The development of Log Aesthetic Patch and its projection onto the plane.

Yee Meng Teh 1, R.U. Gobithaasan 2 *, Kenjiro T. Miura 3, Diya' J. Albayari 4, and Wen Eng Ong 1

1 School of Mathematical Sciences, Universiti Sains Malaysia, Minden 11800, Malaysia; veostly_0405@hotmail.com (Y.M.T.); weneng@usm.my (W.E.O.)

2 Special Interest Group on Modelling & Data Analytics, Faculty of Ocean Engineering Technology and Informatics, University Malaysia Terengganu, 21030 Kuala Nerus, Terengganu, Malaysia; gr@umt.edu.my

3 Graduate School of Science & Technology, Shizuoka University, Hamamatsu, Shizuoka, 432-8561, Japan; miura.kenjiro@shizuoka.ac.jp

4 Mathematics Department, Ibn Rushd National Academy-International section IB®, 17126 Amman, Jordan; diyabayari@gmail.com

* Correspondence: gr@umt.edu.my

Figure 1

Set up

```
kernelcodeHP = "
#define min 0
#define max 1
#define uuu 0
#define vvv 1
#define sign1 1
#define h 0.001
#define n 1402
#define n_j 302


#define inidxdu1 (tu1)
#define inidxdu2 (0)
#define inidxdu3 (tu2)
#define inidxdv1 (0)
#define inidxdv2 (tv1)
```

```
#define inidxdv3 (tv2)

#define inid2xdu21 (dtu1)
#define inid2xdu22 (0)
#define inid2xdu23 (dtu2)
#define inid2xdv21 (0)
#define inid2xdv22 (dtv1)
#define inid2xdv23 (dtv2)
#define inid2xdudv1 (0)
#define inid2xdudv2 (0)
#define inid2xdudv3 (0)

#define inid3xdu31 (d2tu1)
#define inid3xdu32 (0)
#define inid3xdu33 (d2tu2)
#define inid3xdv31 (0)
#define inid3xdv32 (d2tv1)
#define inid3xdv33 (d2tv2)
#define inid3xdu2dv1 (0)
#define inid3xdu2dv2 (0)
#define inid3xdu2dv3 (0)
#define inid3xdudv21 (0)
#define inid3xdudv22 (0)
#define inid3xdudv23 (0)



#define repiniSurNorvec iniSurNorvec(xu1,xu2,tu1,tu2,xv1,xv2,tv1,tv2,normv_C)
#define repiniDSurNvDu
    iniDSurNvDu(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,dnormvdu_C)
#define repiniDSurNvDv
    iniDSurNvDv(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,dnormvdv_C)
#define repiniScalarSurNor
    iniScalarSurNor(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,Scal)
#define repinicoefE inicoefE(xu1,xu2,tu1,tu2,xv1,xv2,tv1,tv2)
#define repiniDEDu
    iniDEDu(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniDEDv
    iniDEDv(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repinicoefF inicoefF(xu1,xu2,tu1,tu2,xv1,xv2,tv1,tv2)
#define repiniDFDu
    iniDFDu(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniDFDv
    iniDFDv(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repinicoefG inicoefG(xu1,xu2,tu1,tu2,xv1,xv2,tv1,tv2)
#define repiniDGDu
    iniDGDu(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
```

```
#define repiniDGDv
    iniDGDv(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repinicoefL
    inicoefL(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repinicoefM
    inicoefM(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repinicoefN
    inicoefN(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniGaussCurva
    iniGaussCurva(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniMeanCurva
    iniMeanCurva(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniMaxPrinci
    iniMaxPrinci(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniMinPrinci
    iniMinPrinci(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniPrinci
    iniPrinci(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)

#define repiniDcoefL_2
    iniDcoefL_2(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
    dtv2,d2tv1,d2tv2,CoeffL)
#define repiniDcoefM_2
    iniDcoefM_2(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
    dtv2,d2tv1,d2tv2,CoeffM)
#define repiniDcoefN_2
    iniDcoefN_2(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
    dtv2,d2tv1,d2tv2,CoeffN)
#define repiniDGaussCurva_2
    iniDGaussCurva_2(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,
    dtv1,dtv2,d2tv1,d2tv2,GCurva)
#define repiniDMeanCurva_2
    iniDMeanCurva_2(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,
    dtv1,dtv2,d2tv1,d2tv2,MCurva)
#define repiniDPrinci_2
    iniDPrinci_2(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
    dtv2,d2tv1,d2tv2,optimum,PCurva)
#define repiniEta
    iniEta(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniMiu
    iniMiu(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDuDs
    iniDuDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDvDs
    iniDvDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDEDs
    iniDEDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
```

```
#define repiniDFDs
    iniDFDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDGDs
    iniDGDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDLDs
    iniDLDs(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2,optimum)
#define repiniDMDs
    iniDMDs(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2,optimum)
#define repiniDNDs
    iniDNDs(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2,optimum)
#define repiniDPrinciDs
    iniDPrinciDs(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
    dtv2,d2tv1,d2tv2,optimum)
#define repiniDvDt
    iniDvDt(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDuDt
    iniDuDt(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDSurDs
    iniDSurDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum,tangv)
#define repiniSurNor iniSurNor(xu1,xu2,tu1,tu2,xv1,xv2,tv1,tv2,Snor_C)
#define repiniBinormalSur
    iniBinormalSur(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    optimum,binormv_C)
#define repiniAlp1
    iniAlp1(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum,Alp1_C)
#define repiniBeta1
    iniBeta1(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2
    ,d2tv1,d2tv2,optimum)


#define opt1 (float xu1,float xu2,float tu1,float
    tu2,float xv1,float xv2,float tv1,float tv2,float normv_C[])
#define opt2 (float xu1,float xu2,float tu1,float tu2,float
    xv1,float xv2,float tv1,float tv2)
#define opt3 (float xu1,float xu2,float tu1,float tu2,float dtu1,float
    dtu2,float xv1,float xv2,float tv1,float tv2,float dtv1,float dtv2)
#define opt4 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float xv1,float xv2,float
    tv1,float tv2,float dtv1,float dtv2,mint optimum)
 #define opt5 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float xv1,float xv2,float tv1,float
    tv2,float dtv1,float dtv2,mint optimum,float tangv_C[])
#define opt6 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float xv1,float xv2,float tv1,float
```

```
    tv2,float dtv1,float dtv2,float dnormv_C[])
#define opt7 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float xv1,float xv2,float
    tv1,float tv2,float dtv1,float dtv2,float Scal[])
#define opt8 (float xu1,float xu2,float tu1,float tu2,float dtu1,float
    dtu2,float d2tu1,float d2tu2,float xv1,float xv2,float tv1,float
    tv2,float dtv1,float dtv2,float d2tv1,float d2tv2,float coeff[])
#define opt9 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float d2tu1,float d2tu2,float xv1,float
    xv2,float tv1,float tv2,float dtv1,float dtv2,float
    d2tv1,float d2tv2,mint optimum,float coeff[])
#define opt10 (float xu1,float xu2,float tu1,float tu2,float dtu1,float
    dtu2,float d2tu1,float d2tu2,float xv1,float xv2,float tv1,float
    tv2,float dtv1,float dtv2,float d2tv1,float d2tv2,mint optimum)
#define opt11 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float xv1,float xv2,float tv1,float
    tv2,float dtv1,float dtv2,mint optimum,float Alph_C[])
#define opt12 (float xu1,float xu2,float tu1,float tu2,float
    xv1,float xv2,float tv1,float tv2,float Snor_C[])
#define opt13 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float xv1,float xv2,float tv1,float
    tv2,float dtv1,float dtv2,mint optimum,float binormv_C[])
#define opt15 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float d2tu1,float d2tu2,float
    xv1,float xv2,float tv1,float tv2,float dtv1,float
    dtv2,float d2tv1,float d2tv2,mint optimum,float x[])



__device__ void iniSurNorvec opt1
{
    normv_C[0] =inidxdu2*inidxdv3-inidxdu3*inidxdv2;
    normv_C[1] =inidxdu3*inidxdv1-inidxdu1*inidxdv3;
    normv_C[2] =inidxdu1*inidxdv2-inidxdu2*inidxdv1;
}
__device__ void iniDSurNvDu opt6
{
    dnormv_C[0]
    =(inid2xdu22*inidxdv3-inid2xdu23*inidxdv2)+(inidxdu2*inid2xdudv3-
    inidxdu3*inid2xdudv2);
    dnormv_C[1]
    =(inid2xdu23*inidxdv1-inid2xdu21*inidxdv3)+(inidxdu3*inid2xdudv1-
    inidxdu1*inid2xdudv3);
    dnormv_C[2]
    =(inid2xdu21*inidxdv2-inid2xdu22*inidxdv1)+(inidxdu1*inid2xdudv2-
    inidxdu2*inid2xdudv1);
```

```
}
__device__ void iniDSurNvDv opt6
{
    dnormv_C[0]
     =(inid2xdudv2*inidxdv3-inid2xdudv3*inidxdv2)+(inidxdu2*inid2xdv23-
     inidxdu3*inid2xdv22);
    dnormv_C[1]
     =(inid2xdudv3*inidxdv1-inid2xdudv1*inidxdv3)+(inidxdu3*inid2xdv21-
     inidxdu1*inid2xdv23);
    dnormv_C[2]
     =(inid2xdudv1*inidxdv2-inid2xdudv2*inidxdv1)+(inidxdu1*inid2xdv22-
     inidxdu2*inid2xdv21);
}

__device__ void iniScalarSurNor opt7
{
float ScalSNor,DScalSNorDu,DScalSNorDv;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];

repiniSurNorvec;
repiniDSurNvDu;
repiniDSurNvDv;


ScalSNor =
     sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
DScalSNorDu
     =
     (normv_C[0]*dnormvdu_C[0]+normv_C[1]*dnormvdu_C[1]+normv_C[2]*dnormvdu_C
     [2])/ScalSNor;
DScalSNorDv =
     (normv_C[0]*dnormvdv_C[0]+normv_C[1]*dnormvdv_C[1]+normv_C[2]*dnormvdv_C
     [2])/ScalSNor;


Scal[0] = ScalSNor;
Scal[1] = DScalSNorDu;
Scal[2] = DScalSNorDv;
}

__device__ float inicoefE opt2
{
    return inidxdu1*inidxdu1+inidxdu2*inidxdu2+inidxdu3*inidxdu3;
}
__device__ float iniDEDu opt3
```

```
{
    return 2*(inidxdu1*inid2xdu21+inidxdu2*inid2xdu22+inidxdu3*inid2xdu23);
}
__device__ float iniDEDv opt3
{
    return 2*(inidxdu1*inid2xdudv1+inidxdu2*inid2xdudv2+inidxdu3*inid2xdudv3);
}


__device__ float inicoefF opt2
{
    return (inidxdu1*inidxdv1+inidxdu2*inidxdv2+inidxdu3*inidxdv3);
}
__device__ float iniDFDu opt3
{
    return inidxdv1*inid2xdu21+inidxdv2*inid2xdu22+inidxdv3*inid2xdu23
     + inidxdu1*inid2xdudv1+inidxdu2*inid2xdudv2+inidxdu3*inid2xdudv3;
}
__device__ float iniDFDv opt3
{
    return
     inidxdv1*inid2xdudv1+inidxdv2*inid2xdudv2+inidxdv3*inid2xdudv3
     + inidxdu1*inid2xdv21+inidxdu2*inid2xdv22+inidxdu3*inid2xdv23;
}

__device__ float inicoefG opt2
{
    return inidxdv1*inidxdv1+inidxdv2*inidxdv2+inidxdv3*inidxdv3;
}
__device__ float iniDGDu opt3
{
    return
     2*(inidxdv1*inid2xdudv1+inidxdv2*inid2xdudv2+inidxdv3*inid2xdudv3);
}
__device__ float iniDGDv opt3
{
    return
     2*(inidxdv1*inid2xdv21+inidxdv2*inid2xdv22+inidxdv3*inid2xdv23);
}

__device__ float inicoefL opt3
{
float normv_C[3];
repiniSurNorvec;

    return
     (normv_C[0]*inid2xdu21+normv_C[1]*inid2xdu22+normv_C[2]*inid2xdu23)/sqrt
```

```
    (normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float inicoefM opt3
{
float normv_C[3];
repiniSurNorvec;

    return
    (normv_C[0]*inid2xdudv1+normv_C[1]*inid2xdudv2+normv_C[2]*inid2xdudv3)/
    sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float inicoefN opt3
{
float normv_C[3];
repiniSurNorvec;

    return
    (normv_C[0]*inid2xdv21+normv_C[1]*inid2xdv22+normv_C[2]*inid2xdv23)/sqrt
    (normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float iniGaussCurva opt3
{
    return
    (repinicoefL*repinicoefN-pow(repinicoefM,2))/(repinicoefE*repinicoefG-
    pow(repinicoefF,2));
}
__device__ float iniMeanCurva opt3
{
    return
    0.5*(repinicoefE*repinicoefN+repinicoefG*repinicoefL-2*repinicoefF*
    repinicoefM)/(repinicoefE*repinicoefG-pow(repinicoefF,2));
}
__device__ float iniMaxPrinci opt3
{
    return repiniMeanCurva+sqrt(pow(repiniMeanCurva,2)-repiniGaussCurva);
}

__device__ float iniMinPrinci opt3
{
    return repiniMeanCurva-sqrt(pow(repiniMeanCurva,2)-repiniGaussCurva);
}
__device__ float iniPrinci opt4
{
    if(optimum==max)
    {return repiniMaxPrinci;}
    else if(optimum==min)
    {return repiniMinPrinci;}
```

```
    else
    {return 0;}
}
__device__ void iniDcoefL_2 opt8
{

float CoeL,DCoeLDu,DCoeLDv;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float Scal[3];

repiniSurNorvec;
repiniDSurNvDu;
repiniDSurNvDv;
repiniScalarSurNor;

CoeL =
     (normv_C[0]*inid2xdu21+normv_C[1]*inid2xdu22+normv_C[2]*inid2xdu23)/Scal[0];
DCoeLDu
     =
     ((dnormvdu_C[0]*inid2xdu21+dnormvdu_C[1]*inid2xdu22+dnormvdu_C[2]*
     inid2xdu23)+(normv_C[0]*inid3xdu31+normv_C[1]*inid3xdu32+normv_C[2]*
     inid3xdu33)-Scal[1]*CoeL)/Scal[0];
DCoeLDv =
     ((dnormvdv_C[0]*inid2xdu21+dnormvdv_C[1]*inid2xdu22+dnormvdv_C[2]*
     inid2xdu23)+(normv_C[0]*inid3xdu2dv1+normv_C[1]*inid3xdu2dv2+normv_C[2]*
     inid3xdu2dv3)-Scal[2]*CoeL)/Scal[0];

coeff[0] = CoeL;
coeff[1] = DCoeLDu;
coeff[2] = DCoeLDv;

}
__device__ void iniDcoefM_2 opt8
{
float CoeM,DCoeMDu,DCoeMDv;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float Scal[3];
repiniSurNorvec;
repiniDSurNvDu;
repiniDSurNvDv;
repiniScalarSurNor;

CoeM =
```

```
     (normv_C[0]*inid2xdudv1+normv_C[1]*inid2xdudv2+normv_C[2]*inid2xdudv3)/
     Scal[0];
DCoeMDu =
     ((dnormvdu_C[0]*inid2xdudv1+dnormvdu_C[1]*inid2xdudv2+dnormvdu_C[2]*
     inid2xdudv3)+(normv_C[0]*inid3xdu2dv1+normv_C[1]*inid3xdu2dv2+normv_C[2]
     *inid3xdu2dv3)-Scal[1]*CoeM)/Scal[0];
DCoeMDv =
     ((dnormvdv_C[0]*inid2xdudv1+dnormvdv_C[1]*inid2xdudv2+dnormvdv_C[2]*
     inid2xdudv3)+(normv_C[0]*inid3xdudv21+normv_C[1]*inid3xdudv22+normv_C[2]
     *inid3xdudv23)-Scal[2]*CoeM)/Scal[0];


coeff[0] = CoeM;
coeff[1] = DCoeMDu;
coeff[2] = DCoeMDv;
}
__device__ void iniDcoefN_2 opt8
{
float CoeN,DCoeNDu,DCoeNDv;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float Scal[3];
repiniSurNorvec;
repiniDSurNvDu;
repiniDSurNvDv;
repiniScalarSurNor;

CoeN =
     (normv_C[0]*inid2xdv21+normv_C[1]*inid2xdv22+normv_C[2]*inid2xdv23)/Scal[0];
DCoeNDu
     =
     ((dnormvdu_C[0]*inid2xdv21+dnormvdu_C[1]*inid2xdv22+dnormvdu_C[2]*
     inid2xdv23)+(normv_C[0]*inid3xdudv21+normv_C[1]*inid3xdudv22+normv_C[2]*
     inid3xdudv23)-Scal[1]*CoeN)/Scal[0];
DCoeNDv =
     ((dnormvdv_C[0]*inid2xdv21+dnormvdv_C[1]*inid2xdv22+dnormvdv_C[2]*
     inid2xdv23)+(normv_C[0]*inid3xdv31+normv_C[1]*inid3xdv32+normv_C[2]*
     inid3xdv33)-Scal[2]*CoeN)/Scal[0];

coeff[0] = CoeN;
coeff[1] = DCoeNDu;
coeff[2] = DCoeNDv;

}
__device__ void iniDGaussCurva_2 opt8
{
```

```
float GCurv,DGCurvDu,DGCurvDv;
float CoeffL[3];
float CoeffM[3];
float CoeffN[3];
repiniDcoefL_2;
repiniDcoefM_2;
repiniDcoefN_2;

GCurv=
    (CoeffL[0]*CoeffN[0]-pow(CoeffM[0],2))/(repinicoefE*repinicoefG-pow(
    repinicoefF,2));
DGCurvDu =
    ((pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repinicoefG*repiniDEDu-2*
    repinicoefF*repiniDFDu+repinicoefE*repiniDGDu)+(-pow(repinicoefF,2)+
    repinicoefE*repinicoefG)*(CoeffN[0]*CoeffL[1]-2*CoeffM[0]*CoeffM[1]+
    CoeffL[0]*CoeffN[1]))/(pow(pow(repinicoefF,2)-repinicoefE*repinicoefG,2)
    );
DGCurvDv
    =((pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repinicoefG*repiniDEDv-2*
    repinicoefF*repiniDFDv+repinicoefE*repiniDGDv)+(-pow(repinicoefF,2)+
    repinicoefE*repinicoefG)*(CoeffN[0]*CoeffL[2]-2*CoeffM[0]*CoeffM[2]+
    CoeffL[0]*CoeffN[2]))/(pow(pow(repinicoefF,2)-repinicoefE*repinicoefG,2)
    );


coeff[0] = GCurv;
coeff[1] = DGCurvDu;
coeff[2] = DGCurvDv;

}
__device__ void iniDMeanCurva_2 opt8
{
float MCurv,DMCurvDu,DMCurvDv;
float CoeffL[3];
float CoeffM[3];
float CoeffN[3];
repiniDcoefL_2;
repiniDcoefM_2;
repiniDcoefN_2;

MCurv =
    0.5*(repinicoefE*CoeffN[0]+repinicoefG*CoeffL[0]-2*repinicoefF*CoeffM[0]
    )/(repinicoefE*repinicoefG-pow(repinicoefF,2));
DMCurvDu =
    0.5*(-(repinicoefG*CoeffL[0]-2*repinicoefF*CoeffM[0]+repinicoefE*CoeffN[
    0])*(repinicoefG*repiniDEDu-2*repinicoefF*repiniDFDu+repinicoefE*
    repiniDGDu)+(-pow(repinicoefF,2)+repinicoefE*repinicoefG)*(CoeffN[0]*
```

```
    repiniDEDu-2*CoeffM[0]*repiniDFDu+CoeffL[0]*repiniDGDu+repinicoefG*
    CoeffL[1]-2*repinicoefF*CoeffM[1]+repinicoefE*CoeffN[1]))/(pow(pow(
    repinicoefF,2)-repinicoefE*repinicoefG,2));
DMCurvDv =
    0.5*(-(repinicoefG*CoeffL[0]-2*repinicoefF*CoeffM[0]+repinicoefE*CoeffN[
    0])*(repinicoefG*repiniDEDv-2*repinicoefF*repiniDFDv+repinicoefE*
    repiniDGDv)+(-pow(repinicoefF,2)+repinicoefE*repinicoefG)*(CoeffN[0]*
    repiniDEDv-2*CoeffM[0]*repiniDFDv+CoeffL[0]*repiniDGDv+repinicoefG*
    CoeffL[2]-2*repinicoefF*CoeffM[2]+repinicoefE*CoeffN[2]))/(pow(pow(
    repinicoefF,2)-repinicoefE*repinicoefG,2));



coeff[0] = MCurv;
coeff[1] = DMCurvDu;
coeff[2] = DMCurvDv;
}
__device__ void iniDPrinci_2 opt9
{
float Princip,DPrincipDu,DPrincipDv;
float GCurva[3];
float MCurva[3];
repiniDGaussCurva_2;
repiniDMeanCurva_2;

if(optimum==max)
{Princip = MCurva[0]+sqrt(pow(MCurva[0],2)-GCurva[0]);
DPrincipDu =
    MCurva[1]+(2*MCurva[0]*MCurva[1]-GCurva[1])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
DPrincipDv =
    MCurva[2]+(2*MCurva[0]*MCurva[2]-GCurva[2])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));

}
else if(optimum==min)
{Princip = MCurva[0]-sqrt(pow(MCurva[0],2)-GCurva[0]);
DPrincipDu =
    MCurva[1]-(2*MCurva[0]*MCurva[1]-GCurva[1])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
DPrincipDv =
    MCurva[2]-(2*MCurva[0]*MCurva[2]-GCurva[2])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));

}
else
{Princip = 0;
```

```
DPrincipDu = 0;
DPrincipDv = 0;
}

coeff[0] = Princip;
coeff[1] = DPrincipDu;
coeff[2] = DPrincipDv;

}

__device__ float iniEta opt4
{
    return
    sign1/sqrt(repinicoefE*pow(repinicoefM-repiniPrinci*repinicoefF,2)-2*
    repinicoefF*(repinicoefM-repiniPrinci*repinicoefF)*(repinicoefL-
    repiniPrinci*repinicoefE)+repinicoefG*pow(repinicoefL-repiniPrinci*
    repinicoefE,2));
}
__device__ float iniMiu opt4
{
    return
    sign1/sqrt(repinicoefE*pow(repinicoefN-repiniPrinci*repinicoefG,2)-2*
    repinicoefF*(repinicoefM-repiniPrinci*repinicoefF)*(repinicoefN-
    repiniPrinci*repinicoefG)+repinicoefG*pow(repinicoefM-repiniPrinci*
    repinicoefF,2));
}
__device__ float iniDuDs opt4
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-
    repiniPrinci*repinicoefG))
    {return repiniEta*(repinicoefM-repiniPrinci*repinicoefF);}
else
    {return repiniMiu*(repinicoefN-repiniPrinci*repinicoefG);}
}
__device__ float iniDvDs opt4
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
    repinicoefG))
    {return -repiniEta*(repinicoefL-repiniPrinci*repinicoefE);}
else
    {return -repiniMiu*(repinicoefM-repiniPrinci*repinicoefF);}
}


__device__ float iniDEDs opt4
{
    return repiniDEDu*repiniDuDs+repiniDEDv*repiniDvDs;
```

```
}
__device__ float iniDFDs opt4
{
    return repiniDFDu*repiniDuDs+repiniDFDv*repiniDvDs;
}
__device__ float iniDGDs opt4
{
    return repiniDGDu*repiniDuDs+repiniDGDv*repiniDvDs;
}
__device__ float iniDLDs opt10
{
float CoeffL[3];
repiniDcoefL_2;

    return CoeffL[1]*repiniDuDs+CoeffL[2]*repiniDvDs;
}
__device__ float iniDMDs opt10
{
float CoeffM[3];
repiniDcoefM_2;

    return CoeffM[1]*repiniDuDs+CoeffM[2]*repiniDvDs;
}
__device__ float iniDNDs opt10
{
float CoeffN[3];
repiniDcoefN_2;

    return CoeffN[1]*repiniDuDs+CoeffN[2]*repiniDvDs;
}
__device__ float iniDPrinciDs opt10
{
float PCurva[3];
repiniDPrinci_2;

    return PCurva[1]*repiniDuDs+PCurva[2]*repiniDvDs;
}
__device__ float iniDuDt opt4
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
     repinicoefG))
    {return (repinicoefM-repiniPrinci*repinicoefF);}
else
    {return (repinicoefN-repiniPrinci*repinicoefG);}
}
__device__ float iniDvDt opt4
{
```

```
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
     repinicoefG))
     {return (repinicoefL-repiniPrinci*repinicoefE);}
else
     {return (repinicoefM-repiniPrinci*repinicoefF);}
}
__device__ void iniDSurDs opt5
{
     tangv_C[0] =inidxdu1*repiniDuDs+inidxdv1*repiniDvDs;
     tangv_C[1] =inidxdu2*repiniDuDs+inidxdv2*repiniDvDs;
     tangv_C[2] =inidxdu3*repiniDuDs+inidxdv3*repiniDvDs;
}

__device__ void iniAlp1 opt11
{
     Alph_C[0]=(inid2xdu21*pow(repiniDuDs,2)+2*inid2xdudv1*repiniDuDs*
     repiniDvDs+inid2xdv21*pow(repiniDvDs,2));

     Alph_C[1]=(inid2xdu22*pow(repiniDuDs,2)+2*inid2xdudv2*repiniDuDs*
     repiniDvDs+inid2xdv22*pow(repiniDvDs,2));

     Alph_C[2]=(inid2xdu23*pow(repiniDuDs,2)+2*inid2xdudv3*repiniDuDs*
     repiniDvDs+inid2xdv23*pow(repiniDvDs,2));
}
__device__ void iniSurNor opt12
{
     float normv_C[3];
     repiniSurNorvec;
     Snor_C[0]
     =normv_C[0]/sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*
     normv_C[2]);
     Snor_C[1]
     =normv_C[1]/sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*
     normv_C[2]);
     Snor_C[2]
     =normv_C[2]/sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*
     normv_C[2]);
}

__device__ void iniBinormalSur opt13
{
     float Snor_C[3];
     float tangv[3];
     repiniSurNor;
     repiniDSurDs;

     binormv_C[0] =Snor_C[1]*tangv[2] -Snor_C[2]*tangv[1];
```

```
    binormv_C[1] =Snor_C[2]*tangv[0] -Snor_C[0]*tangv[2];
    binormv_C[2] =Snor_C[0]*tangv[1] -Snor_C[1]*tangv[0];
}
__device__ float iniBeta1 opt10
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
    repinicoefG))
    {return
    (-(repiniDLDs-repiniDPrinciDs*repinicoefE-repiniPrinci*repiniDEDs)*
    repiniDuDs-(repiniDMDs-repiniDPrinciDs*repinicoefF-repiniPrinci*
    repiniDFDs)*repiniDvDs);}
else
    {return
    (-(repiniDMDs-repiniDPrinciDs*repinicoefF-repiniPrinci*repiniDFDs)*
    repiniDuDs-(repiniDNDs-repiniDPrinciDs*repinicoefG-repiniPrinci*
    repiniDGDs)*repiniDvDs);}
}

__device__ void LUDecomposition3x3 opt15
{
    mint rc=3;
    float binormv_C[3];
    float Alp1_C[3];
    repiniBinormalSur;
    repiniAlp1;
    float matrixA[3][3]={

    {repinicoefE,repinicoefF,-(binormv_C[0]*inidxdu1+binormv_C[1]*inidxdu2+
    binormv_C[2]*inidxdu3)},

    {repinicoefF,repinicoefG,-(binormv_C[0]*inidxdv1+binormv_C[1]*inidxdv2+
    binormv_C[2]*inidxdv3)},
    {repiniDvDt,repiniDuDt,0}};

    float
    matrixB[3]={-(Alp1_C[0]*inidxdu1+Alp1_C[1]*inidxdu2+Alp1_C[2]*inidxdu3),
    -(Alp1_C[0]*inidxdv1+Alp1_C[1]*inidxdv2+Alp1_C[2]*inidxdv3),repiniBeta1}
    ;
    float Lower[3][3];
    float Upper[3][3];
    float y[3];
    float sum;
    for(mint ii=0; ii<rc; ii++)
            {
                for(mint jj=0; jj<rc; jj++)
                {
                    sum = 0;
```

```
                    if(ii==jj)
                    {
                    Lower[ii][jj]=1;
                    for (mint kk = 0; kk < ii; kk++)
                        {sum += Lower[ii][kk] * Upper[kk][jj];}
                    Upper[ii][jj] = matrixA[ii][jj] - sum;
                    }
                    else if(ii < jj)
                    {
                    Lower[ii][jj]=0;
                    for (mint kk = 0; kk < ii; kk++)
                        {sum += Lower[ii][kk] * Upper[kk][jj];}
                    Upper[ii][jj] = matrixA[ii][jj] - sum;
                    }
                    else
                    {
                    Upper[ii][jj]=0;
                    for (mint kk = 0; kk < jj; kk++)
                        {sum += Lower[ii][kk] * Upper[kk][jj];}
                    Lower[ii][jj] =(matrixA[ii][jj] - sum)/Upper[jj][jj];
                    }
                }
            }
    for (mint ii = 0; ii < rc; ii++)
    {
        sum = 0;
        for (mint jj = 0; jj < ii; jj++)
            {sum += Lower[ii][jj] * y[jj];}
        y[ii] = matrixB[ii] - sum;
    }
    for (mint ii = rc - 1; ii >= 0; ii--)
    {
        sum = 0;
        for (mint jj = ii + 1; jj < rc; jj++)
            {sum += Upper[ii][jj] * x[jj];}
        x[ii] = (y[ii] - sum)/Upper[ii][ii];
    }
}


__device__ float RK4_Init(float x1, float t1, float SSize)

{
float k1,k2,k3,k4;

k1=SSize*x1;
k2=SSize*(x1+0.5*SSize*t1);
```

```
k3=SSize*(x1+0.5*SSize*t1);
k4=SSize*(x1+SSize*t1);

return (1.0/6.0)*(k1+2*k2+2*k3+k4);
}

__device__ void RK4_func_LoC( float xu1, float xu2,
    float tu1, float tu2, float dtu1, float dtu2, float xv1,
    float xv2, float tv1, float tv2, float dtv1, float dtv2,
    mint optimum, float& increment_u, float& increment_v)
{

float u_k1,u_k2,u_k3,u_k4;
float v_k1,v_k2,v_k3,v_k4;
float newxu1,newxu2,newtu1,newtu2,newdtu1,newdtu2;
float newxv1,newxv2,newtv1,newtv2,newdtv1,newdtv2;


u_k1=h*iniDuDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum);
v_k1=h*iniDvDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum);

newxu1=xu1+0.5*u_k1*tu1;
newxu2=xu2+0.5*u_k1*tu2;
newtu1=tu1+0.5*u_k1*dtu1;
newtu2=tu2+0.5*u_k1*dtu2;
newdtu1=0;
newdtu2=2;
newxv1=xv1+0.5*v_k1*tv1;
newxv2=xv2+0.5*v_k1*tv2;
newtv1=tv1+0.5*v_k1*dtv1;
newtv2=tv2+0.5*v_k1*dtv2;
newdtv1=0;
newdtv2=-2;
u_k2=h*iniDuDs(newxu1,newxu2,newtu1,newtu2,newdtu1,newdtu2,newxv1,newxv2,newtv1
    ,newtv2,newdtv1,newdtv2,optimum);
v_k2=h*iniDvDs(newxu1,newxu2,newtu1,newtu2,newdtu1,newdtu2,newxv1,newxv2
    ,newtv1,newtv2,newdtv1,newdtv2,optimum);

newxu1=xu1+0.5*u_k2*tu1;
newxu2=xu2+0.5*u_k2*tu2;
newtu1=tu1+0.5*u_k2*dtu1;
newtu2=tu2+0.5*u_k2*dtu2;
newdtu1=0;
newdtu2=2;
newxv1=xv1+0.5*v_k2*tv1;
newxv2=xv2+0.5*v_k2*tv2;
newtv1=tv1+0.5*v_k2*dtv1;
```

```
newtv2=tv2+0.5*v_k2*dtv2;
newdtv1=0;
newdtv2=-2;

u_k3=h*iniDuDs(newxu1,newxu2,newtu1,newtu2,newdtu1,newdtu2,newxv1,newxv2,newtv1
     ,newtv2,newdtv1,newdtv2,optimum);
v_k3=h*iniDvDs(newxu1,newxu2,newtu1,newtu2,newdtu1,newdtu2,newxv1,newxv2
     ,newtv1,newtv2,newdtv1,newdtv2,optimum);

newxu1=xu1+u_k3*tu1;
newxu2=xu2+u_k3*tu2;
newtu1=tu1+u_k3*dtu1;
newtu2=tu2+u_k3*dtu2;
newdtu1=0;
newdtu2=2;
newxv1=xv1+v_k3*tv1;
newxv2=xv2+v_k3*tv2;
newtv1=tv1+v_k3*dtv1;
newtv2=tv2+v_k3*dtv2;
newdtv1=0;
newdtv2=-2;

u_k4=h*iniDuDs(newxu1,newxu2,newtu1,newtu2,newdtu1,newdtu2,newxv1,newxv2,newtv1
     ,newtv2,newdtv1,newdtv2,optimum);
v_k4=h*iniDvDs(newxu1,newxu2,newtu1,newtu2,newdtu1,newdtu2,newxv1,newxv2
     ,newtv1,newtv2,newdtv1,newdtv2,optimum);

increment_u=(1.0/6.0)*(u_k1+2*u_k2+2*u_k3+u_k4);
increment_v=(1.0/6.0)*(v_k1+2*v_k2+2*v_k3+v_k4);

}

__device__ void ReturnGeoCur( int k, float x[],
     float y[], float z[], float GeoCur[], mint optimum)
{
float iniSSize_u,iniSSize_v,SSize_u,SSize_v;

float xu1, xu2, tu1, tu2, dtu1, dtu2, d2tu1, d2tu2;
float xv1, xv2, tv1, tv2, dtv1, dtv2, d2tv1, d2tv2;
float u_s,v_s;
float coef[3];

u_s=0;
v_s=0;
xu1=0;
xu2=0;
tu1=1;
```

```
tu2=0;
dtu1=0;
dtu2=2;
d2tu1=0;
d2tu2=0;
xv1=0;
xv2=0;
tv1=1;
tv2=0;
dtv1=0;
dtv2=-2;
d2tv1=0;
d2tv2=0;

if(optimum==min)
{
    for(mint i=1;i<=k;i++)
    {

    RK4_func_LoC(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,max,
    SSize_u,SSize_v);
        u_s+=SSize_u;
        v_s+=SSize_v;
        xu1+=RK4_Init(tu1,dtu1,SSize_u);
        xu2+=RK4_Init(tu2,dtu2,SSize_u);
        tu1+=RK4_Init(dtu1,d2tu1,SSize_u);
        tu2+=RK4_Init(dtu2,d2tu2,SSize_u);
        dtu1=0;
        dtu2=2;
        d2tu1=0;
        d2tu2=0;
        xv1+=RK4_Init(tv1,dtv1,SSize_v);
        xv2+=RK4_Init(tv2,dtv2,SSize_v);
        tv1+=RK4_Init(dtv1,d2tv1,SSize_v);
        tv2+=RK4_Init(dtv2,d2tv2,SSize_v);
        dtv1=0;
        dtv2=-2;
        d2tv1=0;
        d2tv2=0;
    }
}
else if(optimum==max)
{
    for(mint i=1;i<=k;i++)
    {
```

```
    RK4_func_LoC(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,min,
    SSize_u,SSize_v);
        u_s+=SSize_u;
        v_s+=SSize_v;
        xu1+=RK4_Init(tu1,dtu1,SSize_u);
        xu2+=RK4_Init(tu2,dtu2,SSize_u);
        tu1+=RK4_Init(dtu1,d2tu1,SSize_u);
        tu2+=RK4_Init(dtu2,d2tu2,SSize_u);
        dtu1=0;
        dtu2=2;
        d2tu1=0;
        d2tu2=0;
        xv1+=RK4_Init(tv1,dtv1,SSize_v);
        xv2+=RK4_Init(tv2,dtv2,SSize_v);
        tv1+=RK4_Init(dtv1,d2tv1,SSize_v);
        tv2+=RK4_Init(dtv2,d2tv2,SSize_v);
        dtv1=0;
        dtv2=-2;
        d2tv1=0;
        d2tv2=0;
    }
}
x[0]=xu1;
y[0]=xv1;
z[0]=xu2+xv2;
LUDecomposition3x3(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
    dtv2,d2tv1,d2tv2,optimum,coef);
GeoCur[0]=coef[2];

for(mint j=1;j<n;j++)
{

    RK4_func_LoC(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum
    ,SSize_u,SSize_v);
    u_s+=SSize_u;
    v_s+=SSize_v;
    xu1+=RK4_Init(tu1,dtu1,SSize_u);
    xu2+=RK4_Init(tu2,dtu2,SSize_u);
    tu1+=RK4_Init(dtu1,d2tu1,SSize_u);
    tu2+=RK4_Init(dtu2,d2tu2,SSize_u);
    dtu1=0;
    dtu2=2;
    d2tu1=0;
    d2tu2=0;

    xv1+=RK4_Init(tv1,dtv1,SSize_v);
    xv2+=RK4_Init(tv2,dtv2,SSize_v);
```

```
    tv1+=RK4_Init(dtv1,d2tv1,SSize_v);
    tv2+=RK4_Init(dtv2,d2tv2,SSize_v);
    dtv1=0;
    dtv2=-2;
    d2tv1=0;
    d2tv2=0;

    x[j]=xu1;
    y[j]=xv1;
    z[j]=xu2+xv2;

    LUDecomposition3x3(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2
    ,dtv1,dtv2,d2tv1,d2tv2,optimum,coef);
    GeoCur[j]=coef[2];

}

}

__device__ void TwoDimensionCurve( float init_x, float init_y,
    float GeoCur[], float x[], float y[], float tangent1[], float
    tangent2[], float normal1[], float normal2[], mint optimum)
{//use to compute plane curve coordinates from given geodesic curvature
float DummyGeoCur1,DummyGeoCur2,DGCDs;
float x1, x2;
float t1, t2, dt1, dt2, d2t1, d2t2;
float n1, n2, dn1, dn2, d2n1, d2n2;

DummyGeoCur1=GeoCur[0];
DummyGeoCur2=GeoCur[1];
DGCDs=(DummyGeoCur2-DummyGeoCur1)/h;
x1=init_x;
x2=init_y;

if (optimum==max)//when curve render in maximum principle direction, the
    tangent is {-1,0,0} at (-x)(-y) plane. We cross {-1,0,0}x{0,0,1}
    then we get normal is {0,1,0}. Then we flip the curve from (-x)(-y)
    plane to (-x)(y) plane we get tangent is {-1,0,0} and normal is
    {0,-1,0}. Finally we flip the curve from (-x)(y) plane to (x)(y)
    plane then we have tangent {1,0,0} and normal is {0,-1,0}. Then
    we reverse the normal={0,1,0} so that we can cut out the pieces
{
    t1=1.0;
    t2=0.0;
    n1=0.0;
    n2=1.0;
}
```

```
else if(optimum==min)//when curve render in minimum principle direction, the
    tangent is {0,-1,0} at (-x)(-y) plane. We cross {0,-1,0}x{0,0,1}
    then we get normal is {-1,0,0}. Then we flip the curve from (-x)(-y)
    plane to (-x)(y) plane we get tangent is {0,1,0} and normal is
    {-1,0,0}. Finally we flip the curve from (-x)(y) plane to (x)(y)
    plane then we have tangent {0,1,0} and normal is {1,0,0}. Then
    we reverse the normal={0,1,0} so that we can cut out the pieces
{
    t1=0.0;
    t2=1.0;
    n1=-1.0;
    n2=0.0;
}
dt1=DummyGeoCur1*n1;
dt2=DummyGeoCur1*n2;
dn1=-DummyGeoCur1*t1;
dn2=-DummyGeoCur1*t2;
d2t1=DummyGeoCur1*dn1+DGCDs*n1;
d2t2=DummyGeoCur1*dn2+DGCDs*n2;
d2n1=-DummyGeoCur1*dt1-DGCDs*t1;
d2n2=-DummyGeoCur1*dt2-DGCDs*t2;

x[0]=x1;
y[0]=x2;
tangent1[0]=t1;
tangent2[0]=t2;
normal1[0]=n1;
normal2[0]=n2;

for(int j=1;j<n;j++)
{

DummyGeoCur1=GeoCur[j];
DummyGeoCur2=GeoCur[j+1];
DGCDs=(DummyGeoCur2-DummyGeoCur1)/h;
x1=x1+RK4_Init(t1,dt1,h);
x2=x2+RK4_Init(t2,dt2,h);
t1=t1+RK4_Init(dt1,d2t1,h);
t2=t2+RK4_Init(dt2,d2t2,h);
n1=n1+RK4_Init(dn1,d2n1,h);
n2=n2+RK4_Init(dn2,d2n2,h);

dt1=DummyGeoCur1*n1;
dt2=DummyGeoCur1*n2;
dn1=-DummyGeoCur1*t1;
dn2=-DummyGeoCur1*t2;
d2t1=DummyGeoCur1*dn1+DGCDs*n1;
```

```
d2t2=DummyGeoCur1*dn2+DGCDs*n2;
d2n1=-DummyGeoCur1*dt1-DGCDs*t1;
d2n2=-DummyGeoCur1*dt2-DGCDs*t2;


x[j]=x1;
y[j]=x2;
tangent1[j]=t1;
tangent2[j]=t2;
normal1[j]=n1;
normal2[j]=n2;
}

}
__device__ void TwoDimensionCurve2( float init_x, float init_y,
     float GeoCur[], float x[], float y[], float tangent1[], float
     tangent2[], float normal1[], float normal2[], mint optimum)
{//use to compute plane curve coordinates from given geodesic curvature
float DummyGeoCur1,DummyGeoCur2,DGCDs;
float x1, x2;
float t1, t2, dt1, dt2, d2t1, d2t2;
float n1, n2, dn1, dn2, d2n1, d2n2;

DummyGeoCur1=GeoCur[0];
DummyGeoCur2=GeoCur[1];
DGCDs=(DummyGeoCur2-DummyGeoCur1)/h;
x1=init_x;
x2=init_y;

if (optimum==max)//when curve render in maximum principle direction, the
     tangent is {-1,0,0} at (-x)(-y) plane. We cross {-1,0,0}x{0,0,1}
     then we get normal is {0,1,0}. Then we flip the curve from (-x)(-y)
     plane to (-x)(y) plane we get tangent is {-1,0,0} and normal
     is {0,-1,0}. Finally we flip the curve from (-x)(y) plane to
     (x)(y) plane then we have tangent {1,0,0} and normal is {0,-1,0}.
{
    t1=1.0;
    t2=0.0;
    n1=0.0;
    n2=-1.0;
}
else if(optimum==min)//when curve render in minimum principle direction, the
     tangent is {0,-1,0} at (-x)(-y) plane. We cross {0,-1,0}x{0,0,1}
     then we get normal is {-1,0,0}. Then we flip the curve from
     (-x)(-y) plane to (-x)(y) plane we get tangent is {0,1,0} and
     normal is {-1,0,0}. Finally we flip the curve from (-x)(y) plane
     to (x)(y) plane then we have tangent {0,1,0} and normal is {1,0,0}.
```

```
{
    t1=0.0;
    t2=1.0;
    n1=1.0;
    n2=0.0;
}
dt1=DummyGeoCur1*n1;
dt2=DummyGeoCur1*n2;
dn1=-DummyGeoCur1*t1;
dn2=-DummyGeoCur1*t2;
d2t1=DummyGeoCur1*dn1+DGCDs*n1;
d2t2=DummyGeoCur1*dn2+DGCDs*n2;
d2n1=-DummyGeoCur1*dt1-DGCDs*t1;
d2n2=-DummyGeoCur1*dt2-DGCDs*t2;

x[0]=x1;
y[0]=x2;
tangent1[0]=t1;
tangent2[0]=t2;
normal1[0]=n1;
normal2[0]=n2;

for(int j=1;j<n;j++)
{

DummyGeoCur1=GeoCur[j];
DummyGeoCur2=GeoCur[j+1];
DGCDs=(DummyGeoCur2-DummyGeoCur1)/h;
x1=x1+RK4_Init(t1,dt1,h);
x2=x2+RK4_Init(t2,dt2,h);
t1=t1+RK4_Init(dt1,d2t1,h);
t2=t2+RK4_Init(dt2,d2t2,h);
n1=n1+RK4_Init(dn1,d2n1,h);
n2=n2+RK4_Init(dn2,d2n2,h);

dt1=DummyGeoCur1*n1;
dt2=DummyGeoCur1*n2;
dn1=-DummyGeoCur1*t1;
dn2=-DummyGeoCur1*t2;
d2t1=DummyGeoCur1*dn1+DGCDs*n1;
d2t2=DummyGeoCur1*dn2+DGCDs*n2;
d2n1=-DummyGeoCur1*dt1-DGCDs*t1;
d2n2=-DummyGeoCur1*dt2-DGCDs*t2;


x[j]=x1;
y[j]=x2;
```

```
tangent1[j]=t1;
tangent2[j]=t2;
normal1[j]=n1;
normal2[j]=n2;
}

}

__device__ void ObtainSmallerPiece( mint m, mint max_size, float x1[],
    float y1[], float z1[], float x2[], float y2[], float z2[],
    float x_2d[], float y_2d[], float t1_2d[], float t2_2d[], float
    n1_2d[], float n2_2d[], float& resx, float& resy, mint optimum)
{
//assume that two curves are genarated in
    maximum principle direction (x1,y1,z1) and (x2,y2,z2)
//this function first is to compute the intesection points between
    every LoCs in Minimum Principle direction and two LoCs in
    maximum principle direction, and also geodesic cruvature
    of each point (LoC in minimum principle direction)
//then use the computed geodesic curvature to compute plane curve
int min_i, min_j, max_j;
float dif1, dif2;
float min_dif1=5;
float min_dif2=5;

float dummy_x_2d, dummy_t1_2d, dummy_dt1_2d,
    dummy_d2t1_2d, dummy_n1_2d, dummy_dn1_2d, dummy_d2n1_2d;
float dummy_y_2d, dummy_t2_2d, dummy_dt2_2d, dummy_d2t2_2d,
    dummy_n2_2d, dummy_dn2_2d, dummy_d2n2_2d;
float GeoCur1,GeoCur2,DGCDs;
float x3[n], y3[n], z3[n], GeoCur3[n];

ReturnGeoCur( m, x3, y3, z3, GeoCur3, optimum);

for(int i=0; i<n; i++)
{
    for(int j=0; j<n; j++)
    {
        dif1=pow(x1[i]-x3[j],2)+pow(y1[i]-y3[j],2)+pow(z1[i]-z3[j],2);
        dif2=pow(x2[i]-x3[j],2)+pow(y2[i]-y3[j],2)+pow(z2[i]-z3[j],2);

        if (dif1<=min_dif1)
        {
            min_dif1=dif1;
            min_j=j;
        }
```

```
        if (dif2<=min_dif2)
        {
            min_dif2=dif2;
            max_j=j;
            min_i=i;
        }
    }
}
if (min_i<=max_size)
{
GeoCur1=GeoCur3[max_j];
GeoCur2=GeoCur3[max_j-1];
DGCDs=(GeoCur2-GeoCur1)/h;
dummy_x_2d=x_2d[min_i];
dummy_y_2d=y_2d[min_i];

    if (optimum==max)//Since the second curve is in minimum principle
    direction, the tangent is tangent2={0,1} and normal is normal2={-1,0}.
    The orthogonal curve is rendered in maximum principle direction ,
    hence tangent is tangento={1,0} and normal is normalo={0,1}. Since
    we render the curve reversely, so tangent of orthogonal curve is
    tangentnew0={-1,0}=normal2 and normal is normalnew0={0,-1}=-tangent2.
    {
        dummy_t1_2d=n1_2d[min_i];
        dummy_t2_2d=n2_2d[min_i];
        dummy_n1_2d=-t2_2d[min_i];
        dummy_n2_2d=-t2_2d[min_i];

    }
    else if(optimum==min)//Since the second curve is in
    maximum principle direction, the tangent is tangent2={1,0}
    and normal is normal2={0,1}. The orthogonal curve is
    rendered in minimum principle direction , hence tangent is
    tangento={0,1} and normal is normalo={-1,0}. Since we render
    the curve reversely, so tangent of orthogonal curve is
    tangentnew0={0,-1}=-normal2 and normal is normalnew0={1,0}=tangent2.
    {
        dummy_t1_2d=-n1_2d[min_i];
        dummy_t2_2d=-n2_2d[min_i];
        dummy_n1_2d=t2_2d[min_i];
        dummy_n2_2d=t2_2d[min_i];
    }


dummy_dt1_2d=GeoCur1*dummy_n1_2d;
dummy_dt2_2d=GeoCur1*dummy_n2_2d;
dummy_dn1_2d=-GeoCur1*dummy_t1_2d;
```

```
dummy_dn2_2d=-GeoCur1*dummy_t2_2d;
dummy_d2t1_2d=GeoCur1*dummy_dn1_2d+DGCDs*dummy_n1_2d;
dummy_d2t2_2d=GeoCur1*dummy_dn2_2d+DGCDs*dummy_n2_2d;
dummy_d2n1_2d=-GeoCur1*dummy_dt1_2d-DGCDs*dummy_t1_2d;
dummy_d2n2_2d=-GeoCur1*dummy_dt2_2d-DGCDs*dummy_t2_2d;

for(int k=max_j-1; k>=min_j; k--)
{
    GeoCur1=GeoCur3[k];
    GeoCur2=GeoCur3[k-1];
    DGCDs=(GeoCur2-GeoCur1)/h;

    dummy_x_2d+=RK4_Init(dummy_t1_2d,dummy_dt1_2d,h);
    dummy_y_2d+=RK4_Init(dummy_t2_2d,dummy_dt2_2d,h);
    dummy_t1_2d+=RK4_Init(dummy_dt1_2d,dummy_d2t1_2d,h);
    dummy_t2_2d+=RK4_Init(dummy_dt2_2d,dummy_d2t2_2d,h);
    dummy_n1_2d+=RK4_Init(dummy_dn1_2d,dummy_d2n1_2d,h);
    dummy_n2_2d+=RK4_Init(dummy_dn2_2d,dummy_d2n2_2d,h);

    dummy_dt1_2d=GeoCur1*dummy_n1_2d;
    dummy_dt2_2d=GeoCur1*dummy_n2_2d;
    dummy_dn1_2d=-GeoCur1*dummy_t1_2d;
    dummy_dn2_2d=-GeoCur1*dummy_t2_2d;
    dummy_d2t1_2d=GeoCur1*dummy_dn1_2d+DGCDs*dummy_n1_2d;
    dummy_d2t2_2d=GeoCur1*dummy_dn2_2d+DGCDs*dummy_n2_2d;
    dummy_d2n1_2d=-GeoCur1*dummy_dt1_2d-DGCDs*dummy_t1_2d;
    dummy_d2n2_2d=-GeoCur1*dummy_dt2_2d-DGCDs*dummy_t2_2d;
}
resx=dummy_x_2d;
resy=dummy_y_2d;


}
else
{
GeoCur1=GeoCur3[max_j];
GeoCur2=GeoCur3[max_j-1];
DGCDs=(GeoCur2-GeoCur1)/h;
dummy_x_2d=x_2d[min_i];
dummy_y_2d=y_2d[min_i];

    if (optimum==max)//Since the second curve is in minimum principle
    direction, the tangent is tangent2={0,1} and normal is normal2={1,0}.
    The orthogonal curve is rendered in maximum principle direction ,
    hence tangent is tangento={1,0} and normal is normalo={0,-1}. Since
    we render the curve reversely, so tangent of orthogonal curve is
    tangentnew0={-1,0}=-normal2 and normal is normalnew0={0,-1}=-tangent2.
```

```
    {
        dummy_t1_2d=n1_2d[min_i];
        dummy_t2_2d=n2_2d[min_i];
        dummy_n1_2d=-t2_2d[min_i];
        dummy_n2_2d=-t2_2d[min_i];

    }
    else if(optimum==min)//Since the second curve is in
    maximum principle direction, the tangent is tangent2={1,0}
    and normal is normal2={0,-1}. The orthogonal curve is
    rendered in minimum principle direction , hence tangent is
    tangento={0,1} and normal is normalo={1,0}. Since we render
    the curve reversely, so tangent of orthogonal curve is
    tangentnew0={0,-1}=normal2 and normal is normalnew0={1,0}=tangent2.
    {
        dummy_t1_2d=-n1_2d[min_i];
        dummy_t2_2d=-n2_2d[min_i];
        dummy_n1_2d=t2_2d[min_i];
        dummy_n2_2d=t2_2d[min_i];
    }


dummy_dt1_2d=GeoCur1*dummy_n1_2d;
dummy_dt2_2d=GeoCur1*dummy_n2_2d;
dummy_dn1_2d=-GeoCur1*dummy_t1_2d;
dummy_dn2_2d=-GeoCur1*dummy_t2_2d;
dummy_d2t1_2d=GeoCur1*dummy_dn1_2d+DGCDs*dummy_n1_2d;
dummy_d2t2_2d=GeoCur1*dummy_dn2_2d+DGCDs*dummy_n2_2d;
dummy_d2n1_2d=-GeoCur1*dummy_dt1_2d-DGCDs*dummy_t1_2d;
dummy_d2n2_2d=-GeoCur1*dummy_dt2_2d-DGCDs*dummy_t2_2d;

for(int k=max_j-1; k>=min_j; k--)
{
    GeoCur1=GeoCur3[k];
    GeoCur2=GeoCur3[k-1];
    DGCDs=(GeoCur2-GeoCur1)/h;

    dummy_x_2d+=RK4_Init(dummy_t1_2d,dummy_dt1_2d,h);
    dummy_y_2d+=RK4_Init(dummy_t2_2d,dummy_dt2_2d,h);
    dummy_t1_2d+=RK4_Init(dummy_dt1_2d,dummy_d2t1_2d,h);
    dummy_t2_2d+=RK4_Init(dummy_dt2_2d,dummy_d2t2_2d,h);
    dummy_n1_2d+=RK4_Init(dummy_dn1_2d,dummy_d2n1_2d,h);
    dummy_n2_2d+=RK4_Init(dummy_dn2_2d,dummy_d2n2_2d,h);

    dummy_dt1_2d=GeoCur1*dummy_n1_2d;
    dummy_dt2_2d=GeoCur1*dummy_n2_2d;
    dummy_dn1_2d=-GeoCur1*dummy_t1_2d;
```

```
    dummy_dn2_2d=-GeoCur1*dummy_t2_2d;
    dummy_d2t1_2d=GeoCur1*dummy_dn1_2d+DGCDs*dummy_n1_2d;
    dummy_d2t2_2d=GeoCur1*dummy_dn2_2d+DGCDs*dummy_n2_2d;
    dummy_d2n1_2d=-GeoCur1*dummy_dt1_2d-DGCDs*dummy_t1_2d;
    dummy_d2n2_2d=-GeoCur1*dummy_dt2_2d-DGCDs*dummy_t2_2d;
}
resx=dummy_x_2d;
resy=dummy_y_2d+0.2;


}
}



__global__ void fun3( float * xvalue, float *
    yvalue, float init_x, float init_y, mint init_s1, mint
    init_s2, mint max_size, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
mint optimum1, ii;
float x1[n], y1[n], z1[n], GeoCur1[n];
float x2[n], y2[n], z2[n], GeoCur2[n];
float x3[n], y3[n], z3[n], GeoCur3[n];
float x_2d[n], y_2d[n], t1_2d[n], t2_2d[n], n1_2d[n], n2_2d[n];
float res_x, res_y;
float dummy_x[n], dummy_y[n];
float Last_x, Last_y;

if(optimum==min)
{
    optimum1=max;
}
else if(optimum==max)
{
    optimum1=min;
}

ReturnGeoCur( init_s1, x1, y1, z1, GeoCur1, optimum);
ReturnGeoCur( init_s2, x2, y2, z2, GeoCur2, optimum);
TwoDimensionCurve( init_x, init_y,
    GeoCur2, x_2d, y_2d, t1_2d, t2_2d, n1_2d, n2_2d, optimum);

if( index < ListSize)
{
    ObtainSmallerPiece( index, max_size, x1, y1, z1, x2, y2, z2, x_2d,
    y_2d, t1_2d, t2_2d, n1_2d, n2_2d, res_x, res_y, optimum1);
```

```
    xvalue[index]= res_x;
    yvalue[index]= res_y;
}

}

__global__ void fun2( float * xvalue, float * yvalue, float
    init_x, float init_y, mint init_s, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float x1[n], y1[n], z1[n];
float GeoCur1[n];
float x_2d[n], y_2d[n], t1_2d[n], t2_2d[n], n1_2d[n], n2_2d[n];

ReturnGeoCur( init_s, x1, y1, z1, GeoCur1, optimum);
TwoDimensionCurve( init_x, init_y,
    GeoCur1, x_2d, y_2d, t1_2d, t2_2d, n1_2d, n2_2d, optimum);

if( index < ListSize)
{
    xvalue[index]= x_2d[index];
    yvalue[index]= y_2d[index];
}

}

__global__ void fun( float * xvalue, float * yvalue, float * zvalue,
    float * Geovalue, mint init_s, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float x1[n], y1[n], z1[n];
float GeoCur1[n];


ReturnGeoCur( init_s, x1, y1, z1, GeoCur1, optimum);

if( index <= ListSize)
{
    xvalue[index]= x1[index];
    yvalue[index]= y1[index];
    zvalue[index]= z1[index];
    Geovalue[index]= GeoCur1[index];
}

}


"
```

```
Needs["CUDALink`"]

AbsoluteTiming[funcLoCHP1 = CUDAFunctionLoad[kernelcodeHP, "fun",
   {{"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, _Integer, _Integer, _Integer}, 160]]

{136.457, CUDAFunction[<>, fun, {{Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, Integer64, Integer64, Integer64}]]}

AbsoluteTiming[
 funcLoCHP2 = CUDAFunctionLoad[kernelcodeHP, "fun2", {{"Float", _, "Output"},
    {"Float", _, "Output"}, "Float", "Float", _Integer, _Integer, _Integer}, 160]]

{0.0110301, CUDAFunction[<>, fun2, {{Float, _, Output},
   {Float, _, Output}, Float, Float, Integer64, Integer64, Integer64}]]}

AbsoluteTiming[funcLoCHP3 = CUDAFunctionLoad[kernelcodeHP,
   "fun3", {{"Float", _, "Output"}, {"Float", _, "Output"}, "Float",
    "Float", _Integer, _Integer, _Integer, _Integer, _Integer}, 160]]

{0.0115131, CUDAFunction[<>, fun3, {{Float, _, Output}, {Float, _, Output},
   Float, Float, Integer64, Integer64, Integer64, Integer64, Integer64}]]}

xyzvalue[v_, nn_, optimum_] := Flatten[
   bufferLoCHP11 = CUDAMemoryAllocate["Float", nn];
   bufferLoCHP12 = CUDAMemoryAllocate["Float", nn];
   bufferLoCHP13 = CUDAMemoryAllocate["Float", nn];
   bufferLoCHP14 = CUDAMemoryAllocate["Float", nn];
   resHP = AbsoluteTiming[funcLoCHP1[bufferLoCHP11,
       bufferLoCHP12, bufferLoCHP13, bufferLoCHP14, v, optimum, nn]];
   Transpose[{CUDAMemoryGet[bufferLoCHP11], CUDAMemoryGet[bufferLoCHP12],
     CUDAMemoryGet[bufferLoCHP13], CUDAMemoryGet[bufferLoCHP14]}], 0];

xyvalueOri[initx_, inity_, v_, nn_, optimum_] := Flatten[
   bufferLoCHP21 = CUDAMemoryAllocate["Float", nn];
   bufferLoCHP22 = CUDAMemoryAllocate["Float", nn];
   resHP2 = AbsoluteTiming[
     funcLoCHP2[bufferLoCHP21, bufferLoCHP22, initx, inity, v, optimum, nn]];
   Transpose[{CUDAMemoryGet[bufferLoCHP21], CUDAMemoryGet[bufferLoCHP22]}], 0];

xyvalue[initx_, inity_, v1_, v2_, maxn_, nn_, optimum_] := Flatten[
   bufferLoCHP31 = CUDAMemoryAllocate["Float", nn];
   bufferLoCHP32 = CUDAMemoryAllocate["Float", nn];
   resHP3 = AbsoluteTiming[funcLoCHP3[bufferLoCHP31,
       bufferLoCHP32, initx, inity, v1, v2, maxn, optimum, nn]];
   Transpose[{CUDAMemoryGet[bufferLoCHP31], CUDAMemoryGet[bufferLoCHP32]}], 0];
```

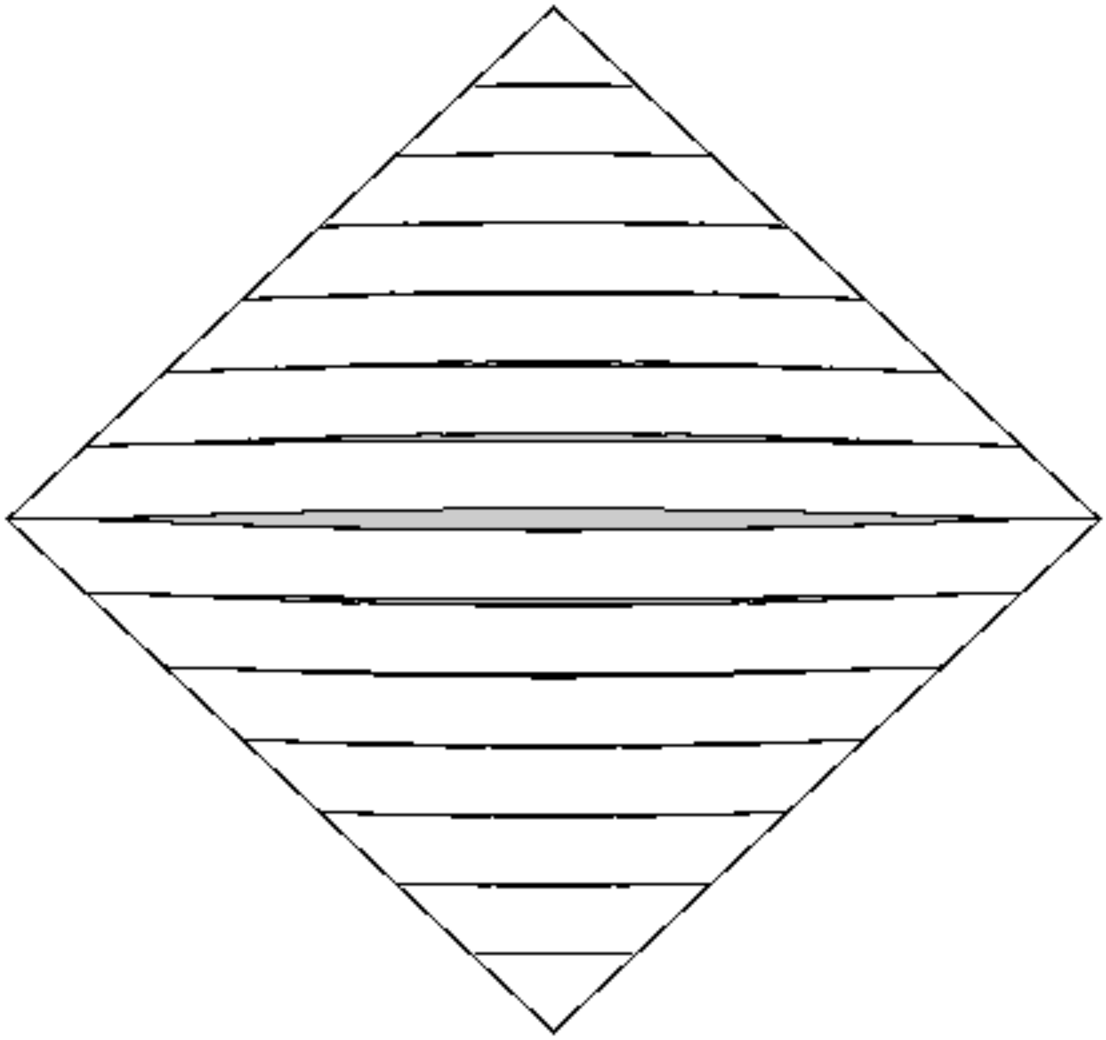
```
Coordinates[v_, nn_, optimum_] := Flatten[
   bufferLoCHP41 = CUDAMemoryAllocate["Float", nn];
   bufferLoCHP42 = CUDAMemoryAllocate["Float", nn];
   bufferLoCHP43 = CUDAMemoryAllocate["Float", nn];
   resHP4 = AbsoluteTiming[
     funcLoCHP4[bufferLoCHP41, bufferLoCHP42, bufferLoCHP43, v, optimum, nn]];
   Transpose[{CUDAMemoryGet[bufferLoCHP41], CUDAMemoryGet[bufferLoCHP42],
     CUDAMemoryGet[bufferLoCHP43]}], 0];

ShortPPCutLineat100 =
  Table[{1, 1}, {i, 298}] * xyvalue[0, 0.116, 0, 100, 300, 298, 1];
ShortPNCutLineat100 = Table[{1, -1}, {i, 298}] *
   xyvalue[0, 0.116, 0, 100, 300, 298, 1];
ShortNPCutLineat100 = Table[{-1, 1}, {i, 298}] *
   xyvalue[0, 0.116, 0, 100, 300, 298, 1];
ShortNNCutLineat100 = Table[{-1, -1}, {i, 298}] *
   xyvalue[0, 0.116, 0, 100, 300, 298, 1];
ShortPPCutLineat200 = Table[{1, 1}, {i, 242}] *
   xyvalue[0, 0.226, 100, 200, 250, 242, 1];
ShortPNCutLineat200 = Table[{1, -1}, {i, 242}] *
   xyvalue[0, 0.226, 100, 200, 250, 242, 1];
ShortNPCutLineat200 = Table[{-1, 1}, {i, 242}] *
   xyvalue[0, 0.226, 100, 200, 250, 242, 1];
ShortNNCutLineat200 = Table[{-1, -1}, {i, 242}] *
   xyvalue[0, 0.226, 100, 200, 250, 242, 1];
ShortPPCutLineat300 = Table[{1, 1}, {i, 187}] *
   xyvalue[0, 0.331, 200, 300, 200, 187, 1];
ShortPNCutLineat300 = Table[{1, -1}, {i, 187}] *
   xyvalue[0, 0.331, 200, 300, 200, 187, 1];
ShortNPCutLineat300 = Table[{-1, 1}, {i, 187}] *
   xyvalue[0, 0.331, 200, 300, 200, 187, 1];
ShortNNCutLineat300 = Table[{-1, -1}, {i, 187}] *
   xyvalue[0, 0.331, 200, 300, 200, 187, 1];
ShortPPCutLineat400 = Table[{1, 1}, {i, 134}] *
   xyvalue[0, 0.434, 300, 400, 150, 134, 1];
ShortPNCutLineat400 = Table[{1, -1}, {i, 134}] *
   xyvalue[0, 0.434, 300, 400, 150, 134, 1];
ShortNPCutLineat400 = Table[{-1, 1}, {i, 134}] *
   xyvalue[0, 0.434, 300, 400, 150, 134, 1];
ShortNNCutLineat400 = Table[{-1, -1}, {i, 134}] *
   xyvalue[0, 0.434, 300, 400, 150, 134, 1];
ShortPPCutLineat500 = Table[{1, 1}, {i, 86}] *
   xyvalue[0, 0.5355, 400, 500, 100, 86, 1];
ShortPNCutLineat500 = Table[{1, -1}, {i, 86}] *
   xyvalue[0, 0.5355, 400, 500, 100, 86, 1];
ShortNPCutLineat500 = Table[{-1, 1}, {i, 86}] *
   xyvalue[0, 0.5355, 400, 500, 100, 86, 1];
```

```
ShortNNCutLineat500 = Table[{-1, -1}, {i, 86}] *
   xyvalue[0, 0.5355, 400, 500, 100, 86, 1];
ShortPPCutLineat600 = Table[{1, 1}, {i, 41}] *
   xyvalue[0, 0.6365, 500, 600, 50, 41, 1];
ShortPNCutLineat600 = Table[{1, -1}, {i, 41}] *
   xyvalue[0, 0.6365, 500, 600, 50, 41, 1];
ShortNPCutLineat600 = Table[{-1, 1}, {i, 41}] *
   xyvalue[0, 0.6365, 500, 600, 50, 41, 1];
ShortNNCutLineat600 = Table[{-1, -1}, {i, 41}] *
   xyvalue[0, 0.6365, 500, 600, 50, 41, 1];

ShortPPOriLineat0 = Table[{1, 1}, {i, 350}] * xyvalueOri[0, 0, 0, 350, 1];
ShortPNOriLineat0 = Table[{1, -1}, {i, 350}] * xyvalueOri[0, 0, 0, 350, 1];
ShortNPOriLineat0 = Table[{-1, 1}, {i, 350}] * xyvalueOri[0, 0, 0, 350, 1];
ShortNNOriLineat0 = Table[{-1, -1}, {i, 350}] * xyvalueOri[0, 0, 0, 350, 1];
ShortPPOriLineat100 = Table[{1, 1}, {i, 300}] * xyvalueOri[0, 0.116, 100, 300, 1];
ShortPNOriLineat100 = Table[{1, -1}, {i, 300}] * xyvalueOri[0, 0.116, 100, 300, 1];
ShortNPOriLineat100 = Table[{-1, 1}, {i, 300}] * xyvalueOri[0, 0.116, 100, 300, 1];
ShortNNOriLineat100 =
  Table[{-1, -1}, {i, 300}] * xyvalueOri[0, 0.116, 100, 300, 1];
ShortPPOriLineat200 = Table[{1, 1}, {i, 250}] * xyvalueOri[0, 0.226, 200, 250, 1];
ShortPNOriLineat200 = Table[{1, -1}, {i, 250}] * xyvalueOri[0, 0.226, 200, 250, 1];
ShortNPOriLineat200 = Table[{-1, 1}, {i, 250}] * xyvalueOri[0, 0.226, 200, 250, 1];
ShortNNOriLineat200 =
  Table[{-1, -1}, {i, 250}] * xyvalueOri[0, 0.226, 200, 250, 1];
ShortPPOriLineat300 = Table[{1, 1}, {i, 200}] * xyvalueOri[0, 0.331, 300, 200, 1];
ShortPNOriLineat300 = Table[{1, -1}, {i, 200}] * xyvalueOri[0, 0.331, 300, 200, 1];
ShortNPOriLineat300 = Table[{-1, 1}, {i, 200}] * xyvalueOri[0, 0.331, 300, 200, 1];
ShortNNOriLineat300 =
  Table[{-1, -1}, {i, 200}] * xyvalueOri[0, 0.331, 300, 200, 1];
ShortPPOriLineat400 = Table[{1, 1}, {i, 150}] * xyvalueOri[0, 0.434, 400, 150, 1];
ShortPNOriLineat400 = Table[{1, -1}, {i, 150}] * xyvalueOri[0, 0.434, 400, 150, 1];
ShortNPOriLineat400 = Table[{-1, 1}, {i, 150}] * xyvalueOri[0, 0.434, 400, 150, 1];
ShortNNOriLineat400 =
  Table[{-1, -1}, {i, 150}] * xyvalueOri[0, 0.434, 400, 150, 1];
ShortPPOriLineat500 = Table[{1, 1}, {i, 100}] * xyvalueOri[0, 0.5355, 500, 100, 1];
ShortPNOriLineat500 =
  Table[{1, -1}, {i, 100}] * xyvalueOri[0, 0.5355, 500, 100, 1];
ShortNPOriLineat500 = Table[{-1, 1}, {i, 100}] *
   xyvalueOri[0, 0.5355, 500, 100, 1];
ShortNNOriLineat500 = Table[{-1, -1}, {i, 100}] *
   xyvalueOri[0, 0.5355, 500, 100, 1];
ShortPPOriLineat600 = Table[{1, 1}, {i, 50}] * xyvalueOri[0, 0.6365, 600, 50, 1];
ShortPNOriLineat600 = Table[{1, -1}, {i, 50}] * xyvalueOri[0, 0.6365, 600, 50, 1];
ShortNPOriLineat600 = Table[{-1, 1}, {i, 50}] * xyvalueOri[0, 0.6365, 600, 50, 1];
ShortNNOriLineat600 = Table[{-1, -1}, {i, 50}] * xyvalueOri[0, 0.6365, 600, 50, 1];
```

Result

```
Show[
 ListLinePlot[{ShortPPCutLineat100, ShortPNCutLineat100},
  Filling → {1 → {2}}, PlotStyle → Black],
 ListLinePlot[{Reverse[ShortNPCutLineat100], Reverse[ShortNNCutLineat100]},
  Filling → {1 → {2}}, PlotStyle → Black],
 ListLinePlot[ShortPPOriLineat0[[293 ;; 350]], PlotStyle → Black],
 ListLinePlot[ShortNNOriLineat0[[293 ;; 350]], PlotStyle → Black],
 ListLinePlot[{ShortPPCutLineat200, ShortPPOriLineat100},
  Filling → {1 → {2}}, PlotStyle → Black],
 ListLinePlot[{Reverse[ShortNPCutLineat200], Reverse[ShortNPOriLineat100]},
  Filling → {1 → {2}}, PlotStyle → Black],
 ListLinePlot[{ShortPNCutLineat200, ShortPNOriLineat100},
  Filling → {1 → {2}}, PlotStyle → Black],
 ListLinePlot[{Reverse[ShortNNCutLineat200], Reverse[ShortNNOriLineat100]},
  Filling → {1 → {2}}, PlotStyle → Black],
 ListLinePlot[{ShortPPCutLineat300, ShortPPOriLineat200},
  Filling → {1 → {2}}, PlotStyle → Black],
 ListLinePlot[{Reverse[ShortNPCutLineat300], Reverse[ShortNPOriLineat200]},
  Filling → {1 → {2}}, PlotStyle → Black],
 ListLinePlot[{ShortPNCutLineat300, ShortPNOriLineat200},
  Filling → {1 → {2}}, PlotStyle → Black],
 ListLinePlot[{Reverse[ShortNNCutLineat300], Reverse[ShortNNOriLineat200]},
  Filling → {1 → {2}}, PlotStyle → Black],
 ListLinePlot[{ShortPPCutLineat400, ShortPPOriLineat300},
  Filling → {1 → {2}}, PlotStyle → Black],
 ListLinePlot[{Reverse[ShortNPCutLineat400], Reverse[ShortNPOriLineat300]},
  Filling → {1 → {2}}, PlotStyle → Black],
 ListLinePlot[{ShortPNCutLineat400, ShortPNOriLineat300},
  Filling → {1 → {2}}, PlotStyle → Black],
 ListLinePlot[{Reverse[ShortNNCutLineat400], Reverse[ShortNNOriLineat300]},
  Filling → {1 → {2}}, PlotStyle → Black],
 ListLinePlot[{ShortPPCutLineat500, ShortPPOriLineat400},
  Filling → {1 → {2}}, PlotStyle → Black],
 ListLinePlot[{Reverse[ShortNPCutLineat500], Reverse[ShortNPOriLineat400]},
  Filling → {1 → {2}}, PlotStyle → Black],
 ListLinePlot[{ShortPNCutLineat500, ShortPNOriLineat400},
  Filling → {1 → {2}}, PlotStyle → Black],
 ListLinePlot[{Reverse[ShortNNCutLineat500], Reverse[ShortNNOriLineat400]},
  Filling → {1 → {2}}, PlotStyle → Black],
 ListLinePlot[{ShortPPCutLineat600, ShortPPOriLineat500},
  Filling → {1 → {2}}, PlotStyle → Black],
 ListLinePlot[{Reverse[ShortNPCutLineat600], Reverse[ShortNPOriLineat500]},
  Filling → {1 → {2}}, PlotStyle → Black],
```

```
ListLinePlot[{ShortPNCutLineat600, ShortPNOriLineat500},
 Filling → {1 → {2}}, PlotStyle → Black],
ListLinePlot[{Reverse[ShortNNCutLineat600], Reverse[ShortNNOriLineat500]},
 Filling → {1 → {2}}, PlotStyle → Black],
ListLinePlot[{ShortPPOriLineat600, ShortNPOriLineat600,
  ShortPNOriLineat600, ShortNNOriLineat600}, PlotStyle → Black],

ListLinePlot[{{{0.35, 0}, {0, 0.75}}, {{-0.35, 0}, {0, 0.75}},
  {{0.35, 0}, {0, -0.75}}, {{-0.35, 0}, {0, -0.75}}}, PlotStyle → Black],
PlotRange → All, Axes → False, AspectRatio → 1]
```



Piece by piece

Section 3.2

Section 5

Obtain the necessary information-Lambda, initRT, arc-length

Hyperbolic Paraboloid-obtain points and tangent

```
Clear["Global`*"]
```

Obtain Points and its max and min tangent

```
DPbval = {{1 / 5}, {3 / 40, 9 / 40}, {44 / 45, -56 / 15, 32 / 9},
    {19 372 / 6561, -25 360 / 2187, 64 448 / 6561, -212 / 729},
    {9017 / 3168, -355 / 33, 46 732 / 5247, 49 / 176, -5103 / 18 656},
    {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84}};
DPcval = {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84, 0};
DPaval = {1 / 5, 3 / 10, 4 / 5, 8 / 9, 1, 1};
DPerrval =
  {71 / 57 600, 0, -71 / 16 695, 71 / 1920, -17 253 / 339 200, 22 / 525, -1 / 40};
DPCoefficients[5, p_] :=
  N[{DPbval, DPcval, DPaval, DPerrval}, p];

ClassicalRK4bval = {{1 / 2}, {0, 1 / 2}, {0, 0, 1}};
ClassicalRK4cval = {1 / 6, 1 / 3, 1 / 3, 1 / 6};
ClassicalRK4aval = {1 / 2, 1 / 2, 1};
ClassicalRK4Coefficients[4, p_] :=
  N[{ClassicalRK4bval, ClassicalRK4cval, ClassicalRK4aval }, p];
Surf[u_, v_][s_] := {u[s], v[s], u[s] ^2 - v[s] ^2};
DuSurf[u_, v_][s_] := {1, 0, 2 u[s]};
DvSurf[u_, v_][s_] := {0, 1, -2 v[s]};
Du2Surf[u_, v_][s_] := {0, 0, 2};
Dv2Surf[u_, v_][s_] := {0, 0, -2};
DuDvSurf[u_, v_][s_] := {0, 0, 0};
CrossN[u_, v_][s_] := Cross[DuSurf[u, v][s], DvSurf[u, v][s]];
ScalarN[u_, v_][s_] := Sqrt[CrossN[u, v][s].CrossN[u, v][s]];
NormalSurf[u_, v_][s_] := CrossN[u, v][s] / ScalarN[u, v][s];
SurfE[u_, v_][s_] := DuSurf[u, v][s].DuSurf[u, v][s];
SurfF[u_, v_][s_] := DuSurf[u, v][s].DvSurf[u, v][s];
SurfG[u_, v_][s_] := DvSurf[u, v][s].DvSurf[u, v][s];
SurfL[u_, v_][s_] := Du2Surf[u, v][s].NormalSurf[u, v][s];
SurfM[u_, v_][s_] := DuDvSurf[u, v][s].NormalSurf[u, v][s];
SurfN[u_, v_][s_] := Dv2Surf[u, v][s].NormalSurf[u, v][s];
GaussCurva[u_, v_][s_] := (SurfL[u, v][s] × SurfN[u, v][s] - SurfM[u, v][s] ^2) /
```

```
    (SurfE[u, v][s] × SurfG[u, v][s] - SurfF[u, v][s]^2);
MeanCurva[u_, v_][s_] := 1/2 (SurfE[u, v][s] × SurfN[u, v][s] +
      SurfG[u, v][s] × SurfL[u, v][s] - 2 SurfF[u, v][s] × SurfM[u, v][s]) /
    (SurfE[u, v][s] × SurfG[u, v][s] - SurfF[u, v][s]^2);
MaxPrinci[u_, v_][s_] := MeanCurva[u, v][s] +
   Sqrt[MeanCurva[u, v][s]^2 - GaussCurva[u, v][s]];
MinPrinci[u_, v_][s_] := MeanCurva[u, v][s] -
   Sqrt[MeanCurva[u, v][s]^2 - GaussCurva[u, v][s]];
Princi[a_, u_, v_][s_] := Piecewise[
   {{MaxPrinci[u, v][s], a === max}, {MinPrinci[u, v][s], a === min}}];
Etaη[a_, u_, v_][s_] := 1 / Sqrt[
    SurfE[u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s])^2 -
     2 SurfF[u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s])
      (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s]) +
     SurfG[u, v][s] (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s])^2];
Miuμ[a_, u_, v_][s_] := 1 / Sqrt[SurfE[u, v][s]
      (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])^2 -
     2 SurfF[u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s])
      (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s]) +
     SurfG[u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s])^2];

DuDs[a_, u_, v_][s_] :=
 Piecewise[{{Etaη[a, u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s]),
    Sign[(SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s])]
      (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s]) ≥
     Sign[(SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])]
      (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])}},
  Miuμ[a, u, v][s] (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])]
DvDs[a_, u_, v_][s_] := Piecewise[
  {{-Etaη[a, u, v][s] (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s]),
    Sign[(SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s])]
      (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s]) ≥
     Sign[(SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])]
      (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])}},
  -Miuμ[a, u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s])]

d1uds1[a_, b_, c_] := DuDs[a, u, v][s] /. {u[s] → b, v[s] → c}
d1vds1[a_, b_, c_] := DvDs[a, u, v][s] /. {u[s] → b, v[s] → c}

DSurDs1[a_, u_, v_][s_] :=
 DuSurf[u, v][s] × DuDs[a, u, v][s] + DvSurf[u, v][s] × DvDs[a, u, v][s]
DSurDs2[a_, b_, c_] := DSurDs1[a, u, v][s] /. {u[s] → b, v[s] → c}
d1uds1[a_, b_, c_] := DuDs[a, u, v][s] /. {u[s] → b, v[s] → c}
d1vds1[a_, b_, c_] := DvDs[a, u, v][s] /. {u[s] → b, v[s] → c}

DEDs[u_, v_][s_] := D[SurfE[u, v][s], s]
DFDs[u_, v_][s_] := D[SurfF[u, v][s], s]
```

```
DGDs[u_, v_][s_] := D[SurfG[u, v][s], s]
DLDs[u_, v_][s_] := D[SurfL[u, v][s], s]
DMDs[u_, v_][s_] := D[SurfM[u, v][s], s]
DNDs[u_, v_][s_] := D[SurfN[u, v][s], s]
DkpDs[a_, u_, v_][s_] := D[Princi[a, u, v][s], s]
D2EDs2[u_, v_][s_] := D[SurfE[u, v][s], {s, 2}]
D2FDs2[u_, v_][s_] := D[SurfF[u, v][s], {s, 2}]
D2GDs2[u_, v_][s_] := D[SurfG[u, v][s], {s, 2}]
D2LDs2[u_, v_][s_] := D[SurfL[u, v][s], {s, 2}]
D2MDs2[u_, v_][s_] := D[SurfM[u, v][s], {s, 2}]
D2NDs2[u_, v_][s_] := D[SurfN[u, v][s], {s, 2}]
D2kpDs2[a_, u_, v_][s_] := D[Princi[a, u, v][s], {s, 2}]
DuDt[a_, u_, v_][s_] :=
  Piecewise[{{(SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s]),
      Sign[(SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s])]
         (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s]) ≥
        Sign[(SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])]
         (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])}},
    (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])];
DvDt[a_, u_, v_][s_] := Piecewise[
    {{(SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s]),
      Sign[(SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s])]
         (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s]) ≥
        Sign[(SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])]
         (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])}},
    (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s])];
BinormalSurf[u_, v_][s_] := Cross[NormalSurf[u, v][s],
    (DuSurf[u, v][s] × u'[s] + DvSurf[u, v][s] × v'[s])];

Alp1[a_, u_, v_][s_] := Du2Surf[u, v][s] (u'[s])^2 +
    2 DuDvSurf[u, v][s] × u'[s] × v'[s] + Dv2Surf[u, v][s] (v'[s])^2;
Beta1[a_, u_, v_][s_] := Piecewise[
    {{- (DLDs[u, v][s] - DkpDs[a, u, v][s] × SurfE[u, v][s] -
           Princi[a, u, v][s] × DEDs[u, v][s]) u'[s] -
        (DMDs[u, v][s] - DkpDs[a, u, v][s] × SurfF[u, v][s] -
           Princi[a, u, v][s] × DFDs[u, v][s]) v'[s],
      Sign[(SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s])]
         (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s]) ≥
        Sign[(SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])]
         (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])}},
    - (DMDs[u, v][s] - DkpDs[a, u, v][s] × SurfF[u, v][s] -
          Princi[a, u, v][s] × DFDs[u, v][s]) u'[s] -
      (DNDs[u, v][s] - DkpDs[a, u, v][s] × SurfG[u, v][s] -
          Princi[a, u, v][s] × DGDs[u, v][s]) v'[s]];
Matris[a_, u_, v_][s_] := {{SurfE[u, v][s], SurfF[u, v][s],
    -BinormalSurf[u, v][s].DuSurf[u, v][s]},
  {SurfF[u, v][s], SurfG[u, v][s], -BinormalSurf[u, v][s].DvSurf[u, v][s]},
```

```
  {DvDt[a, u, v][s], DuDt[a, u, v][s], 0}}
MatricR2[a_, u_, v_][s_] := {-Alp1[a, u, v][s].DuSurf[u, v][s],
  -Alp1[a, u, v][s].DvSurf[u, v][s],
  Beta1[a, u, v][s]}

InverResult[a_, b_, c_] :=
 N[(Inverse[Matris[a, u, v][s] /. {u'[s] → d1uds1[a, b, c],
       v'[s] → d1vds1[a, b, c], u[s] → b, v[s] → c}]).(MatricR2[a, u, v][s] /.
     {u'[s] → d1uds1[a, b, c], v'[s] → d1vds1[a, b, c], u[s] → b, v[s] → c})]
D2uDs2[a_, b_, c_] := InverResult[a, b, c][[1]]
D2vDs2[a_, b_, c_] := InverResult[a, b, c][[2]]
GeoCur[a_, b_, c_] := InverResult[a, b, c][[3]]

Princi1[a_, b_, c_] :=
  Piecewise[{{MaxPrinci[u, v][s], a === max}, {MinPrinci[u, v][s], a === min}}] /.
   {u[s] → b, v[s] → c};
Curofacurve[a_, b_, c_] := Sqrt[Princi1[a, b, c]^2 + GeoCur[a, b, c]^2]

SolvedudvDP[a_] := ParametricNDSolve[
  {u'[ArcL] == DuDs[a, u, v][ArcL], v'[ArcL] == DvDs[a, u, v][ArcL],
   u[0] == iniu, v[0] == iniv},
  {u, v}, {ArcL, 0, 2}, {iniu, iniv}, Method →
   {"ExplicitRungeKutta", "DifferenceOrder" → 5, "Coefficients" → DPCoefficients,
    "EmbeddedDifferenceOrder" → 4, "StepSizeControlParameters" -> {1, 0},
    "StepSizeSafetyFactors" → {1, 1}, "StepSizeRatioBounds" → {1/25, 25},
    "StiffnessTest" → False, "DiscontinuityProcessing" → False},
  MaxStepSize → 0.01, MaxSteps → Automatic]

solution41 = SolvedudvDP[max];
solution42 = SolvedudvDP[min];
MaximumusDP[t_, s_] :=
 u[u[0, 0][t] /. solution42, v[0, 0][t] /. solution42][s] /. solution41
MaximumvsDP[t_, s_] :=
 v[u[0, 0][t] /. solution42, v[0, 0][t] /. solution42][s] /. solution41
MinimumusDP[t_, s_] :=
 u[u[0, 0][t] /. solution41, v[0, 0][t] /. solution41][s] /. solution42
MinimumvsDP[t_, s_] :=
 v[u[0, 0][t] /. solution41, v[0, 0][t] /. solution41][s] /. solution42
```

```
NOIteration[a_, b_, tol_] := Ceiling[N[1 / Log[2] Log[Abs[a - b] / tol]]];
MultipleBisectionMethod[x1_, x2_, MaxIteration_] :=
 Module[{i = 1, j, k, s0, s1, sE = 0.10383330285549164, u0, u1,
   uE, t1 = x1, t2 = x2, tole0 = 10^-50, maxIte = MaxIteration},
  While[i < maxIte,
   {
    j = 1;
    s0 = 0;
    s1 = 2;
    While[j < NOIteration[s0, s1, tole0], {sE = (s0 + s1) / 2;
      If[MaximumusDP[t1, sE] < MinimumusDP[t2, uE], s1 = sE, s0 = sE]}; j++];

    k = 1;
    u0 = 0;
    u1 = 2;
    While[k < NOIteration[u0, u1, tole0], {uE = (u0 + u1) / 2;
      If[MaximumvsDP[t1, sE] > MinimumvsDP[t2, uE], u1 = uE, u0 = uE]}; k++];

   };
   i++];
  N[{sE, uE}]]
```

```
MultipleBisectionMethod[.22, .9, 30]
MultipleBisectionMethod[.44, .9, 30]
MultipleBisectionMethod[.66, .9, 30]
```

{0.930554, 0.30895}

{1.00313, 0.60727}

{1.09171, 0.892787}

```
MultipleBisectionMethod[.9, .22, 30]
MultipleBisectionMethod[.9, .44, 30]
MultipleBisectionMethod[.9, .66, 30]
```

{0.30895, 0.930554}

{0.60727, 1.00313}

{0.892787, 1.09171}

```
MultipleBisectionMethod[0, .9, 30]
MultipleBisectionMethod[.1, .9, 30]
MultipleBisectionMethod[.3, .9, 30]
MultipleBisectionMethod[.5, .9, 30]
MultipleBisectionMethod[.7, .9, 30]
MultipleBisectionMethod[.9, .9, 30]
```

$\{0.9, 8.67362\times10^{-19}\}$

{0.906697, 0.141252}

{0.953734, 0.418938}

{1.02645, 0.686351}

{1.10839, 0.943466}

{1.19207, 1.19207}

```
1.1920739744274533`, 1.1920739744274533`
```

```
Surf[u, v][s] /. {u[s] → MaximumusDP[0.9, 1.1920739744274533`] ,
  v[s] → MaximumvsDP[0.9, 1.1920739744274533`] }
```

$\{-1.00131, -1.00131, 2.84217\times10^{-14}\}$

```
Point00 = {MaximumusDP[0, 0], MaximumvsDP[0, 0],
  MaximumusDP[0, 0]^2 - MaximumvsDP[0, 0]^2}
Point03 = {MaximumusDP[0, 0.9], MaximumvsDP[0, 0.9],
  MaximumusDP[0, 0.9]^2 - MaximumvsDP[0, 0.9]^2}
Point30 = {MinimumusDP[0, 0.9], MinimumvsDP[0, 0.9],
  MinimumusDP[0, 0.9]^2 - MinimumvsDP[0, 0.9]^2}
Point33 = {-1.0013149889663, -1.0013149889663, 0}
```

{0., 0., 0.}

{-0.707726, 0., 0.500877}

{0., -0.707726, -0.500877}

{-1.00131, -1.00131, 0}

```
MaxPoint33 = {MaximumusDP[0.9, 1.1920739744274533],
  MaximumvsDP[0.9, 1.1920739744274533], MaximumusDP[0.9, 1.1920739744274533]^2 -
   MaximumvsDP[0.9, 1.1920739744274533]^2}
MinPoint33 = {MinimumusDP[0.9, 1.1920739744274533],
  MinimumvsDP[0.9, 1.1920739744274533], MinimumusDP[0.9, 1.1920739744274533]^2 -
   MinimumvsDP[0.9, 1.1920739744274533]^2}
```

$\{-1.00131, -1.00131, 2.84217\times10^{-14}\}$

$\{-1.00131, -1.00131, -2.84217\times10^{-14}\}$

```
Point00 = {MaximumusDP[0, 0], MaximumvsDP[0, 0],
  MaximumusDP[0, 0]^2 - MaximumvsDP[0, 0]^2}
Point03 = {MaximumusDP[0, 0.9], MaximumvsDP[0, 0.9],
  MaximumusDP[0, 0.9]^2 - MaximumvsDP[0, 0.9]^2}
Point30 = {MinimumusDP[0, 0.9], MinimumvsDP[0, 0.9],
  MinimumusDP[0, 0.9]^2 - MinimumvsDP[0, 0.9]^2}
Point33 = {-1.0013149889663, -1.0013149889663, 0}
```

```
{0., 0., 0.}

{-0.707726, 0., 0.500877}

{0., -0.707726, -0.500877}

{-1.00131, -1.00131, 0}
```

```
MaxTangentPoint00 = DSurDs2[max, 0, 0]
MaxTangentPoint03 = DSurDs2[max, -0.7077264466498845, 0]
MaxTangentPoint30 = DSurDs2[max, 0., -0.7077264466498845]
MaxTangentPoint33 = DSurDs2[max, -1.0013149889663, -1.0013149889663]
```

```
{-1, 0, 0}

{-0.577013, 0., 0.816735}

{-1., 0., 0.}

{-0.666472, -0.333528, 0.666764}
```

```
MinTangentPoint00 = DSurDs2[min, 0, 0]
MinTangentPoint03 = DSurDs2[min, -0.7077264466498845, 0]
MinTangentPoint30 = DSurDs2[min, 0., -0.7077264466498845]
MinTangentPoint33 = DSurDs2[min, -1.0013149889663, -1.0013149889663]
```

```
{0, -1, 0}

{0., -1., 0.}

{0., -0.577013, -0.816735}

{-0.333528, -0.666472, -0.666764}
```

Log Aesthetic Curves-Bisection

General equations

```
Point00 = {0, 0, 0};(*According to Point*)
Point03 = {-0.7077264466498845, 0, 0.5008767232876717};
Point30 = {0, -0.7077264466498845, -0.5008767232876717};

MaxTangentPoint00 = {-1, 0, 0};(*According to LoC*)
MaxTangentPoint03 = {-0.5770131138020663, 0, 0.8167348814030436};
MinTangentPoint00 = {0, -1, 0};
MinTangentPoint30 = {0., -0.5770131138020663, -0.8167348814030436};



DPbval = {{1 / 5}, {3 / 40, 9 / 40}, {44 / 45, -56 / 15, 32 / 9},
    {19 372 / 6561, -25 360 / 2187, 64 448 / 6561, -212 / 729},
    {9017 / 3168, -355 / 33, 46 732 / 5247, 49 / 176, -5103 / 18 656},
    {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84}};
DPcval = {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84, 0};
DPaval  = {1 / 5, 3 / 10, 4 / 5, 8 / 9, 1, 1};
DPerrval =
   {71 / 57 600, 0, -71 / 16 695, 71 / 1920, -17 253 / 339 200, 22 / 525, -1 / 40};
DPCoefficients[5, p_] :=
   N[{DPbval, DPcval, DPaval, DPerrval}, p];
CurvatureLASC[ArcL_, Alpha_, Lambda_, InitRC_] :=
  Piecewise[{{1 / Exp[Lambda * ArcL + Log[InitRC]], Alpha == 0}},
   (InitRC ^ (Alpha) + Lambda * Alpha * ArcL) ^ (-1 / Alpha)]
ArcLengths[Alpha_, InitRC_, Lambda_, Theta_] :=
  Piecewise[{{(Exp[Lambda * Theta + Log[InitRC]] - InitRC) / Lambda, Alpha == 1},
    {(Log[(InitRC ^ -1 - Lambda * Theta) ^ -1] - Log[InitRC]) / Lambda, Alpha == 0}},
   ((InitRC ^ (Alpha - 1) + (Alpha - 1) * Lambda * Theta) ^ (Alpha / (Alpha - 1)) -
      InitRC ^ Alpha) / (Alpha * Lambda)]

LACEquation[Alpha_, Lambda_, InitRC_] :=
  Flatten[NDSolve[{x'[ArcL] == t1[ArcL], y'[ArcL] == t2[ArcL], z'[ArcL] == t3[ArcL],
     t1'[ArcL] == CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × n1[ArcL],
     t2'[ArcL] == CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × n2[ArcL],
     t3'[ArcL] == CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × n3[ArcL],
     n1'[ArcL] == -CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × t1[ArcL],
     n2'[ArcL] == -CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × t2[ArcL],
     n3'[ArcL] == -CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × t3[ArcL],

     x[0] == 0, y[0] == 0, z[0] == 0,
     t1[0] == 1, t2[0] == 0, t3[0] == 0,
     n1[0] == 0, n2[0] == 1, n3[0] == 0},
    {x, y, z, t1, t2, t3, n1, n2, n3},
```

```
    {ArcL, -8, 8}, Method → {"ExplicitRungeKutta",
      "DifferenceOrder" → 5, "Coefficients" → DPCoefficients,
      "EmbeddedDifferenceOrder" → 4, "StepSizeControlParameters" -> {1, 0},
      "StepSizeSafetyFactors" → {1, 1}, "StepSizeRatioBounds" → {1 / 25, 25},
      "StiffnessTest" → False, "DiscontinuityProcessing" → False},
    MaxStepSize → 0.01, MaxSteps → Automatic]]


LACEquation2[Alpha_, InitRC_] := Flatten[ParametricNDSolve[
    {x'[ArcL] == t1[ArcL], y'[ArcL] == t2[ArcL], z'[ArcL] == t3[ArcL],
     t1'[ArcL] == CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × n1[ArcL],
     t2'[ArcL] == CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × n2[ArcL],
     t3'[ArcL] == CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × n3[ArcL],
     n1'[ArcL] ==
      -CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × t1[ArcL] + 0 * b1[ArcL],
     n2'[ArcL] == -CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × t2[ArcL] +
       0 * b2[ArcL],
     n3'[ArcL] == -CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × t3[ArcL] +
       0 * b3[ArcL],
     b1'[ArcL] == -0 * n1[ArcL],
     b2'[ArcL] == -0 * n2[ArcL],
     b3'[ArcL] == -0 * n3[ArcL],
     x[0] == 0, y[0] == 0, z[0] == 0,
     t1[0] == 1, t2[0] == 0, t3[0] == 0,
     n1[0] == 0, n2[0] == 0, n3[0] == 1,
     b1[0] == 0, b2[0] == 1, b3[0] == 0},
    {x, y, z, t1, t2, t3, n1, n2, n3, b1, b2, b3, t1', t2', t3'},
    {ArcL, -8, 8}, {Lambda}, Method → {"ExplicitRungeKutta",
      "DifferenceOrder" → 5, "Coefficients" → DPCoefficients,
      "EmbeddedDifferenceOrder" → 4, "StepSizeControlParameters" -> {1, 0},
      "StepSizeSafetyFactors" → {1, 1}, "StepSizeRatioBounds" → {1 / 25, 25},
      "StiffnessTest" → False, "DiscontinuityProcessing" → False},
    MaxStepSize → 0.01, MaxSteps → Automatic]]

NOIteration[a_, b_, tol_] := Ceiling[N[1 / Log[2] Log[Abs[a - b] / tol]]];
LACvectorchord[Alpha_, InitRC_, Lambda_, s_] :=
  (Re[{x[Lambda][s], y[Lambda][s], z[Lambda][s]}] - {0, 0, 0}) /
    Sqrt[(Re[{x[Lambda][s], y[Lambda][s], z[Lambda][s]}] - {0, 0, 0}).
      (Re[{x[Lambda][s], y[Lambda][s], z[Lambda][s]}] - {0, 0, 0})] /.
   LACEquation2[Alpha, InitRC]
LACtangentA[Alpha_, InitRC_, Lambda_, s_] := {1, 0, 0}
LACtangentB[Alpha_, InitRC_, Lambda_, s_] :=
  Re[{t1[Lambda][s], t2[Lambda][s], t3[Lambda][s]}] /.
   LACEquation2[Alpha, InitRC]
ThetaA[Alpha_, InitRC_, Lambda_, s_] := ArcCos[
   LACvectorchord[Alpha, InitRC, Lambda, s].LACtangentA[Alpha, InitRC, Lambda, s]]
ThetaB[Alpha_, InitRC_, Lambda_, s_] := ArcCos[
```

```
    LACvectorchord[Alpha, InitRC, Lambda, s].LACtangentB[Alpha, InitRC, Lambda, s]]

  VectorBetweenTwoPoints[PointA_, PointB_] :=
   (PointB - PointA) / Sqrt[(PointB - PointA).(PointB - PointA)]
  AngleAlphaofTwoPoint[TangentA_, TangentB_, PointA_, PointB_] :=
   ArcCos[TangentA.VectorBetweenTwoPoints[PointA, PointB]]
  AngleBetaofTwoPoint[TangentA_, TangentB_, PointA_, PointB_] :=
   ArcCos[TangentB.VectorBetweenTwoPoints[PointA, PointB]]

  BisectionMethodforLAC[Alpha_,
    InitRC_, {TangentA_, TangentB_, PointA_, PointB_}] :=
   Module[{i = 1, j, k, lamb0, lamb1, sE, lambE, Alph = Alpha, IRC = InitRC,
     AngA = AngleAlphaofTwoPoint[TangentA, TangentB, PointA, PointB], AngB =
      AngleBetaofTwoPoint[TangentA, TangentB, PointA, PointB], tole0 = 10^-16},

    k = 1;
    lamb0 = 0;
    lamb1 = 14;
    While[k < NOIteration[lamb0, lamb1, tole0], {lambE = (lamb0 + lamb1) / 2;
      If[ThetaA[Alph, IRC, lambE, ArcLengths[Alph, IRC, lambE, AngA + AngB]] < AngA,
       lamb0 = lambE, lamb1 = lambE]};
     k++];

    sE = ArcLengths[Alph, IRC, lambE, AngA + AngB];

    N[{lambE, sE}]]
```

Bisection for point 00 to 03(MAX)

```
BisectionMethodforLAC[2, 1,
 {MaxTangentPoint00, MaxTangentPoint03, Point00, Point03}]
{12.7184, 6.76436}
```

Bisection for point 00 to 30(MIN)

```
BisectionMethodforLAC[2, 1,
 {MinTangentPoint00, MinTangentPoint30, Point00, Point30}]
{12.7184, 6.76436}
```

Log Aesthetic Space Curves-Downhill Simplex

General Equations

```
DPbval = {{1 / 5}, {3 / 40, 9 / 40}, {44 / 45, -56 / 15, 32 / 9},
   {19 372 / 6561, -25 360 / 2187, 64 448 / 6561, -212 / 729},
   {9017 / 3168, -355 / 33, 46 732 / 5247, 49 / 176, -5103 / 18 656},
   {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84}};
DPcval = {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84, 0};
DPaval = {1 / 5, 3 / 10, 4 / 5, 8 / 9, 1, 1};
DPerrval =
  {71 / 57 600, 0, -71 / 16 695, 71 / 1920, -17 253 / 339 200, 22 / 525, -1 / 40};
DPCoefficients[5, p_] :=
  N[{DPbval, DPcval, DPaval, DPerrval}, p];
CurvatureLASC[ArcL_, Alpha_, Lambda_, InitRC_] :=
 Piecewise[{{1 / Exp[Lambda * ArcL + Log[InitRC]], Alpha == 0}},
  (InitRC ^ (Alpha) + Lambda * Alpha * ArcL) ^ (-1 / Alpha)]
TorsionLASC[ArcL_, BetaT_, Omega_, InitRT_] :=
 Piecewise[{{1 / Exp[Omega * ArcL + Log[InitRT]], BetaT == 0}},
  (InitRT ^ (BetaT) + Omega * BetaT * ArcL) ^ (-1 / BetaT)]

LASCEquation[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_] :=
 Flatten[NDSolve[{x'[ArcL] == t1[ArcL], y'[ArcL] == t2[ArcL], z'[ArcL] == t3[ArcL],
     t1'[ArcL] == CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × n1[ArcL],
     t2'[ArcL] == CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × n2[ArcL],
     t3'[ArcL] == CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × n3[ArcL],
     n1'[ArcL] == -CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × t1[ArcL] +
       TorsionLASC[ArcL, BetaT, Omega, InitRT] × b1[ArcL],
     n2'[ArcL] == -CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × t2[ArcL] +
       TorsionLASC[ArcL, BetaT, Omega, InitRT] × b2[ArcL],
     n3'[ArcL] == -CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × t3[ArcL] +
       TorsionLASC[ArcL, BetaT, Omega, InitRT] × b3[ArcL],
     b1'[ArcL] == -TorsionLASC[ArcL, BetaT, Omega, InitRT] × n1[ArcL],
     b2'[ArcL] == -TorsionLASC[ArcL, BetaT, Omega, InitRT] × n2[ArcL],
     b3'[ArcL] == -TorsionLASC[ArcL, BetaT, Omega, InitRT] × n3[ArcL],
     x[0] == 0, y[0] == 0, z[0] == 0,
     t1[0] == 1, t2[0] == 0, t3[0] == 0,
     n1[0] == 0, n2[0] == 0, n3[0] == 1,
     b1[0] == 0, b2[0] == 1, b3[0] == 0},
    {x, y, z, t1, t2, t3, n1, n2, n3, b1, b2, b3, t1', t2', t3'},
    {ArcL, -8, 8}, Method → {"ExplicitRungeKutta",
      "DifferenceOrder" → 5, "Coefficients" → DPCoefficients,
      "EmbeddedDifferenceOrder" → 4, "StepSizeControlParameters" -> {1, 0},
      "StepSizeSafetyFactors" → {1, 1}, "StepSizeRatioBounds" → {1 / 25, 25},
      "StiffnessTest" → False, "DiscontinuityProcessing" → False},
    MaxStepSize → 0.01, MaxSteps → Automatic]]
LASCEquation2[Alpha_, BetaT_, Omega_, InitRC_] := Flatten[ParametricNDSolve[
    {x'[ArcL] == t1[ArcL], y'[ArcL] == t2[ArcL], z'[ArcL] == t3[ArcL],
     t1'[ArcL] == CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × n1[ArcL],
     t2'[ArcL] == CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × n2[ArcL],
     t3'[ArcL] == CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × n3[ArcL],
```

```
    n1'[ArcL] == -CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × t1[ArcL] +
      TorsionLASC[ArcL, BetaT, Omega, InitRT] × b1[ArcL],
    n2'[ArcL] == -CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × t2[ArcL] +
      TorsionLASC[ArcL, BetaT, Omega, InitRT] × b2[ArcL],
    n3'[ArcL] == -CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × t3[ArcL] +
      TorsionLASC[ArcL, BetaT, Omega, InitRT] × b3[ArcL],
    b1'[ArcL] == -TorsionLASC[ArcL, BetaT, Omega, InitRT] × n1[ArcL],
    b2'[ArcL] == -TorsionLASC[ArcL, BetaT, Omega, InitRT] × n2[ArcL],
    b3'[ArcL] == -TorsionLASC[ArcL, BetaT, Omega, InitRT] × n3[ArcL],
    x[0] == 0, y[0] == 0, z[0] == 0,
    t1[0] == 1, t2[0] == 0, t3[0] == 0,
    n1[0] == 0, n2[0] == 0, n3[0] == 1,
    b1[0] == 0, b2[0] == 1, b3[0] == 0},
   {x, y, z, t1, t2, t3, n1, n2, n3, b1, b2, b3, t1', t2', t3'},
   {ArcL, -8, 8}, {Lambda, InitRT}, Method → {"ExplicitRungeKutta",
     "DifferenceOrder" → 5, "Coefficients" → DPCoefficients,
     "EmbeddedDifferenceOrder" → 4, "StepSizeControlParameters" -> {1, 0},
     "StepSizeSafetyFactors" → {1, 1}, "StepSizeRatioBounds" → {1 / 25, 25},
     "StiffnessTest" → False, "DiscontinuityProcessing" → False},
   MaxStepSize → 0.01, MaxSteps → Automatic]]
```

```
NoArcCosThetaB[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_, s_] :=
 vectorchord[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s].
  tangentB[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s]
NormalVectorB[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_, s_] :=
 Cross[vectorchord[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s],
    tangentA[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s]] /
  Sqrt[Cross[vectorchord[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s],
      tangentA[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s]].
     Cross[vectorchord[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s],
      tangentA[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s]]]
NoArcSinThetaC[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_, s_] :=
 Abs[tangentB[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s].
    NormalVectorB[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s]]
tangentB[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_, s_] :=
 Re[{t1[s], t2[s], t3[s]}] /.
  LASCEquation[Alpha, BetaT, Lambda, Omega, InitRC, InitRT]
AngleAlphaBetaofTwoPoint[TangentA_, TangentB_, PointA_, PointB_] :=
 {ArcCos[TangentA.VectorBetweenTwoPoints[PointA, PointB]],
  ArcCos[TangentB.VectorBetweenTwoPoints[PointA, PointB]]}
AngleBetaofTwoPoint[TangentA_, TangentB_, PointA_, PointB_] :=
 ArcCos[TangentB.VectorBetweenTwoPoints[PointA, PointB]]
```

```
NOIteration[a_, b_, tol_] := Ceiling[N[1 / Log[2] Log[Abs[a - b] / tol]]];
vectorchord[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_, s_] :=
 (Re[{x[Lambda, InitRT][s], y[Lambda, InitRT][s], z[Lambda, InitRT][s]}] -
```

```
    {0, 0, 0}) / Sqrt[(Re[{x[Lambda, InitRT][s],
        y[Lambda, InitRT][s], z[Lambda, InitRT][s]}] - {0, 0, 0}).
    (Re[{x[Lambda, InitRT][s], y[Lambda, InitRT][s], z[Lambda, InitRT][s]}] -
      {0, 0, 0})] /. LASCEquation2[Alpha, BetaT, Omega, InitRC]
tangentA[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_, s_] := {1, 0, 0}

NoArcCosThetaA[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_, s_] :=
 vectorchord[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s].
  tangentA[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s]



RotateAngle1TAtoXYPlane[TangentA_] :=
 Piecewise[{{-ArcCos[0], TangentA == {1, 0, 0} || TangentA == {-1, 0, 0}}},
  Piecewise[{{-ArcCos[{0, 1, 0}.({0, TangentA[[2]], TangentA[[3]]} /
          Sqrt[{0, TangentA[[2]], TangentA[[3]]}.
            {0, TangentA[[2]], TangentA[[3]]}])], Sign[TangentA[[3]]] == 1}},
   ArcCos[{0, 1, 0}.({0, TangentA[[2]], TangentA[[3]]} / Sqrt[{0,
          TangentA[[2]], TangentA[[3]]}.{0, TangentA[[2]], TangentA[[3]]}])]]]
TangentAonXYPlane[TangentA_] := RotationTransform[
   RotateAngle1TAtoXYPlane[TangentA], {1, 0, 0}][TangentA]
TangentBAfterRotateAngle1[TangentA_, TangentB_] :=
 RotationTransform[RotateAngle1TAtoXYPlane[TangentA], {1, 0, 0}][TangentB]
PointBAfterRotateAngle1[TangentA_, PointA_, PointB_] :=
 RotationTransform[RotateAngle1TAtoXYPlane[TangentA], {1, 0, 0}][PointB - PointA]

RotateAngle2TAtoXAxis[TangentA_] :=
 Piecewise[{{-ArcCos[{1, 0, 0}.TangentAonXYPlane[TangentA]],
     Sign[TangentAonXYPlane[TangentA][[2]]] == 1}},
  ArcCos[{1, 0, 0}.TangentAonXYPlane[TangentA]]]
TangentAonXAxis[TangentA_] := RotationTransform[
   RotateAngle2TAtoXAxis[TangentA], {0, 0, 1}][TangentAonXYPlane[TangentA]]
TangentBAfterRotateAngle2[TangentA_, TangentB_] :=
 RotationTransform[RotateAngle2TAtoXAxis[TangentA], {0, 0, 1}][
  TangentBAfterRotateAngle1[TangentA, TangentB]]
PointBAfterRotateAngle2[TangentA_, PointA_, PointB_] :=
 RotationTransform[RotateAngle2TAtoXAxis[TangentA], {0, 0, 1}][
  PointBAfterRotateAngle1[TangentA, PointA, PointB]]

RotateAngle3PointBtoXYPlane[TangentA_, PointA_, PointB_] := Piecewise[
  {{-ArcCos[{0, 1, 0}.({0, PointBAfterRotateAngle2[TangentA, PointA, PointB][[
          2]], PointBAfterRotateAngle2[TangentA, PointA, PointB][[3]]} /
        Sqrt[{0, PointBAfterRotateAngle2[TangentA, PointA, PointB][[2]],
           PointBAfterRotateAngle2[TangentA, PointA, PointB][[3]]}.
          {0, PointBAfterRotateAngle2[TangentA, PointA, PointB][[2]],
           PointBAfterRotateAngle2[TangentA, PointA, PointB][[3]]}])],
    Sign[PointBAfterRotateAngle2[TangentA, PointA, PointB][[3]]] == 1}},
```

```
ArcCos[{0, 1, 0}.({0, PointBAfterRotateAngle2[TangentA, PointA, PointB][[2]],
     PointBAfterRotateAngle2[TangentA, PointA, PointB][[3]]} /
    Sqrt[{0, PointBAfterRotateAngle2[TangentA, PointA, PointB][[2]],
       PointBAfterRotateAngle2[TangentA, PointA, PointB][[3]]}.
      {0, PointBAfterRotateAngle2[TangentA, PointA, PointB][[2]],
       PointBAfterRotateAngle2[TangentA, PointA, PointB][[3]]}])]]

TangentBAfterRotateAngle3[TangentA_, TangentB_, PointA_, PointB_] :=
 RotationTransform[RotateAngle3PointBtoXYPlane[TangentA, PointA, PointB],
   {1, 0, 0}][TangentBAfterRotateAngle2[TangentA, TangentB]]
PointBAfterRotateAngle3[TangentA_, TangentB_, PointA_, PointB_] :=
 RotationTransform[RotateAngle3PointBtoXYPlane[TangentA, PointA, PointB],
   {1, 0, 0}][PointBAfterRotateAngle2[TangentA, PointA, PointB]]

VectorBetweenTwoPoints[PointA_, PointB_] :=
 (PointB - PointA) / Sqrt[(PointB - PointA).(PointB - PointA)]

AngleAlphaofTwoPoint[TangentA_, TangentB_, PointA_, PointB_] :=
 ArcCos[TangentA.VectorBetweenTwoPoints[PointA, PointB]]

AngleTangentBtoXYPlane[TangentA_, TangentB_, PointA_, PointB_] :=
 ArcSin[TangentB.(Cross[TangentA, VectorBetweenTwoPoints[PointA, PointB]] /
     Sqrt[Cross[TangentA, VectorBetweenTwoPoints[PointA, PointB]].
       Cross[TangentA, VectorBetweenTwoPoints[PointA, PointB]]])]
PointBonXYPlane[TangentA_, TangentB_, PointA_, PointB_] :=
 Piecewise[{{PointBAfterRotateAngle3[TangentA, TangentB, PointA, PointB], Sign[
      PointBAfterRotateAngle3[TangentA, TangentB, PointA, PointB][[1]]] == 1}},
  {-1, 1, 1} * PointBAfterRotateAngle3[TangentA, TangentB, PointA, PointB]]
TangentPointBonXYPlane[TangentA_, TangentB_, PointA_, PointB_] :=
 Piecewise[{{TangentBAfterRotateAngle3[TangentA, TangentB, PointA, PointB], Sign[
      PointBAfterRotateAngle3[TangentA, TangentB, PointA, PointB][[1]]] == 1}},
  {1, -1, 1} * TangentBAfterRotateAngle3[TangentA, TangentB, PointA, PointB]]
TangentPointBonXYPlanewithNegz[TangentA_, TangentB_, PointA_, PointB_] :=
 Piecewise[
  {{{1, 1, -1} * TangentPointBonXYPlane[TangentA, TangentB, PointA, PointB],
    Sign[TangentPointBonXYPlane[TangentA, TangentB, PointA, PointB][[3]]] == 1}},
  TangentPointBonXYPlane[TangentA, TangentB, PointA, PointB]]

Angle1PointEndtoXYPlane[
  {Alpha_, BetaT_, Omega_, InitRC_}, {Lambda_, InitRT_, s_}] :=
 ArcCos[{0, 1, 0}.((Re[{0, y[Lambda, InitRT][s], z[Lambda, InitRT][s]}]) /
      Sqrt[(Re[{0, y[Lambda, InitRT][s], z[Lambda, InitRT][s]}].
         Re[{0, y[Lambda, InitRT][s], z[Lambda, InitRT][s]}])]) /.
   LASCEquation2[Alpha, BetaT, Omega, InitRC]]

IdealTangent[{Alpha_, BetaT_, Omega_, InitRC_}, {Lambda_, InitRT_, s_}] :=
 RotationTransform[-Angle1PointEndtoXYPlane[
```

```
      {Alpha, BetaT, Omega, InitRC}, {Lambda, InitRT, s}], {1, 0, 0}][
   (Re[{t1[Lambda, InitRT][s], t2[Lambda, InitRT][s], t3[Lambda, InitRT][s]}] /.
     LASCEquation2[Alpha, BetaT, Omega, InitRC])]

TangentX[{Alpha_, BetaT_, Omega_, InitRC_}, {Lambda_, InitRT_, s_}] :=
 RotationTransform[-Angle1PointEndtoXYPlane[
      {Alpha, BetaT, Omega, InitRC}, {Lambda, InitRT, s}], {1, 0, 0}][
   (Re[{t1[Lambda, InitRT][s], t2[Lambda, InitRT][s], t3[Lambda, InitRT][s]}] /.
     LASCEquation2[Alpha, BetaT, Omega, InitRC])]

BisectionMethodtoFindArcLength[Alpha_, Beta_, Omega_,
  InitRC_, Lambda_, IRT_, {TangentA_, TangentB_, PointA_, PointB_}] :=
 Module[{i = 1, j, k, s0, s1, lamb0, lamb1, sE, Bet = Beta, Omeg = Omega,
    iRTE = IRT, lambE = Lambda, Alph = Alpha, IRC = InitRC, AngA =
     AngleAlphaofTwoPoint[TangentA, TangentB, PointA, PointB], tole0 = 10^-16},

  j = 1;
  s0 = 0;
  s1 = 4;
  While[j < NOIteration[s0, s1, tole0], {sE = (s0 + s1) / 2;
     If[ArcCos[NoArcCosThetaA[Alph, Bet, lambE, Omeg, IRC, iRTE, sE]] < AngA,
      s0 = sE, s1 = sE]};
   j++];

  N[sE]]


MinimizeFunction[{Alpha_, BetaT_, Omega_, InitRC_},
  {Lambda_, InitRT_, s_}, {TangentA_, TangentB_, PointA_, PointB_}] :=
 Abs[TangentX[{Alpha, BetaT, Omega, InitRC}, {Lambda, InitRT, s}].
     TangentPointBonXYPlanewithNegz[TangentA, TangentB, PointA, PointB] - 1]

NelderMeadDownhill[Alpha_, Beta_, Omega_, InitRC_, s_, MaxIteration_,
  {Pointx1_, Pointx2_, Pointx3_}, {TangentA_, TangentB_, PointA_, PointB_}] :=
 Quiet[Module[{i = 1, x1 = Pointx1, x2 = Pointx2, x3 = Pointx3, Alph = Alpha,
     Bet = Beta, Omeg = Omega, IRC = InitRC, xnew, Minimizex1, Minimizex2,
     Minimizex3, xBest, xSecond, xWorst, xMiddle, xReflection, xExpansion,
     xContraction, xContractionReflection, MinimizexBest, MinimizexSecond,
     MinimizexWorst, MinimizexReflection, MinimizexExpansion,
     MinimizexContraction, MinimizexContractionReflection},

    While[i ≤ MaxIteration,

     Minimizex1 = MinimizeFunction[{Alph, Bet, Omeg, IRC},
       {x1[[1]], x1[[2]], s}, {TangentA, TangentB, PointA, PointB}];
     Minimizex2 = MinimizeFunction[{Alph, Bet, Omeg, IRC},
       {x2[[1]], x2[[2]], s}, {TangentA, TangentB, PointA, PointB}];
```

```
Minimizex3 = MinimizeFunction[{Alph, Bet, Omeg, IRC},
   {x3[[1]], x3[[2]], s}, {TangentA, TangentB, PointA, PointB}];

xBest = Piecewise[{{x1, Minimizex1 < Minimizex2 && Minimizex1 < Minimizex3},
    {x2, Minimizex2 < Minimizex1 && Minimizex2 < Minimizex3},
    {x3, Minimizex3 < Minimizex1 && Minimizex3 < Minimizex2}}];

xSecond =
  Piecewise[{{x1, ((Minimizex1 < Minimizex2) && (Minimizex1 > Minimizex3)) ||
       ((Minimizex1 > Minimizex2) && (Minimizex1 < Minimizex3))},
    {x2, ((Minimizex2 < Minimizex1) && (Minimizex2 > Minimizex3)) ||
       ((Minimizex2 > Minimizex1) && (Minimizex2 < Minimizex3))},
    {x3, ((Minimizex3 < Minimizex1) && (Minimizex3 > Minimizex2)) ||
       ((Minimizex3 > Minimizex1) && (Minimizex3 < Minimizex2))}}];

xWorst = Piecewise[{{x1, Minimizex1 > Minimizex2 && Minimizex1 > Minimizex3},
    {x2, Minimizex2 > Minimizex1 && Minimizex2 > Minimizex3},
    {x3, Minimizex3 > Minimizex1 && Minimizex3 > Minimizex2}}];

xMiddle = (xBest + xSecond) / 2;
xReflection = xMiddle + 1 (xMiddle - xWorst);
xExpansion = xMiddle + 2 (xReflection - xMiddle);
xContraction = xMiddle + .5 (xWorst - xMiddle);
xContractionReflection = xMiddle + .5 (xReflection - xMiddle);


MinimizexBest = MinimizeFunction[{Alph, Bet, Omeg, IRC},
   {xBest[[1]], xBest[[2]], s}, {TangentA, TangentB, PointA, PointB}];
MinimizexSecond = MinimizeFunction[{Alph, Bet, Omeg, IRC},
   {xSecond[[1]], xSecond[[2]], s}, {TangentA, TangentB, PointA, PointB}];
MinimizexWorst = MinimizeFunction[{Alph, Bet, Omeg, IRC},
   {xWorst[[1]], xWorst[[2]], s}, {TangentA, TangentB, PointA, PointB}];
MinimizexReflection = MinimizeFunction[{Alph, Bet, Omeg, IRC},
   {xReflection[[1]], xReflection[[2]], s},
   {TangentA, TangentB, PointA, PointB}];
MinimizexExpansion = MinimizeFunction[{Alph, Bet, Omeg, IRC}, {xExpansion[[
     1]], xExpansion[[2]], s}, {TangentA, TangentB, PointA, PointB}];
MinimizexContraction = MinimizeFunction[{Alph, Bet, Omeg, IRC},
   {xContraction[[1]], xContraction[[2]], s},
   {TangentA, TangentB, PointA, PointB}];
MinimizexContractionReflection = MinimizeFunction[{Alph, Bet, Omeg, IRC},
   {xContractionReflection[[1]], xContractionReflection[[2]], s},
   {TangentA, TangentB, PointA, PointB}];


xnew = Piecewise[
   {{Piecewise[{{xExpansion, MinimizexExpansion ≤ MinimizexReflection}},
```

```
        xReflection], MinimizexReflection < MinimizexBest}},
      Piecewise[{{xReflection, MinimizexReflection ≤ MinimizexSecond}},
       Piecewise[{{Piecewise[{{xReflection, MinimizexReflection ≤
               MinimizexContractionReflection}}, xContractionReflection],
          MinimizexReflection ≤ MinimizexWorst}}, Piecewise[
         {{xWorst, MinimizexWorst ≤ MinimizexContraction}}, xContraction]]]];
    x1 = Sign[xnew] * xnew;
    x2 = xBest;
    x3 = xSecond;


    i++];

   N[{x1, x2, x3}]]]
```

```
Point00 = {0, 0, 0};(*According to Point*)
Point03 = {-0.7077264466498845, 0, 0.5008767232876717};
Point30 = {0, -0.7077264466498845, -0.5008767232876717};
Point33 = {-1.0013149889663, -1.0013149889663, 0};

MaxTangentPoint00 = {-1, 0, 0};(*According to LoC*)
MaxTangentPoint03 = {-0.5770131138020663, 0, 0.8167348814030436};
MaxTangentPoint30 = {-1, 0, 0};
MaxTangentPoint33 =
  {-0.6664720667513934, -0.3335279332486068, 0.6667639027294741};

MinTangentPoint00 = {0, -1, 0};
MinTangentPoint30 = {0., -0.5770131138020663, -0.8167348814030436};
MinTangentPoint03 = {0, -1, 0};
MinTangentPoint33 =
  {-0.3335279332486068, -0.6664720667513934, -0.6667639027294741};
```

DHS for point 30 to 33(MAX)

4th Iteration with s =P30P33ArcLength4

```
LambdaAndInitRT4P30P33 =
 AbsoluteTiming[NelderMeadDownhill[2, 2, 1, 1, 3.108419254422188,
   25, {{6.28520, 7.301}, {6.28521, 7.303}, {6.28519, 7.305}},
   {MaxTangentPoint10, MaxTangentPoint11, Point10, Point11}]]
```

```
{121.673, {{6.28525, 7.30313}, {6.28524, 7.30326}, {6.28524, 7.3032}}}
```

```
{6.285247558593747`, 7.303131835937503`, 3.1084204465150833`}
```

```
P30P33ArcLength4t1 = BisectionMethodtoFindArcLength[2, 2, 1, 1,
  LambdaAndInitRT4P30P33[[2, 1, 1]], LambdaAndInitRT4P30P33[[2, 1, 2]],
  {MaxTangentPoint10, MaxTangentPoint11, Point10, Point11}]
P30P33ArcLength4t2 = BisectionMethodtoFindArcLength[2, 2, 1, 1,
  LambdaAndInitRT4P30P33[[2, 2, 1]], LambdaAndInitRT4P30P33[[2, 2, 2]],
  {MaxTangentPoint10, MaxTangentPoint11, Point10, Point11}]
```

3.10842

3.10842

```
MinimizeFunction[{2, 2, 1, 1}, {LambdaAndInitRT4P30P33[[2, 1, 1]],
  LambdaAndInitRT4P30P33[[2, 1, 2]], P30P33ArcLength4t1},
 {MaxTangentPoint10, MaxTangentPoint11, Point10, Point11}]
MinimizeFunction[{2, 2, 1, 1}, {LambdaAndInitRT4P30P33[[2, 2, 1]],
  LambdaAndInitRT4P30P33[[2, 2, 2]], P30P33ArcLength4t2},
 {MaxTangentPoint10, MaxTangentPoint11, Point10, Point11}]
```

6.59472×10^{-14}

4.81393×10^{-13}

DHS for point 03 to 33(MIN)

4th Iteration with s =P03P33ArcLength4

```
LambdaAndInitRT4P03P33 =
 AbsoluteTiming[NelderMeadDownhill[2, 2, 1, 1, 3.108419254422188,
   25, {{6.28520, 7.301}, {6.28521, 7.303}, {6.28519, 7.305}},
   {MinTangentPoint01, MinTangentPoint11, Point01, Point11}]]
```

{118.867, {{6.28525, 7.30313}, {6.28524, 7.30326}, {6.28524, 7.3032}}}

{6.285247558593747`, 7.303131835937503`, 3.1084204465150833`}

```
P03P33ArcLength4t1 = BisectionMethodtoFindArcLength[2, 2, 1, 1,
  LambdaAndInitRT4P03P33[[2, 1, 1]], LambdaAndInitRT4P03P33[[2, 1, 2]],
  {MinTangentPoint01, MinTangentPoint11, Point01, Point11}]
P03P33ArcLength4t2 = BisectionMethodtoFindArcLength[2, 2, 1, 1,
  LambdaAndInitRT4P03P33[[2, 2, 1]], LambdaAndInitRT4P03P33[[2, 2, 2]],
  {MinTangentPoint01, MinTangentPoint11, Point01, Point11}]
```

3.10842

3.10842

```
MinimizeFunction[{2, 2, 1, 1}, {LambdaAndInitRT4P03P33[[2, 1, 1]],
   LambdaAndInitRT4P03P33[[2, 1, 2]], P03P33ArcLength4t1},
  {MinTangentPoint01, MinTangentPoint11, Point01, Point11}]
MinimizeFunction[{2, 2, 1, 1}, {LambdaAndInitRT4P03P33[[2, 2, 1]],
   LambdaAndInitRT4P03P33[[2, 2, 2]], P03P33ArcLength4t2},
  {MinTangentPoint01, MinTangentPoint11, Point01, Point11}]
```

6.59472×10^{-14}

4.81393×10^{-13}

Table 1

Calculate vectors of LAC and LASC after inverse transformation

LAC

```
Clear["Global`*"]

Point00 = {0, 0, 0};(*According to Point*)
Point03 = {-0.7077264466498845, 0, 0.5008767232876717};
Point30 = {0, -0.7077264466498845, -0.5008767232876717};
Point33 = {-1.0013149889663, -1.0013149889663, 0};

MaxTangentPoint00 = {-1, 0, 0};(*According to LoC*)
MaxTangentPoint03 = {-0.5770131138020663, 0, 0.8167348814030436};
MaxTangentPoint30 = {-1, 0, 0};
MaxTangentPoint33 =
  {-0.6664720667513934, -0.3335279332486068, 0.6667639027294741};

MinTangentPoint00 = {0, -1, 0};
MinTangentPoint30 = {0., -0.5770131138020663, -0.8167348814030436};
MinTangentPoint03 = {0, -1, 0};
MinTangentPoint33 =
  {-0.3335279332486068, -0.6664720667513934, -0.6667639027294741};

DPbval = {{1 / 5}, {3 / 40, 9 / 40}, {44 / 45, -56 / 15, 32 / 9},
    {19 372 / 6561, -25 360 / 2187, 64 448 / 6561, -212 / 729},
    {9017 / 3168, -355 / 33, 46 732 / 5247, 49 / 176, -5103 / 18 656},
    {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84}};
DPcval = {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84, 0};
DPaval = {1 / 5, 3 / 10, 4 / 5, 8 / 9, 1, 1};
DPerrval =
  {71 / 57 600, 0, -71 / 16 695, 71 / 1920, -17 253 / 339 200, 22 / 525, -1 / 40};
DPCoefficients[5, p_] :=
```

```
  N[{DPbval, DPcval, DPaval, DPerrval}, p];
CurvatureLASC[ArcL_, Alpha_, Lambda_, InitRC_] :=
 Piecewise[{{1 / Exp[Lambda * ArcL + Log[InitRC]], Alpha == 0}},
  (InitRC ^ (Alpha) + Lambda * Alpha * ArcL) ^ (-1 / Alpha)]
TorsionLASC[ArcL_, BetaT_, Omega_, InitRT_] :=
 Piecewise[{{1 / Exp[Omega * ArcL + Log[InitRT]], BetaT == 0}},
  (InitRT ^ (BetaT) + Omega * BetaT * ArcL) ^ (-1 / BetaT)]

LASCEquation3[Alpha_, BetaT_, Omega_, InitRC_] := Flatten[ParametricNDSolve[
   {x'[ArcL] == t1[ArcL], y'[ArcL] == t2[ArcL], z'[ArcL] == t3[ArcL],
    t1'[ArcL] == CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × n1[ArcL],
    t2'[ArcL] == CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × n2[ArcL],
    t3'[ArcL] == CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × n3[ArcL],
    n1'[ArcL] ==
     -CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × t1[ArcL] + 0 * b1[ArcL],
    n2'[ArcL] == -CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × t2[ArcL] +
      0 * b2[ArcL],
    n3'[ArcL] == -CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × t3[ArcL] +
      0 * b3[ArcL],
    b1'[ArcL] == -0 * n1[ArcL],
    b2'[ArcL] == -0 * n2[ArcL],
    b3'[ArcL] == -0 * n3[ArcL],
    x[0] == 0, y[0] == 0, z[0] == 0,
    t1[0] == 1, t2[0] == 0, t3[0] == 0,
    n1[0] == 0, n2[0] == 0, n3[0] == 1,
    b1[0] == 0, b2[0] == 1, b3[0] == 0},
   {x, y, z, t1, t2, t3, n1, n2, n3, b1, b2, b3, t1', t2', t3'},
   {ArcL, -8, 8}, {Lambda, InitRT}, Method → {"ExplicitRungeKutta",
     "DifferenceOrder" → 5, "Coefficients" → DPCoefficients,
     "EmbeddedDifferenceOrder" → 4, "StepSizeControlParameters" -> {1, 0},
     "StepSizeSafetyFactors" → {1, 1}, "StepSizeRatioBounds" → {1 / 25, 25},
     "StiffnessTest" → False, "DiscontinuityProcessing" → False},
   MaxStepSize → 0.01, MaxSteps → Automatic]]
NoArcCosThetaB[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_, s_] :=
 vectorchord[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s].
  tangentB[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s]
NormalVectorB[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_, s_] :=
 Cross[vectorchord[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s],
   tangentA[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s]] /
  Sqrt[Cross[vectorchord[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s],
     tangentA[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s]].
    Cross[vectorchord[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s],
     tangentA[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s]]]
NoArcSinThetaC[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_, s_] :=
 Abs[tangentB[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s].
   NormalVectorB[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s]]
tangentB[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_, s_] :=
```

```
 Re[{t1[Lambda, InitRT][s], t2[Lambda, InitRT][s], t3[Lambda, InitRT][s]}] /.
  LASCEquation3[Alpha, BetaT, Omega, InitRC]
AngleAlphaBetaofTwoPoint[TangentA_, TangentB_, PointA_, PointB_] :=
 {ArcCos[TangentA.VectorBetweenTwoPoints[PointA, PointB]],
  ArcCos[TangentB.VectorBetweenTwoPoints[PointA, PointB]]}
AngleBetaofTwoPoint[TangentA_, TangentB_, PointA_, PointB_] :=
 ArcCos[TangentB.VectorBetweenTwoPoints[PointA, PointB]]
NOIteration[a_, b_, tol_] := Ceiling[N[1 / Log[2] Log[Abs[a - b] / tol]]];
vectorchord[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_, s_] :=
 (Re[{x[Lambda, InitRT][s], y[Lambda, InitRT][s], z[Lambda, InitRT][s]}] -
     {0, 0, 0}) / Sqrt[(Re[{x[Lambda, InitRT][s],
         y[Lambda, InitRT][s], z[Lambda, InitRT][s]}] - {0, 0, 0}).
     (Re[{x[Lambda, InitRT][s], y[Lambda, InitRT][s], z[Lambda, InitRT][s]}] -
       {0, 0, 0})] /. LASCEquation3[Alpha, BetaT, Omega, InitRC]
tangentA[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_, s_] := {1, 0, 0}

NoArcCosThetaA[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_, s_] :=
 vectorchord[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s].
  tangentA[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s]
```

```
RotateAngle1TAtoXYPlane[TangentA_] :=
 Piecewise[{{-ArcCos[0], TangentA == {1, 0, 0} || TangentA == {-1, 0, 0}}},
  Piecewise[{{-ArcCos[{0, 1, 0}.({0, TangentA[[2]], TangentA[[3]]} /
          Sqrt[{0, TangentA[[2]], TangentA[[3]]}.
            {0, TangentA[[2]], TangentA[[3]]}])], Sign[TangentA[[3]]] == 1}},
   ArcCos[{0, 1, 0}.({0, TangentA[[2]], TangentA[[3]]} / Sqrt[{0,
          TangentA[[2]], TangentA[[3]]}.{0, TangentA[[2]], TangentA[[3]]}])]]]
TangentAonXYPlane[TangentA_] := RotationTransform[
   RotateAngle1TAtoXYPlane[TangentA], {1, 0, 0}][TangentA]
TangentBAfterRotateAngle1[TangentA_, TangentB_] :=
 RotationTransform[RotateAngle1TAtoXYPlane[TangentA], {1, 0, 0}][TangentB]
PointBAfterRotateAngle1[TangentA_, PointA_, PointB_] :=
 RotationTransform[RotateAngle1TAtoXYPlane[TangentA], {1, 0, 0}][PointB - PointA]

RotateAngle2TAtoXAxis[TangentA_] :=
 Piecewise[{{-ArcCos[{1, 0, 0}.TangentAonXYPlane[TangentA]],
     Sign[TangentAonXYPlane[TangentA][[2]]] == 1}},
  ArcCos[{1, 0, 0}.TangentAonXYPlane[TangentA]]]
TangentAonXAxis[TangentA_] := RotationTransform[
   RotateAngle2TAtoXAxis[TangentA], {0, 0, 1}][TangentAonXYPlane[TangentA]]
TangentBAfterRotateAngle2[TangentA_, TangentB_] :=
 RotationTransform[RotateAngle2TAtoXAxis[TangentA], {0, 0, 1}][
  TangentBAfterRotateAngle1[TangentA, TangentB]]
PointBAfterRotateAngle2[TangentA_, PointA_, PointB_] :=
 RotationTransform[RotateAngle2TAtoXAxis[TangentA], {0, 0, 1}][
```

```
  PointBAfterRotateAngle1[TangentA, PointA, PointB]]

RotateAngle3PointBtoXYPlane[TangentA_, PointA_, PointB_] := Piecewise[
  {{-ArcCos[{0, 1, 0}.({0, PointBAfterRotateAngle2[TangentA, PointA, PointB][[
          2]], PointBAfterRotateAngle2[TangentA, PointA, PointB][[3]]} /
        Sqrt[{0, PointBAfterRotateAngle2[TangentA, PointA, PointB][[2]],
           PointBAfterRotateAngle2[TangentA, PointA, PointB][[3]]}.
          {0, PointBAfterRotateAngle2[TangentA, PointA, PointB][[2]],
           PointBAfterRotateAngle2[TangentA, PointA, PointB][[3]]}])],
    Sign[PointBAfterRotateAngle2[TangentA, PointA, PointB][[3]]] == 1}},
  ArcCos[{0, 1, 0}.({0, PointBAfterRotateAngle2[TangentA, PointA, PointB][[2]],
      PointBAfterRotateAngle2[TangentA, PointA, PointB][[3]]} /
     Sqrt[{0, PointBAfterRotateAngle2[TangentA, PointA, PointB][[2]],
        PointBAfterRotateAngle2[TangentA, PointA, PointB][[3]]}.
       {0, PointBAfterRotateAngle2[TangentA, PointA, PointB][[2]],
        PointBAfterRotateAngle2[TangentA, PointA, PointB][[3]]}])]]

TangentBAfterRotateAngle3[TangentA_, TangentB_, PointA_, PointB_] :=
 RotationTransform[RotateAngle3PointBtoXYPlane[TangentA, PointA, PointB],
   {1, 0, 0}][TangentBAfterRotateAngle2[TangentA, TangentB]]
PointBAfterRotateAngle3[TangentA_, TangentB_, PointA_, PointB_] :=
 RotationTransform[RotateAngle3PointBtoXYPlane[TangentA, PointA, PointB],
   {1, 0, 0}][PointBAfterRotateAngle2[TangentA, PointA, PointB]]

VectorBetweenTwoPoints[PointA_, PointB_] :=
 (PointB - PointA) / Sqrt[(PointB - PointA).(PointB - PointA)]

AngleAlphaofTwoPoint[TangentA_, TangentB_, PointA_, PointB_] :=
 ArcCos[TangentA.VectorBetweenTwoPoints[PointA, PointB]]

AngleTangentBtoXYPlane[TangentA_, TangentB_, PointA_, PointB_] :=
 ArcSin[TangentB.(Cross[TangentA, VectorBetweenTwoPoints[PointA, PointB]] /
     Sqrt[Cross[TangentA, VectorBetweenTwoPoints[PointA, PointB]].
       Cross[TangentA, VectorBetweenTwoPoints[PointA, PointB]]])]
PointBonXYPlane[TangentA_, TangentB_, PointA_, PointB_] :=
 Piecewise[{{PointBAfterRotateAngle3[TangentA, TangentB, PointA, PointB], Sign[
      PointBAfterRotateAngle3[TangentA, TangentB, PointA, PointB][[1]]] == 1}},
  {-1, 1, 1} * PointBAfterRotateAngle3[TangentA, TangentB, PointA, PointB]]
TangentPointBonXYPlane[TangentA_, TangentB_, PointA_, PointB_] :=
 Piecewise[{{TangentBAfterRotateAngle3[TangentA, TangentB, PointA, PointB], Sign[
      PointBAfterRotateAngle3[TangentA, TangentB, PointA, PointB][[1]]] == 1}},
  {1, -1, 1} * TangentBAfterRotateAngle3[TangentA, TangentB, PointA, PointB]]
TangentPointBonXYPlanewithNegz[TangentA_, TangentB_, PointA_, PointB_] :=
 Piecewise[
  {{{1, 1, -1} * TangentPointBonXYPlane[TangentA, TangentB, PointA, PointB],
    Sign[TangentPointBonXYPlane[TangentA, TangentB, PointA, PointB][[3]]] == 1}},
  TangentPointBonXYPlane[TangentA, TangentB, PointA, PointB]]
```

```
Angle1PointEndtoXYPlane[
  {Alpha_, BetaT_, Omega_, InitRC_}, {Lambda_, InitRT_, s_}] :=
 ArcCos[{0, 1, 0}.((Re[{0, y[Lambda, InitRT][s], z[Lambda, InitRT][s]}]) /
      Sqrt[(Re[{0, y[Lambda, InitRT][s], z[Lambda, InitRT][s]}].
         Re[{0, y[Lambda, InitRT][s], z[Lambda, InitRT][s]}])]) /.
   LASCEquation3[Alpha, BetaT, Omega, InitRC]]

IdealTangent[{Alpha_, BetaT_, Omega_, InitRC_}, {Lambda_, InitRT_, s_}] :=
 RotationTransform[-Angle1PointEndtoXYPlane[
     {Alpha, BetaT, Omega, InitRC}, {Lambda, InitRT, s}], {1, 0, 0}][
  (Re[{t1[Lambda, InitRT][s], t2[Lambda, InitRT][s], t3[Lambda, InitRT][s]}] /.
    LASCEquation3[Alpha, BetaT, Omega, InitRC])]

TangentX[{Alpha_, BetaT_, Omega_, InitRC_}, {Lambda_, InitRT_, s_}] :=
 RotationTransform[-Angle1PointEndtoXYPlane[
     {Alpha, BetaT, Omega, InitRC}, {Lambda, InitRT, s}], {1, 0, 0}][
  (Re[{t1[Lambda, InitRT][s], t2[Lambda, InitRT][s], t3[Lambda, InitRT][s]}] /.
    LASCEquation3[Alpha, BetaT, Omega, InitRC])]

SignofPoints[TangentA_, TangentB_, PointA_, PointB_] :=
 Piecewise[{{{1, 1, -1}, Sign[TangentPointBonXYPlane[
        TangentA, TangentB, PointA, PointB][[3]]] == 1}}, {1, 1, 1}]


TangentofCoonMaxLAC =
 RotationTransform[-RotateAngle1TAtoXYPlane[MaxTangentPoint00], {1, 0, 0}][
  RotationTransform[-RotateAngle2TAtoXAxis[MaxTangentPoint00], {0, 0, 1}][
   RotationTransform[-RotateAngle3PointBtoXYPlane[
       MaxTangentPoint00, Point00, Point03], {1, 0, 0}][
    SignofPoints[MaxTangentPoint00, MaxTangentPoint03, Point00, Point03] *
     RotationTransform[-Angle1PointEndtoXYPlane[{2, 2, 1, 1},
        {12.718433357775211, 1, 6.764357890136049`}], {1, 0, 0}][{1, 0, 0}]]]]
NormalofCoonMaxLAC = RotationTransform[-RotateAngle1TAtoXYPlane[
     MaxTangentPoint00], {1, 0, 0}][
  RotationTransform[-RotateAngle2TAtoXAxis[MaxTangentPoint00], {0, 0, 1}][
   RotationTransform[-RotateAngle3PointBtoXYPlane[
       MaxTangentPoint00, Point00, Point03], {1, 0, 0}][
    SignofPoints[MaxTangentPoint00, MaxTangentPoint03, Point00, Point03] *
     RotationTransform[-Angle1PointEndtoXYPlane[{2, 2, 1, 1},
        {12.718433357775211, 1, 6.764357890136049`}], {1, 0, 0}][{0, 0, 1}]]]]
BinormalofCoonMaxLAC = RotationTransform[-RotateAngle1TAtoXYPlane[
     MaxTangentPoint00], {1, 0, 0}][
  RotationTransform[-RotateAngle2TAtoXAxis[MaxTangentPoint00], {0, 0, 1}][
   RotationTransform[-RotateAngle3PointBtoXYPlane[
       MaxTangentPoint00, Point00, Point03], {1, 0, 0}][
    SignofPoints[MaxTangentPoint00, MaxTangentPoint03, Point00, Point03] *
     RotationTransform[-Angle1PointEndtoXYPlane[{2, 2, 1, 1},
```

```
        {12.718433357775211, 1, 6.764357890136049`}], {1, 0, 0}][{0, 1, 0}]]]]
TangentofCoonMinLAC = RotationTransform[-RotateAngle1TAtoXYPlane[
     MinTangentPoint00], {1, 0, 0}][
  RotationTransform[-RotateAngle2TAtoXAxis[MinTangentPoint00], {0, 0, 1}][
   RotationTransform[-RotateAngle3PointBtoXYPlane[
       MinTangentPoint00, Point00, Point30], {1, 0, 0}][
    SignofPoints[MaxTangentPoint00, MinTangentPoint30, Point00, Point30] *
     RotationTransform[-Angle1PointEndtoXYPlane[{2, 2, 1, 1},
         {12.718433357775211, 1, 6.764357890136049`}], {1, 0, 0}][{1, 0, 0}]]]]
NormalofCoonMinLAC = RotationTransform[-RotateAngle1TAtoXYPlane[
     MinTangentPoint00], {1, 0, 0}][
  RotationTransform[-RotateAngle2TAtoXAxis[MinTangentPoint00], {0, 0, 1}][
   RotationTransform[-RotateAngle3PointBtoXYPlane[
       MinTangentPoint00, Point00, Point30], {1, 0, 0}][
    SignofPoints[MaxTangentPoint00, MinTangentPoint30, Point00, Point30] *
     RotationTransform[-Angle1PointEndtoXYPlane[{2, 2, 1, 1},
         {12.718433357775211, 1, 6.764357890136049`}], {1, 0, 0}][{0, 0, 1}]]]]
BinormalofCoonMinLAC = RotationTransform[-RotateAngle1TAtoXYPlane[
     MinTangentPoint00], {1, 0, 0}][
  RotationTransform[-RotateAngle2TAtoXAxis[MinTangentPoint00], {0, 0, 1}][
   RotationTransform[-RotateAngle3PointBtoXYPlane[
       MinTangentPoint00, Point00, Point30], {1, 0, 0}][
    SignofPoints[MaxTangentPoint00, MinTangentPoint30, Point00, Point30] *
     RotationTransform[-Angle1PointEndtoXYPlane[{2, 2, 1, 1},
         {12.718433357775211, 1, 6.764357890136049`}], {1, 0, 0}][{0, 1, 0}]]]]
```

{-1., 0., 0.}

$\{0., 1.83697\times 10^{-16}, 1.\}$

$\{0., -1., 1.83697\times 10^{-16}\}$

{0., -1., 0.}

{0., 0., -1.}

{-1., 0., 0.}

LASC

```
Clear["Global`*"]
```

```
Point00 = {0, 0, 0};(*According to Point*)
Point03 = {-0.7077264466498845, 0, 0.5008767232876717};
Point30 = {0, -0.7077264466498845, -0.5008767232876717};
Point33 = {-1.0013149889663, -1.0013149889663, 0};

MaxTangentPoint00 = {-1, 0, 0};(*According to LoC*)
MaxTangentPoint03 = {-0.5770131138020663, 0, 0.8167348814030436};
MaxTangentPoint30 = {-1, 0, 0};
MaxTangentPoint33 =
  {-0.6664720667513934, -0.3335279332486068, 0.6667639027294741};

MinTangentPoint00 = {0, -1, 0};
MinTangentPoint30 = {0., -0.5770131138020663, -0.8167348814030436};
MinTangentPoint03 = {0, -1, 0};
MinTangentPoint33 =
  {-0.3335279332486068, -0.6664720667513934, -0.6667639027294741};
DPbval = {{1 / 5}, {3 / 40, 9 / 40}, {44 / 45, -56 / 15, 32 / 9},
   {19 372 / 6561, -25 360 / 2187, 64 448 / 6561, -212 / 729},
   {9017 / 3168, -355 / 33, 46 732 / 5247, 49 / 176, -5103 / 18 656},
   {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84}};
DPcval = {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84, 0};
DPaval = {1 / 5, 3 / 10, 4 / 5, 8 / 9, 1, 1};
DPerrval =
  {71 / 57 600, 0, -71 / 16 695, 71 / 1920, -17 253 / 339 200, 22 / 525, -1 / 40};
DPCoefficients[5, p_] :=
  N[{DPbval, DPcval, DPaval, DPerrval}, p];
CurvatureLASC[ArcL_, Alpha_, Lambda_, InitRC_] :=
 Piecewise[{{1 / Exp[Lambda * ArcL + Log[InitRC]], Alpha == 0}},
  (InitRC ^ (Alpha) + Lambda * Alpha * ArcL) ^ (-1 / Alpha)]
TorsionLASC[ArcL_, BetaT_, Omega_, InitRT_] :=
 Piecewise[{{1 / Exp[Omega * ArcL + Log[InitRT]], BetaT == 0}},
  (InitRT ^ (BetaT) + Omega * BetaT * ArcL) ^ (-1 / BetaT)]

LASCEquation[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_] :=
 Flatten[NDSolve[{x'[ArcL] == t1[ArcL], y'[ArcL] == t2[ArcL], z'[ArcL] == t3[ArcL],
    t1'[ArcL] == CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × n1[ArcL],
    t2'[ArcL] == CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × n2[ArcL],
    t3'[ArcL] == CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × n3[ArcL],
    n1'[ArcL] == -CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × t1[ArcL] +
      TorsionLASC[ArcL, BetaT, Omega, InitRT] × b1[ArcL],
    n2'[ArcL] == -CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × t2[ArcL] +
      TorsionLASC[ArcL, BetaT, Omega, InitRT] × b2[ArcL],
    n3'[ArcL] == -CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × t3[ArcL] +
      TorsionLASC[ArcL, BetaT, Omega, InitRT] × b3[ArcL],
    b1'[ArcL] == -TorsionLASC[ArcL, BetaT, Omega, InitRT] × n1[ArcL],
    b2'[ArcL] == -TorsionLASC[ArcL, BetaT, Omega, InitRT] × n2[ArcL],
```

```
      b3'[ArcL] == -TorsionLASC[ArcL, BetaT, Omega, InitRT] × n3[ArcL],
      x[0] == 0, y[0] == 0, z[0] == 0,
      t1[0] == 1, t2[0] == 0, t3[0] == 0,
      n1[0] == 0, n2[0] == 0, n3[0] == 1,
      b1[0] == 0, b2[0] == 1, b3[0] == 0},
    {x, y, z, t1, t2, t3, n1, n2, n3, b1, b2, b3, t1', t2', t3'},
    {ArcL, -8, 8}, Method → {"ExplicitRungeKutta",
      "DifferenceOrder" → 5, "Coefficients" → DPCoefficients,
      "EmbeddedDifferenceOrder" → 4, "StepSizeControlParameters" -> {1, 0},
      "StepSizeSafetyFactors" → {1, 1}, "StepSizeRatioBounds" → {1 / 25, 25},
      "StiffnessTest" → False, "DiscontinuityProcessing" → False},
    MaxStepSize → 0.01, MaxSteps → Automatic]]
LASCEquation2[Alpha_, BetaT_, Omega_, InitRC_] := Flatten[ParametricNDSolve[
    {x'[ArcL] == t1[ArcL], y'[ArcL] == t2[ArcL], z'[ArcL] == t3[ArcL],
      t1'[ArcL] == CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × n1[ArcL],
      t2'[ArcL] == CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × n2[ArcL],
      t3'[ArcL] == CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × n3[ArcL],
      n1'[ArcL] == -CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × t1[ArcL] +
        TorsionLASC[ArcL, BetaT, Omega, InitRT] × b1[ArcL],
      n2'[ArcL] == -CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × t2[ArcL] +
        TorsionLASC[ArcL, BetaT, Omega, InitRT] × b2[ArcL],
      n3'[ArcL] == -CurvatureLASC[ArcL, Alpha, Lambda, InitRC] × t3[ArcL] +
        TorsionLASC[ArcL, BetaT, Omega, InitRT] × b3[ArcL],
      b1'[ArcL] == -TorsionLASC[ArcL, BetaT, Omega, InitRT] × n1[ArcL],
      b2'[ArcL] == -TorsionLASC[ArcL, BetaT, Omega, InitRT] × n2[ArcL],
      b3'[ArcL] == -TorsionLASC[ArcL, BetaT, Omega, InitRT] × n3[ArcL],
      x[0] == 0, y[0] == 0, z[0] == 0,
      t1[0] == 1, t2[0] == 0, t3[0] == 0,
      n1[0] == 0, n2[0] == 0, n3[0] == 1,
      b1[0] == 0, b2[0] == 1, b3[0] == 0},
    {x, y, z, t1, t2, t3, n1, n2, n3, b1, b2, b3, t1', t2', t3'},
    {ArcL, -8, 8}, {Lambda, InitRT}, Method → {"ExplicitRungeKutta",
      "DifferenceOrder" → 5, "Coefficients" → DPCoefficients,
      "EmbeddedDifferenceOrder" → 4, "StepSizeControlParameters" -> {1, 0},
      "StepSizeSafetyFactors" → {1, 1}, "StepSizeRatioBounds" → {1 / 25, 25},
      "StiffnessTest" → False, "DiscontinuityProcessing" → False},
    MaxStepSize → 0.01, MaxSteps → Automatic]]
NoArcCosThetaB[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_, s_] :=
  vectorchord[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s].
   tangentB[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s]
NormalVectorB[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_, s_] :=
  Cross[vectorchord[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s],
    tangentA[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s]] /
   Sqrt[Cross[vectorchord[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s],
      tangentA[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s]].
     Cross[vectorchord[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s],
      tangentA[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s]]]
```

```
NoArcSinThetaC[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_, s_] :=
 Abs[tangentB[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s].
   NormalVectorB[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s]]
tangentB[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_, s_] :=
 Re[{t1[s], t2[s], t3[s]}] /.
  LASCEquation[Alpha, BetaT, Lambda, Omega, InitRC, InitRT]
AngleAlphaBetaofTwoPoint[TangentA_, TangentB_, PointA_, PointB_] :=
 {ArcCos[TangentA.VectorBetweenTwoPoints[PointA, PointB]],
  ArcCos[TangentB.VectorBetweenTwoPoints[PointA, PointB]]}
AngleBetaofTwoPoint[TangentA_, TangentB_, PointA_, PointB_] :=
 ArcCos[TangentB.VectorBetweenTwoPoints[PointA, PointB]]
NOIteration[a_, b_, tol_] := Ceiling[N[1 / Log[2] Log[Abs[a - b] / tol]]];
vectorchord[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_, s_] :=
 (Re[{x[Lambda, InitRT][s], y[Lambda, InitRT][s], z[Lambda, InitRT][s]}] -
     {0, 0, 0}) / Sqrt[(Re[{x[Lambda, InitRT][s],
         y[Lambda, InitRT][s], z[Lambda, InitRT][s]}] - {0, 0, 0}).
     (Re[{x[Lambda, InitRT][s], y[Lambda, InitRT][s], z[Lambda, InitRT][s]}] -
       {0, 0, 0})] /. LASCEquation2[Alpha, BetaT, Omega, InitRC]
tangentA[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_, s_] := {1, 0, 0}

NoArcCosThetaA[Alpha_, BetaT_, Lambda_, Omega_, InitRC_, InitRT_, s_] :=
 vectorchord[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s].
  tangentA[Alpha, BetaT, Lambda, Omega, InitRC, InitRT, s]



RotateAngle1TAtoXYPlane[TangentA_] :=
 Piecewise[{{-ArcCos[0], TangentA == {1, 0, 0} || TangentA == {-1, 0, 0}}},
  Piecewise[{{-ArcCos[{0, 1, 0}.({0, TangentA[[2]], TangentA[[3]]} /
          Sqrt[{0, TangentA[[2]], TangentA[[3]]}.
            {0, TangentA[[2]], TangentA[[3]]}])], Sign[TangentA[[3]]] == 1}},
   ArcCos[{0, 1, 0}.({0, TangentA[[2]], TangentA[[3]]} / Sqrt[{0,
          TangentA[[2]], TangentA[[3]]}.{0, TangentA[[2]], TangentA[[3]]}])]]]
TangentAonXYPlane[TangentA_] := RotationTransform[
   RotateAngle1TAtoXYPlane[TangentA], {1, 0, 0}][TangentA]
TangentBAfterRotateAngle1[TangentA_, TangentB_] :=
 RotationTransform[RotateAngle1TAtoXYPlane[TangentA], {1, 0, 0}][TangentB]
PointBAfterRotateAngle1[TangentA_, PointA_, PointB_] :=
 RotationTransform[RotateAngle1TAtoXYPlane[TangentA], {1, 0, 0}][PointB - PointA]

RotateAngle2TAtoXAxis[TangentA_] :=
 Piecewise[{{-ArcCos[{1, 0, 0}.TangentAonXYPlane[TangentA]],
     Sign[TangentAonXYPlane[TangentA][[2]]] == 1}},
  ArcCos[{1, 0, 0}.TangentAonXYPlane[TangentA]]]
TangentAonXAxis[TangentA_] := RotationTransform[
   RotateAngle2TAtoXAxis[TangentA], {0, 0, 1}][TangentAonXYPlane[TangentA]]
TangentBAfterRotateAngle2[TangentA_, TangentB_] :=
```

```
 RotationTransform[RotateAngle2TAtoXAxis[TangentA], {0, 0, 1}][
  TangentBAfterRotateAngle1[TangentA, TangentB]]
PointBAfterRotateAngle2[TangentA_, PointA_, PointB_] :=
 RotationTransform[RotateAngle2TAtoXAxis[TangentA], {0, 0, 1}][
  PointBAfterRotateAngle1[TangentA, PointA, PointB]]

RotateAngle3PointBtoXYPlane[TangentA_, PointA_, PointB_] := Piecewise[
  {{-ArcCos[{0, 1, 0}.({0, PointBAfterRotateAngle2[TangentA, PointA, PointB][[
          2]], PointBAfterRotateAngle2[TangentA, PointA, PointB][[3]]} /
        Sqrt[{0, PointBAfterRotateAngle2[TangentA, PointA, PointB][[2]],
           PointBAfterRotateAngle2[TangentA, PointA, PointB][[3]]}.
          {0, PointBAfterRotateAngle2[TangentA, PointA, PointB][[2]],
           PointBAfterRotateAngle2[TangentA, PointA, PointB][[3]]}])],
    Sign[PointBAfterRotateAngle2[TangentA, PointA, PointB][[3]]] == 1}},
  ArcCos[{0, 1, 0}.({0, PointBAfterRotateAngle2[TangentA, PointA, PointB][[2]],
       PointBAfterRotateAngle2[TangentA, PointA, PointB][[3]]} /
      Sqrt[{0, PointBAfterRotateAngle2[TangentA, PointA, PointB][[2]],
         PointBAfterRotateAngle2[TangentA, PointA, PointB][[3]]}.
        {0, PointBAfterRotateAngle2[TangentA, PointA, PointB][[2]],
         PointBAfterRotateAngle2[TangentA, PointA, PointB][[3]]}])]]

TangentBAfterRotateAngle3[TangentA_, TangentB_, PointA_, PointB_] :=
 RotationTransform[RotateAngle3PointBtoXYPlane[TangentA, PointA, PointB],
   {1, 0, 0}][TangentBAfterRotateAngle2[TangentA, TangentB]]
PointBAfterRotateAngle3[TangentA_, TangentB_, PointA_, PointB_] :=
 RotationTransform[RotateAngle3PointBtoXYPlane[TangentA, PointA, PointB],
   {1, 0, 0}][PointBAfterRotateAngle2[TangentA, PointA, PointB]]

VectorBetweenTwoPoints[PointA_, PointB_] :=
 (PointB - PointA) / Sqrt[(PointB - PointA).(PointB - PointA)]

AngleAlphaofTwoPoint[TangentA_, TangentB_, PointA_, PointB_] :=
 ArcCos[TangentA.VectorBetweenTwoPoints[PointA, PointB]]

AngleTangentBtoXYPlane[TangentA_, TangentB_, PointA_, PointB_] :=
 ArcSin[TangentB.(Cross[TangentA, VectorBetweenTwoPoints[PointA, PointB]] /
     Sqrt[Cross[TangentA, VectorBetweenTwoPoints[PointA, PointB]].
       Cross[TangentA, VectorBetweenTwoPoints[PointA, PointB]]])]
PointBonXYPlane[TangentA_, TangentB_, PointA_, PointB_] :=
 Piecewise[{{PointBAfterRotateAngle3[TangentA, TangentB, PointA, PointB], Sign[
      PointBAfterRotateAngle3[TangentA, TangentB, PointA, PointB][[1]]] == 1}},
  {-1, 1, 1} * PointBAfterRotateAngle3[TangentA, TangentB, PointA, PointB]]
TangentPointBonXYPlane[TangentA_, TangentB_, PointA_, PointB_] :=
 Piecewise[{{TangentBAfterRotateAngle3[TangentA, TangentB, PointA, PointB], Sign[
      PointBAfterRotateAngle3[TangentA, TangentB, PointA, PointB][[1]]] == 1}},
  {1, -1, 1} * TangentBAfterRotateAngle3[TangentA, TangentB, PointA, PointB]]
TangentPointBonXYPlanewithNegz[TangentA_, TangentB_, PointA_, PointB_] :=
```

```
 Piecewise[
  {{{1, 1, -1} * TangentPointBonXYPlane[TangentA, TangentB, PointA, PointB],
    Sign[TangentPointBonXYPlane[TangentA, TangentB, PointA, PointB][[3]]] == 1}},
  TangentPointBonXYPlane[TangentA, TangentB, PointA, PointB]]

Angle1PointEndtoXYPlane[
  {Alpha_, BetaT_, Omega_, InitRC_}, {Lambda_, InitRT_, s_}] :=
 ArcCos[{0, 1, 0}.((Re[{0, y[Lambda, InitRT][s], z[Lambda, InitRT][s]}]) /
      Sqrt[(Re[{0, y[Lambda, InitRT][s], z[Lambda, InitRT][s]}].
          Re[{0, y[Lambda, InitRT][s], z[Lambda, InitRT][s]}])]) /.
   LASCEquation2[Alpha, BetaT, Omega, InitRC]]

IdealTangent[{Alpha_, BetaT_, Omega_, InitRC_}, {Lambda_, InitRT_, s_}] :=
 RotationTransform[-Angle1PointEndtoXYPlane[
     {Alpha, BetaT, Omega, InitRC}, {Lambda, InitRT, s}], {1, 0, 0}][
  (Re[{t1[Lambda, InitRT][s], t2[Lambda, InitRT][s], t3[Lambda, InitRT][s]}] /.
    LASCEquation2[Alpha, BetaT, Omega, InitRC])]

TangentX[{Alpha_, BetaT_, Omega_, InitRC_}, {Lambda_, InitRT_, s_}] :=
 RotationTransform[-Angle1PointEndtoXYPlane[
     {Alpha, BetaT, Omega, InitRC}, {Lambda, InitRT, s}], {1, 0, 0}][
  (Re[{t1[Lambda, InitRT][s], t2[Lambda, InitRT][s], t3[Lambda, InitRT][s]}] /.
    LASCEquation2[Alpha, BetaT, Omega, InitRC])]

SignofPoints[TangentA_, TangentB_, PointA_, PointB_] :=
 Piecewise[{{{1, 1, -1}, Sign[TangentPointBonXYPlane[
        TangentA, TangentB, PointA, PointB][[3]]] == 1}}, {1, 1, 1}]


tangentofCoonMaxLASC =
 RotationTransform[-RotateAngle1TAtoXYPlane[MaxTangentPoint30], {1, 0, 0}][
  RotationTransform[-RotateAngle2TAtoXAxis[MaxTangentPoint30], {0, 0, 1}][
   RotationTransform[-RotateAngle3PointBtoXYPlane[MaxTangentPoint30,
       Point30, Point33], {1, 0, 0}][SignofPoints[MaxTangentPoint30,
      MaxTangentPoint33, Point30, Point33] * RotationTransform[
       -Angle1PointEndtoXYPlane[{2, 2, 1, 1}, {6.285247558593747`,
          7.303131835937503`, 3.1084204465150833`}], {1, 0, 0}][{1, 0, 0}]]]]
NormalofCoonMaxLASC = RotationTransform[-RotateAngle1TAtoXYPlane[
     MaxTangentPoint30], {1, 0, 0}][
  RotationTransform[-RotateAngle2TAtoXAxis[MaxTangentPoint30], {0, 0, 1}][
   RotationTransform[-RotateAngle3PointBtoXYPlane[MaxTangentPoint30,
       Point30, Point33], {1, 0, 0}][SignofPoints[MaxTangentPoint30,
      MaxTangentPoint33, Point30, Point33] * RotationTransform[
       -Angle1PointEndtoXYPlane[{2, 2, 1, 1}, {6.285247558593747`,
          7.303131835937503`, 3.1084204465150833`}], {1, 0, 0}][{0, 0, 1}]]]]
BinormalofCoonMaxLASC = RotationTransform[-RotateAngle1TAtoXYPlane[
     MaxTangentPoint30], {1, 0, 0}][
  RotationTransform[-RotateAngle2TAtoXAxis[MaxTangentPoint30], {0, 0, 1}][
```

```
    RotationTransform[-RotateAngle3PointBtoXYPlane[MaxTangentPoint30,
        Point30, Point33], {1, 0, 0}][SignofPoints[MaxTangentPoint30,
       MaxTangentPoint33, Point30, Point33] * RotationTransform[
        -Angle1PointEndtoXYPlane[{2, 2, 1, 1}, {6.285247558593747`,
           7.303131835937503`, 3.1084204465150833`}], {1, 0, 0}][{0, 1, 0}]]]]

tangentofCoonMinLASC =
  RotationTransform[-RotateAngle1TAtoXYPlane[MinTangentPoint03], {1, 0, 0}][
   RotationTransform[-RotateAngle2TAtoXAxis[MinTangentPoint03], {0, 0, 1}][
    RotationTransform[-RotateAngle3PointBtoXYPlane[MinTangentPoint03,
        Point03, Point33], {1, 0, 0}][SignofPoints[MinTangentPoint03,
       MinTangentPoint33, Point03, Point33] * RotationTransform[
        -Angle1PointEndtoXYPlane[{2, 2, 1, 1}, {6.285247558593747`,
           7.303131835937503`, 3.1084204465150833`}], {1, 0, 0}][{1, 0, 0}]]]]
NormalofCoonMinLASC = RotationTransform[-RotateAngle1TAtoXYPlane[
      MinTangentPoint03], {1, 0, 0}][
   RotationTransform[-RotateAngle2TAtoXAxis[MinTangentPoint03], {0, 0, 1}][
    RotationTransform[-RotateAngle3PointBtoXYPlane[MinTangentPoint03,
        Point03, Point33], {1, 0, 0}][SignofPoints[MinTangentPoint03,
       MinTangentPoint33, Point03, Point33] * RotationTransform[
        -Angle1PointEndtoXYPlane[{2, 2, 1, 1}, {6.285247558593747`,
           7.303131835937503`, 3.1084204465150833`}], {1, 0, 0}][{0, 0, 1}]]]]
BinormalofCoonMinLASC = RotationTransform[-RotateAngle1TAtoXYPlane[
      MinTangentPoint03], {1, 0, 0}][
   RotationTransform[-RotateAngle2TAtoXAxis[MinTangentPoint03], {0, 0, 1}][
    RotationTransform[-RotateAngle3PointBtoXYPlane[MinTangentPoint03,
        Point03, Point33], {1, 0, 0}][SignofPoints[MinTangentPoint03,
       MinTangentPoint33, Point30, Point33] * RotationTransform[
        -Angle1PointEndtoXYPlane[{2, 2, 1, 1}, {6.285247558593747`,
           7.303131835937503`, 3.1084204465150833`}], {1, 0, 0}][{0, 1, 0}]]]]
```

```
{-1., 0., 0.}

{0., -0.593059, 0.805159}

{0., 0.805159, 0.593059}

{0., -1., 0.}

{-0.593059, 0., -0.805159}

{0.805159, 0., -0.593059}
```

Figure 2

Set up

```
Cross[{-1, 0, 0}, {0, 0, 1}]
{0, 1, 0}

Cross[{0, -1, 0}, {0, 0, -1}]
{1, 0, 0}

Point00 = {0, 0, 0};(*According to Point*)
Point03 = {-0.7077264466498845, 0, 0.5008767232876717};
Point30 = {0, -0.7077264466498845, -0.5008767232876717};
Point33 = {-1.0013149889663, -1.0013149889663, 0};

MaxTangentPoint00 = {-1, 0, 0};(*According to LoC*)
MaxTangentPoint03 = {-0.5770131138020663, 0, 0.8167348814030436};
MaxTangentPoint30 = {-1, 0, 0};
MaxTangentPoint33 =
  {-0.6664720667513934`, -0.3335279332486068`, 0.6667639027294741`};

MinTangentPoint00 = {0, -1, 0};
MinTangentPoint30 = {0., -0.5770131138020663, -0.8167348814030436};
MinTangentPoint03 = {0, -1, 0};
MinTangentPoint33 =
  {-0.3335279332486068`, -0.6664720667513934`, -0.6667639027294741`};


DPbval = {{1 / 5}, {3 / 40, 9 / 40}, {44 / 45, -56 / 15, 32 / 9},
   {19 372 / 6561, -25 360 / 2187, 64 448 / 6561, -212 / 729},
   {9017 / 3168, -355 / 33, 46 732 / 5247, 49 / 176, -5103 / 18 656},
   {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84}};
DPcval = {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84, 0};
DPaval  = {1 / 5, 3 / 10, 4 / 5, 8 / 9, 1, 1};
DPerrval =
  {71 / 57 600, 0, -71 / 16 695, 71 / 1920, -17 253 / 339 200, 22 / 525, -1 / 40};
DPCoefficients[5, p_] :=
  N[{DPbval, DPcval, DPaval, DPerrval}, p];

Surf[u_, v_][s_] := {u[s], v[s], u[s]^2 - v[s]^2};
DuSurf[u_, v_][s_] := {1, 0, 2 u[s]};
DvSurf[u_, v_][s_] := {0, 1, -2 v[s]};
Du2Surf[u_, v_][s_] := {0, 0, 2};
Dv2Surf[u_, v_][s_] := {0, 0, -2};
DuDvSurf[u_, v_][s_] := {0, 0, 0};
```

```
CrossN[u_, v_][s_] := Cross[DuSurf[u, v][s], DvSurf[u, v][s]];
ScalarN[u_, v_][s_] := Sqrt[CrossN[u, v][s].CrossN[u, v][s]];
NormalSurf[u_, v_][s_] := CrossN[u, v][s] / ScalarN[u, v][s];
SurfE[u_, v_][s_] := DuSurf[u, v][s].DuSurf[u, v][s];
SurfF[u_, v_][s_] := DuSurf[u, v][s].DvSurf[u, v][s];
SurfG[u_, v_][s_] := DvSurf[u, v][s].DvSurf[u, v][s];
SurfL[u_, v_][s_] := Du2Surf[u, v][s].NormalSurf[u, v][s];
SurfM[u_, v_][s_] := DuDvSurf[u, v][s].NormalSurf[u, v][s];
SurfN[u_, v_][s_] := Dv2Surf[u, v][s].NormalSurf[u, v][s];
GaussCurva[u_, v_][s_] := (SurfL[u, v][s] × SurfN[u, v][s] - SurfM[u, v][s]^2) /
   (SurfE[u, v][s] × SurfG[u, v][s] - SurfF[u, v][s]^2);
MeanCurva[u_, v_][s_] := 1 / 2 (SurfE[u, v][s] × SurfN[u, v][s] +
      SurfG[u, v][s] × SurfL[u, v][s] - 2 SurfF[u, v][s] × SurfM[u, v][s]) /
    (SurfE[u, v][s] × SurfG[u, v][s] - SurfF[u, v][s]^2);
MaxPrinci[u_, v_][s_] := MeanCurva[u, v][s] +
   Sqrt[MeanCurva[u, v][s]^2 - GaussCurva[u, v][s]];
MinPrinci[u_, v_][s_] := MeanCurva[u, v][s] -
   Sqrt[MeanCurva[u, v][s]^2 - GaussCurva[u, v][s]];
Princi[a_, u_, v_][s_] := Piecewise[
   {{MaxPrinci[u, v][s], a === max}, {MinPrinci[u, v][s], a === min}}];

Etaη[a_, u_, v_][s_] :=
  1 / Sqrt[SurfE[u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s])^2 -
     2 SurfF[u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s])
      (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s]) +
     SurfG[u, v][s] (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s])^2];
Miuμ[a_, u_, v_][s_] := 1 / Sqrt[SurfE[u, v][s]
      (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])^2 -
     2 SurfF[u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s])
      (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s]) +
     SurfG[u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s])^2];

DuDs[a_, u_, v_][s_] :=
 Piecewise[{{Etaη[a, u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s]),
    Sign[(SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s])]
       (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s]) ≥
     Sign[(SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])]
      (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])}},
  Miuμ[a, u, v][s] (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])]
DvDs[a_, u_, v_][s_] := Piecewise[
  {{-Etaη[a, u, v][s] (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s]),
    Sign[(SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s])]
       (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s]) ≥
     Sign[(SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])]
      (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])}},
  -Miuμ[a, u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s])]
```

```
d1uds1[a_, b_, c_] := DuDs[a, u, v][s] /. {u[s] → b, v[s] → c}
d1vds1[a_, b_, c_] := DvDs[a, u, v][s] /. {u[s] → b, v[s] → c}

SolvedudvDP[a_] := ParametricNDSolve[
  {u'[ArcL] == DuDs[a, u, v][ArcL], v'[ArcL] == DvDs[a, u, v][ArcL],
   u[0] == iniu, v[0] == iniv},
  {u, v}, {ArcL, 0, 2}, {iniu, iniv}, Method →
   {"ExplicitRungeKutta", "DifferenceOrder" → 5, "Coefficients" → DPCoefficients,
    "EmbeddedDifferenceOrder" → 4, "StepSizeControlParameters" -> {1, 0},
    "StepSizeSafetyFactors" → {1, 1}, "StepSizeRatioBounds" → {1 / 25, 25},
    "StiffnessTest" → False, "DiscontinuityProcessing" → False},
  MaxStepSize → 0.01, MaxSteps → Automatic]

solution41 = SolvedudvDP[max];
solution42 = SolvedudvDP[min];

MaximumusDP[t_, s_] :=
 u[u[0, 0][t] /. solution42, v[0, 0][t] /. solution42][s] /. solution41
MaximumvsDP[t_, s_] :=
 v[u[0, 0][t] /. solution42, v[0, 0][t] /. solution42][s] /. solution41
MinimumusDP[t_, s_] :=
 u[u[0, 0][t] /. solution41, v[0, 0][t] /. solution41][s] /. solution42
MinimumvsDP[t_, s_] :=
 v[u[0, 0][t] /. solution41, v[0, 0][t] /. solution41][s] /. solution42
```

Result

```
Show[ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MaximumusDP[0, s], v[s] → MaximumvsDP[0, s]},
  {s, 0, .9}], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MaximumusDP[0.9, s], v[s] → MaximumvsDP[0.9, s]},
  {s, 0, 1.1920739744274536`}], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MinimumusDP[0, s], v[s] → MinimumvsDP[0, s]},
  {s, 0, .9}], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MinimumusDP[0.9, s], v[s] → MinimumvsDP[0.9, s]},
  {s, 0, 1.1920739744274536`}],
 Graphics3D[{{PointSize[Large], Point[{Point00, Point03, Point30, Point33}]},
   Text[Style["PA", Large], Point00 + {0.1, .1, 0.1}], Text[Style["PB", Large],
    Point03 + {0, .15, .1}], Text[Style["PC", Large], Point30 + {0.3, 0, 0.1}],
   Text[Style["PD", Large], Point33 + {-0.05, 0.1, .2}],
   Text[Style["tA2", Medium], (Point00 + Point00 + .5 * MinTangentPoint00) / 2 +
     {-0.15, 0, 0}], Text[Style["tB2", Medium],
    (Point03 + Point03 + .5 * MinTangentPoint03) / 2 + {-0.2, 0, 0}],
   Text[Style["tC2", Medium], (Point30 + Point30 + .5 * MinTangentPoint30) / 2 +
     {0.2, 0, 0}], Text[Style["tD2", Medium],
    (Point33 + Point33 + .5 * MinTangentPoint33) / 2 + {0.15, 0, 0}],
   Text[Style["tA1", Medium], (Point00 + Point00 + .5 * MaxTangentPoint00) / 2 +
     {0, 0.15, 0}], Text[Style["tB1", Medium],
    (Point03 + Point03 + .5 * MaxTangentPoint03) / 2 + {0, 0.15, 0}],
   Text[Style["tC1", Medium], (Point30 + Point30 + .5 * MaxTangentPoint30) / 2 +
     {0, -0.15, 0}], Text[Style["tD1", Medium],
    (Point33 + Point33 + .5 * MaxTangentPoint33) / 2 + {0, -0.15, 0}],
   Arrowheads[.04], Arrow[{Point00, Point00 + .5 * MinTangentPoint00}],
   Arrow[{Point03, Point03 + .5 * MinTangentPoint03}],
   Arrow[{Point30, Point30 + .5 * MinTangentPoint30}],
   Arrow[{Point33, Point33 + .5 * MinTangentPoint33}],
   Arrow[{Point00, Point00 + .5 * MaxTangentPoint00}],
   Arrow[{Point03, Point03 + .5 * MaxTangentPoint03}],
   Arrow[{Point30, Point30 + .5 * MaxTangentPoint30}],
   Arrow[{Point33, Point33 + .5 * MaxTangentPoint33}]}],
 PlotRange → All, Boxed → False, Axes → False]
```

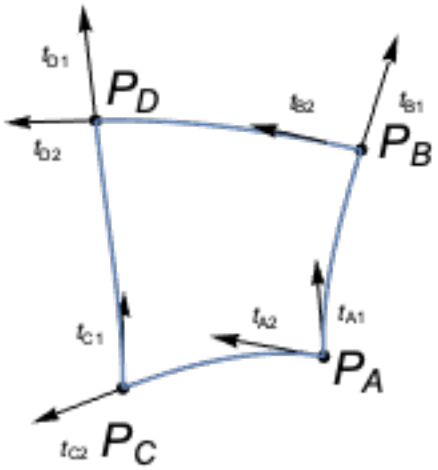


Figure 3

Set up

```
Point00 = {0, 0, 0};(*According to Point*)
Point03 = {-0.7077264466498845, 0, 0.5008767232876717};
Point30 = {0, -0.7077264466498845, -0.5008767232876717};
Point33 = {-1.0013149889663, -1.0013149889663, 0};

MaxTangentPoint00 = {-1, 0, 0};(*According to LoC*)
MaxTangentPoint03 = {-0.5770131138020663, 0, 0.8167348814030436};
MaxTangentPoint30 = {-1, 0, 0};
MaxTangentPoint33 =
  {-0.6664720667513934`, -0.3335279332486068`, 0.6667639027294741`};

MinTangentPoint00 = {0, -1, 0};
MinTangentPoint30 = {0., -0.5770131138020663, -0.8167348814030436};
MinTangentPoint03 = {0, -1, 0};
MinTangentPoint33 =
  {-0.3335279332486068`, -0.6664720667513934`, -0.6667639027294741`};


DPbval = {{1 / 5}, {3 / 40, 9 / 40}, {44 / 45, -56 / 15, 32 / 9},
   {19 372 / 6561, -25 360 / 2187, 64 448 / 6561, -212 / 729},
   {9017 / 3168, -355 / 33, 46 732 / 5247, 49 / 176, -5103 / 18 656},
   {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84}};
DPcval = {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84, 0};
DPaval  = {1 / 5, 3 / 10, 4 / 5, 8 / 9, 1, 1};
```

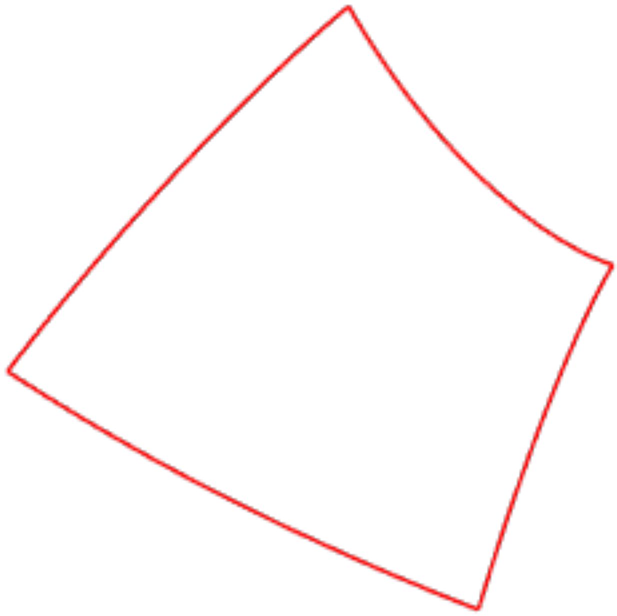
```
DPerrval =
  {71 / 57 600, 0, -71 / 16 695, 71 / 1920, -17 253 / 339 200, 22 / 525, -1 / 40};
DPCoefficients[5, p_] :=
  N[{DPbval, DPcval, DPaval, DPerrval}, p];

Surf[u_, v_][s_] := {u[s], v[s], u[s]^2 - v[s]^2};
DuSurf[u_, v_][s_] := {1, 0, 2 u[s]};
DvSurf[u_, v_][s_] := {0, 1, -2 v[s]};
Du2Surf[u_, v_][s_] := {0, 0, 2};
Dv2Surf[u_, v_][s_] := {0, 0, -2};
DuDvSurf[u_, v_][s_] := {0, 0, 0};
CrossN[u_, v_][s_] := Cross[DuSurf[u, v][s], DvSurf[u, v][s]];
ScalarN[u_, v_][s_] := Sqrt[CrossN[u, v][s].CrossN[u, v][s]];
NormalSurf[u_, v_][s_] := CrossN[u, v][s] / ScalarN[u, v][s];
SurfE[u_, v_][s_] := DuSurf[u, v][s].DuSurf[u, v][s];
SurfF[u_, v_][s_] := DuSurf[u, v][s].DvSurf[u, v][s];
SurfG[u_, v_][s_] := DvSurf[u, v][s].DvSurf[u, v][s];
SurfL[u_, v_][s_] := Du2Surf[u, v][s].NormalSurf[u, v][s];
SurfM[u_, v_][s_] := DuDvSurf[u, v][s].NormalSurf[u, v][s];
SurfN[u_, v_][s_] := Dv2Surf[u, v][s].NormalSurf[u, v][s];
GaussCurva[u_, v_][s_] := (SurfL[u, v][s] × SurfN[u, v][s] - SurfM[u, v][s]^2) /
   (SurfE[u, v][s] × SurfG[u, v][s] - SurfF[u, v][s]^2);
MeanCurva[u_, v_][s_] := 1 / 2 (SurfE[u, v][s] × SurfN[u, v][s] +
      SurfG[u, v][s] × SurfL[u, v][s] - 2 SurfF[u, v][s] × SurfM[u, v][s]) /
    (SurfE[u, v][s] × SurfG[u, v][s] - SurfF[u, v][s]^2);
MaxPrinci[u_, v_][s_] := MeanCurva[u, v][s] +
   Sqrt[MeanCurva[u, v][s]^2 - GaussCurva[u, v][s]];
MinPrinci[u_, v_][s_] := MeanCurva[u, v][s] -
   Sqrt[MeanCurva[u, v][s]^2 - GaussCurva[u, v][s]];
Princi[a_, u_, v_][s_] := Piecewise[
   {{MaxPrinci[u, v][s], a === max}, {MinPrinci[u, v][s], a === min}}];

Etaη[a_, u_, v_][s_] :=
  1 / Sqrt[SurfE[u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s])^2 -
     2 SurfF[u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s])
      (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s]) +
     SurfG[u, v][s] (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s])^2];
Miuμ[a_, u_, v_][s_] := 1 / Sqrt[SurfE[u, v][s]
      (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])^2 -
     2 SurfF[u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s])
      (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s]) +
     SurfG[u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s])^2];

DuDs[a_, u_, v_][s_] :=
 Piecewise[{{Etaη[a, u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s]),
    Sign[(SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s])]
      (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s]) ≥
```

```
      Sign[(SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])]
        (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])}},
   Miuμ[a, u, v][s] (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])]
DvDs[a_, u_, v_][s_] := Piecewise[
  {{-Etaη[a, u, v][s] (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s]),
     Sign[(SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s])]
       (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s]) ≥
      Sign[(SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])]
        (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])}},
   -Miuμ[a, u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s])]

d1uds1[a_, b_, c_] := DuDs[a, u, v][s] /. {u[s] → b, v[s] → c}
d1vds1[a_, b_, c_] := DvDs[a, u, v][s] /. {u[s] → b, v[s] → c}

SolvedudvDP[a_] := ParametricNDSolve[
  {u'[ArcL] == DuDs[a, u, v][ArcL], v'[ArcL] == DvDs[a, u, v][ArcL],
   u[0] == iniu, v[0] == iniv},
  {u, v}, {ArcL, 0, 2}, {iniu, iniv}, Method →
   {"ExplicitRungeKutta", "DifferenceOrder" → 5, "Coefficients" → DPCoefficients,
    "EmbeddedDifferenceOrder" → 4, "StepSizeControlParameters" -> {1, 0},
    "StepSizeSafetyFactors" →  {1, 1}, "StepSizeRatioBounds" →  {1 / 25, 25},
    "StiffnessTest" → False, "DiscontinuityProcessing" → False},
  MaxStepSize →  0.01, MaxSteps → Automatic]

solution41 = SolvedudvDP[max];
solution42 = SolvedudvDP[min];

MaximumusDP[t_, s_] :=
 u[u[0, 0][t] /. solution42, v[0, 0][t] /. solution42][s] /. solution41
MaximumvsDP[t_, s_] :=
 v[u[0, 0][t] /. solution42, v[0, 0][t] /. solution42][s] /. solution41
MinimumusDP[t_, s_] :=
 u[u[0, 0][t] /. solution41, v[0, 0][t] /. solution41][s] /. solution42
MinimumvsDP[t_, s_] :=
 v[u[0, 0][t] /. solution41, v[0, 0][t] /. solution41][s] /. solution42

Point00 = {0, 0, 0};(*According to Point*)
Point03 = {-0.7077264466498845, 0, 0.5008767232876717};
Point30 = {0, -0.7077264466498845, -0.5008767232876717};
Point33 = {-1.0013149889663, -1.0013149889663, 0};

MaxTangentPoint00 = {-1, 0, 0};(*According to LoC*)
MaxTangentPoint03 = {-0.5770131138020663, 0, 0.8167348814030436};
MaxTangentPoint30 = {-1, 0, 0};
MaxTangentPoint33 =
  {-0.6664720667513934`, -0.3335279332486068`, 0.6667639027294741`};

MinTangentPoint00 = {0, -1, 0};
MinTangentPoint30 = {0., -0.5770131138020663, -0.8167348814030436};
```

```
MinTangentPoint03 = {0, -1, 0};
MinTangentPoint33 =
  {-0.3335279332486068`, -0.6664720667513934`, -0.6667639027294741`};

DPbval = {{1 / 5}, {3 / 40, 9 / 40}, {44 / 45, -56 / 15, 32 / 9},
   {19 372 / 6561, -25 360 / 2187, 64 448 / 6561, -212 / 729},
   {9017 / 3168, -355 / 33, 46 732 / 5247, 49 / 176, -5103 / 18 656},
   {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84}};
DPcval = {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84, 0};
DPaval = {1 / 5, 3 / 10, 4 / 5, 8 / 9, 1, 1};
DPerrval =
  {71 / 57 600, 0, -71 / 16 695, 71 / 1920, -17 253 / 339 200, 22 / 525, -1 / 40};
DPCoefficients[5, p_] :=
  N[{DPbval, DPcval, DPaval, DPerrval}, p];
CurvatureLASC[ArcL_, Alpha_, Lambda_, InitRC_] :=
 Piecewise[{{1 / Exp[Lambda * ArcL + Log[InitRC]], Alpha == 0}},
  (InitRC ^ (Alpha) + Lambda * Alpha * ArcL) ^ (-1 / Alpha)]
TorsionLASC[ArcL_, BetaT_, Omega_, InitRT_] :=
 Piecewise[{{1 / Exp[Omega * ArcL + Log[InitRT]], BetaT == 0}},
  (InitRT ^ (BetaT) + Omega * BetaT * ArcL) ^ (-1 / BetaT)]

CoonMaxLAC[Alpha_, InitRC_, Lambda_,
  ArcL_, Scalar_, {Pointx_, Pointy_, Pointz_}] :=
 Flatten[NDSolve[{xu01'[t] == tu01[t], xu02'[t] == tu02[t], xu03'[t] == tu03[t],
    tu01'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nu01[t],
    tu02'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nu02[t],
    tu03'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nu03[t],
    nu01'[t] == -ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tu01[t],
    nu02'[t] == -ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tu02[t],
    nu03'[t] == -ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tu03[t],

    xu01[0] == Pointx, xu02[0] == Pointy, xu03[0] == Pointz,
    tu01[0] == -1 * Scalar * ArcL, tu02[0] == 0, tu03[0] == 0,
    nu01[0] == 0, nu02[0] == 0, nu03[0] == 1 * Scalar * ArcL},
   {xu01, xu02, xu03, tu01, tu02, tu03, tu01', tu02', tu03', nu01, nu02, nu03},
   {t, 0, 2}, Method → {"ExplicitRungeKutta",
     "DifferenceOrder" → 5, "Coefficients" → DPCoefficients,
     "EmbeddedDifferenceOrder" → 4, "StepSizeControlParameters" -> {1, 0},
     "StepSizeSafetyFactors" → {1, 1}, "StepSizeRatioBounds" → {1 / 25, 25},
     "StiffnessTest" → False, "DiscontinuityProcessing" → False},
   MaxStepSize → 0.01, MaxSteps → Automatic]]

CoonMinLAC[Alpha_, InitRC_, Lambda_,
  ArcL_, Scalar_, {Pointx_, Pointy_, Pointz_}] :=
 Flatten[NDSolve[{xv01'[t] == tv01[t], xv02'[t] == tv02[t], xv03'[t] == tv03[t],
    tv01'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nv01[t],
    tv02'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nv02[t],
```

```
    tv03'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nv03[t],
    nv01'[t] == -ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tv01[t],
    nv02'[t] == -ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tv02[t],
    nv03'[t] == -ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tv03[t],

    xv01[0] == Pointx, xv02[0] == Pointy, xv03[0] == Pointz,
    tv01[0] == 0, tv02[0] == -1 * Scalar * ArcL, tv03[0] == 0,
    nv01[0] == 0, nv02[0] == 0, nv03[0] == -1 * Scalar * ArcL},
   {xv01, xv02, xv03, tv01, tv02, tv03, tv01', tv02', tv03', nv01, nv02, nv03},
   {t, 0, 2}, Method → {"ExplicitRungeKutta",
     "DifferenceOrder" → 5, "Coefficients" → DPCoefficients,
     "EmbeddedDifferenceOrder" → 4, "StepSizeControlParameters" -> {1, 0},
     "StepSizeSafetyFactors" → {1, 1}, "StepSizeRatioBounds" → {1 / 25, 25},
     "StiffnessTest" → False, "DiscontinuityProcessing" → False},
   MaxStepSize → 0.01, MaxSteps → Automatic]]

CoonMaxLASC[Alpha_, BetaT_, Omega_, InitRC_, Lambda_,
  InitRT_, ArcL_, Scalar_, {Pointx_, Pointy_, Pointz_}] :=
 Flatten[NDSolve[{xu11'[t] == tu11[t], xu12'[t] == tu12[t], xu13'[t] == tu13[t],
    tu11'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nu11[t],
    tu12'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nu12[t],
    tu13'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nu13[t],
    nu11'[t] == ArcL * (-CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tu11[t] +
        TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × bu11[t]),
    nu12'[t] == ArcL * (-CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tu12[t] +
        TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × bu12[t]),
    nu13'[t] == ArcL * (-CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tu13[t] +
        TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × bu13[t]),
    bu11'[t] == -ArcL * TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × nu11[t],
    bu12'[t] == -ArcL * TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × nu12[t],
    bu13'[t] == -ArcL * TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × nu13[t],
    xu11[0] == Pointx, xu12[0] == Pointy, xu13[0] == Pointz,
    tu11[0] == -1 * Scalar * ArcL, tu12[0] == 0, tu13[0] == 0,
    nu11[0] == 0, nu12[0] == -0.593058402842245 * Scalar * ArcL,
    nu13[0] == 0.8051594443451592 * Scalar * ArcL,
    bu11[0] == 0, bu12[0] == 0.8051594443451592 * Scalar * ArcL,
    bu13[0] == 0.593058402842245 * Scalar * ArcL},
   {xu11, xu12, xu13, tu11, tu12, tu13, nu11, nu12, nu13, bu11,
    bu12, bu13, tu11', tu12', tu13', bu11', bu12', bu13'},
   {t, 0, 2}, Method → {"ExplicitRungeKutta", "DifferenceOrder" → 5,
     "Coefficients" → DPCoefficients, "EmbeddedDifferenceOrder" → 4,
     "StepSizeControlParameters" -> {1, 0},
     "StepSizeSafetyFactors" → {1, 1}, "StepSizeRatioBounds" → {1 / 25, 25},
     "StiffnessTest" → False, "DiscontinuityProcessing" → False},
   MaxStepSize → 0.01, MaxSteps → Automatic]]

CoonMinLASC[Alpha_, BetaT_, Omega_, InitRC_, Lambda_,
```

```
  InitRT_, ArcL_, Scalar_, {Pointx_, Pointy_, Pointz_}] :=
 Flatten[NDSolve[{xv11'[t] == tv11[t], xv12'[t] == tv12[t], xv13'[t] == tv13[t],
    tv11'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nv11[t],
    tv12'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nv12[t],
    tv13'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nv13[t],
    nv11'[t] == ArcL * (-CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tv11[t] +
        TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × bv11[t]),
    nv12'[t] == ArcL * (-CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tv12[t] +
        TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × bv12[t]),
    nv13'[t] == ArcL * (-CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tv13[t] +
        TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × bv13[t]),
    bv11'[t] == -ArcL * TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × nv11[t],
    bv12'[t] == -ArcL * TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × nv12[t],
    bv13'[t] == -ArcL * TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × nv13[t],
    xv11[0] == Pointx, xv12[0] == Pointy, xv13[0] == Pointz,
    tv11[0] == 0, tv12[0] == -1 * Scalar * ArcL, tv13[0] == 0,
    nv11[0] == -0.593058402842245 * Scalar * ArcL,
    nv12[0] == 0, nv13[0] == -0.8051594443451592 * Scalar * ArcL,
    bv11[0] == 0.8051594443451592 * Scalar * ArcL, bv12[0] == 0,
    bv13[0] == -0.593058402842245 * Scalar * ArcL},
   {xv11, xv12, xv13, tv11, tv12, tv13, nv11, nv12, nv13, bv11,
    bv12, bv13, tv11', tv12', tv13', bv11', bv12', bv13'},
   {t, 0, 2}, Method → {"ExplicitRungeKutta", "DifferenceOrder" → 5,
     "Coefficients" → DPCoefficients, "EmbeddedDifferenceOrder" → 4,
     "StepSizeControlParameters" -> {1, 0},
     "StepSizeSafetyFactors" → {1, 1}, "StepSizeRatioBounds" → {1 / 25, 25},
     "StiffnessTest" → False, "DiscontinuityProcessing" → False},
   MaxStepSize → 0.01, MaxSteps → Automatic]]

CoonsMaxLoC0[u_] :=
  Evaluate[{xu01[u], xu02[u], xu03[u]} /. CoonMaxLAC[2, 1, 12.718433357775211,
     6.764357890136049`, 0.1319221327518171`, Point00]];
CoonsMaxLoC1[u_] := Evaluate[{xu11[u], xu12[u], xu13[u]} /.
    CoonMaxLASC[2, 2, 1, 1, 6.285247558593747`, 7.303131835937503`,
     3.1084204465150833`, 0.3816425818064554`, Point30]];
CoonsMinLoC0[v_] := Evaluate[{xv01[v], xv02[v], xv03[v]} /. CoonMinLAC[2, 1,
     12.718433357775211, 6.764357890136049`, 0.1319221327518171`, Point00]];
CoonsMinLoC1[v_] := Evaluate[{xv11[v], xv12[v], xv13[v]} /.
    CoonMinLASC[2, 2, 1, 1, 6.285247558593747`, 7.303131835937503`,
     3.1084204465150833`, 0.3816425818064554`, Point03]];


ParametricPlot3D[
 {CoonsMaxLoC0[s], CoonsMaxLoC1[s], CoonsMinLoC0[s], CoonsMinLoC1[s]},
 {s, 0, 1}, PlotStyle → Red, Axes → False, Boxed → False]
```



result

```
Show[ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MaximumusDP[0, s], v[s] → MaximumvsDP[0, s]},
  {s, 0, .9}], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MaximumusDP[0.9, s], v[s] → MaximumvsDP[0.9, s]},
  {s, 0, 1.1920739744274536`}], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MinimumusDP[0, s], v[s] → MinimumvsDP[0, s]},
  {s, 0, .9}], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MinimumusDP[0.9, s], v[s] → MinimumvsDP[0.9, s]},
  {s, 0, 1.1920739744274536`}],
 Graphics3D[{{PointSize[Large], Point[{Point00, Point03, Point30, Point33}]},
   Text[Style["C_AB", Large], Point00 + {-0.5, .3, 0.2}],
   Text[Style["C_BD", Large], Point03 + {-0.8, -.15, -.1}],
   Text[Style["C_AC", Large], Point30 + {0.0, 0.5, 0.2}],
   Text[Style["C_CD", Large], Point33 + {0.2, 0.3, -.5}], Text[Style["P_A", Large],
    Point00 + {0.1, .1, 0}], Text[Style["P_B", Large], Point03 + {0, .15, .1}],
   Text[Style["P_C", Large], Point30 + {0.15, -0.1, -0.05}],
   Text[Style["P_D", Large], Point33 + {-0.2, 0.1, .1}]}], ParametricPlot3D[
  {CoonsMaxLoC0[s], CoonsMaxLoC1[s], CoonsMinLoC0[s], CoonsMinLoC1[s]},
  {s, 0, 1}, PlotStyle → Red, Axes → False, Boxed → False],
 PlotRange → All, Boxed → False, Axes → False]
```

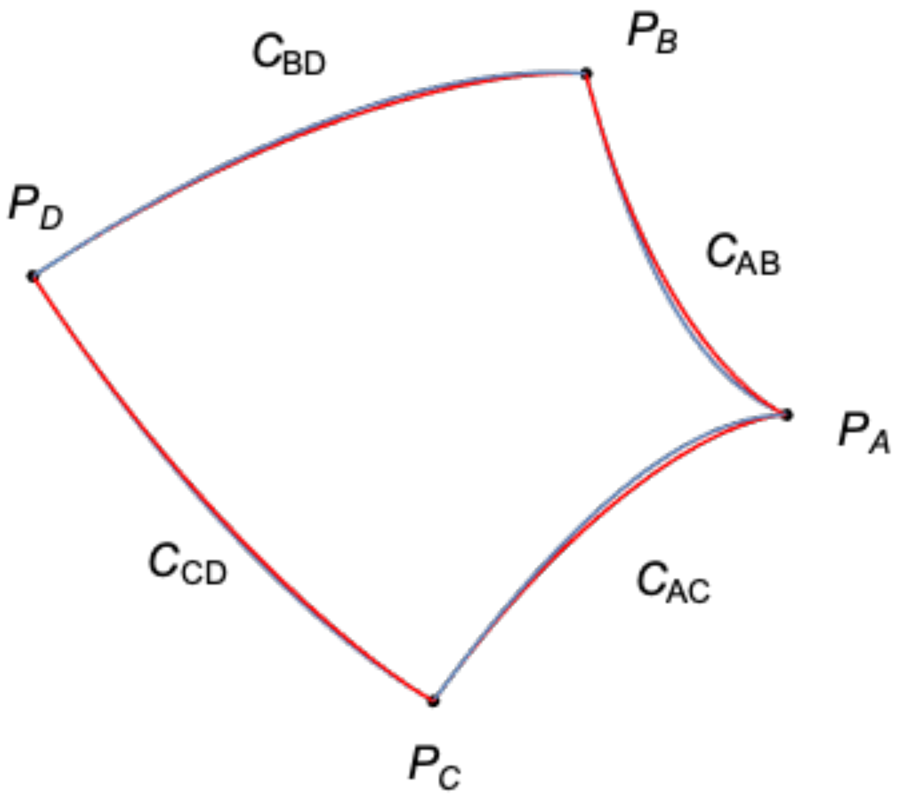


Figure 4

Set up

```
Point00 = {0, 0, 0};(*According to Point*)
Point03 = {-0.7077264466498845, 0, 0.5008767232876717};
Point30 = {0, -0.7077264466498845, -0.5008767232876717};
Point33 = {-1.0013149889663, -1.0013149889663, 0};

MaxTangentPoint00 = {-1, 0, 0};(*According to LoC*)
MaxTangentPoint03 = {-0.5770131138020663, 0, 0.8167348814030436};
MaxTangentPoint30 = {-1, 0, 0};
MaxTangentPoint33 =
  {-0.6664720667513934`, -0.3335279332486068`, 0.6667639027294741`};

MinTangentPoint00 = {0, -1, 0};
```

```
MinTangentPoint30 = {0., -0.5770131138020663, -0.8167348814030436};
MinTangentPoint03 = {0, -1, 0};
MinTangentPoint33 =
  {-0.3335279332486068`, -0.6664720667513934`, -0.6667639027294741`};

DPbval = {{1 / 5}, {3 / 40, 9 / 40}, {44 / 45, -56 / 15, 32 / 9},
   {19 372 / 6561, -25 360 / 2187, 64 448 / 6561, -212 / 729},
   {9017 / 3168, -355 / 33, 46 732 / 5247, 49 / 176, -5103 / 18 656},
   {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84}};
DPcval = {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84, 0};
DPaval = {1 / 5, 3 / 10, 4 / 5, 8 / 9, 1, 1};
DPerrval =
  {71 / 57 600, 0, -71 / 16 695, 71 / 1920, -17 253 / 339 200, 22 / 525, -1 / 40};
DPCoefficients[5, p_] :=
  N[{DPbval, DPcval, DPaval, DPerrval}, p];
CurvatureLASC[ArcL_, Alpha_, Lambda_, InitRC_] :=
 Piecewise[{{1 / Exp[Lambda * ArcL + Log[InitRC]], Alpha == 0}},
  (InitRC ^ (Alpha) + Lambda * Alpha * ArcL) ^ (-1 / Alpha)]
TorsionLASC[ArcL_, BetaT_, Omega_, InitRT_] :=
 Piecewise[{{1 / Exp[Omega * ArcL + Log[InitRT]], BetaT == 0}},
  (InitRT ^ (BetaT) + Omega * BetaT * ArcL) ^ (-1 / BetaT)]

CoonMaxLAC[Alpha_, InitRC_, Lambda_,
  ArcL_, Scalar_, {Pointx_, Pointy_, Pointz_}] :=
 Flatten[NDSolve[{xu01'[t] == tu01[t], xu02'[t] == tu02[t], xu03'[t] == tu03[t],
    tu01'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nu01[t],
    tu02'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nu02[t],
    tu03'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nu03[t],
    nu01'[t] == -ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tu01[t],
    nu02'[t] == -ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tu02[t],
    nu03'[t] == -ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tu03[t],

    xu01[0] == Pointx, xu02[0] == Pointy, xu03[0] == Pointz,
    tu01[0] == -1 * Scalar * ArcL, tu02[0] == 0, tu03[0] == 0,
    nu01[0] == 0, nu02[0] == 0, nu03[0] == 1 * Scalar * ArcL},
   {xu01, xu02, xu03, tu01, tu02, tu03, tu01', tu02', tu03', nu01, nu02, nu03},
   {t, 0, 2}, Method → {"ExplicitRungeKutta",
     "DifferenceOrder" → 5, "Coefficients" → DPCoefficients,
     "EmbeddedDifferenceOrder" → 4, "StepSizeControlParameters" -> {1, 0},
     "StepSizeSafetyFactors" → {1, 1}, "StepSizeRatioBounds" → {1 / 25, 25},
     "StiffnessTest" → False, "DiscontinuityProcessing" → False},
   MaxStepSize → 0.01, MaxSteps → Automatic]]

CoonMinLAC[Alpha_, InitRC_, Lambda_,
  ArcL_, Scalar_, {Pointx_, Pointy_, Pointz_}] :=
 Flatten[NDSolve[{xv01'[t] == tv01[t], xv02'[t] == tv02[t], xv03'[t] == tv03[t],
    tv01'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nv01[t],
```

```
    tv02'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nv02[t],
    tv03'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nv03[t],
    nv01'[t] == -ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tv01[t],
    nv02'[t] == -ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tv02[t],
    nv03'[t] == -ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tv03[t],

    xv01[0] == Pointx, xv02[0] == Pointy, xv03[0] == Pointz,
    tv01[0] == 0, tv02[0] == -1 * Scalar * ArcL, tv03[0] == 0,
    nv01[0] == 0, nv02[0] == 0, nv03[0] == -1 * Scalar * ArcL},
   {xv01, xv02, xv03, tv01, tv02, tv03, tv01', tv02', tv03', nv01, nv02, nv03},
   {t, 0, 2}, Method → {"ExplicitRungeKutta",
     "DifferenceOrder" → 5, "Coefficients" → DPCoefficients,
     "EmbeddedDifferenceOrder" → 4, "StepSizeControlParameters" -> {1, 0},
     "StepSizeSafetyFactors" → {1, 1}, "StepSizeRatioBounds" → {1 / 25, 25},
     "StiffnessTest" → False, "DiscontinuityProcessing" → False},
   MaxStepSize → 0.01, MaxSteps → Automatic]]

CoonMaxLASC[Alpha_, BetaT_, Omega_, InitRC_, Lambda_,
   InitRT_, ArcL_, Scalar_, {Pointx_, Pointy_, Pointz_}] :=
  Flatten[NDSolve[{xu11'[t] == tu11[t], xu12'[t] == tu12[t], xu13'[t] == tu13[t],
    tu11'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nu11[t],
    tu12'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nu12[t],
    tu13'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nu13[t],
    nu11'[t] == ArcL * (-CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tu11[t] +
        TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × bu11[t]),
    nu12'[t] == ArcL * (-CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tu12[t] +
        TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × bu12[t]),
    nu13'[t] == ArcL * (-CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tu13[t] +
        TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × bu13[t]),
    bu11'[t] == -ArcL * TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × nu11[t],
    bu12'[t] == -ArcL * TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × nu12[t],
    bu13'[t] == -ArcL * TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × nu13[t],
    xu11[0] == Pointx, xu12[0] == Pointy, xu13[0] == Pointz,
    tu11[0] == -1 * Scalar * ArcL, tu12[0] == 0, tu13[0] == 0,
    nu11[0] == 0, nu12[0] == -0.593058402842245 * Scalar * ArcL,
    nu13[0] == 0.8051594443451592 * Scalar * ArcL,
    bu11[0] == 0, bu12[0] == 0.8051594443451592 * Scalar * ArcL,
    bu13[0] == 0.593058402842245 * Scalar * ArcL},
   {xu11, xu12, xu13, tu11, tu12, tu13, nu11, nu12, nu13, bu11,
    bu12, bu13, tu11', tu12', tu13', bu11', bu12', bu13'},
   {t, 0, 2}, Method → {"ExplicitRungeKutta", "DifferenceOrder" → 5,
     "Coefficients" → DPCoefficients, "EmbeddedDifferenceOrder" → 4,
     "StepSizeControlParameters" -> {1, 0},
     "StepSizeSafetyFactors" → {1, 1}, "StepSizeRatioBounds" → {1 / 25, 25},
     "StiffnessTest" → False, "DiscontinuityProcessing" → False},
   MaxStepSize → 0.01, MaxSteps → Automatic]]
```

```
CoonMinLASC[Alpha_, BetaT_, Omega_, InitRC_, Lambda_,
  InitRT_, ArcL_, Scalar_, {Pointx_, Pointy_, Pointz_}] :=
 Flatten[NDSolve[{xv11'[t] == tv11[t], xv12'[t] == tv12[t], xv13'[t] == tv13[t],
    tv11'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nv11[t],
    tv12'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nv12[t],
    tv13'[t] == ArcL * CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × nv13[t],
    nv11'[t] == ArcL * (-CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tv11[t] +
        TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × bv11[t]),
    nv12'[t] == ArcL * (-CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tv12[t] +
        TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × bv12[t]),
    nv13'[t] == ArcL * (-CurvatureLASC[ArcL * t, Alpha, Lambda, InitRC] × tv13[t] +
        TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × bv13[t]),
    bv11'[t] == -ArcL * TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × nv11[t],
    bv12'[t] == -ArcL * TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × nv12[t],
    bv13'[t] == -ArcL * TorsionLASC[ArcL * t, BetaT, Omega, InitRT] × nv13[t],
    xv11[0] == Pointx, xv12[0] == Pointy, xv13[0] == Pointz,
    tv11[0] == 0, tv12[0] == -1 * Scalar * ArcL, tv13[0] == 0,
    nv11[0] == -0.593058402842245 * Scalar * ArcL,
    nv12[0] == 0, nv13[0] == -0.8051594443451592 * Scalar * ArcL,
    bv11[0] == 0.8051594443451592 * Scalar * ArcL, bv12[0] == 0,
    bv13[0] == -0.593058402842245 * Scalar * ArcL},
   {xv11, xv12, xv13, tv11, tv12, tv13, nv11, nv12, nv13, bv11,
    bv12, bv13, tv11', tv12', tv13', bv11', bv12', bv13'},
   {t, 0, 2}, Method → {"ExplicitRungeKutta", "DifferenceOrder" → 5,
     "Coefficients" → DPCoefficients, "EmbeddedDifferenceOrder" → 4,
     "StepSizeControlParameters" -> {1, 0},
     "StepSizeSafetyFactors" → {1, 1}, "StepSizeRatioBounds" → {1 / 25, 25},
     "StiffnessTest" → False, "DiscontinuityProcessing" → False},
   MaxStepSize → 0.01, MaxSteps → Automatic]]
```

```
CoonsMaxLoC0[u_] :=
  Evaluate[{xu01[u], xu02[u], xu03[u]} /. CoonMaxLAC[2, 1, 12.718433357775211,
     6.764357890136049`, 0.1319221327518171`, Point00]];
CoonsMaxLoC1[u_] := Evaluate[{xu11[u], xu12[u], xu13[u]} /.
    CoonMaxLASC[2, 2, 1, 1, 6.285247558593747`, 7.303131835937503`,
     3.1084204465150833`, 0.3816425818064554`, Point30]];
CoonsMinLoC0[v_] := Evaluate[{xv01[v], xv02[v], xv03[v]} /. CoonMinLAC[2, 1,
     12.718433357775211, 6.764357890136049`, 0.1319221327518171`, Point00]];
CoonsMinLoC1[v_] := Evaluate[{xv11[v], xv12[v], xv13[v]} /.
    CoonMinLASC[2, 2, 1, 1, 6.285247558593747`, 7.303131835937503`,
     3.1084204465150833`, 0.3816425818064554`, Point03]];
```

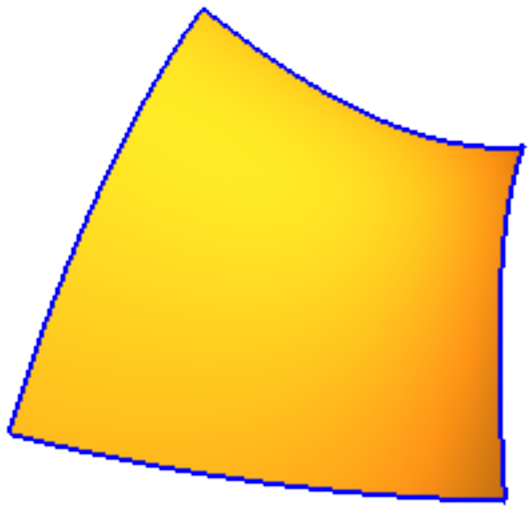
```
DCoonsMaxLoC0Du[u_] :=
```

```
  Evaluate[{tu01[u], tu02[u], tu03[u]} /. CoonMaxLAC[2, 1, 12.718433357775211,
     6.764357890136049`, 0.1319221327518171`, Point00]];
DCoonsMaxLoC1Du[u_] := Evaluate[{tu11[u], tu12[u], tu13[u]} /.
    CoonMaxLASC[2, 2, 1, 1, 6.285247558593747`, 7.303131835937503`,
     3.1084204465150833`, 0.3816425818064554`, Point30]];
DCoonsMinLoC0Dv[v_] := Evaluate[{tv01[v], tv02[v], tv03[v]} /. CoonMinLAC[2,
     1, 12.718433357775211, 6.764357890136049`, 0.1319221327518171`, Point00]];
DCoonsMinLoC1Dv[v_] := Evaluate[{tv11[v], tv12[v], tv13[v]} /.
    CoonMinLASC[2, 2, 1, 1, 6.285247558593747`, 7.303131835937503`,
     3.1084204465150833`, 0.3816425818064554`, Point03]];

D2CoonsMaxLoC0Du2[u_] :=
  Evaluate[{tu01'[u], tu02'[u], tu03'[u]} /. CoonMaxLAC[2, 1,
     12.718433357775211, 6.764357890136049`, 0.1319221327518171`, Point00]];
D2CoonsMaxLoC1Du2[u_] := Evaluate[{tu11'[u], tu12'[u], tu13'[u]} /.
    CoonMaxLASC[2, 2, 1, 1, 6.285247558593747`, 7.303131835937503`,
     3.1084204465150833`, 0.3816425818064554`, Point30]];
D2CoonsMinLoC0Dv2[v_] := Evaluate[{tv01'[v], tv02'[v], tv03'[v]} /. CoonMinLAC[2,
     1, 12.718433357775211, 6.764357890136049`, 0.1319221327518171`, Point00]];
D2CoonsMinLoC1Dv2[v_] := Evaluate[{tv11'[v], tv12'[v], tv13'[v]} /.
    CoonMinLASC[2, 2, 1, 1, 6.285247558593747`, 7.303131835937503`,
     3.1084204465150833`, 0.3816425818064554`, Point03]];

Coonslc[u_, v_] := (1 - v) * CoonsMaxLoC0[u] + v * CoonsMaxLoC1[u];
Coonsld[u_, v_] := (1 - u) * CoonsMinLoC0[v] + u * CoonsMinLoC1[v];
Coonsb[u_, v_] := CoonsMaxLoC0[0] * (1 - u) * (1 - v) +
   CoonsMaxLoC0[1] * u * (1 - v) + CoonsMaxLoC1[0] * (1 - u) * v + CoonsMaxLoC1[1] * u * v;
Coonspatch[u_, v_] := Coonslc[u, v] + Coonsld[u, v] - Coonsb[u, v];

Surf[u_, v_][s_] := Coonspatch[u, v] /. {u → u[s], v → v[s]};
DuSurf[u_, v_][s_] :=
  ((1 - v) * CoonsMaxLoC0[0] - (1 - v) * CoonsMaxLoC0[1] + v * CoonsMaxLoC1[0] -
     v * CoonsMaxLoC1[1] - CoonsMinLoC0[v] + CoonsMinLoC1[v] +
     (1 - v) * DCoonsMaxLoC0Du[u] + v * DCoonsMaxLoC1Du[u]) /. {u → u[s], v → v[s]};
DvSurf[u_, v_][s_] := ((1 - u) * CoonsMaxLoC0[0] - (1 - u) * CoonsMaxLoC1[0] +
     u * CoonsMaxLoC0[1] - u * CoonsMaxLoC1[1] - CoonsMaxLoC0[u] + CoonsMaxLoC1[u] +
     (1 - u) * DCoonsMinLoC0Dv[v] + u * DCoonsMinLoC1Dv[v]) /. {u → u[s], v → v[s]};
Du2Surf[u_, v_][s_] := ((1 - v) * D2CoonsMaxLoC0Du2[u] + v * D2CoonsMaxLoC1Du2[u]) /.
   {u → u[s], v → v[s]};
Dv2Surf[u_, v_][s_] := ((1 - u) * D2CoonsMinLoC0Dv2[v] + u * D2CoonsMinLoC1Dv2[v]) /.
   {u → u[s], v → v[s]};
DuDvSurf[u_, v_][s_] := (-CoonsMaxLoC0[0] + CoonsMaxLoC0[1] + CoonsMaxLoC1[0] -
     CoonsMaxLoC1[1] - DCoonsMaxLoC0Du[u] + DCoonsMaxLoC1Du[u] -
     DCoonsMinLoC0Dv[v] + DCoonsMinLoC1Dv[v]) /. {u → u[s], v → v[s]};
CrossN[u_, v_][s_] := Cross[DuSurf[u, v][s], DvSurf[u, v][s]];
ScalarN[u_, v_][s_] := Sqrt[CrossN[u, v][s].CrossN[u, v][s]];
NormalSurf[u_, v_][s_] := CrossN[u, v][s] / ScalarN[u, v][s];
```

```
L = {1/2, 1/2, -√2/2};

cvalueonCS = Table[
   Table[Dot[NormalSurf[m, n][s], L], {n[s], 0, 1, 0.001}], {m[s], 0, 1, 0.001}];

PointsonCS =
  Table[Table[Evaluate[Surf[u, v][s]], {v[s], 0, 1, 0.001}], {u[s], 0, 1, 0.001}];

mincvalueonCS = Min[cvalueonCS];
maxcvalueonCS = Max[cvalueonCS];

PositionofconCS = Table[Partition[Flatten[
      Table[Table[If[ mincvalueonCS + (k - 1) (maxcvalueonCS - mincvalueonCS) / 10 ≤
           cvalueonCS[[i, j]] < mincvalueonCS + k (maxcvalueonCS - mincvalueonCS) /
              10, {i, j}, {}], {j, 1, Dimensions[cvalueonCS][[2]]}],
       {i, 1, Dimensions[cvalueonCS][[1]]}]], 2], {k, 1, 10}];

RearrangedPointonCS = Table[
   Table[PointsonCS[[PositionofconCS[[k, i, 1]], PositionofconCS[[k, i, 2]]]],
    {i, 1, Dimensions[PositionofconCS[[k]]][[1]]}], {k, 1, 10}];
```

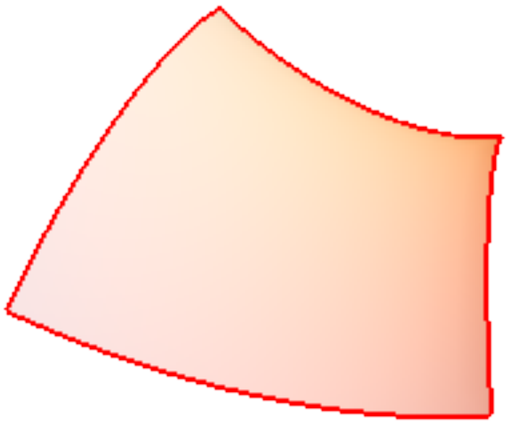
(a)

```
Show[ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MaximumusDP[0, s], v[s] → MaximumvsDP[0, s]},
  {s, 0, .9}, PlotStyle → Blue], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MaximumusDP[0.9, s], v[s] → MaximumvsDP[0.9, s]},
  {s, 0, 1.1920739744274536`}, PlotStyle → Blue],
 ParametricPlot3D[Surf[u, v][s] /. {u[s] → MinimumusDP[0, s],
    v[s] → MinimumvsDP[0, s]}, {s, 0, .9}, PlotStyle → Blue], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MinimumusDP[0.9, s], v[s] → MinimumvsDP[0.9, s]},
  {s, 0, 1.1920739744274536`}, PlotStyle → Blue],
 ParametricPlot3D[HPSurf[u, v][s], {v[s], 0, 1}, {u[s], 0, 1}, Axes → False,
  Boxed → False, Mesh → None], Axes → False, Boxed → False, PlotRange → All]
```

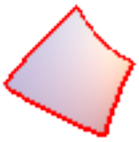
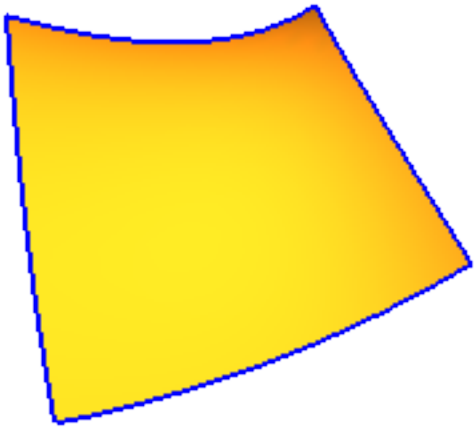


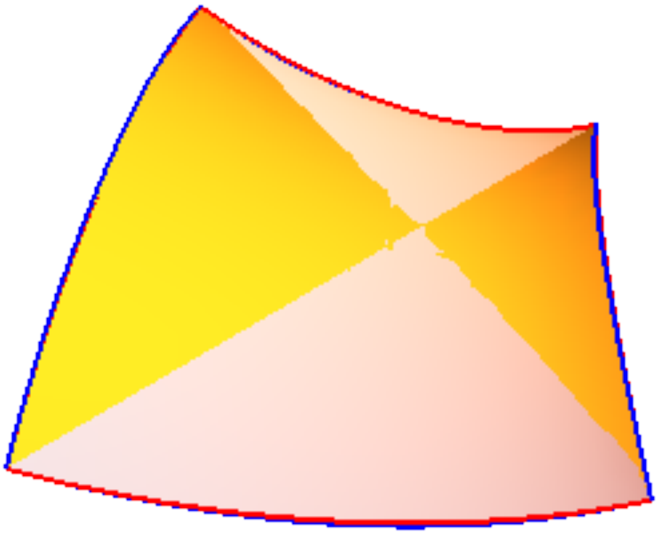
(b)

```
Show[ParametricPlot3D[CoonsMaxLoC0[u], {u, 0, 1}, PlotStyle → Red],
 ParametricPlot3D[CoonsMaxLoC1[u], {u, 0, 1}, PlotStyle → Red],
 ParametricPlot3D[CoonsMinLoC0[v], {v, 0, 1}, PlotStyle → Red],
 ParametricPlot3D[CoonsMinLoC1[v], {v, 0, 1}, PlotStyle → Red],
 ParametricPlot3D[Coonspatch[u, v], {u, 0, 1}, {v, 0, 1},
  PlotStyle → LightPink, Axes → False, Boxed → False, Mesh → None],
 Axes → False, Boxed → False, PlotRange → All]
```



(c)

Show[ , ]



Show[ , ]

(d)

```
Show[Table[Graphics3D[Point[RearrangedPointonCS[[i]]]],
  {i, 1, Dimensions[RearrangedPointonCS][[1]], 2}], Axes → False, Boxed → False]
```

Figure 5

(a)

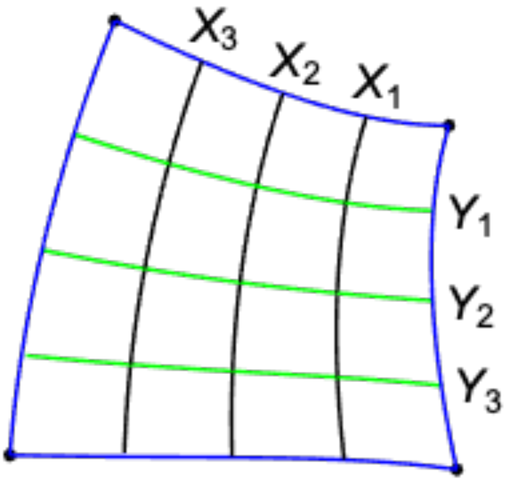
Set up

```
Point00 = {0, 0, 0};(*According to Point*)
Point03 = {-0.7077264466498845, 0, 0.5008767232876717};
Point30 = {0, -0.7077264466498845, -0.5008767232876717};
Point33 = {-1.0013149889663, -1.0013149889663, 0};

MaxTangentPoint00 = {-1, 0, 0};(*According to LoC*)
MaxTangentPoint03 = {-0.5770131138020663, 0, 0.8167348814030436};
MaxTangentPoint30 = {-1, 0, 0};
MaxTangentPoint33 =
  {-0.6664720667513934`, -0.3335279332486068`, 0.6667639027294741`};

MinTangentPoint00 = {0, -1, 0};
MinTangentPoint30 = {0., -0.5770131138020663, -0.8167348814030436};
MinTangentPoint03 = {0, -1, 0};
MinTangentPoint33 =
  {-0.3335279332486068`, -0.6664720667513934`, -0.6667639027294741`};


DPbval = {{1 / 5}, {3 / 40, 9 / 40}, {44 / 45, -56 / 15, 32 / 9},
    {19 372 / 6561, -25 360 / 2187, 64 448 / 6561, -212 / 729},
    {9017 / 3168, -355 / 33, 46 732 / 5247, 49 / 176, -5103 / 18 656},
    {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84}};
DPcval = {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84, 0};
DPaval  = {1 / 5, 3 / 10, 4 / 5, 8 / 9, 1, 1};
DPerrval =
  {71 / 57 600, 0, -71 / 16 695, 71 / 1920, -17 253 / 339 200, 22 / 525, -1 / 40};
DPCoefficients[5, p_] :=
  N[{DPbval, DPcval, DPaval, DPerrval}, p];

Surf[u_, v_][s_] := {u[s], v[s], u[s] ^2 - v[s] ^2};
DuSurf[u_, v_][s_] := {1, 0, 2 u[s]};
DvSurf[u_, v_][s_] := {0, 1, -2 v[s]};
Du2Surf[u_, v_][s_] := {0, 0, 2};
Dv2Surf[u_, v_][s_] := {0, 0, -2};
DuDvSurf[u_, v_][s_] := {0, 0, 0};
CrossN[u_, v_][s_] := Cross[DuSurf[u, v][s], DvSurf[u, v][s]];
ScalarN[u_, v_][s_] := Sqrt[CrossN[u, v][s].CrossN[u, v][s]];
NormalSurf[u_, v_][s_] := CrossN[u, v][s] / ScalarN[u, v][s];
```

```
SurfE[u_, v_][s_] := DuSurf[u, v][s].DuSurf[u, v][s];
SurfF[u_, v_][s_] := DuSurf[u, v][s].DvSurf[u, v][s];
SurfG[u_, v_][s_] := DvSurf[u, v][s].DvSurf[u, v][s];
SurfL[u_, v_][s_] := Du2Surf[u, v][s].NormalSurf[u, v][s];
SurfM[u_, v_][s_] := DuDvSurf[u, v][s].NormalSurf[u, v][s];
SurfN[u_, v_][s_] := Dv2Surf[u, v][s].NormalSurf[u, v][s];
GaussCurva[u_, v_][s_] := (SurfL[u, v][s] × SurfN[u, v][s] - SurfM[u, v][s] ^ 2) /
   (SurfE[u, v][s] × SurfG[u, v][s] - SurfF[u, v][s] ^ 2);
MeanCurva[u_, v_][s_] := 1 / 2 (SurfE[u, v][s] × SurfN[u, v][s] +
      SurfG[u, v][s] × SurfL[u, v][s] - 2 SurfF[u, v][s] × SurfM[u, v][s]) /
    (SurfE[u, v][s] × SurfG[u, v][s] - SurfF[u, v][s] ^ 2);
MaxPrinci[u_, v_][s_] := MeanCurva[u, v][s] +
   Sqrt[MeanCurva[u, v][s] ^ 2 - GaussCurva[u, v][s]];
MinPrinci[u_, v_][s_] := MeanCurva[u, v][s] -
   Sqrt[MeanCurva[u, v][s] ^ 2 - GaussCurva[u, v][s]];
Princi[a_, u_, v_][s_] := Piecewise[
   {{MaxPrinci[u, v][s], a === max}, {MinPrinci[u, v][s], a === min}}];

Etaη[a_, u_, v_][s_] :=
  1 / Sqrt[SurfE[u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s]) ^ 2 -
     2 SurfF[u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s])
      (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s]) +
     SurfG[u, v][s] (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s]) ^ 2];
Miuμ[a_, u_, v_][s_] := 1 / Sqrt[SurfE[u, v][s]
      (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s]) ^ 2 -
     2 SurfF[u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s])
      (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s]) +
     SurfG[u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s]) ^ 2];

DuDs[a_, u_, v_][s_] :=
 Piecewise[{{Etaη[a, u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s]),
    Sign[(SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s])]
      (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s]) ≥
     Sign[(SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])]
      (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])}},
  Miuμ[a, u, v][s] (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])]
DvDs[a_, u_, v_][s_] := Piecewise[
  {{-Etaη[a, u, v][s] (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s]),
    Sign[(SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s])]
      (SurfL[u, v][s] - Princi[a, u, v][s] × SurfE[u, v][s]) ≥
     Sign[(SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])]
      (SurfN[u, v][s] - Princi[a, u, v][s] × SurfG[u, v][s])}},
  -Miuμ[a, u, v][s] (SurfM[u, v][s] - Princi[a, u, v][s] × SurfF[u, v][s])]

d1uds1[a_, b_, c_] := DuDs[a, u, v][s] /. {u[s] → b, v[s] → c}
d1vds1[a_, b_, c_] := DvDs[a, u, v][s] /. {u[s] → b, v[s] → c}
```

```
SolvedudvDP[a_] := ParametricNDSolve[
  {u'[ArcL] == DuDs[a, u, v][ArcL], v'[ArcL] == DvDs[a, u, v][ArcL],
   u[0] == iniu, v[0] == iniv},
  {u, v}, {ArcL, 0, 2}, {iniu, iniv}, Method →
   {"ExplicitRungeKutta", "DifferenceOrder" → 5, "Coefficients" → DPCoefficients,
    "EmbeddedDifferenceOrder" → 4, "StepSizeControlParameters" -> {1, 0},
    "StepSizeSafetyFactors" → {1, 1}, "StepSizeRatioBounds" → {1 / 25, 25},
    "StiffnessTest" → False, "DiscontinuityProcessing" → False},
  MaxStepSize → 0.01, MaxSteps → Automatic]

solution41 = SolvedudvDP[max];
solution42 = SolvedudvDP[min];

MaximumusDP[t_, s_] :=
 u[u[0, 0][t] /. solution42, v[0, 0][t] /. solution42][s] /. solution41
MaximumvsDP[t_, s_] :=
 v[u[0, 0][t] /. solution42, v[0, 0][t] /. solution42][s] /. solution41
MinimumusDP[t_, s_] :=
 u[u[0, 0][t] /. solution41, v[0, 0][t] /. solution41][s] /. solution42
MinimumvsDP[t_, s_] :=
 v[u[0, 0][t] /. solution41, v[0, 0][t] /. solution41][s] /. solution42

MaximumusDP[0, .9]

-0.707726

{MaximumusDP[0.22, 0], MaximumvsDP[0.22, 0],
 MaximumusDP[0.22, 0] ^ 2 - MaximumvsDP[0.22, 0] ^ 2}
{MaximumusDP[0.44, 0], MaximumvsDP[0.44, 0],
 MaximumusDP[0.44, 0] ^ 2 - MaximumvsDP[0.44, 0] ^ 2}
{MaximumusDP[0.66, 0], MaximumvsDP[0.66, 0],
 MaximumusDP[0.66, 0] ^ 2 - MaximumvsDP[0.66, 0] ^ 2}

{0., -0.213665, -0.0456525}

{0., -0.400553, -0.160442}

{0., -0.559105, -0.312598}
```

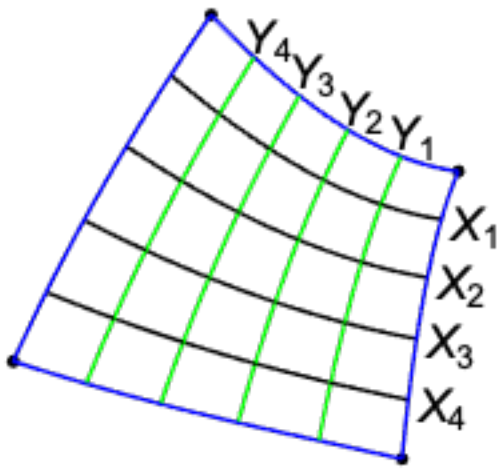
```
Show[ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MaximumusDP[0, s], v[s] → MaximumvsDP[0, s]},
  {s, 0, .9}, PlotStyle → Blue], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MaximumusDP[0.9, s], v[s] → MaximumvsDP[0.9, s]},
  {s, 0, 1.1920739744274536`}, PlotStyle → Blue],
 ParametricPlot3D[Surf[u, v][s] /.
   {u[s] → MaximumusDP[0.22, s], v[s] → MaximumvsDP[0.22, s]},
  {s, 0, .93}, PlotStyle → Black], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MaximumusDP[0.44, s], v[s] → MaximumvsDP[0.44, s]},
  {s, 0, 1}, PlotStyle → Black],
 ParametricPlot3D[Surf[u, v][s] /.
   {u[s] → MaximumusDP[0.66, s], v[s] → MaximumvsDP[0.66, s]},
  {s, 0, 1.08}, PlotStyle → Black], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MinimumusDP[0, s], v[s] → MinimumvsDP[0, s]},
  {s, 0, .9}, PlotStyle → Blue], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MinimumusDP[0.9, s], v[s] → MinimumvsDP[0.9, s]},
  {s, 0, 1.1920739744274536`}, PlotStyle → Blue], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MinimumusDP[0.22, s], v[s] → MinimumvsDP[0.22, s]},
  {s, 0, .93}, PlotStyle → Green], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MinimumusDP[0.44, s], v[s] → MinimumvsDP[0.44, s]},
  {s, 0, 1}, PlotStyle → Green], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MinimumusDP[0.66, s], v[s] → MinimumvsDP[0.66, s]},
  {s, 0, 1.08}, PlotStyle → Green],
 Graphics3D[{{PointSize[Large], Point[{Point00, Point03, Point30, Point33}]},
   Text[Style["X1", Large], {0.`, -0.21366455477174995`, -0.04565254196581013`} +
     {0.1, 0, 0}], Text[Style["X2", Large],
    {0.`, -0.4005526798927377`, -0.16044244936925398`} + {0.1, 0, 0}],
   Text[Style["X3", Large], {0.`, -0.5591045516980968`, -0.3125978997295298`} +
     {0.1, 0, 0}], Text[Style["Y1", Large],
    {-0.21366455477174995`, 0.`, 0.04565254196581013`} + {0, .1, 0}],
   Text[Style["Y2", Large], {-0.4005526798927377`, 0.`, 0.16044244936925398`} +
     {0, .1, 0}], Text[Style["Y3", Large],
    {-0.5591045516980968`, 0.`, 0.3125978997295298`} + {0, 0.1, 0}]
  }], PlotRange → All, Boxed → False, Axes → False]
```




```
Surf[u, v][s] /. {u[s] → MaximumusDP[0.18, 0], v[s] → MaximumvsDP[0.18, 0]}
Surf[u, v][s] /. {u[s] → MaximumusDP[0.36, 0], v[s] → MaximumvsDP[0.36, 0]}
Surf[u, v][s] /. {u[s] → MaximumusDP[0.54, 0], v[s] → MaximumvsDP[0.54, 0]}
Surf[u, v][s] /. {u[s] → MaximumusDP[0.72, 0], v[s] → MaximumvsDP[0.72, 0]}
Surf[u, v][s] /. {u[s] → MinimumusDP[0.18, 0], v[s] → MinimumvsDP[0.18, 0]}
Surf[u, v][s] /. {u[s] → MinimumusDP[0.36, 0], v[s] → MinimumvsDP[0.36, 0]}
Surf[u, v][s] /. {u[s] → MinimumusDP[0.54, 0], v[s] → MinimumvsDP[0.54, 0]}
Surf[u, v][s] /. {u[s] → MinimumusDP[0.72, 0], v[s] → MinimumvsDP[0.72, 0]}
```

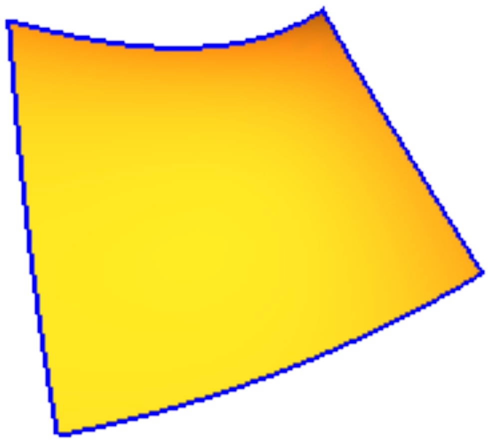
```
{0., -0.176406, -0.031119}

{0., -0.336163, -0.113006}

{0., -0.475739, -0.226328}

{0., -0.598329, -0.357997}

{-0.176406, 0., 0.031119}

{-0.336163, 0., 0.113006}

{-0.475739, 0., 0.226328}

{-0.598329, 0., 0.357997}
```

```
Show[ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MaximumusDP[0, s], v[s] → MaximumvsDP[0, s]},
  {s, 0, .9}, PlotStyle → Blue], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MaximumusDP[0.9, s], v[s] → MaximumvsDP[0.9, s]},
  {s, 0, 1.1920739744274536`}, PlotStyle → Blue],
 ParametricPlot3D[Surf[u, v][s] /.
   {u[s] → MaximumusDP[0.18, s], v[s] → MaximumvsDP[0.18, s]},
  {s, 0, .92}, PlotStyle → Black], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MaximumusDP[0.36, s], v[s] → MaximumvsDP[0.36, s]},
  {s, 0, .97}, PlotStyle → Black],
 ParametricPlot3D[Surf[u, v][s] /. {u[s] → MaximumusDP[0.54, s],
    v[s] → MaximumvsDP[0.54, s]}, {s, 0, 1.04}, PlotStyle → Black],
 ParametricPlot3D[Surf[u, v][s] /. {u[s] → MaximumusDP[0.72, s],
    v[s] → MaximumvsDP[0.72, s]},
  {s, 0, 1.115}, PlotStyle → Black], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MinimumusDP[0, s], v[s] → MinimumvsDP[0, s]},
  {s, 0, .9}, PlotStyle → Blue], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MinimumusDP[0.9, s], v[s] → MinimumvsDP[0.9, s]},
  {s, 0, 1.1920739744274536`}, PlotStyle → Blue],
 ParametricPlot3D[Surf[u, v][s] /.
   {u[s] → MinimumusDP[0.18, s], v[s] → MinimumvsDP[0.18, s]},
  {s, 0, .92}, PlotStyle → Green], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MinimumusDP[0.36, s], v[s] → MinimumvsDP[0.36, s]},
  {s, 0, .97}, PlotStyle → Green],
 ParametricPlot3D[Surf[u, v][s] /. {u[s] → MinimumusDP[0.54, s],
    v[s] → MinimumvsDP[0.54, s]}, {s, 0, 1.04}, PlotStyle → Green],
 ParametricPlot3D[Surf[u, v][s] /. {u[s] → MinimumusDP[0.72, s],
    v[s] → MinimumvsDP[0.72, s]}, {s, 0, 1.115}, PlotStyle → Green],
 Graphics3D[{{PointSize[Large], Point[{Point00, Point03, Point30, Point33}]},
   Text[Style["X1", Large],
    {0.`, -0.1764057531631565`, -0.031118989749060494`} + {0.1, 0, 0}],
   Text[Style["X2", Large], {0.`, -0.33616302504958406`, -0.11300557941048728`} +
     {0.1, 0, 0}], Text[Style["X3", Large],
    {0.`, -0.47573948940047167`, -0.2263280617750215`} + {0.1, 0, 0}],
   Text[Style["X4", Large], {0.`, -0.5983288570755986`, -0.35799742120939204`} +
     {0.1, 0, 0}], Text[Style["Y1", Large],
    {-0.1764057531631565`, 0.`, 0.031118989749060494`} + {0, .1, 0}],
   Text[Style["Y2", Large],
    {-0.33616302504958406`, 0.`, 0.11300557941048728`} + {0, .1, 0}],
   Text[Style["Y3", Large], {-0.47573948940047167`, 0.`, 0.2263280617750215`} +
     {0, 0.1, 0}], Text[Style["Y4", Large],
    {-0.5983288570755986`, 0.`, 0.35799742120939204`} + {0, .1, 0}]
  }], PlotRange → All, Boxed → False, Axes → False]
```

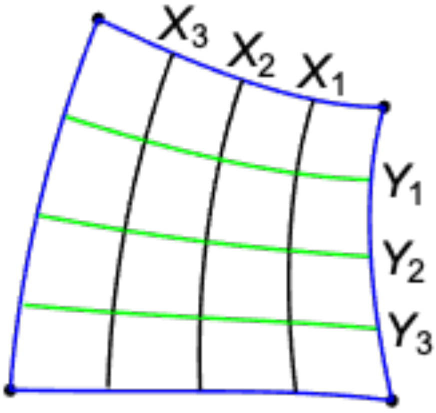

Y_4
Y_3
Y_2
Y_1
X_1
X_2
X_3
X_4

```
DPbval = {{1 / 5}, {3 / 40, 9 / 40}, {44 / 45, -56 / 15, 32 / 9},
    {19 372 / 6561, -25 360 / 2187, 64 448 / 6561, -212 / 729},
    {9017 / 3168, -355 / 33, 46 732 / 5247, 49 / 176, -5103 / 18 656},
    {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84}};
DPcval = {35 / 384, 0, 500 / 1113, 125 / 192, -2187 / 6784, 11 / 84, 0};
DPaval = {1 / 5, 3 / 10, 4 / 5, 8 / 9, 1, 1};
DPerrval =
   {71 / 57 600, 0, -71 / 16 695, 71 / 1920, -17 253 / 339 200, 22 / 525, -1 / 40};
DPCoefficients[5, p_] :=
   N[{DPbval, DPcval, DPaval, DPerrval}, p];
CoonsMax0[u_] := {MaximumusDP[0, u * .9], MaximumvsDP[0, u * .9],
    MaximumusDP[0, u * .9] ^ 2 - MaximumvsDP[0, u * .9] ^ 2};
CoonsMax1[u_] := {MaximumusDP[0.9, u * 1.1920739744274533`], MaximumvsDP[0.9,
     u * 1.1920739744274533`], MaximumusDP[0.9, u * 1.1920739744274533`] ^ 2 -
     MaximumvsDP[0.9, u * 1.1920739744274533`] ^ 2};
CoonsMin0[v_] := {MinimumusDP[0, v * .9], MinimumvsDP[0, v * .9],
    MinimumusDP[0, v * .9] ^ 2 - MinimumvsDP[0, v * .9] ^ 2};
CoonsMin1[v_] := {MinimumusDP[0.9, v * 1.1920739744274533`], MinimumvsDP[0.9,
     v * 1.1920739744274533`], MinimumusDP[0.9, v * 1.1920739744274533`] ^ 2 -
     MinimumvsDP[0.9, v * 1.1920739744274533`] ^ 2};

DCoonsMax0Du[u_] := D[CoonsMax0[u], u];
DCoonsMax1Du[u_] := D[CoonsMax1[u], u];
DCoonsMin0Dv[v_] := D[CoonsMin0[v], v];
DCoonsMin1Dv[v_] := D[CoonsMin1[v], v];


Coonslc1[u_, v_] := (1 - v) * CoonsMax0[u] + v * CoonsMax1[u];
Coonsld1[u_, v_] := (1 - u) * CoonsMin0[v] + u * CoonsMin1[v];
Coonsb1[u_, v_] := CoonsMax0[0] * (1 - u) * (1 - v) +
    CoonsMax0[1] * u * (1 - v) + CoonsMax1[0] * (1 - u) * v + CoonsMax1[1] * u * v;
Coonspatch1[u_, v_] := Coonslc1[u, v] + Coonsld1[u, v] - Coonsb1[u, v];


HPSurf[u_, v_][s_] := Coonspatch1[u[s], v[s]]
```

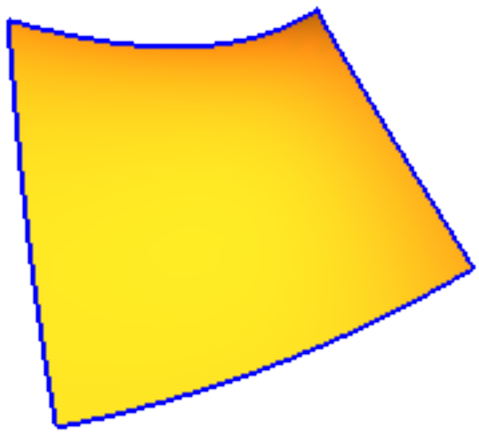
```
Show[ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MaximumusDP[0, s], v[s] → MaximumvsDP[0, s]},
  {s, 0, .9}, PlotStyle → Blue], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MaximumusDP[0.9, s], v[s] → MaximumvsDP[0.9, s]},
  {s, 0, 1.1920739744274536`}, PlotStyle → Blue],
 ParametricPlot3D[Surf[u, v][s] /. {u[s] → MinimumusDP[0, s],
    v[s] → MinimumvsDP[0, s]}, {s, 0, .9}, PlotStyle → Blue], ParametricPlot3D[
  Surf[u, v][s] /. {u[s] → MinimumusDP[0.9, s], v[s] → MinimumvsDP[0.9, s]},
  {s, 0, 1.1920739744274536`}, PlotStyle → Blue],
 ParametricPlot3D[HPSurf[u, v][s], {v[s], 0, 1}, {u[s], 0, 1}, Axes → False,
  Boxed → False, Mesh → None], Axes → False, Boxed → False, PlotRange → All]
```

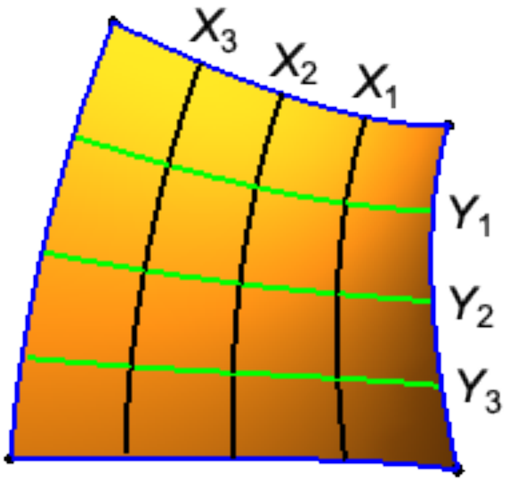


(a)

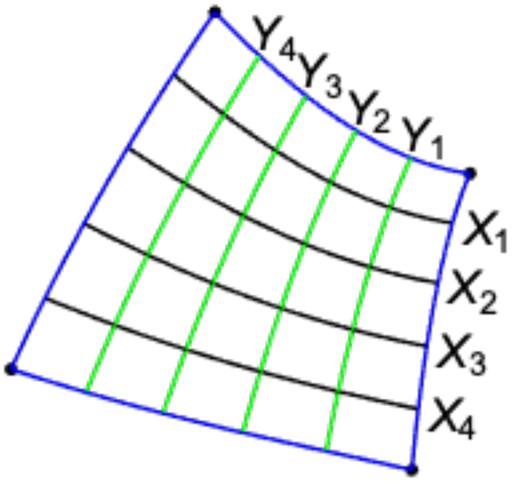
Show[

,

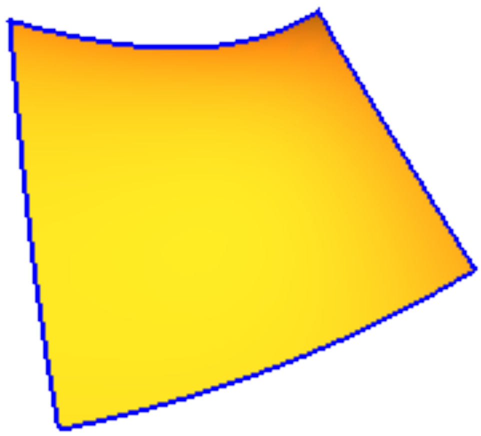

]


X_3
X_2
X_1
Y_1
Y_2
Y_3

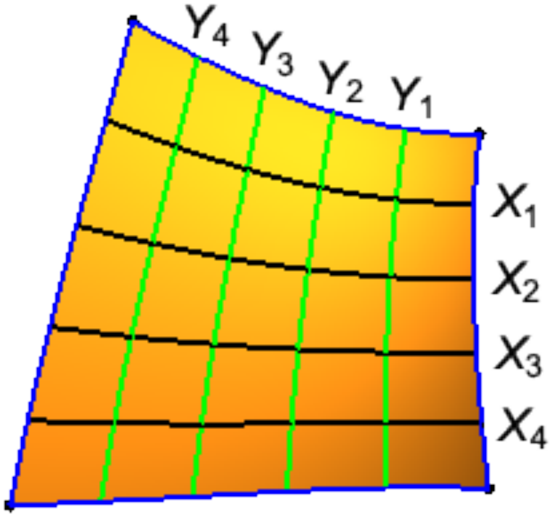
Show[



,



]



- (b)

Set up

```
kernelcodeCS = "
#define min 0
#define max 1
#define vvv 0
#define uuu 1
#define sign1 -1
#define h 0.001
#define n 1201
#define AlphaLAC (2.0)
#define BetaLASC (2.0)
#define InitRCLAC (1.0)
#define OmegaLASC (1.0)
#define Lambda_Max_LAC (12.718433357775211)
#define ArcLength_Max_LAC (6.764357890136049)
#define Lambda_Min_LAC (12.718433357775211)
#define ArcLength_Min_LAC (6.764357890136049)
#define Lambda_Max_LASC (6.285247558593747)
#define InitRT_Max_LASC (7.303131835937503)
#define ArcLength_Max_LASC (3.1084204465150833)
#define Lambda_Min_LASC (6.285247558593747)
#define InitRT_Min_LASC (7.303131835937503)
#define ArcLength_Min_LASC (3.1084204465150833)
```

```
#define Point00x (0.0)
#define Point00y (0.0)
#define Point00z (0.0)
#define Point03x (-0.7077264466498845)
#define Point03y (0.0)
#define Point03z (0.5008767232876717)
#define Point30x (0.0)
#define Point30y (-0.7077264466498845)
#define Point30z (-0.5008767232876717)
#define Point33x (-1.0013149889663)
#define Point33y (-1.0013149889663)
#define Point33z (0.0)

#define maxPoint00x (0.0)
#define maxPoint00y (0.0)
#define maxPoint00z (0.0)
#define maxPoint03x (-0.7077264466498845)
#define maxPoint03y (0.0)
#define maxPoint03z (0.5008767232876717)
#define maxPoint30x (0.0)
#define maxPoint30y (-0.7077264466498845)
#define maxPoint30z (-0.5008767232876717)
#define maxPoint33x (-1.0013149889663)
#define maxPoint33y (-1.0013149889663)
#define maxPoint33z (0.0)


#define ScaleToOrigin_Max_LAC (0.13192365851313362)
#define ScaleToOrigin_Min_LAC (0.13192365851313362)
#define ScaleToOrigin_Max_LASC (0.38164320308766897)
#define ScaleToOrigin_Min_LASC (0.38164320308766897)

#define inidxdu1
    ((1-pv)*(maxPoint00x-maxPoint03x)+pv*(maxPoint30x-maxPoint33x)-v1_x1+
    v2_x1+(1-pv)*u1_t1+pv*u2_t1)
#define inidxdu2
    ((1-pv)*(maxPoint00y-maxPoint03y)+pv*(maxPoint30y-maxPoint33y)-v1_x2+
    v2_x2+(1-pv)*u1_t2+pv*u2_t2)
#define inidxdu3
    ((1-pv)*(maxPoint00z-maxPoint03z)+pv*(maxPoint30z-maxPoint33z)-v1_x3+
    v2_x3+(1-pv)*u1_t3+pv*u2_t3)
#define inidxdv1
    ((1-pu)*(maxPoint00x-maxPoint30x)+pu*(maxPoint03x-maxPoint33x)-u1_x1+
    u2_x1+(1-pu)*v1_t1+pu*v2_t1)
#define inidxdv2
    ((1-pu)*(maxPoint00y-maxPoint30y)+pu*(maxPoint03y-maxPoint33y)-u1_x2+
    u2_x2+(1-pu)*v1_t2+pu*v2_t2)
```

```
#define inidxdv3
    ((1-pu)*(maxPoint00z-maxPoint30z)+pu*(maxPoint03z-maxPoint33z)-u1_x3+
    u2_x3+(1-pu)*v1_t3+pu*v2_t3)

#define inid2xdu21 ((1-pv)*u1_dt1+pv*u2_dt1)
#define inid2xdu22 ((1-pv)*u1_dt2+pv*u2_dt2)
#define inid2xdu23 ((1-pv)*u1_dt3+pv*u2_dt3)
#define inid2xdv21 ((1-pu)*v1_dt1+pu*v2_dt1)
#define inid2xdv22 ((1-pu)*v1_dt2+pu*v2_dt2)
#define inid2xdv23 ((1-pu)*v1_dt3+pu*v2_dt3)
#define inid2xdudv1
    (-maxPoint00x+maxPoint03x+maxPoint30x-maxPoint33x-u1_t1+u2_t1-v1_t1+v2_t1)
#define inid2xdudv2
    (-maxPoint00y+maxPoint03y+maxPoint30y-maxPoint33y-u1_t2+u2_t2-v1_t2+v2_t2)
#define inid2xdudv3
    (-maxPoint00z+maxPoint03z+maxPoint30z-maxPoint33z-u1_t3+u2_t3-v1_t3+v2_t3)


#define dxdu1
    ((1-pv[i])*(maxPoint00x-maxPoint03x)+pv[i]*(maxPoint30x-maxPoint33x)-
    v1_x1[i]+v2_x1[i]+(1-pv[i])*u1_t1[i]+pv[i]*u2_t1[i])
#define dxdu2
    ((1-pv[i])*(maxPoint00y-maxPoint03y)+pv[i]*(maxPoint30y-maxPoint33y)-
    v1_x2[i]+v2_x2[i]+(1-pv[i])*u1_t2[i]+pv[i]*u2_t2[i])
#define dxdu3
    ((1-pv[i])*(maxPoint00z-maxPoint03z)+pv[i]*(maxPoint30z-maxPoint33z)-
    v1_x3[i]+v2_x3[i]+(1-pv[i])*u1_t3[i]+pv[i]*u2_t3[i])
#define dxdv1
    ((1-pu[i])*(maxPoint00x-maxPoint30x)+pu[i]*(maxPoint03x-maxPoint33x)-
    u1_x1[i]+u2_x1[i]+(1-pu[i])*v1_t1[i]+pu[i]*v2_t1[i])
#define dxdv2
    ((1-pu[i])*(maxPoint00y-maxPoint30y)+pu[i]*(maxPoint03y-maxPoint33y)-
    u1_x2[i]+u2_x2[i]+(1-pu[i])*v1_t2[i]+pu[i]*v2_t2[i])
#define dxdv3
    ((1-pu[i])*(maxPoint00z-maxPoint30z)+pu[i]*(maxPoint03z-maxPoint33z)-
    u1_x3[i]+u2_x3[i]+(1-pu[i])*v1_t3[i]+pu[i]*v2_t3[i])


#define d2xdu21 ((1-pv[i])*u1_dt1[i]+pv[i]*u2_dt1[i])
#define d2xdu22 ((1-pv[i])*u1_dt2[i]+pv[i]*u2_dt2[i])
#define d2xdu23 ((1-pv[i])*u1_dt3[i]+pv[i]*u2_dt3[i])
#define d2xdv21 ((1-pu[i])*v1_dt1[i]+pu[i]*v2_dt1[i])
#define d2xdv22 ((1-pu[i])*v1_dt2[i]+pu[i]*v2_dt2[i])
#define d2xdv23 ((1-pu[i])*v1_dt3[i]+pu[i]*v2_dt3[i])
#define d2xdudv1
    (-maxPoint00x+maxPoint03x+maxPoint30x-maxPoint33x-u1_t1[i]+u2_t1[i]-
    v1_t1[i]+v2_t1[i])
```

```
#define d2xdudv2
    (-maxPoint00y+maxPoint03y+maxPoint30y-maxPoint33y-u1_t2[i]+u2_t2[i]-
    v1_t2[i]+v2_t2[i])
#define d2xdudv3
    (-maxPoint00z+maxPoint03z+maxPoint30z-maxPoint33z-u1_t3[i]+u2_t3[i]-
    v1_t3[i]+v2_t3[i])


#define d3xdu31 ((1-pv[i])*u1_d2t1[i]+pv[i]*u2_d2t1[i])
#define d3xdu32 ((1-pv[i])*u1_d2t2[i]+pv[i]*u2_d2t2[i])
#define d3xdu33 ((1-pv[i])*u1_d2t3[i]+pv[i]*u2_d2t3[i])
#define d3xdv31 ((1-pu[i])*v1_d2t1[i]+pu[i]*v2_d2t1[i])
#define d3xdv32 ((1-pu[i])*v1_d2t2[i]+pu[i]*v2_d2t2[i])
#define d3xdv33 ((1-pu[i])*v1_d2t3[i]+pu[i]*v2_d2t3[i])
#define d3xdu2dv1 (-u1_dt1[i]+u2_dt1[i])
#define d3xdu2dv2 (-u1_dt2[i]+u2_dt2[i])
#define d3xdu2dv3 (-u1_dt3[i]+u2_dt3[i])
#define d3xdudv21 (-v1_dt1[i]+v2_dt1[i])
#define d3xdudv22 (-v1_dt2[i]+v2_dt2[i])
#define d3xdudv23 (-v1_dt3[i]+v2_dt3[i])

#define d4xdu41 ((1-pv[i])*u1_d3t1[i]+pv[i]*u2_d3t1[i])
#define d4xdu42 ((1-pv[i])*u1_d3t2[i]+pv[i]*u2_d3t2[i])
#define d4xdu43 ((1-pv[i])*u1_d3t3[i]+pv[i]*u2_d3t3[i])
#define d4xdv41 ((1-pu[i])*v1_d3t1[i]+pu[i]*v2_d3t1[i])
#define d4xdv42 ((1-pu[i])*v1_d3t2[i]+pu[i]*v2_d3t2[i])
#define d4xdv43 ((1-pu[i])*v1_d3t3[i]+pu[i]*v2_d3t3[i])
#define d4xdu3dv1 (-u1_d2t1[i]+u2_d2t1[i])
#define d4xdu3dv2 (-u1_d2t2[i]+u2_d2t2[i])
#define d4xdu3dv3 (-u1_d2t3[i]+u2_d2t3[i])
#define d4xdudv31 (-v1_d2t1[i]+v2_d2t1[i])
#define d4xdudv32 (-v1_d2t2[i]+v2_d2t2[i])
#define d4xdudv33 (-v1_d2t3[i]+v2_d2t3[i])
#define d4xdu2dv21 (0.0)
#define d4xdu2dv22 (0.0)
#define d4xdu2dv23 (0.0)

#define Coons_x1
    ((1-pv[index])*u1_x1[index]+pv[index]*u2_x1[index]+(1-pu[index])*v1_x1[
    index]+pu[index]*v2_x1[index]-((1-pu[index])*(1-pv[index])*maxPoint00x+
    pu[index]*(1-pv[index])*maxPoint03x+(1-pu[index])*pv[index]*maxPoint30x+
    pu[index]*pv[index]*maxPoint33x))
#define Coons_x2
    ((1-pv[index])*u1_x2[index]+pv[index]*u2_x2[index]+(1-pu[index])*v1_x2[
    index]+pu[index]*v2_x2[index]-((1-pu[index])*(1-pv[index])*maxPoint00y+
    pu[index]*(1-pv[index])*maxPoint03y+(1-pu[index])*pv[index]*maxPoint30y+
    pu[index]*pv[index]*maxPoint33y))
```

```
#define Coons_x3
    ((1-pv[index])*u1_x3[index]+pv[index]*u2_x3[index]+(1-pu[index])*v1_x3[
    index]+pu[index]*v2_x3[index]-((1-pu[index])*(1-pv[index])*maxPoint00z+
    pu[index]*(1-pv[index])*maxPoint03z+(1-pu[index])*pv[index]*maxPoint30z+
    pu[index]*pv[index]*maxPoint33z))

#define repiniSurNorvec
    iniSurNorvec(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3
    ,v2_t1,v2_t2,v2_t3,normv_C)
#define repinicoefE
    inicoefE(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3)
#define repinicoefF
    inicoefF(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3)
#define repinicoefG
    inicoefG(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3)
#define repinicoefL
    inicoefL(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repinicoefM
    inicoefM(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repinicoefN
    inicoefN(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniMeanCurva
    iniMeanCurva(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniGaussCurva
    iniGaussCurva(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
```

```
#define repiniMaxPrinci
    iniMaxPrinci(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniMinPrinci
    iniMinPrinci(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniPrinci
    iniPrinci(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3
    ,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniEta
    iniEta(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniMiu
    iniMiu(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniDuDs
    iniDuDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniDvDs
    iniDvDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)

#define repSurNorvec
    SurNorvec(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,normv_C)
#define repDSurNvDu
    DSurNvDu(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,dnormvdu_C)
#define repDSurNvDv
    DSurNvDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
```

```
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,dnormvdv_C)
#define repD2SurNvDu2
    D2SurNvDu2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    d2normvdu2_C)
#define repD2SurNvDuDv
    D2SurNvDuDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    d2normvdudv_C)
#define repD2SurNvDv2
    D2SurNvDv2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    d2normvdv2_C)
#define repScalarSurNor
    ScalarSurNor(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    Scal)
#define repcoefE
    coefE(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3)
#define repDEDu
    DEDu(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repDEDv
    DEDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repD2EDu2
    D2EDu2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
```

```
        u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
        u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
        v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
        v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repD2EDuDv
        D2EDuDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3
        ,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
        u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
        v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
        v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repD2EDv2
        D2EDv2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
        u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
        u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
        v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
        v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repcoefF
        coefF(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
        u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
        v2_t1,v2_t2,v2_t3)
#define repDFDu
        DFDu(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
        u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
        v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
        v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repDFDv
        DFDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
        u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
        v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
        v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repD2FDu2
        D2FDu2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
        u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
        u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
        v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
        v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repD2FDuDv
        D2FDuDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3
        ,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
        u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
        v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
        v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repD2FDv2
        D2FDv2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
        u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
        u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
        v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
```

```
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repcoefG
    coefG(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3)
#define repDGDu
    DGDu(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repDGDv
    DGDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repD2GDu2
    D2GDu2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repD2GDuDv
    D2GDuDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3
    ,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repD2GDv2
    D2GDv2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repcoefL
    coefL(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repcoefM
    coefM(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repcoefN
    coefN(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
```

```
        v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
    #define repGaussCurva
        GaussCurva(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
        u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
        v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
        v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
    #define repMeanCurva
        MeanCurva(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
        u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
        v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
        v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
    #define repMaxPrinci
        MaxPrinci(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
        u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
        v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
        v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
    #define repMinPrinci
        MinPrinci(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
        u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
        v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
        v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
    #define repPrinci
        Princi(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
        u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
        v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
        v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
    #define repDcoefL_2
        DcoefL_2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
        u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
        ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
        u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
        ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
        ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
        v2_d3t2,v2_d3t3,CoeffL)
    #define repDcoefM_2
        DcoefM_2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
        u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
        ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
        u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
        ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
        ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
        v2_d3t2,v2_d3t3,CoeffM)
    #define repDcoefN_2
        DcoefN_2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
        u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
        ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
        u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
```

```
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,CoeffN)
#define repDGaussCurva_2
    DGaussCurva_2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,GCurva)
#define repDMeanCurva_2
    DMeanCurva_2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,MCurva)
#define repDPrinci_2
    DPrinci_2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,optimum,PCurva)
#define repEta
    Eta(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repMiu
    Miu(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDuDs
    DuDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDuDs1
    DuDs(i+1,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
```

```
#define repDvDs
    DvDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDvDs1
    DvDs(i+1,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDEDs
    DEDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDFDs
    DFDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDGDs
    DGDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDLDs
    DLDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repDMDs
    DMDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repDNDs
    DNDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
```

```
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repDPrinciDs
    DPrinciDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,optimum)
#define repDvDt
    DvDt(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDuDt
    DuDt(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDSurDs
    DSurDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum,tangv)
#define repD2uDs2
    D2uDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repD2vDs2
    D2vDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repGeoCur
    GeoCur(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
```

```
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repCurvecurvature
    Curvecurvature(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2
    ,u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,
    u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,
    u2_d3t1,u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,
    v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,
    v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,
    v2_d2t3,v2_d3t1,v2_d3t2,v2_d3t3,optimum)
#define repD2EDs2
    D2EDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repD2FDs2
    D2FDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repD2GDs2
    D2GDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repD2LDs2
    D2LDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repD2MDs2
    D2MDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
```

```
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repD2NDs2
    D2NDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repD2PrinciDs2
    D2PrinciDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,optimum)
#define repD2SurDs2
    D2SurDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,optimum,dtangds)
#define repCurvenormaloS
    CurvenormaloS(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,optimum,norv)
#define repCurvebinormaloS
    CurvebinormaloS(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,
    u1_dt2,u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,
    u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,
    u2_d2t3,u2_d3t1,u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,
    v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,
    v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2
    ,v2_d2t3,v2_d3t1,v2_d3t2,v2_d3t3,optimum,binorv)
#define repAlp2
    Alp2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
```

```
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum,Alp2_C)
#define repBeta2
    Beta2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)


#define opt1 (float pu,float pv,float u1_x1,float u1_x2,float u1_x3,float
    u1_t1,float u1_t2,float u1_t3,float u2_x1,float u2_x2,float u2_x3,float
    u2_t1,float u2_t2,float u2_t3,float v1_x1,float v1_x2,float
    v1_x3,float v1_t1,float v1_t2,float v1_t3,float v2_x1,float
    v2_x2,float v2_x3,float v2_t1,float v2_t2,float v2_t3,float normv_C[])
#define opt2 (float pu,float pv,float u1_x1,float u1_x2,float u1_x3,float
    u1_t1,float u1_t2,float u1_t3,float u2_x1,float u2_x2,float
    u2_x3,float u2_t1,float u2_t2,float u2_t3,float v1_x1,float
    v1_x2,float v1_x3,float v1_t1,float v1_t2,float v1_t3,float
    v2_x1,float v2_x2,float v2_x3,float v2_t1,float v2_t2,float v2_t3)
#define opt3 (float pu,float pv,float u1_x1,float u1_x2,float
    u1_x3,float u1_t1,float u1_t2,float u1_t3,float u1_dt1,float
    u1_dt2,float u1_dt3,float u2_x1,float u2_x2,float
    u2_x3,float u2_t1,float u2_t2,float u2_t3,float u2_dt1,float
    u2_dt2,float u2_dt3,float v1_x1,float v1_x2,float v1_x3,float
    v1_t1,float v1_t2,float v1_t3,float v1_dt1,float v1_dt2,float
    v1_dt3,float v2_x1,float v2_x2,float v2_x3,float v2_t1,float
    v2_t2,float v2_t3,float v2_dt1,float v2_dt2,float v2_dt3)
#define opt4 (float pu,float pv,float u1_x1,float u1_x2,float
    u1_x3,float u1_t1,float u1_t2,float u1_t3,float u1_dt1,float
    u1_dt2,float u1_dt3,float u2_x1,float u2_x2,float u2_x3,float
    u2_t1,float u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float
    u2_dt3,float v1_x1,float v1_x2,float v1_x3,float v1_t1,float
    v1_t2,float v1_t3,float v1_dt1,float v1_dt2,float v1_dt3,float
    v2_x1,float v2_x2,float v2_x3,float v2_t1,float v2_t2,float
    v2_t3,float v2_dt1,float v2_dt2,float v2_dt3,mint optimum)
#define opt5 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u2_x1[],float u2_x2[],float u2_x3[],float u2_t1[],float u2_t2[],float
    u2_t3[],float v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float
    v1_t2[],float v1_t3[],float v2_x1[],float v2_x2[],float
    v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float normv_C[])
#define opt6 (int i,float pu[],float pv[],float u1_x1[],float
```

```
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u2_x1[],float u2_x2[],float u2_x3[],float u2_t1[],float u2_t2[],float
    u2_t3[],float v1_x1[],float v1_x2[],float v1_x3[],float
    v1_t1[],float v1_t2[],float v1_t3[],float v2_x1[],float
    v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[])
#define opt7 (int i,float pu[],float pv[],float u1_x1[],float u1_x2[],float
    u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float u1_dt1[],float
    u1_dt2[],float u1_dt3[],float u2_x1[],float u2_x2[],float u2_x3[],float
    u2_t1[],float u2_t2[],float u2_t3[],float u2_dt1[],float u2_dt2[],float
    u2_dt3[],float v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float
    v1_t2[],float v1_t3[],float v1_dt1[],float v1_dt2[],float
    v1_dt3[],float v2_x1[],float v2_x2[],float v2_x3[],float v2_t1[],float
    v2_t2[],float v2_t3[],float v2_dt1[],float v2_dt2[],float v2_dt3[])
#define opt8 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u2_x1[],float
    u2_x2[],float u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float
    u2_dt1[],float u2_dt2[],float u2_dt3[],float v1_x1[],float
    v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float v1_t3[],float
    v1_dt1[],float v1_dt2[],float v1_dt3[],float v2_x1[],float
    v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float
    v2_t3[],float v2_dt1[],float v2_dt2[],float v2_dt3[],mint optimum)
#define opt9 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u2_x1[],float
    u2_x2[],float u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float
    u2_dt1[],float u2_dt2[],float u2_dt3[],float v1_x1[],float
    v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float v1_t3[],float
    v1_dt1[],float v1_dt2[],float v1_dt3[],float v2_x1[],float
    v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float
    v2_dt1[],float v2_dt2[],float v2_dt3[],mint optimum,float tangv_C[])
#define opt10 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u2_x1[],float
    u2_x2[],float u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float
    u2_dt1[],float u2_dt2[],float u2_dt3[],float v1_x1[],float
    v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float v1_t3[],float
    v1_dt1[],float v1_dt2[],float v1_dt3[],float v2_x1[],float
    v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float
    v2_dt1[],float v2_dt2[],float v2_dt3[],float dnormv_C[])
#define opt11 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u1_d2t1[],float
    u1_d2t2[],float u1_d2t3[],float u2_x1[],float u2_x2[],float
    u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float u2_dt1[],float
    u2_dt2[],float u2_dt3[],float u2_d2t1[],float u2_d2t2[],float
    u2_d2t3[],float v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float
```

```
    v1_t2[],float v1_t3[],float v1_dt1[],float v1_dt2[],float
    v1_dt3[],float v1_d2t1[],float v1_d2t2[],float v1_d2t3[],float
    v2_x1[],float v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float
    v2_t3[],float v2_dt1[],float v2_dt2[],float v2_dt3[],float
    v2_d2t1[],float v2_d2t2[],float v2_d2t3[],float d2normv_C[])
#define opt12 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u1_d2t1[],float
    u1_d2t2[],float u1_d2t3[],float u2_x1[],float u2_x2[],float
    u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float u2_dt1[],float
    u2_dt2[],float u2_dt3[],float u2_d2t1[],float u2_d2t2[],float
    u2_d2t3[],float v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float
    v1_t2[],float v1_t3[],float v1_dt1[],float v1_dt2[],float
    v1_dt3[],float v1_d2t1[],float v1_d2t2[],float v1_d2t3[],float
    v2_x1[],float v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float
    v2_t3[],float v2_dt1[],float v2_dt2[],float v2_dt3[],float
    v2_d2t1[],float v2_d2t2[],float v2_d2t3[],float Scal[])
#define opt13 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u1_d2t1[],float
    u1_d2t2[],float u1_d2t3[],float u2_x1[],float u2_x2[],float
    u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float u2_dt1[],float
    u2_dt2[],float u2_dt3[],float u2_d2t1[],float u2_d2t2[],float
    u2_d2t3[],float v1_x1[],float v1_x2[],float v1_x3[],float
    v1_t1[],float v1_t2[],float v1_t3[],float v1_dt1[],float
    v1_dt2[],float v1_dt3[],float v1_d2t1[],float v1_d2t2[],float
    v1_d2t3[],float v2_x1[],float v2_x2[],float v2_x3[],float v2_t1[],float
    v2_t2[],float v2_t3[],float v2_dt1[],float v2_dt2[],float
    v2_dt3[],float v2_d2t1[],float v2_d2t2[],float v2_d2t3[])
#define opt14 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u1_d2t1[],float
    u1_d2t2[],float u1_d2t3[],float u1_d3t1[],float u1_d3t2[],float
    u1_d3t3[],float u2_x1[],float u2_x2[],float u2_x3[],float
    u2_t1[],float u2_t2[],float u2_t3[],float u2_dt1[],float
    u2_dt2[],float u2_dt3[],float u2_d2t1[],float u2_d2t2[],float
    u2_d2t3[],float u2_d3t1[],float u2_d3t2[],float u2_d3t3[],float
    v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float
    v1_t3[],float v1_dt1[],float v1_dt2[],float v1_dt3[],float
    v1_d2t1[],float v1_d2t2[],float v1_d2t3[],float v1_d3t1[],float
    v1_d3t2[],float v1_d3t3[],float v2_x1[],float v2_x2[],float
    v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float v2_dt1[],float
    v2_dt2[],float v2_dt3[],float v2_d2t1[],float v2_d2t2[],float
    v2_d2t3[],float v2_d3t1[],float v2_d3t2[],float v2_d3t3[],float coeff[])
#define opt15 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u1_d2t1[],float
```

```
    u1_d2t2[],float u1_d2t3[],float u1_d3t1[],float u1_d3t2[],float
    u1_d3t3[],float u2_x1[],float u2_x2[],float u2_x3[],float u2_t1[],float
    u2_t2[],float u2_t3[],float u2_dt1[],float u2_dt2[],float
    u2_dt3[],float u2_d2t1[],float u2_d2t2[],float u2_d2t3[],float
    u2_d3t1[],float u2_d3t2[],float u2_d3t3[],float v1_x1[],float
    v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float v1_t3[],float
    v1_dt1[],float v1_dt2[],float v1_dt3[],float v1_d2t1[],float
    v1_d2t2[],float v1_d2t3[],float v1_d3t1[],float v1_d3t2[],float
    v1_d3t3[],float v2_x1[],float v2_x2[],float v2_x3[],float v2_t1[],float
    v2_t2[],float v2_t3[],float v2_dt1[],float v2_dt2[],float
    v2_dt3[],float v2_d2t1[],float v2_d2t2[],float v2_d2t3[],float
    v2_d3t1[],float v2_d3t2[],float v2_d3t3[],mint optimum,float coeff[])
#define opt16 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u1_d2t1[],float
    u1_d2t2[],float u1_d2t3[],float u1_d3t1[],float u1_d3t2[],float
    u1_d3t3[],float u2_x1[],float u2_x2[],float u2_x3[],float
    u2_t1[],float u2_t2[],float u2_t3[],float u2_dt1[],float
    u2_dt2[],float u2_dt3[],float u2_d2t1[],float u2_d2t2[],float
    u2_d2t3[],float u2_d3t1[],float u2_d3t2[],float u2_d3t3[],float
    v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float
    v1_t3[],float v1_dt1[],float v1_dt2[],float v1_dt3[],float
    v1_d2t1[],float v1_d2t2[],float v1_d2t3[],float v1_d3t1[],float
    v1_d3t2[],float v1_d3t3[],float v2_x1[],float v2_x2[],float
    v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float v2_dt1[],float
    v2_dt2[],float v2_dt3[],float v2_d2t1[],float v2_d2t2[],float
    v2_d2t3[],float v2_d3t1[],float v2_d3t2[],float v2_d3t3[],mint optimum)
#define opt17 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u2_x1[],float
    u2_x2[],float u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float
    u2_dt1[],float u2_dt2[],float u2_dt3[],float v1_x1[],float
    v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float v1_t3[],float
    v1_dt1[],float v1_dt2[],float v1_dt3[],float v2_x1[],float
    v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float
    v2_dt1[],float v2_dt2[],float v2_dt3[],mint optimum,float Alph_C[])
#define opt18 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u2_x1[],float u2_x2[],float u2_x3[],float u2_t1[],float u2_t2[],float
    u2_t3[],float v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float
    v1_t2[],float v1_t3[],float v2_x1[],float v2_x2[],float
    v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float Snor_C[])
#define opt19 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u2_x1[],float
    u2_x2[],float u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float
    u2_dt1[],float u2_dt2[],float u2_dt3[],float v1_x1[],float
```

```
    v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float v1_t3[],float
    v1_dt1[],float v1_dt2[],float v1_dt3[],float v2_x1[],float
    v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float
    v2_dt1[],float v2_dt2[],float v2_dt3[],mint optimum,float binormv_C[])

__device__ float inicurvature_LAC(float
    t,float Alpha,float Lambda,float InitRC,float Total_s)
{
if(Alpha==-1)
    {return ((float) 1)/InitRC-Lambda*t*Total_s;}
else if(Alpha==0)
    {return ((float) 1)/exp(Lambda*t*Total_s+log(InitRC));}
else if(Alpha==1)
    {return ((float) 1)/(InitRC+Lambda*t*Total_s);}
else if(Alpha==2)
    {return ((float)
    1)/sqrt(pow(InitRC,2)+((float) 2)*Lambda*t*Total_s);}
else
    {return pow(pow(InitRC,Alpha)+Lambda*Alpha*t*Total_s,-((float) 1)/Alpha);}
}

__device__ float inidcLACdx(float
    t,float Alpha,float Lambda,float InitRC,float Total_s)
{
if(Alpha==-1)
    {return -Total_s*Lambda;}
else if(Alpha==0)
    {return -Total_s*(Lambda*exp(-Lambda*t*Total_s))/InitRC;}
else if(Alpha==1)
    {return -Total_s*Lambda/pow(InitRC+Lambda*t*Total_s,2);}
else if(Alpha==2)
    {return -Total_s*Lambda/sqrt(pow(pow(InitRC,2)+((float)
    2)*Lambda*t*Total_s,3));}
else
    {return
    -Total_s*Lambda*pow(pow(InitRC,Alpha)+Lambda*Alpha*t*Total_s,-((float)
    1)-((float) 1)/Alpha);}
}
__device__ float inid2cLACdx2(float
    t,float Alpha,float Lambda,float InitRC,float Total_s)
{
if(Alpha==-1)
    {return 0;}
else if(Alpha==0)
    {return
    pow(Total_s,2)*(pow(Lambda,2)*exp(-Lambda*t*Total_s))/InitRC;}
else if(Alpha==1)
```

```
    {return pow(Total_s,2)*((float)
    2)*pow(Lambda,2)/pow(InitRC+Lambda*t*Total_s,3);}
else if(Alpha==2)
    {return pow(Total_s,2)*(((float)
    3)*pow(Lambda,2))/sqrt(pow(pow(InitRC,2)+((float)
    2)*Lambda*t*Total_s,5));}
else
    {return (((float)
    1)+Alpha)*pow(Total_s,2)*pow(Lambda,2)*pow(pow(InitRC,Alpha)+Lambda*
    Alpha*t*Total_s,-((float) 2)-((float) 1)/Alpha);}
}

__device__ float iniTorsion_LAC(float
    t,float Beta,float Omega,float InitRT,float Total_s)
{
if(Beta==-1)
    {return ((float) 1)/InitRT-Omega*t*Total_s;}
else if(Beta==0)
    {return ((float) 1)/exp(Omega*t*Total_s+log(InitRT));}
else if(Beta==1)
    {return ((float) 1)/(InitRT+Omega*t*Total_s);}
else if(Beta==2)
    {return ((float)
    1)/sqrt(pow(InitRT,2)+((float) 2)*Omega*t*Total_s);}
else
    {return pow(pow(InitRT,Beta)+Omega*Beta*t*Total_s,-((float) 1)/Beta);}
}

__device__ float inidtLACdx(float
    t,float Beta,float Omega,float InitRT,float Total_s)
{
if(Beta==-1)
    {return -Total_s*Omega;}
else if(Beta==0)
    {return -Total_s*(Omega*exp(-Omega*t*Total_s))/InitRT;}
else if(Beta==1)
    {return -Total_s*Omega/pow(InitRT+Omega*t*Total_s,2);}
else if(Beta==2)
    {return
    -Total_s*Omega/sqrt(pow(pow(InitRT,2)+((float) 2)*Omega*t*Total_s,3));}
else
    {return -Total_s*Omega*pow(pow(InitRT,Beta)+Omega*Beta*t*Total_s,-((float)
    1)-((float) 1)/Beta);}
}


__device__ void iniSurNorvec opt1
```

```
{
    normv_C[0] =inidxdu2*inidxdv3-inidxdu3*inidxdv2;
    normv_C[1] =inidxdu3*inidxdv1-inidxdu1*inidxdv3;
    normv_C[2] =inidxdu1*inidxdv2-inidxdu2*inidxdv1;
}
__device__ float inicoefE opt2
{
    return inidxdu1*inidxdu1+inidxdu2*inidxdu2+inidxdu3*inidxdu3;
}
__device__ float inicoefF opt2
{
    return (inidxdu1*inidxdv1+inidxdu2*inidxdv2+inidxdu3*inidxdv3);
}

__device__ float inicoefG opt2
{
    return inidxdv1*inidxdv1+inidxdv2*inidxdv2+inidxdv3*inidxdv3;
}
__device__ float inicoefL opt3
{
float normv_C[3];
repiniSurNorvec;

    return
     (normv_C[0]*inid2xdu21+normv_C[1]*inid2xdu22+normv_C[2]*inid2xdu23)/sqrt
     (normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float inicoefM opt3
{
float normv_C[3];
repiniSurNorvec;

    return
     (normv_C[0]*inid2xdudv1+normv_C[1]*inid2xdudv2+normv_C[2]*inid2xdudv3)/
     sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float inicoefN opt3
{
float normv_C[3];
repiniSurNorvec;

    return
     (normv_C[0]*inid2xdv21+normv_C[1]*inid2xdv22+normv_C[2]*inid2xdv23)/sqrt
     (normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float iniGaussCurva opt3
{
```

```
    return
    (repinicoefL*repinicoefN-pow(repinicoefM,2))/(repinicoefE*repinicoefG-
    pow(repinicoefF,2));
}
__device__ float iniMeanCurva opt3
{
    return
    0.5*(repinicoefE*repinicoefN+repinicoefG*repinicoefL-2*repinicoefF*
    repinicoefM)/(repinicoefE*repinicoefG-pow(repinicoefF,2));
}
__device__ float iniMaxPrinci opt3
{
    return repiniMeanCurva+sqrt(pow(repiniMeanCurva,2)-repiniGaussCurva);
}

__device__ float iniMinPrinci opt3
{
    return repiniMeanCurva-sqrt(pow(repiniMeanCurva,2)-repiniGaussCurva);
}
__device__ float iniPrinci opt4
{
    if(optimum==max)
    {return repiniMaxPrinci;}
    else if(optimum==min)
    {return repiniMinPrinci;}
    else
    {return 0;}
}
__device__ float iniEta opt4
{
    return
    sign1/sqrt(repinicoefE*pow(repinicoefM-repiniPrinci*repinicoefF,2)-2*
    repinicoefF*(repinicoefM-repiniPrinci*repinicoefF)*(repinicoefL-
    repiniPrinci*repinicoefE)+repinicoefG*pow(repinicoefL-repiniPrinci*
    repinicoefE,2));
}
__device__ float iniMiu opt4
{
    return
    sign1/sqrt(repinicoefE*pow(repinicoefN-repiniPrinci*repinicoefG,2)-2*
    repinicoefF*(repinicoefM-repiniPrinci*repinicoefF)*(repinicoefN-
    repiniPrinci*repinicoefG)+repinicoefG*pow(repinicoefM-repiniPrinci*
    repinicoefF,2));
}

__device__ float iniDuDs opt4
{
```

```
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-
    repiniPrinci*repinicoefG))
    {return repiniEta*(repinicoefM-repiniPrinci*repinicoefF);}
else
    {return repiniMiu*(repinicoefN-repiniPrinci*repinicoefG);}
}
__device__ float iniDvDs opt4
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
    repinicoefG))
    {return -repiniEta*(repinicoefL-repiniPrinci*repinicoefE);}
else
    {return -repiniMiu*(repinicoefM-repiniPrinci*repinicoefF);}
}

__device__ void SurNorvec opt5
{
    normv_C[0] =dxdu2*dxdv3-dxdu3*dxdv2;
    normv_C[1] =dxdu3*dxdv1-dxdu1*dxdv3;
    normv_C[2] =dxdu1*dxdv2-dxdu2*dxdv1;
}
__device__ void DSurNvDu opt10
{
    dnormv_C[0]
    =(d2xdu22*dxdv3-d2xdu23*dxdv2)+(dxdu2*d2xdudv3-dxdu3*d2xdudv2);
    dnormv_C[1]
    =(d2xdu23*dxdv1-d2xdu21*dxdv3)+(dxdu3*d2xdudv1-dxdu1*d2xdudv3);
    dnormv_C[2]
    =(d2xdu21*dxdv2-d2xdu22*dxdv1)+(dxdu1*d2xdudv2-dxdu2*d2xdudv1);

}
__device__ void DSurNvDv opt10
{
    dnormv_C[0]
    =(d2xdudv2*dxdv3-d2xdudv3*dxdv2)+(dxdu2*d2xdv23-dxdu3*d2xdv22);
    dnormv_C[1]
    =(d2xdudv3*dxdv1-d2xdudv1*dxdv3)+(dxdu3*d2xdv21-dxdu1*d2xdv23);

    dnormv_C[2]
    =(d2xdudv1*dxdv2-d2xdudv2*dxdv1)+(dxdu1*d2xdv22-dxdu2*d2xdv21);

}
__device__ void D2SurNvDu2 opt11
{
    d2normv_C[0]
    =(d3xdu32*dxdv3-d3xdu33*dxdv2)+2*(d2xdu22*d2xdudv3-d2xdu23*d2xdudv2)+(
    dxdu2*d3xdu2dv3-dxdu3*d3xdu2dv2);
```

```
    d2normv_C[1]
     =(d3xdu33*dxdv1-d3xdu31*dxdv3)+2*(d2xdu23*d2xdudv1-d2xdu21*d2xdudv3)+(
     dxdu3*d3xdu2dv1-dxdu1*d3xdu2dv3);
    d2normv_C[2]
     =(d3xdu31*dxdv2-d3xdu32*dxdv1)+2*(d2xdu21*d2xdudv2-d2xdu22*d2xdudv1)+(
     dxdu1*d3xdu2dv2-dxdu2*d3xdu2dv1);
}
__device__ void D2SurNvDuDv opt11
{
    d2normv_C[0]
     =(d3xdu2dv2*dxdv3-d3xdu2dv3*dxdv2)+(d2xdu22*d2xdv23-d2xdu23*d2xdv22)+(
     dxdu2*d3xdudv23-dxdu3*d3xdudv22);
    d2normv_C[1]
     =(d3xdu2dv3*dxdv1-d3xdu2dv1*dxdv3)+(d2xdu23*d2xdv21-d2xdu21*d2xdv23)+(
     dxdu3*d3xdudv21-dxdu1*d3xdudv23);
    d2normv_C[2]
     =(d3xdu2dv1*dxdv2-d3xdu2dv2*dxdv1)+(d2xdu21*d2xdv22-d2xdu22*d2xdv21)+(
     dxdu1*d3xdudv22-dxdu2*d3xdudv21);
}
__device__ void D2SurNvDv2 opt11
{
    d2normv_C[0]
     =(d3xdudv22*dxdv3-d3xdudv23*dxdv2)+2*(d2xdudv2*d2xdv23-d2xdudv3*d2xdv22)
     +(dxdu2*d3xdv33-dxdu3*d3xdv32);
    d2normv_C[1]
     =(d3xdudv23*dxdv1-d3xdudv21*dxdv3)+2*(d2xdudv3*d2xdv21-d2xdudv1*d2xdv23)
     +(dxdu3*d3xdv31-dxdu1*d3xdv33);
    d2normv_C[2]
     =(d3xdudv21*dxdv2-d3xdudv22*dxdv1)+2*(d2xdudv1*d2xdv22-d2xdudv2*d2xdv21)
     +(dxdu1*d3xdv32-dxdu2*d3xdv31);
}
__device__ void ScalarSurNor opt12
{
float
     ScalSNor,DScalSNorDu,DScalSNorDv,D2ScalSNorDu2,D2ScalSNorDuDv,D2ScalSNorDv2;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float d2normvdu2_C[3];
float d2normvdudv_C[3];
float d2normvdv2_C[3];
repSurNorvec;
repDSurNvDu;
repDSurNvDv;
repD2SurNvDu2;
repD2SurNvDuDv;
repD2SurNvDv2;
```

```
ScalSNor =
     sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
DScalSNorDu
     =
     (normv_C[0]*dnormvdu_C[0]+normv_C[1]*dnormvdu_C[1]+normv_C[2]*dnormvdu_C
     [2])/ScalSNor;
DScalSNorDv =
     (normv_C[0]*dnormvdv_C[0]+normv_C[1]*dnormvdv_C[1]+normv_C[2]*dnormvdv_C
     [2])/ScalSNor;
D2ScalSNorDu2 =
     ((normv_C[0]*d2normvdu2_C[0]+normv_C[1]*d2normvdu2_C[1]+normv_C[2]*
     d2normvdu2_C[2])+(dnormvdu_C[0]*dnormvdu_C[0]+dnormvdu_C[1]*dnormvdu_C[1
     ]+dnormvdu_C[2]*dnormvdu_C[2])-pow(DScalSNorDu,2))/ScalSNor;
D2ScalSNorDuDv =
     ((normv_C[0]*d2normvdudv_C[0]+normv_C[1]*d2normvdudv_C[1]+normv_C[2]*
     d2normvdudv_C[2])+(dnormvdu_C[0]*dnormvdv_C[0]+dnormvdu_C[1]*dnormvdv_C[
     1]+dnormvdu_C[2]*dnormvdv_C[2])-DScalSNorDu*DScalSNorDv)/ScalSNor;
D2ScalSNorDv2 =
     ((normv_C[0]*d2normvdv2_C[0]+normv_C[1]*d2normvdv2_C[1]+normv_C[2]*
     d2normvdv2_C[2])+(dnormvdv_C[0]*dnormvdv_C[0]+dnormvdv_C[1]*dnormvdv_C[1
     ]+dnormvdv_C[2]*dnormvdv_C[2])-pow(DScalSNorDv,2))/ScalSNor;

Scal[0] = ScalSNor;
Scal[1] = DScalSNorDu;
Scal[2] = DScalSNorDv;
Scal[3] = D2ScalSNorDu2;
Scal[4] = D2ScalSNorDuDv;
Scal[5] = D2ScalSNorDv2;
}

__device__ float coefE opt6
{
    return dxdu1*dxdu1+dxdu2*dxdu2+dxdu3*dxdu3;
}
__device__ float DEDu opt7
{
    return 2*(dxdu1*d2xdu21+dxdu2*d2xdu22+dxdu3*d2xdu23);
}
__device__ float DEDv opt7
{
    return 2*(dxdu1*d2xdudv1+dxdu2*d2xdudv2+dxdu3*d2xdudv3);
}
__device__ float D2EDu2 opt13
{
    return
     2*((d2xdu21*d2xdu21+d2xdu22*d2xdu22+d2xdu23*d2xdu23)+(dxdu1*d3xdu31+
```

```
    dxdu2*d3xdu32+dxdu3*d3xdu33));
}
__device__ float D2EDuDv opt13
{
    return
    2*((d2xdudv1*d2xdu21+d2xdudv2*d2xdu22+d2xdudv3*d2xdu23)+(dxdu1*d3xdu2dv1
    +dxdu2*d3xdu2dv2+dxdu3*d3xdu2dv3));
}
__device__ float D2EDv2 opt13
{
    return
    2*((d2xdudv1*d2xdudv1+d2xdudv2*d2xdudv2+d2xdudv3*d2xdudv3)+(dxdu1*
    d3xdudv21+dxdu2*d3xdudv22+dxdu3*d3xdudv23));
}

__device__ float coefF opt6
{
    return (dxdu1*dxdv1+dxdu2*dxdv2+dxdu3*dxdv3);
}
__device__ float DFDu opt7
{
    return dxdv1*d2xdu21+dxdv2*d2xdu22+dxdv3*d2xdu23
     + dxdu1*d2xdudv1+dxdu2*d2xdudv2+dxdu3*d2xdudv3;
}
__device__ float DFDv opt7
{
    return dxdv1*d2xdudv1+dxdv2*d2xdudv2+dxdv3*d2xdudv3
     + dxdu1*d2xdv21+dxdu2*d2xdv22+dxdu3*d2xdv23;
}
__device__ float D2FDu2 opt13
{
    return
    (dxdv1*d3xdu31+dxdv2*d3xdu32+dxdv3*d3xdu33)+2*(d2xdu21*d2xdudv1+d2xdu22*
    d2xdudv2+d2xdu23*d2xdudv3)+(dxdu1*d3xdu2dv1+dxdu2*d3xdu2dv2+dxdu3*
    d3xdu2dv3);
}
__device__ float D2FDuDv opt13
{
    return
    (dxdv1*d3xdu2dv1+dxdv2*d3xdu2dv2+dxdv3*d3xdu2dv3)+(d2xdu21*d2xdv21+
    d2xdu22*d2xdv22+d2xdu23*d2xdv23)+(d2xdudv1*d2xdudv1+d2xdudv2*d2xdudv2+
    d2xdudv3*d2xdudv3)+(dxdu1*d3xdudv21+dxdu2*d3xdudv22+dxdu3*d3xdudv23);
}
__device__ float D2FDv2 opt13
{
    return
    (dxdv1*d3xdudv21+dxdv2*d3xdudv22+dxdv3*d3xdudv23)+2*(d2xdudv1*d2xdv21+
```

```
    d2xdudv2*d2xdv22+d2xdudv3*d2xdv23)+(dxdu1*d3xdv31+dxdu2*d3xdv32+dxdu3*
    d3xdv33);
}
__device__ float coefG opt6
{
    return dxdv1*dxdv1+dxdv2*dxdv2+dxdv3*dxdv3;
}
__device__ float DGDu opt7
{
    return 2*(dxdv1*d2xdudv1+dxdv2*d2xdudv2+dxdv3*d2xdudv3);
}
__device__ float DGDv opt7
{
    return 2*(dxdv1*d2xdv21+dxdv2*d2xdv22+dxdv3*d2xdv23);
}
__device__ float D2GDu2 opt13
{
    return
    2*((d2xdudv1*d2xdudv1+d2xdudv2*d2xdudv2+d2xdudv3*d2xdudv3)+(dxdv1*
    d3xdu2dv1+dxdv2*d3xdu2dv2+dxdv3*d3xdu2dv3));
}
__device__ float D2GDuDv opt13
{
    return
    2*((d2xdv21*d2xdudv1+d2xdv22*d2xdudv2+d2xdv23*d2xdudv3)+(dxdv1*d3xdudv21
    +dxdv2*d3xdudv22+dxdv3*d3xdudv23));
}
__device__ float D2GDv2 opt13
{
    return
    2*((d2xdv21*d2xdv21+d2xdv22*d2xdv22+d2xdv23*d2xdv23)+(dxdv1*d3xdv31+
    dxdv2*d3xdv32+dxdv3*d3xdv33));
}
__device__ float coefL opt7
{
float normv_C[3];
repSurNorvec;

    return
    (normv_C[0]*d2xdu21+normv_C[1]*d2xdu22+normv_C[2]*d2xdu23)/sqrt(normv_C[
    0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float coefM opt7
{
float normv_C[3];
repSurNorvec;
```

```
    return
    (normv_C[0]*d2xdudv1+normv_C[1]*d2xdudv2+normv_C[2]*d2xdudv3)/sqrt(
    normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float coefN opt7
{
float normv_C[3];
repSurNorvec;

    return
    (normv_C[0]*d2xdv21+normv_C[1]*d2xdv22+normv_C[2]*d2xdv23)/sqrt(normv_C[
    0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float GaussCurva opt7
{
    return
    (repcoefL*repcoefN-pow(repcoefM,2))/(repcoefE*repcoefG-pow(repcoefF,2));

}
__device__ float MeanCurva opt7
{
    return
    0.5*(repcoefE*repcoefN+repcoefG*repcoefL-2*repcoefF*repcoefM)/(repcoefE*
    repcoefG-pow(repcoefF,2));
}
__device__ float MaxPrinci opt7
{
    return repMeanCurva+sqrt(pow(repMeanCurva,2)-repGaussCurva);
}

__device__ float MinPrinci opt7
{
    return repMeanCurva-sqrt(pow(repMeanCurva,2)-repGaussCurva);
}
__device__ float Princi opt8
{
    if(optimum==max)
    {return repMaxPrinci;}
    else if(optimum==min)
    {return repMinPrinci;}
    else
    {return 0;}
}
__device__ void DcoefL_2 opt14
{
float CoeL,DCoeLDu,DCoeLDv,D2CoeLDu2,D2CoeLDuDv,D2CoeLDv2;
float normv_C[3];
```

```
float dnormvdu_C[3];
float dnormvdv_C[3];
float d2normvdu2_C[3];
float d2normvdudv_C[3];
float d2normvdv2_C[3];
float Scal[6];
repSurNorvec;
repDSurNvDu;
repDSurNvDv;
repD2SurNvDu2;
repD2SurNvDuDv;
repD2SurNvDv2;
repScalarSurNor;

CoeL = (normv_C[0]*d2xdu21+normv_C[1]*d2xdu22+normv_C[2]*d2xdu23)/Scal[0];
DCoeLDu =
    ((dnormvdu_C[0]*d2xdu21+dnormvdu_C[1]*d2xdu22+dnormvdu_C[2]*d2xdu23)+(
    normv_C[0]*d3xdu31+normv_C[1]*d3xdu32+normv_C[2]*d3xdu33)-Scal[1]*CoeL)/
    Scal[0];
DCoeLDv =
    ((dnormvdv_C[0]*d2xdu21+dnormvdv_C[1]*d2xdu22+dnormvdv_C[2]*d2xdu23)+(
    normv_C[0]*d3xdu2dv1+normv_C[1]*d3xdu2dv2+normv_C[2]*d3xdu2dv3)-Scal[2]*
    CoeL)/Scal[0];
D2CoeLDu2 =
    ((d2normvdu2_C[0]*d2xdu21+d2normvdu2_C[1]*d2xdu22+d2normvdu2_C[2]*
    d2xdu23)+2*(dnormvdu_C[0]*d3xdu31+dnormvdu_C[1]*d3xdu32+dnormvdu_C[2]*
    d3xdu33)+(normv_C[0]*d4xdu41+normv_C[1]*d4xdu42+normv_C[2]*d4xdu43)-Scal
    [3]*CoeL-2*Scal[1]*DCoeLDu)/Scal[0];
D2CoeLDuDv =
    ((d2normvdudv_C[0]*d2xdu21+d2normvdudv_C[1]*d2xdu22+d2normvdudv_C[2]*
    d2xdu23)+(dnormvdu_C[0]*d3xdu2dv1+dnormvdu_C[1]*d3xdu2dv2+dnormvdu_C[2]*
    d3xdu2dv3)+(dnormvdv_C[0]*d3xdu31+dnormvdv_C[1]*d3xdu32+dnormvdv_C[2]*
    d3xdu33)+(normv_C[0]*d4xdu3dv1+normv_C[1]*d4xdu3dv2+normv_C[2]*d4xdu3dv3
    )-Scal[4]*CoeL-Scal[1]*DCoeLDv-Scal[2]*DCoeLDu)/Scal[0];
D2CoeLDv2 =
    ((d2normvdv2_C[0]*d2xdu21+d2normvdv2_C[1]*d2xdu22+d2normvdv2_C[2]*
    d2xdu23)+2*(dnormvdv_C[0]*d3xdu2dv1+dnormvdv_C[1]*d3xdu2dv2+dnormvdv_C[2
    ]*d3xdu2dv3)+(normv_C[0]*d4xdu2dv21+normv_C[1]*d4xdu2dv22+normv_C[2]*
    d4xdu2dv23)-Scal[5]*CoeL-2*Scal[2]*DCoeLDv)/Scal[0];

coeff[0] = CoeL;
coeff[1] = DCoeLDu;
coeff[2] = DCoeLDv;
coeff[3] = D2CoeLDu2;
coeff[4] = D2CoeLDuDv;
coeff[5] = D2CoeLDv2;
}
```

```
__device__ void DcoefM_2 opt14
{
float CoeM,DCoeMDu,DCoeMDv,D2CoeMDu2,D2CoeMDuDv,D2CoeMDv2;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float d2normvdu2_C[3];
float d2normvdudv_C[3];
float d2normvdv2_C[3];
float Scal[6];
repSurNorvec;
repDSurNvDu;
repDSurNvDv;
repD2SurNvDu2;
repD2SurNvDuDv;
repD2SurNvDv2;
repScalarSurNor;

CoeM = (normv_C[0]*d2xdudv1+normv_C[1]*d2xdudv2+normv_C[2]*d2xdudv3)/Scal[0];
DCoeMDu =
    ((dnormvdu_C[0]*d2xdudv1+dnormvdu_C[1]*d2xdudv2+dnormvdu_C[2]*d2xdudv3)+
    (normv_C[0]*d3xdu2dv1+normv_C[1]*d3xdu2dv2+normv_C[2]*d3xdu2dv3)-Scal[1]
    *CoeM)/Scal[0];
DCoeMDv =
    ((dnormvdv_C[0]*d2xdudv1+dnormvdv_C[1]*d2xdudv2+dnormvdv_C[2]*d2xdudv3)+
    (normv_C[0]*d3xdudv21+normv_C[1]*d3xdudv22+normv_C[2]*d3xdudv23)-Scal[2]
    *CoeM)/Scal[0];
D2CoeMDu2 =
    ((d2normvdu2_C[0]*d2xdudv1+d2normvdu2_C[1]*d2xdudv2+d2normvdu2_C[2]*
    d2xdudv3)+2*(dnormvdu_C[0]*d3xdu2dv1+dnormvdu_C[1]*d3xdu2dv2+dnormvdu_C[
    2]*d3xdu2dv3)+(normv_C[0]*d4xdu3dv1+normv_C[1]*d4xdu3dv2+normv_C[2]*
    d4xdu3dv3)-Scal[3]*CoeM-2*Scal[1]*DCoeMDu)/Scal[0];
D2CoeMDuDv =
    ((d2normvdudv_C[0]*d2xdudv1+d2normvdudv_C[1]*d2xdudv2+d2normvdudv_C[2]*
    d2xdudv3)+(dnormvdu_C[0]*d3xdudv21+dnormvdu_C[1]*d3xdudv22+dnormvdu_C[2]
    *d3xdudv23)+(dnormvdv_C[0]*d3xdu2dv1+dnormvdv_C[1]*d3xdu2dv2+dnormvdv_C[
    2]*d3xdu2dv3)+(normv_C[0]*d4xdu2dv21+normv_C[1]*d4xdu2dv22+normv_C[2]*
    d4xdu2dv23)-Scal[4]*CoeM-Scal[1]*DCoeMDv-Scal[2]*DCoeMDu)/Scal[0];
D2CoeMDv2 =
    ((d2normvdv2_C[0]*d2xdudv1+d2normvdv2_C[1]*d2xdudv2+d2normvdv2_C[2]*
    d2xdudv3)+2*(dnormvdv_C[0]*d3xdudv21+dnormvdv_C[1]*d3xdudv22+dnormvdv_C[
    2]*d3xdudv23)+(normv_C[0]*d4xdudv31+normv_C[1]*d4xdudv32+normv_C[2]*
    d4xdudv33)-Scal[5]*CoeM-2*Scal[2]*DCoeMDv)/Scal[0];

coeff[0] = CoeM;
coeff[1] = DCoeMDu;
coeff[2] = DCoeMDv;
```

```
coeff[3] = D2CoeMDu2;
coeff[4] = D2CoeMDuDv;
coeff[5] = D2CoeMDv2;
}
__device__ void DcoefN_2 opt14
{
float CoeN,DCoeNDu,DCoeNDv,D2CoeNDu2,D2CoeNDuDv,D2CoeNDv2;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float d2normvdu2_C[3];
float d2normvdudv_C[3];
float d2normvdv2_C[3];
float Scal[6];
repSurNorvec;
repDSurNvDu;
repDSurNvDv;
repD2SurNvDu2;
repD2SurNvDuDv;
repD2SurNvDv2;
repScalarSurNor;

CoeN = (normv_C[0]*d2xdv21+normv_C[1]*d2xdv22+normv_C[2]*d2xdv23)/Scal[0];
DCoeNDu =
    ((dnormvdu_C[0]*d2xdv21+dnormvdu_C[1]*d2xdv22+dnormvdu_C[2]*d2xdv23)+(
    normv_C[0]*d3xdudv21+normv_C[1]*d3xdudv22+normv_C[2]*d3xdudv23)-Scal[1]*
    CoeN)/Scal[0];
DCoeNDv =
    ((dnormvdv_C[0]*d2xdv21+dnormvdv_C[1]*d2xdv22+dnormvdv_C[2]*d2xdv23)+(
    normv_C[0]*d3xdv31+normv_C[1]*d3xdv32+normv_C[2]*d3xdv33)-Scal[2]*CoeN)/
    Scal[0];
D2CoeNDu2 =
    ((d2normvdu2_C[0]*d2xdv21+d2normvdu2_C[1]*d2xdv22+d2normvdu2_C[2]*
    d2xdv23)+2*(dnormvdu_C[0]*d3xdudv21+dnormvdu_C[1]*d3xdudv22+dnormvdu_C[2
    ]*d3xdudv23)+(normv_C[0]*d4xdu2dv21+normv_C[1]*d4xdu2dv22+normv_C[2]*
    d4xdu2dv23)-Scal[3]*CoeN-2*Scal[1]*DCoeNDu)/Scal[0];
D2CoeNDuDv =
    ((d2normvdudv_C[0]*d2xdv21+d2normvdudv_C[1]*d2xdv22+d2normvdudv_C[2]*
    d2xdv23)+(dnormvdu_C[0]*d3xdv31+dnormvdu_C[1]*d3xdv32+dnormvdu_C[2]*
    d3xdv33)+(dnormvdv_C[0]*d3xdudv21+dnormvdv_C[1]*d3xdudv22+dnormvdv_C[2]*
    d3xdudv23)+(normv_C[0]*d4xdudv31+normv_C[1]*d4xdudv32+normv_C[2]*
    d4xdudv33)-Scal[4]*CoeN-Scal[1]*DCoeNDv-Scal[2]*DCoeNDu)/Scal[0];
D2CoeNDv2 =
    ((d2normvdv2_C[0]*d2xdv21+d2normvdv2_C[1]*d2xdv22+d2normvdv2_C[2]*
    d2xdv23)+2*(dnormvdv_C[0]*d3xdv31+dnormvdv_C[1]*d3xdv32+dnormvdv_C[2]*
    d3xdv33)+(normv_C[0]*d4xdv41+normv_C[1]*d4xdv42+normv_C[2]*d4xdv43)-Scal
    [5]*CoeN-2*Scal[2]*DCoeNDv)/Scal[0];
```

```
coeff[0] = CoeN;
coeff[1] = DCoeNDu;
coeff[2] = DCoeNDv;
coeff[3] = D2CoeNDu2;
coeff[4] = D2CoeNDuDv;
coeff[5] = D2CoeNDv2;
}
__device__ void DGaussCurva_2 opt14
{
float GCurv,DGCurvDu,DGCurvDv,D2GCurvDu2,D2GCurvDuDv,D2GCurvDv2;
float CoeffL[6];
float CoeffM[6];
float CoeffN[6];
repDcoefL_2;
repDcoefM_2;
repDcoefN_2;

GCurv=
    (CoeffL[0]*CoeffN[0]-pow(CoeffM[0],2))/(repcoefE*repcoefG-pow(repcoefF,2));
DGCurvDu
    =
    ((pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repcoefG*repDEDu-2*repcoefF*
    repDFDu+repcoefE*repDGDu)+(-pow(repcoefF,2)+repcoefE*repcoefG)*(CoeffN[0
    ]*CoeffL[1]-2*CoeffM[0]*CoeffM[1]+CoeffL[0]*CoeffN[1]))/(pow(pow(
    repcoefF,2)-repcoefE*repcoefG,2));
DGCurvDv
    =((pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repcoefG*repDEDv-2*repcoefF*
    repDFDv+repcoefE*repDGDv)+(-pow(repcoefF,2)+repcoefE*repcoefG)*(CoeffN[0
    ]*CoeffL[2]-2*CoeffM[0]*CoeffM[2]+CoeffL[0]*CoeffN[2]))/(pow(pow(
    repcoefF,2)-repcoefE*repcoefG,2));
D2GCurvDu2 =
    ((-2*(pow(repcoefF,2)-repcoefE*repcoefG)*(repcoefG*repDEDu-2*repcoefF*
    repDFDu+repcoefE*repDGDu)*(CoeffN[0]*CoeffL[1]-2*CoeffM[0]*CoeffM[1]+
    CoeffL[0]*CoeffN[1])+(pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(2*pow(
    repcoefG*repDEDu-2*repcoefF*repDFDu+repcoefE*repDGDu,2)-(pow(repcoefF,2)
    -repcoefE*repcoefG)*(2*pow(repDFDu,2)-2*repDEDu*repDGDu-repcoefG*
    repD2EDu2+2*repcoefF*repD2FDu2-repcoefE*repD2GDu2))+pow(pow(repcoefF,2)-
    repcoefE*repcoefG,2)*(2*pow(CoeffM[1],2)-2*CoeffL[1]*CoeffN[1]-CoeffN[0]
    *CoeffL[3]+2*CoeffM[0]*CoeffM[3]-CoeffL[0]*CoeffN[3]))/pow(pow(repcoefF,
    2)-repcoefE*repcoefG,3));
D2GCurvDuDv =
    -((2*(pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repcoefG*repDEDv-2*repcoefF
    *repDFDv+repcoefE*repDGDv)*(repcoefG*repDEDu-2*repcoefF*repDFDu+repcoefE
    *repDGDu)+(-pow(repcoefF,2)+repcoefE*repcoefG)*(CoeffN[0]*CoeffL[2]-2*
    CoeffM[0]*CoeffM[2]+CoeffL[0]*CoeffN[2])*(repcoefG*repDEDu-2*repcoefF*
    repDFDu+repcoefE*repDGDu)+(-pow(repcoefF,2)+repcoefE*repcoefG)*(repcoefG
```

```
     *repDEDv-2*repcoefF*repDFDv+repcoefE*repDGDv)*(CoeffN[0]*CoeffL[1]-2*
     CoeffM[0]*CoeffM[1]+CoeffL[0]*CoeffN[1])+(-pow(repcoefF,2)+repcoefE*
     repcoefG)*(-pow(CoeffM[0],2)+CoeffL[0]*CoeffN[0])*(repDGDv*repDEDu-2*
     repDFDv*repDFDu+repDEDv*repDGDu+repcoefG*repD2EDuDv-2*repcoefF*
     repD2FDuDv+repcoefE*repD2GDuDv)-pow(pow(repcoefF,2)-repcoefE*repcoefG,2)
     *(CoeffN[2]*CoeffL[1]-2*CoeffM[2]*CoeffM[1]+CoeffL[2]*CoeffN[1]+CoeffN[0
     ]*CoeffL[4]-2*CoeffM[0]*CoeffM[4]+CoeffL[0]*CoeffN[4]))/pow(-pow(
     repcoefF,2)+repcoefE*repcoefG,3));
D2GCurvDv2 =
     ((-2*(pow(repcoefF,2)-repcoefE*repcoefG)*(repcoefG*repDEDv-2*repcoefF*
     repDFDv+repcoefE*repDGDv)*(CoeffN[0]*CoeffL[2]-2*CoeffM[0]*CoeffM[2]+
     CoeffL[0]*CoeffN[2])+(pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(2*pow(
     repcoefG*repDEDv-2*repcoefF*repDFDv+repcoefE*repDGDv,2)-(pow(repcoefF,2)
     -repcoefE*repcoefG)*(2*pow(repDFDv,2)-2*repDEDv*repDGDv-repcoefG*
     repD2EDv2+2*repcoefF*repD2FDv2-repcoefE*repD2GDv2))+pow(pow(repcoefF,2)-
     repcoefE*repcoefG,2)*(2*pow(CoeffM[2],2)-2*CoeffL[2]*CoeffN[2]-CoeffN[0]
     *CoeffL[5]+2*CoeffM[0]*CoeffM[5]-CoeffL[0]*CoeffN[5]))/pow(pow(repcoefF,
     2)-repcoefE*repcoefG,3));

coeff[0] = GCurv;
coeff[1] = DGCurvDu;
coeff[2] = DGCurvDv;
coeff[3] = D2GCurvDu2;
coeff[4] = D2GCurvDuDv;
coeff[5] = D2GCurvDv2;
}
__device__ void DMeanCurva_2 opt14
{
float MCurv,DMCurvDu,DMCurvDv,D2MCurvDu2,D2MCurvDuDv,D2MCurvDv2;
float CoeffL[6];
float CoeffM[6];
float CoeffN[6];
repDcoefL_2;
repDcoefM_2;
repDcoefN_2;

MCurv =
     0.5*(repcoefE*CoeffN[0]+repcoefG*CoeffL[0]-2*repcoefF*CoeffM[0])/(
     repcoefE*repcoefG-pow(repcoefF,2));
DMCurvDu =
     0.5*(-(repcoefG*CoeffL[0]-2*repcoefF*CoeffM[0]+repcoefE*CoeffN[0])*(
     repcoefG*repDEDu-2*repcoefF*repDFDu+repcoefE*repDGDu)+(-pow(repcoefF,2)+
     repcoefE*repcoefG)*(CoeffN[0]*repDEDu-2*CoeffM[0]*repDFDu+CoeffL[0]*
     repDGDu+repcoefG*CoeffL[1]-2*repcoefF*CoeffM[1]+repcoefE*CoeffN[1]))/(
     pow(pow(repcoefF,2)-repcoefE*repcoefG,2));
DMCurvDv =
     0.5*(-(repcoefG*CoeffL[0]-2*repcoefF*CoeffM[0]+repcoefE*CoeffN[0])*(
```

```
    repcoefG*repDEDv-2*repcoefF*repDFDv+repcoefE*repDGDv)+(-pow(repcoefF,2)+
    repcoefE*repcoefG)*(CoeffN[0]*repDEDv-2*CoeffM[0]*repDFDv+CoeffL[0]*
    repDGDv+repcoefG*CoeffL[2]-2*repcoefF*CoeffM[2]+repcoefE*CoeffN[2]))/(
    pow(pow(repcoefF,2)-repcoefE*repcoefG,2));
D2MCurvDu2 =
    -((2*(pow(repcoefF,2)-repcoefE*repcoefG)*(repcoefG*repDEDu-2*repcoefF*
    repDFDu+repcoefE*repDGDu)*(CoeffN[0]*repDEDu-2*CoeffM[0]*repDFDu+CoeffL[
    0]*repDGDu+repcoefG*CoeffL[1]-2*repcoefF*CoeffM[1]+repcoefE*CoeffN[1])+(
    repcoefG*CoeffL[0]-2*repcoefF*CoeffM[0]+repcoefE*CoeffN[0])*(2*pow(
    repcoefG*repDEDu-2*repcoefF*repDFDu+repcoefE*repDGDu,2)-(pow(repcoefF,2)
    -repcoefE*repcoefG)*(2*pow(repDFDu,2)-2*repDEDu*repDGDu-repcoefG*
    repD2EDu2+2*repcoefF*repD2FDu2-repcoefE*repD2GDu2))+pow(pow(repcoefF,2)-
    repcoefE*repcoefG,2)*(2*repDGDu*CoeffL[1]-4*repDFDu*CoeffM[1]+2*repDEDu*
    CoeffN[1]+CoeffN[0]*repD2EDu2-2*CoeffM[0]*repD2FDu2+CoeffL[0]*repD2GDu2+
    repcoefG*CoeffL[3]-2*repcoefF*CoeffM[3]+repcoefE*CoeffN[3]))/(2*pow(pow(
    repcoefF,2)-repcoefE*repcoefG,3)));
D2MCurvDuDv =
    -(0.5*(-2*(repcoefG*CoeffL[0]-2*repcoefF*CoeffM[0]+repcoefE*CoeffN[0])*(
    repcoefG*repDEDv-2*repcoefF*repDFDv+repcoefE*repDGDv)*(repcoefG*repDEDu-
    2*repcoefF*repDFDu+repcoefE*repDGDu)+(-pow(repcoefF,2)+repcoefE*repcoefG
    )*(CoeffN[0]*repDEDv-2*CoeffM[0]*repDFDv+CoeffL[0]*repDGDv+repcoefG*
    CoeffL[2]-2*repcoefF*CoeffM[2]+repcoefE*CoeffN[2])*(repcoefG*repDEDu-2*
    repcoefF*repDFDu+repcoefE*repDGDu)+(-pow(repcoefF,2)+repcoefE*repcoefG)*
    (repcoefG*repDEDv-2*repcoefF*repDFDv+repcoefE*repDGDv)*(CoeffN[0]*
    repDEDu-2*CoeffM[0]*repDFDu+CoeffL[0]*repDGDu+repcoefG*CoeffL[1]-2*
    repcoefF*CoeffM[1]+repcoefE*CoeffN[1])+(-pow(repcoefF,2)+repcoefE*
    repcoefG)*(repcoefG*CoeffL[0]-2*repcoefF*CoeffM[0]+repcoefE*CoeffN[0])*(
    repDGDv*repDEDu-2*repDFDv*repDFDu+repDEDv*repDGDu+repcoefG*repD2EDuDv-2*
    repcoefF*repD2FDuDv+repcoefE*repD2GDuDv)-pow(pow(repcoefF,2)-repcoefE*
    repcoefG,2)*(CoeffN[2]*repDEDu-2*CoeffM[2]*repDFDu+CoeffL[2]*repDGDu+
    repDGDv*CoeffL[1]-2*repDFDv*CoeffM[1]+repDEDv*CoeffN[1]+CoeffN[0]*
    repD2EDuDv-2*CoeffM[0]*repD2FDuDv+CoeffL[0]*repD2GDuDv+repcoefG*CoeffL[4
    ]-2*repcoefF*CoeffM[4]+repcoefE*CoeffN[4]))/(pow(-pow(repcoefF,2)+
    repcoefE*repcoefG,3)));
D2MCurvDv2 =
    -(0.5*(2*(pow(repcoefF,2)-repcoefE*repcoefG)*(repcoefG*repDEDv-2*
    repcoefF*repDFDv+repcoefE*repDGDv)*(CoeffN[0]*repDEDv-2*CoeffM[0]*
    repDFDv+CoeffL[0]*repDGDv+repcoefG*CoeffL[2]-2*repcoefF*CoeffM[2]+
    repcoefE*CoeffN[2])+(repcoefG*CoeffL[0]-2*repcoefF*CoeffM[0]+repcoefE*
    CoeffN[0])*(2*pow(repcoefG*repDEDv-2*repcoefF*repDFDv+repcoefE*repDGDv,2
    )-(pow(repcoefF,2)-repcoefE*repcoefG)*(2*pow(repDFDv,2)-2*repDEDv*
    repDGDv-repcoefG*repD2EDv2+2*repcoefF*repD2FDv2-repcoefE*repD2GDv2))+pow
    (pow(repcoefF,2)-repcoefE*repcoefG,2)*(2*repDGDv*CoeffL[2]-4*repDFDv*
    CoeffM[2]+2*repDEDv*CoeffN[2]+CoeffN[0]*repD2EDv2-2*CoeffM[0]*repD2FDv2+
    CoeffL[0]*repD2GDv2+repcoefG*CoeffL[5]-2*repcoefF*CoeffM[5]+repcoefE*
    CoeffN[5]))/(pow(pow(repcoefF,2)-repcoefE*repcoefG,3)));
```

```
coeff[0] = MCurv;
coeff[1] = DMCurvDu;
coeff[2] = DMCurvDv;
coeff[3] = D2MCurvDu2;
coeff[4] = D2MCurvDuDv;
coeff[5] = D2MCurvDv2;
}
__device__ void DPrinci_2 opt15
{
float Princip,DPrincipDu,DPrincipDv,D2PrincipDu2,D2PrincipDuDv,D2PrincipDv2;
float GCurva[6];
float MCurva[6];
repDGaussCurva_2;
repDMeanCurva_2;

if(optimum==max)
{Princip = MCurva[0]+sqrt(pow(MCurva[0],2)-GCurva[0]);
DPrincipDu =
    MCurva[1]+(2*MCurva[0]*MCurva[1]-GCurva[1])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
DPrincipDv =
    MCurva[2]+(2*MCurva[0]*MCurva[2]-GCurva[2])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
D2PrincipDu2 =
    -(pow(-2*MCurva[0]*MCurva[1]+GCurva[1],2))/(4*sqrt(pow(pow(MCurva[0],2)-
    GCurva[0],3)))+MCurva[3]+(2*pow(MCurva[1],2)+2*MCurva[0]*MCurva[3]-
    GCurva[3])/(2*sqrt(pow(MCurva[0],2)-GCurva[0]));
D2PrincipDuDv =
    -(((2*MCurva[0]*MCurva[2]-GCurva[2])*(2*MCurva[0]*MCurva[1]-GCurva[1]))/
    (4*sqrt(pow(pow(MCurva[0],2)-GCurva[0],3))))+MCurva[4]+(2*MCurva[2]*
    MCurva[1]+2*MCurva[0]*MCurva[4]-GCurva[4])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
D2PrincipDv2 =
    -(pow(-2*MCurva[0]*MCurva[2]+GCurva[2],2))/(4*sqrt(pow(pow(MCurva[0],2)-
    GCurva[0],3)))+MCurva[5]+(2*pow(MCurva[2],2)+2*MCurva[0]*MCurva[5]-
    GCurva[5])/(2*sqrt(pow(MCurva[0],2)-GCurva[0]));
}
else if(optimum==min)
{Princip = MCurva[0]-sqrt(pow(MCurva[0],2)-GCurva[0]);
DPrincipDu =
    MCurva[1]-(2*MCurva[0]*MCurva[1]-GCurva[1])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
DPrincipDv =
    MCurva[2]-(2*MCurva[0]*MCurva[2]-GCurva[2])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
D2PrincipDu2 =
```

```
    (pow(-2*MCurva[0]*MCurva[1]+GCurva[1],2))/(4*sqrt(pow(pow(MCurva[0],2)-
    GCurva[0],3)))+MCurva[3]+(-2*(pow(MCurva[1],2)+MCurva[0]*MCurva[3])+
    GCurva[3])/(2*sqrt(pow(MCurva[0],2)-GCurva[0]));
D2PrincipDuDv =
    ((2*MCurva[0]*MCurva[2]-GCurva[2])*(2*MCurva[0]*MCurva[1]-GCurva[1]))/(4
    *sqrt(pow(pow(MCurva[0],2)-GCurva[0],3)))+MCurva[4]+(-2*MCurva[2]*MCurva
    [1]-2*MCurva[0]*MCurva[4]+GCurva[4])/(2*sqrt(pow(MCurva[0],2)-GCurva[0])
    );
D2PrincipDv2
    =(pow(-2*MCurva[0]*MCurva[2]+GCurva[2],2))/(4*sqrt(pow(pow(MCurva[0],2)-
    GCurva[0],3)))+MCurva[5]+(-2*(pow(MCurva[2],2)+MCurva[0]*MCurva[5])+
    GCurva[5])/(2*sqrt(pow(MCurva[0],2)-GCurva[0]));
}
else
{Princip = 0;
DPrincipDu = 0;
DPrincipDv = 0;
D2PrincipDu2 = 0;
D2PrincipDuDv = 0;
D2PrincipDv2 = 0;
}

coeff[0] = Princip;
coeff[1] = DPrincipDu;
coeff[2] = DPrincipDv;
coeff[3] = D2PrincipDu2;
coeff[4] = D2PrincipDuDv;
coeff[5] = D2PrincipDv2;
}

__device__ float Eta opt8
{
    return
    sign1/sqrt(repcoefE*pow(repcoefM-repPrinci*repcoefF,2)-2*repcoefF*(
    repcoefM-repPrinci*repcoefF)*(repcoefL-repPrinci*repcoefE)+repcoefG*pow(
    repcoefL-repPrinci*repcoefE,2));
}
__device__ float Miu opt8
{
    return
    sign1/sqrt(repcoefE*pow(repcoefN-repPrinci*repcoefG,2)-2*repcoefF*(
    repcoefM-repPrinci*repcoefF)*(repcoefN-repPrinci*repcoefG)+repcoefG*pow(
    repcoefM-repPrinci*repcoefF,2));
}
__device__ float DuDs opt8
{
if(abs(repcoefL-repPrinci*repcoefE)>=abs(repcoefN-repPrinci*repcoefG))
```

```
    {return repEta*(repcoefM-repPrinci*repcoefF);}
else
    {return repMiu*(repcoefN-repPrinci*repcoefG);}
}
__device__ float DvDs opt8
{
if(abs(repcoefL-repPrinci*repcoefE)>=abs(repcoefN-repPrinci*repcoefG))
    {return -repEta*(repcoefL-repPrinci*repcoefE);}
else
    {return -repMiu*(repcoefM-repPrinci*repcoefF);}
}
__device__ float DEDs opt8
{
    return repDEDu*repDuDs+repDEDv*repDvDs;
}
__device__ float DFDs opt8
{
    return repDFDu*repDuDs+repDFDv*repDvDs;
}
__device__ float DGDs opt8
{
    return repDGDu*repDuDs+repDGDv*repDvDs;
}
__device__ float DLDs opt16
{
float CoeffL[6];
repDcoefL_2;

    return CoeffL[1]*repDuDs+CoeffL[2]*repDvDs;
}
__device__ float DMDs opt16
{
float CoeffM[6];
repDcoefM_2;

    return CoeffM[1]*repDuDs+CoeffM[2]*repDvDs;
}
__device__ float DNDs opt16
{
float CoeffN[6];
repDcoefN_2;

    return CoeffN[1]*repDuDs+CoeffN[2]*repDvDs;
}
__device__ float DPrinciDs opt16
{
float PCurva[6];
```

```
repDPrinci_2;

    return PCurva[1]*repDuDs+PCurva[2]*repDvDs;
}
__device__ float DuDt opt8
{
if(abs(repcoefL-repPrinci*repcoefE)>=abs(repcoefN-repPrinci*repcoefG))
    {return (repcoefM-repPrinci*repcoefF);}
else
    {return (repcoefN-repPrinci*repcoefG);}
}
__device__ float DvDt opt8
{
if(abs(repcoefL-repPrinci*repcoefE)>=abs(repcoefN-repPrinci*repcoefG))
    {return (repcoefL-repPrinci*repcoefE);}
else
    {return (repcoefM-repPrinci*repcoefF);}
}
__device__ void DSurDs opt9
{
    tangv_C[0] =dxdu1*repDuDs+
                dxdv1*repDvDs;
    tangv_C[1] =dxdu2*repDuDs+
                dxdv2*repDvDs;
    tangv_C[2] =dxdu3*repDuDs+
                dxdv3*repDvDs;
}



__device__ float RK4_Init(float x1,float t1,float SSize)

{
float k1,k2,k3,k4;

k1=SSize*x1;
k2=SSize*(x1+0.5*SSize*t1);
k3=SSize*(x1+0.5*SSize*t1);
k4=SSize*(x1+SSize*t1);

return (1.0/6.0)*(k1+2*k2+2*k3+k4);
}

__device__ void InitialFS_LAC( float t1, float t2, float t3, float& dt1,
    float& dt2, float& dt3, float& d2t1, float& d2t2, float& d2t3,
    float& d3t1, float& d3t2, float& d3t3, float n1, float n2, float n3,
    float& dn1, float& dn2, float& dn3, float& d2n1, float& d2n2, float&
```

```
    d2n3, float Alpha, float InitRC, float Lambda, float ArcLength)
{
dt1=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*n1);
dt2=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*n2);
dt3=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*n3);
dn1=ArcLength*(-inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*t1);
dn2=ArcLength*(-inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*t2);
dn3=ArcLength*(-inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*t3);

d2t1=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dn1+
    inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*n1);
d2t2=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dn2+
    inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*n2);
d2t3=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dn3+
    inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*n3);
d2n1=ArcLength*(-inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dt1-
    inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*t1);
d2n2=ArcLength*(-inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dt2-
    inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*t2);
d2n3=ArcLength*(-inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dt3-
    inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*t3);

d3t1=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*d2n1+2.0*
    inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*dn1+inid2cLACdx2(0,Alpha,
    Lambda,InitRC,ArcLength)*n1);
d3t2=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*d2n2+
    2.0*inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*dn2+inid2cLACdx2(0,Alpha
    ,Lambda,InitRC,ArcLength)*n2);
d3t3=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*d2n3+
    2.0*inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*dn3+inid2cLACdx2(0,Alpha
    ,Lambda,InitRC,ArcLength)*n3);
}

__device__ void InitialFS_LASC( float t1, float t2, float t3, float& dt1,
    float& dt2, float& dt3, float& d2t1, float& d2t2, float& d2t3, float&
    d3t1, float& d3t2, float& d3t3, float n1, float n2, float n3, float&
    dn1, float& dn2, float& dn3, float& d2n1, float& d2n2, float& d2n3,
    float b1, float b2, float b3, float& db1, float& db2, float& db3,
    float& d2b1, float& d2b2, float& d2b3, float Alpha, float Beta, float
    InitRC, float Omega, float Lambda, float InitRT, float ArcLength)
{
dt1=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*n1);
dt2=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*n2);
dt3=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*n3);
dn1=ArcLength*(-inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*t1+
    iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)*b1);
dn2=ArcLength*(-inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*t2+
```

```
iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)*b2);
dn3=ArcLength*(-inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*t3+
    iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)*b3);
db1=ArcLength*(-iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)*n1);
db2=ArcLength*(-iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)*n2);
db3=ArcLength*(-iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)*n3);

d2t1=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dn1+
    inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*n1);
d2t2=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dn2+
    inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*n2);
d2t3=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dn3+
    inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*n3);
d2n1=ArcLength*(-inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*t1-
    inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dt1+inidtLACdx(0,Beta,
    Omega,InitRT,ArcLength)*b1+iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)
    *db1);
d2n2=ArcLength*(-inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*t2-
    inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dt2+inidtLACdx(0,Beta,
    Omega,InitRT,ArcLength)*b2+iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)
    *db2);
d2n3=ArcLength*(-inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*t3-
    inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dt3+inidtLACdx(0,Beta,
    Omega,InitRT,ArcLength)*b3+iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)
    *db3);
d2b1=ArcLength*(-iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)*dn1-
    inidtLACdx(0,Beta,Omega,InitRT,ArcLength)*n1);
d2b2=ArcLength*(-iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)*dn2-
    inidtLACdx(0,Beta,Omega,InitRT,ArcLength)*n2);
d2b3=ArcLength*(-iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)*dn3-
    inidtLACdx(0,Beta,Omega,InitRT,ArcLength)*n3);
d3t1=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*d2n1+
    2.0*inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*dn1+inid2cLACdx2(0,Alpha
    ,Lambda,InitRC,ArcLength)*n1);
d3t2=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*d2n2+
    2.0*inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*dn2+inid2cLACdx2(0,Alpha
    ,Lambda,InitRC,ArcLength)*n2);
d3t3=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*d2n3+
    2.0*inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*dn3+inid2cLACdx2(0,Alpha
    ,Lambda,InitRC,ArcLength)*n3);
}
__device__ void FS_LAC( float& x1, float& x2, float& x3, float& t1, float&
    t2, float& t3, float& dt1, float& dt2, float& dt3, float& d2t1,
    float& d2t2, float& d2t3, float& d3t1, float& d3t2, float& d3t3,
    float& n1, float& n2, float& n3, float& dn1, float& dn2, float&
    dn3, float& d2n1, float& d2n2, float& d2n3, float Alpha, float
    InitRC, float Lambda, float ArcLength, float initp, float stepsize)
```

```
{
x1=x1+RK4_Init(t1,dt1,stepsize);
x2=x2+RK4_Init(t2,dt2,stepsize);
x3=x3+RK4_Init(t3,dt3,stepsize);
t1=t1+RK4_Init(dt1,d2t1,stepsize);
t2=t2+RK4_Init(dt2,d2t2,stepsize);
t3=t3+RK4_Init(dt3,d2t3,stepsize);
n1=n1+RK4_Init(dn1,d2n1,stepsize);
n2=n2+RK4_Init(dn2,d2n2,stepsize);
n3=n3+RK4_Init(dn3,d2n3,stepsize);

dt1=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
    n1);
dt2=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*n2);
dt3=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*n3);
dn1=ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*t1);
dn2=ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*t2);
dn3=ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*t3);

d2t1=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
    dn1+inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*n1);
d2t2=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*dn2+inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
    n2);
d2t3=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*dn3+inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
    n3);
d2n1=ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*dt1-inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
    t1);
d2n2=ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*dt2-inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
    t2);
d2n3=ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*dt3-inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
    t3);

d3t1=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
    d2n1+2.0*inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*dn1+
    inid2cLACdx2(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*n1);
d3t2=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*d2n2+2.0*inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,
```

```
    ArcLength)*dn2+inid2cLACdx2(initp+stepsize,Alpha,Lambda,InitRC,ArcLength
    )*n2);
d3t3=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*d2n3+2.0*inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*dn3+inid2cLACdx2(initp+stepsize,Alpha,Lambda,InitRC,ArcLength
    )*n3);

}


__device__ void FS_LASC( float& x1, float& x2, float& x3, float& t1, float&
    t2, float& t3, float& dt1, float& dt2, float& dt3, float& d2t1,
    float& d2t2, float& d2t3, float& d3t1, float& d3t2, float& d3t3,
    float& n1, float& n2, float& n3, float& dn1, float& dn2, float& dn3,
    float& d2n1, float& d2n2, float& d2n3, float& b1, float& b2, float&
    b3, float& db1, float& db2, float& db3, float& d2b1, float& d2b2,
    float& d2b3, float Alpha, float Beta, float InitRC, float Omega, float
    Lambda, float InitRT, float ArcLength, float initp, float stepsize)
{
x1=x1+RK4_Init(t1,dt1,stepsize);
x2=x2+RK4_Init(t2,dt2,stepsize);
x3=x3+RK4_Init(t3,dt3,stepsize);
t1=t1+RK4_Init(dt1,d2t1,stepsize);
t2=t2+RK4_Init(dt2,d2t2,stepsize);
t3=t3+RK4_Init(dt3,d2t3,stepsize);
n1=n1+RK4_Init(dn1,d2n1,stepsize);
n2=n2+RK4_Init(dn2,d2n2,stepsize);
n3=n3+RK4_Init(dn3,d2n3,stepsize);
b1=b1+RK4_Init(db1,d2b1,stepsize);
b2=b2+RK4_Init(db2,d2b2,stepsize);
b3=b3+RK4_Init(db3,d2b3,stepsize);

dt1=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
    n1);
dt2=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*n2);
dt3=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*n3);
dn1=ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*t1+iniTorsion_LAC(initp+stepsize,Beta,Omega,InitRT,ArcLength)
    *b1);
dn2=ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*t2+iniTorsion_LAC(initp+stepsize,Beta,Omega,InitRT,ArcLength)
    *b2);
dn3=ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*t3+iniTorsion_LAC(initp+stepsize,Beta,Omega,InitRT,ArcLength)
    *b3);
```

```
db1=ArcLength*(-iniTorsion_LAC(initp+stepsize,Beta,Omega,InitRT,
     ArcLength)*n1);
db2=ArcLength*(-iniTorsion_LAC(initp+stepsize,Beta,Omega,InitRT,
     ArcLength)*n2);
db3=ArcLength*(-iniTorsion_LAC(initp+stepsize,Beta,Omega,InitRT,
     ArcLength)*n3);

d2t1=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
     dn1+inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*n1);
d2t2=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
     ArcLength)*dn2+inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
     n2);
d2t3=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
     ArcLength)*dn3+inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
     n3);
d2n1=ArcLength*(-inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength
     )*t1-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*dt1+
     inidtLACdx(initp+stepsize,Beta,Omega,InitRT,ArcLength)*b1+iniTorsion_LAC
     (initp+stepsize,Beta,Omega,InitRT,ArcLength)*db1);
d2n2=ArcLength*(-inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength
     )*t2-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*dt2+
     inidtLACdx(initp+stepsize,Beta,Omega,InitRT,ArcLength)*b2+iniTorsion_LAC
     (initp+stepsize,Beta,Omega,InitRT,ArcLength)*db2);
d2n3=ArcLength*(-inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength
     )*t3-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*dt3+
     inidtLACdx(initp+stepsize,Beta,Omega,InitRT,ArcLength)*b3+iniTorsion_LAC
     (initp+stepsize,Beta,Omega,InitRT,ArcLength)*db3);
d2b1=ArcLength*(-iniTorsion_LAC(initp+stepsize,Beta,Omega,InitRT,
     ArcLength)*dn1-inidtLACdx(initp+stepsize,Beta,Omega,InitRT,ArcLength)*n1
     );
d2b2=ArcLength*(-iniTorsion_LAC(initp+stepsize,Beta,Omega,InitRT,
     ArcLength)*dn2-inidtLACdx(initp+stepsize,Beta,Omega,InitRT,ArcLength)*n2
     );
d2b3=ArcLength*(-iniTorsion_LAC(initp+stepsize,Beta,Omega,InitRT,
     ArcLength)*dn3-inidtLACdx(initp+stepsize,Beta,Omega,InitRT,ArcLength)*n3
     );
d3t1=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
     ArcLength)*d2n1+2.0*inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,
     ArcLength)*dn1+inid2cLACdx2(initp+stepsize,Alpha,Lambda,InitRC,ArcLength
     )*n1);
d3t2=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
     ArcLength)*d2n2+2.0*inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,
     ArcLength)*dn2+inid2cLACdx2(initp+stepsize,Alpha,Lambda,InitRC,ArcLength
     )*n2);
d3t3=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
     ArcLength)*d2n3+2.0*inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,
     ArcLength)*dn3+inid2cLACdx2(initp+stepsize,Alpha,Lambda,InitRC,ArcLength
```

```
    )*n3);
}
__device__ void RK4_part( float old_x1, float old_x2, float old_x3, float
    old_t1, float old_t2, float old_t3, float old_dt1, float old_dt2,
    float old_dt3, float old_d2t1, float old_d2t2, float old_d2t3,
    float old_n1, float old_n2, float old_n3, float old_dn1, float
    old_dn2, float old_dn3, float old_d2n1, float old_d2n2, float
    old_d2n3, float& x1, float& x2, float& x3, float& t1, float& t2,
    float& t3, float& dt1, float& dt2, float& dt3, float Alpha, float
    InitRC, float Lambda, float ArcLength, float initp, float stepsize)
{
float n1,n2,n3;

x1=old_x1+RK4_Init(old_t1,old_dt1,stepsize);
x2=old_x2+RK4_Init(old_t2,old_dt2,stepsize);
x3=old_x3+RK4_Init(old_t3,old_dt3,stepsize);
t1=old_t1+RK4_Init(old_dt1,old_d2t1,stepsize);
t2=old_t2+RK4_Init(old_dt2,old_d2t2,stepsize);
t3=old_t3+RK4_Init(old_dt3,old_d2t3,stepsize);
n1=old_n1+RK4_Init(old_dn1,old_d2n1,stepsize);
n2=old_n2+RK4_Init(old_dn2,old_d2n2,stepsize);
n3=old_n3+RK4_Init(old_dn3,old_d2n3,stepsize);

dt1=ArcLength*inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*n1
    ;
dt2=ArcLength*inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*n2;
dt3=ArcLength*inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*n3;
}


__device__ float RK4_func(float (f)(float, float, float, float, float, float,
    float, float, float, float, float, float, float, float, float, float,
    float, float, float, float, float, float, float, float, float, float,
    float, float, float, float, float, float, float, float, float, float,
    float, float, mint), float u_s, float v_s, float u1_x1, float u1_x2,
    float u1_x3, float u1_t1, float u1_t2, float u1_t3, float u1_dt1,
    float u1_dt2, float u1_dt3, float i1_d2tu1, float i1_d2tu2, float
    i1_d2tu3, float i1_nu1, float i1_nu2, float i1_nu3, float i1_dnu1,
    float i1_dnu2, float i1_dnu3, float i1_d2nu1, float i1_d2nu2, float
    i1_d2nu3, float u2_x1, float u2_x2, float u2_x3, float u2_t1, float
    u2_t2, float u2_t3, float u2_dt1, float u2_dt2, float u2_dt3, float
    i2_d2tu1, float i2_d2tu2, float i2_d2tu3, float i2_nu1, float i2_nu2,
    float i2_nu3, float i2_dnu1, float i2_dnu2, float i2_dnu3, float
    i2_d2nu1, float i2_d2nu2, float i2_d2nu3, float i2_bu1, float i2_bu2,
    float i2_bu3, float i2_dbu1, float i2_dbu2, float i2_dbu3, float
```

```
    i2_d2bu1, float i2_d2bu2, float i2_d2bu3, float v1_x1, float v1_x2,
    float v1_x3, float v1_t1, float v1_t2, float v1_t3, float v1_dt1,
    float v1_dt2, float v1_dt3, float j1_d2tv1, float j1_d2tv2, float
    j1_d2tv3, float j1_nv1, float j1_nv2, float j1_nv3, float j1_dnv1,
    float j1_dnv2, float j1_dnv3, float j1_d2nv1, float j1_d2nv2, float
    j1_d2nv3, float v2_x1, float v2_x2, float v2_x3, float v2_t1, float
    v2_t2, float v2_t3, float v2_dt1, float v2_dt2, float v2_dt3, float
    j2_d2tv1, float j2_d2tv2, float j2_d2tv3, float j2_nv1, float j2_nv2,
    float j2_nv3, float j2_dnv1, float j2_dnv2, float j2_dnv3, float
    j2_d2nv1, float j2_d2nv2, float j2_d2nv3, float j2_bv1, float j2_bv2,
    float j2_bv3, float j2_dbv1, float j2_dbv2, float j2_dbv3, float
    j2_d2bv1, float j2_d2bv2, float j2_d2bv3, mint optimum, int u_or_v)

{

float k1,k2,k3,k4;
float
    new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,new_u1_t2,new_u1_t3,new_u1_dt1,
    new_u1_dt2,new_u1_dt3,new_i1_d2tu1,new_i1_d2tu2,new_i1_d2tu3;
float
    new_u2_x1,new_u2_x2,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,
    new_u2_dt2,new_u2_dt3,new_i2_d2tu1,new_i2_d2tu2,new_i2_d2tu3;
float
    new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,new_v1_dt1,
    new_v1_dt2,new_v1_dt3,new_j1_d2tv1,new_j1_d2tv2,new_j1_d2tv3;
float
    new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,new_v2_t2,new_v2_t3,new_v2_dt1,
    new_v2_dt2,new_v2_dt3,new_j2_d2tv1,new_j2_d2tv2,new_j2_d2tv3;


if(u_or_v==uuu)
{
k1=h*f(u_s,v_s,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,u2_x1,
    u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,v1_x3,
    v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,
    v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum);

RK4_part(u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,i1_d2tu1,
    i1_d2tu2,i1_d2tu3,i1_nu1,i1_nu2,i1_nu3,i1_dnu1,i1_dnu2,i1_dnu3,i1_d2nu1,
    i1_d2nu2,i1_d2nu3,new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,new_u1_t2,
    new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3,AlphaLAC,InitRCLAC,Lambda_Max
    _LAC,ArcLength_Max_LAC,u_s,0.5*k1);
RK4_part(u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,
    i2_d2tu1,i2_d2tu2,i2_d2tu3,i2_nu1,i2_nu2,i2_nu3,i2_dnu1,i2_dnu2,i2_dnu3,
    i2_d2nu1,i2_d2nu2,i2_d2nu3,new_u2_x1,new_u2_x2,new_u2_x3,new_u2_t1,
    new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2,new_u2_dt3,AlphaLAC,InitRCLAC,
    Lambda_Max_LASC,ArcLength_Max_LASC,u_s,0.5*k1);
```

```
k2=h*f(u_s,v_s,new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,new_u1_t2,new_u1_t3,
     new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u2_x1,new_u2_x2,new_u2_x3,new_u2_t1
     ,new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2,new_u2_dt3,v1_x1,v1_x2,v1_x3,
     v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,
     v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum);

RK4_part(u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,i1_d2tu1,
     i1_d2tu2,i1_d2tu3,i1_nu1,i1_nu2,i1_nu3,i1_dnu1,i1_dnu2,i1_dnu3,i1_d2nu1,
     i1_d2nu2,i1_d2nu3,new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,new_u1_t2,
     new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3,AlphaLAC,InitRCLAC,Lambda_Max
     _LAC,ArcLength_Max_LAC,u_s,0.5*k2);
RK4_part(u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,
     i2_d2tu1,i2_d2tu2,i2_d2tu3,i2_nu1,i2_nu2,i2_nu3,i2_dnu1,i2_dnu2,i2_dnu3,
     i2_d2nu1,i2_d2nu2,i2_d2nu3,new_u2_x1,new_u2_x2,new_u2_x3,new_u2_t1,
     new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2,new_u2_dt3,AlphaLAC,InitRCLAC,
     Lambda_Max_LASC,ArcLength_Max_LASC,u_s,0.5*k2);

k3=h*f(u_s,v_s,new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,new_u1_t2,new_u1_t3,
     new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u2_x1,new_u2_x2,new_u2_x3,new_u2_t1
     ,new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2,new_u2_dt3,v1_x1,v1_x2,v1_x3,
     v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,
     v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum);

RK4_part(u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,i1_d2tu1,
     i1_d2tu2,i1_d2tu3,i1_nu1,i1_nu2,i1_nu3,i1_dnu1,i1_dnu2,i1_dnu3,i1_d2nu1,
     i1_d2nu2,i1_d2nu3,new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,new_u1_t2,
     new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3,AlphaLAC,InitRCLAC,Lambda_Max
     _LAC,ArcLength_Max_LAC,u_s,k3);

RK4_part(u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,i2_d2tu1,
     i2_d2tu2,i2_d2tu3,i2_nu1,i2_nu2,i2_nu3,i2_dnu1,i2_dnu2,i2_dnu3,i2_d2nu1,
     i2_d2nu2,i2_d2nu3,new_u2_x1,new_u2_x2,new_u2_x3,new_u2_t1,new_u2_t2,
     new_u2_t3,new_u2_dt1,new_u2_dt2,new_u2_dt3,AlphaLAC,InitRCLAC,Lambda_Max
     _LASC,ArcLength_Max_LASC,u_s,k3);

k4=h*f(u_s,v_s,new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,new_u1_t2,new_u1_t3,
     new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u2_x1,new_u2_x2,new_u2_x3,new_u2_t1
     ,new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2,new_u2_dt3,v1_x1,v1_x2,v1_x3,
     v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,
     v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum);

return (1.0/6.0)*(k1+2*k2+2*k3+k4);
}
else if (u_or_v==vvv)
{
```

```
k1=h*f(u_s,v_s,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,u2_x1,
    u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,v1_x3,
    v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,
    v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum);

RK4_part(v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,j1_d2tv1,
    j1_d2tv2,j1_d2tv3,j1_nv1,j1_nv2,j1_nv3,j1_dnv1,j1_dnv2,j1_dnv3,j1_d2nv1,
    j1_d2nv2,j1_d2nv3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,
    new_v1_t3,new_v1_dt1,new_v1_dt2,new_v1_dt3,AlphaLAC,InitRCLAC,Lambda_Min
    _LAC,ArcLength_Min_LAC,v_s,0.5*k1);

RK4_part(v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,j2_d2tv1,
    j2_d2tv2,j2_d2tv3,j2_nv1,j2_nv2,j2_nv3,j2_dnv1,j2_dnv2,j2_dnv3,j2_d2nv1,
    j2_d2nv2,j2_d2nv3,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,new_v2_t2,
    new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,AlphaLAC,InitRCLAC,Lambda_Min
    _LASC,ArcLength_Min_LASC,v_s,0.5*k1);



k2=h*f(u_s,v_s,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,u2_x1,
    u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,new_v1_x1,new_v1_x2,
    new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,new_v1_dt1,new_v1_dt2,new_v1_dt3
    ,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,new_v2_t2,new_v2_t3,new_v2_dt1,
    new_v2_dt2,new_v2_dt3,optimum);

RK4_part(v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,j1_d2tv1,
    j1_d2tv2,j1_d2tv3,j1_nv1,j1_nv2,j1_nv3,j1_dnv1,j1_dnv2,j1_dnv3,j1_d2nv1,
    j1_d2nv2,j1_d2nv3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,
    new_v1_t3,new_v1_dt1,new_v1_dt2,new_v1_dt3,AlphaLAC,InitRCLAC,Lambda_Min
    _LAC,ArcLength_Min_LAC,v_s,0.5*k2);
RK4_part(v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,
    j2_d2tv1,j2_d2tv2,j2_d2tv3,j2_nv1,j2_nv2,j2_nv3,j2_dnv1,j2_dnv2,j2_dnv3,
    j2_d2nv1,j2_d2nv2,j2_d2nv3,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,
    new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,AlphaLAC,InitRCLAC,
    Lambda_Min_LASC,ArcLength_Min_LASC,v_s,0.5*k2);


k3=h*f(u_s,v_s,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,u2_x1,
    u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,new_v1_x1,new_v1_x2,
    new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,new_v1_dt1,new_v1_dt2,new_v1_dt3
    ,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,new_v2_t2,new_v2_t3,new_v2_dt1,
    new_v2_dt2,new_v2_dt3,optimum);

RK4_part(v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,j1_d2tv1,
    j1_d2tv2,j1_d2tv3,j1_nv1,j1_nv2,j1_nv3,j1_dnv1,j1_dnv2,j1_dnv3,j1_d2nv1,
    j1_d2nv2,j1_d2nv3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,
```

```
    new_v1_t3,new_v1_dt1,new_v1_dt2,new_v1_dt3,AlphaLAC,InitRCLAC,Lambda_Min
    _LAC,ArcLength_Min_LAC,v_s,k3);
RK4_part(v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,
    j2_d2tv1,j2_d2tv2,j2_d2tv3,j2_nv1,j2_nv2,j2_nv3,j2_dnv1,j2_dnv2,j2_dnv3,
    j2_d2nv1,j2_d2nv2,j2_d2nv3,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,
    new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,AlphaLAC,InitRCLAC,
    Lambda_Min_LASC,ArcLength_Min_LASC,v_s,k3);

k4=h*f(u_s,v_s,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,u2_x1,
    u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,new_v1_x1,new_v1_x2,
    new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,new_v1_dt1,new_v1_dt2,new_v1_dt3
    ,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,new_v2_t2,new_v2_t3,new_v2_dt1,
    new_v2_dt2,new_v2_dt3,optimum);

return (1.0/6.0)*(k1+2*k2+2*k3+k4);
}
}


__device__ void Integration( float pu[], float pv[], float u1_x1[], float
    u1_x2[], float u1_x3[], float u1_t1[], float u1_t2[], float u1_t3[],
    float u1_dt1[], float u1_dt2[], float u1_dt3[], float u1_d2t1[], float
    u1_d2t2[], float u1_d2t3[], float u1_d3t1[], float u1_d3t2[], float
    u1_d3t3[], float u2_x1[], float u2_x2[], float u2_x3[], float u2_t1[],
    float u2_t2[], float u2_t3[], float u2_dt1[], float u2_dt2[], float
    u2_dt3[], float u2_d2t1[], float u2_d2t2[], float u2_d2t3[], float
    u2_d3t1[], float u2_d3t2[], float u2_d3t3[], float v1_x1[], float
    v1_x2[], float v1_x3[], float v1_t1[], float v1_t2[], float v1_t3[],
    float v1_dt1[], float v1_dt2[], float v1_dt3[], float v1_d2t1[], float
    v1_d2t2[], float v1_d2t3[], float v1_d3t1[], float v1_d3t2[], float
    v1_d3t3[], float v2_x1[], float v2_x2[], float v2_x3[], float v2_t1[],
    float v2_t2[], float v2_t3[], float v2_dt1[], float v2_dt2[], float
    v2_dt3[], float v2_d2t1[], float v2_d2t2[], float v2_d2t3[], float
    v2_d3t1[], float v2_d3t2[], float v2_d3t3[], mint idx, mint optimum)
{
float t1_Max_LAC=-1*ScaleToOrigin_Max_LAC*ArcLength_Max_LAC,
    t2_Max_LAC=0, t3_Max_LAC=0;
float n1_Max_LAC=0, n2_Max_LAC=0,
    n3_Max_LAC=1*ScaleToOrigin_Max_LAC*ArcLength_Max_LAC;
float t1_Min_LAC=0, t2_Min_LAC=-1*ScaleToOrigin_Min_LAC*ArcLength_Min_LAC,
    t3_Min_LAC=0;
float n1_Min_LAC=0, n2_Min_LAC=0,
    n3_Min_LAC=-1*ScaleToOrigin_Min_LAC*ArcLength_Min_LAC;

float t1_Max_LASC=-1*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC,
    t2_Max_LASC=0, t3_Max_LASC=0;
float n1_Max_LASC=0,
```

```
    n2_Max_LASC=-0.5930588126117428*ScaleToOrigin_Max_LASC*ArcLength_Max
    _LASC,
    n3_Max_LASC=0.8051591425200051*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC;
float b1_Max_LASC=0,
    b2_Max_LASC=0.8051591425200051*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC
    ,
    b3_Max_LASC=0.5930588126117428*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC;

float t1_Min_LASC=0,
    t2_Min_LASC=-1*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC, t3_Min_LASC=0;
float
    n1_Min_LASC=-0.5930588126117428*ScaleToOrigin_Min_LASC*ArcLength_Min
    _LASC, n2_Min_LASC=0,
    n3_Min_LASC=-0.8051591425200051*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC;
float
    b1_Min_LASC=0.8051591425200051*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC
    , b2_Min_LASC=0,
    b3_Min_LASC=-0.5930588126117428*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC;

float Init_u=0.0, Init_v=0.0;

float dt1_Max_LAC, dt2_Max_LAC, dt3_Max_LAC;
float d2t1_Max_LAC, d2t2_Max_LAC, d2t3_Max_LAC;
float d3t1_Max_LAC, d3t2_Max_LAC, d3t3_Max_LAC;
float dt1_Min_LAC, dt2_Min_LAC, dt3_Min_LAC;
float d2t1_Min_LAC, d2t2_Min_LAC, d2t3_Min_LAC;
float d3t1_Min_LAC, d3t2_Min_LAC, d3t3_Min_LAC;
float dn1_Max_LAC, dn2_Max_LAC, dn3_Max_LAC;
float d2n1_Max_LAC, d2n2_Max_LAC, d2n3_Max_LAC;
float dn1_Min_LAC, dn2_Min_LAC, dn3_Min_LAC;
float d2n1_Min_LAC, d2n2_Min_LAC, d2n3_Min_LAC;

float dt1_Max_LASC, dt2_Max_LASC, dt3_Max_LASC;
float d2t1_Max_LASC, d2t2_Max_LASC, d2t3_Max_LASC;
float d3t1_Max_LASC, d3t2_Max_LASC, d3t3_Max_LASC;
float dt1_Min_LASC, dt2_Min_LASC, dt3_Min_LASC;
float d2t1_Min_LASC, d2t2_Min_LASC, d2t3_Min_LASC;
float d3t1_Min_LASC, d3t2_Min_LASC, d3t3_Min_LASC;
float dn1_Max_LASC, dn2_Max_LASC, dn3_Max_LASC;
float d2n1_Max_LASC, d2n2_Max_LASC, d2n3_Max_LASC;
float dn1_Min_LASC, dn2_Min_LASC, dn3_Min_LASC;
float d2n1_Min_LASC, d2n2_Min_LASC, d2n3_Min_LASC;
float db1_Max_LASC, db2_Max_LASC, db3_Max_LASC;
float d2b1_Max_LASC, d2b2_Max_LASC, d2b3_Max_LASC;
float db1_Min_LASC, db2_Min_LASC, db3_Min_LASC;
float d2b1_Min_LASC, d2b2_Min_LASC, d2b3_Min_LASC;
```

```
float dummy_u1_x1=Point00x, dummy_u1_x2=Point00y, dummy_u1_x3=Point00z;
float dummy_u2_x1=Point30x, dummy_u2_x2=Point30y, dummy_u2_x3=Point30z;
float dummy_v1_x1=Point00x, dummy_v1_x2=Point00y, dummy_v1_x3=Point00z;
float dummy_v2_x1=Point03x, dummy_v2_x2=Point03y, dummy_v2_x3=Point03z;
float SSize_u, SSize_v, ratio;

InitialFS_LAC(t1_Max_LAC,t2_Max_LAC,t3_Max_LAC,dt1_Max_LAC,dt2_Max_LAC,dt3_Max
    _LAC,d2t1_Max_LAC,d2t2_Max_LAC,d2t3_Max_LAC,d3t1_Max_LAC,d3t2_Max_LAC,
    d3t3_Max_LAC,n1_Max_LAC,n2_Max_LAC,n3_Max_LAC,dn1_Max_LAC,dn2_Max_LAC,
    dn3_Max_LAC,d2n1_Max_LAC,d2n2_Max_LAC,d2n3_Max_LAC,AlphaLAC,InitRCLAC,
    Lambda_Max_LAC,ArcLength_Max_LAC);

InitialFS_LASC(t1_Max_LASC,t2_Max_LASC,t3_Max_LASC,dt1_Max_LASC,dt2_Max_LASC,
    dt3_Max_LASC,d2t1_Max_LASC,d2t2_Max_LASC,d2t3_Max_LASC,d3t1_Max_LASC,
    d3t2_Max_LASC,d3t3_Max_LASC,n1_Max_LASC,n2_Max_LASC,n3_Max_LASC,dn1_Max
    _LASC,dn2_Max_LASC,dn3_Max_LASC,d2n1_Max_LASC,d2n2_Max_LASC,d2n3_Max
    _LASC,b1_Max_LASC,b2_Max_LASC,b3_Max_LASC,db1_Max_LASC,db2_Max_LASC,
    db3_Max_LASC,d2b1_Max_LASC,d2b2_Max_LASC,d2b3_Max_LASC,AlphaLAC,BetaLASC
    ,InitRCLAC,OmegaLASC,Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC)
    ;

InitialFS_LAC(t1_Min_LAC,t2_Min_LAC,t3_Min_LAC,dt1_Min_LAC,dt2_Min_LAC,dt3_Min
    _LAC,d2t1_Min_LAC,d2t2_Min_LAC,d2t3_Min_LAC,d3t1_Min_LAC,d3t2_Min_LAC,
    d3t3_Min_LAC,n1_Min_LAC,n2_Min_LAC,n3_Min_LAC,dn1_Min_LAC,dn2_Min_LAC,
    dn3_Min_LAC,d2n1_Min_LAC,d2n2_Min_LAC,d2n3_Min_LAC,AlphaLAC,InitRCLAC,
    Lambda_Min_LAC,ArcLength_Min_LAC);

InitialFS_LASC(t1_Min_LASC,t2_Min_LASC,t3_Min_LASC,dt1_Min_LASC,dt2_Min_LASC,
    dt3_Min_LASC,d2t1_Min_LASC,d2t2_Min_LASC,d2t3_Min_LASC,d3t1_Min_LASC,
    d3t2_Min_LASC,d3t3_Min_LASC,n1_Min_LASC,n2_Min_LASC,n3_Min_LASC,dn1_Min
    _LASC,dn2_Min_LASC,dn3_Min_LASC,d2n1_Min_LASC,d2n2_Min_LASC,d2n3_Min
    _LASC,b1_Min_LASC,b2_Min_LASC,b3_Min_LASC,db1_Min_LASC,db2_Min_LASC,
    db3_Min_LASC,d2b1_Min_LASC,d2b2_Min_LASC,d2b3_Min_LASC,AlphaLAC,BetaLASC
    ,InitRCLAC,OmegaLASC,Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC)
    ;


if(optimum==min)
    {

    for(int i=1;i<idx+1;i++)
    {

    FS_LAC(dummy_u1_x1,dummy_u1_x2,dummy_u1_x3,t1_Max_LAC,t2_Max_LAC,t3_Max
    _LAC,dt1_Max_LAC,dt2_Max_LAC,dt3_Max_LAC,d2t1_Max_LAC,d2t2_Max_LAC,
    d2t3_Max_LAC,d3t1_Max_LAC,d3t2_Max_LAC,d3t3_Max_LAC,n1_Max_LAC,n2_Max
    _LAC,n3_Max_LAC,dn1_Max_LAC,dn2_Max_LAC,dn3_Max_LAC,d2n1_Max_LAC,
```

```
     d2n2_Max_LAC,d2n3_Max_LAC,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
     ArcLength_Max_LAC,Init_u,h);

     FS_LASC(dummy_u2_x1,dummy_u2_x2,dummy_u2_x3,t1_Max_LASC,t2_Max_LASC,
     t3_Max_LASC,dt1_Max_LASC,dt2_Max_LASC,dt3_Max_LASC,d2t1_Max_LASC,
     d2t2_Max_LASC,d2t3_Max_LASC,d3t1_Max_LASC,d3t2_Max_LASC,d3t3_Max_LASC,
     n1_Max_LASC,n2_Max_LASC,n3_Max_LASC,dn1_Max_LASC,dn2_Max_LASC,dn3_Max
     _LASC,d2n1_Max_LASC,d2n2_Max_LASC,d2n3_Max_LASC,b1_Max_LASC,b2_Max_LASC,
     b3_Max_LASC,db1_Max_LASC,db2_Max_LASC,db3_Max_LASC,d2b1_Max_LASC,
     d2b2_Max_LASC,d2b3_Max_LASC,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
     Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,Init_u,h);

     Init_u+=h;
     }
     }
else if(optimum==max)
     {
     for(int i=1;i<idx+1;i++)
     {

     FS_LAC(dummy_v1_x1,dummy_v1_x2,dummy_v1_x3,t1_Min_LAC,t2_Min_LAC,t3_Min
     _LAC,dt1_Min_LAC,dt2_Min_LAC,dt3_Min_LAC,d2t1_Min_LAC,d2t2_Min_LAC,
     d2t3_Min_LAC,d3t1_Min_LAC,d3t2_Min_LAC,d3t3_Min_LAC,n1_Min_LAC,n2_Min
     _LAC,n3_Min_LAC,dn1_Min_LAC,dn2_Min_LAC,dn3_Min_LAC,d2n1_Min_LAC,
     d2n2_Min_LAC,d2n3_Min_LAC,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
     ArcLength_Min_LAC,Init_v,h);

     FS_LASC(dummy_v2_x1,dummy_v2_x2,dummy_v2_x3,t1_Min_LASC,t2_Min_LASC,
     t3_Min_LASC,dt1_Min_LASC,dt2_Min_LASC,dt3_Min_LASC,d2t1_Min_LASC,
     d2t2_Min_LASC,d2t3_Min_LASC,d3t1_Min_LASC,d3t2_Min_LASC,d3t3_Min_LASC,
     n1_Min_LASC,n2_Min_LASC,n3_Min_LASC,dn1_Min_LASC,dn2_Min_LASC,dn3_Min
     _LASC,d2n1_Min_LASC,d2n2_Min_LASC,d2n3_Min_LASC,b1_Min_LASC,b2_Min_LASC,
     b3_Min_LASC,db1_Min_LASC,db2_Min_LASC,db3_Min_LASC,d2b1_Min_LASC,
     d2b2_Min_LASC,d2b3_Min_LASC,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
     Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,Init_v,h);
     Init_v+=h;
     }
     }

pu[0]=Init_u;
pv[0]=Init_v;
u1_x1[0]=dummy_u1_x1;
u1_x2[0]=dummy_u1_x2;
u1_x3[0]=dummy_u1_x3;
u1_t1[0]=t1_Max_LAC;
u1_t2[0]=t2_Max_LAC;
u1_t3[0]=t3_Max_LAC;
```

```
u1_dt1[0]=dt1_Max_LAC;
u1_dt2[0]=dt2_Max_LAC;
u1_dt3[0]=dt3_Max_LAC;
u1_d2t1[0]=d2t1_Max_LAC;
u1_d2t2[0]=d2t2_Max_LAC;
u1_d2t3[0]=d2t3_Max_LAC;
u1_d3t1[0]=d3t1_Max_LAC;
u1_d3t2[0]=d3t2_Max_LAC;
u1_d3t3[0]=d3t3_Max_LAC;
u2_x1[0]=dummy_u2_x1;
u2_x2[0]=dummy_u2_x2;
u2_x3[0]=dummy_u2_x3;
u2_t1[0]=t1_Max_LASC;
u2_t2[0]=t2_Max_LASC;
u2_t3[0]=t3_Max_LASC;
u2_dt1[0]=dt1_Max_LASC;
u2_dt2[0]=dt2_Max_LASC;
u2_dt3[0]=dt3_Max_LASC;
u2_d2t1[0]=d2t1_Max_LASC;
u2_d2t2[0]=d2t2_Max_LASC;
u2_d2t3[0]=d2t3_Max_LASC;
u2_d3t1[0]=d3t1_Max_LASC;
u2_d3t2[0]=d3t2_Max_LASC;
u2_d3t3[0]=d3t3_Max_LASC;
v1_x1[0]=dummy_v1_x1;
v1_x2[0]=dummy_v1_x2;
v1_x3[0]=dummy_v1_x3;
v1_t1[0]=t1_Min_LAC;
v1_t2[0]=t2_Min_LAC;
v1_t3[0]=t3_Min_LAC;
v1_dt1[0]=dt1_Min_LAC;
v1_dt2[0]=dt2_Min_LAC;
v1_dt3[0]=dt3_Min_LAC;
v1_d2t1[0]=d2t1_Min_LAC;
v1_d2t2[0]=d2t2_Min_LAC;
v1_d2t3[0]=d2t3_Min_LAC;
v1_d3t1[0]=d3t1_Min_LAC;
v1_d3t2[0]=d3t2_Min_LAC;
v1_d3t3[0]=d3t3_Min_LAC;
v2_x1[0]=dummy_v2_x1;
v2_x2[0]=dummy_v2_x2;
v2_x3[0]=dummy_v2_x3;
v2_t1[0]=t1_Min_LASC;
v2_t2[0]=t2_Min_LASC;
v2_t3[0]=t3_Min_LASC;
v2_dt1[0]=dt1_Min_LASC;
v2_dt2[0]=dt2_Min_LASC;
```

```
v2_dt3[0]=dt3_Min_LASC;
v2_d2t1[0]=d2t1_Min_LASC;
v2_d2t2[0]=d2t2_Min_LASC;
v2_d2t3[0]=d2t3_Min_LASC;
v2_d3t1[0]=d3t1_Min_LASC;
v2_d3t2[0]=d3t2_Min_LASC;
v2_d3t3[0]=d3t3_Min_LASC;



for(int j=1;j<n;j++)
    {

    SSize_u=RK4_func(iniDuDs,Init_u,Init_v,dummy_u1_x1,dummy_u1_x2,dummy_u1
    _x3,t1_Max_LAC,t2_Max_LAC,t3_Max_LAC,dt1_Max_LAC,dt2_Max_LAC,dt3_Max_LAC
    ,d2t1_Max_LAC,d2t2_Max_LAC,d2t3_Max_LAC,n1_Max_LAC,n2_Max_LAC,n3_Max_LAC
    ,dn1_Max_LAC,dn2_Max_LAC,dn3_Max_LAC,d2n1_Max_LAC,d2n2_Max_LAC,d2n3_Max
    _LAC,dummy_u2_x1,dummy_u2_x2,dummy_u2_x3,t1_Max_LASC,t2_Max_LASC,t3_Max
    _LASC,dt1_Max_LASC,dt2_Max_LASC,dt3_Max_LASC,d2t1_Max_LASC,d2t2_Max_LASC
    ,d2t3_Max_LASC,n1_Max_LASC,n2_Max_LASC,n3_Max_LASC,dn1_Max_LASC,dn2_Max
    _LASC,dn3_Max_LASC,d2n1_Max_LASC,d2n2_Max_LASC,d2n3_Max_LASC,b1_Max_LASC
    ,b2_Max_LASC,b3_Max_LASC,db1_Max_LASC,db2_Max_LASC,db3_Max_LASC,d2b1_Max
    _LASC,d2b2_Max_LASC,d2b3_Max_LASC,dummy_v1_x1,dummy_v1_x2,dummy_v1_x3,
    t1_Min_LAC,t2_Min_LAC,t3_Min_LAC,dt1_Min_LAC,dt2_Min_LAC,dt3_Min_LAC,
    d2t1_Min_LAC,d2t2_Min_LAC,d2t3_Min_LAC,n1_Min_LAC,n2_Min_LAC,n3_Min_LAC,
    dn1_Min_LAC,dn2_Min_LAC,dn3_Min_LAC,d2n1_Min_LAC,d2n2_Min_LAC,d2n3_Min
    _LAC,dummy_v2_x1,dummy_v2_x2,dummy_v2_x3,t1_Min_LASC,t2_Min_LASC,t3_Min
    _LASC,dt1_Min_LASC,dt2_Min_LASC,dt3_Min_LASC,d2t1_Min_LASC,d2t2_Min_LASC
    ,d2t3_Min_LASC,n1_Min_LASC,n2_Min_LASC,n3_Min_LASC,dn1_Min_LASC,dn2_Min
    _LASC,dn3_Min_LASC,d2n1_Min_LASC,d2n2_Min_LASC,d2n3_Min_LASC,b1_Min_LASC
    ,b2_Min_LASC,b3_Min_LASC,db1_Min_LASC,db2_Min_LASC,db3_Min_LASC,d2b1_Min
    _LASC,d2b2_Min_LASC,d2b3_Min_LASC,optimum,uuu);

    SSize_v=RK4_func(iniDvDs,Init_u,Init_v,dummy_u1_x1,dummy_u1_x2,dummy_u1
    _x3,t1_Max_LAC,t2_Max_LAC,t3_Max_LAC,dt1_Max_LAC,dt2_Max_LAC,dt3_Max_LAC
    ,d2t1_Max_LAC,d2t2_Max_LAC,d2t3_Max_LAC,n1_Max_LAC,n2_Max_LAC,n3_Max_LAC
    ,dn1_Max_LAC,dn2_Max_LAC,dn3_Max_LAC,d2n1_Max_LAC,d2n2_Max_LAC,d2n3_Max
    _LAC,dummy_u2_x1,dummy_u2_x2,dummy_u2_x3,t1_Max_LASC,t2_Max_LASC,t3_Max
    _LASC,dt1_Max_LASC,dt2_Max_LASC,dt3_Max_LASC,d2t1_Max_LASC,d2t2_Max_LASC
    ,d2t3_Max_LASC,n1_Max_LASC,n2_Max_LASC,n3_Max_LASC,dn1_Max_LASC,dn2_Max
    _LASC,dn3_Max_LASC,d2n1_Max_LASC,d2n2_Max_LASC,d2n3_Max_LASC,b1_Max_LASC
    ,b2_Max_LASC,b3_Max_LASC,db1_Max_LASC,db2_Max_LASC,db3_Max_LASC,d2b1_Max
    _LASC,d2b2_Max_LASC,d2b3_Max_LASC,dummy_v1_x1,dummy_v1_x2,dummy_v1_x3,
    t1_Min_LAC,t2_Min_LAC,t3_Min_LAC,dt1_Min_LAC,dt2_Min_LAC,dt3_Min_LAC,
    d2t1_Min_LAC,d2t2_Min_LAC,d2t3_Min_LAC,n1_Min_LAC,n2_Min_LAC,n3_Min_LAC,
    dn1_Min_LAC,dn2_Min_LAC,dn3_Min_LAC,d2n1_Min_LAC,d2n2_Min_LAC,d2n3_Min
    _LAC,dummy_v2_x1,dummy_v2_x2,dummy_v2_x3,t1_Min_LASC,t2_Min_LASC,t3_Min
```

```
_LASC,dt1_Min_LASC,dt2_Min_LASC,dt3_Min_LASC,d2t1_Min_LASC,d2t2_Min_LASC
,d2t3_Min_LASC,n1_Min_LASC,n2_Min_LASC,n3_Min_LASC,dn1_Min_LASC,dn2_Min
_LASC,dn3_Min_LASC,d2n1_Min_LASC,d2n2_Min_LASC,d2n3_Min_LASC,b1_Min_LASC
,b2_Min_LASC,b3_Min_LASC,db1_Min_LASC,db2_Min_LASC,db3_Min_LASC,d2b1_Min
_LASC,d2b2_Min_LASC,d2b3_Min_LASC,optimum,vvv);

if((Init_u+SSize_u) > 1.0)
    {
    if(Init_u==1.0)
        {
        SSize_u=0.0;
        SSize_v=0.0;
        }
    else
        {
        ratio=(1.0-Init_u)/SSize_u;
        SSize_u*=ratio;
        SSize_v*=ratio;
        }
    }

if((Init_v+SSize_v) > 1.0)
    {
    if(Init_v==1.0)
        {
        SSize_u=0.0;
        SSize_v=0.0;
        }
    else
        {
        ratio=(1.0-Init_v)/SSize_v;
        SSize_u*=ratio;
        SSize_v*=ratio;
        }
    }

FS_LAC(dummy_u1_x1,dummy_u1_x2,dummy_u1_x3,t1_Max_LAC,t2_Max_LAC,t3_Max
_LAC,dt1_Max_LAC,dt2_Max_LAC,dt3_Max_LAC,d2t1_Max_LAC,d2t2_Max_LAC,
d2t3_Max_LAC,d3t1_Max_LAC,d3t2_Max_LAC,d3t3_Max_LAC,n1_Max_LAC,n2_Max
_LAC,n3_Max_LAC,dn1_Max_LAC,dn2_Max_LAC,dn3_Max_LAC,d2n1_Max_LAC,
d2n2_Max_LAC,d2n3_Max_LAC,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
ArcLength_Max_LAC,Init_u,SSize_u);

FS_LASC(dummy_u2_x1,dummy_u2_x2,dummy_u2_x3,t1_Max_LASC,t2_Max_LASC,
t3_Max_LASC,dt1_Max_LASC,dt2_Max_LASC,dt3_Max_LASC,d2t1_Max_LASC,
d2t2_Max_LASC,d2t3_Max_LASC,d3t1_Max_LASC,d3t2_Max_LASC,d3t3_Max_LASC,
n1_Max_LASC,n2_Max_LASC,n3_Max_LASC,dn1_Max_LASC,dn2_Max_LASC,dn3_Max
```

```
_LASC,d2n1_Max_LASC,d2n2_Max_LASC,d2n3_Max_LASC,b1_Max_LASC,b2_Max_LASC,
b3_Max_LASC,db1_Max_LASC,db2_Max_LASC,db3_Max_LASC,d2b1_Max_LASC,
d2b2_Max_LASC,d2b3_Max_LASC,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,Init_u,SSize_u);


FS_LAC(dummy_v1_x1,dummy_v1_x2,dummy_v1_x3,t1_Min_LAC,t2_Min_LAC,t3_Min
_LAC,dt1_Min_LAC,dt2_Min_LAC,dt3_Min_LAC,d2t1_Min_LAC,d2t2_Min_LAC,
d2t3_Min_LAC,d3t1_Min_LAC,d3t2_Min_LAC,d3t3_Min_LAC,n1_Min_LAC,n2_Min
_LAC,n3_Min_LAC,dn1_Min_LAC,dn2_Min_LAC,dn3_Min_LAC,d2n1_Min_LAC,
d2n2_Min_LAC,d2n3_Min_LAC,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
ArcLength_Min_LAC,Init_v,SSize_v);

FS_LASC(dummy_v2_x1,dummy_v2_x2,dummy_v2_x3,t1_Min_LASC,t2_Min_LASC,
t3_Min_LASC,dt1_Min_LASC,dt2_Min_LASC,dt3_Min_LASC,d2t1_Min_LASC,
d2t2_Min_LASC,d2t3_Min_LASC,d3t1_Min_LASC,d3t2_Min_LASC,d3t3_Min_LASC,
n1_Min_LASC,n2_Min_LASC,n3_Min_LASC,dn1_Min_LASC,dn2_Min_LASC,dn3_Min
_LASC,d2n1_Min_LASC,d2n2_Min_LASC,d2n3_Min_LASC,b1_Min_LASC,b2_Min_LASC,
b3_Min_LASC,db1_Min_LASC,db2_Min_LASC,db3_Min_LASC,d2b1_Min_LASC,
d2b2_Min_LASC,d2b3_Min_LASC,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,Init_v,SSize_v);

Init_u+=SSize_u;
Init_v+=SSize_v;
pu[j]=Init_u;
pv[j]=Init_v;
u1_x1[j]=dummy_u1_x1;
u1_x2[j]=dummy_u1_x2;
u1_x3[j]=dummy_u1_x3;
u1_t1[j]=t1_Max_LAC;
u1_t2[j]=t2_Max_LAC;
u1_t3[j]=t3_Max_LAC;
u1_dt1[j]=dt1_Max_LAC;
u1_dt2[j]=dt2_Max_LAC;
u1_dt3[j]=dt3_Max_LAC;
u1_d2t1[j]=d2t1_Max_LAC;
u1_d2t2[j]=d2t2_Max_LAC;
u1_d2t3[j]=d2t3_Max_LAC;
u1_d3t1[j]=d3t1_Max_LAC;
u1_d3t2[j]=d3t2_Max_LAC;
u1_d3t3[j]=d3t3_Max_LAC;
u2_x1[j]=dummy_u2_x1;
u2_x2[j]=dummy_u2_x2;
u2_x3[j]=dummy_u2_x3;
u2_t1[j]=t1_Max_LASC;
u2_t2[j]=t2_Max_LASC;
u2_t3[j]=t3_Max_LASC;
```

```
    u2_dt1[j]=dt1_Max_LASC;
    u2_dt2[j]=dt2_Max_LASC;
    u2_dt3[j]=dt3_Max_LASC;
    u2_d2t1[j]=d2t1_Max_LASC;
    u2_d2t2[j]=d2t2_Max_LASC;
    u2_d2t3[j]=d2t3_Max_LASC;
    u2_d3t1[j]=d3t1_Max_LASC;
    u2_d3t2[j]=d3t2_Max_LASC;
    u2_d3t3[j]=d3t3_Max_LASC;
    v1_x1[j]=dummy_v1_x1;
    v1_x2[j]=dummy_v1_x2;
    v1_x3[j]=dummy_v1_x3;
    v1_t1[j]=t1_Min_LAC;
    v1_t2[j]=t2_Min_LAC;
    v1_t3[j]=t3_Min_LAC;
    v1_dt1[j]=dt1_Min_LAC;
    v1_dt2[j]=dt2_Min_LAC;
    v1_dt3[j]=dt3_Min_LAC;
    v1_d2t1[j]=d2t1_Min_LAC;
    v1_d2t2[j]=d2t2_Min_LAC;
    v1_d2t3[j]=d2t3_Min_LAC;
    v1_d3t1[j]=d3t1_Min_LAC;
    v1_d3t2[j]=d3t2_Min_LAC;
    v1_d3t3[j]=d3t3_Min_LAC;
    v2_x1[j]=dummy_v2_x1;
    v2_x2[j]=dummy_v2_x2;
    v2_x3[j]=dummy_v2_x3;
    v2_t1[j]=t1_Min_LASC;
    v2_t2[j]=t2_Min_LASC;
    v2_t3[j]=t3_Min_LASC;
    v2_dt1[j]=dt1_Min_LASC;
    v2_dt2[j]=dt2_Min_LASC;
    v2_dt3[j]=dt3_Min_LASC;
    v2_d2t1[j]=d2t1_Min_LASC;
    v2_d2t2[j]=d2t2_Min_LASC;
    v2_d2t3[j]=d2t3_Min_LASC;
    v2_d3t1[j]=d3t1_Min_LASC;
    v2_d3t2[j]=d3t2_Min_LASC;
    v2_d3t3[j]=d3t3_Min_LASC;

    }

}


__global__ void fun( float * xvalue, float * yvalue,
    float * zvalue, mint idx, mint optimum, mint ListSize)
```

```
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float pu[n], pv[n];
float u1_x1[n], u1_x2[n], u1_x3[n], u1_t1[n], u1_t2[n],
    u1_t3[n], u1_dt1[n], u1_dt2[n], u1_dt3[n], u1_d2t1[n],
    u1_d2t2[n], u1_d2t3[n], u1_d3t1[n], u1_d3t2[n], u1_d3t3[n];
float u2_x1[n], u2_x2[n], u2_x3[n], u2_t1[n], u2_t2[n], u2_t3[n],
    u2_dt1[n], u2_dt2[n], u2_dt3[n], u2_d2t1[n], u2_d2t2[n],
    u2_d2t3[n], u2_d3t1[n], u2_d3t2[n], u2_d3t3[n];
float v1_x1[n], v1_x2[n], v1_x3[n], v1_t1[n], v1_t2[n], v1_t3[n],
    v1_dt1[n], v1_dt2[n], v1_dt3[n], v1_d2t1[n], v1_d2t2[n],
    v1_d2t3[n], v1_d3t1[n], v1_d3t2[n], v1_d3t3[n];
float v2_x1[n], v2_x2[n], v2_x3[n], v2_t1[n], v2_t2[n], v2_t3[n],
    v2_dt1[n], v2_dt2[n], v2_dt3[n], v2_d2t1[n], v2_d2t2[n],
    v2_d2t3[n], v2_d3t1[n], v2_d3t2[n], v2_d3t3[n];

Integration( pu, pv, u1_x1, u1_x2, u1_x3, u1_t1, u1_t2, u1_t3, u1_dt1, u1_dt2,
    u1_dt3, u1_d2t1, u1_d2t2, u1_d2t3, u1_d3t1, u1_d3t2, u1_d3t3,
    u2_x1, u2_x2, u2_x3, u2_t1, u2_t2, u2_t3, u2_dt1, u2_dt2, u2_dt3,
    u2_d2t1, u2_d2t2, u2_d2t3, u2_d3t1, u2_d3t2, u2_d3t3, v1_x1, v1_x2,
    v1_x3, v1_t1, v1_t2, v1_t3, v1_dt1, v1_dt2, v1_dt3, v1_d2t1,
    v1_d2t2, v1_d2t3, v1_d3t1, v1_d3t2, v1_d3t3, v2_x1, v2_x2,
    v2_x3, v2_t1, v2_t2, v2_t3, v2_dt1, v2_dt2, v2_dt3, v2_d2t1,
    v2_d2t2, v2_d2t3, v2_d3t1, v2_d3t2, v2_d3t3, idx, optimum);

if( index < ListSize)
{
    xvalue[index]= Coons_x1;
    yvalue[index]= Coons_x2;
    zvalue[index]= Coons_x3;
}

}


";
```

```
Needs["CUDALink`"]
```

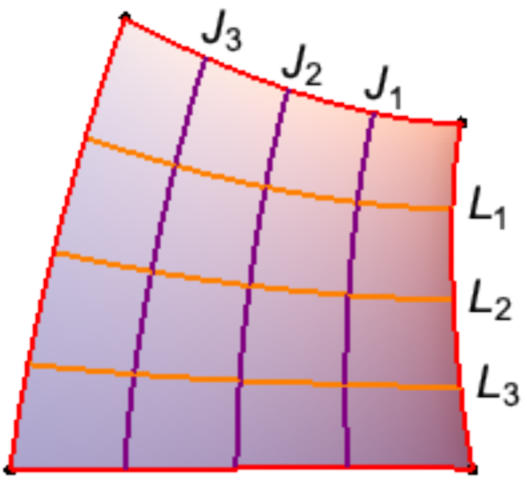
```
AbsoluteTiming[funcCS = CUDAFunctionLoad[kernelcodeCS,
   "fun", {{"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, _Integer, _Integer, _Integer}, 256]]
```

```
{102.767, CUDAFunction[<>, fun, {{Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, Integer64, Integer64, Integer64}]]}
```

```
CoordCS[idx_, optimum_, nn_] := Flatten[
   bufferCP2LAS1 = CUDAMemoryAllocate["Float", nn];
   bufferCP2LAS2 = CUDAMemoryAllocate["Float", nn];
   bufferCP2LAS3 = CUDAMemoryAllocate["Float", nn];
   resCS1 =
    funcCS[bufferCP2LAS1, bufferCP2LAS2, bufferCP2LAS3, idx, optimum, nn];
   Transpose[{CUDAMemoryGet[bufferCP2LAS1], CUDAMemoryGet[bufferCP2LAS2],
     CUDAMemoryGet[bufferCP2LAS3]}], 0];
```

Result

```
Show[Graphics3D[
  {Thick, Purple, {Line[CoordCS[250, 0, 934]], Line[CoordCS[500, 0, 1003]],
    Line[CoordCS[750, 0, 1095]]}, Thick, Orange, {Line[CoordCS[250, 1, 934]],
    Line[CoordCS[500, 1, 1003]], Line[CoordCS[750, 1, 1095]]}}],
 ParametricPlot3D[CoonsMaxLoC0[u], {u, 0, 1}, PlotStyle → Red],
 ParametricPlot3D[CoonsMaxLoC1[u], {u, 0, 1}, PlotStyle → Red],
 ParametricPlot3D[CoonsMinLoC0[v], {v, 0, 1}, PlotStyle → Red],
 ParametricPlot3D[CoonsMinLoC1[v], {v, 0, 1}, PlotStyle → Red],
 Graphics3D[{{PointSize[Large], Point[{Point00, Point03, Point30, Point33}]},
   Text[Style["L1", Large], {0.`, -0.21366455477174995`, -0.04565254196581013`} +
     {0.1, 0, 0}], Text[Style["L2", Large],
    {0.`, -0.4005526798927377`, -0.16044244936925398`} + {0.1, 0, 0}],
   Text[Style["L3", Large], {0.`, -0.5591045516980968`, -0.3125978997295298`} +
     {0.1, 0, 0}], Text[Style["J1", Large],
    {-0.21366455477174995`, 0.`, 0.04565254196581013`} + {0, .1, 0}],
   Text[Style["J2", Large], {-0.4005526798927377`, 0.`, 0.16044244936925398`} +
     {0, .1, 0}], Text[Style["J3", Large],
    {-0.5591045516980968`, 0.`, 0.3125978997295298`} + {0, 0.1, 0}]
  }], ParametricPlot3D[Coonspatch[u, v], {u, 0, 1}, {v, 0, 1},
  PlotStyle → LightPink, Axes → False, Boxed → False, Mesh → None],
 PlotRange → All, Boxed → False, Axes → False]
```



```
Show[Graphics3D[
  {Thick, Purple, {Line[CoordCS[200, 0, 923]], Line[CoordCS[400, 0, 975]],
    Line[CoordCS[600, 0, 1040]], Line[CoordCS[800, 0, 1112]]}, Thick,
   Orange, {Line[CoordCS[200, 1, 923]], Line[CoordCS[400, 1, 975]],
    Line[CoordCS[600, 1, 1040]], Line[CoordCS[800, 1, 1112]]}}],
 ParametricPlot3D[CoonsMaxLoC0[u], {u, 0, 1}, PlotStyle → Red],
 ParametricPlot3D[CoonsMaxLoC1[u], {u, 0, 1}, PlotStyle → Red],
 ParametricPlot3D[CoonsMinLoC0[v], {v, 0, 1}, PlotStyle → Red],
 ParametricPlot3D[CoonsMinLoC1[v], {v, 0, 1}, PlotStyle → Red],
 Graphics3D[{{PointSize[Large], Point[{Point00, Point03, Point30, Point33}]},
   Text[Style["L₁", Large],
    {0.`, -0.17241576313972473`, -0.0423455648124218`} + {0.1, 0, 0}],
   Text[Style["L₂", Large], {0.`, -0.32973816990852356`, -0.12606291472911835`} +
     {0.1, 0, 0}], Text[Style["L₃", Large],
    {0.`, -0.47130563855171204`, -0.23448483645915985`} + {0.1, 0, 0}],
   Text[Style["L₄", Large], {0.`, -0.5972213745117188`, -0.36080968379974365`} +
     {0.1, 0, 0}], Text[Style["J₁", Large],
    {-0.17241576313972473`, 0.`, 0.0423455648124218`} + {0, .1, 0}],
   Text[Style["J₂", Large],
    {-0.32973816990852356`, 0.`, 0.12606291472911835`} + {0, .1, 0}],
   Text[Style["J₃", Large], {-0.47130563855171204`, 0.`, 0.23448483645915985`} +
     {0, 0.1, 0}], Text[Style["J₄", Large],
    {-0.5972214937210083`, 0.`, 0.36080968379974365`} + {0, 0.1, 0}]
  }], ParametricPlot3D[Coonspatch[u, v], {u, 0, 1}, {v, 0, 1},
  PlotStyle → LightPink, Axes → False, Boxed → False, Mesh → None],
 PlotRange → All, Boxed → False, Axes → False]
```

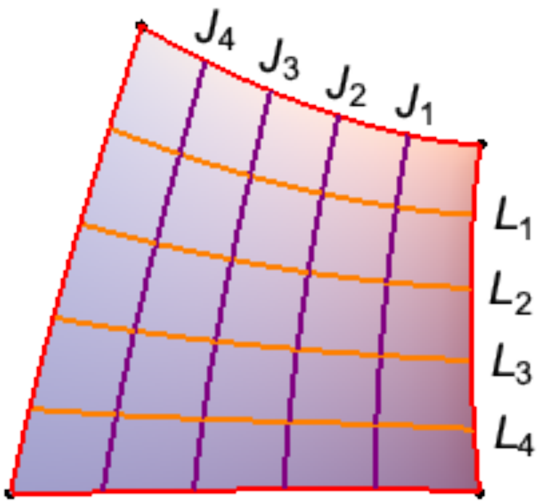


Figure 6

Hyperbolic Paraboloid

Set up

```
kernelcodeHP = "
#define min 0
#define max 1
#define uuu 0
#define vvv 1
#define sign1 1
#define h 0.001
#define n 1202
#define n_j 302


#define inidxdu1 (tu1)
#define inidxdu2 (0)
#define inidxdu3 (tu2)
#define inidxdv1 (0)
#define inidxdv2 (tv1)
#define inidxdv3 (tv2)

#define inid2xdu21 (dtu1)
#define inid2xdu22 (0)
#define inid2xdu23 (dtu2)
#define inid2xdv21 (0)
#define inid2xdv22 (dtv1)
#define inid2xdv23 (dtv2)
#define inid2xdudv1 (0)
#define inid2xdudv2 (0)
#define inid2xdudv3 (0)

#define inid3xdu31 (d2tu1)
#define inid3xdu32 (0)
#define inid3xdu33 (d2tu2)
#define inid3xdv31 (0)
#define inid3xdv32 (d2tv1)
#define inid3xdv33 (d2tv2)
#define inid3xdu2dv1 (0)
#define inid3xdu2dv2 (0)
#define inid3xdu2dv3 (0)
#define inid3xdudv21 (0)
```

```
#define inid3xdudv22 (0)
#define inid3xdudv23 (0)

#define dxdu1 (tu1[i])
#define dxdu2 (0)
#define dxdu3 (tu2[i])
#define dxdv1 (0)
#define dxdv2 (tv1[i])
#define dxdv3 (tv2[i])

#define d2xdu21 (dtu1[i])
#define d2xdu22 (0)
#define d2xdu23 (dtu2[i])
#define d2xdv21 (0)
#define d2xdv22 (dtv1[i])
#define d2xdv23 (dtv2[i])
#define d2xdudv1 (0)
#define d2xdudv2 (0)
#define d2xdudv3 (0)

#define d3xdu31 (d2tu1[i])
#define d3xdu32 (0)
#define d3xdu33 (d2tu2[i])
#define d3xdv31 (0)
#define d3xdv32 (d2tv1[i])
#define d3xdv33 (d2tv2[i])
#define d3xdu2dv1 (0)
#define d3xdu2dv2 (0)
#define d3xdu2dv3 (0)
#define d3xdudv21 (0)
#define d3xdudv22 (0)
#define d3xdudv23 (0)

#define d4xdu41 (d3tu1[i])
#define d4xdu42 (0)
#define d4xdu43 (d3tu2[i])
#define d4xdv41 (0)
#define d4xdv42 (d3tv1[i])
#define d4xdv43 (d3tv2[i])
#define d4xdu3dv1 (0)
#define d4xdu3dv2 (0)
#define d4xdu3dv3 (0)
#define d4xdudv31 (0)
#define d4xdudv32 (0)
#define d4xdudv33 (0)
#define d4xdu2dv21 (0)
#define d4xdu2dv22 (0)
```

```
#define d4xdu2dv23 (0)



#define repiniSurNorvec iniSurNorvec(xu1,xu2,tu1,tu2,xv1,xv2,tv1,tv2,normv_C)
#define repiniDSurNvDu
    iniDSurNvDu(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,dnormvdu_C)
#define repiniDSurNvDv
    iniDSurNvDv(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,dnormvdv_C)
#define repiniScalarSurNor
    iniScalarSurNor(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,Scal)
#define repinicoefE inicoefE(xu1,xu2,tu1,tu2,xv1,xv2,tv1,tv2)
#define repiniDEDu
    iniDEDu(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniDEDv
    iniDEDv(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repinicoefF inicoefF(xu1,xu2,tu1,tu2,xv1,xv2,tv1,tv2)
#define repiniDFDu
    iniDFDu(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniDFDv
    iniDFDv(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repinicoefG inicoefG(xu1,xu2,tu1,tu2,xv1,xv2,tv1,tv2)
#define repiniDGDu
    iniDGDu(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniDGDv
    iniDGDv(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repinicoefL
    inicoefL(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repinicoefM
    inicoefM(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repinicoefN
    inicoefN(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniGaussCurva
    iniGaussCurva(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniMeanCurva
    iniMeanCurva(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniMaxPrinci
    iniMaxPrinci(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniMinPrinci
    iniMinPrinci(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniPrinci
    iniPrinci(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)

#define repiniDcoefL_2
    iniDcoefL_2(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
    dtv2,d2tv1,d2tv2,CoeffL)
#define repiniDcoefM_2
```

```
    iniDcoefM_2(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
    dtv2,d2tv1,d2tv2,CoeffM)
#define repiniDcoefN_2
    iniDcoefN_2(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
    dtv2,d2tv1,d2tv2,CoeffN)
#define repiniDGaussCurva_2
    iniDGaussCurva_2(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,
    dtv1,dtv2,d2tv1,d2tv2,GCurva)
#define repiniDMeanCurva_2
    iniDMeanCurva_2(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,
    dtv1,dtv2,d2tv1,d2tv2,MCurva)
#define repiniDPrinci_2
    iniDPrinci_2(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
    dtv2,d2tv1,d2tv2,optimum,PCurva)
#define repiniEta
    iniEta(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniMiu
    iniMiu(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDuDs
    iniDuDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDvDs
    iniDvDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDEDs
    iniDEDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDFDs
    iniDFDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDGDs
    iniDGDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDLDs
    iniDLDs(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2,optimum)
#define repiniDMDs
    iniDMDs(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2,optimum)
#define repiniDNDs
    iniDNDs(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2,optimum)
#define repiniDPrinciDs
    iniDPrinciDs(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
    dtv2,d2tv1,d2tv2,optimum)
#define repiniDvDt
    iniDvDt(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDuDt
    iniDuDt(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDSurDs
    iniDSurDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum,tangv)
#define repiniSurNor iniSurNor(xu1,xu2,tu1,tu2,xv1,xv2,tv1,tv2,Snor_C)
```

```
#define repiniBinormalSur
    iniBinormalSur(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    optimum,binormv_C)
#define repiniAlp1
    iniAlp1(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum,Alp1_C)
#define repiniBeta1
    iniBeta1(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2
    ,d2tv1,d2tv2,optimum)

#define repSurNorvec
    SurNorvec(i,xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,normv_C)
#define repDSurNvDu
    DSurNvDu(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
    dtv2,d2tv1,d2tv2,dnormvdu_C)
#define repDSurNvDv
    DSurNvDv(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
    dtv2,d2tv1,d2tv2,dnormvdv_C)
#define repD2SurNvDu2
    D2SurNvDu2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,
    tv1,tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,d2normvdu2_C)
#define repD2SurNvDuDv
    D2SurNvDuDv(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,
    tv1,tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,d2normvdudv_C)
#define repD2SurNvDv2
    D2SurNvDv2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,
    tv1,tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,d2normvdv2_C)
#define repScalarSurNor
    ScalarSurNor(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2
    ,tv1,tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,Scal)
#define repcoefE
    coefE(i,xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repDEDu
    DEDu(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2)
#define repDEDv
    DEDv(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2)
#define repD2EDu2
    D2EDu2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1,
    tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2)
#define repD2EDuDv
    D2EDuDv(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1,
    tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2)
#define repD2EDv2
    D2EDv2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1,
    tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2)
#define repcoefF
```

```
    coefF(i,xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repDFDu
    DFDu(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2)
#define repDFDv
    DFDv(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2)
#define repD2FDu2
    D2FDu2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1,
    tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2)
#define repD2FDuDv
    D2FDuDv(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1,
    tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2)
#define repD2FDv2
    D2FDv2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1,
    tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2)
#define repcoefG
    coefG(i,xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repDGDu
    DGDu(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2)
#define repDGDv
    DGDv(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2)
#define repD2GDu2
    D2GDu2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1,
    tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2)
#define repD2GDuDv
    D2GDuDv(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1,
    tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2)
#define repD2GDv2
    D2GDv2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1,
    tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2)
#define repcoefL
    coefL(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2)
#define repcoefM
    coefM(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2)
#define repcoefN
    coefN(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2)
#define repGaussCurva
    GaussCurva(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
    dtv2,d2tv1,d2tv2)
#define repMeanCurva
    MeanCurva(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
```

```
    dtv2,d2tv1,d2tv2)
#define repMaxPrinci
    MaxPrinci(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
    dtv2,d2tv1,d2tv2)
#define repMinPrinci
    MinPrinci(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
    dtv2,d2tv1,d2tv2)
#define repPrinci
    Princi(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2
    ,d2tv1,d2tv2,optimum)
#define repDcoefL_2
    DcoefL_2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1
    ,tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,CoeffL)
#define repDcoefM_2
    DcoefM_2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1
    ,tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,CoeffM)
#define repDcoefN_2
    DcoefN_2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1
    ,tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,CoeffN)
#define repDGaussCurva_2
    DGaussCurva_2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,
    xv2,tv1,tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,GCurva)
#define repDMeanCurva_2
    DMeanCurva_2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2
    ,tv1,tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,MCurva)
#define repDPrinci_2
    DPrinci_2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,
    tv1,tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,optimum,PCurva)
#define repEta
    Eta(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2,optimum)
#define repMiu
    Miu(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2,optimum)
#define repDuDs
    DuDs(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2,optimum)
#define repDvDs
    DvDs(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2,optimum)
#define repDEDs
    DEDs(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2,optimum)
#define repDFDs
    DFDs(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2,optimum)
#define repDGDs
```

```
    DGDs(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2,optimum)
#define repDLDs
    DLDs(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1,tv2
    ,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,optimum)
#define repDMDs
    DMDs(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1,tv2
    ,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,optimum)
#define repDNDs
    DNDs(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1,tv2
    ,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,optimum)
#define repDPrinciDs
    DPrinciDs(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,
    tv1,tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,optimum)
#define repDvDt
    DvDt(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2,optimum)
#define repDuDt
    DuDt(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2,optimum)
#define repDSurDs
    DSurDs(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2
    ,d2tv1,d2tv2,optimum,tangv)

#define repCurvecurvature
    Curvecurvature(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,
    xv2,tv1,tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,D2uDs2,D2vDs2,GeoCur,
    optimum)
#define repD2EDs2
    D2EDs2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1,
    tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,D2uDs2,D2vDs2,GeoCur,optimum)
#define repD2FDs2
    D2FDs2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1,
    tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,D2uDs2,D2vDs2,GeoCur,optimum)
#define repD2GDs2
    D2GDs2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1,
    tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,D2uDs2,D2vDs2,GeoCur,optimum)
#define repD2LDs2
    D2LDs2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1,
    tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,D2uDs2,D2vDs2,GeoCur,optimum)
#define repD2MDs2
    D2MDs2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1,
    tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,D2uDs2,D2vDs2,GeoCur,optimum)
#define repD2NDs2
    D2NDs2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1,
    tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,D2uDs2,D2vDs2,GeoCur,optimum)
#define repD2PrinciDs2
```

```
    D2PrinciDs2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,
    tv1,tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,D2uDs2,D2vDs2,GeoCur,optimum)
#define repD2SurDs2
    D2SurDs2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1
    ,tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,D2uDs2,D2vDs2,GeoCur,optimum,
    dtangds)
#define repCurvenormaloS
    CurvenormaloS(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,
    xv2,tv1,tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,D2uDs2,D2vDs2,GeoCur,
    optimum,norv)
#define repCurvebinormaloS
    CurvebinormaloS(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,
    xv2,tv1,tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,D2uDs2,D2vDs2,GeoCur,
    optimum,binorv)
#define repAlp2
    Alp2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1,tv2
    ,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,D2uDs2,D2vDs2,GeoCur,optimum,Alp2_C)
#define repBeta2
    Beta2(i,xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,d3tu1,d3tu2,xv1,xv2,tv1,
    tv2,dtv1,dtv2,d2tv1,d2tv2,d3tv1,d3tv2,D2uDs2,D2vDs2,GeoCur,optimum)
```

```
#define opt1 (float xu1,float xu2,float tu1,float
    tu2,float xv1,float xv2,float tv1,float tv2,float normv_C[])
#define opt2 (float xu1,float xu2,float tu1,float tu2,float
    xv1,float xv2,float tv1,float tv2)
#define opt3 (float xu1,float xu2,float tu1,float tu2,float dtu1,float
    dtu2,float xv1,float xv2,float tv1,float tv2,float dtv1,float dtv2)
#define opt4 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float xv1,float xv2,float
    tv1,float tv2,float dtv1,float dtv2,mint optimum)
 #define opt5 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float xv1,float xv2,float tv1,float
    tv2,float dtv1,float dtv2,mint optimum,float tangv_C[])
#define opt6 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float xv1,float xv2,float tv1,float
    tv2,float dtv1,float dtv2,float dnormv_C[])
#define opt7 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float xv1,float xv2,float
    tv1,float tv2,float dtv1,float dtv2,float Scal[])
#define opt8 (float xu1,float xu2,float tu1,float tu2,float dtu1,float
    dtu2,float d2tu1,float d2tu2,float xv1,float xv2,float tv1,float
    tv2,float dtv1,float dtv2,float d2tv1,float d2tv2,float coeff[])
#define opt9 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float d2tu1,float d2tu2,float xv1,float
```

```
    xv2,float tv1,float tv2,float dtv1,float dtv2,float
    d2tv1,float d2tv2,mint optimum,float coeff[])
#define opt10 (float xu1,float xu2,float tu1,float tu2,float dtu1,float
    dtu2,float d2tu1,float d2tu2,float xv1,float xv2,float tv1,float
    tv2,float dtv1,float dtv2,float d2tv1,float d2tv2,mint optimum)
#define opt11 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float xv1,float xv2,float tv1,float
    tv2,float dtv1,float dtv2,mint optimum,float Alph_C[])
#define opt12 (float xu1,float xu2,float tu1,float tu2,float
    xv1,float xv2,float tv1,float tv2,float Snor_C[])
#define opt13 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float xv1,float xv2,float tv1,float
    tv2,float dtv1,float dtv2,mint optimum,float binormv_C[])
#define opt14 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float d2tu1,float d2tu2,float
    xv1,float xv2,float tv1,float tv2,float dtv1,float
    dtv2,float d2tv1,float d2tv2,mint optimum,float x[])

#define opt15 (mint i,float xu1[],float xu2[],float tu1[],float
    tu2[],float dtu1[],float dtu2[],float xv1[],float xv2[],float
    tv1[],float tv2[],float dtv1[],float dtv2[],float normv_C[])
#define opt16 (mint i,float xu1[],float xu2[],float tu1[],float
    tu2[],float dtu1[],float dtu2[],float xv1[],float
    xv2[],float tv1[],float tv2[],float dtv1[],float dtv2[])
#define opt17 (mint i,float xu1[],float xu2[],float tu1[],float
    tu2[],float dtu1[],float dtu2[],float d2tu1[],float
    d2tu2[],float xv1[],float xv2[],float tv1[],float
    tv2[],float dtv1[],float dtv2[],float d2tv1[],float d2tv2[])
#define opt18 (mint i,float xu1[],float xu2[],float tu1[],float
    tu2[],float dtu1[],float dtu2[],float d2tu1[],float
    d2tu2[],float xv1[],float xv2[],float tv1[],float tv2[],float
    dtv1[],float dtv2[],float d2tv1[],float d2tv2[],mint optimum)
#define opt19 (mint i,float xu1[],float xu2[],float tu1[],float
    tu2[],float dtu1[],float dtu2[],float d2tu1[],float d2tu2[],float
    xv1[],float xv2[],float tv1[],float tv2[],float dtv1[],float
    dtv2[],float d2tv1[],float d2tv2[],mint optimum,float tangv_C[])
#define opt20 (mint i,float xu1[],float xu2[],float tu1[],float
    tu2[],float dtu1[],float dtu2[],float d2tu1[],float d2tu2[],float
    xv1[],float xv2[],float tv1[],float tv2[],float dtv1[],float
    dtv2[],float d2tv1[],float d2tv2[],float dnormv_C[])
#define opt21 (mint i,float xu1[],float xu2[],float tu1[],float
    tu2[],float dtu1[],float dtu2[],float d2tu1[],float
    d2tu2[],float d3tu1[],float d3tu2[],float xv1[],float xv2[],float
    tv1[],float tv2[],float dtv1[],float dtv2[],float d2tv1[],float
    d2tv2[],float d3tv1[],float d3tv2[],float d2normv_C[])
#define opt22 (mint i,float xu1[],float xu2[],float tu1[],float
    tu2[],float dtu1[],float dtu2[],float d2tu1[],float
```

```
    d2tu2[],float d3tu1[],float d3tu2[],float xv1[],float
    xv2[],float tv1[],float tv2[],float dtv1[],float dtv2[],float
    d2tv1[],float d2tv2[],float d3tv1[],float d3tv2[],float Scal[])
#define opt23 (mint i,float xu1[],float xu2[],float tu1[],float
    tu2[],float dtu1[],float dtu2[],float d2tu1[],float
    d2tu2[],float d3tu1[],float d3tu2[],float xv1[],float
    xv2[],float tv1[],float tv2[],float dtv1[],float dtv2[],float
    d2tv1[],float d2tv2[],float d3tv1[],float d3tv2[])
#define opt24 (mint i,float xu1[],float xu2[],float tu1[],float
    tu2[],float dtu1[],float dtu2[],float d2tu1[],float
    d2tu2[],float d3tu1[],float d3tu2[],float xv1[],float
    xv2[],float tv1[],float tv2[],float dtv1[],float dtv2[],float
    d2tv1[],float d2tv2[],float d3tv1[],float d3tv2[],float coeff[])
#define opt25 (mint i,float xu1[],float xu2[],float tu1[],float
    tu2[],float dtu1[],float dtu2[],float d2tu1[],float
    d2tu2[],float d3tu1[],float d3tu2[],float xv1[],float xv2[],float
    tv1[],float tv2[],float dtv1[],float dtv2[],float d2tv1[],float
    d2tv2[],float d3tv1[],float d3tv2[],mint optimum,float coeff[])
#define opt26 (mint i,float xu1[],float xu2[],float tu1[],float
    tu2[],float dtu1[],float dtu2[],float d2tu1[],float
    d2tu2[],float d3tu1[],float d3tu2[],float xv1[],float
    xv2[],float tv1[],float tv2[],float dtv1[],float dtv2[],float
    d2tv1[],float d2tv2[],float d3tv1[],float d3tv2[],mint optimum)
#define opt27 (mint i,float xu1[],float xu2[],float tu1[],float tu2[],float
    dtu1[],float dtu2[],float d2tu1[],float d2tu2[],float d3tu1[],float
    d3tu2[],float xv1[],float xv2[],float tv1[],float tv2[],float
    dtv1[],float dtv2[],float d2tv1[],float d2tv2[],float d3tv1[],float
    d3tv2[],float D2uDs2[],float D2vDs2[],float GeoCur[],mint optimum)
#define opt28 (mint i,float xu1[],float xu2[],float tu1[],float
    tu2[],float dtu1[],float dtu2[],float d2tu1[],float
    d2tu2[],float d3tu1[],float d3tu2[],float xv1[],float
    xv2[],float tv1[],float tv2[],float dtv1[],float dtv2[],float
    d2tv1[],float d2tv2[],float d3tv1[],float d3tv2[],float
    D2uDs2[],float D2vDs2[],float GeoCur[],mint optimum,float coeff[])
#define opt29 ( float * pvalue, float xu1[], float xu2[], float tu1[],
    float tu2[], float dtu1[], float dtu2[], float d2tu1[], float
    d2tu2[], float d3tu1[], float d3tu2[], float xv1[], float xv2[],
    float tv1[], float tv2[], float dtv1[], float dtv2[], float d2tv1[],
    float d2tv2[], float d3tv1[], float d3tv2[], float D2uDs2[],
    float D2vDs2[], float GeoCur[], mint optimum, mint ListSize)
#define opt30 ( index, xu1, xu2, tu1, tu2, dtu1, dtu2, d2tu1,
    d2tu2, d3tu1, d3tu2, xv1, xv2, tv1, tv2, dtv1, dtv2, d2tv1,
    d2tv2, d3tv1, d3tv2, D2uDs2, D2vDs2, GeoCur, optimum)

__device__ void iniSurNorvec opt1
{
    normv_C[0] =inidxdu2*inidxdv3-inidxdu3*inidxdv2;
```

```
    normv_C[1] =inidxdu3*inidxdv1-inidxdu1*inidxdv3;
    normv_C[2] =inidxdu1*inidxdv2-inidxdu2*inidxdv1;
}
__device__ void iniDSurNvDu opt6
{
    dnormv_C[0]
     =(inid2xdu22*inidxdv3-inid2xdu23*inidxdv2)+(inidxdu2*inid2xdudv3-
     inidxdu3*inid2xdudv2);
    dnormv_C[1]
     =(inid2xdu23*inidxdv1-inid2xdu21*inidxdv3)+(inidxdu3*inid2xdudv1-
     inidxdu1*inid2xdudv3);
    dnormv_C[2]
     =(inid2xdu21*inidxdv2-inid2xdu22*inidxdv1)+(inidxdu1*inid2xdudv2-
     inidxdu2*inid2xdudv1);
}
__device__ void iniDSurNvDv opt6
{
    dnormv_C[0]
     =(inid2xdudv2*inidxdv3-inid2xdudv3*inidxdv2)+(inidxdu2*inid2xdv23-
     inidxdu3*inid2xdv22);
    dnormv_C[1]
     =(inid2xdudv3*inidxdv1-inid2xdudv1*inidxdv3)+(inidxdu3*inid2xdv21-
     inidxdu1*inid2xdv23);
    dnormv_C[2]
     =(inid2xdudv1*inidxdv2-inid2xdudv2*inidxdv1)+(inidxdu1*inid2xdv22-
     inidxdu2*inid2xdv21);
}

__device__ void iniScalarSurNor opt7
{
float ScalSNor,DScalSNorDu,DScalSNorDv;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];

repiniSurNorvec;
repiniDSurNvDu;
repiniDSurNvDv;


ScalSNor =
     sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
DScalSNorDu
     =
     (normv_C[0]*dnormvdu_C[0]+normv_C[1]*dnormvdu_C[1]+normv_C[2]*dnormvdu_C
     [2])/ScalSNor;
DScalSNorDv =
```

```
    (normv_C[0]*dnormvdv_C[0]+normv_C[1]*dnormvdv_C[1]+normv_C[2]*dnormvdv_C
    [2])/ScalSNor;


Scal[0] = ScalSNor;
Scal[1] = DScalSNorDu;
Scal[2] = DScalSNorDv;
}

__device__ float inicoefE opt2
{
    return inidxdu1*inidxdu1+inidxdu2*inidxdu2+inidxdu3*inidxdu3;
}
__device__ float iniDEDu opt3
{
    return 2*(inidxdu1*inid2xdu21+inidxdu2*inid2xdu22+inidxdu3*inid2xdu23);
}
__device__ float iniDEDv opt3
{
    return 2*(inidxdu1*inid2xdudv1+inidxdu2*inid2xdudv2+inidxdu3*inid2xdudv3);
}


__device__ float inicoefF opt2
{
    return (inidxdu1*inidxdv1+inidxdu2*inidxdv2+inidxdu3*inidxdv3);
}
__device__ float iniDFDu opt3
{
    return inidxdv1*inid2xdu21+inidxdv2*inid2xdu22+inidxdv3*inid2xdu23
    + inidxdu1*inid2xdudv1+inidxdu2*inid2xdudv2+inidxdu3*inid2xdudv3;
}
__device__ float iniDFDv opt3
{
    return
    inidxdv1*inid2xdudv1+inidxdv2*inid2xdudv2+inidxdv3*inid2xdudv3
    + inidxdu1*inid2xdv21+inidxdu2*inid2xdv22+inidxdu3*inid2xdv23;
}

__device__ float inicoefG opt2
{
    return inidxdv1*inidxdv1+inidxdv2*inidxdv2+inidxdv3*inidxdv3;
}
__device__ float iniDGDu opt3
{
    return
    2*(inidxdv1*inid2xdudv1+inidxdv2*inid2xdudv2+inidxdv3*inid2xdudv3);
```

```
}
__device__ float iniDGDv opt3
{
    return
    2*(inidxdv1*inid2xdv21+inidxdv2*inid2xdv22+inidxdv3*inid2xdv23);
}

__device__ float inicoefL opt3
{
float normv_C[3];
repiniSurNorvec;

    return
    (normv_C[0]*inid2xdu21+normv_C[1]*inid2xdu22+normv_C[2]*inid2xdu23)/sqrt
    (normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float inicoefM opt3
{
float normv_C[3];
repiniSurNorvec;

    return
    (normv_C[0]*inid2xdudv1+normv_C[1]*inid2xdudv2+normv_C[2]*inid2xdudv3)/
    sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float inicoefN opt3
{
float normv_C[3];
repiniSurNorvec;

    return
    (normv_C[0]*inid2xdv21+normv_C[1]*inid2xdv22+normv_C[2]*inid2xdv23)/sqrt
    (normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float iniGaussCurva opt3
{
    return
    (repinicoefL*repinicoefN-pow(repinicoefM,2))/(repinicoefE*repinicoefG-
    pow(repinicoefF,2));
}
__device__ float iniMeanCurva opt3
{
    return
    0.5*(repinicoefE*repinicoefN+repinicoefG*repinicoefL-2*repinicoefF*
    repinicoefM)/(repinicoefE*repinicoefG-pow(repinicoefF,2));
}
__device__ float iniMaxPrinci opt3
```

```
{
    return repiniMeanCurva+sqrt(pow(repiniMeanCurva,2)-repiniGaussCurva);
}

__device__ float iniMinPrinci opt3
{
    return repiniMeanCurva-sqrt(pow(repiniMeanCurva,2)-repiniGaussCurva);
}
__device__ float iniPrinci opt4
{
    if(optimum==max)
    {return repiniMaxPrinci;}
    else if(optimum==min)
    {return repiniMinPrinci;}
    else
    {return 0;}
}
__device__ void iniDcoefL_2 opt8
{

float CoeL,DCoeLDu,DCoeLDv;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float Scal[3];

repiniSurNorvec;
repiniDSurNvDu;
repiniDSurNvDv;
repiniScalarSurNor;

CoeL =
    (normv_C[0]*inid2xdu21+normv_C[1]*inid2xdu22+normv_C[2]*inid2xdu23)/Scal[0];
DCoeLDu
    =
    ((dnormvdu_C[0]*inid2xdu21+dnormvdu_C[1]*inid2xdu22+dnormvdu_C[2]*
    inid2xdu23)+(normv_C[0]*inid3xdu31+normv_C[1]*inid3xdu32+normv_C[2]*
    inid3xdu33)-Scal[1]*CoeL)/Scal[0];
DCoeLDv =
    ((dnormvdv_C[0]*inid2xdu21+dnormvdv_C[1]*inid2xdu22+dnormvdv_C[2]*
    inid2xdu23)+(normv_C[0]*inid3xdu2dv1+normv_C[1]*inid3xdu2dv2+normv_C[2]*
    inid3xdu2dv3)-Scal[2]*CoeL)/Scal[0];

coeff[0] = CoeL;
coeff[1] = DCoeLDu;
coeff[2] = DCoeLDv;
```

```
}
__device__ void iniDcoefM_2 opt8
{
float CoeM,DCoeMDu,DCoeMDv;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float Scal[3];
repiniSurNorvec;
repiniDSurNvDu;
repiniDSurNvDv;
repiniScalarSurNor;

CoeM =
    (normv_C[0]*inid2xdudv1+normv_C[1]*inid2xdudv2+normv_C[2]*inid2xdudv3)/
    Scal[0];
DCoeMDu =
    ((dnormvdu_C[0]*inid2xdudv1+dnormvdu_C[1]*inid2xdudv2+dnormvdu_C[2]*
    inid2xdudv3)+(normv_C[0]*inid3xdu2dv1+normv_C[1]*inid3xdu2dv2+normv_C[2]
    *inid3xdu2dv3)-Scal[1]*CoeM)/Scal[0];
DCoeMDv =
    ((dnormvdv_C[0]*inid2xdudv1+dnormvdv_C[1]*inid2xdudv2+dnormvdv_C[2]*
    inid2xdudv3)+(normv_C[0]*inid3xdudv21+normv_C[1]*inid3xdudv22+normv_C[2]
    *inid3xdudv23)-Scal[2]*CoeM)/Scal[0];


coeff[0] = CoeM;
coeff[1] = DCoeMDu;
coeff[2] = DCoeMDv;
}
__device__ void iniDcoefN_2 opt8
{
float CoeN,DCoeNDu,DCoeNDv;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float Scal[3];
repiniSurNorvec;
repiniDSurNvDu;
repiniDSurNvDv;
repiniScalarSurNor;

CoeN =
    (normv_C[0]*inid2xdv21+normv_C[1]*inid2xdv22+normv_C[2]*inid2xdv23)/Scal[0];
DCoeNDu
    =
    ((dnormvdu_C[0]*inid2xdv21+dnormvdu_C[1]*inid2xdv22+dnormvdu_C[2]*
```

```
        inid2xdv23)+(normv_C[0]*inid3xdudv21+normv_C[1]*inid3xdudv22+normv_C[2]*
        inid3xdudv23)-Scal[1]*CoeN)/Scal[0];
DCoeNDv =
        ((dnormvdv_C[0]*inid2xdv21+dnormvdv_C[1]*inid2xdv22+dnormvdv_C[2]*
        inid2xdv23)+(normv_C[0]*inid3xdv31+normv_C[1]*inid3xdv32+normv_C[2]*
        inid3xdv33)-Scal[2]*CoeN)/Scal[0];

coeff[0] = CoeN;
coeff[1] = DCoeNDu;
coeff[2] = DCoeNDv;

}
__device__ void iniDGaussCurva_2 opt8
{
float GCurv,DGCurvDu,DGCurvDv;
float CoeffL[3];
float CoeffM[3];
float CoeffN[3];
repiniDcoefL_2;
repiniDcoefM_2;
repiniDcoefN_2;

GCurv=
        (CoeffL[0]*CoeffN[0]-pow(CoeffM[0],2))/(repinicoefE*repinicoefG-pow(
        repinicoefF,2));
DGCurvDu =
        ((pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repinicoefG*repiniDEDu-2*
        repinicoefF*repiniDFDu+repinicoefE*repiniDGDu)+(-pow(repinicoefF,2)+
        repinicoefE*repinicoefG)*(CoeffN[0]*CoeffL[1]-2*CoeffM[0]*CoeffM[1]+
        CoeffL[0]*CoeffN[1]))/(pow(pow(repinicoefF,2)-repinicoefE*repinicoefG,2)
        );
DGCurvDv
        =((pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repinicoefG*repiniDEDv-2*
        repinicoefF*repiniDFDv+repinicoefE*repiniDGDv)+(-pow(repinicoefF,2)+
        repinicoefE*repinicoefG)*(CoeffN[0]*CoeffL[2]-2*CoeffM[0]*CoeffM[2]+
        CoeffL[0]*CoeffN[2]))/(pow(pow(repinicoefF,2)-repinicoefE*repinicoefG,2)
        );


coeff[0] = GCurv;
coeff[1] = DGCurvDu;
coeff[2] = DGCurvDv;

}
__device__ void iniDMeanCurva_2 opt8
{
float MCurv,DMCurvDu,DMCurvDv;
```

```
float CoeffL[3];
float CoeffM[3];
float CoeffN[3];
repiniDcoefL_2;
repiniDcoefM_2;
repiniDcoefN_2;

MCurv =
    0.5*(repinicoefE*CoeffN[0]+repinicoefG*CoeffL[0]-2*repinicoefF*CoeffM[0]
    )/(repinicoefE*repinicoefG-pow(repinicoefF,2));
DMCurvDu =
    0.5*(-(repinicoefG*CoeffL[0]-2*repinicoefF*CoeffM[0]+repinicoefE*CoeffN[
    0])*(repinicoefG*repiniDEDu-2*repinicoefF*repiniDFDu+repinicoefE*
    repiniDGDu)+(-pow(repinicoefF,2)+repinicoefE*repinicoefG)*(CoeffN[0]*
    repiniDEDu-2*CoeffM[0]*repiniDFDu+CoeffL[0]*repiniDGDu+repinicoefG*
    CoeffL[1]-2*repinicoefF*CoeffM[1]+repinicoefE*CoeffN[1]))/(pow(pow(
    repinicoefF,2)-repinicoefE*repinicoefG,2));
DMCurvDv =
    0.5*(-(repinicoefG*CoeffL[0]-2*repinicoefF*CoeffM[0]+repinicoefE*CoeffN[
    0])*(repinicoefG*repiniDEDv-2*repinicoefF*repiniDFDv+repinicoefE*
    repiniDGDv)+(-pow(repinicoefF,2)+repinicoefE*repinicoefG)*(CoeffN[0]*
    repiniDEDv-2*CoeffM[0]*repiniDFDv+CoeffL[0]*repiniDGDv+repinicoefG*
    CoeffL[2]-2*repinicoefF*CoeffM[2]+repinicoefE*CoeffN[2]))/(pow(pow(
    repinicoefF,2)-repinicoefE*repinicoefG,2));



coeff[0] = MCurv;
coeff[1] = DMCurvDu;
coeff[2] = DMCurvDv;
}
__device__ void iniDPrinci_2 opt9
{
float Princip,DPrincipDu,DPrincipDv;
float GCurva[3];
float MCurva[3];
repiniDGaussCurva_2;
repiniDMeanCurva_2;

if(optimum==max)
{Princip = MCurva[0]+sqrt(pow(MCurva[0],2)-GCurva[0]);
DPrincipDu =
    MCurva[1]+(2*MCurva[0]*MCurva[1]-GCurva[1])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
DPrincipDv =
    MCurva[2]+(2*MCurva[0]*MCurva[2]-GCurva[2])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
```

```
}
else if(optimum==min)
{Princip = MCurva[0]-sqrt(pow(MCurva[0],2)-GCurva[0]);
DPrincipDu =
    MCurva[1]-(2*MCurva[0]*MCurva[1]-GCurva[1])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
DPrincipDv =
    MCurva[2]-(2*MCurva[0]*MCurva[2]-GCurva[2])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));

}
else
{Princip = 0;
DPrincipDu = 0;
DPrincipDv = 0;
}

coeff[0] = Princip;
coeff[1] = DPrincipDu;
coeff[2] = DPrincipDv;

}

__device__ float iniEta opt4
{
    return
    sign1/sqrt(repinicoefE*pow(repinicoefM-repiniPrinci*repinicoefF,2)-2*
    repinicoefF*(repinicoefM-repiniPrinci*repinicoefF)*(repinicoefL-
    repiniPrinci*repinicoefE)+repinicoefG*pow(repinicoefL-repiniPrinci*
    repinicoefE,2));
}
__device__ float iniMiu opt4
{
    return
    sign1/sqrt(repinicoefE*pow(repinicoefN-repiniPrinci*repinicoefG,2)-2*
    repinicoefF*(repinicoefM-repiniPrinci*repinicoefF)*(repinicoefN-
    repiniPrinci*repinicoefG)+repinicoefG*pow(repinicoefM-repiniPrinci*
    repinicoefF,2));
}
__device__ float iniDuDs opt4
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-
    repiniPrinci*repinicoefG))
    {return repiniEta*(repinicoefM-repiniPrinci*repinicoefF);}
else
    {return repiniMiu*(repinicoefN-repiniPrinci*repinicoefG);}
```

```
}
__device__ float iniDvDs opt4
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
     repinicoefG))
    {return -repiniEta*(repinicoefL-repiniPrinci*repinicoefE);}
else
    {return -repiniMiu*(repinicoefM-repiniPrinci*repinicoefF);}
}


__device__ float iniDEDs opt4
{
    return repiniDEDu*repiniDuDs+repiniDEDv*repiniDvDs;
}
__device__ float iniDFDs opt4
{
    return repiniDFDu*repiniDuDs+repiniDFDv*repiniDvDs;
}
__device__ float iniDGDs opt4
{
    return repiniDGDu*repiniDuDs+repiniDGDv*repiniDvDs;
}
__device__ float iniDLDs opt10
{
float CoeffL[3];
repiniDcoefL_2;

    return CoeffL[1]*repiniDuDs+CoeffL[2]*repiniDvDs;
}
__device__ float iniDMDs opt10
{
float CoeffM[3];
repiniDcoefM_2;

    return CoeffM[1]*repiniDuDs+CoeffM[2]*repiniDvDs;
}
__device__ float iniDNDs opt10
{
float CoeffN[3];
repiniDcoefN_2;

    return CoeffN[1]*repiniDuDs+CoeffN[2]*repiniDvDs;
}
__device__ float iniDPrinciDs opt10
{
float PCurva[3];
```

```
repiniDPrinci_2;

    return PCurva[1]*repiniDuDs+PCurva[2]*repiniDvDs;
}
__device__ float iniDuDt opt4
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
    repinicoefG))
    {return (repinicoefM-repiniPrinci*repinicoefF);}
else
    {return (repinicoefN-repiniPrinci*repinicoefG);}
}
__device__ float iniDvDt opt4
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
    repinicoefG))
    {return (repinicoefL-repiniPrinci*repinicoefE);}
else
    {return (repinicoefM-repiniPrinci*repinicoefF);}
}
__device__ void iniDSurDs opt5
{
    tangv_C[0] =inidxdu1*repiniDuDs+inidxdv1*repiniDvDs;
    tangv_C[1] =inidxdu2*repiniDuDs+inidxdv2*repiniDvDs;
    tangv_C[2] =inidxdu3*repiniDuDs+inidxdv3*repiniDvDs;
}

__device__ void iniAlp1 opt11
{
    Alph_C[0]=(inid2xdu21*pow(repiniDuDs,2)+2*inid2xdudv1*repiniDuDs*
    repiniDvDs+inid2xdv21*pow(repiniDvDs,2));

    Alph_C[1]=(inid2xdu22*pow(repiniDuDs,2)+2*inid2xdudv2*repiniDuDs*
    repiniDvDs+inid2xdv22*pow(repiniDvDs,2));

    Alph_C[2]=(inid2xdu23*pow(repiniDuDs,2)+2*inid2xdudv3*repiniDuDs*
    repiniDvDs+inid2xdv23*pow(repiniDvDs,2));
}
__device__ void iniSurNor opt12
{
    float normv_C[3];
    repiniSurNorvec;
    Snor_C[0]
    =normv_C[0]/sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*
    normv_C[2]);
    Snor_C[1]
    =normv_C[1]/sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*
```

```
    normv_C[2]);
    Snor_C[2]
    =normv_C[2]/sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*
    normv_C[2]);
}

__device__ void iniBinormalSur opt13
{
    float Snor_C[3];
    float tangv[3];
    repiniSurNor;
    repiniDSurDs;

    binormv_C[0] =Snor_C[1]*tangv[2] -Snor_C[2]*tangv[1];
    binormv_C[1] =Snor_C[2]*tangv[0] -Snor_C[0]*tangv[2];
    binormv_C[2] =Snor_C[0]*tangv[1] -Snor_C[1]*tangv[0];
}
__device__ float iniBeta1 opt10
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
    repinicoefG))
    {return
    (-(repiniDLDs-repiniDPrinciDs*repinicoefE-repiniPrinci*repiniDEDs)*
    repiniDuDs-(repiniDMDs-repiniDPrinciDs*repinicoefF-repiniPrinci*
    repiniDFDs)*repiniDvDs);}
else
    {return
    (-(repiniDMDs-repiniDPrinciDs*repinicoefF-repiniPrinci*repiniDFDs)*
    repiniDuDs-(repiniDNDs-repiniDPrinciDs*repinicoefG-repiniPrinci*
    repiniDGDs)*repiniDvDs);}
}

__device__ void LUDecomposition3x3 opt14
{
    mint rc=3;
    float binormv_C[3];
    float Alp1_C[3];
    repiniBinormalSur;
    repiniAlp1;
    float matrixA[3][3]={

    {repinicoefE,repinicoefF,-(binormv_C[0]*inidxdu1+binormv_C[1]*inidxdu2+
    binormv_C[2]*inidxdu3)},

    {repinicoefF,repinicoefG,-(binormv_C[0]*inidxdv1+binormv_C[1]*inidxdv2+
    binormv_C[2]*inidxdv3)},
    {repiniDvDt,repiniDuDt,0}};
```

```
float
matrixB[3]={-(Alp1_C[0]*inidxdu1+Alp1_C[1]*inidxdu2+Alp1_C[2]*inidxdu3),
-(Alp1_C[0]*inidxdv1+Alp1_C[1]*inidxdv2+Alp1_C[2]*inidxdv3),repiniBeta1}
;
float Lower[3][3];
float Upper[3][3];
float y[3];
float sum;
for(mint ii=0; ii<rc; ii++)
        {
            for(mint jj=0; jj<rc; jj++)
            {
                sum = 0;
                if(ii==jj)
                {
                Lower[ii][jj]=1;
                for (mint kk = 0; kk < ii; kk++)
                    {sum += Lower[ii][kk] * Upper[kk][jj];}
                Upper[ii][jj] = matrixA[ii][jj] - sum;
                }
                else if(ii < jj)
                {
                Lower[ii][jj]=0;
                for (mint kk = 0; kk < ii; kk++)
                    {sum += Lower[ii][kk] * Upper[kk][jj];}
                Upper[ii][jj] = matrixA[ii][jj] - sum;
                }
                else
                {
                Upper[ii][jj]=0;
                for (mint kk = 0; kk < jj; kk++)
                    {sum += Lower[ii][kk] * Upper[kk][jj];}
                Lower[ii][jj] =(matrixA[ii][jj] - sum)/Upper[jj][jj];
                }
            }
        }
for (mint ii = 0; ii < rc; ii++)
{
    sum = 0;
    for (mint jj = 0; jj < ii; jj++)
        {sum += Lower[ii][jj] * y[jj];}
    y[ii] = matrixB[ii] - sum;
}
for (mint ii = rc - 1; ii >= 0; ii--)
{
    sum = 0;
```

```
        for (mint jj = ii + 1; jj < rc; jj++)
            {sum += Upper[ii][jj] * x[jj];}
        x[ii] = (y[ii] - sum)/Upper[ii][ii];
    }
}

__device__ void SurNorvec opt15
{
    normv_C[0] =dxdu2*dxdv3-dxdu3*dxdv2;
    normv_C[1] =dxdu3*dxdv1-dxdu1*dxdv3;
    normv_C[2] =dxdu1*dxdv2-dxdu2*dxdv1;
}
__device__ void DSurNvDu opt20
{
    dnormv_C[0]
     =(d2xdu22*dxdv3-d2xdu23*dxdv2)+(dxdu2*d2xdudv3-dxdu3*d2xdudv2);
    dnormv_C[1]
     =(d2xdu23*dxdv1-d2xdu21*dxdv3)+(dxdu3*d2xdudv1-dxdu1*d2xdudv3);
    dnormv_C[2]
     =(d2xdu21*dxdv2-d2xdu22*dxdv1)+(dxdu1*d2xdudv2-dxdu2*d2xdudv1);

}
__device__ void DSurNvDv opt20
{
    dnormv_C[0]
     =(d2xdudv2*dxdv3-d2xdudv3*dxdv2)+(dxdu2*d2xdv23-dxdu3*d2xdv22);
    dnormv_C[1]
     =(d2xdudv3*dxdv1-d2xdudv1*dxdv3)+(dxdu3*d2xdv21-dxdu1*d2xdv23);

    dnormv_C[2]
     =(d2xdudv1*dxdv2-d2xdudv2*dxdv1)+(dxdu1*d2xdv22-dxdu2*d2xdv21);

}
__device__ void D2SurNvDu2 opt21
{
    d2normv_C[0]
     =(d3xdu32*dxdv3-d3xdu33*dxdv2)+2*(d2xdu22*d2xdudv3-d2xdu23*d2xdudv2)+(
     dxdu2*d3xdu2dv3-dxdu3*d3xdu2dv2);
    d2normv_C[1]
     =(d3xdu33*dxdv1-d3xdu31*dxdv3)+2*(d2xdu23*d2xdudv1-d2xdu21*d2xdudv3)+(
     dxdu3*d3xdu2dv1-dxdu1*d3xdu2dv3);
    d2normv_C[2]
     =(d3xdu31*dxdv2-d3xdu32*dxdv1)+2*(d2xdu21*d2xdudv2-d2xdu22*d2xdudv1)+(
     dxdu1*d3xdu2dv2-dxdu2*d3xdu2dv1);
}
__device__ void D2SurNvDuDv opt21
{
```

```
        d2normv_C[0]
         =(d3xdu2dv2*dxdv3-d3xdu2dv3*dxdv2)+(d2xdu22*d2xdv23-d2xdu23*d2xdv22)+(
         dxdu2*d3xdudv23-dxdu3*d3xdudv22);
        d2normv_C[1]
         =(d3xdu2dv3*dxdv1-d3xdu2dv1*dxdv3)+(d2xdu23*d2xdv21-d2xdu21*d2xdv23)+(
         dxdu3*d3xdudv21-dxdu1*d3xdudv23);
        d2normv_C[2]
         =(d3xdu2dv1*dxdv2-d3xdu2dv2*dxdv1)+(d2xdu21*d2xdv22-d2xdu22*d2xdv21)+(
         dxdu1*d3xdudv22-dxdu2*d3xdudv21);
}
__device__ void D2SurNvDv2 opt21
{
        d2normv_C[0]
         =(d3xdudv22*dxdv3-d3xdudv23*dxdv2)+2*(d2xdudv2*d2xdv23-d2xdudv3*d2xdv22)
         +(dxdu2*d3xdv33-dxdu3*d3xdv32);
        d2normv_C[1]
         =(d3xdudv23*dxdv1-d3xdudv21*dxdv3)+2*(d2xdudv3*d2xdv21-d2xdudv1*d2xdv23)
         +(dxdu3*d3xdv31-dxdu1*d3xdv33);
        d2normv_C[2]
         =(d3xdudv21*dxdv2-d3xdudv22*dxdv1)+2*(d2xdudv1*d2xdv22-d2xdudv2*d2xdv21)
         +(dxdu1*d3xdv32-dxdu2*d3xdv31);
}
__device__ void ScalarSurNor opt22
{
float
         ScalSNor,DScalSNorDu,DScalSNorDv,D2ScalSNorDu2,D2ScalSNorDuDv,D2ScalSNorDv2;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float d2normvdu2_C[3];
float d2normvdudv_C[3];
float d2normvdv2_C[3];
repSurNorvec;
repDSurNvDu;
repDSurNvDv;
repD2SurNvDu2;
repD2SurNvDuDv;
repD2SurNvDv2;

ScalSNor =
         sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
DScalSNorDu
         =
         (normv_C[0]*dnormvdu_C[0]+normv_C[1]*dnormvdu_C[1]+normv_C[2]*dnormvdu_C
         [2])/ScalSNor;
DScalSNorDv =
         (normv_C[0]*dnormvdv_C[0]+normv_C[1]*dnormvdv_C[1]+normv_C[2]*dnormvdv_C
```

```
    [2])/ScalSNor;
D2ScalSNorDu2 =
    ((normv_C[0]*d2normvdu2_C[0]+normv_C[1]*d2normvdu2_C[1]+normv_C[2]*
    d2normvdu2_C[2])+(dnormvdu_C[0]*dnormvdu_C[0]+dnormvdu_C[1]*dnormvdu_C[1
    ]+dnormvdu_C[2]*dnormvdu_C[2])-pow(DScalSNorDu,2))/ScalSNor;
D2ScalSNorDuDv =
    ((normv_C[0]*d2normvdudv_C[0]+normv_C[1]*d2normvdudv_C[1]+normv_C[2]*
    d2normvdudv_C[2])+(dnormvdu_C[0]*dnormvdv_C[0]+dnormvdu_C[1]*dnormvdv_C[
    1]+dnormvdu_C[2]*dnormvdv_C[2])-DScalSNorDu*DScalSNorDv)/ScalSNor;
D2ScalSNorDv2 =
    ((normv_C[0]*d2normvdv2_C[0]+normv_C[1]*d2normvdv2_C[1]+normv_C[2]*
    d2normvdv2_C[2])+(dnormvdv_C[0]*dnormvdv_C[0]+dnormvdv_C[1]*dnormvdv_C[1
    ]+dnormvdv_C[2]*dnormvdv_C[2])-pow(DScalSNorDv,2))/ScalSNor;

Scal[0] = ScalSNor;
Scal[1] = DScalSNorDu;
Scal[2] = DScalSNorDv;
Scal[3] = D2ScalSNorDu2;
Scal[4] = D2ScalSNorDuDv;
Scal[5] = D2ScalSNorDv2;
}
__device__ float coefE opt16
{
    return dxdu1*dxdu1+dxdu2*dxdu2+dxdu3*dxdu3;
}
__device__ float DEDu opt17
{
    return 2*(dxdu1*d2xdu21+dxdu2*d2xdu22+dxdu3*d2xdu23);
}
__device__ float DEDv opt17
{
    return 2*(dxdu1*d2xdudv1+dxdu2*d2xdudv2+dxdu3*d2xdudv3);
}
__device__ float D2EDu2 opt23
{
    return
    2*((d2xdu21*d2xdu21+d2xdu22*d2xdu22+d2xdu23*d2xdu23)+(dxdu1*d3xdu31+
    dxdu2*d3xdu32+dxdu3*d3xdu33));
}
__device__ float D2EDuDv opt23
{
    return
    2*((d2xdudv1*d2xdu21+d2xdudv2*d2xdu22+d2xdudv3*d2xdu23)+(dxdu1*d3xdu2dv1
    +dxdu2*d3xdu2dv2+dxdu3*d3xdu2dv3));
}
__device__ float D2EDv2 opt23
{
```

```
    return
    2*((d2xdudv1*d2xdudv1+d2xdudv2*d2xdudv2+d2xdudv3*d2xdudv3)+(dxdu1*
    d3xdudv21+dxdu2*d3xdudv22+dxdu3*d3xdudv23));
}

__device__ float coefF opt16
{
    return (dxdu1*dxdv1+dxdu2*dxdv2+dxdu3*dxdv3);
}
__device__ float DFDu opt17
{
    return dxdv1*d2xdu21+dxdv2*d2xdu22+dxdv3*d2xdu23
    + dxdu1*d2xdudv1+dxdu2*d2xdudv2+dxdu3*d2xdudv3;
}
__device__ float DFDv opt17
{
    return dxdv1*d2xdudv1+dxdv2*d2xdudv2+dxdv3*d2xdudv3
    + dxdu1*d2xdv21+dxdu2*d2xdv22+dxdu3*d2xdv23;
}
__device__ float D2FDu2 opt23
{
    return
    (dxdv1*d3xdu31+dxdv2*d3xdu32+dxdv3*d3xdu33)+2*(d2xdu21*d2xdudv1+d2xdu22*
    d2xdudv2+d2xdu23*d2xdudv3)+(dxdu1*d3xdu2dv1+dxdu2*d3xdu2dv2+dxdu3*
    d3xdu2dv3);
}
__device__ float D2FDuDv opt23
{
    return
    (dxdv1*d3xdu2dv1+dxdv2*d3xdu2dv2+dxdv3*d3xdu2dv3)+(d2xdu21*d2xdv21+
    d2xdu22*d2xdv22+d2xdu23*d2xdv23)+(d2xdudv1*d2xdudv1+d2xdudv2*d2xdudv2+
    d2xdudv3*d2xdudv3)+(dxdu1*d3xdudv21+dxdu2*d3xdudv22+dxdu3*d3xdudv23);
}
__device__ float D2FDv2 opt23
{
    return
    (dxdv1*d3xdudv21+dxdv2*d3xdudv22+dxdv3*d3xdudv23)+2*(d2xdudv1*d2xdv21+
    d2xdudv2*d2xdv22+d2xdudv3*d2xdv23)+(dxdu1*d3xdv31+dxdu2*d3xdv32+dxdu3*
    d3xdv33);
}
__device__ float coefG opt16
{
    return dxdv1*dxdv1+dxdv2*dxdv2+dxdv3*dxdv3;
}
__device__ float DGDu opt17
{
    return 2*(dxdv1*d2xdudv1+dxdv2*d2xdudv2+dxdv3*d2xdudv3);
```

```
}
__device__ float DGDv opt17
{
    return 2*(dxdv1*d2xdv21+dxdv2*d2xdv22+dxdv3*d2xdv23);
}
__device__ float D2GDu2 opt23
{
    return
    2*((d2xdudv1*d2xdudv1+d2xdudv2*d2xdudv2+d2xdudv3*d2xdudv3)+(dxdv1*
    d3xdu2dv1+dxdv2*d3xdu2dv2+dxdv3*d3xdu2dv3));
}
__device__ float D2GDuDv opt23
{
    return
    2*((d2xdv21*d2xdudv1+d2xdv22*d2xdudv2+d2xdv23*d2xdudv3)+(dxdv1*d3xdudv21
    +dxdv2*d3xdudv22+dxdv3*d3xdudv23));
}
__device__ float D2GDv2 opt23
{
    return
    2*((d2xdv21*d2xdv21+d2xdv22*d2xdv22+d2xdv23*d2xdv23)+(dxdv1*d3xdv31+
    dxdv2*d3xdv32+dxdv3*d3xdv33));
}
__device__ float coefL opt17
{
float normv_C[3];
repSurNorvec;

    return
    (normv_C[0]*d2xdu21+normv_C[1]*d2xdu22+normv_C[2]*d2xdu23)/sqrt(normv_C[
    0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float coefM opt17
{
float normv_C[3];
repSurNorvec;

    return
    (normv_C[0]*d2xdudv1+normv_C[1]*d2xdudv2+normv_C[2]*d2xdudv3)/sqrt(
    normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float coefN opt17
{
float normv_C[3];
repSurNorvec;

    return
```

```
    (normv_C[0]*d2xdv21+normv_C[1]*d2xdv22+normv_C[2]*d2xdv23)/sqrt(normv_C[
    0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float GaussCurva opt17
{
    return
    (repcoefL*repcoefN-pow(repcoefM,2))/(repcoefE*repcoefG-pow(repcoefF,2));

}
__device__ float MeanCurva opt17
{
    return
    0.5*(repcoefE*repcoefN+repcoefG*repcoefL-2*repcoefF*repcoefM)/(repcoefE*
    repcoefG-pow(repcoefF,2));
}
__device__ float MaxPrinci opt17
{
    return repMeanCurva+sqrt(pow(repMeanCurva,2)-repGaussCurva);
}

__device__ float MinPrinci opt17
{
    return repMeanCurva-sqrt(pow(repMeanCurva,2)-repGaussCurva);
}
__device__ float Princi opt18
{
    if(optimum==max)
    {return repMaxPrinci;}
    else if(optimum==min)
    {return repMinPrinci;}
    else
    {return 0;}
}
__device__ void DcoefL_2 opt24
{
float CoeL,DCoeLDu,DCoeLDv,D2CoeLDu2,D2CoeLDuDv,D2CoeLDv2;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float d2normvdu2_C[3];
float d2normvdudv_C[3];
float d2normvdv2_C[3];
float Scal[6];
repSurNorvec;
repDSurNvDu;
repDSurNvDv;
repD2SurNvDu2;
```

```
repD2SurNvDuDv;
repD2SurNvDv2;
repScalarSurNor;

CoeL = (normv_C[0]*d2xdu21+normv_C[1]*d2xdu22+normv_C[2]*d2xdu23)/Scal[0];
DCoeLDu =
    ((dnormvdu_C[0]*d2xdu21+dnormvdu_C[1]*d2xdu22+dnormvdu_C[2]*d2xdu23)+(
    normv_C[0]*d3xdu31+normv_C[1]*d3xdu32+normv_C[2]*d3xdu33)-Scal[1]*CoeL)/
    Scal[0];
DCoeLDv =
    ((dnormvdv_C[0]*d2xdu21+dnormvdv_C[1]*d2xdu22+dnormvdv_C[2]*d2xdu23)+(
    normv_C[0]*d3xdu2dv1+normv_C[1]*d3xdu2dv2+normv_C[2]*d3xdu2dv3)-Scal[2]*
    CoeL)/Scal[0];
D2CoeLDu2 =
    ((d2normvdu2_C[0]*d2xdu21+d2normvdu2_C[1]*d2xdu22+d2normvdu2_C[2]*
    d2xdu23)+2*(dnormvdu_C[0]*d3xdu31+dnormvdu_C[1]*d3xdu32+dnormvdu_C[2]*
    d3xdu33)+(normv_C[0]*d4xdu41+normv_C[1]*d4xdu42+normv_C[2]*d4xdu43)-Scal
    [3]*CoeL-2*Scal[1]*DCoeLDu)/Scal[0];
D2CoeLDuDv =
    ((d2normvdudv_C[0]*d2xdu21+d2normvdudv_C[1]*d2xdu22+d2normvdudv_C[2]*
    d2xdu23)+(dnormvdu_C[0]*d3xdu2dv1+dnormvdu_C[1]*d3xdu2dv2+dnormvdu_C[2]*
    d3xdu2dv3)+(dnormvdv_C[0]*d3xdu31+dnormvdv_C[1]*d3xdu32+dnormvdv_C[2]*
    d3xdu33)+(normv_C[0]*d4xdu3dv1+normv_C[1]*d4xdu3dv2+normv_C[2]*d4xdu3dv3
    )-Scal[4]*CoeL-Scal[1]*DCoeLDv-Scal[2]*DCoeLDu)/Scal[0];
D2CoeLDv2 =
    ((d2normvdv2_C[0]*d2xdu21+d2normvdv2_C[1]*d2xdu22+d2normvdv2_C[2]*
    d2xdu23)+2*(dnormvdv_C[0]*d3xdu2dv1+dnormvdv_C[1]*d3xdu2dv2+dnormvdv_C[2
    ]*d3xdu2dv3)+(normv_C[0]*d4xdu2dv21+normv_C[1]*d4xdu2dv22+normv_C[2]*
    d4xdu2dv23)-Scal[5]*CoeL-2*Scal[2]*DCoeLDv)/Scal[0];

coeff[0] = CoeL;
coeff[1] = DCoeLDu;
coeff[2] = DCoeLDv;
coeff[3] = D2CoeLDu2;
coeff[4] = D2CoeLDuDv;
coeff[5] = D2CoeLDv2;
}
__device__ void DcoefM_2 opt24
{
float CoeM,DCoeMDu,DCoeMDv,D2CoeMDu2,D2CoeMDuDv,D2CoeMDv2;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float d2normvdu2_C[3];
float d2normvdudv_C[3];
float d2normvdv2_C[3];
float Scal[6];
```

```
repSurNorvec;
repDSurNvDu;
repDSurNvDv;
repD2SurNvDu2;
repD2SurNvDuDv;
repD2SurNvDv2;
repScalarSurNor;

CoeM = (normv_C[0]*d2xdudv1+normv_C[1]*d2xdudv2+normv_C[2]*d2xdudv3)/Scal[0];
DCoeMDu =
    ((dnormvdu_C[0]*d2xdudv1+dnormvdu_C[1]*d2xdudv2+dnormvdu_C[2]*d2xdudv3)+
    (normv_C[0]*d3xdu2dv1+normv_C[1]*d3xdu2dv2+normv_C[2]*d3xdu2dv3)-Scal[1]
    *CoeM)/Scal[0];
DCoeMDv =
    ((dnormvdv_C[0]*d2xdudv1+dnormvdv_C[1]*d2xdudv2+dnormvdv_C[2]*d2xdudv3)+
    (normv_C[0]*d3xdudv21+normv_C[1]*d3xdudv22+normv_C[2]*d3xdudv23)-Scal[2]
    *CoeM)/Scal[0];
D2CoeMDu2 =
    ((d2normvdu2_C[0]*d2xdudv1+d2normvdu2_C[1]*d2xdudv2+d2normvdu2_C[2]*
    d2xdudv3)+2*(dnormvdu_C[0]*d3xdu2dv1+dnormvdu_C[1]*d3xdu2dv2+dnormvdu_C[
    2]*d3xdu2dv3)+(normv_C[0]*d4xdu3dv1+normv_C[1]*d4xdu3dv2+normv_C[2]*
    d4xdu3dv3)-Scal[3]*CoeM-2*Scal[1]*DCoeMDu)/Scal[0];
D2CoeMDuDv =
    ((d2normvdudv_C[0]*d2xdudv1+d2normvdudv_C[1]*d2xdudv2+d2normvdudv_C[2]*
    d2xdudv3)+(dnormvdu_C[0]*d3xdudv21+dnormvdu_C[1]*d3xdudv22+dnormvdu_C[2]
    *d3xdudv23)+(dnormvdv_C[0]*d3xdu2dv1+dnormvdv_C[1]*d3xdu2dv2+dnormvdv_C[
    2]*d3xdu2dv3)+(normv_C[0]*d4xdu2dv21+normv_C[1]*d4xdu2dv22+normv_C[2]*
    d4xdu2dv23)-Scal[4]*CoeM-Scal[1]*DCoeMDv-Scal[2]*DCoeMDu)/Scal[0];
D2CoeMDv2 =
    ((d2normvdv2_C[0]*d2xdudv1+d2normvdv2_C[1]*d2xdudv2+d2normvdv2_C[2]*
    d2xdudv3)+2*(dnormvdv_C[0]*d3xdudv21+dnormvdv_C[1]*d3xdudv22+dnormvdv_C[
    2]*d3xdudv23)+(normv_C[0]*d4xdudv31+normv_C[1]*d4xdudv32+normv_C[2]*
    d4xdudv33)-Scal[5]*CoeM-2*Scal[2]*DCoeMDv)/Scal[0];

coeff[0] = CoeM;
coeff[1] = DCoeMDu;
coeff[2] = DCoeMDv;
coeff[3] = D2CoeMDu2;
coeff[4] = D2CoeMDuDv;
coeff[5] = D2CoeMDv2;
}
__device__ void DcoefN_2 opt24
{
float CoeN,DCoeNDu,DCoeNDv,D2CoeNDu2,D2CoeNDuDv,D2CoeNDv2;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
```

```
float d2normvdu2_C[3];
float d2normvdudv_C[3];
float d2normvdv2_C[3];
float Scal[6];
repSurNorvec;
repDSurNvDu;
repDSurNvDv;
repD2SurNvDu2;
repD2SurNvDuDv;
repD2SurNvDv2;
repScalarSurNor;

CoeN = (normv_C[0]*d2xdv21+normv_C[1]*d2xdv22+normv_C[2]*d2xdv23)/Scal[0];
DCoeNDu =
    ((dnormvdu_C[0]*d2xdv21+dnormvdu_C[1]*d2xdv22+dnormvdu_C[2]*d2xdv23)+(
    normv_C[0]*d3xdudv21+normv_C[1]*d3xdudv22+normv_C[2]*d3xdudv23)-Scal[1]*
    CoeN)/Scal[0];
DCoeNDv =
    ((dnormvdv_C[0]*d2xdv21+dnormvdv_C[1]*d2xdv22+dnormvdv_C[2]*d2xdv23)+(
    normv_C[0]*d3xdv31+normv_C[1]*d3xdv32+normv_C[2]*d3xdv33)-Scal[2]*CoeN)/
    Scal[0];
D2CoeNDu2 =
    ((d2normvdu2_C[0]*d2xdv21+d2normvdu2_C[1]*d2xdv22+d2normvdu2_C[2]*
    d2xdv23)+2*(dnormvdu_C[0]*d3xdudv21+dnormvdu_C[1]*d3xdudv22+dnormvdu_C[2
    ]*d3xdudv23)+(normv_C[0]*d4xdu2dv21+normv_C[1]*d4xdu2dv22+normv_C[2]*
    d4xdu2dv23)-Scal[3]*CoeN-2*Scal[1]*DCoeNDu)/Scal[0];
D2CoeNDuDv =
    ((d2normvdudv_C[0]*d2xdv21+d2normvdudv_C[1]*d2xdv22+d2normvdudv_C[2]*
    d2xdv23)+(dnormvdu_C[0]*d3xdv31+dnormvdu_C[1]*d3xdv32+dnormvdu_C[2]*
    d3xdv33)+(dnormvdv_C[0]*d3xdudv21+dnormvdv_C[1]*d3xdudv22+dnormvdv_C[2]*
    d3xdudv23)+(normv_C[0]*d4xdudv31+normv_C[1]*d4xdudv32+normv_C[2]*
    d4xdudv33)-Scal[4]*CoeN-Scal[1]*DCoeNDv-Scal[2]*DCoeNDu)/Scal[0];
D2CoeNDv2 =
    ((d2normvdv2_C[0]*d2xdv21+d2normvdv2_C[1]*d2xdv22+d2normvdv2_C[2]*
    d2xdv23)+2*(dnormvdv_C[0]*d3xdv31+dnormvdv_C[1]*d3xdv32+dnormvdv_C[2]*
    d3xdv33)+(normv_C[0]*d4xdv41+normv_C[1]*d4xdv42+normv_C[2]*d4xdv43)-Scal
    [5]*CoeN-2*Scal[2]*DCoeNDv)/Scal[0];

coeff[0] = CoeN;
coeff[1] = DCoeNDu;
coeff[2] = DCoeNDv;
coeff[3] = D2CoeNDu2;
coeff[4] = D2CoeNDuDv;
coeff[5] = D2CoeNDv2;
}
__device__ void DGaussCurva_2 opt24
{
```

```
float GCurv,DGCurvDu,DGCurvDv,D2GCurvDu2,D2GCurvDuDv,D2GCurvDv2;
float CoeffL[6];
float CoeffM[6];
float CoeffN[6];
repDcoefL_2;
repDcoefM_2;
repDcoefN_2;

GCurv=
    (CoeffL[0]*CoeffN[0]-pow(CoeffM[0],2))/(repcoefE*repcoefG-pow(repcoefF,2));
DGCurvDu
    =
    ((pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repcoefG*repDEDu-2*repcoefF*
    repDFDu+repcoefE*repDGDu)+(-pow(repcoefF,2)+repcoefE*repcoefG)*(CoeffN[0
    ]*CoeffL[1]-2*CoeffM[0]*CoeffM[1]+CoeffL[0]*CoeffN[1]))/(pow(pow(
    repcoefF,2)-repcoefE*repcoefG,2));
DGCurvDv
    =((pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repcoefG*repDEDv-2*repcoefF*
    repDFDv+repcoefE*repDGDv)+(-pow(repcoefF,2)+repcoefE*repcoefG)*(CoeffN[0
    ]*CoeffL[2]-2*CoeffM[0]*CoeffM[2]+CoeffL[0]*CoeffN[2]))/(pow(pow(
    repcoefF,2)-repcoefE*repcoefG,2));
D2GCurvDu2 =
    ((-2*(pow(repcoefF,2)-repcoefE*repcoefG)*(repcoefG*repDEDu-2*repcoefF*
    repDFDu+repcoefE*repDGDu)*(CoeffN[0]*CoeffL[1]-2*CoeffM[0]*CoeffM[1]+
    CoeffL[0]*CoeffN[1])+(pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(2*pow(
    repcoefG*repDEDu-2*repcoefF*repDFDu+repcoefE*repDGDu,2)-(pow(repcoefF,2)
    -repcoefE*repcoefG)*(2*pow(repDFDu,2)-2*repDEDu*repDGDu-repcoefG*
    repD2EDu2+2*repcoefF*repD2FDu2-repcoefE*repD2GDu2))+pow(pow(repcoefF,2)-
    repcoefE*repcoefG,2)*(2*pow(CoeffM[1],2)-2*CoeffL[1]*CoeffN[1]-CoeffN[0]
    *CoeffL[3]+2*CoeffM[0]*CoeffM[3]-CoeffL[0]*CoeffN[3]))/pow(pow(repcoefF,
    2)-repcoefE*repcoefG,3));
D2GCurvDuDv =
    -((2*(pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repcoefG*repDEDv-2*repcoefF
    *repDFDv+repcoefE*repDGDv)*(repcoefG*repDEDu-2*repcoefF*repDFDu+repcoefE
    *repDGDu)+(-pow(repcoefF,2)+repcoefE*repcoefG)*(CoeffN[0]*CoeffL[2]-2*
    CoeffM[0]*CoeffM[2]+CoeffL[0]*CoeffN[2])*(repcoefG*repDEDu-2*repcoefF*
    repDFDu+repcoefE*repDGDu)+(-pow(repcoefF,2)+repcoefE*repcoefG)*(repcoefG
    *repDEDv-2*repcoefF*repDFDv+repcoefE*repDGDv)*(CoeffN[0]*CoeffL[1]-2*
    CoeffM[0]*CoeffM[1]+CoeffL[0]*CoeffN[1])+(-pow(repcoefF,2)+repcoefE*
    repcoefG)*(-pow(CoeffM[0],2)+CoeffL[0]*CoeffN[0])*(repDGDv*repDEDu-2*
    repDFDv*repDFDu+repDEDv*repDGDu+repcoefG*repD2EDuDv-2*repcoefF*
    repD2FDuDv+repcoefE*repD2GDuDv)-pow(pow(repcoefF,2)-repcoefE*repcoefG,2)
    *(CoeffN[2]*CoeffL[1]-2*CoeffM[2]*CoeffM[1]+CoeffL[2]*CoeffN[1]+CoeffN[0
    ]*CoeffL[4]-2*CoeffM[0]*CoeffM[4]+CoeffL[0]*CoeffN[4]))/pow(-pow(
    repcoefF,2)+repcoefE*repcoefG,3));
D2GCurvDv2 =
    ((-2*(pow(repcoefF,2)-repcoefE*repcoefG)*(repcoefG*repDEDv-2*repcoefF*
```

```
    repDFDv+repcoefE*repDGDv)*(CoeffN[0]*CoeffL[2]-2*CoeffM[0]*CoeffM[2]+
    CoeffL[0]*CoeffN[2])+(pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(2*pow(
    repcoefG*repDEDv-2*repcoefF*repDFDv+repcoefE*repDGDv,2)-(pow(repcoefF,2)
    -repcoefE*repcoefG)*(2*pow(repDFDv,2)-2*repDEDv*repDGDv-repcoefG*
    repD2EDv2+2*repcoefF*repD2FDv2-repcoefE*repD2GDv2))+pow(pow(repcoefF,2)-
    repcoefE*repcoefG,2)*(2*pow(CoeffM[2],2)-2*CoeffL[2]*CoeffN[2]-CoeffN[0]
    *CoeffL[5]+2*CoeffM[0]*CoeffM[5]-CoeffL[0]*CoeffN[5]))/pow(pow(repcoefF,
    2)-repcoefE*repcoefG,3));

coeff[0] = GCurv;
coeff[1] = DGCurvDu;
coeff[2] = DGCurvDv;
coeff[3] = D2GCurvDu2;
coeff[4] = D2GCurvDuDv;
coeff[5] = D2GCurvDv2;
}
__device__ void DMeanCurva_2 opt24
{
float MCurv,DMCurvDu,DMCurvDv,D2MCurvDu2,D2MCurvDuDv,D2MCurvDv2;
float CoeffL[6];
float CoeffM[6];
float CoeffN[6];
repDcoefL_2;
repDcoefM_2;
repDcoefN_2;

MCurv =
    0.5*(repcoefE*CoeffN[0]+repcoefG*CoeffL[0]-2*repcoefF*CoeffM[0])/(
    repcoefE*repcoefG-pow(repcoefF,2));
DMCurvDu =
    0.5*(-(repcoefG*CoeffL[0]-2*repcoefF*CoeffM[0]+repcoefE*CoeffN[0])*(
    repcoefG*repDEDu-2*repcoefF*repDFDu+repcoefE*repDGDu)+(-pow(repcoefF,2)+
    repcoefE*repcoefG)*(CoeffN[0]*repDEDu-2*CoeffM[0]*repDFDu+CoeffL[0]*
    repDGDu+repcoefG*CoeffL[1]-2*repcoefF*CoeffM[1]+repcoefE*CoeffN[1]))/(
    pow(pow(repcoefF,2)-repcoefE*repcoefG,2));
DMCurvDv =
    0.5*(-(repcoefG*CoeffL[0]-2*repcoefF*CoeffM[0]+repcoefE*CoeffN[0])*(
    repcoefG*repDEDv-2*repcoefF*repDFDv+repcoefE*repDGDv)+(-pow(repcoefF,2)+
    repcoefE*repcoefG)*(CoeffN[0]*repDEDv-2*CoeffM[0]*repDFDv+CoeffL[0]*
    repDGDv+repcoefG*CoeffL[2]-2*repcoefF*CoeffM[2]+repcoefE*CoeffN[2]))/(
    pow(pow(repcoefF,2)-repcoefE*repcoefG,2));
D2MCurvDu2 =
    -((2*(pow(repcoefF,2)-repcoefE*repcoefG)*(repcoefG*repDEDu-2*repcoefF*
    repDFDu+repcoefE*repDGDu)*(CoeffN[0]*repDEDu-2*CoeffM[0]*repDFDu+CoeffL[
    0]*repDGDu+repcoefG*CoeffL[1]-2*repcoefF*CoeffM[1]+repcoefE*CoeffN[1])+(
    repcoefG*CoeffL[0]-2*repcoefF*CoeffM[0]+repcoefE*CoeffN[0])*(2*pow(
    repcoefG*repDEDu-2*repcoefF*repDFDu+repcoefE*repDGDu,2)-(pow(repcoefF,2)
```

```
        -repcoefE*repcoefG)*(2*pow(repDFDu,2)-2*repDEDu*repDGDu-repcoefG*
        repD2EDu2+2*repcoefF*repD2FDu2-repcoefE*repD2GDu2))+pow(pow(repcoefF,2)-
        repcoefE*repcoefG,2)*(2*repDGDu*CoeffL[1]-4*repDFDu*CoeffM[1]+2*repDEDu*
        CoeffN[1]+CoeffN[0]*repD2EDu2-2*CoeffM[0]*repD2FDu2+CoeffL[0]*repD2GDu2+
        repcoefG*CoeffL[3]-2*repcoefF*CoeffM[3]+repcoefE*CoeffN[3]))/(2*pow(pow(
        repcoefF,2)-repcoefE*repcoefG,3)));
D2MCurvDuDv =
        -(0.5*(-2*(repcoefG*CoeffL[0]-2*repcoefF*CoeffM[0]+repcoefE*CoeffN[0])*(
        repcoefG*repDEDv-2*repcoefF*repDFDv+repcoefE*repDGDv)*(repcoefG*repDEDu-
        2*repcoefF*repDFDu+repcoefE*repDGDu)+(-pow(repcoefF,2)+repcoefE*repcoefG
        )*(CoeffN[0]*repDEDv-2*CoeffM[0]*repDFDv+CoeffL[0]*repDGDv+repcoefG*
        CoeffL[2]-2*repcoefF*CoeffM[2]+repcoefE*CoeffN[2])*(repcoefG*repDEDu-2*
        repcoefF*repDFDu+repcoefE*repDGDu)+(-pow(repcoefF,2)+repcoefE*repcoefG)*
        (repcoefG*repDEDv-2*repcoefF*repDFDv+repcoefE*repDGDv)*(CoeffN[0]*
        repDEDu-2*CoeffM[0]*repDFDu+CoeffL[0]*repDGDu+repcoefG*CoeffL[1]-2*
        repcoefF*CoeffM[1]+repcoefE*CoeffN[1])+(-pow(repcoefF,2)+repcoefE*
        repcoefG)*(repcoefG*CoeffL[0]-2*repcoefF*CoeffM[0]+repcoefE*CoeffN[0])*(
        repDGDv*repDEDu-2*repDFDv*repDFDu+repDEDv*repDGDu+repcoefG*repD2EDuDv-2*
        repcoefF*repD2FDuDv+repcoefE*repD2GDuDv)-pow(pow(repcoefF,2)-repcoefE*
        repcoefG,2)*(CoeffN[2]*repDEDu-2*CoeffM[2]*repDFDu+CoeffL[2]*repDGDu+
        repDGDv*CoeffL[1]-2*repDFDv*CoeffM[1]+repDEDv*CoeffN[1]+CoeffN[0]*
        repD2EDuDv-2*CoeffM[0]*repD2FDuDv+CoeffL[0]*repD2GDuDv+repcoefG*CoeffL[4
        ]-2*repcoefF*CoeffM[4]+repcoefE*CoeffN[4]))/(pow(-pow(repcoefF,2)+
        repcoefE*repcoefG,3)));
D2MCurvDv2 =
        -(0.5*(2*(pow(repcoefF,2)-repcoefE*repcoefG)*(repcoefG*repDEDv-2*
        repcoefF*repDFDv+repcoefE*repDGDv)*(CoeffN[0]*repDEDv-2*CoeffM[0]*
        repDFDv+CoeffL[0]*repDGDv+repcoefG*CoeffL[2]-2*repcoefF*CoeffM[2]+
        repcoefE*CoeffN[2])+(repcoefG*CoeffL[0]-2*repcoefF*CoeffM[0]+repcoefE*
        CoeffN[0])*(2*pow(repcoefG*repDEDv-2*repcoefF*repDFDv+repcoefE*repDGDv,2
        )-(pow(repcoefF,2)-repcoefE*repcoefG)*(2*pow(repDFDv,2)-2*repDEDv*
        repDGDv-repcoefG*repD2EDv2+2*repcoefF*repD2FDv2-repcoefE*repD2GDv2))+pow
        (pow(repcoefF,2)-repcoefE*repcoefG,2)*(2*repDGDv*CoeffL[2]-4*repDFDv*
        CoeffM[2]+2*repDEDv*CoeffN[2]+CoeffN[0]*repD2EDv2-2*CoeffM[0]*repD2FDv2+
        CoeffL[0]*repD2GDv2+repcoefG*CoeffL[5]-2*repcoefF*CoeffM[5]+repcoefE*
        CoeffN[5]))/(pow(pow(repcoefF,2)-repcoefE*repcoefG,3)));


coeff[0] = MCurv;
coeff[1] = DMCurvDu;
coeff[2] = DMCurvDv;
coeff[3] = D2MCurvDu2;
coeff[4] = D2MCurvDuDv;
coeff[5] = D2MCurvDv2;
}
__device__ void DPrinci_2 opt25
{
```

```
float Princip,DPrincipDu,DPrincipDv,D2PrincipDu2,D2PrincipDuDv,D2PrincipDv2;
float GCurva[6];
float MCurva[6];
repDGaussCurva_2;
repDMeanCurva_2;

if(optimum==max)
{Princip = MCurva[0]+sqrt(pow(MCurva[0],2)-GCurva[0]);
DPrincipDu =
    MCurva[1]+(2*MCurva[0]*MCurva[1]-GCurva[1])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
DPrincipDv =
    MCurva[2]+(2*MCurva[0]*MCurva[2]-GCurva[2])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
D2PrincipDu2 =
    -(pow(-2*MCurva[0]*MCurva[1]+GCurva[1],2))/(4*sqrt(pow(pow(MCurva[0],2)-
    GCurva[0],3)))+MCurva[3]+(2*pow(MCurva[1],2)+2*MCurva[0]*MCurva[3]-
    GCurva[3])/(2*sqrt(pow(MCurva[0],2)-GCurva[0]));
D2PrincipDuDv =
    -(((2*MCurva[0]*MCurva[2]-GCurva[2])*(2*MCurva[0]*MCurva[1]-GCurva[1]))/
    (4*sqrt(pow(pow(MCurva[0],2)-GCurva[0],3))))+MCurva[4]+(2*MCurva[2]*
    MCurva[1]+2*MCurva[0]*MCurva[4]-GCurva[4])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
D2PrincipDv2 =
    -(pow(-2*MCurva[0]*MCurva[2]+GCurva[2],2))/(4*sqrt(pow(pow(MCurva[0],2)-
    GCurva[0],3)))+MCurva[5]+(2*pow(MCurva[2],2)+2*MCurva[0]*MCurva[5]-
    GCurva[5])/(2*sqrt(pow(MCurva[0],2)-GCurva[0]));
}
else if(optimum==min)
{Princip = MCurva[0]-sqrt(pow(MCurva[0],2)-GCurva[0]);
DPrincipDu =
    MCurva[1]-(2*MCurva[0]*MCurva[1]-GCurva[1])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
DPrincipDv =
    MCurva[2]-(2*MCurva[0]*MCurva[2]-GCurva[2])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
D2PrincipDu2 =
    (pow(-2*MCurva[0]*MCurva[1]+GCurva[1],2))/(4*sqrt(pow(pow(MCurva[0],2)-
    GCurva[0],3)))+MCurva[3]+(-2*(pow(MCurva[1],2)+MCurva[0]*MCurva[3])+
    GCurva[3])/(2*sqrt(pow(MCurva[0],2)-GCurva[0]));
D2PrincipDuDv =
    ((2*MCurva[0]*MCurva[2]-GCurva[2])*(2*MCurva[0]*MCurva[1]-GCurva[1]))/(4
    *sqrt(pow(pow(MCurva[0],2)-GCurva[0],3)))+MCurva[4]+(-2*MCurva[2]*MCurva
    [1]-2*MCurva[0]*MCurva[4]+GCurva[4])/(2*sqrt(pow(MCurva[0],2)-GCurva[0])
    );
D2PrincipDv2
    =(pow(-2*MCurva[0]*MCurva[2]+GCurva[2],2))/(4*sqrt(pow(pow(MCurva[0],2)-
```

```
    GCurva[0],3)))+MCurva[5]+(-2*(pow(MCurva[2],2)+MCurva[0]*MCurva[5])+
    GCurva[5])/(2*sqrt(pow(MCurva[0],2)-GCurva[0]));
}
else
{Princip = 0;
DPrincipDu = 0;
DPrincipDv = 0;
D2PrincipDu2 = 0;
D2PrincipDuDv = 0;
D2PrincipDv2 = 0;
}

coeff[0] = Princip;
coeff[1] = DPrincipDu;
coeff[2] = DPrincipDv;
coeff[3] = D2PrincipDu2;
coeff[4] = D2PrincipDuDv;
coeff[5] = D2PrincipDv2;
}

__device__ float Eta opt18
{
    return
    sign1/sqrt(repcoefE*pow(repcoefM-repPrinci*repcoefF,2)-2*repcoefF*(
    repcoefM-repPrinci*repcoefF)*(repcoefL-repPrinci*repcoefE)+repcoefG*pow(
    repcoefL-repPrinci*repcoefE,2));
}
__device__ float Miu opt18
{
    return
    sign1/sqrt(repcoefE*pow(repcoefN-repPrinci*repcoefG,2)-2*repcoefF*(
    repcoefM-repPrinci*repcoefF)*(repcoefN-repPrinci*repcoefG)+repcoefG*pow(
    repcoefM-repPrinci*repcoefF,2));
}
__device__ float DuDs opt18
{
if(abs(repcoefL-repPrinci*repcoefE)>=abs(repcoefN-repPrinci*repcoefG))
    {return repEta*(repcoefM-repPrinci*repcoefF);}
else
    {return repMiu*(repcoefN-repPrinci*repcoefG);}
}
__device__ float DvDs opt18
{
if(abs(repcoefL-repPrinci*repcoefE)>=abs(repcoefN-repPrinci*repcoefG))
    {return -repEta*(repcoefL-repPrinci*repcoefE);}
else
    {return -repMiu*(repcoefM-repPrinci*repcoefF);}
```

```
}
__device__ float DEDs opt18
{
    return repDEDu*repDuDs+repDEDv*repDvDs;
}
__device__ float DFDs opt18
{
    return repDFDu*repDuDs+repDFDv*repDvDs;
}
__device__ float DGDs opt18
{
    return repDGDu*repDuDs+repDGDv*repDvDs;
}
__device__ float DLDs opt26
{
float CoeffL[6];
repDcoefL_2;

    return CoeffL[1]*repDuDs+CoeffL[2]*repDvDs;
}
__device__ float DMDs opt26
{
float CoeffM[6];
repDcoefM_2;

    return CoeffM[1]*repDuDs+CoeffM[2]*repDvDs;
}
__device__ float DNDs opt26
{
float CoeffN[6];
repDcoefN_2;

    return CoeffN[1]*repDuDs+CoeffN[2]*repDvDs;
}
__device__ float DPrinciDs opt26
{
float PCurva[6];
repDPrinci_2;

    return PCurva[1]*repDuDs+PCurva[2]*repDvDs;
}
__device__ float DuDt opt18
{
if(abs(repcoefL-repPrinci*repcoefE)>=abs(repcoefN-repPrinci*repcoefG))
    {return (repcoefM-repPrinci*repcoefF);}
else
    {return (repcoefN-repPrinci*repcoefG);}
```

```
}
__device__ float DvDt opt18
{
if(abs(repcoefL-repPrinci*repcoefE)>=abs(repcoefN-repPrinci*repcoefG))
    {return (repcoefL-repPrinci*repcoefE);}
else
    {return (repcoefM-repPrinci*repcoefF);}
}
__device__ void DSurDs opt19
{
    tangv_C[0] =dxdu1*repDuDs+
                dxdv1*repDvDs;
    tangv_C[1] =dxdu2*repDuDs+
                dxdv2*repDvDs;
    tangv_C[2] =dxdu3*repDuDs+
                dxdv3*repDvDs;
}

__device__ float Curvecurvature opt27
{
    return sqrt(pow(repPrinci,2)+pow(GeoCur[i],2));
}
__device__ float D2EDs2 opt27
{
    return
    repD2EDu2*pow(repDuDs,2)+2*repD2EDuDv*repDuDs*repDvDs+repD2EDv2*pow(
    repDvDs,2)+repDEDu*D2uDs2[i]+repDEDv*D2vDs2[i];
}
__device__ float D2FDs2 opt27
{
    return
    repD2FDu2*pow(repDuDs,2)+2*repD2FDuDv*repDuDs*repDvDs+repD2FDv2*pow(
    repDvDs,2)+repDFDu*D2uDs2[i]+repDFDv*D2vDs2[i];
}
__device__ float D2GDs2 opt27
{
    return
    repD2GDu2*pow(repDuDs,2)+2*repD2GDuDv*repDuDs*repDvDs+repD2GDv2*pow(
    repDvDs,2)+repDGDu*D2uDs2[i]+repDGDv*D2vDs2[i];
}
__device__ float D2LDs2 opt27
{
float CoeffL[6];
repDcoefL_2;

    return
    CoeffL[3]*pow(repDuDs,2)+2*CoeffL[4]*repDuDs*repDvDs+CoeffL[5]*pow(
```

```
    repDvDs,2)+CoeffL[1]*D2uDs2[i]+CoeffL[2]*D2vDs2[i];
}

__device__ float D2MDs2 opt27
{
float CoeffM[6];
repDcoefM_2;

    return
    CoeffM[3]*pow(repDuDs,2)+2*CoeffM[4]*repDuDs*repDvDs+CoeffM[5]*pow(
    repDvDs,2)+CoeffM[1]*D2uDs2[i]+CoeffM[2]*D2vDs2[i];
}
__device__ float D2NDs2 opt27
{
float CoeffN[6];
repDcoefN_2;

    return
    CoeffN[3]*pow(repDuDs,2)+2*CoeffN[4]*repDuDs*repDvDs+CoeffN[5]*pow(
    repDvDs,2)+CoeffN[1]*D2uDs2[i]+CoeffN[2]*D2vDs2[i];
}
__device__ float D2PrinciDs2 opt27
{
float PCurva[6];
repDPrinci_2;

    return
    PCurva[3]*pow(repDuDs,2)+2*PCurva[4]*repDuDs*repDvDs+PCurva[5]*pow(
    repDvDs,2)+PCurva[1]*D2uDs2[i]+PCurva[2]*D2vDs2[i];
}

__device__ void D2SurDs2 opt28
{
    coeff[0] =
    d2xdu21*pow(repDuDs,2)+2*d2xdudv1*repDuDs*repDvDs+d2xdv21*pow(repDvDs,2)
    +dxdu1*D2uDs2[i]+dxdv1*D2vDs2[i];

    coeff[1] =
    d2xdu22*pow(repDuDs,2)+2*d2xdudv2*repDuDs*repDvDs+d2xdv22*pow(repDvDs,2)
    +dxdu2*D2uDs2[i]+dxdv2*D2vDs2[i];

    coeff[2] =
    d2xdu23*pow(repDuDs,2)+2*d2xdudv3*repDuDs*repDvDs+d2xdv23*pow(repDvDs,2)
    +dxdu3*D2uDs2[i]+dxdv3*D2vDs2[i];
}
__device__ void CurvenormaloS opt28
{
```

```
        float dtangds[3];
        repD2SurDs2;

        coeff[0] = dtangds[0]/repCurvecurvature;
        coeff[1] = dtangds[1]/repCurvecurvature;
        coeff[2] = dtangds[2]/repCurvecurvature;
    }
    __device__ void CurvebinormaloS opt28
    {
        float tangv[3];
        float norv[3];
        repDSurDs;
        repCurvenormaloS;

        coeff[0] =tangv[1]*norv[2]-tangv[2]*norv[1];
        coeff[1] =tangv[2]*norv[0]-tangv[0]*norv[2];
        coeff[2] =tangv[0]*norv[1]-tangv[1]*norv[0];
    }

    __device__ void Alp2 opt28
    {
        coeff[0] =
        (d3xdu31*pow(repDuDs,3)+3*d3xdu2dv1*pow(repDuDs,2)*repDvDs+3*d3xdudv21*
        repDuDs*pow(repDvDs,2)+d3xdv31*pow(repDvDs,3))+3*(d2xdu21*repDuDs*D2uDs2
        [i]+d2xdudv1*(repDvDs*D2uDs2[i]+repDuDs*D2vDs2[i])+d2xdv21*repDvDs*
        D2vDs2[i]);

        coeff[1] =
        (d3xdu32*pow(repDuDs,3)+3*d3xdu2dv2*pow(repDuDs,2)*repDvDs+3*d3xdudv22*
        repDuDs*pow(repDvDs,2)+d3xdv32*pow(repDvDs,3))+3*(d2xdu22*repDuDs*D2uDs2
        [i]+d2xdudv2*(repDvDs*D2uDs2[i]+repDuDs*D2vDs2[i])+d2xdv22*repDvDs*
        D2vDs2[i]);

        coeff[2] =
        (d3xdu33*pow(repDuDs,3)+3*d3xdu2dv3*pow(repDuDs,2)*repDvDs+3*d3xdudv23*
        repDuDs*pow(repDvDs,2)+d3xdv33*pow(repDvDs,3))+3*(d2xdu23*repDuDs*D2uDs2
        [i]+d2xdudv3*(repDvDs*D2uDs2[i]+repDuDs*D2vDs2[i])+d2xdv23*repDvDs*
        D2vDs2[i]);
    }

    __device__ float Beta2 opt27
    {
    if(abs(repcoefL-repPrinci*repcoefE)>=abs(repcoefN-repPrinci*repcoefG))
    {return
        -(2*(repDLDs-repDPrinciDs*repcoefE-repPrinci*repDEDs)*D2uDs2[i]+2*(
        repDMDs-repDPrinciDs*repcoefF-repPrinci*repDFDs)*D2vDs2[i]+(repD2LDs2-
        repD2PrinciDs2*repcoefE-2*repDPrinciDs*repDEDs-repPrinci*repD2EDs2)*
```

```
    repDuDs+(repD2MDs2-repD2PrinciDs2*repcoefF-2*repDPrinciDs*repDFDs-
    repPrinci*repD2FDs2)*repDvDs);}
else
{return
    -(2*(repDMDs-repDPrinciDs*repcoefF-repPrinci*repDFDs)*D2uDs2[i]+2*(
    repDNDs-repDPrinciDs*repcoefG-repPrinci*repDGDs)*D2vDs2[i]+(repD2MDs2-
    repD2PrinciDs2*repcoefF-2*repDPrinciDs*repDFDs-repPrinci*repD2FDs2)*
    repDuDs+(repD2NDs2-repD2PrinciDs2*repcoefG-2*repDPrinciDs*repDGDs-
    repPrinci*repD2GDs2)*repDvDs);}
}

__device__ float Torsion opt27
{
    mint rc=4;
    float tangv[3];
    float norv[3];
    float binorv[3];
    float Alp2_C[3];
    repDSurDs;
    repCurvenormaloS;
    repCurvebinormaloS;
    repAlp2;
    float matrixA[4][4]={
    {repcoefE,repcoefF,-(norv[0]*dxdu1+norv[1]*dxdu2+norv[2]*dxdu3),-
    repCurvecurvature*(binorv[0]*dxdu1+binorv[1]*dxdu2+binorv[2]*dxdu3)},

    {repcoefF,repcoefG,-(norv[0]*dxdv1+norv[1]*dxdv2+norv[2]*dxdv3),-
    repCurvecurvature*(binorv[0]*dxdv1+binorv[1]*dxdv2+binorv[2]*dxdv3)},

    {(norv[0]*dxdu1+norv[1]*dxdu2+norv[2]*dxdu3),(norv[0]*dxdv1+norv[1]*
    dxdv2+norv[2]*dxdv3),-1,0},
    {repDvDt,repDuDt,0,0}};

    float
    matrixB[4]={-(Alp2_C[0]*dxdu1+Alp2_C[1]*dxdu2+Alp2_C[2]*dxdu3)-(pow(
    repCurvecurvature,2)*(tangv[0]*dxdu1+tangv[1]*dxdu2+tangv[2]*dxdu3)),-(
    Alp2_C[0]*dxdv1+Alp2_C[1]*dxdv2+Alp2_C[2]*dxdv3)-(pow(repCurvecurvature,
    2)*(tangv[0]*dxdv1+tangv[1]*dxdv2+tangv[2]*dxdv3)),-(Alp2_C[0]*norv[0]+
    Alp2_C[1]*norv[1]+Alp2_C[2]*norv[2]),repBeta2};
    float Lower[4][4];
    float Upper[4][4];
    float x[4];
    float y[4];
    float sum;

    for(mint ii=0; ii<rc; ii++)
            {
```

```
                for(mint jj=0; jj<rc; jj++)
                {
                    sum = 0;
                    if(ii==jj)
                    {
                    Lower[ii][jj]=1;
                    for (mint kk = 0; kk < ii; kk++)
                        {sum += Lower[ii][kk] * Upper[kk][jj];}
                    Upper[ii][jj] = matrixA[ii][jj] - sum;
                    }
                    else if(ii < jj)
                    {
                    Lower[ii][jj]=0;
                    for (mint kk = 0; kk < ii; kk++)
                        {sum += Lower[ii][kk] * Upper[kk][jj];}
                    Upper[ii][jj] = matrixA[ii][jj] - sum;
                    }
                    else
                    {
                    Upper[ii][jj]=0;
                    for (mint kk = 0; kk < jj; kk++)
                        {sum += Lower[ii][kk] * Upper[kk][jj];}
                    Lower[ii][jj] =(matrixA[ii][jj] - sum)/Upper[jj][jj];
                    }
                }
            }
    for (mint ii = 0; ii < rc; ii++)
    {
        sum = 0;
        for (mint jj = 0; jj < ii; jj++)
            {sum += Lower[ii][jj] * y[jj];}
        y[ii] = matrixB[ii] - sum;
    }

    for (mint ii = rc - 1; ii >= 0; ii--)
    {
        sum = 0;
        for (mint jj = ii + 1; jj < rc; jj++)
            {sum += Upper[ii][jj] * x[jj];}
        x[ii] = (y[ii] - sum)/Upper[ii][ii];
    }
return x[3];
}

__device__ float DCurvatureDs opt27
{
    mint rc=4;
```

```
float tangv[3];
float norv[3];
float binorv[3];
float Alp2_C[3];
repDSurDs;
repCurvenormaloS;
repCurvebinormaloS;
repAlp2;
float matrixA[4][4]={
{repcoefE,repcoefF,-(norv[0]*dxdu1+norv[1]*dxdu2+norv[2]*dxdu3),-
repCurvecurvature*(binorv[0]*dxdu1+binorv[1]*dxdu2+binorv[2]*dxdu3)},

{repcoefF,repcoefG,-(norv[0]*dxdv1+norv[1]*dxdv2+norv[2]*dxdv3),-
repCurvecurvature*(binorv[0]*dxdv1+binorv[1]*dxdv2+binorv[2]*dxdv3)},

{(norv[0]*dxdu1+norv[1]*dxdu2+norv[2]*dxdu3),(norv[0]*dxdv1+norv[1]*
dxdv2+norv[2]*dxdv3),-1,0},
{repDvDt,repDuDt,0,0}};

float
matrixB[4]={-(Alp2_C[0]*dxdu1+Alp2_C[1]*dxdu2+Alp2_C[2]*dxdu3)-(pow(
repCurvecurvature,2)*(tangv[0]*dxdu1+tangv[1]*dxdu2+tangv[2]*dxdu3)),-(
Alp2_C[0]*dxdv1+Alp2_C[1]*dxdv2+Alp2_C[2]*dxdv3)-(pow(repCurvecurvature,
2)*(tangv[0]*dxdv1+tangv[1]*dxdv2+tangv[2]*dxdv3)),-(Alp2_C[0]*norv[0]+
Alp2_C[1]*norv[1]+Alp2_C[2]*norv[2]),repBeta2};
float Lower[4][4];
float Upper[4][4];
float x[4];
float y[4];
float sum;

for(mint ii=0; ii<rc; ii++)
        {
            for(mint jj=0; jj<rc; jj++)
            {
                sum = 0;
                if(ii==jj)
                {
                Lower[ii][jj]=1;
                for (mint kk = 0; kk < ii; kk++)
                    {sum += Lower[ii][kk] * Upper[kk][jj];}
                Upper[ii][jj] = matrixA[ii][jj] - sum;
                }
                else if(ii < jj)
                {
                Lower[ii][jj]=0;
                for (mint kk = 0; kk < ii; kk++)
```

```
                        {sum += Lower[ii][kk] * Upper[kk][jj];}
                    Upper[ii][jj] = matrixA[ii][jj] - sum;
                    }
                    else
                    {
                    Upper[ii][jj]=0;
                    for (mint kk = 0; kk < jj; kk++)
                        {sum += Lower[ii][kk] * Upper[kk][jj];}
                    Lower[ii][jj] =(matrixA[ii][jj] - sum)/Upper[jj][jj];
                    }
                }
            }
    for (mint ii = 0; ii < rc; ii++)
    {
        sum = 0;
        for (mint jj = 0; jj < ii; jj++)
            {sum += Lower[ii][jj] * y[jj];}
        y[ii] = matrixB[ii] - sum;
    }

    for (mint ii = rc - 1; ii >= 0; ii--)
    {
        sum = 0;
        for (mint jj = ii + 1; jj < rc; jj++)
            {sum += Upper[ii][jj] * x[jj];}
        x[ii] = (y[ii] - sum)/Upper[ii][ii];
    }
return x[2];
}

__device__ float RK4_Init(float x1, float t1, float SSize)

{
float k1,k2,k3,k4;

k1=SSize*x1;
k2=SSize*(x1+0.5*SSize*t1);
k3=SSize*(x1+0.5*SSize*t1);
k4=SSize*(x1+SSize*t1);

return (1.0/6.0)*(k1+2*k2+2*k3+k4);
}


__device__ void RK4_func_LoC( float xu1, float xu2, float
    tu1, float tu2, float dtu1, float dtu2, float xv1, float xv2,
    float tv1, float tv2, float dtv1, float dtv2, mint optimum,
```

```
    float stepsize, float& increment_u, float& increment_v)

{

float u_k1,u_k2,u_k3,u_k4;
float v_k1,v_k2,v_k3,v_k4;
float newxu1,newxu2,newtu1,newtu2,newdtu1,newdtu2;
float newxv1,newxv2,newtv1,newtv2,newdtv1,newdtv2;


u_k1=stepsize*iniDuDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    optimum);
v_k1=stepsize*iniDvDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,
    dtv2,optimum);

newxu1=xu1+0.5*u_k1*tu1;
newxu2=xu2+0.5*u_k1*tu2;
newtu1=tu1+0.5*u_k1*dtu1;
newtu2=tu2+0.5*u_k1*dtu2;
newdtu1=0;
newdtu2=2;
newxv1=xv1+0.5*v_k1*tv1;
newxv2=xv2+0.5*v_k1*tv2;
newtv1=tv1+0.5*v_k1*dtv1;
newtv2=tv2+0.5*v_k1*dtv2;
newdtv1=0;
newdtv2=-2;
u_k2=stepsize*iniDuDs(newxu1,newxu2,newtu1,newtu2,newdtu1,newdtu2,newxv1,newxv2
    ,newtv1,newtv2,newdtv1,newdtv2,optimum);
v_k2=stepsize*iniDvDs(newxu1,newxu2,newtu1,newtu2,newdtu1,newdtu2,newxv1
    ,newxv2,newtv1,newtv2,newdtv1,newdtv2,optimum);

newxu1=xu1+0.5*u_k2*tu1;
newxu2=xu2+0.5*u_k2*tu2;
newtu1=tu1+0.5*u_k2*dtu1;
newtu2=tu2+0.5*u_k2*dtu2;
newdtu1=0;
newdtu2=2;
newxv1=xv1+0.5*v_k2*tv1;
newxv2=xv2+0.5*v_k2*tv2;
newtv1=tv1+0.5*v_k2*dtv1;
newtv2=tv2+0.5*v_k2*dtv2;
newdtv1=0;
newdtv2=-2;

u_k3=stepsize*iniDuDs(newxu1,newxu2,newtu1,newtu2,newdtu1,newdtu2,newxv1,newxv2
    ,newtv1,newtv2,newdtv1,newdtv2,optimum);
```

```
v_k3=stepsize*iniDvDs(newxu1,newxu2,newtu1,newtu2,newdtu1,newdtu2,newxv1
     ,newxv2,newtv1,newtv2,newdtv1,newdtv2,optimum);

newxu1=xu1+u_k3*tu1;
newxu2=xu2+u_k3*tu2;
newtu1=tu1+u_k3*dtu1;
newtu2=tu2+u_k3*dtu2;
newdtu1=0;
newdtu2=2;
newxv1=xv1+v_k3*tv1;
newxv2=xv2+v_k3*tv2;
newtv1=tv1+v_k3*dtv1;
newtv2=tv2+v_k3*dtv2;
newdtv1=0;
newdtv2=-2;

u_k4=stepsize*iniDuDs(newxu1,newxu2,newtu1,newtu2,newdtu1,newdtu2,newxv1,newxv2
     ,newtv1,newtv2,newdtv1,newdtv2,optimum);
v_k4=stepsize*iniDvDs(newxu1,newxu2,newtu1,newtu2,newdtu1,newdtu2,newxv1
     ,newxv2,newtv1,newtv2,newdtv1,newdtv2,optimum);

increment_u=(1.0/6.0)*(u_k1+2*u_k2+2*u_k3+u_k4);
increment_v=(1.0/6.0)*(v_k1+2*v_k2+2*v_k3+v_k4);

}



__device__ void AllDetails( mint i, float xu1[],float
     xu2[],float tu1[],float tu2[],float dtu1[],float dtu2[],float
     d2tu1[],float d2tu2[],float d3tu1[],float d3tu2[],float
     xv1[],float xv2[],float tv1[],float tv2[],float dtv1[],float
     dtv2[],float d2tv1[],float d2tv2[],float d3tv1[],float d3tv2[],
     float D2uDs2[], float D2vDs2[],float GeoCur[], mint optimum)
{
float iniSSize_u,iniSSize_v,SSize_u,SSize_v;

float dummy_xu1, dummy_xu2, dummy_tu1, dummy_tu2, dummy_dtu1,
     dummy_dtu2, dummy_d2tu1, dummy_d2tu2, dummy_d3tu1, dummy_d3tu2;
float dummy_xv1, dummy_xv2, dummy_tv1, dummy_tv2, dummy_dtv1,
     dummy_dtv2, dummy_d2tv1, dummy_d2tv2, dummy_d3tv1, dummy_d3tv2;
float u_s, v_s;
float coef[3];

u_s=0;
v_s=0;
dummy_xu1=0;
```

```
dummy_xu2=0;
dummy_tu1=1;
dummy_tu2=0;
dummy_dtu1=0;
dummy_dtu2=2;
dummy_d2tu1=0;
dummy_d2tu2=0;
dummy_d3tu1=0;
dummy_d3tu2=0;
dummy_xv1=0;
dummy_xv2=0;
dummy_tv1=1;
dummy_tv2=0;
dummy_dtv1=0;
dummy_dtv2=-2;
dummy_d2tv1=0;
dummy_d2tv2=0;
dummy_d3tv1=0;
dummy_d3tv2=0;


if(optimum==min)
{
    for(mint k=1;k<=i;k++)
    {

    RK4_func_LoC(dummy_xu1,dummy_xu2,dummy_tu1,dummy_tu2,dummy_dtu1,
    dummy_dtu2,dummy_xv1,dummy_xv2,dummy_tv1,dummy_tv2,dummy_dtv1,dummy_dtv2
    ,max,h,SSize_u,SSize_v);
        u_s+=SSize_u;
        v_s+=SSize_v;
        dummy_xu1+=RK4_Init(dummy_tu1,dummy_dtu1,SSize_u);
        dummy_xu2+=RK4_Init(dummy_tu2,dummy_dtu2,SSize_u);
        dummy_tu1+=RK4_Init(dummy_dtu1,dummy_d2tu1,SSize_u);
        dummy_tu2+=RK4_Init(dummy_dtu2,dummy_d2tu2,SSize_u);
        dummy_dtu1=0;
        dummy_dtu2=2;
        dummy_d2tu1=0;
        dummy_d2tu2=0;
        dummy_d3tu1=0;
        dummy_d3tu2=0;
        dummy_xv1+=RK4_Init(dummy_tv1,dummy_dtv1,SSize_v);
        dummy_xv2+=RK4_Init(dummy_tv2,dummy_dtv2,SSize_v);
        dummy_tv1+=RK4_Init(dummy_dtv1,dummy_d2tv1,SSize_v);
        dummy_tv2+=RK4_Init(dummy_dtv2,dummy_d2tv2,SSize_v);
        dummy_dtv1=0;
        dummy_dtv2=-2;
```

```
            dummy_d2tv1=0;
            dummy_d2tv2=0;
            dummy_d3tv1=0;
            dummy_d3tv2=0;
        }
    }
    else if(optimum==max)
    {
        for(mint k=1;k<=i;k++)
        {

            RK4_func_LoC(dummy_xu1,dummy_xu2,dummy_tu1,dummy_tu2,dummy_dtu1,
        dummy_dtu2,dummy_xv1,dummy_xv2,dummy_tv1,dummy_tv2,dummy_dtv1,dummy_dtv2
        ,min,h,SSize_u,SSize_v);
            u_s+=SSize_u;
            v_s+=SSize_v;
            dummy_xu1+=RK4_Init(dummy_tu1,dummy_dtu1,SSize_u);
            dummy_xu2+=RK4_Init(dummy_tu2,dummy_dtu2,SSize_u);
            dummy_tu1+=RK4_Init(dummy_dtu1,dummy_d2tu1,SSize_u);
            dummy_tu2+=RK4_Init(dummy_dtu2,dummy_d2tu2,SSize_u);
            dummy_dtu1=0;
            dummy_dtu2=2;
            dummy_d2tu1=0;
            dummy_d2tu2=0;
            dummy_d3tu1=0;
            dummy_d3tu2=0;
            dummy_xv1+=RK4_Init(dummy_tv1,dummy_dtv1,SSize_v);
            dummy_xv2+=RK4_Init(dummy_tv2,dummy_dtv2,SSize_v);
            dummy_tv1+=RK4_Init(dummy_dtv1,dummy_d2tv1,SSize_v);
            dummy_tv2+=RK4_Init(dummy_dtv2,dummy_d2tv2,SSize_v);
            dummy_dtv1=0;
            dummy_dtv2=-2;
            dummy_d2tv1=0;
            dummy_d2tv2=0;
            dummy_d3tv1=0;
            dummy_d3tv2=0;

        }
    }
    xu1[0]=dummy_xu1;
    xu2[0]=dummy_xu2;
    tu1[0]=dummy_tu1;
    tu2[0]=dummy_tu2;
    dtu1[0]=dummy_dtu1;
    dtu2[0]=dummy_dtu2;
    d2tu1[0]=dummy_d2tu1;
    d2tu2[0]=dummy_d2tu2;
```

```
d3tu1[0]=dummy_d3tu1;
d3tu2[0]=dummy_d3tu2;

xv1[0]=dummy_xv1;
xv2[0]=dummy_xv2;
tv1[0]=dummy_tv1;
tv2[0]=dummy_tv2;
dtv1[0]=dummy_dtv1;
dtv2[0]=dummy_dtv2;
d2tv1[0]=dummy_d2tv1;
d2tv2[0]=dummy_d2tv2;
d3tv1[0]=dummy_d3tv1;
d3tv2[0]=dummy_d3tv2;

LUDecomposition3x3(dummy_xu1,dummy_xu2,dummy_tu1,dummy_tu2,dummy_dtu1,
    dummy_dtu2,dummy_d2tu1,dummy_d2tu2,dummy_xv1,dummy_xv2,dummy_tv1,
    dummy_tv2,dummy_dtv1,dummy_dtv2,dummy_d2tv1,dummy_d2tv2,optimum,coef);
D2uDs2[0]=coef[0];
D2vDs2[0]=coef[1];
GeoCur[0]=coef[2];

for(mint j=1;j<n;j++)
{
    RK4_func_LoC(dummy_xu1,dummy_xu2,dummy_tu1,dummy_tu2,dummy_dtu1,
    dummy_dtu2,dummy_xv1,dummy_xv2,dummy_tv1,dummy_tv2,dummy_dtv1,dummy_dtv2
    ,optimum,h,SSize_u,SSize_v);
    u_s+=SSize_u;
    v_s+=SSize_v;
    dummy_xu1+=RK4_Init(dummy_tu1,dummy_dtu1,SSize_u);
    dummy_xu2+=RK4_Init(dummy_tu2,dummy_dtu2,SSize_u);
    dummy_tu1+=RK4_Init(dummy_dtu1,dummy_d2tu1,SSize_u);
    dummy_tu2+=RK4_Init(dummy_dtu2,dummy_d2tu2,SSize_u);
    dummy_dtu1=0;
    dummy_dtu2=2;
    dummy_d2tu1=0;
    dummy_d2tu2=0;
    dummy_d3tu1=0;
    dummy_d3tu2=0;

    dummy_xv1+=RK4_Init(dummy_tv1,dummy_dtv1,SSize_v);
    dummy_xv2+=RK4_Init(dummy_tv2,dummy_dtv2,SSize_v);
    dummy_tv1+=RK4_Init(dummy_dtv1,dummy_d2tv1,SSize_v);
    dummy_tv2+=RK4_Init(dummy_dtv2,dummy_d2tv2,SSize_v);
    dummy_dtv1=0;
    dummy_dtv2=-2;
    dummy_d2tv1=0;
    dummy_d2tv2=0;
```

```
    dummy_d3tv1=0;
    dummy_d3tv2=0;

    xu1[j]=dummy_xu1;
    xu2[j]=dummy_xu2;
    tu1[j]=dummy_tu1;
    tu2[j]=dummy_tu2;
    dtu1[j]=dummy_dtu1;
    dtu2[j]=dummy_dtu2;
    d2tu1[j]=dummy_d2tu1;
    d2tu2[j]=dummy_d2tu2;
    d3tu1[j]=dummy_d3tu1;
    d3tu2[j]=dummy_d3tu2;
    xv1[j]=dummy_xv1;
    xv2[j]=dummy_xv2;
    tv1[j]=dummy_tv1;
    tv2[j]=dummy_tv2;
    dtv1[j]=dummy_dtv1;
    dtv2[j]=dummy_dtv2;
    d2tv1[j]=dummy_d2tv1;
    d2tv2[j]=dummy_d2tv2;
    d3tv1[j]=dummy_d3tv1;
    d3tv2[j]=dummy_d3tv2;

    LUDecomposition3x3(dummy_xu1,dummy_xu2,dummy_tu1,dummy_tu2,dummy_dtu1,
    dummy_dtu2,dummy_d2tu1,dummy_d2tu2,dummy_xv1,dummy_xv2,dummy_tv1,
    dummy_tv2,dummy_dtv1,dummy_dtv2,dummy_d2tv1,dummy_d2tv2,optimum,coef);
    D2uDs2[j]=coef[0];
    D2vDs2[j]=coef[1];
    GeoCur[j]=coef[2];

}

}

__global__ void ReturnEveryDetails( float * xu1, float * xu2, float * tu1,
    float * tu2, float * dtu1, float * dtu2, float * d2tu1, float *
    d2tu2, float * d3tu1, float * d3tu2, float * xv1, float * xv2, float
    * tv1, float * tv2, float * dtv1, float * dtv2, float * d2tv1,
    float * d2tv2, float * d3tv1, float * d3tv2, float * D2uDs2, float
    * D2vDs2, float * GeoCur, mint init_s, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float Txu1[n], Txu2[n], Ttu1[n], Ttu2[n], Tdtu1[n],
    Tdtu2[n], Td2tu1[n], Td2tu2[n], Td3tu1[n], Td3tu2[n];
float Txv1[n], Txv2[n], Ttv1[n], Ttv2[n], Tdtv1[n], Tdtv2[n],
    Td2tv1[n], Td2tv2[n], Td3tv1[n], Td3tv2[n];
```

```
float TD2uDs2[n], TD2vDs2[n], TGeoCur[n];

AllDetails( init_s, Txu1, Txu2, Ttu1, Ttu2, Tdtu1, Tdtu2, Td2tu1, Td2tu2,
    Td3tu1, Td3tu2, Txv1, Txv2, Ttv1, Ttv2, Tdtv1, Tdtv2, Td2tv1,
    Td2tv2, Td3tv1, Td3tv2, TD2uDs2, TD2vDs2, TGeoCur, optimum);

if( index < ListSize)
{
    xu1[index]=Txu1[index];
    xu2[index]=Txu2[index];
    tu1[index]=Ttu1[index];
    tu2[index]=Ttu2[index];
    dtu1[index]=Tdtu1[index];
    dtu2[index]=Tdtu2[index];
    d2tu1[index]=Td2tu1[index];
    d2tu2[index]=Td2tu2[index];
    d3tu1[index]=Td3tu1[index];
    d3tu2[index]=Td3tu2[index];
    xv1[index]=Txv1[index];
    xv2[index]=Txv2[index];
    tv1[index]=Ttv1[index];
    tv2[index]=Ttv2[index];
    dtv1[index]=Tdtv1[index];
    dtv2[index]=Tdtv2[index];
    d2tv1[index]=Td2tv1[index];
    d2tv2[index]=Td2tv2[index];
    d3tv1[index]=Td3tv1[index];
    d3tv2[index]=Td3tv2[index];
    D2uDs2[index]=TD2uDs2[index];
    D2vDs2[index]=TD2vDs2[index];
    GeoCur[index]=TGeoCur[index];

}

}
__global__ void CurvatureofLoC opt29
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;

if( index < ListSize)
{
    pvalue[index]=Curvecurvature opt30;

}
}
```

```
__global__ void TorsionofLoC opt29
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;

if( index < ListSize)
{
    pvalue[index]=Torsion opt30;

}
}

__global__ void DCurvatureDsofLoC opt29
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;

if( index < ListSize)
{
    pvalue[index]=DCurvatureDs opt30;

}
}
";
```

```
Needs["CUDALink`"]
```

```
HPCurvature =
  CUDAFunctionLoad[kernelcodeHP, "CurvatureofLoC", {{"Float", _, "Output"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, _Integer, _Integer}, 160];
HPTorsion = CUDAFunctionLoad[kernelcodeHP, "TorsionofLoC",
   {{"Float", _, "Output"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, _Integer, _Integer}, 160];
HPDerivativeofCurvature = CUDAFunctionLoad[kernelcodeHP,
   "DCurvatureDsofLoC", {{"Float", _, "Output"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, _Integer, _Integer}, 160];
```

```
AbsoluteTiming[
 HPAllDetails = CUDAFunctionLoad[kernelcodeHP, "ReturnEveryDetails",
   {{"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, _Integer, _Integer, _Integer}, 512]]
```

```
{165.929,
 CUDAFunction[<>, ReturnEveryDetails, {{Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, {Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, {Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, {Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, {Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, Integer64, Integer64, Integer64}]}
```

```
HPReturnAll[v_, nn_, optimum_] :=
  Module[{opt1 = optimum, xu1, xu2, tu1, tu2, dtu1, dtu2,
    d2tu1, d2tu2, d3tu1, d3tu2, xv1, xv2, tv1, tv2, dtv1,
    dtv2, d2tv1, d2tv2, d3tv1, d3tv2, D2uDs2, D2vDs2, GeoCur},
   xu1 = CUDAMemoryAllocate["Float", nn];
   xu2 = CUDAMemoryAllocate["Float", nn];
   tu1 = CUDAMemoryAllocate["Float", nn];
   tu2 = CUDAMemoryAllocate["Float", nn];
   dtu1 = CUDAMemoryAllocate["Float", nn];
   dtu2 = CUDAMemoryAllocate["Float", nn];
   d2tu1 = CUDAMemoryAllocate["Float", nn];
   d2tu2 = CUDAMemoryAllocate["Float", nn];
   d3tu1 = CUDAMemoryAllocate["Float", nn];
   d3tu2 = CUDAMemoryAllocate["Float", nn];
   xv1 = CUDAMemoryAllocate["Float", nn];
   xv2 = CUDAMemoryAllocate["Float", nn];
   tv1 = CUDAMemoryAllocate["Float", nn];
   tv2 = CUDAMemoryAllocate["Float", nn];
   dtv1 = CUDAMemoryAllocate["Float", nn];
   dtv2 = CUDAMemoryAllocate["Float", nn];
   d2tv1 = CUDAMemoryAllocate["Float", nn];
   d2tv2 = CUDAMemoryAllocate["Float", nn];
   d3tv1 = CUDAMemoryAllocate["Float", nn];
   d3tv2 = CUDAMemoryAllocate["Float", nn];
   D2uDs2 = CUDAMemoryAllocate["Float", nn];
   D2vDs2 = CUDAMemoryAllocate["Float", nn];
   GeoCur = CUDAMemoryAllocate["Float", nn];

   HPAllDetails[xu1, xu2, tu1, tu2, dtu1, dtu2, d2tu1,
    d2tu2, d3tu1, d3tu2, xv1, xv2, tv1, tv2, dtv1, dtv2, d2tv1,
    d2tv2, d3tv1, d3tv2, D2uDs2, D2vDs2, GeoCur, v, optimum, nn];
   Transpose[{CUDAMemoryGet[D2uDs2], CUDAMemoryGet[D2vDs2],
     CUDAMemoryGet[GeoCur]}]];
```

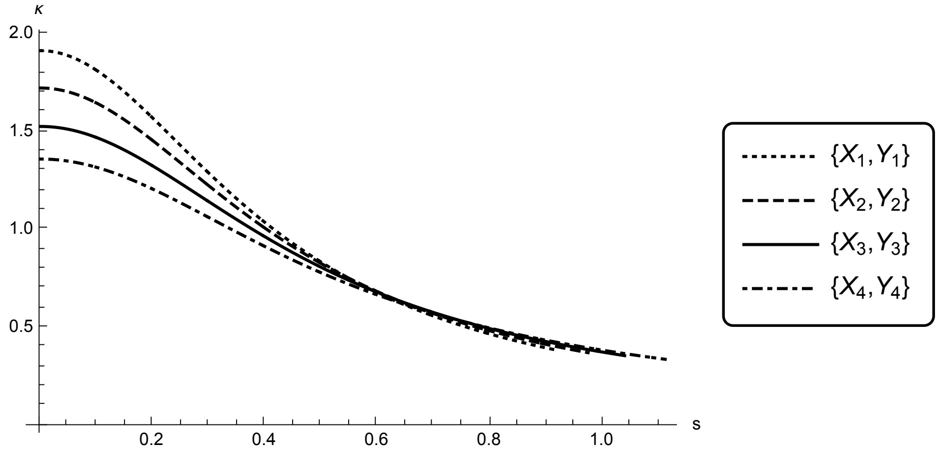
```
HPCurvatureofLoC[v_, nn_, optimum_] :=
  Module[{opt1 = optimum, xu1, xu2, tu1, tu2, dtu1, dtu2,
    d2tu1, d2tu2, d3tu1, d3tu2, xv1, xv2, tv1, tv2, dtv1, dtv2,
    d2tv1, d2tv2, d3tv1, d3tv2, D2uDs2, D2vDs2, GeoCur, Cur},
   xu1 = CUDAMemoryAllocate["Float", nn];
   xu2 = CUDAMemoryAllocate["Float", nn];
   tu1 = CUDAMemoryAllocate["Float", nn];
   tu2 = CUDAMemoryAllocate["Float", nn];
   dtu1 = CUDAMemoryAllocate["Float", nn];
   dtu2 = CUDAMemoryAllocate["Float", nn];
   d2tu1 = CUDAMemoryAllocate["Float", nn];
   d2tu2 = CUDAMemoryAllocate["Float", nn];
   d3tu1 = CUDAMemoryAllocate["Float", nn];
   d3tu2 = CUDAMemoryAllocate["Float", nn];
   xv1 = CUDAMemoryAllocate["Float", nn];
   xv2 = CUDAMemoryAllocate["Float", nn];
   tv1 = CUDAMemoryAllocate["Float", nn];
   tv2 = CUDAMemoryAllocate["Float", nn];
   dtv1 = CUDAMemoryAllocate["Float", nn];
   dtv2 = CUDAMemoryAllocate["Float", nn];
   d2tv1 = CUDAMemoryAllocate["Float", nn];
   d2tv2 = CUDAMemoryAllocate["Float", nn];
   d3tv1 = CUDAMemoryAllocate["Float", nn];
   d3tv2 = CUDAMemoryAllocate["Float", nn];
   D2uDs2 = CUDAMemoryAllocate["Float", nn];
   D2vDs2 = CUDAMemoryAllocate["Float", nn];
   GeoCur = CUDAMemoryAllocate["Float", nn];
   Cur = CUDAMemoryAllocate["Float", nn];

   HPAllDetails[xu1, xu2, tu1, tu2, dtu1, dtu2, d2tu1,
    d2tu2, d3tu1, d3tu2, xv1, xv2, tv1, tv2, dtv1, dtv2, d2tv1,
    d2tv2, d3tv1, d3tv2, D2uDs2, D2vDs2, GeoCur, v, optimum, nn];
   HPCurvature[Cur, xu1, xu2, tu1, tu2, dtu1, dtu2, d2tu1, d2tu2,
    d3tu1, d3tu2, xv1, xv2, tv1, tv2, dtv1, dtv2, d2tv1,
    d2tv2, d3tv1, d3tv2, D2uDs2, D2vDs2, GeoCur, optimum, nn];
   Transpose[{Table[i, {i, 0, (nn - 1) / 1000, 0.001}], CUDAMemoryGet[Cur]}]];
```

```
HPTorsionofLoC[v_, nn_, optimum_] :=
  Module[{opt1 = optimum, xu1, xu2, tu1, tu2, dtu1, dtu2, d2tu1,
    d2tu2, d3tu1, d3tu2, xv1, xv2, tv1, tv2, dtv1, dtv2, d2tv1,
    d2tv2, d3tv1, d3tv2, D2uDs2, D2vDs2, GeoCur, Torsion},
   xu1 = CUDAMemoryAllocate["Float", nn];
   xu2 = CUDAMemoryAllocate["Float", nn];
   tu1 = CUDAMemoryAllocate["Float", nn];
   tu2 = CUDAMemoryAllocate["Float", nn];
   dtu1 = CUDAMemoryAllocate["Float", nn];
   dtu2 = CUDAMemoryAllocate["Float", nn];
   d2tu1 = CUDAMemoryAllocate["Float", nn];
   d2tu2 = CUDAMemoryAllocate["Float", nn];
   d3tu1 = CUDAMemoryAllocate["Float", nn];
   d3tu2 = CUDAMemoryAllocate["Float", nn];
   xv1 = CUDAMemoryAllocate["Float", nn];
   xv2 = CUDAMemoryAllocate["Float", nn];
   tv1 = CUDAMemoryAllocate["Float", nn];
   tv2 = CUDAMemoryAllocate["Float", nn];
   dtv1 = CUDAMemoryAllocate["Float", nn];
   dtv2 = CUDAMemoryAllocate["Float", nn];
   d2tv1 = CUDAMemoryAllocate["Float", nn];
   d2tv2 = CUDAMemoryAllocate["Float", nn];
   d3tv1 = CUDAMemoryAllocate["Float", nn];
   d3tv2 = CUDAMemoryAllocate["Float", nn];
   D2uDs2 = CUDAMemoryAllocate["Float", nn];
   D2vDs2 = CUDAMemoryAllocate["Float", nn];
   GeoCur = CUDAMemoryAllocate["Float", nn];
   Torsion = CUDAMemoryAllocate["Float", nn];

   HPAllDetails[xu1, xu2, tu1, tu2, dtu1, dtu2, d2tu1,
    d2tu2, d3tu1, d3tu2, xv1, xv2, tv1, tv2, dtv1, dtv2, d2tv1,
    d2tv2, d3tv1, d3tv2, D2uDs2, D2vDs2, GeoCur, v, optimum, nn];
   HPTorsion[Torsion, xu1, xu2, tu1, tu2, dtu1, dtu2, d2tu1,
    d2tu2, d3tu1, d3tu2, xv1, xv2, tv1, tv2, dtv1, dtv2, d2tv1,
    d2tv2, d3tv1, d3tv2, D2uDs2, D2vDs2, GeoCur, optimum, nn];
   Transpose[{Table[i, {i, 0, (nn - 1) / 1000, 0.001}], CUDAMemoryGet[Torsion]}]];
```

```
HPDerivativeCurvatureofLoC[v_, nn_, optimum_] :=
  Module[{opt1 = optimum, xu1, xu2, tu1, tu2, dtu1, dtu2,
    d2tu1, d2tu2, d3tu1, d3tu2, xv1, xv2, tv1, tv2, dtv1, dtv2,
    d2tv1, d2tv2, d3tv1, d3tv2, D2uDs2, D2vDs2, GeoCur, DCurDs},
   xu1 = CUDAMemoryAllocate["Float", nn];
   xu2 = CUDAMemoryAllocate["Float", nn];
   tu1 = CUDAMemoryAllocate["Float", nn];
   tu2 = CUDAMemoryAllocate["Float", nn];
   dtu1 = CUDAMemoryAllocate["Float", nn];
   dtu2 = CUDAMemoryAllocate["Float", nn];
   d2tu1 = CUDAMemoryAllocate["Float", nn];
   d2tu2 = CUDAMemoryAllocate["Float", nn];
   d3tu1 = CUDAMemoryAllocate["Float", nn];
   d3tu2 = CUDAMemoryAllocate["Float", nn];
   xv1 = CUDAMemoryAllocate["Float", nn];
   xv2 = CUDAMemoryAllocate["Float", nn];
   tv1 = CUDAMemoryAllocate["Float", nn];
   tv2 = CUDAMemoryAllocate["Float", nn];
   dtv1 = CUDAMemoryAllocate["Float", nn];
   dtv2 = CUDAMemoryAllocate["Float", nn];
   d2tv1 = CUDAMemoryAllocate["Float", nn];
   d2tv2 = CUDAMemoryAllocate["Float", nn];
   d3tv1 = CUDAMemoryAllocate["Float", nn];
   d3tv2 = CUDAMemoryAllocate["Float", nn];
   D2uDs2 = CUDAMemoryAllocate["Float", nn];
   D2vDs2 = CUDAMemoryAllocate["Float", nn];
   GeoCur = CUDAMemoryAllocate["Float", nn];
   DCurDs = CUDAMemoryAllocate["Float", nn];

   HPAllDetails[xu1, xu2, tu1, tu2, dtu1, dtu2, d2tu1,
    d2tu2, d3tu1, d3tu2, xv1, xv2, tv1, tv2, dtv1, dtv2, d2tv1,
    d2tv2, d3tv1, d3tv2, D2uDs2, D2vDs2, GeoCur, v, optimum, nn];
   HPDerivativeofCurvature[DCurDs, xu1, xu2, tu1, tu2, dtu1, dtu2,
    d2tu1, d2tu2, d3tu1, d3tu2, xv1, xv2, tv1, tv2, dtv1, dtv2,
    d2tv1, d2tv2, d3tv1, d3tv2, D2uDs2, D2vDs2, GeoCur, optimum, nn];
   Transpose[{Table[i, {i, 0, (nn - 1) / 1000, 0.001}], CUDAMemoryGet[DCurDs]}]];
```

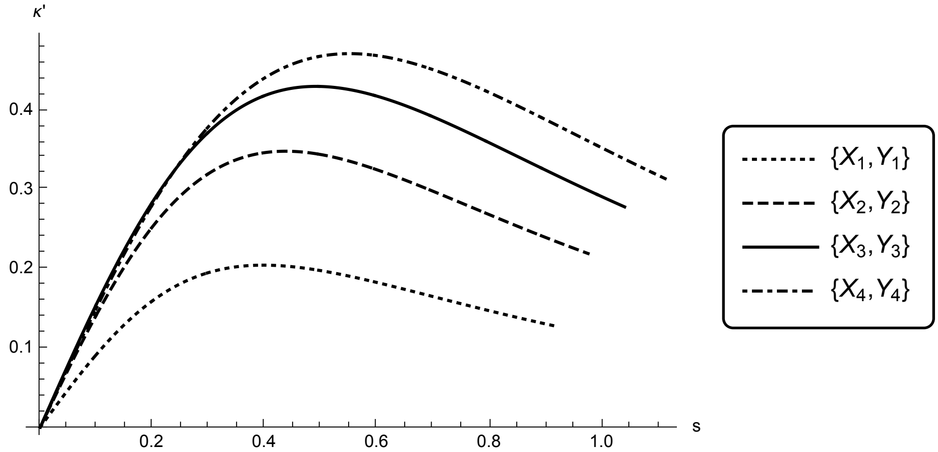
Curvature

```
ListLinePlot[{HPCurvatureofLoC[180, 923, 1],
   HPCurvatureofLoC[360, 975, 1], HPCurvatureofLoC[540, 1040, 1],
   HPCurvatureofLoC[720, 1112, 1]}, AxesLabel → {"s", "κ"},
  PlotStyle → {{Dotted, Black}, {Dashed, Black}, {Black}, {DotDashed, Black}},
  PlotLegends → LineLegend[{{{"X₁,Y₁"}, {"X₂,Y₂"}, {"X₃,Y₃"}, {"X₄,Y₄"}}},
    LegendFunction → (Framed[#, RoundingRadius → 5] &), LegendMargins → 5]]
```



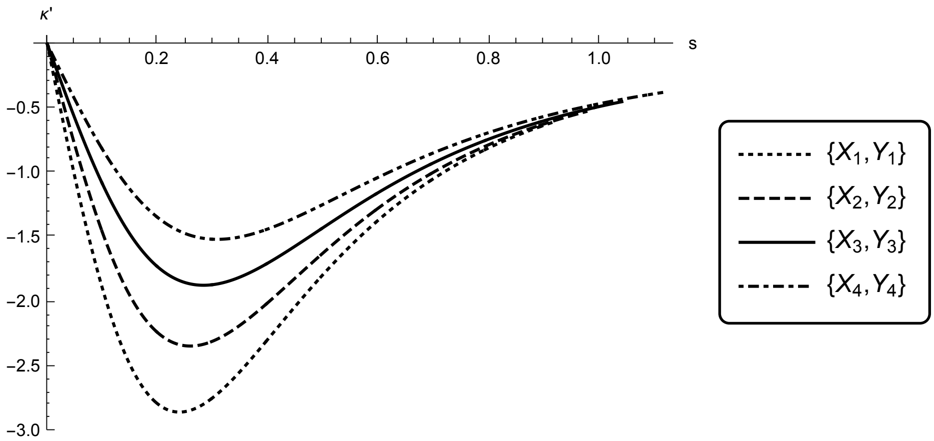
Torsion

```
ListLinePlot[{HPTorsionofLoC[180, 923, 1],
   HPTorsionofLoC[360, 975, 1], HPTorsionofLoC[540, 1040, 1],
   HPTorsionofLoC[720, 1112, 1]}, AxesLabel → {"s", "κ'"},
  PlotStyle → {{Dotted, Black}, {Dashed, Black}, {Black}, {DotDashed, Black}},
  PlotLegends → LineLegend[{{{"X₁,Y₁"}, {"X₂,Y₂"}, {"X₃,Y₃"}, {"X₄,Y₄"}}},
    LegendFunction → (Framed[#, RoundingRadius → 5] &), LegendMargins → 5]]
```



Derivative of Curvature

```
ListLinePlot[{HPDerivativeCurvatureofLoC[180, 923, 1],
  HPDerivativeCurvatureofLoC[360, 975, 1],
  HPDerivativeCurvatureofLoC[540, 1040, 1],
  HPDerivativeCurvatureofLoC[720, 1112, 1]}, AxesLabel → {"s", "κ'"},
 PlotStyle → {{Dotted, Black}, {Dashed, Black}, {Black}, {DotDashed, Black}},
 PlotLegends → LineLegend[{{{"X₁,Y₁"}, {"X₂,Y₂"}, {"X₃,Y₃"}, {"X₄,Y₄"}}},
   LegendFunction → (Framed[#, RoundingRadius → 5] &), LegendMargins → 5]]
```



LCG

Set up

```
HPCurvatureofLoC2[v_, nn_, optimum_] :=
  Module[{opt1 = optimum, xu1, xu2, tu1, tu2, dtu1, dtu2,
    d2tu1, d2tu2, d3tu1, d3tu2, xv1, xv2, tv1, tv2, dtv1, dtv2,
    d2tv1, d2tv2, d3tv1, d3tv2, D2uDs2, D2vDs2, GeoCur, Cur},
   xu1 = CUDAMemoryAllocate["Float", nn];
   xu2 = CUDAMemoryAllocate["Float", nn];
   tu1 = CUDAMemoryAllocate["Float", nn];
   tu2 = CUDAMemoryAllocate["Float", nn];
   dtu1 = CUDAMemoryAllocate["Float", nn];
   dtu2 = CUDAMemoryAllocate["Float", nn];
   d2tu1 = CUDAMemoryAllocate["Float", nn];
   d2tu2 = CUDAMemoryAllocate["Float", nn];
   d3tu1 = CUDAMemoryAllocate["Float", nn];
   d3tu2 = CUDAMemoryAllocate["Float", nn];
   xv1 = CUDAMemoryAllocate["Float", nn];
   xv2 = CUDAMemoryAllocate["Float", nn];
   tv1 = CUDAMemoryAllocate["Float", nn];
   tv2 = CUDAMemoryAllocate["Float", nn];
   dtv1 = CUDAMemoryAllocate["Float", nn];
   dtv2 = CUDAMemoryAllocate["Float", nn];
   d2tv1 = CUDAMemoryAllocate["Float", nn];
   d2tv2 = CUDAMemoryAllocate["Float", nn];
   d3tv1 = CUDAMemoryAllocate["Float", nn];
   d3tv2 = CUDAMemoryAllocate["Float", nn];
   D2uDs2 = CUDAMemoryAllocate["Float", nn];
   D2vDs2 = CUDAMemoryAllocate["Float", nn];
   GeoCur = CUDAMemoryAllocate["Float", nn];
   Cur = CUDAMemoryAllocate["Float", nn];

   HPAllDetails[xu1, xu2, tu1, tu2, dtu1, dtu2, d2tu1,
    d2tu2, d3tu1, d3tu2, xv1, xv2, tv1, tv2, dtv1, dtv2, d2tv1,
    d2tv2, d3tv1, d3tv2, D2uDs2, D2vDs2, GeoCur, v, optimum, nn];
   HPCurvature[Cur, xu1, xu2, tu1, tu2, dtu1, dtu2, d2tu1, d2tu2,
    d3tu1, d3tu2, xv1, xv2, tv1, tv2, dtv1, dtv2, d2tv1,
    d2tv2, d3tv1, d3tv2, D2uDs2, D2vDs2, GeoCur, optimum, nn];
   CUDAMemoryGet[Cur]];
```

Four

```
1 / HPCurvatureofLoC2[180, 923, 0]
```

```
1 / HPCurvatureofLoC2[360, 975, 0]
```

```
1 / HPCurvatureofLoC2[540, 1040, 0]
```

```
1 / HPCurvatureofLoC2[720, 1112, 0]
HP2rho180[s_] := {0.5230430594479403`, 0.5230459945954584`,
    0.5230546697795766`, 0.523069248650644`, 0.5230896669592048`,
    0.5231158934512358`, 0.5231478646225383`, 0.5231857783614371`,
    0.5232294088004825`, 0.52327895465076`, 0.5233343539734718`,
    0.5233954145449204`, 0.5234624016434081`, 0.5235352544669367`,
    0.5236138144879833`, 0.5236982830021968`, 0.5237885676007926`,
    0.5238846742863837`, 0.5239865767217409`, 0.5240943143949364`,
    0.524207763449207`, 0.5243272917764275`, 0.5244524486209207`,
    0.5245834716404014`, 0.5247202711183507`, 0.5248629874026997`,
    0.5250113672244039`, 0.5251655845145728`, 0.5253258139783776`,
    0.5254916715168388`, 0.5256634312217562`, 0.5258409397553324`,
    0.5260243735806664`, 0.5262134478941853`, 0.5264084059597312`,
    0.5266092937669246`, 0.5268157938829992`, 0.5270282504489436`,
    0.5272465119904062`, 0.5274704932208479`, 0.5277003412755873`,
    0.5279359383459823`, 0.5281773660794676`, 0.528424673476686`,
    0.5286777101544833`, 0.5289365589483173`, 0.529201269965759`,
    0.5294716599185173`, 0.5297479465387415`, 0.5300299469224982`,
    0.5303178129680365`, 0.5306114290528804`, 0.5309108809665473`,
    0.5312160197262663`, 0.5315271337987724`, 0.531843906377911`,
    0.532166559613021`, 0.5324949114175633`, 0.532829083702665`,
    0.5331689959658024`, 0.5335147372445923`, 0.5338662614295137`,
    0.5342235564813854`, 0.5345865765149329`, 0.5349555144795164`,
    0.5353301205643507`, 0.5357105201564994`, 0.5360967025859934`,
    0.5364886230829181`, 0.5368863743275807`, 0.5372898777517138`,
    0.5376990890417201`, 0.5381141709806336`, 0.5385349073145754`,
    0.5389614615873551`, 0.5393937211611171`, 0.5398317813543725`,
    0.5402757035505418`, 0.5407252012343988`, 0.5411805099076015`,
    0.5416415171578408`, 0.542108390296472`, 0.5425810172630842`,
    0.5430593205932098`, 0.5435433985808118`, 0.5440331387373327`,
    0.5445287107399595`, 0.5450299670036473`, 0.5455370073988134`,
    0.5460497905593871`, 0.5465682393894412`, 0.5470924904736957`,
    0.5476223956364683`, 0.5481581281064045`, 0.548699467856345`,
    0.5492465888680473`, 0.5497994144418324`, 0.5503579757479516`,
    0.5509222681903804`, 0.5514922147802612`, 0.5520678470374836`,
    0.5526492333008826`, 0.5532362966878004`, 0.5538291791893104`,
    0.5544276943659279`, 0.5550318379128081`, 0.5556417896330241`,
    0.556257362183766`, 0.5568786624250122`, 0.5575056500843575`,
    0.558138284639608`, 0.5587767114110814`, 0.559420666788039`,
    0.5600704457198009`, 0.5607257463084387`, 0.561386827452048`,
    0.5620535740009699`, 0.5627259830421655`, 0.5634040138753369`,
    0.5640878530723547`, 0.5647771567564481`, 0.5654722638700872`,
    0.5661730200043426`, 0.5668793458951396`, 0.5675913919650365`,
    0.5683090406957149`, 0.5690323664237714`, 0.5697612894961374`,
    0.5704959234736685`, 0.5712361886353756`, 0.5719820824666378`,
    0.5727335242488777`, 0.5734905892626251`, 0.5742534322299312`,
    0.5750218942157045`, 0.5757958936400581`, 0.5765755065685184`,
```

```
0.577360770160513`, 0.5781516024452359`, 0.5789480404187973`,
0.5797502016158432`, 0.5805577629023901`, 0.5813710824749698`,
0.5821899972738716`, 0.5830145044579516`, 0.5838446011303826`,
0.5846802435823831`, 0.5855214693094656`, 0.5863682752180198`,
0.5872207403541214`, 0.5880787385125098`, 0.5889422660066704`,
0.5898114019970855`, 0.5906860601574514`, 0.5915663199470478`,
0.5924521361902693`, 0.5933436309800254`, 0.5942404226763505`,
0.5951429694006022`, 0.5960510571536575`, 0.5969646391692859`,
0.5978837958649938`, 0.5988084803352285`, 0.5997385592504183`,
0.6006743276450991`, 0.6016156526924042`, 0.6025623998001948`,
0.6035146930931498`, 0.6044725710546072`, 0.6054359414654876`,
0.6064047984916449`, 0.6073791361128641`, 0.6083591687143876`,
0.6093444494073378`, 0.610335369092261`, 0.6113316999298929`,
0.6123336132481053`, 0.6133409685873444`, 0.6143538034737622`,
0.6153722009398197`, 0.6163959279374578`, 0.6174253389392197`,
0.6184600178874456`, 0.6195002749811738`, 0.6205459654157249`,
0.6215971725751168`, 0.6226537959570073`, 0.6237158726563835`,
0.6247833470513041`, 0.625856302676431`, 0.6269347303468003`,
0.6280186676023876`, 0.6291079164358055`, 0.6302027020123555`,
0.6313028248829073`, 0.6324084163629623`, 0.6335193702180325`,
0.63463586661387`, 0.6357577026224696`, 0.6368849623742759`,
0.6380175854351477`, 0.639155656418542`, 0.6402991632137696`,
0.6414480442949404`, 0.6426022370558511`, 0.6437619742137837`,
0.6449269462516031`, 0.6460974361079594`, 0.6472732312040863`,
0.6484543161563886`, 0.6496409770146142`, 0.6508328473906039`,
0.6520301634589853`, 0.6532327577312101`, 0.6544407153077162`,
0.6556541735874151`, 0.6568728603063931`, 0.6580967570608084`,
0.6593262077083858`, 0.6605609368576131`, 0.6618009780185657`,
0.6630464173941429`, 0.6642970792369468`, 0.6655531017421898`,
0.6668146249797299`, 0.6680811524376943`, 0.66935314117556`,
0.6706305188799427`, 0.6719131042554456`, 0.6732009290911909`,
0.674494079668959`, 0.6757925344838267`, 0.6770963807672719`,
0.6784053776793473`, 0.6797197210174799`, 0.6810393329350879`,
0.6823642451433064`, 0.6836943224394089`, 0.6850298188493672`,
0.6863703743826653`, 0.687716412472704`, 0.6890675705491457`,
0.6904240483242051`, 0.6917857631770359`, 0.6931526882054505`,
0.6945247383322629`, 0.6959021155367635`, 0.6972847349682982`,
0.6986726265529827`, 0.7000656451745623`, 0.7014638777721723`,
0.7028674124330311`, 0.704276101894347`, 0.7056899150019912`,
0.7071091774820764`, 0.708533501923029`, 0.7099629149973281`,
0.7113976245553133`, 0.7128374781216479`, 0.7142825029478622`,
0.7157327264778366`, 0.7171882376658634`, 0.718648695697467`,
0.7201144957434944`, 0.7215854805640656`, 0.7230614277531857`,
0.7245427368041378`, 0.7260291223205491`, 0.7275206725175742`,
0.7290172233913443`, 0.7305190532957232`, 0.7320259977799962`,
0.733537953552959`, 0.7350551374071433`, 0.7365774461019455`,
0.7381048393660467`, 0.7396374064344495`, 0.741175106803473`,
```

```
0.7427178332768898`, 0.7442656747155928`, 0.7458185885988358`,
0.7473765980275784`, 0.74893972623898`, 0.7505078623152295`,
0.7520811629337287`, 0.7536595840258749`, 0.7552430805238025`,
0.7568314697814442`, 0.7584251829541016`, 0.7600236942048679`,
0.7616274366475496`, 0.7632361560652041`, 0.7648499423577626`,
0.7664687466421454`, 0.768092589286123`, 0.7697214907584143`,
0.7713554716292701`, 0.7729944101110808`, 0.7746383966923085`,
0.7762873794889164`, 0.7779413776298713`, 0.7796004103293099`,
0.7812644241246803`, 0.7829334374680043`, 0.7846075422743551`,
0.7862865369906568`, 0.7879705124255109`, 0.7896595607163994`,
0.7913534767313848`, 0.7930525014133614`, 0.7947562772478944`,
0.7964653477708324`, 0.7981792024114911`, 0.7998980079997353`,
0.8016219340278946`, 0.8033505372089155`, 0.8050842935484676`,
0.8068229886720183`, 0.8085664821820305`, 0.8103150223912027`,
0.8120685472480635`, 0.8138269144331948`, 0.8155901375192592`,
0.817358469019653`, 0.8191315257240215`, 0.8209097208610159`,
0.8226925871237739`, 0.8244804591752039`, 0.8262733517592988`,
0.8280708709375101`, 0.8298736008522108`, 0.8316810637131768`,
0.8334934356716488`, 0.8353105630449463`, 0.8371327904938626`,
0.8389597972560762`, 0.8407914252707066`, 0.8426281059991114`,
0.844469768200387`, 0.8463162537743746`, 0.8481675731466233`,
0.8500236505889949`, 0.8518845818644603`, 0.8537503773307232`,
0.8556210473233895`, 0.857496426846075`, 0.8593766999580997`,
0.8612618769137765`, 0.8631518791279834`, 0.865046626428620 7`,
0.866946126634617`, 0.8688506574893631`, 0.8707596879526444`,
0.8726738576244456`, 0.8745924494403765`, 0.8765161072716098`,
0.8784444743009212`, 0.8803775567851169`, 0.8823154537163477`,
0.8842583588667092`, 0.8862057202609364`, 0.8881578222949148`,
0.8901149542374222`, 0.8920766513520926`, 0.8940430122035273`,
0.8960144240603238`, 0.897990415706885`, 0.8999711832690447`,
0.9019565382647325`, 0.9039468731556397`, 0.9059418044776287`,
0.9079416296890547`, 0.9099458617114876`, 0.9119549960868573`,
0.9139688399309627`, 0.9159874968251781`, 0.9180106701703504`,
0.9200386633908294`, 0.9220714810815374`, 0.9241087205392843`,
0.9261506875298233`, 0.9281973849923087`, 0.9302488157597657`,
0.9323049825577803`, 0.934365783928759`, 0.9364312209974904`,
0.938501504760413`, 0.940576111569268`, 0.942655567630691`,
0.9447397699745667`, 0.9468283999037438`, 0.94892188432182`,
0.9510199028551657`, 0.9531223458835723`, 0.9552297543661471`,
0.9573416954287737`, 0.9594582770554085`, 0.9615794987952471`,
0.9637052493429575`, 0.9658356376960388`, 0.9679707746632051`,
0.9701104363156747`, 0.9722547326759847`, 0.9744035492295418`,
0.976556996678573 2`, 0.9787150733831429`, 0.9808776628553422`,
0.9830449920253501`, 0.9852165974649542`, 0.987393053414116`,
0.9895740103001195`, 0.991759581230667`, 0.9939496456674674`,
0.9961443174073815`, 0.9983434745285555`, 1.0005472315190087`,
1.00275558492466`, 1.0049684709203535`, 1.0071858849070963`,
```

```
1.0094077613715446`, 1.0116342775413087`, 1.0138652460690036`,
1.016100845423538`, 1.018340948081001`, 1.0205854241570682`,
1.0228347026026134`, 1.025088280367964`, 1.027346337401618`,
1.0296091831340441`, 1.0318760534585731`, 1.0341477005503836`,
1.0364239934532602`, 1.0387045415512994`, 1.040989786927764`,
1.0432793360160053`, 1.045573244338914`, 1.047871895578165`,
1.0501748912719615`, 1.0524825521128047`, 1.0547944085865208`,
1.0571107155004322`, 1.0594317990009`, 1.0617569829858435`,
1.0640866591876488`, 1.0664208872657561`, 1.068759727852952`,
1.0711027638909265`, 1.073450257734795`, 1.0758023383410902`,
1.078158582573565`, 1.0805194644603067`, 1.0828846275736816`,
1.0852542707605006`, 1.0876282437671736`, 1.090006606094343`,
1.0923896316193404`, 1.0947768131383533`, 1.0971685672478766`,
1.0995644530338595`, 1.1019648903543615`, 1.104369797517257`,
1.1067789453074055`, 1.1091926137727592`, 1.11161049860426`,
1.1140327339599645`, 1.1164597534982832`, 1.1188906552391946`,
1.121326170180727`, 1.123765913002334`, 1.126210247595917 5`,
1.1286586330240982`, 1.1311114339755828`, 1.1335684853505203`,
1.1360302345235394`, 1.1384961322709588`, 1.1409662404969725`,
1.143440778034061`, 1.1459194196177622`, 1.1484026989888203`,
1.1508902117557218`, 1.1533821014891685`, 1.1558782753280796`,
1.1583785587904245`, 1.1608832563099942`, 1.1633923548971552`,
1.165905841302779`, 1.168423376523886`, 1.1709452694650482`,
1.173471341914385`, 1.1760019071495467`, 1.1785367043390342`,
1.1810758838349351`, 1.1836190140510707`, 1.1861665784317719`,
1.1887185634325912`, 1.1912747014839795`, 1.193835145137659`,
1.196399708309569`, 1.1989683730693887`, 1.201541637497471`,
1.2041189711930966`, 1.2067005289361292`, 1.209286381017252`,
1.2118765113950465`, 1.2144708158320017`, 1.217069364878893`,
1.2196722303030232`, 1.2222790398636183`, 1.2248903092251517`,
1.227505934234712`, 1.2301253579631903`, 1.2327491914894744`,
1.235377237688763`, 1.23800938740982`, 1.240645804878662`,
1.2432864728734792`, 1.2459312813544259`, 1.2485803042546444`,
1.251233430152212`, 1.2538909205923652`, 1.2565524769238685`,
1.2592182679693358`, 1.2618880857997636`, 1.264561909597311`,
1.267240483978871`, 1.2699229321043162`, 1.2726092310124284`,
1.275300035992537`, 1.2779947479258846`, 1.2806938329244688`,
1.2833969796306095`, 1.2861040686038037`, 1.288815572458895`,
1.2915311764468147`, 1.294250959875651`, 1.296974602368415`,
1.2997024829753716`, 1.3024346838618643`, 1.305170983988387`,
1.3079109546831877`, 1.3106552863952343`, 1.3134039615172974`,
1.316156652397046`, 1.318913233261153`, 1.3216740968683178`,
1.3244390152906518`, 1.327207966784234`, 1.329981034716204`,
1.3327584097664982`, 1.3355395416349112`, 1.3383250451899253`,
1.3411145808061693`, 1.3439078032000247`, 1.3467054423794567`,
1.3495071551030855`, 1.3523129194235965`, 1.355122603617647`,
1.3579364035504842`, 1.3607539665706707`, 1.363575931638236`,
```

```
1.3664019468350492`, 1.3692321013994029`, 1.3720659250501974`,
1.374904066498706`, 1.3777460538845583`, 1.3805924300368675`,
1.3834423778971352`, 1.3862966706089306`, 1.389154713924091`,
1.3920169432756873`, 1.394883105528362`, 1.3977535256643638`,
1.4006277155753022`, 1.4035058836146626`, 1.4063881237555318`,
1.4092744131997845`, 1.4121644910983102`, 1.4150586891345776`,
1.4179573445179543`, 1.4208597170410862`, 1.4237657793300929`,
1.4266758676168647`, 1.4295900801001526`, 1.4325080273891804`,
1.4354306647232375`, 1.4383564967446854`, 1.4412866038443206`,
1.4442204679999142`, 1.447158436578367`, 1.4501003613492596`,
1.4530462175685463`, 1.4559959801594138`, 1.4589501311925122`,
1.461907632014607`, 1.4648696014788731`, 1.467835378830424`,
1.4708049373243706`, 1.4737785087963953`, 1.4767559389655307`,
1.4797372016330965`, 1.482722663376444`, 1.4857119073711242`,
1.488704906407214`, 1.4917020308207616`, 1.4947031243499123`,
1.497708028270493`, 1.500716581376398`, 1.503729293850573`,
1.5067458722636866`, 1.5097662902513296`, 1.5127909303323526`,
1.5158189486584266`, 1.5188514129427433`, 1.5218873373383088`,
1.52492738350571`, 1.5279712497017321`, 1.5310190485628938`,
1.5340706140702192`, 1.5371256368036115`, 1.5401853578243245`,
1.5432484828950626`, 1.5463155499349712`, 1.5493863901221032`,
1.5524609756846859`, 1.5555398554090683`, 1.5586222837103818`,
1.561708520822202`, 1.564798977171689`, 1.5678930441003376`,
1.57099083889194`, 1.5740927761734003`, 1.5771980910668113`,
1.5803074952255585`, 1.5834205161354014`, 1.586537724113181`,
1.5896584936007523`, 1.5927837013910309`, 1.5959119610525254`,
1.5990444552844227`, 1.602180396486078`, 1.6053203671837204`,
1.6084643423472136`, 1.6116118321928024`, 1.6147632718415168`,
1.6179186352761512`, 1.6210775828833726`, 1.6242403988123457`,
1.6274068981880307`, 1.6305773687677652`, 1.6337514665303878`,
1.636929321312883`, 1.6401110650887054`, 1.6432965100698487`,
1.6464857888034519`, 1.6496788737922923`, 1.6528757372058946`,
1.6560760239367374`, 1.6592806863085046`, 1.6624888793936676`,
1.665700572111188`, 1.6689163970889587`, 1.672135330288572`,
1.6753586741260127`, 1.678585400046864`, 1.6818163181898969`,
1.685050728142054`, 1.6882884284447932`, 1.6915305790350101`,
1.6947759610181294`, 1.6980252272472407`, 1.70127817816616`,
1.704535130781789`, 1.7077958855444375`, 1.7110597164085322`,
1.7143279884019444`, 1.7175996257918982`, 1.7208747720462638`,
1.7241541045273852`, 1.7274365324286847`, 1.730722912322378`,
1.7340132178951746`, 1.7373070627141447`, 1.740604957480381`,
1.7439059706156095`, 1.7472113399710612`, 1.7505201305658145`,
1.753832126650619`, 1.7571480295153499`, 1.7604679970304224`,
1.7637912624018097`, 1.7671183502192411`, 1.7704492326498842`,
1.7737836940026224`, 1.7771218919171707`, 1.7804634196789138`,
1.7838091921820467`, 1.7871584248622845`, 1.7905108946868284`,
1.7938675262407353`, 1.79722733288663`, 1.8005910524752253`,
```

1.8039584642217719`, 1.8073293441246716`, 1.8107040512106316`,
1.8140819692391799`, 1.8174644417123789`, 1.8208502632856758`,
1.8242394008021026`, 1.8276320198488936`, 1.8310286882933433`,
1.8344287791129952`, 1.8378326626540933`, 1.8412403107581177`,
1.8446512893096594`, 1.8480661757132841`, 1.8514845344487343`,
1.854906128718015`, 1.8583311300295555`, 1.8617607455735896`,
1.8651932974418597`, 1.868629372377164`, 1.8720693570617515`,
1.8755132246741342`, 1.8789603167839046`, 1.882410810197044`,
1.8858650964553005`, 1.8893233602248105`, 1.8927845077294945`,
1.8962495724818702`, 1.8997185277750288`, 1.903190698909057`,
1.90666605146216`, 1.9101458555252908`, 1.913628562372116`,
1.9171154485840416`, 1.9206056108848408`, 1.9240987948991224`,
1.9275965144466771`, 1.931096971426993`, 1.9346014647755316`,
1.9381092974678178`, 1.9416206612535298`, 1.9451357508032263`,
1.9486540847280196`, 1.952176310540162`, 1.9557019453953879`,
1.9592311859531344`, 1.9627640022228072`, 1.9663003638962506`,
1.9698400090641766`, 1.9733838327403181`, 1.976930413463834`,
1.980481114844115`, 1.984034739377799`, 1.9875921916603236`,
1.9911529715590537`, 1.9947177576698212`, 1.9982858103881416`,
2.001857454478375`, 2.0054326599216927`, 2.009010915241954`,
2.012593150361697`, 2.016178854885962`, 2.0197679975330183`,
2.023360546701918`, 2.0269565929155293`, 2.0305561052360854`,
2.034159052409566`, 2.0377656503719175`, 2.0413754972876106`,
2.0449890581634476`, 2.0486065545896186`, 2.0522267051965426`,
2.055850855742875`, 2.0594784751241724`, 2.0631092785168432`,
2.066743613697245`, 2.0703815779999153`, 2.07402301385524`,
2.077667761586081`, 2.081316046694221`, 2.084967709327653`,
2.088623107540359`, 2.0922819522209615`, 2.0959438193846136`,
2.09960959302864`, 2.1032785876159017`, 2.106951166506526`,
2.110627034473215`, 2.1143068248559986`, 2.1179895756917464`,
2.121675921558312`, 2.125365294436326`, 2.1290587396025398`,
2.132755283624389`, 2.1364551638941305`, 2.140158894259946`,
2.143865900020107`, 2.147576010949079`, 2.151290020089305`,
2.1550074859759976`, 2.158728238888634`, 2.1624522462462243`,
2.1661793353128713`, 2.169910032733183`, 2.1736444501392955`,
2.177382135638246`, 2.1811226312187992`, 2.1848671799004706`,
2.1886149006398976`, 2.1923657598353494`, 2.196121017184815`,
2.199878921020289`, 2.2036408795930074`, 2.207405706722225`,
2.211174094340676`, 2.2149458667440136`, 2.218721286066816`,
2.2224995876830835`, 2.226281326043507`, 2.230066618677275`,
2.233855287399618`, 2.237646704048325`, 2.2414423282590574`,
2.2452412358664673`, 2.249043846942854`, 2.252848923562439`,
2.256657639434951`, 2.2604696615314634`, 2.264285263609307`,
2.268104723253418`, 2.271926937438719`, 2.275752177481697`,
2.2795811831331916`, 2.283413927311828`, 2.287249603107249`,
2.291088644338092`, 2.294931019928929`, 2.298777013484948`,
2.3026261224677462`, 2.3064784718845446`, 2.3103345067088967`,

```
    2.3141935607569026`, 2.318055919874804`, 2.321921552529944`,
    2.325790426897031`, 2.329662996098`, 2.333538907995185`,
    2.3374179686849654`, 2.3413004719103383`, 2.3451858959090264`,
    2.3490750277371264`, 2.3529675100561174`, 2.356863146430091`,
    2.3607620701324596`, 2.3646649162705584`, 2.3685701549461458`,
    2.3724790878213224`, 2.3763910143349336`, 2.380307082191398`,
    2.3842257437784014`, 2.3881479807085904`, 2.3920734234611234`,
    2.396002211028709`, 2.3999343127109767`, 2.4038693530949473`,
    2.4078083342075884`, 2.411750191207482`, 2.415695584518956`,
    2.4196437875035812`, 2.423595814659785`, 2.4275511132776524`,
    2.431509652108159`, 2.4354712228471755`, 2.4394363239998507`,
    2.4434040372956205`, 2.4473757531888998`, 2.45135019410595`,
    2.45532876095157`, 2.4593096284756157`, 2.463294383674487`,
    2.4672815501569954`, 2.4712727266299663`, 2.4752675241100155`,
    2.479264999086746`, 2.483265666189568`, 2.487268940627133`,
    2.4912768197301474`, 2.4952874299264503`, 2.49930110745754`,
    2.5033180062970835`, 2.5073382825364874`, 2.5113620943984265`,
    2.515389225121064`, 2.5194188877547785`, 2.5234525626882327`,
    2.527488893699648`, 2.5315289901755698`, 2.53557110181365`,
    2.5396176847573813`, 2.5436675629857395`, 2.547719738947508`,
    2.551775531210633`, 2.555834717910059`, 2.5598970744508183`,
    2.563962765327454`, 2.5680311710917834`, 2.5721034390191515`,
    2.576178555420754`, 2.58025708115886`, 2.584338191042011`,
    2.588423046280291`, 2.592511420802979`, 2.5966020810777284`,
    2.6006963985306264`, 2.6047941444199014`, 2.6088948843646373`,
    2.612998992551328`, 2.61710643976634`, 2.621216991773122`,
    2.6253304115035467`, 2.629446871148183`, 2.6335669586021067`}[[s]];
HP2rho360[s_] := {0.5810529914752516`, 0.5810557283349908`, 0.5810638585703859`,
    0.5810775033876413`, 0.5810965830561525`, 0.5811210986466083`,
    0.581151091796235`, 0.5811865642010452`, 0.581227437323377`,
    0.581273793998`, 0.5813254757057104`, 0.5813828479263158`,
    0.5814454707031085`, 0.5815136699185538`, 0.5815871672400191`,
    0.5816662086758644`, 0.58175079873977`, 0.5818406597655896`,
    0.5819360388484218`, 0.5820369010182965`, 0.582143130764798`,
    0.5822548956049882`, 0.5823721210318709`, 0.5824946922733736`,
    0.5826227374477445`, 0.5827563042256496`, 0.5828952381508273`,
    0.5830396279047785`, 0.5831894410039364`, 0.5833446857555648`,
    0.5835054113630626`, 0.5836715862087872`, 0.5838432601649607`,
    0.5840203615344781`, 0.5842027780491896`, 0.5843907638397848`,
    0.584584044381682`, 0.584782874660374`, 0.5849871434891494`,
    0.5851967804790048`, 0.5854119195827379`, 0.5856324501196069`,
    0.5858583431830671`, 0.5860897747882007`, 0.5863265939319316`,
    0.5865688544723822`, 0.5868166107151327`, 0.587069670909293`,
    0.5873281718761192`, 0.5875920865460131`, 0.5878615528006895`,
    0.5881363794177844`, 0.5884166223644258`, 0.5887022967620703`,
    0.5889933766920308`, 0.589289836386277`, 0.5895917745398032`,
    0.5898990830926699`, 0.590211819886994`, 0.5905299601132541`,
```

```
0.5908535207154585`, 0.591182477290844`, 0.5915168472773017`,
0.5918565649125332`, 0.5922017733985696`, 0.592552365704046`,
0.5929084021499217`, 0.5932697757464169`, 0.5936365892544138`,
0.594008777918354`, 0.5943863611113`, 0.5947694006941451`,
0.5951577901664611`, 0.5955514226847043`, 0.5959506144211869`,
0.5963551747159183`, 0.5967651668064685`, 0.5971804419213635`,
0.5976011912403202`, 0.5980272235226384`, 0.5984586881905934`,
0.5988955219383846`, 0.599337789745565`, 0.5997853427462586`,
0.6002382890641977`, 0.600696565600899`, 0.6011603244597187`,
0.6016294168289686`, 0.602103822795055`, 0.6025836523462137`,
0.6030687995695718`, 0.6035592448890051`, 0.6040550992546566`,
0.6045563873729229`, 0.6050629597104736`, 0.6055748407521463`,
0.6060920115088444`, 0.6066147600773154`, 0.6071427175585425`,
0.6076760408704234`, 0.6082146676853113`, 0.6087586237275723`,
0.6093078907860628`, 0.6098625393277575`, 0.610422596049452`,
0.6109878654717966`, 0.6115585074650587`, 0.6121345045455793`,
0.6127157049609356`, 0.613302314631698`, 0.6138941368430975`,
0.6144915136287387`, 0.6150939779786212`, 0.6157018275579995`,
0.6163149549651746`, 0.6169333427862126`, 0.6175572008811847`,
0.6181861486817112`, 0.6188205786453032`, 0.6194602003883826`,
0.6201051337627578`, 0.6207554078321992`, 0.6214108678572375`,
0.6220716346476499`, 0.6227376914289651`, 0.6234091603440333`,
0.6240856539371498`, 0.6247676190057695`, 0.6254548068213176`,
0.626147246946951`, 0.6268448755820294`, 0.6275479100870117`,
0.6282561934133895`, 0.6289696614614477`, 0.6296883441260108`,
0.6304122716170534`, 0.6311415219490115`, 0.6318759835154691`,
0.6326156868269519`, 0.6333606148921069`, 0.6341108464554579`,
0.6348661727511545`, 0.6356269609958828`, 0.6363928577741536`,
0.6371638934424424`, 0.6379402442066326`, 0.6387217962101294`,
0.6395085807475823`, 0.6403005316826562`, 0.6410976802169364`,
0.6419000578647642`, 0.6427075979724766`, 0.6435203813095888`,
0.6443384399675335`, 0.6451616078833834`, 0.6459899660109769`,
0.6468234464986703`, 0.6476622307953268`, 0.6485061011435392`,
0.6493551893482504`, 0.6502094773796907`, 0.6510688459282895`,
0.6519335290494686`, 0.6528033060518994`, 0.653678259471667`,
0.6545582683959444`, 0.6554435178420001`, 0.6563339893850288`,
0.6572294583867503`, 0.6581300591356196`, 0.6590359273463195`,
0.6599468367487906`, 0.6608629750149979`, 0.6617841663914692`,
0.662710494873293`, 0.6636419928354328`, 0.6645785876542005`,
0.6655203114168129`, 0.6664670906068956`, 0.6674189570003458`,
0.6683760491719566`, 0.669338240181337`, 0.6703054015444034`,
0.6712778329444371`, 0.6722550832400315`, 0.6732376684968554`,
0.6742253520900355`, 0.6752179488706777`, 0.6762156529836699`,
0.6772184420955211`, 0.6782264031833258`, 0.6792394041717564`,
0.680257421631721`, 0.6812804317517018`, 0.6823085768219462`,
0.6833417780913109`, 0.6843800115800975`, 0.6854232529081599`,
0.6864715896402022`, 0.6875249412576778`, 0.6885832826592032`,
```

0.6896467584101128`, 0.6907151735123178`, 0.6917886727238046`,
0.6928670591197096`, 0.6939505924459175`, 0.6950390748861923`,
0.6961326526200465`, 0.6972310678872479`, 0.6983345827466085`,
0.6994429959004504`, 0.7005566299020795`, 0.7016751069855695`,
0.7027984565693991`, 0.703926826454083`, 0.7050603069479632`,
0.7061986921286104`, 0.7073420714834228`, 0.7084903560066472`,
0.7096435754544397`, 0.7108018802855067`, 0.7119649997980027`,
0.7131331445838114`, 0.7143062237380781`, 0.7154842060291636`,
0.7166671208874614`, 0.7178549979812862`, 0.7190478672192796`,
0.7202455732320232`, 0.7214482066026991`, 0.7226556722075729`,
0.7238681233413636`, 0.7250855271818648`, 0.7263077245297502`,
0.7275349324585507`, 0.728766927638637`, 0.7300038647021808`,
0.7312457089936704`, 0.7324923612556741`, 0.7337440416045052`,
0.7350003934087475`, 0.736261765796028`, 0.7375279937879112`,
0.7387990400888327`, 0.7400749973003513`, 0.7413556315436154`,
0.7426413629756903`, 0.7439318257360471`, 0.7452271783167046`,
0.7465272490729379`, 0.7478322637884284`, 0.7491421167303514`,
0.7504567004985864`, 0.7517761081217301`, 0.7531003661536984`,
0.7544294334940591`, 0.7557633363686005`, 0.7571019645317842`,
0.7584453431368339`, 0.7597936351515854`, 0.7611466603708152`,
0.7625045130668967`, 0.7638671495642366`, 0.7652344555684879`,
0.7666067353053474`, 0.7679835941004678`, 0.7693654075901183`,
0.7707517776632331`, 0.7721429399190597`, 0.7735389191987979`,
0.7749395973377562`, 0.7763450698118518`, 0.77775528908046`,
0.7791702067205711`, 0.7805897007779555`, 0.7820141576440446`,
0.7834431643280884`, 0.7848771093914761`, 0.7863155024438411`,
0.787758733411828`, 0.7892066037918686`, 0.7906593590958257`,
0.7921167246893235`, 0.7935786469026802`, 0.7950453724945992`,
0.796516773543279`, 0.797992795944877`, 0.7994735369898877`,
0.8009589422761283`, 0.8024489563993201`, 0.8039435229399122`,
0.8054430484635521`, 0.8069470139352766`, 0.8084555935810117`,
0.809969042268475`, 0.8114869904982225`, 0.8130095357904704`,
0.8145366978601513`, 0.8160684965329331`, 0.8176049517459932`,
0.8191460035593356`, 0.8206916712289956`, 0.8222419741159579`,
0.8237967698861842`, 0.825356319893896`, 0.8269203186744367`,
0.8284890286986248`, 0.8300622234377819`, 0.8316400025141705`,
0.8332224668888101`, 0.8348094696438185`, 0.8364009448212806`,
0.8379970764133408`, 0.8395977149143804`, 0.8412028770502834`,
0.8428124949450488`, 0.8444269245156373`, 0.8460457590037636`,
0.8476690991430149`, 0.8492970470437786`, 0.8509293609403448`,
0.8525664017038709`, 0.8542077527481999`, 0.8558538640562227`,
0.8575043158331173`, 0.859159297564253`, 0.860818913164567`,
0.8624828245930852`, 0.8641511334877663`, 0.8658242109964297`,
0.8675016264476574`, 0.8691836627695771`, 0.8708699742516075`,
0.8725607541923411`, 0.8742561076245073`, 0.8759559581050231`,
0.8776600441685872`, 0.8793686531990167`, 0.881081799280628`,
0.8827994036379265`, 0.8845214793551222`, 0.8862478522883621`,

```
0.8879787213884999`, 0.8897140054677172`, 0.8914537166292315`,
0.8931977718950158`, 0.8949463732547783`, 0.8966993422971113`,
0.898456786321912`, 0.900218427728152`, 0.9019845663003077`,
0.9037551169215773`, 0.9055300907170031`, 0.9073094006902969`,
0.9090930568335965`, 0.9108810691413733`, 0.9126734476101613`,
0.9144700028594194`, 0.9162711428605446`, 0.9180766790184339`,
0.9198864195873795`, 0.9217005747245784`, 0.923519154225028`,
0.9253419645258342`, 0.9271691150829909`, 0.9290006148590343`,
0.9308363693177379`, 0.9326765933710268`, 0.9345210887562468`,
0.9363697584312098`, 0.9382230283505573`, 0.9400802775747821`,
0.9419418270685117`, 0.9438080027722632`, 0.9456781761683826`,
0.9475525648540183`, 0.949431497223499`, 0.9513144436784532`,
0.953201840859783`, 0.9550933710302967`, 0.9569892571253341`,
0.9588895062819971`, 0.9607936854270143`, 0.9627024592809129`,
0.964615171970324`, 0.9665323820185466`, 0.9684536504754523`,
0.9703793170102869`, 0.972309050395989`, 0.9742431928703761`,
0.9761812968236027`, 0.9781237057241042`, 0.9800704248965644`,
0.9820212296946004`, 0.9839762380158433`, 0.9859355695257509`,
0.9878989966509495`, 0.989866638472223`, 0.9918383813160084`,
0.9938143450517841`, 0.9957945331423718`, 0.9977788303044908`,
0.9997673576018764`, 1.0017599388478722`, 1.0037568753499224`,
1.0057578701923031`, 1.007762863722806`, 1.0097722206613697`,
1.0117855182229711`, 1.0138031844020017`, 1.0158247928536068`,
1.0178505278109198`, 1.0198806383591277`, 1.0219145682623383`,
1.0239528146743726`, 1.025995004201896`, 1.0280413873091925`,
1.0300919021463972`, 1.0321466127638417`, 1.0342052653728409`,
1.0362678580494993`, 1.0383348385910491`, 1.0404055650469903`,
1.0424805510629007`, 1.0445596030413309`, 1.0466427853602656`,
1.04873016333401`, 1.0508210792357873`, 1.052916516725637`,
1.0550158165441856`, 1.0571192412883772`, 1.0592266564337736`,
1.0613381937783621`, 1.0634537849168482`, 1.0655732925816235`,
1.0676967810727933`, 1.069824315645187`, 1.0719558940266687`,
1.074091582590222`, 1.076231310608597`, 1.078374936824574`,
1.0805224568279612`, 1.0826741455425206`, 1.0848297911879874`,
1.086989248942018`, 1.0891529372538613`, 1.0913205003188315`,
1.0934920754619093`, 1.0956675874718682`, 1.097847103545339`,
1.1000306918839735`, 1.1022179872275228`, 1.104409272731797`,
1.1066046170482986`, 1.1088037967300721`, 1.1110071732777376`,
1.1132143752430999`, 1.115425395975239`, 1.1176404520280148`,
1.1198595391019985`, 1.1220823525745094`, 1.1243089596359046`,
1.1265398067254038`, 1.1287744360001815`, 1.131012839732616`,
1.1332553162242827`, 1.135501707359184`, 1.1377519295995813`,
1.140006052955341`, 1.1422641485026401`, 1.1445259760800766`,
1.146791605213477`, 1.149061421291984`, 1.1513349466592295`,
1.153612409923538`, 1.1558936450988522`, 1.1581786434248786`,
1.1604676367747784`, 1.1627606188992152`, 1.1650570979653394`,
1.1673577114544866`, 1.1696619652814888`, 1.1719702580313194`,
```

```
1.174282172656523`, 1.176598028106844 7`, 1.1789177341460548`,
1.1812411158990528`, 1.1835684966721955`, 1.1858997857250804`,
1.188234638367919`, 1.1905735493118683`, 1.192916003213302`,
1.1952623280990964`, 1.1976126006896766`, 1.1999665555738577`,
1.202324353939036`, 1.2046858999854309`, 1.207051183217967`,
1.2094202801389897`, 1.2117931809071179`, 1.2141696118904732`,
1.2165499126375119`, 1.2189341623461067`, 1.2213219968856217`,
1.2237137604282746`, 1.2261090856897219`, 1.2285081393382837`,
1.2309110003449764`, 1.2333174769135453`, 1.2357279203299043`,
1.2381420477574103`, 1.2405598470915247`, 1.242981490208711`,
1.2454065045206923`, 1.24783570937472`, 1.2502680734889124`,
1.2527045136803399`, 1.2551445526066152`, 1.2575883653023416`,
1.2600358456318619`, 1.2624870757964093`, 1.2649418531578014`,
1.267400450628827`, 1.2698627609157063`, 1.2723284820604523`,
1.2747979854545577`, 1.2772715510438784`, 1.2797483893128008`,
1.2822290704039172`, 1.2847131895777921`, 1.287201125866425`,
1.2896925702149655`, 1.2921880049650436`, 1.2946867210237223`,
1.297189201852706`, 1.299695334326575`, 1.3022052057111562`,
1.304718498767933`, 1.3072358080603292`, 1.3097564081175592`,
1.3122804860373418`, 1.3148086440051547`, 1.3173400453968835`,
1.3198752943562506`, 1.3224137535565472`, 1.3249562407581208`,
1.3275022208493104`, 1.330051677673257`, 1.3326048065452396`,
1.3351616995529332`, 1.3377219168924745`, 1.340285975654035`,
1.342853326078481`, 1.345424165229465`, 1.3479990181596277`,
1.3505771133216773`, 1.35315897699369`, 1.3557441575548854`,
1.3583329663012755`, 1.3609254985419539`, 1.3635212969743171`,
1.3661206759950681`, 1.3687237315218148`, 1.3713306730443853`,
1.3739408131246715`, 1.3765546962375832`, 1.3791718548327896`,
1.381792610893874`, 1.384417062875342`, 1.387044622710449`,
1.3896758444944837`, 1.3923108286229926`, 1.39494921309481`,
1.397591212540756`, 1.4002364605629485`, 1.4028852890761905`,
1.4055380350769253`, 1.408193622325604`, 1.4108529752367671`,
1.4135158413765312`, 1.4161824426701242`, 1.4188521644764822`,
1.4215253464526252`, 1.4242023334478486`, 1.426882383353633`,
1.429566082816116`, 1.4322531714988618`, 1.4349439988176427`,
1.4376380571493401`, 1.4403355731778407`, 1.443036528816321`,
1.4457411549131731`, 1.4484490605663547`, 1.4511604766304933`,
1.4538753854968938`, 1.4565935164060657`, 1.4593151023207918`,
1.4620403797508839`, 1.4647689493585159`, 1.4675007909462394`,
1.4702361417546332`, 1.47297524262251`, 1.4757174293908604`,
1.478462940028069`, 1.48121227766762`, 1.4839646404003797`,
1.4867204001164407`, 1.489479537609429`, 1.4922424316167384`,
1.495008534012276`, 1.4977779571203753`, 1.5005509492067763`,
1.503327088741572`, 1.506106624009905`, 1.5088895356345888`,
1.511675940208071`, 1.5144656826783447`, 1.5172588801729998`,
1.5200553761628812`, 1.5228550118549156`, 1.5256584586019175`,
1.528465005234358`, 1.53127476869168`, 1.5340881482632256`,
```

```
1.5369045636417968`, 1.53972455648712`, 1.5425478255384963`,
1.5453744921893136`, 1.5482043939388905`, 1.551037796132819`,
1.5538745363188606`, 1.5567147385964715`, 1.559557659107785`,
1.5624041436841998`, 1.5652544664969232`, 1.5681075866661072`,
1.5709645075398209`, 1.5738243279321`, 1.5766874671077362`,
1.5795536069697176`, 1.582423768138524`, 1.5852970388105994`,
1.588173395972148`, 1.5910528163557782`, 1.5939360336062791`,
1.5968222722668275`, 1.5997116608582476`, 1.60260448293626`,
1.6055005653199308`, 1.6084000407631216`, 1.6113027345021795`,
1.6142084693042156`, 1.617117844801096`, 1.6200300614690029`,
1.6229458782522828`, 1.6258649622035115`, 1.6287872918752744`,
1.6317128455783652`, 1.6346416013805758`, 1.63757369694413`,
1.6405087907434137`, 1.6434471803640711`, 1.646389167748577`,
1.6493339242297276`, 1.6522820738221893`, 1.6552334330551692`,
1.6581881436675958`, 1.661146185021995`, 1.6641072061252482`,
1.6670716792840616`, 1.6700392524348744`, 1.6730099022123992`,
1.6759839398511622`, 1.6789611770370063`, 1.6819414228357525`,
1.6849251605388518`, 1.6879120313066074`, 1.690902352931335`,
1.6938952506320852`, 1.6968913819417966`, 1.6998912415946288`,
1.7028937757937836`, 1.7058996504257322`, 1.708908670565214`,
1.7119209879606883`, 1.7149365812821125`, 1.7179552530487758`,
1.7209773326702675`, 1.724002268497591`, 1.7270303900730537`,
1.7300614964648504`, 1.7330962791438076`, 1.7361336427985945`,
1.7391748205310174`, 1.7422185299798543`, 1.7452656494802292`,
1.7483163410399498`, 1.7513694892125713`, 1.7544263480345126`,
1.757485979942689`, 1.7605485427993945`, 1.7636147529515227`,
1.7666840345712564`, 1.7697561772155344`, 1.7728317169321794`,
1.7759100694191208`, 1.7789919627753614`, 1.7820764316867104`,
1.7851642070299898`, 1.7882554581219776`, 1.7913494003535886`,
1.7944469656674316`, 1.797547173086232`, 1.8006503809200705`,
1.8037571464772357`, 1.8068668672786838`, 1.8099793231107553`,
1.8130954664551397`, 1.8162142968877284`, 1.8193363785706471`,
1.8224616896371515`, 1.8255896120424469`, 1.8287211139740638`,
1.8318553768963615`, 1.8349927752943176`, 1.8381334875079856`,
1.841277088165933`, 1.8444235516783742`, 1.847573869516734`,
1.850726392759304`, 1.8538825227991906`, 1.8570414192275972`,
1.8602036739907801`, 1.8633688524307948`, 1.8665371370134323`,
1.8697082960509064`, 1.872882303862283`, 1.876059763872454`,
1.8792402353181277`, 1.8824236939303978`, 1.8856099032750882`,
1.888799474380792`, 1.8919919598425168`, 1.895187762601334`,
1.8983862171036998`, 1.9015879420678836`, 1.9047924833816745`,
1.9080002488317813`, 1.9112107809263688`, 1.9144242714423998`,
1.9176413532199146`, 1.9208611282340624`, 1.924083789748299`,
1.9273095337354091`, 1.9305390031863783`, 1.9337703980067102`,
1.9370050249260484`, 1.9402430864189821`, 1.9434838872003826`,
1.9467274004068615`, 1.9499747321303627`, 1.9532240471766746`,
1.9564769082255402`, 1.9597326098017842`, 1.9629913557886263`,
```

1.9662533528834945`, 1.9695178857907987`, 1.9727856216270117`,
1.9760565382578206`, 1.9793301462953545`, 1.982606652808176`,
1.9858862677205842`, 1.9891689679882727`, 1.9924544937185702`,
1.995743056474983`, 1.999034394339816`, 2.002328958946758`,
2.0056262489426278`, 2.008926598368991`, 2.012229983759761`,
2.0155362603543883`, 2.0188454030403826`, 2.0221578739327293`,
2.0254729186577918`, 2.0287908770039147`, 2.0321118471976214`,
2.0354358053747093`, 2.038762603569739`, 2.042092589089813`,
2.045425116392481`, 2.0487606574144928`, 2.0520995648716926`,
2.055441062974132`, 2.0587855027632553`, 2.062132986506459`,
2.065483490748964`, 2.068836736700848`, 2.072192825947616`,
2.075552246657101`, 2.0789139478001926`, 2.082279319957206`,
2.0856470494633976`, 2.089017756546836`, 2.0923915476800845`,
2.0957677454041606`, 2.099147502094178`, 2.102529481966314`,
2.105914711112941`, 2.109302903035801`, 2.112693767614737`,
2.1160875448844556`, 2.1194846119527817`, 2.122884276395797`,
2.1262869148794064`, 2.1296923683303577`, 2.133100475723469`,
2.136511890299034`, 2.1399256360862045`, 2.143343052158119`,
2.14676261054847`, 2.150185381734374`, 2.1536110677521916`,
2.1570396442376816`, 2.160470808393547`, 2.163904951405308`,
2.167342189562296`, 2.170782219404424`, 2.1742250154605536`,
2.177670410705013`, 2.1811186614289464`, 2.1845700272830104`,
2.1880244859441094`, 2.1914811561106196`, 2.1949410119650676`,
2.198403599795323`, 2.2018694716506557`, 2.2053378821483767`,
2.2088089489775258`, 2.2122832290005405`, 2.215759969338725`,
2.2192398754850102`, 2.222722336773538`, 2.226207916543954`,
2.22969614817208`, 2.2331871537686028`, 2.2366809079871692`,
2.2401776843814947`, 2.243677309943507`, 2.2471793085897214`,
2.2506847080439245`, 2.2541927317361785`, 2.25770320110121`,
2.261217002132811`, 2.2647333505533824`, 2.2682526790406996`,
2.2717745029127006`, 2.2752991029792513`, 2.278827073181979`,
2.2823574624761633`, 2.285890398164499`, 2.2894266340799083`,
2.2929653665823837`, 2.2965068823825`, 2.3000513141874612`,
2.3035983219936433`, 2.307148672138001`, 2.3107013901423143`,
2.314256766496471`, 2.317814935283826`, 2.321376353536487`,
2.324940356354155`, 2.328507240249944`, 2.3320764950719064`,
2.3356487425532753`, 2.3392236337043526`, 2.3428014696466795`,
2.3463818989812872`, 2.3499650598297017`, 2.3535509270257804`,
2.3571399719470656`, 2.3607311770099235`, 2.3643258446363244`,
2.367922953800833`, 2.371522644378382`, 2.3751253943036796`,
2.3787306756319144`, 2.3823386306564283`, 2.3859495731634257`,
2.3895631401563215`, 2.393179305691662`, 2.396798728428714`,
2.4004207013267385`, 2.404045370058378`, 2.407672882162742`,
2.4113025206460716`, 2.4149352979985603`, 2.4185703214377976`,
2.4222091354382522`, 2.4258499709840153`, 2.4294936764034865`,
2.433140228186081`, 2.4367894256581457`, 2.4404410658234936`,
2.444096011514506`, 2.447753350407714`, 2.451413054786338`,

```
    2.4550759948695324`, 2.4587416100571304`, 2.4624095138165125`,
    2.466080402564032`, 2.469753890434254`, 2.473430680996271`,
    2.4771094747368574`, 2.4807911576910855`, 2.4844753388602583`,
    2.4881627297093956`, 2.491852384785058`, 2.495545017887831`,
    2.499240606796545`, 2.5029383822904583`, 2.5066386896500927`,
    2.510342253983604`, 2.514048113798885`, 2.517757374422709`,
    2.521468501662522`, 2.525182603782397`, 2.5288990868903136`,
    2.532618688272271`, 2.536340620248769`, 2.54006523963883`,
    2.5437929075685015`, 2.547523022428528`, 2.5512559455230157`,
    2.5549920424984873`, 2.5587301224177303`, 2.5624711330506713`,
    2.566214267031523`, 2.5699608728226577`, 2.573708563971336`,
    2.5774604698208874`, 2.5812139980708455`, 2.584971305600935`,
    2.5887305801416045`, 2.5924925922838296`, 2.596257920829158`,
    2.6000255396490655`, 2.6037954210138032`, 2.607568144901646`,
    2.6113430787769665`, 2.6151214171820945`, 2.618901915760546`,
    2.622685365754145`, 2.62647133414948`, 2.6302602077262685`,
    2.63405113717496`, 2.6378445076954353`, 2.641641540927754`,
    2.6454405527396876`, 2.6492423496552218`, 2.653046488992695`,
    2.6568535749201447`, 2.6606633747519983`, 2.664476076363724`,
    2.6682914454981015`, 2.672108819701074`, 2.675929236243932`,
    2.6797520322895023`, 2.6835776106101763`, 2.6874055177361753`,
    2.6912363743575973`, 2.6950695095533015`, 2.698905547324065`,
    2.702744465784311`, 2.706585369591678`, 2.7104288858821866`}[[s]];
HP2rho540[s_] := {0.6558877334192567`, 0.6558902462709697`,
   0.6558978362257051`, 0.655910606208102`, 0.6559283516678397`,
   0.6559511759973676`, 0.6559790802569926`, 0.6560120144412283`,
   0.6560500826807197`, 0.656093235447897`, 0.656141423426462`,
   0.6561947514952028`, 0.6562530168005181`, 0.656316530067845`,
   0.6563849861799816`, 0.6564584395983842`, 0.6565372533624355`,
   0.6566208659581552`, 0.6567096409229781`, 0.6568034795847029`,
   0.656902437716059`, 0.6570063655605747`, 0.6571154222459592`,
   0.6572295613718053`, 0.6573487882165997`, 0.6574730052356547`,
   0.6576023211648093`, 0.6577366904222979`, 0.6578762223607904`,
   0.6580207170509693`, 0.6581702842747462`, 0.6583249825718974`,
   0.6584847674421251`, 0.6586494910950966`, 0.658819264370239`,
   0.658994146783842`, 0.6591741464046338`, 0.6593592197173418`,
   0.6595493233308412`, 0.659744465858457`, 0.6599446561495413`,
   0.660149851339331`, 0.6603601646214934`, 0.6605755536407848`,
   0.6607959240952703`, 0.6610215462590925`, 0.6612520137440272`,
   0.661487649498148`, 0.6617281512026005`, 0.6619739471999616`,
   0.6622245785926553`, 0.6624803700712936`, 0.6627412286575146`,
   0.6630070612879836`, 0.663278037022749`, 0.6635539582617792`,
   0.6638349946734031`, 0.6641210012331424`, 0.6644120431082504`,
   0.6647081858886372`, 0.6650092847213062`, 0.6653155637898824`,
   0.6656267201351216`, 0.6659429785181477`, 0.6662642472363316`,
   0.6665905403608879`, 0.6669219252262027`, 0.6672582573190253`,
   0.6675997105049244`, 0.6679461404341765`, 0.668297508820362`,
```

```
0.6686539905818127`, 0.669015494784761`, 0.6693820371087505`,
0.6697535265300739`, 0.6701299788574457`, 0.6705116781112188`,
0.6708983195647059`, 0.6712897584116201`, 0.6716864409324299`,
0.6720880079843524`, 0.6724946919018283`, 0.6729061325477662`,
0.6733228328911862`, 0.6737443786419557`, 0.6741710579442906`,
0.6746026723041392`, 0.6750392939671569`, 0.6754809413112576`,
0.6759275240332512`, 0.6763790605704829`, 0.6768355696012156`,
0.6772971247316805`, 0.677763690593729`, 0.6782352317665332`,
0.6787117676838001`, 0.6791931530503627`, 0.6796796824372577`,
0.6801709904422145`, 0.6806675381306043`, 0.6811688490964692`,
0.6816752197650032`, 0.6821865601097145`, 0.6827028908109382`,
0.6832241215040853`, 0.6837502728526638`, 0.6842814773992206`,
0.6848175331930381`, 0.6853586290879664`, 0.685904674827643`,
0.6864556358644618`, 0.6870115337993283`, 0.6875724468346174`,
0.6881382279920215`, 0.6887089555970091`, 0.6892846519562006`,
0.6898652828912899`, 0.6904508709190792`, 0.691041324950163`,
0.6916367244354743`, 0.6922370352154605`, 0.6928423374284164`,
0.6934525398875776`, 0.6940677231264343`, 0.694687680688105`,
0.6953126081425748`, 0.6959424715980976`, 0.6965772369943061`,
0.697216928044499`, 0.6978613945401378`, 0.6985109505388836`,
0.6991652132341015`, 0.6998245559881636`, 0.7004887116175582`,
0.7011578212088546`, 0.701831733396227`, 0.7025105310319992`,
0.7031942976651836`, 0.7038828813037235`, 0.7045761879916264`,
0.7052746567395047`, 0.7059777794703109`, 0.7066858775592866`,
0.7073989170868475`, 0.7081166248043176`, 0.7088393841419752`,
0.7095668614676193`, 0.7102992015894375`, 0.7110364300082075`,
0.7117784516801268`, 0.7125254128810127`, 0.7132771579002771`,
0.7140338336258322`, 0.7147952227736271`, 0.7155614114133838`,
0.7163325475424008`, 0.7171084739442544`, 0.7178892777982216`,
0.7186748623474993`, 0.7194651299977889`, 0.7202604770426744`,
0.7210603735987596`, 0.7218654032011911`, 0.7226749717149424`,
0.7234894156374853`, 0.7243086992684995`, 0.7251327865841141`,
0.7259615784037065`, 0.7267951005708237`, 0.7276336316201035`,
0.7284767567800395`, 0.7293248183210365`, 0.7301774625665375`,
0.7310350332819926`, 0.7318973665993861`, 0.732764297083813`,
0.7336361709941038`, 0.7345127590451781`, 0.7353940876075966`,
0.7362802479051809`, 0.7371710728954689`, 0.7380666536391096`,
0.7389670169575705`, 0.7398721899044048`, 0.7407819380993856`,
0.7416965494423665`, 0.7426158544606749`, 0.7435398137686537`,
0.7444685196862064`, 0.7454019329462346`, 0.7463401466485902`,
0.747282921938825`, 0.7482306181463992`, 0.7491829291197318`,
0.750139813852186`, 0.7511015671259115`, 0.7520680821520139`,
0.7530389805876707`, 0.754014693412054`, 0.7549951803693881`,
0.7559803325982255`, 0.756970176652375`, 0.757964807790605`,
0.7589637728731193`, 0.7599675777916026`, 0.7609761124522882`,
0.7619892655230207`, 0.7630069938147975`, 0.7640293927800084`,
0.7650565588265605`, 0.7660883094794365`, 0.7671246005191891`,
```

```
0.7681655981990259`, 0.7692110473001164`, 0.7702614677357479`,
0.7713163921449492`, 0.7723758457606885`, 0.7734399253243847`,
0.7745086570551951`, 0.7755819956715528`, 0.7766598953029462`,
0.7777424536960161`, 0.7788295526752546`, 0.7799212176748503`,
0.7810176197578295`, 0.7821184942979321`, 0.7832240123484224`,
0.7843340534696038`, 0.7854486428444288`, 0.7865678796012011`,
0.7876915680480318`, 0.7888198810329933`, 0.7899528445140254`,
0.791090111628229`, 0.7922320795982537`, 0.7933786999311755`,
0.7945297729588302`, 0.7956852475300022`, 0.7968454502596162`,
0.7980099523312083`, 0.7991791573024609`, 0.8003528624804732`,
0.801531015845907`, 0.8027137950931599`, 0.803900994712252`,
0.8050926384470862`, 0.8062889052095772`, 0.807489587287023`,
0.808694708253776`, 0.8099044482428626`, 0.8111187541469164`,
0.8123371787757916`, 0.8135604529655417`, 0.814787970879172`,
0.8160199924026535`, 0.8172564619763625`, 0.8184975628337712`,
0.8197429195781816`, 0.8209928752613862`, 0.8222469710511237`,
0.8235057933281821`, 0.8247689624611808`, 0.8260365820634648`,
0.827308675312277`, 0.828585101859292`, 0.8298659657834696`,
0.8311512076282815`, 0.8324408496791048`, 0.833734997236987`,
0.8350335074220816`, 0.8363364022035887`, 0.837643703698526`,
0.8389553502679752`, 0.8402714477064095`, 0.8415918497114988`,
0.8429166621030412`, 0.8442459069420204`, 0.845579350731235`,
0.8469172699135062`, 0.8482594296546303`, 0.849606108294038`,
0.8509569833265909`, 0.8523123341830099`, 0.8536719224879741`,
0.8550359424411301`, 0.8564042408743102`, 0.8577769256529619`,
0.8591538419152703`, 0.8605350092238194`, 0.8619205358233647`,
0.8633104423764552`, 0.864704482273039`, 0.8661031209562456`,
0.8675057532207746`, 0.8689127559017926`, 0.8703240591262862`,
0.8717397731541692`, 0.8731596460025187`, 0.8745837869881833`,
0.876012123763981`, 0.8774447664600232`, 0.8788817344356067`,
0.8803229547827204`, 0.8817684461828595`, 0.8832180414427503`,
0.8846719442220387`, 0.8861300799181118`, 0.8875923728032543`,
0.8890590286896826`, 0.8905298778296001`, 0.8920048429332454`,
0.8934840358820529`, 0.8949675699124067`, 0.8964551764289986`,
0.8979471598324296`, 0.8994430569326133`, 0.9009431723058886`,
0.9024476203057944`, 0.9039561270377948`, 0.9054690012142206`,
0.9069858691074654`, 0.908507040581697`, 0.9100322371935373`,
0.91156157283613`, 0.9130950634178455`, 0.9146329243769624`,
0.9161746734944474`, 0.9177206250856745`, 0.9192706946390502`,
0.9208248974410651`, 0.9223831474296426`, 0.9239456625393183`,
0.9255123571989282`, 0.9270829396168822`, 0.9286578335433954`,
0.9302366431226362`, 0.9318195880096353`, 0.933406786870985`,
0.9349978387822993`, 0.9365930692000335`, 0.9381922832679025`,
0.9397957044769595`, 0.9414032423088116`, 0.9430146988563459`,
0.9446301924854191`, 0.9462498430645354`, 0.9478734506646611`,
0.9495011348930751`, 0.9511329090357427`, 0.9527685700023454`,
0.9544084547087621`, 0.9560522508281367`, 0.957699969597399`,
```

```
0.9593518417179616`, 0.9610077706762653`, 0.9626675480151298`,
0.9643315165485595`, 0.9659992450367152`, 0.9676710775103955`,
0.9693469146079745`, 0.9710268802616628`, 0.972710423249109`,
0.9743983427600817`, 0.9760900860730837`, 0.9777857764627608`,
0.9794853100696356`, 0.9811889257672323`, 0.9828965201446961`,
0.9846078726534656`, 0.9863233394007968`, 0.9880425831035763`,
0.989765962329557`, 0.991493020220122`, 0.9932243517928484`,
0.9949592621704362`, 0.9966983492696874`, 0.9984409120670359`,
1.0001876706354118`, 1.0019382780045818`, 1.0036928625715265`,
1.0054511326296844`, 1.0072135783393694`, 1.0089796650846472`,
1.0107498248967905`, 1.0125237624987922`, 1.0143015462685012`,
1.016083245440872`, 1.0178686213472519`, 1.0196579276152473`,
1.0214511725897835`, 1.0232481149867965`, 1.025048948405088`,
1.026853617679845`, 1.0286622559694691`, 1.0304745551548309`,
1.0322906472668831`, 1.0341106023664475`, 1.0359343634707707`,
1.0377620010678723`, 1.0395932000314037`, 1.041428545712209`,
1.0432675287924016`, 1.0451102185570986`, 1.0469566851819876`,
1.0488069341768191`, 1.0506609052338234`, 1.0525188012117643`,
1.054380297639738`, 1.056245464331855`, 1.0581144387401757`,
1.0599874941155563`, 1.0618640337857912`, 1.063744261849798`,
1.065628385225704`, 1.0675159344283742`, 1.0694075238610514`,
1.0713028188687603`, 1.073201548583606`, 1.0751040582492941`,
1.0770105596646666`, 1.0789204349349644`, 1.0808343106263771`,
1.0827514944675964`, 1.0846726166328076`, 1.0865974013189`,
1.0885257805742725`, 1.0904581106530673`, 1.092393828135723`,
1.0943332180548764`, 1.096276354531091`, 1.0982230251010476`,
1.1001735192030169`, 1.1021275511129132`, 1.104085194241668`,
1.1060465958899133`, 1.1080114663213958`, 1.1099800256449257`,
1.1119521285486802`, 1.1139279973702156`, 1.115907337982344`,
1.1178905223667177`, 1.1198768812114086`, 1.1218673109580626`,
1.1238610645065303`, 1.1258584417493025`, 1.1278594435113523`,
1.1298639944653068`, 1.13187217086199`, 1.133883973337275`,
1.135899017934272`, 1.1379180728526779`, 1.1399405230083335`,
1.1419663663005484`, 1.143995834564117`, 1.1460288497453044`,
1.1480654897339209`, 1.1501056757810415`, 1.1521493279478696`,
1.1541965239022993`, 1.1562471829889056`, 1.158301463250689`,
1.1603590434011715`, 1.1624204834462355`, 1.1644850581075907`,
1.166553248399078`, 1.1686250537392038`, 1.1707002284011834`,
1.1727790969654304`, 1.1748611658587513`, 1.1769469251964697`,
1.1790359605906138`, 1.181128516913251`, 1.1832246755836053`,
1.1853242678895854`, 1.1874272067021077`, 1.1895336566263197`,
1.1916435306458046`, 1.1937568253948376`, 1.19587379313781`,
1.197993834234977`, 1.2001175423351267`, 1.2022444856692756`,
1.2043752637700056`, 1.2065088361233747`, 1.2086462371077353`,
1.2107869429239337`, 1.2129311237936544`, 1.2150785123117003`,
1.2172294554272447`, 1.2193840391029338`, 1.2215415500075648`,
1.2237025142142264`, 1.2258670173900337`, 1.2280346969317901`,
```

```
1.2302059969075496`, 1.2323807339729211`, 1.2345583587509612`,
1.236739590386978`, 1.238924334798431`, 1.2411122210806895`,
1.2433034258619198`, 1.2454981282505806`, 1.2476960459097448`,
1.2498974511717833`, 1.2521020595197936`, 1.2543100512550678`,
1.2565213268850162`, 1.258735879917978`, 1.2609540828321248`,
1.2631749818529636`, 1.265399709407568`, 1.2676274025827623`,
1.2698585318453592`, 1.2720928026648322`, 1.274330303498928 6`,
1.27657141544054`, 1.2788156476481018`, 1.2810632848721408`,
1.2833141250472049`, 1.2855680617420269`, 1.2878252835758295`,
1.2900858826113089`, 1.2923499521835904`, 1.2946168876150117`,
1.2968871784521456`, 1.2991606169140837`, 1.3014373960975378`,
1.303717509181738`, 1.3060006442344982`, 1.3082869950289322`,
1.3105766556782852`, 1.3128697216011749`, 1.3151656709571071`,
1.3174651125956036`, 1.3197675218967908`, 1.3220733045024144`,
1.3243821399141156`, 1.3266941227740479`, 1.3290096648818965`,
1.3313279160403706`, 1.3336496043538564`, 1.3359746170479696`,
1.33830262623652`, 1.3406337279509863`, 1.3429680195699227`,
1.3453054919543315`, 1.3476461358308927`, 1.3499899417911336`,
1.3523369002905992`, 1.354687111033093`, 1.3570401262790588`,
1.3593964833756222`, 1.3617560639855588`, 1.3641186371965979`,
1.366484414076343`, 1.3688536086333352`, 1.3712254291899952`,
1.3736005350562464`, 1.3759789176203818`, 1.3783601151737856`,
1.3807445686708095`, 1.3831321550631368`, 1.3855228640123969`,
1.3879162257702042`, 1.390313377092104`, 1.392713042630815`,
1.3951161335864917`, 1.397522293263158`, 1.3999315104613432`,
1.4023435394019859`, 1.4047587190292663`, 1.4071771569365217`,
1.4095986069039952`, 1.4120228193146613`, 1.4144504951229613`,
1.4168810287080875`, 1.419314646593765`, 1.4217509760270635`,
1.4241907272175498`, 1.4266335283559428`, 1.4290792459683141`,
1.4315278669075924`, 1.433979622999072`, 1.4364342573301554`,
1.438892003098505`, 1.4413530967214772`, 1.4438165353375938`,
1.4462836712218945`, 1.4487536233485958`, 1.4512262517973582`,
1.4537015407287668`, 1.4561804852454805`, 1.4586623185868068`,
1.4611471542703145`, 1.463635107388274`, 1.466125782132686`,
1.468619420103694`, 1.4711158792450116`, 1.4736154038916403`,
1.4761178521120875`, 1.4786233406903673`, 1.4811318572644092`,
1.4836432581189447`, 1.4861576608992764`, 1.488675052782199`,
1.4911951557090966`, 1.493718352823004`, 1.4962446321788965`,
1.4987737139004045`, 1.5013057173714057`, 1.5038406288280126`,
1.5063784343473205`, 1.5089192555566537`, 1.511463079586421`,
1.5140094835315123`, 1.516558998420456`, 1.5191116122938797`,
1.5216666229802713`, 1.5242248421193403`, 1.5267859807960698`,
1.5293498856261192`, 1.5319166809489708`, 1.5344862121187455`,
1.537059026539373`, 1.53963412479805`, 1.5422126205547628`,
1.544793793354977`, 1.547377769382716`, 1.54996510614034`,
1.5525547882284974`, 1.5551473723473757`, 1.5577428441732188`,
1.5603410441046197`, 1.5629425384498292`, 1.56554644024499`,
```

```
1.568153169868143`, 1.5707628592440643`, 1.573375494539678`,
1.5759907656791075`, 1.5786089526308618`, 1.5812301901089798`,
1.5838538670289806`, 1.5864808644143222`, 1.5891101183155252`,
1.591742060841581`, 1.5943775844630537`, 1.5970154652541046`,
1.5996561408098806`, 1.6022995955097987`, 1.6049458135675247`,
1.6075949330701218`, 1.6102466303261291`, 1.6129011976384668`,
1.6155587756538996`, 1.6182193508262486`, 1.6208822830678866`,
1.6235477094501511`, 1.626216557069463`, 1.6288878675096345`,
1.6315613044192403`, 1.6342385965424955`, 1.6369181414871499`,
1.6396003988466947`, 1.642285673564367`, 1.6449734681951707`,
1.647663764247954`, 1.6503573547719965`, 1.6530532516312322`,
1.6557519244401806`, 1.6584535211459739`, 1.6611578627192538`,
1.6638646028148374`, 1.6665745494684254`, 1.6692868604229687`,
1.6720023487658118`, 1.6747198321728134`, 1.6774407964106808`,
1.6801640544860064`, 1.6828897544358725`, 1.6856185543653461`,
1.6883501015593363`, 1.691084038748983`, 1.6938206881122315`,
1.6965605476330265`, 1.6993029169758855`, 1.7020477774496436`,
1.7047951101858392`, 1.707545243717725`, 1.7102983365353228`,
1.7130536746503995`, 1.7158119387146906`, 1.7185731140651983`,
1.7213364794489825`, 1.724103076133529`, 1.7268718259090226`,
1.7296435982103155`, 1.7324176637207582`, 1.7351947194992543`,
1.7379745714296073`, 1.7407566617039607`, 1.7435415125789127`,
1.7463291077577463`, 1.7491192484253228`, 1.7519122820867212`,
1.7547074596122834`, 1.757505679319525`, 1.7603063735133802`,
1.7631095231384522`, 1.7659158524615615`, 1.7687236709784089`,
1.771535007877142`, 1.7743491034771592`, 1.7771653766838735`,
1.7799843721569766`, 1.7828060732153546`, 1.7856302729668507`,
1.788457333897178`, 1.791286666878781`, 1.7941186346608948`,
1.796953219730907`, 1.799790597483737`, 1.8026305582184046`,
1.805472889629851`, 1.8083177666091854`, 1.8111649752818082`,
1.814015672106481`, 1.8168680768465633`, 1.819723347475588`,
1.8225814686795303`, 1.8254420277563281`, 1.8283048061010565`,
1.8311703818962155`, 1.8340385386911886`, 1.836909459606085`,
1.8397827248633065`, 1.8426581118582426`, 1.8455364103464431`,
1.8484174018044341`, 1.851300660730487`, 1.8541867809689523`,
1.8570757471708657`, 1.8599663066291863`, 1.862860086873873`,
1.8657562459826205`, 1.8686547641005489`, 1.8715562475339715`,
1.8744596347794764`, 1.8773663721092098`, 1.8802749736460054`,
1.883186683054681`, 1.8861002150825816`, 1.8890166082021471`,
1.8919354203934187`, 1.8948564182197072`, 1.897780868122047`,
1.9007070376009323`, 1.9036357647446476`, 1.9065672481032871`,
1.9095012540721636`, 1.9124375462158858`, 1.9153765412606092`,
1.9183182224507003`, 1.9212619128271125`, 1.9242080306236111`,
1.9271569993022948`, 1.930108581446554`, 1.9330625367694247`,
1.936019068913406`, 1.938977487717579`, 1.9419384433000513`,
1.9449023683637308`, 1.9478685701978902`, 1.9508372550351116`,
1.9538081766349351`, 1.9567819987237984`, 1.9597577907458383`,
```

```
1.9627359885594917`, 1.965716803106301`, 1.9687002169449903`,
1.9716862124768042`, 1.97467477194489`, 1.9776656443102678`,
1.9806588093793027`, 1.983654246788619`, 1.9866524064981654`,
1.9896525641959177`, 1.9926554068892468`, 1.995661155206126`,
1.9986690800808273`, 2.001679398967019`, 2.004692092483499`,
2.007706780694934`, 2.0107240430936826`, 2.0137443453815758`,
2.016767067965329`, 2.0197913406759755`, 2.0228185999254036`,
2.0258480978832956`, 2.0288804272336414`, 2.0319145882613117`,
2.034951667336202`, 2.0379906596435426`, 2.041032162357353`,
2.0440764063170005`, 2.0471225011398007`, 2.050171550114788`,
2.0532226595664937`, 2.0562761853326044`, 2.0593324875739842`,
2.062391043459688`, 2.0654517054068786`, 2.0685148342241155`,
2.0715802835066266`, 2.074648418272524`, 2.077718836011926`,
2.0807916456367037`, 2.0838664397549946`, 2.086943585019901`,
2.090023843056765`, 2.0931053732181404`, 2.096189717907623`,
2.099277124474069`, 2.1023661308001507`, 2.1054581640483057`,
2.1085518857704177`, 2.111648200943452`, 2.1147466924814458`,
2.1178474731420773`, 2.1209511933404737`, 2.1240568977090564`,
2.127164833373071`, 2.1302756561779104`, 2.1333882663884407`,
2.1365029118184644`, 2.1396202525834016`, 2.142739861932546`,
2.145862131572002`, 2.1489867696899805`, 2.1521137571879443`,
2.1552426595097653`, 2.1583740089676153`, 2.16150792643624`,
2.164643556746006`, 2.167782275645919`, 2.170922946749603`,
2.1740659700078133`, 2.1772116088253735`, 2.1803587129304316`,
2.183508818486014`, 2.1866607720428`, 2.1898159792509873`,
2.192972995617836`, 2.196131941101958`, 2.1992933695966714`,
2.202457263204104`, 2.2056230239521373`, 2.208791501034671`,
2.2119623869716207`, 2.2151352245602878`, 2.2183104313392707`,
2.221488135576164`, 2.2246681720807806`, 2.2278506696645395`,
2.231034720288471`, 2.234221190776488 6`, 2.2374100628279114`,
2.2406016172269156`, 2.243794787634318`, 2.246990602190282`,
2.2501894972614016`, 2.253389795621372`, 2.2565920773749464`,
2.259797385918035`, 2.263004334615218`, 2.2662139700154995`,
2.269425201779698`, 2.2726393900621633`, 2.275855749473248`,
2.279074104953292`, 2.2822945900098355`, 2.2855182737958972`,
2.288743271437883`, 2.2919708079055514`, 2.295200395146419`,
2.298432484295149`, 2.3016667422355073`, 2.3049039404461267`,
2.3081420012531795`, 2.3113837594351243`, 2.3146261741296086`,
2.3178720945373485`, 2.3211199062211563`, 2.3243700698084813`,
2.3276219209299933`, 2.3308760831671775`, 2.3341328624410416`,
2.3373915912246335`, 2.340652574241727`, 2.3439162834351643`,
2.347182046757805`, 2.3504498432971888`, 2.353719817092076`,
2.3569919483005384`, 2.360266382956274`, 2.3635431023030335`,
2.3668217535444733`, 2.3701031519126574`, 2.373386610888013`,
2.37667194140234`, 2.3799603028243097`, 2.383249651242745`,
2.386541824222227`, 2.389836296825613`, 2.393132709032858`,
2.3964320661250755`, 2.3997331534522126`, 2.403035087197883`,
```

```
    2.406341288151249`, 2.409648466157923`, 2.412957979878085`,
    2.416270333055613`, 2.4195842902829208`, 2.422900877017593`,
    2.4262195511703464`, 2.4295402924953917`, 2.43286308059986`,
    2.436188425574872`, 2.4395159563620608`, 2.442845120051253`,
    2.4461769627394028`, 2.4495107538663943`, 2.4528464721651857`,
    2.45618553456081`, 2.4595257678442515`, 2.46286786752622`,
    2.466213080595194`, 2.469560304842591`, 2.4729093373566786`,
    2.4762605213334323`, 2.479613837299699`, 2.4829690819052535`,
    2.4863273392972114`, 2.4896871191996004`, 2.493049138009304`,
    2.4964128198529596`, 2.4997798174843022`, 2.5031480669036297`,
    2.5065182904232497`, 2.5098914059953272`, 2.5132658943258477`,
    2.516643428603144`, 2.5200219149111143`, 2.523403601530567`,
    2.526786955458969`, 2.530173289076607`, 2.5335606768417533`,
    2.536951008835967`, 2.5403427336564355`, 2.543737368752339`,
    2.5471339342965718`, 2.5505322155040475`, 2.5539321897642937`,
    2.5573353935646126`, 2.5607398622023134`, 2.5641475289735234`,
    2.567556418521459`, 2.5709676863236073`, 2.5743807226371804`,
    2.5777964955970827`, 2.5812143951967292`, 2.5846344022626546`,
    2.5880556990174863`, 2.591480261128011`, 2.5949060717808563`,
    2.5983349157464706`, 2.6017649654999584`, 2.6051972060136004`,
    2.6086318215555235`, 2.612068795061966`, 2.6155079054684642`,
    2.6189485206144125`, 2.6223914363930163`, 2.62583663521609`,
    2.629283893339928`, 2.632734017570688`, 2.636184716594936`,
    2.6396386587781135`, 2.643094168450224`, 2.646552266052611`,
    2.650011471336073`, 2.653473855821763`, 2.6569375155000543`,
    2.660403690487574`, 2.6638721535661163`, 2.6673430992626943`,
    2.670814810899847`, 2.6742893905067375`, 2.677766611471793`,
    2.6812447450948493`, 2.6847259112880493`, 2.688209021623565`,
    2.691693840092149`, 2.6951814272148935`, 2.698670250048839`,
    2.7021615886965833`, 2.705655427597959`, 2.7091502199369866`,
    2.7126474731449015`, 2.7161467315552463`, 2.71964885744041`,
    2.7231520700242147`, 2.7266583368979718`, 2.7301652032174752`,
    2.7336757566963765`, 2.7371879813918643`, 2.740701855353019`,
    2.744217132056041`, 2.7477358126237648`, 2.7512563094250266`,
    2.7547786017030904`, 2.758303348795603`, 2.761828716311066`,
    2.7653576371355864`, 2.7688875905855403`, 2.772420611625096`,
    2.7759553112853697`, 2.7794918981264805`, 2.7830310445712185`,
    2.7865718100624557`, 2.7901146365297587`, 2.793659272703809`,
    2.7972063975623254`, 2.800755761950344`, 2.804306879857214`,
    2.807859260397937`, 2.811414998941355`, 2.8149721975367927`,
    2.818531780061257`, 2.8220923063103993`, 2.8256563667481944`}[[s]];
HP2rho720[s_] := {0.7362689388121301`, 0.7362713944713782`,
    0.7362785030510159`, 0.7362904586990564`, 0.7363071326295152`,
    0.7363283962177865`, 0.736354638083524`, 0.7363855360766769`,
    0.7364210913692928`, 0.7364615639350325`, 0.7365066320740843`,
    0.736556556164557`, 0.7366112088153712`, 0.7366706568232072`,
    0.7367348378040535`, 0.7368037542296392`, 0.7368774087547832`,
```

```
0.7369558689606811`, 0.7370388788823968`, 0.7371268302276017`,
0.737219596949039`, 0.7373169234490571`, 0.7374192022771486`,
0.7375261133283061`, 0.7376377903457879`, 0.7377541078791318`,
0.7378752650172637`, 0.7380012015843882`, 0.7381318575051209`,
0.7382672377702109`, 0.7384073475536572`, 0.7385523222608699`,
0.7387019724414039`, 0.738856238668147`, 0.7390154522146348`,
0.7391792937648471`, 0.7393478345941056`, 0.7395211462769599`,
0.7396994311481714`, 0.7398822395356544`, 0.7400697086667545`,
0.7402620415349741`, 0.7404591149312627`, 0.740661001744683`,
0.7408672517398035`, 0.7410786577483355`, 0.7412946389318142`,
0.7415152687329529`, 0.7417407521013801`, 0.7419709665196159`,
0.742205920671653`, 0.7424454920061905`, 0.7426898865859812`,
0.7429389821889126`, 0.743192853958482`, 0.7434513797388378`,
0.7437146350002665`, 0.7439827615688561`, 0.7442553735903602`,
0.7445328772024279`, 0.7448150847330012`, 0.745102006646312`,
0.7453935873607187`, 0.745689903843376`, 0.7459909007800856`,
0.746296788407367`, 0.7466071797613985`, 0.7469223517943402`,
0.7472423828534377`, 0.7475668187446725`, 0.7478962040242779`,
0.7482302177113629`, 0.7485690054212886`, 0.7489124459994967`,
0.7492606187579538`, 0.7496134694740522`, 0.7499710109114304`,
0.7503332560215739`, 0.7507002179450167`, 0.7510719100125636`,
0.7514482784320773`, 0.7518293367137509`, 0.7522150311046348`,
0.7526055103274342`, 0.7530005858939677`, 0.7534005423048918`,
0.7538048525590709`, 0.754214140292363 6`, 0.7546280137053832`,
0.755046555129255`, 0.7554697113095471`, 0.7558975651013297`,
0.7563302680495171`, 0.7567674946270967`, 0.7572096014223099`,
0.7576560572099053`, 0.7581073560545415`, 0.7585633085431567`,
0.7590239991820363`, 0.759489237953783`, 0.7599591782563063`,
0.7604338364904732`, 0.7609130912103401`, 0.7613970277897034`,
0.7618855937495033`, 0.7623789443071106`, 0.762876749951346`,
0.7633793742344465`, 0.7638864175576967`, 0.7643984534537134`,
0.7649148729141555`, 0.7654360415407837`, 0.7659616975687775`,
0.7664922082138289`, 0.7670272414833276`, 0.7675670254852579`,
0.7681112270438331`, 0.7686602155079795`, 0.7692138686874337`,
0.769771922502763`, 0.7703348188232749`, 0.7709023645909073`,
0.7714744366451387`, 0.7720510532836758`, 0.7726324464791864`,
0.7732183507880163`, 0.7738088561299521`, 0.7744041245385692`,
0.7750038179198236`, 0.7756081698499462`, 0.776217128134462`,
0.7768307122599042`, 0.777448797798908`, 0.7780716204383895`,
0.7786989837359507`, 0.7793309072517579`, 0.7799674832548804`,
0.7806085867607101`, 0.781254237540699`, 0.7819046013167446`,
0.7825594069547586`, 0.7832188934408398`, 0.7838829351765146`,
0.7845514791389714`, 0.7852246924970142`, 0.785902375477276`,
0.7865846957540633`, 0.7872715266742573`, 0.7879629627890139`,
0.7886588027752234`, 0.7893594381462953`, 0.7900645191428892`,
0.7907742153515782`, 0.7914883242232598`, 0.7922070158502637`,
0.7929303116950635`, 0.7936580832351503`, 0.79439042671716`,
```

```
0.79512721302898`, 0.7958686896660602`, 0.7966146521079462`,
0.7973649701171399`, 0.7981200440437473`, 0.7988794404211317`,
0.7996435609813665`, 0.8004120468357868`, 0.8011849954695451`,
0.8019625051385743`, 0.8027444444263688`, 0.8035310657480818`,
0.8043221607965856`, 0.8051176741809376`, 0.805917704934255`,
0.8067221977258968`, 0.8075311745280694`, 0.8083448132847256`,
0.8091626689744973`, 0.809985153279642`, 0.8108120542814211`,
0.8116433939212626`, 0.8124792730135412`, 0.8133195566391704`,
0.8141643458161848`, 0.8150135840325655`, 0.8158672936497972`,
0.8167254176943959`, 0.8175880580757333`, 0.8184550779954183`,
0.8193265795555524`, 0.820202665719675`7, 0.8210829581157649`,
0.8219676388887937`, 0.8228569724569869`, 0.823750578390468`,
0.8246486404015909`, 0.8255512625452456`, 0.8264582240770975`,
0.82736954709732`, 0.8282852538784796`, 0.8292054488332409`,
0.8301299908343983`, 0.8310589844656335`, 0.8319922876691473`,
0.8329300051910378`, 0.8338721598039728`, 0.8348186913782158`,
0.8357695392053412`, 0.8367248923672541`, 0.8376846070432505`,
0.8386485380550183`, 0.8396169590333054`, 0.8405898934107459`,
0.8415670270890101`, 0.8425485507517855`, 0.843534317532859`,
0.8445246892811329`, 0.8455191789825698`, 0.8465181492224213`,
0.8475213665887605`, 0.8485290245587246`, 0.8495408882304782`,
0.8505570653730646`, 0.851577751189615`, 0.8526027094295009`,
0.8536319621091404`, 0.8546655314064271`, 0.8557033523737672`,
0.8567456218804408`, 0.857792012375524`, 0.8588428085523404`,
0.8598979452614375`, 0.8609572682906803`, 0.8620209761731366`,
0.8630888253428467`, 0.8641610148643877`, 0.8652377457260064`,
0.8663184156568785`, 0.8674034028949816`, 0.8684929095820293`,
0.8695864186831842`, 0.8706842212170615`, 0.8717864298838534`,
0.8728927043092571`, 0.8740032469555363`, 0.8751182622838117`,
0.8762373160836702`, 0.8773606119575542`, 0.8784882635332419`,
0.8796199243590083`, 0.8807559841236027`, 0.8818962801981326`,
0.8830407411446528`, 0.8841894811001859`, 0.8853423349982079`,
0.8864992296533424`, 0.8876605606861114`, 0.8888261627383355`,
0.8899957742057221`, 0.8911696040128196`, 0.8923476733678312`,
0.8935300036237109`, 0.894716329992007`, 0.8959069588811929`,
0.8971018161359039`, 0.8983007305285637`, 0.8995037218182899`,
0.9007110033180441`, 0.9019224026641124`, 0.9031379400150146`,
0.9043576356621406`, 0.905581314509086`, 0.9068093876291323`,
0.9080413858469687`, 0.9092776232676688`, 0.9105181208519912`,
0.9117625033001171`, 0.9130109873365369`, 0.9142638916242037`,
0.9155205390571411`, 0.9167815474911536`, 0.9180465368143029`,
0.9193156263561881`, 0.9205887346235347`, 0.9218659817424861`,
0.9231472859037879`, 0.9244326661656931`, 0.925722039548464`,
0.9270157318241061`, 0.9283134559397352`, 0.9296150245397032`,
0.9309208679175851`, 0.9322305923280361`, 0.9335444228324765`,
0.9348622743880456`, 0.9361840608917478`, 0.9375100094962389`,
0.938840034618616`1, 0.9401741549749294`, 0.9415121780509776`,
```

```
0.9428541209843232`, 0.9442002135664803`, 0.9455501552188207`,
0.9469042831332087`, 0.9482624023121999`, 0.9496244229140398`,
0.9509902539633343`, 0.9523603439465527`, 0.9537343873979752`,
0.9551122927068093`, 0.9564944033511821`, 0.9578801916672062`,
0.9592700013253617`, 0.9606638498550463`, 0.9620616445520155`,
0.9634632915277079`, 0.964869028637786`, 0.9662784285912905`,
0.967692175296937`, 0.9691095051337824`, 0.9705308805564159`,
0.9719562062807827`, 0.973385498744426`, 0.9748186611950006`,
0.9762558228659365`, 0.9776967725640967`, 0.9791415250693071`,
0.9805902098647613`, 0.9820429578703903`, 0.983499440226784`,
0.98495990230958`, 0.9864241283396703`, 0.9878924815619723`,
0.9893642792729997`, 0.9908402349307053`, 0.9923200139681276`,
0.9938035131748797`, 0.9952909822707956`, 0.9967822001005741`,
0.9982771803450756`, 0.9997761750630416`, 1.0012790209470939`,
1.002785432740245`, 1.0042960234112834`, 1.005810146951185`,
1.007328117133634`, 1.0088499481149806`, 1.0103756541205324`,
1.0119051884126677`, 1.0134383199309347`, 1.0149753059330433`,
1.0165161603509514`, 1.0180607118517502`, 1.019609034773222`,
1.0211612664160044`, 1.0227169845641644`, 1.0242766372143355`,
1.0258401132255606`, 1.0274072370743674`, 1.0289782096414453`,
1.0305527277538582`, 1.0321311193830647`, 1.0337131432127442`,
1.0352990020665602`, 1.0368884525845798`, 1.0384817625526577`,
1.0400786871359353`, 1.0416793017628536`, 1.0432835529463473`,
1.0448918419766704`, 1.0465034652934528`, 1.0481188236706003`,
1.0497379286697963`, 1.0513607918975385`, 1.052987358916762`,
1.0546173756363326`, 1.0562512497059886`, 1.0578886599919504`,
1.0595298834797375`, 1.0611744621852752`, 1.0628228749327548`,
1.0644747959634864`, 1.0661306405953834`, 1.0677899457577074`,
1.0694527878720503`, 1.0711194493813354`, 1.0727895301898411`,
1.0744633132212684`, 1.0761406706918526`, 1.0778215423667326`,
1.07950614500495`, 1.0811942102560232`, 1.082886095365169`,
1.0845813907183732`, 1.0862803144675304`, 1.0879828757578072`,
1.0896888006543375`, 1.0913983796745037`, 1.0931114083604974`,
1.0948281082616205`, 1.0965485602928258`, 1.0982722711326154`,
1.0999995344879212`, 1.101730502996508`, 1.1034648230966808`,
1.1052027919235663`, 1.1069441260040769`, 1.1086891243206665`,
1.1104377220706203`, 1.1121896323543645`, 1.1139451562534133`,
1.1157040050179101`, 1.1174664071689573`, 1.119232295383328`,
1.1210016763025554`, 1.122774781987945`, 1.1245509428357168`,
1.1263306148993153`, 1.1281142599552767`, 1.1299005953503407`,
1.1316909175165313`, 1.133484318500061`, 1.135281645627652`,
1.137082292670519`, 1.1388862640829887`, 1.140693486749204`,
1.1425043533710868`, 1.144318481075063`, 1.1461359518209644`,
1.1479571627341925`, 1.149781570356481`, 1.1516091775555723`,
1.1534406215697839`, 1.1552751953069589`, 1.1571130604291189`,
1.1589546207790007`, 1.1607992411901094`, 1.1626473257272862`,
1.164498717763003`, 1.1663535829368776`, 1.1682117635606597`,
```

```
1.170073263064749`, 1.1719383304470088`, 1.1738063965713008`,
1.1756780383689225`, 1.177552765644679`, 1.1794309109979444`,
1.1813124785112261`, 1.183197305360484`, 1.1850853939911428`,
1.186976830796392`, 1.1888716189662016`, 1.1907695926345705`,
1.1926709228909111`, 1.194575697899633`, 1.1964834949163121`,
1.1983947417312935`, 1.200309184561491`, 1.202226997054485`,
1.2041477500483568`, 1.2060721359444408`, 1.207999638259996`,
1.209930344621977`, 1.2118645188016441`, 1.2138015497276018`,
1.215742051800075`, 1.2176857634574427`, 1.2196327743682247`,
1.22158273056522`, 1.22353607656552`, 1.225492635926435`,
1.227452589092691`, 1.229415488134758`, 1.231381511973466`,
1.2333510231197593`, 1.2353234790979581`, 1.23729915166045`,
1.239278041379402`, 1.2412601487820945`, 1.2432453822220793`,
1.2452337412509298`, 1.2472254108070613`, 1.249220020054872`,
1.2512179398641539`, 1.2532189841445227`, 1.2552231521908364`,
1.2572304432442738`, 1.2592408564919233`, 1.2612542962495363`,
1.2632710460452867`, 1.265291011300441`, 1.267313617520051`,
1.269339819389425`, 1.2713689459572473`, 1.273401284132419`,
1.2754362518150235`, 1.2774747184787152`, 1.2795162961389641`,
1.2815608854656657`, 1.2836082876096326`, 1.2856591878389256`,
1.2877128964509332`, 1.289769706384943`, 1.2918295164267826`,
1.2938924236042981`, 1.2959584260468147`, 1.2980276222478002`,
1.3000996081850744`, 1.3021746821134248`, 1.304252841904466`,
1.3063339836498955`, 1.3084181041182315`, 1.3105052000070614`,
1.3125953706357958`, 1.3146888195841837`, 1.3167848228473549`,
1.3188839949666558`, 1.3209861262417204`, 1.3230914216247782`,
1.3251996699563084`, 1.3273108673612075`, 1.3294247992021493`,
1.3315416707970091`, 1.3336619021015577`, 1.3357848548613638`,
1.3379111629493732`, 1.3400399698496885`, 1.34217180393904`,
1.3443067694673778`, 1.3464444316929212`, 1.3485851091176748`,
1.350728580556071`, 1.352875276536742`, 1.3550246480954948`,
1.3571772371213815`, 1.3593323808865598`, 1.3614905130644033`,
1.363651740330304`, 1.3658157259318477`, 1.367982686521308`,
1.3701523939056157`, 1.3723251782314745`, 1.3745004729623336`,
1.376679059928299`, 1.378860257677614`, 1.3810445129026439`,
1.383231593701047`, 1.3854213796082504`, 1.3876139781554309`,
1.3898097284931432`, 1.3920082810856413`, 1.3942095136600774`,
1.3964137672955566`, 1.398620571220399`, 1.4008305013576032`,
1.403043084918932`, 1.4052586670392309`, 1.4074770068348625`,
1.4096979788063204`, 1.4119220499797307`, 1.4141486204958467`,
1.416378159066413`, 1.4186104206192967`, 1.4208452773022022`,
1.4230833239039125`, 1.425324072909863`, 1.427567273571362`,
1.4298136468104923`, 1.4320624557954464`, 1.4343144258524474`,
1.4365686923505778`, 1.4388259838833326`, 1.441085924189566`,
1.4433486289439938`, 1.4456144644027298`, 1.4478825518892695`,
1.4501532551186673`, 1.4524269424023215`, 1.4547033560147484`,
1.45698286762356`, 1.4592647112290005`, 1.4615493844357665`,
```

```
1.4638367525003`, 1.4661265508630652`, 1.468419282791839`,
1.4707148132482726`, 1.4730128761635046`, 1.475313721067555`,
1.4776170800717758`, 1.4799233346600162`, 1.4822323472879482`,
1.484543715962316`, 1.4868582196092424`, 1.489174796353952`,
1.4914942273684644`, 1.4938163725580107`, 1.4961414903399926`,
1.4984690393326758`, 1.5007994111739371`, 1.5031319250287551`,
1.505467647757948`, 1.5078054913718184`, 1.510146257361722`,
1.5124893936886092`, 1.5148355724356024`, 1.5171839648461327`,
1.5195351081272481`, 1.5218892700794815`, 1.5242454754967771`,
1.5266048207224678`, 1.5289661862868709`, 1.5313302558649504`,
1.5336968804200797`, 1.536066752900883`, 1.5384384587607152`,
1.5408128300657875`, 1.5431899994996219`, 1.5455701013925889`,
1.5479522719784573`, 1.5503369261900533`, 1.5527244841239505`,
1.555114362047504`, 1.5575065473939789`, 1.5599020427229326`,
1.5622996807369915`, 1.5646998849689462`, 1.567102645069318`,
1.56950809740498`, 1.5719154963650215`, 1.5743258599563357`,
1.576738588716366`, 1.5791542649228958`, 1.5815719870885356`,
1.5839923381718262`, 1.5864151582820452`, 1.5888407369845674`,
1.59126861277536`, 1.5936992268421204`, 1.5961315072892903`,
1.59856695848372`, 1.6010052698799981`, 1.6034453608919388`,
1.6058882897342397`, 1.6083335857756005`, 1.6107812366035987`,
1.6132318501733935`, 1.6156843303365023`, 1.6181394404089267`,
1.620597013800833`, 1.6230571959462499`, 1.6255196615877703`,
1.6279847134232632`, 1.630452340957141`, 1.6329222157195136`,
1.6353943241035518`, 1.6378694518582213`, 1.6403464697220305`,
1.6428258434123257`, 1.645308044341643`, 1.6477925786153516`,
1.6502795956871963`, 1.6527689212120018`, 1.6552605420743225`,
1.6577546088383852`, 1.660251438216226`, 1.6627501972241934`,
1.6652516957268884`, 1.6677552613337954`, 1.6702612112737247`,
1.67276986664077`, 1.675280381428183`, 1.6777937451310407`,
1.6803099494461269`, 1.6828272980221994`, 1.6853476294886733`,
1.6878705971133832`, 1.6903956796875936`, 1.6929233747357775`,
1.695452986422605`, 1.6979848417054604`, 1.7005197885978154`,
1.703056610752882`, 1.7055958112236662`, 1.7081372034864406`,
1.7106811221116123`, 1.71322720590608`, 1.7157757912746812`,
1.718326690697189`, 1.7208798909613992`, 1.723435555773332`,
1.7259929629939403`, 1.7285533409527312`, 1.7311154319364759`,
1.7336806505667328`, 1.7362475534044224`, 1.738817022672008`,
1.7413888666306567`, 1.7439630725745197`, 1.7465390822256723`,
1.749117607227077`, 1.7516988190149956`, 1.7542817904715817`,
1.756867423299174`, 1.7594547847335782`, 1.7620447816387572`,
1.764636846788012`, 1.7672313370522978`, 1.76982786675563`,
1.7724271692039224`, 1.7750284839894535`, 1.7776319835286405`,
1.7802378423162721`, 1.78284547911509`, 1.785455635984157`,
1.7880683014752838`, 1.7906825084046898`, 1.793299770188659`,
1.7959191188473809`, 1.7985405391092575`, 1.8011642089435236`,
1.8037903084477196`, 1.806418436364043`, 1.8090485769170013`,
```

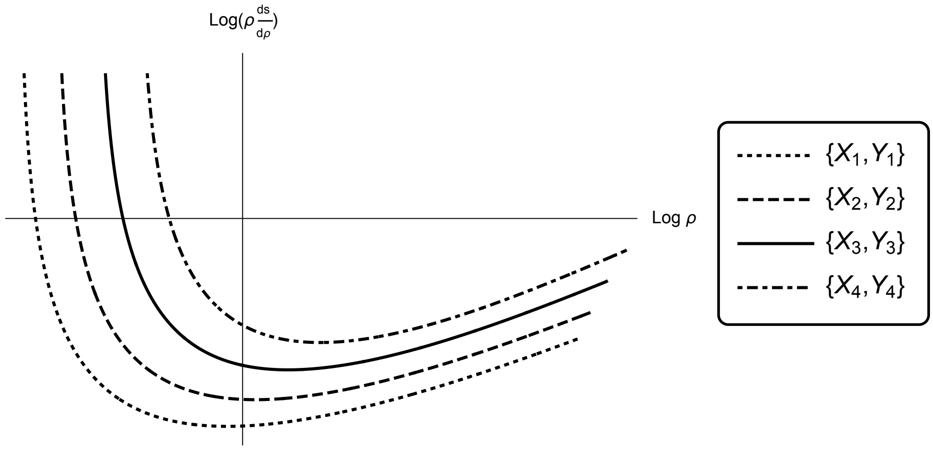
```
1.811680909835021`, 1.8143160093996733`, 1.81695288242754`,
1.8195919056154028`, 1.8222336580264193`, 1.824876939049547`,
1.827522922988171`, 1.8301704020846896`, 1.8328205573026806`,
1.8354731777334818`, 1.8381278486533699`, 1.8407841505664588`,
1.843443279605662`, 1.8461042112795438`, 1.8487673353453493`,
1.8514330461259894`, 1.8541009229274414`, 1.856770334732303`,
1.8594420858909984`, 1.862116163386408 3`, 1.8647927613582835`,
1.8674706211371979`, 1.8701515965919693`, 1.8728344270246406`,
1.8755190952371796`, 1.8782062146924279`, 1.8808949298779514`,
1.8835864902208461`, 1.8862800387662266`, 1.8889751341248529`,
1.891673036243276`, 1.8943730944446364`, 1.8970748651389024`,
1.8997791904555923`, 1.9024854120471577`, 1.9051937294870953`,
1.9079039107489562`, 1.9106163739662538`, 1.9133311056851785`,
1.9160478735134603`, 1.9187666619013581`, 1.9214878953089476`,
1.9242106789163727`, 1.9269356570045857`, 1.9296632592912084`,
1.9323921398252057`, 1.9351236158628957`, 1.937856781325663`,
1.940592291169534`, 1.943330582720623`, 1.94607006666802`,
1.94881162585854`, 1.9515561531911725`, 1.9543015922512945`,
1.957049972663244`, 1.9597999130442705`, 1.9625520826448737`,
1.9653064676574954`, 1.9680628232944692`, 1.9708213650872262`,
1.9735809180697435`, 1.9763437863300308`, 1.9791083309868318`,
1.9818747677289688`, 1.9846433147292424`, 1.9874139572305778`,
1.9901866803575006`, 1.992961232372344`, 1.995738070988216`,
1.9985164687572776`, 2.001297838716819`, 2.0040807370029388`,
2.0068657441806037`, 2.009652725172358`, 2.0124417848402723`,
2.0152324241040893`, 2.0180252307040925`, 2.0208203121565287`,
2.023617777211682`, 2.0264161449309146`, 2.029217599550063`,
2.0320201656359136`, 2.0348251776325417`, 2.037632128758912`,
2.0404413752205652`, 2.043251784153451`, 2.0460655788686113`,
2.0488797527502807`, 2.0516965350352603`, 2.054515663689288`,
2.057336621530596`, 2.060159391683053`, 2.062984337653695`,
2.065811063896855`, 2.068639935765346`, 2.0714705553888377`,
2.0743034182045137`, 2.077137739312851`, 2.079974788208429`,
2.082813909050099`, 2.0856541795681993`, 2.0884963573778696`,
2.0913413387989808`, 2.0941876773427213`, 2.097036400868402`,
2.0998863144672257`, 2.1027389774631193`, 2.105593059375906`,
2.1084490701176906`, 2.1113069938267373`, 2.114167214145798`,
2.11702931751952`, 2.1198931538074697`, 2.1227589741779442`,
2.1256273018912544`, 2.1284967756423026`, 2.131368591740186`,
2.134242602009841`, 2.1371179761318957`, 2.139995103215455`,
2.1428745134944305`, 2.1457556452298285`, 2.1486388937821865`,
2.1515243829130615`, 2.154411407793483`, 2.1573002269527257`,
2.160191101767614`, 2.16308345981886`, 2.1659782605818916`,
2.168875352355294`, 2.171773879011086`, 2.174673398125504`,
2.1775760063469094`, 2.1804799970471556`, 2.1833863447645894`,
2.1862940401873123`, 2.1892042057441827`, 2.1921159709408893`,
2.195029030412673`, 2.1979440834345967`, 2.200861836796724`,
```

```
2.2037809784900637`, 2.206702214322256`, 2.2096252391084805`,
2.212549598579144`, 2.2154762965113464`, 2.2184050273855846`,
2.2213357765902795`, 2.224268381962035`, 2.227201940509503`,
2.23013805954699`, 2.2330758370183093`, 2.236016000429685`,
2.2389574922504267`, 2.2419013409075315`, 2.244846032599824`,
2.2477931997929756`, 2.250742529544724`, 2.2536929488004414`,
2.2566454980253092`, 2.2596000113672163`, 2.2625563212767035`,
2.2655150233094417`, 2.2684751856861696`, 2.2714367894912777`,
2.2744010490009376`, 2.277366254473704`, 2.280334242650835`,
2.283303295926356`, 2.2862747943731683`, 2.289247321218049`,
2.29222226358987`, 2.295198354187353`, 2.2981763588270505`,
2.3011570513638193`, 2.3041386821751684`, 2.3071217043630123`,
2.3101076891227437`, 2.3130947159901765`, 2.316083882102536`,
2.319075013791663`, 2.322068096327285`, 2.3250619871117806`,
2.328058600769733`, 2.331056794897223`, 2.3340570389062`,
2.3370588304534192`, 2.3400628044295058`, 2.343068293409488`,
2.3460754433164803`, 2.3490847305507647`, 2.352095647200009`,
2.3551081760801464`, 2.3581227970627925`, 2.361138998112816`,
2.364156761712004`, 2.367176070227906`, 2.3701975756116775`,
2.3732212651297626`, 2.376246284551581`, 2.3792727832254705`,
2.3823019269318486`, 2.385332349848411`, 2.3883642022739204`,
2.391398147413679`, 2.394434000953223`, 2.3974717480447825`,
2.4005110302684134`, 2.4035516577874745`, 2.4065949923615944`,
2.409639121797883`, 2.412685757657989`, 2.4157336722833156`,
2.4187833692150233`, 2.4218348327599286`, 2.424888397599685`,
2.427943172071184`, 2.4310001929328893`, 2.4340589183527883`,
2.4371189780879283`, 2.4401810629726453`, 2.443244803438702`,
2.446310361296739`, 2.4493780788375115`, 2.4524472263598502`,
2.4555179653579864`, 2.458590639157926`, 2.4616639692202793`,
2.4647406473059865`, 2.467818310190633`, 2.4708976642239553`,
2.473978875674614`, 2.4770611982151274`, 2.480146077939771`,
2.4832325863651166`, 2.4863199700110132`, 2.48941060768762`,
2.4925020880768805`, 2.495595688061249`, 2.498691022670373`,
2.5017873300873297`, 2.5048853369420816`, 2.5079857771516574`,
2.5110875127502243`, 2.514191089999524`, 2.517296116381397`,
2.5204037100850454`, 2.5235117740682083`, 2.52662237617063`,
2.5297345531853748`, 2.532848288386813`, 2.5359647149169926`,
2.539081710856239`, 2.5421998299110347`, 2.5453204036504253`,
2.5484432283743788`, 2.55156751631894`, 2.554693055533138`,
2.5578202169934223`, 2.5609497662974525`, 2.5640799294287238`,
2.567212646920995`, 2.570346728062901`, 2.5734833390305516`,
2.576620293883282`, 2.5797597488391655`, 2.582900501262837`,
2.586043130580934`, 2.589187622555412`, 2.5923339628537865`,
2.5954807317004875`, 2.5986305206053864`, 2.601781911544586`,
2.6049346861056852`, 2.608088218377046`, 2.6112451278111535`,
2.6144021519570324`, 2.6175617142257583`, 2.620722577194493`,
2.6238855433228`, 2.6270499831436913`, 2.6302160858445576`,
```

```
2.6333840424977537`, 2.63655301022761`, 2.6397240070894172`,
2.6428970201826076`, 2.646071619185089`, 2.649246742172727`,
2.6524253008109104`, 2.655603719443519`, 2.658785340435275`,
2.661968681136608`, 2.6651518213092484`, 2.66833791503178`,
2.67152589266343`, 2.674714887545287`, 2.6779046659378087`,
2.681097776908434`, 2.6842924991574195`, 2.687488816957889`,
2.690686067198446`, 2.693885310718497`, 2.6970867513689223`,
2.7002890738362706`, 2.703492910734237`, 2.7066989007656`,
2.709905937899744`, 2.713114660648741`, 2.7163263733324525`,
2.7195382047891106`, 2.7227518963429875`, 2.7259676556898746`,
2.729185026585909`, 2.7324031035022176`, 2.7356236487763392`,
2.7388455352307366`, 2.7420694177390175`, 2.745295508185259`,
2.7485215439654165`, 2.751749528902019`, 2.7549801285843367`,
2.7582126545206807`, 2.7614452754661327`, 2.7646811544838195`,
2.7679180045575937`, 2.7711569501512003`, 2.7743963735698034`,
2.7776376311494713`, 2.780880939649511`, 2.784125593861428`,
2.78737134512154`, 2.790619799206704`, 2.7938693201394633`,
2.7971205884064285`, 2.800374057189355`, 2.8036280756249594`,
2.80688426473612`, 2.8101416718946823`, 2.813400986899584`,
2.8166619602586653`, 2.8199241032166`, 2.8231892978995132`,
2.826454206033847`, 2.8297216630055213`, 2.8329904643413717`,
2.836260593154078`, 2.8395322727550836`, 2.8428071736706273`,
2.8460823952044207`, 2.8493591232368045`, 2.8526380696371727`,
2.8559177651986465`, 2.8591981890330533`, 2.8624822504754888`,
2.8657677458123896`, 2.8690541682834136`, 2.872342727692335`,
2.8756311948062074`, 2.878922507154529`, 2.8822154205387114`,
2.8855106644878994`, 2.888805493360456`, 2.892103866857023`,
2.895402535502611`, 2.8987039796497256`, 2.9020059336669175`,
2.9053106392882575`, 2.9086150640465256`, 2.9119222127408015`,
2.915230811909217`, 2.9185416073395234`, 2.921853570643003`,
2.9251661741325847`, 2.9284822080021224`, 2.931797315230296`,
2.9351168555798344`, 2.9384359444711587`, 2.9417573902878194`,
2.9450804080684017`, 2.9484044648500394`, 2.951730840519035`,
2.9550579638861234`, 2.9583871187590134`, 2.961718554918147`,
2.9650501663522477`, 2.9683837670759043`, 2.971719871581977`,
2.975056888785251`, 2.978394799548551`, 2.9817343795306046`}[[s]];
```

LCG

```
ListLinePlot[{Table[{Log[HP2rho180[s]],
    Log[.001 * HP2rho180[s] / (HP2rho180[s + 1] - HP2rho180[s])]},
   {s, 1, 921, 1}], Table[{Log[HP2rho360[s]],
    Log[.001 * HP2rho360[s] / (HP2rho360[s + 1] - HP2rho360[s])]},
   {s, 1, 973, 1}], Table[{Log[HP2rho540[s]],
    Log[.001 * HP2rho540[s] / (HP2rho540[s + 1] - HP2rho540[s])]},
   {s, 1, 1038, 1}], Table[{Log[HP2rho720[s]],
    Log[.001 * HP2rho720[s] / (HP2rho720[s + 1] - HP2rho720[s])]}, {s, 1, 1110, 1}]},
 AxesLabel → {"Log ρ", "Log(ρ ds/dρ)"}, PlotStyle →
  {{Dotted, Black}, {Dashed, Black}, {Black}, {DotDashed, Black}}, Ticks → None,
 PlotLegends → LineLegend[{{{"X1,Y1"}, {"X2,Y2"}, {"X3,Y3"}, {"X4,Y4"}}},
   LegendFunction → (Framed[#, RoundingRadius → 5] &), LegendMargins → 5]]
```



LAPSurface

Set up

```
kernelcodeCPLAS2 = "
#define min 0
#define max 1
#define vvv 0
#define uuu 1
#define sign1 -1
#define h 0.001
#define n 1201
#define AlphaLAC (2.0)
#define BetaLASC (2.0)
#define InitRCLAC (1.0)
#define OmegaLASC (1.0)
#define Lambda_Max_LAC (12.718433357775211)
#define ArcLength_Max_LAC (6.764357890136049)
#define Lambda_Min_LAC (12.718433357775211)
```

```
#define ArcLength_Min_LAC (6.764357890136049)
#define Lambda_Max_LASC (6.285247558593747)
#define InitRT_Max_LASC (7.303131835937503)
#define ArcLength_Max_LASC (3.1084204465150833)
#define Lambda_Min_LASC (6.285247558593747)
#define InitRT_Min_LASC (7.303131835937503)
#define ArcLength_Min_LASC (3.1084204465150833)
#define Point00x (0.0)
#define Point00y (0.0)
#define Point00z (0.0)
#define Point03x (-0.7077264466498845)
#define Point03y (0.0)
#define Point03z (0.5008767232876717)
#define Point30x (0.0)
#define Point30y (-0.7077264466498845)
#define Point30z (-0.5008767232876717)
#define Point33x (-1.0013149889663)
#define Point33y (-1.0013149889663)
#define Point33z (0.0)

#define maxPoint00x (0.0)
#define maxPoint00y (0.0)
#define maxPoint00z (0.0)
#define maxPoint03x (-0.7077264466498845)
#define maxPoint03y (0.0)
#define maxPoint03z (0.5008767232876717)
#define maxPoint30x (0.0)
#define maxPoint30y (-0.7077264466498845)
#define maxPoint30z (-0.5008767232876717)
#define maxPoint33x (-1.0013149889663)
#define maxPoint33y (-1.0013149889663)
#define maxPoint33z (0.0)


#define ScaleToOrigin_Max_LAC (0.13192365851313362)
#define ScaleToOrigin_Min_LAC (0.13192365851313362)
#define ScaleToOrigin_Max_LASC (0.38164320308766897)
#define ScaleToOrigin_Min_LASC (0.38164320308766897)

#define inidxdu1
    ((1-pv)*(maxPoint00x-maxPoint03x)+pv*(maxPoint30x-maxPoint33x)-v1_x1+
    v2_x1+(1-pv)*u1_t1+pv*u2_t1)
#define inidxdu2
    ((1-pv)*(maxPoint00y-maxPoint03y)+pv*(maxPoint30y-maxPoint33y)-v1_x2+
    v2_x2+(1-pv)*u1_t2+pv*u2_t2)
#define inidxdu3
    ((1-pv)*(maxPoint00z-maxPoint03z)+pv*(maxPoint30z-maxPoint33z)-v1_x3+
```

```
    v2_x3+(1-pv)*u1_t3+pv*u2_t3)
#define inidxdv1
    ((1-pu)*(maxPoint00x-maxPoint30x)+pu*(maxPoint03x-maxPoint33x)-u1_x1+
    u2_x1+(1-pu)*v1_t1+pu*v2_t1)
#define inidxdv2
    ((1-pu)*(maxPoint00y-maxPoint30y)+pu*(maxPoint03y-maxPoint33y)-u1_x2+
    u2_x2+(1-pu)*v1_t2+pu*v2_t2)
#define inidxdv3
    ((1-pu)*(maxPoint00z-maxPoint30z)+pu*(maxPoint03z-maxPoint33z)-u1_x3+
    u2_x3+(1-pu)*v1_t3+pu*v2_t3)

#define inid2xdu21 ((1-pv)*u1_dt1+pv*u2_dt1)
#define inid2xdu22 ((1-pv)*u1_dt2+pv*u2_dt2)
#define inid2xdu23 ((1-pv)*u1_dt3+pv*u2_dt3)
#define inid2xdv21 ((1-pu)*v1_dt1+pu*v2_dt1)
#define inid2xdv22 ((1-pu)*v1_dt2+pu*v2_dt2)
#define inid2xdv23 ((1-pu)*v1_dt3+pu*v2_dt3)
#define inid2xdudv1
    (-maxPoint00x+maxPoint03x+maxPoint30x-maxPoint33x-u1_t1+u2_t1-v1_t1+v2_t1)
#define inid2xdudv2
    (-maxPoint00y+maxPoint03y+maxPoint30y-maxPoint33y-u1_t2+u2_t2-v1_t2+v2_t2)
#define inid2xdudv3
    (-maxPoint00z+maxPoint03z+maxPoint30z-maxPoint33z-u1_t3+u2_t3-v1_t3+v2_t3)


#define inid3xdu31 ((1-pv)*u1_d2t1+pv*u2_d2t1)
#define inid3xdu32 ((1-pv)*u1_d2t2+pv*u2_d2t2)
#define inid3xdu33 ((1-pv)*u1_d2t3+pv*u2_d2t3)
#define inid3xdv31 ((1-pu)*v1_d2t1+pu*v2_d2t1)
#define inid3xdv32 ((1-pu)*v1_d2t2+pu*v2_d2t2)
#define inid3xdv33 ((1-pu)*v1_d2t3+pu*v2_d2t3)
#define inid3xdu2dv1 (-u1_dt1+u2_dt1)
#define inid3xdu2dv2 (-u1_dt2+u2_dt2)
#define inid3xdu2dv3 (-u1_dt3+u2_dt3)
#define inid3xdudv21 (-v1_dt1+v2_dt1)
#define inid3xdudv22 (-v1_dt2+v2_dt2)
#define inid3xdudv23 (-v1_dt3+v2_dt3)


#define dxdu1
    ((1-pv[i])*(maxPoint00x-maxPoint03x)+pv[i]*(maxPoint30x-maxPoint33x)-
    v1_x1[i]+v2_x1[i]+(1-pv[i])*u1_t1[i]+pv[i]*u2_t1[i])
#define dxdu2
    ((1-pv[i])*(maxPoint00y-maxPoint03y)+pv[i]*(maxPoint30y-maxPoint33y)-
    v1_x2[i]+v2_x2[i]+(1-pv[i])*u1_t2[i]+pv[i]*u2_t2[i])
#define dxdu3
    ((1-pv[i])*(maxPoint00z-maxPoint03z)+pv[i]*(maxPoint30z-maxPoint33z)-
```

```
    v1_x3[i]+v2_x3[i]+(1-pv[i])*u1_t3[i]+pv[i]*u2_t3[i])
#define dxdv1
    ((1-pu[i])*(maxPoint00x-maxPoint30x)+pu[i]*(maxPoint03x-maxPoint33x)-
    u1_x1[i]+u2_x1[i]+(1-pu[i])*v1_t1[i]+pu[i]*v2_t1[i])
#define dxdv2
    ((1-pu[i])*(maxPoint00y-maxPoint30y)+pu[i]*(maxPoint03y-maxPoint33y)-
    u1_x2[i]+u2_x2[i]+(1-pu[i])*v1_t2[i]+pu[i]*v2_t2[i])
#define dxdv3
    ((1-pu[i])*(maxPoint00z-maxPoint30z)+pu[i]*(maxPoint03z-maxPoint33z)-
    u1_x3[i]+u2_x3[i]+(1-pu[i])*v1_t3[i]+pu[i]*v2_t3[i])


#define d2xdu21 ((1-pv[i])*u1_dt1[i]+pv[i]*u2_dt1[i])
#define d2xdu22 ((1-pv[i])*u1_dt2[i]+pv[i]*u2_dt2[i])
#define d2xdu23 ((1-pv[i])*u1_dt3[i]+pv[i]*u2_dt3[i])
#define d2xdv21 ((1-pu[i])*v1_dt1[i]+pu[i]*v2_dt1[i])
#define d2xdv22 ((1-pu[i])*v1_dt2[i]+pu[i]*v2_dt2[i])
#define d2xdv23 ((1-pu[i])*v1_dt3[i]+pu[i]*v2_dt3[i])
#define d2xdudv1
    (-maxPoint00x+maxPoint03x+maxPoint30x-maxPoint33x-u1_t1[i]+u2_t1[i]-
    v1_t1[i]+v2_t1[i])
#define d2xdudv2
    (-maxPoint00y+maxPoint03y+maxPoint30y-maxPoint33y-u1_t2[i]+u2_t2[i]-
    v1_t2[i]+v2_t2[i])
#define d2xdudv3
    (-maxPoint00z+maxPoint03z+maxPoint30z-maxPoint33z-u1_t3[i]+u2_t3[i]-
    v1_t3[i]+v2_t3[i])


#define d3xdu31 ((1-pv[i])*u1_d2t1[i]+pv[i]*u2_d2t1[i])
#define d3xdu32 ((1-pv[i])*u1_d2t2[i]+pv[i]*u2_d2t2[i])
#define d3xdu33 ((1-pv[i])*u1_d2t3[i]+pv[i]*u2_d2t3[i])
#define d3xdv31 ((1-pu[i])*v1_d2t1[i]+pu[i]*v2_d2t1[i])
#define d3xdv32 ((1-pu[i])*v1_d2t2[i]+pu[i]*v2_d2t2[i])
#define d3xdv33 ((1-pu[i])*v1_d2t3[i]+pu[i]*v2_d2t3[i])
#define d3xdu2dv1 (-u1_dt1[i]+u2_dt1[i])
#define d3xdu2dv2 (-u1_dt2[i]+u2_dt2[i])
#define d3xdu2dv3 (-u1_dt3[i]+u2_dt3[i])
#define d3xdudv21 (-v1_dt1[i]+v2_dt1[i])
#define d3xdudv22 (-v1_dt2[i]+v2_dt2[i])
#define d3xdudv23 (-v1_dt3[i]+v2_dt3[i])

#define d4xdu41 ((1-pv[i])*u1_d3t1[i]+pv[i]*u2_d3t1[i])
#define d4xdu42 ((1-pv[i])*u1_d3t2[i]+pv[i]*u2_d3t2[i])
#define d4xdu43 ((1-pv[i])*u1_d3t3[i]+pv[i]*u2_d3t3[i])
#define d4xdv41 ((1-pu[i])*v1_d3t1[i]+pu[i]*v2_d3t1[i])
#define d4xdv42 ((1-pu[i])*v1_d3t2[i]+pu[i]*v2_d3t2[i])
```

```
#define d4xdv43 ((1-pu[i])*v1_d3t3[i]+pu[i]*v2_d3t3[i])
#define d4xdu3dv1 (-u1_d2t1[i]+u2_d2t1[i])
#define d4xdu3dv2 (-u1_d2t2[i]+u2_d2t2[i])
#define d4xdu3dv3 (-u1_d2t3[i]+u2_d2t3[i])
#define d4xdudv31 (-v1_d2t1[i]+v2_d2t1[i])
#define d4xdudv32 (-v1_d2t2[i]+v2_d2t2[i])
#define d4xdudv33 (-v1_d2t3[i]+v2_d2t3[i])
#define d4xdu2dv21 (0.0)
#define d4xdu2dv22 (0.0)
#define d4xdu2dv23 (0.0)

#define Coons_x1
    ((1-pv[index])*u1_x1[index]+pv[index]*u2_x1[index]+(1-pu[index])*v1_x1[
    index]+pu[index]*v2_x1[index]-((1-pu[index])*(1-pv[index])*maxPoint00x+
    pu[index]*(1-pv[index])*maxPoint03x+(1-pu[index])*pv[index]*maxPoint30x+
    pu[index]*pv[index]*maxPoint33x))
#define Coons_x2
    ((1-pv[index])*u1_x2[index]+pv[index]*u2_x2[index]+(1-pu[index])*v1_x2[
    index]+pu[index]*v2_x2[index]-((1-pu[index])*(1-pv[index])*maxPoint00y+
    pu[index]*(1-pv[index])*maxPoint03y+(1-pu[index])*pv[index]*maxPoint30y+
    pu[index]*pv[index]*maxPoint33y))
#define Coons_x3
    ((1-pv[index])*u1_x3[index]+pv[index]*u2_x3[index]+(1-pu[index])*v1_x3[
    index]+pu[index]*v2_x3[index]-((1-pu[index])*(1-pv[index])*maxPoint00z+
    pu[index]*(1-pv[index])*maxPoint03z+(1-pu[index])*pv[index]*maxPoint30z+
    pu[index]*pv[index]*maxPoint33z))


#define repiniSurNorvec
    iniSurNorvec(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3
    ,v2_t1,v2_t2,v2_t3,normv_C)
#define repiniDSurNvDu
    iniDSurNvDu(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,dnormvdu_C)
#define repiniDSurNvDv
    iniDSurNvDv(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,dnormvdv_C)
#define repiniScalarSurNor
    iniScalarSurNor(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,Scal)
```

```
#define repinicoefE
    inicoefE(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3)
#define repiniDEDu
    iniDEDu(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniDEDv
    iniDEDv(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repinicoefF
    inicoefF(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3)
#define repiniDFDu
    iniDFDu(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniDFDv
    iniDFDv(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repinicoefG
    inicoefG(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3)
#define repiniDGDu
    iniDGDu(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniDGDv
    iniDGDv(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repinicoefL
    inicoefL(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
```

```
#define repinicoefM
    inicoefM(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repinicoefN
    inicoefN(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniGaussCurva
    iniGaussCurva(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniMeanCurva
    iniMeanCurva(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniMaxPrinci
    iniMaxPrinci(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniMinPrinci
    iniMinPrinci(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniPrinci
    iniPrinci(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3
    ,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)

#define repiniDcoefL_2
    iniDcoefL_2(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    CoeffL)
#define repiniDcoefM_2
    iniDcoefM_2(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
```

```
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    CoeffM)
#define repiniDcoefN_2
    iniDcoefN_2(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    CoeffN)
#define repiniDGaussCurva_2
    iniDGaussCurva_2(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2
    ,u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    GCurva)
#define repiniDMeanCurva_2
    iniDMeanCurva_2(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    MCurva)
#define repiniDPrinci_2
    iniDPrinci_2(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    optimum,PCurva)
#define repiniEta
    iniEta(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniMiu
    iniMiu(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniDuDs
    iniDuDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniDvDs
```

```
    iniDvDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniDEDs
    iniDEDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniDFDs
    iniDFDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniDGDs
    iniDGDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniDLDs
    iniDLDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,optimum)
#define repiniDMDs
    iniDMDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,optimum)
#define repiniDNDs
    iniDNDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,optimum)
#define repiniDPrinciDs
    iniDPrinciDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    optimum)
#define repiniDvDt
    iniDvDt(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
```

```
v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniDuDt
    iniDuDt(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniDSurDs
    iniDSurDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3
    ,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum,tangv)
#define repiniSurNor
    iniSurNor(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,Snor_C)
#define repiniBinormalSur
    iniBinormalSur(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum,binormv_C)
#define repiniAlp1
    iniAlp1(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum,Alp1_C)
#define repiniBeta1
    iniBeta1(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,optimum)

#define repSurNorvec
    SurNorvec(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,normv_C)
#define repDSurNvDu
    DSurNvDu(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,dnormvdu_C)
#define repDSurNvDv
    DSurNvDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,dnormvdv_C)
```

```
#define repD2SurNvDu2
    D2SurNvDu2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    d2normvdu2_C)
#define repD2SurNvDuDv
    D2SurNvDuDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    d2normvdudv_C)
#define repD2SurNvDv2
    D2SurNvDv2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    d2normvdv2_C)
#define repScalarSurNor
    ScalarSurNor(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    Scal)
#define repcoefE
    coefE(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3)
#define repDEDu
    DEDu(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repDEDv
    DEDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repD2EDu2
    D2EDu2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
```

```
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repD2EDuDv
    D2EDuDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3
    ,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repD2EDv2
    D2EDv2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repcoefF
    coefF(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3)
#define repDFDu
    DFDu(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repDFDv
    DFDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repD2FDu2
    D2FDu2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repD2FDuDv
    D2FDuDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3
    ,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repD2FDv2
    D2FDv2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repcoefG
    coefG(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
```

```
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3)
#define repDGDu
    DGDu(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repDGDv
    DGDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repD2GDu2
    D2GDu2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repD2GDuDv
    D2GDuDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3
    ,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repD2GDv2
    D2GDv2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repcoefL
    coefL(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repcoefM
    coefM(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repcoefN
    coefN(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repGaussCurva
    GaussCurva(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
```

```
u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repMeanCurva
    MeanCurva(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repMaxPrinci
    MaxPrinci(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repMinPrinci
    MinPrinci(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repPrinci
    Princi(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDcoefL_2
    DcoefL_2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,CoeffL)
#define repDcoefM_2
    DcoefM_2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,CoeffM)
#define repDcoefN_2
    DcoefN_2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,CoeffN)
```

```
#define repDGaussCurva_2
    DGaussCurva_2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,GCurva)
#define repDMeanCurva_2
    DMeanCurva_2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,MCurva)
#define repDPrinci_2
    DPrinci_2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,optimum,PCurva)
#define repEta
    Eta(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repMiu
    Miu(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDuDs
    DuDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDvDs
    DvDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDEDs
    DEDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
```

```
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDFDs
    DFDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDGDs
    DGDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDLDs
    DLDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repDMDs
    DMDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repDNDs
    DNDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repDPrinciDs
    DPrinciDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,optimum)
#define repDvDt
    DvDt(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
```

```
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDuDt
    DuDt(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDSurDs
    DSurDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum,tangv)
#define repSurNor
    SurNor(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,Snor_C)
#define repBinormalSur
    BinormalSur(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum,binormv_C)
#define repAlp1
    Alp1(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum,Alp1_C)
#define repBeta1
    Beta1(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)

#define repCurvecurvature
    Curvecurvature(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2
    ,u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,
    u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,
    u2_d3t1,u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,
    v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,
    v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,
    v2_d2t3,v2_d3t1,v2_d3t2,v2_d3t3,D2uDs2,D2vDs2,GeoCur,optimum)
#define repD2EDs2
    D2EDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
```

```
,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
,v2_d3t3,D2uDs2,D2vDs2,GeoCur,optimum)
#define repD2FDs2
D2FDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
,v2_d3t3,D2uDs2,D2vDs2,GeoCur,optimum)
#define repD2GDs2
D2GDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
,v2_d3t3,D2uDs2,D2vDs2,GeoCur,optimum)
#define repD2LDs2
D2LDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
,v2_d3t3,D2uDs2,D2vDs2,GeoCur,optimum)
#define repD2MDs2
D2MDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
,v2_d3t3,D2uDs2,D2vDs2,GeoCur,optimum)
#define repD2NDs2
D2NDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
,v2_d3t3,D2uDs2,D2vDs2,GeoCur,optimum)
#define repD2PrinciDs2
D2PrinciDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
```

```
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,D2uDs2,D2vDs2,GeoCur,optimum)
#define repD2SurDs2
    D2SurDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,D2uDs2,D2vDs2,GeoCur,optimum,dtangds)
#define repCurvenormaloS
    CurvenormaloS(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,D2uDs2,D2vDs2,GeoCur,optimum,norv)
#define repCurvebinormaloS
    CurvebinormaloS(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,
    u1_dt2,u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,
    u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,
    u2_d2t3,u2_d3t1,u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,
    v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,
    v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2
    ,v2_d2t3,v2_d3t1,v2_d3t2,v2_d3t3,D2uDs2,D2vDs2,GeoCur,optimum,binorv)
#define repAlp2
    Alp2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,D2uDs2,D2vDs2,GeoCur,optimum,Alp2_C)
#define repBeta2
    Beta2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,D2uDs2,D2vDs2,GeoCur,optimum)
```

```
#define opt1 (float pu,float pv,float u1_x1,float u1_x2,float u1_x3,float
    u1_t1,float u1_t2,float u1_t3,float u2_x1,float u2_x2,float u2_x3,float
    u2_t1,float u2_t2,float u2_t3,float v1_x1,float v1_x2,float
    v1_x3,float v1_t1,float v1_t2,float v1_t3,float v2_x1,float
    v2_x2,float v2_x3,float v2_t1,float v2_t2,float v2_t3,float normv_C[])
#define opt2 (float pu,float pv,float u1_x1,float u1_x2,float u1_x3,float
    u1_t1,float u1_t2,float u1_t3,float u2_x1,float u2_x2,float
    u2_x3,float u2_t1,float u2_t2,float u2_t3,float v1_x1,float
    v1_x2,float v1_x3,float v1_t1,float v1_t2,float v1_t3,float
    v2_x1,float v2_x2,float v2_x3,float v2_t1,float v2_t2,float v2_t3)
#define opt3 (float pu,float pv,float u1_x1,float u1_x2,float
    u1_x3,float u1_t1,float u1_t2,float u1_t3,float u1_dt1,float
    u1_dt2,float u1_dt3,float u2_x1,float u2_x2,float
    u2_x3,float u2_t1,float u2_t2,float u2_t3,float u2_dt1,float
    u2_dt2,float u2_dt3,float v1_x1,float v1_x2,float v1_x3,float
    v1_t1,float v1_t2,float v1_t3,float v1_dt1,float v1_dt2,float
    v1_dt3,float v2_x1,float v2_x2,float v2_x3,float v2_t1,float
    v2_t2,float v2_t3,float v2_dt1,float v2_dt2,float v2_dt3)
#define opt4 (float pu,float pv,float u1_x1,float u1_x2,float
    u1_x3,float u1_t1,float u1_t2,float u1_t3,float u1_dt1,float
    u1_dt2,float u1_dt3,float u2_x1,float u2_x2,float u2_x3,float
    u2_t1,float u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float
    u2_dt3,float v1_x1,float v1_x2,float v1_x3,float v1_t1,float
    v1_t2,float v1_t3,float v1_dt1,float v1_dt2,float v1_dt3,float
    v2_x1,float v2_x2,float v2_x3,float v2_t1,float v2_t2,float
    v2_t3,float v2_dt1,float v2_dt2,float v2_dt3,mint optimum)
 #define opt5 (float pu,float pv,float u1_x1,float u1_x2,float u1_x3,float
    u1_t1,float u1_t2,float u1_t3,float u1_dt1,float u1_dt2,float
    u1_dt3,float u2_x1,float u2_x2,float u2_x3,float u2_t1,float
    u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float u2_dt3,float
    v1_x1,float v1_x2,float v1_x3,float v1_t1,float v1_t2,float
    v1_t3,float v1_dt1,float v1_dt2,float v1_dt3,float v2_x1,float
    v2_x2,float v2_x3,float v2_t1,float v2_t2,float v2_t3,float
    v2_dt1,float v2_dt2,float v2_dt3,mint optimum,float tangv_C[])
#define opt6 (float pu,float pv,float u1_x1,float u1_x2,float
    u1_x3,float u1_t1,float u1_t2,float u1_t3,float u1_dt1,float
    u1_dt2,float u1_dt3,float u2_x1,float u2_x2,float u2_x3,float
    u2_t1,float u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float
    u2_dt3,float v1_x1,float v1_x2,float v1_x3,float v1_t1,float
    v1_t2,float v1_t3,float v1_dt1,float v1_dt2,float v1_dt3,float
    v2_x1,float v2_x2,float v2_x3,float v2_t1,float v2_t2,float
    v2_t3,float v2_dt1,float v2_dt2,float v2_dt3,float dnormv_C[])
#define opt7 (float pu,float pv,float u1_x1,float u1_x2,float
    u1_x3,float u1_t1,float u1_t2,float u1_t3,float u1_dt1,float
```

```
    u1_dt2,float u1_dt3,float u2_x1,float u2_x2,float u2_x3,float
    u2_t1,float u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float
    u2_dt3,float v1_x1,float v1_x2,float v1_x3,float v1_t1,float
    v1_t2,float v1_t3,float v1_dt1,float v1_dt2,float v1_dt3,float
    v2_x1,float v2_x2,float v2_x3,float v2_t1,float v2_t2,float
    v2_t3,float v2_dt1,float v2_dt2,float v2_dt3,float Scal[])
#define opt8 (float pu,float pv,float u1_x1,float u1_x2,float
    u1_x3,float u1_t1,float u1_t2,float u1_t3,float u1_dt1,float
    u1_dt2,float u1_dt3,float u1_d2t1,float u1_d2t2,float
    u1_d2t3,float u2_x1,float u2_x2,float u2_x3,float u2_t1,float
    u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float u2_dt3,float
    u2_d2t1,float u2_d2t2,float u2_d2t3,float v1_x1,float
    v1_x2,float v1_x3,float v1_t1,float v1_t2,float v1_t3,float
    v1_dt1,float v1_dt2,float v1_dt3,float v1_d2t1,float
    v1_d2t2,float v1_d2t3,float v2_x1,float v2_x2,float v2_x3,float
    v2_t1,float v2_t2,float v2_t3,float v2_dt1,float v2_dt2,float
    v2_dt3,float v2_d2t1,float v2_d2t2,float v2_d2t3,float coeff[])
#define opt9 (float pu,float pv,float u1_x1,float u1_x2,float
    u1_x3,float u1_t1,float u1_t2,float u1_t3,float u1_dt1,float
    u1_dt2,float u1_dt3,float u1_d2t1,float u1_d2t2,float
    u1_d2t3,float u2_x1,float u2_x2,float u2_x3,float u2_t1,float
    u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float u2_dt3,float
    u2_d2t1,float u2_d2t2,float u2_d2t3,float v1_x1,float v1_x2,float
    v1_x3,float v1_t1,float v1_t2,float v1_t3,float v1_dt1,float
    v1_dt2,float v1_dt3,float v1_d2t1,float v1_d2t2,float
    v1_d2t3,float v2_x1,float v2_x2,float v2_x3,float v2_t1,float
    v2_t2,float v2_t3,float v2_dt1,float v2_dt2,float v2_dt3,float
    v2_d2t1,float v2_d2t2,float v2_d2t3,mint optimum,float coeff[])
#define opt10 (float pu,float pv,float u1_x1,float u1_x2,float
    u1_x3,float u1_t1,float u1_t2,float u1_t3,float u1_dt1,float
    u1_dt2,float u1_dt3,float u1_d2t1,float u1_d2t2,float
    u1_d2t3,float u2_x1,float u2_x2,float u2_x3,float u2_t1,float
    u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float u2_dt3,float
    u2_d2t1,float u2_d2t2,float u2_d2t3,float v1_x1,float
    v1_x2,float v1_x3,float v1_t1,float v1_t2,float v1_t3,float
    v1_dt1,float v1_dt2,float v1_dt3,float v1_d2t1,float
    v1_d2t2,float v1_d2t3,float v2_x1,float v2_x2,float v2_x3,float
    v2_t1,float v2_t2,float v2_t3,float v2_dt1,float v2_dt2,float
    v2_dt3,float v2_d2t1,float v2_d2t2,float v2_d2t3,mint optimum)

#define opt11 (float pu,float pv,float u1_x1,float u1_x2,float u1_x3,float
    u1_t1,float u1_t2,float u1_t3,float u1_dt1,float u1_dt2,float
    u1_dt3,float u2_x1,float u2_x2,float u2_x3,float u2_t1,float
    u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float u2_dt3,float
    v1_x1,float v1_x2,float v1_x3,float v1_t1,float v1_t2,float
    v1_t3,float v1_dt1,float v1_dt2,float v1_dt3,float v2_x1,float
    v2_x2,float v2_x3,float v2_t1,float v2_t2,float v2_t3,float
```

```
    v2_dt1,float v2_dt2,float v2_dt3,mint optimum,float Alph_C[])
#define opt12 (float pu,float pv,float u1_x1,float u1_x2,float u1_x3,float
    u1_t1,float u1_t2,float u1_t3,float u2_x1,float u2_x2,float u2_x3,float
    u2_t1,float u2_t2,float u2_t3,float v1_x1,float v1_x2,float
    v1_x3,float v1_t1,float v1_t2,float v1_t3,float v2_x1,float
    v2_x2,float v2_x3,float v2_t1,float v2_t2,float v2_t3,float Snor_C[])
#define opt13 (float pu,float pv,float u1_x1,float u1_x2,float u1_x3,float
    u1_t1,float u1_t2,float u1_t3,float u1_dt1,float u1_dt2,float
    u1_dt3,float u2_x1,float u2_x2,float u2_x3,float u2_t1,float
    u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float u2_dt3,float
    v1_x1,float v1_x2,float v1_x3,float v1_t1,float v1_t2,float
    v1_t3,float v1_dt1,float v1_dt2,float v1_dt3,float v2_x1,float
    v2_x2,float v2_x3,float v2_t1,float v2_t2,float v2_t3,float
    v2_dt1,float v2_dt2,float v2_dt3,mint optimum,float binormv_C[])
#define opt14 (float pu,float pv,float u1_x1,float u1_x2,float
    u1_x3,float u1_t1,float u1_t2,float u1_t3,float u1_dt1,float
    u1_dt2,float u1_dt3,float u1_d2t1,float u1_d2t2,float
    u1_d2t3,float u2_x1,float u2_x2,float u2_x3,float u2_t1,float
    u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float u2_dt3,float
    u2_d2t1,float u2_d2t2,float u2_d2t3,float v1_x1,float v1_x2,float
    v1_x3,float v1_t1,float v1_t2,float v1_t3,float v1_dt1,float
    v1_dt2,float v1_dt3,float v1_d2t1,float v1_d2t2,float
    v1_d2t3,float v2_x1,float v2_x2,float v2_x3,float v2_t1,float
    v2_t2,float v2_t3,float v2_dt1,float v2_dt2,float v2_dt3,float
    v2_d2t1,float v2_d2t2,float v2_d2t3,mint optimum,float x[])

#define opt15 (int i,float pu[],float pv[],float u1_x1[],float u1_x2[],float
    u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float u2_x1[],float
    u2_x2[],float u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float
    v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float
    v1_t2[],float v1_t3[],float v2_x1[],float v2_x2[],float
    v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float normv_C[])
#define opt16 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u2_x1[],float u2_x2[],float u2_x3[],float u2_t1[],float u2_t2[],float
    u2_t3[],float v1_x1[],float v1_x2[],float v1_x3[],float
    v1_t1[],float v1_t2[],float v1_t3[],float v2_x1[],float
    v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[])
#define opt17 (int i,float pu[],float pv[],float u1_x1[],float u1_x2[],float
    u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float u1_dt1[],float
    u1_dt2[],float u1_dt3[],float u2_x1[],float u2_x2[],float u2_x3[],float
    u2_t1[],float u2_t2[],float u2_t3[],float u2_dt1[],float u2_dt2[],float
    u2_dt3[],float v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float
    v1_t2[],float v1_t3[],float v1_dt1[],float v1_dt2[],float
    v1_dt3[],float v2_x1[],float v2_x2[],float v2_x3[],float v2_t1[],float
    v2_t2[],float v2_t3[],float v2_dt1[],float v2_dt2[],float v2_dt3[])
#define opt18 (int i,float pu[],float pv[],float u1_x1[],float
```

```
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u2_x1[],float
    u2_x2[],float u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float
    u2_dt1[],float u2_dt2[],float u2_dt3[],float v1_x1[],float
    v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float v1_t3[],float
    v1_dt1[],float v1_dt2[],float v1_dt3[],float v2_x1[],float
    v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float
    v2_t3[],float v2_dt1[],float v2_dt2[],float v2_dt3[],mint optimum)
#define opt19 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u2_x1[],float
    u2_x2[],float u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float
    u2_dt1[],float u2_dt2[],float u2_dt3[],float v1_x1[],float
    v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float v1_t3[],float
    v1_dt1[],float v1_dt2[],float v1_dt3[],float v2_x1[],float
    v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float
    v2_dt1[],float v2_dt2[],float v2_dt3[],mint optimum,float tangv_C[])
#define opt20 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u2_x1[],float
    u2_x2[],float u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float
    u2_dt1[],float u2_dt2[],float u2_dt3[],float v1_x1[],float
    v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float v1_t3[],float
    v1_dt1[],float v1_dt2[],float v1_dt3[],float v2_x1[],float
    v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float
    v2_dt1[],float v2_dt2[],float v2_dt3[],float dnormv_C[])
#define opt21 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u1_d2t1[],float
    u1_d2t2[],float u1_d2t3[],float u2_x1[],float u2_x2[],float
    u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float u2_dt1[],float
    u2_dt2[],float u2_dt3[],float u2_d2t1[],float u2_d2t2[],float
    u2_d2t3[],float v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float
    v1_t2[],float v1_t3[],float v1_dt1[],float v1_dt2[],float
    v1_dt3[],float v1_d2t1[],float v1_d2t2[],float v1_d2t3[],float
    v2_x1[],float v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float
    v2_t3[],float v2_dt1[],float v2_dt2[],float v2_dt3[],float
    v2_d2t1[],float v2_d2t2[],float v2_d2t3[],float d2normv_C[])
#define opt22 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u1_d2t1[],float
    u1_d2t2[],float u1_d2t3[],float u2_x1[],float u2_x2[],float
    u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float u2_dt1[],float
    u2_dt2[],float u2_dt3[],float u2_d2t1[],float u2_d2t2[],float
    u2_d2t3[],float v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float
    v1_t2[],float v1_t3[],float v1_dt1[],float v1_dt2[],float
    v1_dt3[],float v1_d2t1[],float v1_d2t2[],float v1_d2t3[],float
```

```
    v2_x1[],float v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float
    v2_t3[],float v2_dt1[],float v2_dt2[],float v2_dt3[],float
    v2_d2t1[],float v2_d2t2[],float v2_d2t3[],float Scal[])
#define opt23 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u1_d2t1[],float
    u1_d2t2[],float u1_d2t3[],float u2_x1[],float u2_x2[],float
    u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float u2_dt1[],float
    u2_dt2[],float u2_dt3[],float u2_d2t1[],float u2_d2t2[],float
    u2_d2t3[],float v1_x1[],float v1_x2[],float v1_x3[],float
    v1_t1[],float v1_t2[],float v1_t3[],float v1_dt1[],float
    v1_dt2[],float v1_dt3[],float v1_d2t1[],float v1_d2t2[],float
    v1_d2t3[],float v2_x1[],float v2_x2[],float v2_x3[],float v2_t1[],float
    v2_t2[],float v2_t3[],float v2_dt1[],float v2_dt2[],float
    v2_dt3[],float v2_d2t1[],float v2_d2t2[],float v2_d2t3[])
#define opt24 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u1_d2t1[],float
    u1_d2t2[],float u1_d2t3[],float u1_d3t1[],float u1_d3t2[],float
    u1_d3t3[],float u2_x1[],float u2_x2[],float u2_x3[],float
    u2_t1[],float u2_t2[],float u2_t3[],float u2_dt1[],float
    u2_dt2[],float u2_dt3[],float u2_d2t1[],float u2_d2t2[],float
    u2_d2t3[],float u2_d3t1[],float u2_d3t2[],float u2_d3t3[],float
    v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float
    v1_t3[],float v1_dt1[],float v1_dt2[],float v1_dt3[],float
    v1_d2t1[],float v1_d2t2[],float v1_d2t3[],float v1_d3t1[],float
    v1_d3t2[],float v1_d3t3[],float v2_x1[],float v2_x2[],float
    v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float v2_dt1[],float
    v2_dt2[],float v2_dt3[],float v2_d2t1[],float v2_d2t2[],float
    v2_d2t3[],float v2_d3t1[],float v2_d3t2[],float v2_d3t3[],float coeff[])
#define opt25 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u1_d2t1[],float
    u1_d2t2[],float u1_d2t3[],float u1_d3t1[],float u1_d3t2[],float
    u1_d3t3[],float u2_x1[],float u2_x2[],float u2_x3[],float u2_t1[],float
    u2_t2[],float u2_t3[],float u2_dt1[],float u2_dt2[],float
    u2_dt3[],float u2_d2t1[],float u2_d2t2[],float u2_d2t3[],float
    u2_d3t1[],float u2_d3t2[],float u2_d3t3[],float v1_x1[],float
    v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float v1_t3[],float
    v1_dt1[],float v1_dt2[],float v1_dt3[],float v1_d2t1[],float
    v1_d2t2[],float v1_d2t3[],float v1_d3t1[],float v1_d3t2[],float
    v1_d3t3[],float v2_x1[],float v2_x2[],float v2_x3[],float v2_t1[],float
    v2_t2[],float v2_t3[],float v2_dt1[],float v2_dt2[],float
    v2_dt3[],float v2_d2t1[],float v2_d2t2[],float v2_d2t3[],float
    v2_d3t1[],float v2_d3t2[],float v2_d3t3[],mint optimum,float coeff[])
#define opt26 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
```

```
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u1_d2t1[],float
    u1_d2t2[],float u1_d2t3[],float u1_d3t1[],float u1_d3t2[],float
    u1_d3t3[],float u2_x1[],float u2_x2[],float u2_x3[],float
    u2_t1[],float u2_t2[],float u2_t3[],float u2_dt1[],float
    u2_dt2[],float u2_dt3[],float u2_d2t1[],float u2_d2t2[],float
    u2_d2t3[],float u2_d3t1[],float u2_d3t2[],float u2_d3t3[],float
    v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float
    v1_t3[],float v1_dt1[],float v1_dt2[],float v1_dt3[],float
    v1_d2t1[],float v1_d2t2[],float v1_d2t3[],float v1_d3t1[],float
    v1_d3t2[],float v1_d3t3[],float v2_x1[],float v2_x2[],float
    v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float v2_dt1[],float
    v2_dt2[],float v2_dt3[],float v2_d2t1[],float v2_d2t2[],float
    v2_d2t3[],float v2_d3t1[],float v2_d3t2[],float v2_d3t3[],mint optimum)
#define opt27 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u1_d2t1[],float
    u1_d2t2[],float u1_d2t3[],float u1_d3t1[],float u1_d3t2[],float
    u1_d3t3[],float u2_x1[],float u2_x2[],float u2_x3[],float
    u2_t1[],float u2_t2[],float u2_t3[],float u2_dt1[],float
    u2_dt2[],float u2_dt3[],float u2_d2t1[],float u2_d2t2[],float
    u2_d2t3[],float u2_d3t1[],float u2_d3t2[],float u2_d3t3[],float
    v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float
    v1_t3[],float v1_dt1[],float v1_dt2[],float v1_dt3[],float
    v1_d2t1[],float v1_d2t2[],float v1_d2t3[],float v1_d3t1[],float
    v1_d3t2[],float v1_d3t3[],float v2_x1[],float v2_x2[],float
    v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float
    v2_dt1[],float v2_dt2[],float v2_dt3[],float v2_d2t1[],float
    v2_d2t2[],float v2_d2t3[],float v2_d3t1[],float v2_d3t2[],float
    v2_d3t3[],float D2uDs2[],float D2vDs2[],float GeoCur[],mint optimum)
#define opt28 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u1_d2t1[],float
    u1_d2t2[],float u1_d2t3[],float u1_d3t1[],float u1_d3t2[],float
    u1_d3t3[],float u2_x1[],float u2_x2[],float u2_x3[],float
    u2_t1[],float u2_t2[],float u2_t3[],float u2_dt1[],float
    u2_dt2[],float u2_dt3[],float u2_d2t1[],float u2_d2t2[],float
    u2_d2t3[],float u2_d3t1[],float u2_d3t2[],float u2_d3t3[],float
    v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float
    v1_t3[],float v1_dt1[],float v1_dt2[],float v1_dt3[],float
    v1_d2t1[],float v1_d2t2[],float v1_d2t3[],float v1_d3t1[],float
    v1_d3t2[],float v1_d3t3[],float v2_x1[],float v2_x2[],float
    v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float v2_dt1[],float
    v2_dt2[],float v2_dt3[],float v2_d2t1[],float v2_d2t2[],float
    v2_d2t3[],float v2_d3t1[],float v2_d3t2[],float v2_d3t3[],float
    D2uDs2[],float D2vDs2[],float GeoCur[],mint optimum,float coeff[])
```

```
__device__ float inicurvature_LAC(float
    t,float Alpha,float Lambda,float InitRC,float Total_s)
{
if(Alpha==-1)
    {return ((float) 1)/InitRC-Lambda*t*Total_s;}
else if(Alpha==0)
    {return ((float) 1)/exp(Lambda*t*Total_s+log(InitRC));}
else if(Alpha==1)
    {return ((float) 1)/(InitRC+Lambda*t*Total_s);}
else if(Alpha==2)
    {return ((float) 1)/sqrt(pow(InitRC,2)+2.0*Lambda*t*Total_s);}
else
    {return pow(pow(InitRC,Alpha)+Lambda*Alpha*t*Total_s,-((float) 1)/Alpha);}
}

__device__ float inidcLACdx(float
    t,float Alpha,float Lambda,float InitRC,float Total_s)
{
if(Alpha==-1)
    {return -Total_s*Lambda;}
else if(Alpha==0)
    {return -Total_s*(Lambda*exp(-Lambda*t*Total_s))/InitRC;}
else if(Alpha==1)
    {return -Total_s*Lambda/pow(InitRC+Lambda*t*Total_s,2);}
else if(Alpha==2)
    {return
    -Total_s*Lambda/((pow(InitRC,2)+2.0*Lambda*t*Total_s)*sqrt(pow(InitRC,2)
    +2.0*Lambda*t*Total_s));}
else
    {return
    -Total_s*Lambda*pow(pow(InitRC,Alpha)+Lambda*Alpha*t*Total_s,-((float)
    1)-((float) 1)/Alpha);}
}
__device__ float inid2cLACdx2(float
    t,float Alpha,float Lambda,float InitRC,float Total_s)
{
if(Alpha==-1)
    {return 0;}
else if(Alpha==0)
    {return
    pow(Total_s,2)*(pow(Lambda,2)*exp(-Lambda*t*Total_s))/InitRC;}
else if(Alpha==1)
    {return pow(Total_s,2)*((float)
    2)*pow(Lambda,2)/pow(InitRC+Lambda*t*Total_s,3);}
else if(Alpha==2)
    {return
    3.0*pow(Total_s,2)*pow(Lambda,2)/(pow(pow(InitRC,2)+2.0*Lambda*t*Total_s
```

```
    ,2)*(sqrt(pow(InitRC,2)+2.0*Lambda*t*Total_s)));}
else
    {return (((float)
    1)+Alpha)*pow(Total_s,2)*pow(Lambda,2)*pow(pow(InitRC,Alpha)+Lambda*
    Alpha*t*Total_s,-((float) 2)-((float) 1)/Alpha);}
}

__device__ float iniTorsion_LASC(float
    t,float Beta,float Omega,float InitRT,float Total_s)
{
if(Beta==-1)
    {return ((float) 1)/InitRT-Omega*t*Total_s;}
else if(Beta==0)
    {return ((float) 1)/exp(Omega*t*Total_s+log(InitRT));}
else if(Beta==1)
    {return ((float) 1)/(InitRT+Omega*t*Total_s);}
else if(Beta==2)
    {return ((float) 1)/sqrt(pow(InitRT,2)+2.0*Omega*t*Total_s);}
else
    {return pow(pow(InitRT,Beta)+Omega*Beta*t*Total_s,-((float) 1)/Beta);}
}

__device__ float inidtLASCdx(float
    t,float Beta,float Omega,float InitRT,float Total_s)
{
if(Beta==-1)
    {return -Total_s*Omega;}
else if(Beta==0)
    {return -Total_s*(Omega*exp(-Omega*t*Total_s))/InitRT;}
else if(Beta==1)
    {return -Total_s*Omega/pow(InitRT+Omega*t*Total_s,2);}
else if(Beta==2)
    {return
    -Total_s*Omega/((pow(InitRT,2)+2.0*Omega*t*Total_s)*sqrt(pow(InitRT,2)+
    2.0*Omega*t*Total_s));}
else
    {return -Total_s*Omega*pow(pow(InitRT,Beta)+Omega*Beta*t*Total_s,-((float)
    1)-((float) 1)/Beta);}
}

__device__ float inid2tLASCdx2(float
    t,float Beta,float Omega,float InitRT,float Total_s)
{
if(Beta==-1)
    {return 0;}
else if(Beta==0)
    {return
```

```
    pow(Total_s,2)*(pow(Omega,2)*exp(-Omega*t*Total_s))/InitRT;}
else if(Beta==1)
    {return pow(Total_s,2)*((float)
    2)*pow(Omega,2)/pow(InitRT+Omega*t*Total_s,3);}
else if(Beta==2)
    {return pow(Total_s,2)*(((float)
    3)*pow(Omega,2))/sqrt(pow(pow(InitRT,2)+((float)
    2)*Omega*t*Total_s,5));}
else
    {return (((float)
    1)+Beta)*pow(Total_s,2)*pow(Omega,2)*pow(pow(InitRT,Beta)+Omega*Beta*t*
    Total_s,-((float) 2)-((float) 1)/Beta);}
}


__device__ void iniSurNorvec opt1
{
    normv_C[0] =inidxdu2*inidxdv3-inidxdu3*inidxdv2;
    normv_C[1] =inidxdu3*inidxdv1-inidxdu1*inidxdv3;
    normv_C[2] =inidxdu1*inidxdv2-inidxdu2*inidxdv1;
}
__device__ void iniDSurNvDu opt6
{
    dnormv_C[0]
    =(inid2xdu22*inidxdv3-inid2xdu23*inidxdv2)+(inidxdu2*inid2xdudv3-
    inidxdu3*inid2xdudv2);
    dnormv_C[1]
    =(inid2xdu23*inidxdv1-inid2xdu21*inidxdv3)+(inidxdu3*inid2xdudv1-
    inidxdu1*inid2xdudv3);
    dnormv_C[2]
    =(inid2xdu21*inidxdv2-inid2xdu22*inidxdv1)+(inidxdu1*inid2xdudv2-
    inidxdu2*inid2xdudv1);
}
__device__ void iniDSurNvDv opt6
{
    dnormv_C[0]
    =(inid2xdudv2*inidxdv3-inid2xdudv3*inidxdv2)+(inidxdu2*inid2xdv23-
    inidxdu3*inid2xdv22);
    dnormv_C[1]
    =(inid2xdudv3*inidxdv1-inid2xdudv1*inidxdv3)+(inidxdu3*inid2xdv21-
    inidxdu1*inid2xdv23);
    dnormv_C[2]
    =(inid2xdudv1*inidxdv2-inid2xdudv2*inidxdv1)+(inidxdu1*inid2xdv22-
    inidxdu2*inid2xdv21);
}

__device__ void iniScalarSurNor opt7
```

```
{
float ScalSNor,DScalSNorDu,DScalSNorDv;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];

repiniSurNorvec;
repiniDSurNvDu;
repiniDSurNvDv;


ScalSNor =
     sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
DScalSNorDu
     =
     (normv_C[0]*dnormvdu_C[0]+normv_C[1]*dnormvdu_C[1]+normv_C[2]*dnormvdu_C
     [2])/ScalSNor;
DScalSNorDv =
     (normv_C[0]*dnormvdv_C[0]+normv_C[1]*dnormvdv_C[1]+normv_C[2]*dnormvdv_C
     [2])/ScalSNor;


Scal[0] = ScalSNor;
Scal[1] = DScalSNorDu;
Scal[2] = DScalSNorDv;
}

__device__ float inicoefE opt2
{
    return inidxdu1*inidxdu1+inidxdu2*inidxdu2+inidxdu3*inidxdu3;
}
__device__ float iniDEDu opt3
{
    return 2.0*(inidxdu1*inid2xdu21+inidxdu2*inid2xdu22+inidxdu3*inid2xdu23);
}
__device__ float iniDEDv opt3
{
    return
     2.0*(inidxdu1*inid2xdudv1+inidxdu2*inid2xdudv2+inidxdu3*inid2xdudv3);
}


__device__ float inicoefF opt2
{
    return (inidxdu1*inidxdv1+inidxdu2*inidxdv2+inidxdu3*inidxdv3);
}
__device__ float iniDFDu opt3
```

```
{
    return inidxdv1*inid2xdu21+inidxdv2*inid2xdu22+inidxdv3*inid2xdu23
    + inidxdu1*inid2xdudv1+inidxdu2*inid2xdudv2+inidxdu3*inid2xdudv3;
}
__device__ float iniDFDv opt3
{
    return
    inidxdv1*inid2xdudv1+inidxdv2*inid2xdudv2+inidxdv3*inid2xdudv3
    + inidxdu1*inid2xdv21+inidxdu2*inid2xdv22+inidxdu3*inid2xdv23;
}

__device__ float inicoefG opt2
{
    return inidxdv1*inidxdv1+inidxdv2*inidxdv2+inidxdv3*inidxdv3;
}
__device__ float iniDGDu opt3
{
    return
    2.0*(inidxdv1*inid2xdudv1+inidxdv2*inid2xdudv2+inidxdv3*inid2xdudv3);
}
__device__ float iniDGDv opt3
{
    return
    2.0*(inidxdv1*inid2xdv21+inidxdv2*inid2xdv22+inidxdv3*inid2xdv23);
}

__device__ float inicoefL opt3
{
float normv_C[3];
repiniSurNorvec;

    return
    (normv_C[0]*inid2xdu21+normv_C[1]*inid2xdu22+normv_C[2]*inid2xdu23)/sqrt
    (normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float inicoefM opt3
{
float normv_C[3];
repiniSurNorvec;

    return
    (normv_C[0]*inid2xdudv1+normv_C[1]*inid2xdudv2+normv_C[2]*inid2xdudv3)/
    sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float inicoefN opt3
{
float normv_C[3];
```

```
repiniSurNorvec;

    return
    (normv_C[0]*inid2xdv21+normv_C[1]*inid2xdv22+normv_C[2]*inid2xdv23)/sqrt
    (normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float iniGaussCurva opt3
{
    return
    (repinicoefL*repinicoefN-pow(repinicoefM,2))/(repinicoefE*repinicoefG-
    pow(repinicoefF,2));
}
__device__ float iniMeanCurva opt3
{
    return
    0.5*(repinicoefE*repinicoefN+repinicoefG*repinicoefL-2*repinicoefF*
    repinicoefM)/(repinicoefE*repinicoefG-pow(repinicoefF,2));
}
__device__ float iniMaxPrinci opt3
{
    return repiniMeanCurva+sqrt(pow(repiniMeanCurva,2)-repiniGaussCurva);
}

__device__ float iniMinPrinci opt3
{
    return repiniMeanCurva-sqrt(pow(repiniMeanCurva,2)-repiniGaussCurva);
}
__device__ float iniPrinci opt4
{
    if(optimum==max)
    {return repiniMaxPrinci;}
    else if(optimum==min)
    {return repiniMinPrinci;}
    else
    {return 0;}
}
__device__ void iniDcoefL_2 opt8
{

float CoeL,DCoeLDu,DCoeLDv;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float Scal[3];

repiniSurNorvec;
repiniDSurNvDu;
```

```
repiniDSurNvDv;
repiniScalarSurNor;

CoeL =
     (normv_C[0]*inid2xdu21+normv_C[1]*inid2xdu22+normv_C[2]*inid2xdu23)/Scal[0];
DCoeLDu
     =
     ((dnormvdu_C[0]*inid2xdu21+dnormvdu_C[1]*inid2xdu22+dnormvdu_C[2]*
     inid2xdu23)+(normv_C[0]*inid3xdu31+normv_C[1]*inid3xdu32+normv_C[2]*
     inid3xdu33)-Scal[1]*CoeL)/Scal[0];
DCoeLDv =
     ((dnormvdv_C[0]*inid2xdu21+dnormvdv_C[1]*inid2xdu22+dnormvdv_C[2]*
     inid2xdu23)+(normv_C[0]*inid3xdu2dv1+normv_C[1]*inid3xdu2dv2+normv_C[2]*
     inid3xdu2dv3)-Scal[2]*CoeL)/Scal[0];

coeff[0] = CoeL;
coeff[1] = DCoeLDu;
coeff[2] = DCoeLDv;

}
__device__ void iniDcoefM_2 opt8
{
float CoeM,DCoeMDu,DCoeMDv;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float Scal[3];
repiniSurNorvec;
repiniDSurNvDu;
repiniDSurNvDv;
repiniScalarSurNor;

CoeM =
     (normv_C[0]*inid2xdudv1+normv_C[1]*inid2xdudv2+normv_C[2]*inid2xdudv3)/
     Scal[0];
DCoeMDu =
     ((dnormvdu_C[0]*inid2xdudv1+dnormvdu_C[1]*inid2xdudv2+dnormvdu_C[2]*
     inid2xdudv3)+(normv_C[0]*inid3xdu2dv1+normv_C[1]*inid3xdu2dv2+normv_C[2]
     *inid3xdu2dv3)-Scal[1]*CoeM)/Scal[0];
DCoeMDv =
     ((dnormvdv_C[0]*inid2xdudv1+dnormvdv_C[1]*inid2xdudv2+dnormvdv_C[2]*
     inid2xdudv3)+(normv_C[0]*inid3xdudv21+normv_C[1]*inid3xdudv22+normv_C[2]
     *inid3xdudv23)-Scal[2]*CoeM)/Scal[0];


coeff[0] = CoeM;
coeff[1] = DCoeMDu;
```

```
coeff[2] = DCoeMDv;
}
__device__ void iniDcoefN_2 opt8
{
float CoeN,DCoeNDu,DCoeNDv;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float Scal[3];
repiniSurNorvec;
repiniDSurNvDu;
repiniDSurNvDv;
repiniScalarSurNor;

CoeN =
    (normv_C[0]*inid2xdv21+normv_C[1]*inid2xdv22+normv_C[2]*inid2xdv23)/Scal[0];
DCoeNDu
    =
    ((dnormvdu_C[0]*inid2xdv21+dnormvdu_C[1]*inid2xdv22+dnormvdu_C[2]*
    inid2xdv23)+(normv_C[0]*inid3xdudv21+normv_C[1]*inid3xdudv22+normv_C[2]*
    inid3xdudv23)-Scal[1]*CoeN)/Scal[0];
DCoeNDv =
    ((dnormvdv_C[0]*inid2xdv21+dnormvdv_C[1]*inid2xdv22+dnormvdv_C[2]*
    inid2xdv23)+(normv_C[0]*inid3xdv31+normv_C[1]*inid3xdv32+normv_C[2]*
    inid3xdv33)-Scal[2]*CoeN)/Scal[0];

coeff[0] = CoeN;
coeff[1] = DCoeNDu;
coeff[2] = DCoeNDv;

}
__device__ void iniDGaussCurva_2 opt8
{
float GCurv,DGCurvDu,DGCurvDv;
float CoeffL[3];
float CoeffM[3];
float CoeffN[3];
repiniDcoefL_2;
repiniDcoefM_2;
repiniDcoefN_2;

GCurv=
    (CoeffL[0]*CoeffN[0]-pow(CoeffM[0],2))/(repinicoefE*repinicoefG-pow(
    repinicoefF,2));
DGCurvDu =
    ((pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repinicoefG*repiniDEDu-2*
    repinicoefF*repiniDFDu+repinicoefE*repiniDGDu)+(-pow(repinicoefF,2)+
```

```
     repinicoefE*repinicoefG)*(CoeffN[0]*CoeffL[1]-2*CoeffM[0]*CoeffM[1]+
     CoeffL[0]*CoeffN[1]))/(pow(pow(repinicoefF,2)-repinicoefE*repinicoefG,2)
     );
DGCurvDv
     =((pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repinicoefG*repiniDEDv-2*
     repinicoefF*repiniDFDv+repinicoefE*repiniDGDv)+(-pow(repinicoefF,2)+
     repinicoefE*repinicoefG)*(CoeffN[0]*CoeffL[2]-2*CoeffM[0]*CoeffM[2]+
     CoeffL[0]*CoeffN[2]))/(pow(pow(repinicoefF,2)-repinicoefE*repinicoefG,2)
     );


coeff[0] = GCurv;
coeff[1] = DGCurvDu;
coeff[2] = DGCurvDv;

}
__device__ void iniDMeanCurva_2 opt8
{
float MCurv,DMCurvDu,DMCurvDv;
float CoeffL[3];
float CoeffM[3];
float CoeffN[3];
repiniDcoefL_2;
repiniDcoefM_2;
repiniDcoefN_2;

MCurv =
     0.5*(repinicoefE*CoeffN[0]+repinicoefG*CoeffL[0]-2*repinicoefF*CoeffM[0]
     )/(repinicoefE*repinicoefG-pow(repinicoefF,2));
DMCurvDu =
     0.5*(-(repinicoefG*CoeffL[0]-2*repinicoefF*CoeffM[0]+repinicoefE*CoeffN[
     0])*(repinicoefG*repiniDEDu-2*repinicoefF*repiniDFDu+repinicoefE*
     repiniDGDu)+(-pow(repinicoefF,2)+repinicoefE*repinicoefG)*(CoeffN[0]*
     repiniDEDu-2*CoeffM[0]*repiniDFDu+CoeffL[0]*repiniDGDu+repinicoefG*
     CoeffL[1]-2*repinicoefF*CoeffM[1]+repinicoefE*CoeffN[1]))/(pow(pow(
     repinicoefF,2)-repinicoefE*repinicoefG,2));
DMCurvDv =
     0.5*(-(repinicoefG*CoeffL[0]-2*repinicoefF*CoeffM[0]+repinicoefE*CoeffN[
     0])*(repinicoefG*repiniDEDv-2*repinicoefF*repiniDFDv+repinicoefE*
     repiniDGDv)+(-pow(repinicoefF,2)+repinicoefE*repinicoefG)*(CoeffN[0]*
     repiniDEDv-2*CoeffM[0]*repiniDFDv+CoeffL[0]*repiniDGDv+repinicoefG*
     CoeffL[2]-2*repinicoefF*CoeffM[2]+repinicoefE*CoeffN[2]))/(pow(pow(
     repinicoefF,2)-repinicoefE*repinicoefG,2));



coeff[0] = MCurv;
```

```
coeff[1] = DMCurvDu;
coeff[2] = DMCurvDv;
}
__device__ void iniDPrinci_2 opt9
{
float Princip,DPrincipDu,DPrincipDv;
float GCurva[3];
float MCurva[3];
repiniDGaussCurva_2;
repiniDMeanCurva_2;

if(optimum==max)
{Princip = MCurva[0]+sqrt(pow(MCurva[0],2)-GCurva[0]);
DPrincipDu =
    MCurva[1]+(2*MCurva[0]*MCurva[1]-GCurva[1])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
DPrincipDv =
    MCurva[2]+(2*MCurva[0]*MCurva[2]-GCurva[2])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));

}
else if(optimum==min)
{Princip = MCurva[0]-sqrt(pow(MCurva[0],2)-GCurva[0]);
DPrincipDu =
    MCurva[1]-(2*MCurva[0]*MCurva[1]-GCurva[1])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
DPrincipDv =
    MCurva[2]-(2*MCurva[0]*MCurva[2]-GCurva[2])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));

}
else
{Princip = 0;
DPrincipDu = 0;
DPrincipDv = 0;
}

coeff[0] = Princip;
coeff[1] = DPrincipDu;
coeff[2] = DPrincipDv;

}

__device__ float iniEta opt4
{
    return
    sign1/sqrt(repinicoefE*pow(repinicoefM-repiniPrinci*repinicoefF,2)-2*
```

```
    repinicoefF*(repinicoefM-repiniPrinci*repinicoefF)*(repinicoefL-
    repiniPrinci*repinicoefE)+repinicoefG*pow(repinicoefL-repiniPrinci*
    repinicoefE,2));
}
__device__ float iniMiu opt4
{
    return
    sign1/sqrt(repinicoefE*pow(repinicoefN-repiniPrinci*repinicoefG,2)-2*
    repinicoefF*(repinicoefM-repiniPrinci*repinicoefF)*(repinicoefN-
    repiniPrinci*repinicoefG)+repinicoefG*pow(repinicoefM-repiniPrinci*
    repinicoefF,2));
}
__device__ float iniDuDs opt4
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-
    repiniPrinci*repinicoefG))
    {return repiniEta*(repinicoefM-repiniPrinci*repinicoefF);}
else
    {return repiniMiu*(repinicoefN-repiniPrinci*repinicoefG);}
}
__device__ float iniDvDs opt4
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
    repinicoefG))
    {return -repiniEta*(repinicoefL-repiniPrinci*repinicoefE);}
else
    {return -repiniMiu*(repinicoefM-repiniPrinci*repinicoefF);}
}


__device__ float iniDEDs opt4
{
    return repiniDEDu*repiniDuDs+repiniDEDv*repiniDvDs;
}
__device__ float iniDFDs opt4
{
    return repiniDFDu*repiniDuDs+repiniDFDv*repiniDvDs;
}
__device__ float iniDGDs opt4
{
    return repiniDGDu*repiniDuDs+repiniDGDv*repiniDvDs;
}
__device__ float iniDLDs opt10
{
float CoeffL[3];
repiniDcoefL_2;
```

```
    return CoeffL[1]*repiniDuDs+CoeffL[2]*repiniDvDs;
}
__device__ float iniDMDs opt10
{
float CoeffM[3];
repiniDcoefM_2;

    return CoeffM[1]*repiniDuDs+CoeffM[2]*repiniDvDs;
}
__device__ float iniDNDs opt10
{
float CoeffN[3];
repiniDcoefN_2;

    return CoeffN[1]*repiniDuDs+CoeffN[2]*repiniDvDs;
}
__device__ float iniDPrinciDs opt10
{
float PCurva[3];
repiniDPrinci_2;

    return PCurva[1]*repiniDuDs+PCurva[2]*repiniDvDs;
}
__device__ float iniDuDt opt4
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
    repinicoefG))
    {return (repinicoefM-repiniPrinci*repinicoefF);}
else
    {return (repinicoefN-repiniPrinci*repinicoefG);}
}
__device__ float iniDvDt opt4
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
    repinicoefG))
    {return (repinicoefL-repiniPrinci*repinicoefE);}
else
    {return (repinicoefM-repiniPrinci*repinicoefF);}
}
__device__ void iniDSurDs opt5
{
    tangv_C[0] =inidxdu1*repiniDuDs+inidxdv1*repiniDvDs;
    tangv_C[1] =inidxdu2*repiniDuDs+inidxdv2*repiniDvDs;
    tangv_C[2] =inidxdu3*repiniDuDs+inidxdv3*repiniDvDs;
}

__device__ void iniAlp1 opt11
```

```
{
    Alph_C[0]=(inid2xdu21*pow(repiniDuDs,2)+2*inid2xdudv1*repiniDuDs*
    repiniDvDs+inid2xdv21*pow(repiniDvDs,2));

    Alph_C[1]=(inid2xdu22*pow(repiniDuDs,2)+2*inid2xdudv2*repiniDuDs*
    repiniDvDs+inid2xdv22*pow(repiniDvDs,2));

    Alph_C[2]=(inid2xdu23*pow(repiniDuDs,2)+2*inid2xdudv3*repiniDuDs*
    repiniDvDs+inid2xdv23*pow(repiniDvDs,2));
}
__device__ void iniSurNor opt12
{
    float normv_C[3];
    repiniSurNorvec;
    Snor_C[0]
    =normv_C[0]/sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*
    normv_C[2]);
    Snor_C[1]
    =normv_C[1]/sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*
    normv_C[2]);
    Snor_C[2]
    =normv_C[2]/sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*
    normv_C[2]);
}

__device__ void iniBinormalSur opt13
{
    float Snor_C[3];
    float tangv[3];
    repiniSurNor;
    repiniDSurDs;

    binormv_C[0] =Snor_C[1]*tangv[2] -Snor_C[2]*tangv[1];
    binormv_C[1] =Snor_C[2]*tangv[0] -Snor_C[0]*tangv[2];
    binormv_C[2] =Snor_C[0]*tangv[1] -Snor_C[1]*tangv[0];
}
__device__ float iniBeta1 opt10
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
    repinicoefG))
    {return
    (-(repiniDLDs-repiniDPrinciDs*repinicoefE-repiniPrinci*repiniDEDs)*
    repiniDuDs-(repiniDMDs-repiniDPrinciDs*repinicoefF-repiniPrinci*
    repiniDFDs)*repiniDvDs);}
else
    {return
    (-(repiniDMDs-repiniDPrinciDs*repinicoefF-repiniPrinci*repiniDFDs)*
```

```
    repiniDuDs-(repiniDNDs-repiniDPrinciDs*repinicoefG-repiniPrinci*
    repiniDGDs)*repiniDvDs);}
}

__device__ void LUDecomposition3x3 opt14
{
    mint rc=3;
    float binormv_C[3];
    float Alp1_C[3];
    repiniBinormalSur;
    repiniAlp1;
    float matrixA[3][3]={

    {repinicoefE,repinicoefF,-(binormv_C[0]*inidxdu1+binormv_C[1]*inidxdu2+
    binormv_C[2]*inidxdu3)},

    {repinicoefF,repinicoefG,-(binormv_C[0]*inidxdv1+binormv_C[1]*inidxdv2+
    binormv_C[2]*inidxdv3)},
    {repiniDvDt,repiniDuDt,0}};

    float
    matrixB[3]={-(Alp1_C[0]*inidxdu1+Alp1_C[1]*inidxdu2+Alp1_C[2]*inidxdu3),
    -(Alp1_C[0]*inidxdv1+Alp1_C[1]*inidxdv2+Alp1_C[2]*inidxdv3),repiniBeta1}
    ;
    float Lower[3][3];
    float Upper[3][3];
    float y[3];
    float sum;
    for(mint ii=0; ii<rc; ii++)
            {
                for(mint jj=0; jj<rc; jj++)
                {
                    sum = 0;
                    if(ii==jj)
                    {
                    Lower[ii][jj]=1;
                    for (mint kk = 0; kk < ii; kk++)
                        {sum += Lower[ii][kk] * Upper[kk][jj];}
                    Upper[ii][jj] = matrixA[ii][jj] - sum;
                    }
                    else if(ii < jj)
                    {
                    Lower[ii][jj]=0;
                    for (mint kk = 0; kk < ii; kk++)
                        {sum += Lower[ii][kk] * Upper[kk][jj];}
                    Upper[ii][jj] = matrixA[ii][jj] - sum;
                    }
```

```
                    else
                    {
                    Upper[ii][jj]=0;
                    for (mint kk = 0; kk < jj; kk++)
                        {sum += Lower[ii][kk] * Upper[kk][jj];}
                    Lower[ii][jj] =(matrixA[ii][jj] - sum)/Upper[jj][jj];
                    }
                }
            }
    for (mint ii = 0; ii < rc; ii++)
    {
        sum = 0;
        for (mint jj = 0; jj < ii; jj++)
            {sum += Lower[ii][jj] * y[jj];}
        y[ii] = matrixB[ii] - sum;
    }
    for (mint ii = rc - 1; ii >= 0; ii--)
    {
        sum = 0;
        for (mint jj = ii + 1; jj < rc; jj++)
            {sum += Upper[ii][jj] * x[jj];}
        x[ii] = (y[ii] - sum)/Upper[ii][ii];
    }
}

__device__ void SurNorvec opt15
{
    normv_C[0] =dxdu2*dxdv3-dxdu3*dxdv2;
    normv_C[1] =dxdu3*dxdv1-dxdu1*dxdv3;
    normv_C[2] =dxdu1*dxdv2-dxdu2*dxdv1;
}
__device__ void DSurNvDu opt20
{
    dnormv_C[0]
     =(d2xdu22*dxdv3-d2xdu23*dxdv2)+(dxdu2*d2xdudv3-dxdu3*d2xdudv2);
    dnormv_C[1]
     =(d2xdu23*dxdv1-d2xdu21*dxdv3)+(dxdu3*d2xdudv1-dxdu1*d2xdudv3);
    dnormv_C[2]
     =(d2xdu21*dxdv2-d2xdu22*dxdv1)+(dxdu1*d2xdudv2-dxdu2*d2xdudv1);

}
__device__ void DSurNvDv opt20
{
    dnormv_C[0]
     =(d2xdudv2*dxdv3-d2xdudv3*dxdv2)+(dxdu2*d2xdv23-dxdu3*d2xdv22);
    dnormv_C[1]
     =(d2xdudv3*dxdv1-d2xdudv1*dxdv3)+(dxdu3*d2xdv21-dxdu1*d2xdv23);
```

```
    dnormv_C[2]
    =(d2xdudv1*dxdv2-d2xdudv2*dxdv1)+(dxdu1*d2xdv22-dxdu2*d2xdv21);

}
__device__ void D2SurNvDu2 opt21
{
    d2normv_C[0]
    =(d3xdu32*dxdv3-d3xdu33*dxdv2)+2.0*(d2xdu22*d2xdudv3-d2xdu23*d2xdudv2)+(
    dxdu2*d3xdu2dv3-dxdu3*d3xdu2dv2);
    d2normv_C[1]
    =(d3xdu33*dxdv1-d3xdu31*dxdv3)+2.0*(d2xdu23*d2xdudv1-d2xdu21*d2xdudv3)+(
    dxdu3*d3xdu2dv1-dxdu1*d3xdu2dv3);
    d2normv_C[2]
    =(d3xdu31*dxdv2-d3xdu32*dxdv1)+2.0*(d2xdu21*d2xdudv2-d2xdu22*d2xdudv1)+(
    dxdu1*d3xdu2dv2-dxdu2*d3xdu2dv1);
}
__device__ void D2SurNvDuDv opt21
{
    d2normv_C[0]
    =(d3xdu2dv2*dxdv3-d3xdu2dv3*dxdv2)+(d2xdu22*d2xdv23-d2xdu23*d2xdv22)+(
    dxdu2*d3xdudv23-dxdu3*d3xdudv22);
    d2normv_C[1]
    =(d3xdu2dv3*dxdv1-d3xdu2dv1*dxdv3)+(d2xdu23*d2xdv21-d2xdu21*d2xdv23)+(
    dxdu3*d3xdudv21-dxdu1*d3xdudv23);
    d2normv_C[2]
    =(d3xdu2dv1*dxdv2-d3xdu2dv2*dxdv1)+(d2xdu21*d2xdv22-d2xdu22*d2xdv21)+(
    dxdu1*d3xdudv22-dxdu2*d3xdudv21);
}
__device__ void D2SurNvDv2 opt21
{
    d2normv_C[0]
    =(d3xdudv22*dxdv3-d3xdudv23*dxdv2)+2.0*(d2xdudv2*d2xdv23-d2xdudv3*
    d2xdv22)+(dxdu2*d3xdv33-dxdu3*d3xdv32);
    d2normv_C[1]
    =(d3xdudv23*dxdv1-d3xdudv21*dxdv3)+2.0*(d2xdudv3*d2xdv21-d2xdudv1*
    d2xdv23)+(dxdu3*d3xdv31-dxdu1*d3xdv33);
    d2normv_C[2]
    =(d3xdudv21*dxdv2-d3xdudv22*dxdv1)+2.0*(d2xdudv1*d2xdv22-d2xdudv2*
    d2xdv21)+(dxdu1*d3xdv32-dxdu2*d3xdv31);
}
__device__ void ScalarSurNor opt22
{
float
    ScalSNor,DScalSNorDu,DScalSNorDv,D2ScalSNorDu2,D2ScalSNorDuDv,D2ScalSNorDv2;
float normv_C[3];
float dnormvdu_C[3];
```

```
float dnormvdv_C[3];
float d2normvdu2_C[3];
float d2normvdudv_C[3];
float d2normvdv2_C[3];
repSurNorvec;
repDSurNvDu;
repDSurNvDv;
repD2SurNvDu2;
repD2SurNvDuDv;
repD2SurNvDv2;

ScalSNor =
    sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
DScalSNorDu
    =
    (normv_C[0]*dnormvdu_C[0]+normv_C[1]*dnormvdu_C[1]+normv_C[2]*dnormvdu_C
    [2])/ScalSNor;
DScalSNorDv =
    (normv_C[0]*dnormvdv_C[0]+normv_C[1]*dnormvdv_C[1]+normv_C[2]*dnormvdv_C
    [2])/ScalSNor;
D2ScalSNorDu2 =
    ((normv_C[0]*d2normvdu2_C[0]+normv_C[1]*d2normvdu2_C[1]+normv_C[2]*
    d2normvdu2_C[2])+(dnormvdu_C[0]*dnormvdu_C[0]+dnormvdu_C[1]*dnormvdu_C[1
    ]+dnormvdu_C[2]*dnormvdu_C[2])-pow(DScalSNorDu,2))/ScalSNor;
D2ScalSNorDuDv =
    ((normv_C[0]*d2normvdudv_C[0]+normv_C[1]*d2normvdudv_C[1]+normv_C[2]*
    d2normvdudv_C[2])+(dnormvdu_C[0]*dnormvdv_C[0]+dnormvdu_C[1]*dnormvdv_C[
    1]+dnormvdu_C[2]*dnormvdv_C[2])-DScalSNorDu*DScalSNorDv)/ScalSNor;
D2ScalSNorDv2 =
    ((normv_C[0]*d2normvdv2_C[0]+normv_C[1]*d2normvdv2_C[1]+normv_C[2]*
    d2normvdv2_C[2])+(dnormvdv_C[0]*dnormvdv_C[0]+dnormvdv_C[1]*dnormvdv_C[1
    ]+dnormvdv_C[2]*dnormvdv_C[2])-pow(DScalSNorDv,2))/ScalSNor;

Scal[0] = ScalSNor;
Scal[1] = DScalSNorDu;
Scal[2] = DScalSNorDv;
Scal[3] = D2ScalSNorDu2;
Scal[4] = D2ScalSNorDuDv;
Scal[5] = D2ScalSNorDv2;
}
__device__ float coefE opt16
{
    return dxdu1*dxdu1+dxdu2*dxdu2+dxdu3*dxdu3;
}
__device__ float DEDu opt17
{
    return 2.0*(dxdu1*d2xdu21+dxdu2*d2xdu22+dxdu3*d2xdu23);
```

```
}
__device__ float DEDv opt17
{
    return 2.0*(dxdu1*d2xdudv1+dxdu2*d2xdudv2+dxdu3*d2xdudv3);
}
__device__ float D2EDu2 opt23
{
    return
    2.0*((d2xdu21*d2xdu21+d2xdu22*d2xdu22+d2xdu23*d2xdu23)+(dxdu1*d3xdu31+
    dxdu2*d3xdu32+dxdu3*d3xdu33));
}
__device__ float D2EDuDv opt23
{
    return
    2.0*((d2xdudv1*d2xdu21+d2xdudv2*d2xdu22+d2xdudv3*d2xdu23)+(dxdu1*
    d3xdu2dv1+dxdu2*d3xdu2dv2+dxdu3*d3xdu2dv3));
}
__device__ float D2EDv2 opt23
{
    return
    2.0*((d2xdudv1*d2xdudv1+d2xdudv2*d2xdudv2+d2xdudv3*d2xdudv3)+(dxdu1*
    d3xdudv21+dxdu2*d3xdudv22+dxdu3*d3xdudv23));
}

__device__ float coefF opt16
{
    return (dxdu1*dxdv1+dxdu2*dxdv2+dxdu3*dxdv3);
}
__device__ float DFDu opt17
{
    return dxdv1*d2xdu21+dxdv2*d2xdu22+dxdv3*d2xdu23
    + dxdu1*d2xdudv1+dxdu2*d2xdudv2+dxdu3*d2xdudv3;
}
__device__ float DFDv opt17
{
    return dxdv1*d2xdudv1+dxdv2*d2xdudv2+dxdv3*d2xdudv3
    + dxdu1*d2xdv21+dxdu2*d2xdv22+dxdu3*d2xdv23;
}
__device__ float D2FDu2 opt23
{
    return
    (dxdv1*d3xdu31+dxdv2*d3xdu32+dxdv3*d3xdu33)+2.0*(d2xdu21*d2xdudv1+
    d2xdu22*d2xdudv2+d2xdu23*d2xdudv3)+(dxdu1*d3xdu2dv1+dxdu2*d3xdu2dv2+
    dxdu3*d3xdu2dv3);
}
__device__ float D2FDuDv opt23
{
```

```
    return
    (dxdv1*d3xdu2dv1+dxdv2*d3xdu2dv2+dxdv3*d3xdu2dv3)+(d2xdu21*d2xdv21+
    d2xdu22*d2xdv22+d2xdu23*d2xdv23)+(d2xdudv1*d2xdudv1+d2xdudv2*d2xdudv2+
    d2xdudv3*d2xdudv3)+(dxdu1*d3xdudv21+dxdu2*d3xdudv22+dxdu3*d3xdudv23);
}
__device__ float D2FDv2 opt23
{
    return
    (dxdv1*d3xdudv21+dxdv2*d3xdudv22+dxdv3*d3xdudv23)+2.0*(d2xdudv1*d2xdv21+
    d2xdudv2*d2xdv22+d2xdudv3*d2xdv23)+(dxdu1*d3xdv31+dxdu2*d3xdv32+dxdu3*
    d3xdv33);
}
__device__ float coefG opt16
{
    return dxdv1*dxdv1+dxdv2*dxdv2+dxdv3*dxdv3;
}
__device__ float DGDu opt17
{
    return 2.0*(dxdv1*d2xdudv1+dxdv2*d2xdudv2+dxdv3*d2xdudv3);
}
__device__ float DGDv opt17
{
    return 2.0*(dxdv1*d2xdv21+dxdv2*d2xdv22+dxdv3*d2xdv23);
}
__device__ float D2GDu2 opt23
{
    return
    2.0*((d2xdudv1*d2xdudv1+d2xdudv2*d2xdudv2+d2xdudv3*d2xdudv3)+(dxdv1*
    d3xdu2dv1+dxdv2*d3xdu2dv2+dxdv3*d3xdu2dv3));
}
__device__ float D2GDuDv opt23
{
    return
    2.0*((d2xdv21*d2xdudv1+d2xdv22*d2xdudv2+d2xdv23*d2xdudv3)+(dxdv1*
    d3xdudv21+dxdv2*d3xdudv22+dxdv3*d3xdudv23));
}
__device__ float D2GDv2 opt23
{
    return
    2.0*((d2xdv21*d2xdv21+d2xdv22*d2xdv22+d2xdv23*d2xdv23)+(dxdv1*d3xdv31+
    dxdv2*d3xdv32+dxdv3*d3xdv33));
}
__device__ float coefL opt17
{
float normv_C[3];
repSurNorvec;
```

```
    return
     (normv_C[0]*d2xdu21+normv_C[1]*d2xdu22+normv_C[2]*d2xdu23)/sqrt(normv_C[
     0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float coefM opt17
{
float normv_C[3];
repSurNorvec;

    return
     (normv_C[0]*d2xdudv1+normv_C[1]*d2xdudv2+normv_C[2]*d2xdudv3)/sqrt(
     normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float coefN opt17
{
float normv_C[3];
repSurNorvec;

    return
     (normv_C[0]*d2xdv21+normv_C[1]*d2xdv22+normv_C[2]*d2xdv23)/sqrt(normv_C[
     0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float GaussCurva opt17
{
    return
     (repcoefL*repcoefN-pow(repcoefM,2))/(repcoefE*repcoefG-pow(repcoefF,2));

}
__device__ float MeanCurva opt17
{
    return
     0.5*(repcoefE*repcoefN+repcoefG*repcoefL-2*repcoefF*repcoefM)/(repcoefE*
     repcoefG-pow(repcoefF,2));
}
__device__ float MaxPrinci opt17
{
    return repMeanCurva+sqrt(pow(repMeanCurva,2)-repGaussCurva);
}

__device__ float MinPrinci opt17
{
    return repMeanCurva-sqrt(pow(repMeanCurva,2)-repGaussCurva);
}
__device__ float Princi opt18
{
    if(optimum==max)
    {return repMaxPrinci;}
```

```
    else if(optimum==min)
    {return repMinPrinci;}
    else
    {return 0;}
}
__device__ void DcoefL_2 opt24
{
float CoeL,DCoeLDu,DCoeLDv,D2CoeLDu2,D2CoeLDuDv,D2CoeLDv2;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float d2normvdu2_C[3];
float d2normvdudv_C[3];
float d2normvdv2_C[3];
float Scal[6];
repSurNorvec;
repDSurNvDu;
repDSurNvDv;
repD2SurNvDu2;
repD2SurNvDuDv;
repD2SurNvDv2;
repScalarSurNor;

CoeL = (normv_C[0]*d2xdu21+normv_C[1]*d2xdu22+normv_C[2]*d2xdu23)/Scal[0];
DCoeLDu =
    ((dnormvdu_C[0]*d2xdu21+dnormvdu_C[1]*d2xdu22+dnormvdu_C[2]*d2xdu23)+(
    normv_C[0]*d3xdu31+normv_C[1]*d3xdu32+normv_C[2]*d3xdu33)-Scal[1]*CoeL)/
    Scal[0];
DCoeLDv =
    ((dnormvdv_C[0]*d2xdu21+dnormvdv_C[1]*d2xdu22+dnormvdv_C[2]*d2xdu23)+(
    normv_C[0]*d3xdu2dv1+normv_C[1]*d3xdu2dv2+normv_C[2]*d3xdu2dv3)-Scal[2]*
    CoeL)/Scal[0];
D2CoeLDu2 =
    ((d2normvdu2_C[0]*d2xdu21+d2normvdu2_C[1]*d2xdu22+d2normvdu2_C[2]*
    d2xdu23)+2.0*(dnormvdu_C[0]*d3xdu31+dnormvdu_C[1]*d3xdu32+dnormvdu_C[2]*
    d3xdu33)+(normv_C[0]*d4xdu41+normv_C[1]*d4xdu42+normv_C[2]*d4xdu43)-Scal
    [3]*CoeL-2*Scal[1]*DCoeLDu)/Scal[0];
D2CoeLDuDv =
    ((d2normvdudv_C[0]*d2xdu21+d2normvdudv_C[1]*d2xdu22+d2normvdudv_C[2]*
    d2xdu23)+(dnormvdu_C[0]*d3xdu2dv1+dnormvdu_C[1]*d3xdu2dv2+dnormvdu_C[2]*
    d3xdu2dv3)+(dnormvdv_C[0]*d3xdu31+dnormvdv_C[1]*d3xdu32+dnormvdv_C[2]*
    d3xdu33)+(normv_C[0]*d4xdu3dv1+normv_C[1]*d4xdu3dv2+normv_C[2]*d4xdu3dv3
    )-Scal[4]*CoeL-Scal[1]*DCoeLDv-Scal[2]*DCoeLDu)/Scal[0];
D2CoeLDv2 =
    ((d2normvdv2_C[0]*d2xdu21+d2normvdv2_C[1]*d2xdu22+d2normvdv2_C[2]*
    d2xdu23)+2.0*(dnormvdv_C[0]*d3xdu2dv1+dnormvdv_C[1]*d3xdu2dv2+dnormvdv_C
    [2]*d3xdu2dv3)+(normv_C[0]*d4xdu2dv21+normv_C[1]*d4xdu2dv22+normv_C[2]*
```

```
     d4xdu2dv23)-Scal[5]*CoeL-2*Scal[2]*DCoeLDv)/Scal[0];

coeff[0] = CoeL;
coeff[1] = DCoeLDu;
coeff[2] = DCoeLDv;
coeff[3] = D2CoeLDu2;
coeff[4] = D2CoeLDuDv;
coeff[5] = D2CoeLDv2;
}
__device__ void DcoefM_2 opt24
{
float CoeM,DCoeMDu,DCoeMDv,D2CoeMDu2,D2CoeMDuDv,D2CoeMDv2;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float d2normvdu2_C[3];
float d2normvdudv_C[3];
float d2normvdv2_C[3];
float Scal[6];
repSurNorvec;
repDSurNvDu;
repDSurNvDv;
repD2SurNvDu2;
repD2SurNvDuDv;
repD2SurNvDv2;
repScalarSurNor;

CoeM = (normv_C[0]*d2xdudv1+normv_C[1]*d2xdudv2+normv_C[2]*d2xdudv3)/Scal[0];
DCoeMDu =
     ((dnormvdu_C[0]*d2xdudv1+dnormvdu_C[1]*d2xdudv2+dnormvdu_C[2]*d2xdudv3)+
     (normv_C[0]*d3xdu2dv1+normv_C[1]*d3xdu2dv2+normv_C[2]*d3xdu2dv3)-Scal[1]
     *CoeM)/Scal[0];
DCoeMDv =
     ((dnormvdv_C[0]*d2xdudv1+dnormvdv_C[1]*d2xdudv2+dnormvdv_C[2]*d2xdudv3)+
     (normv_C[0]*d3xdudv21+normv_C[1]*d3xdudv22+normv_C[2]*d3xdudv23)-Scal[2]
     *CoeM)/Scal[0];
D2CoeMDu2 =
     ((d2normvdu2_C[0]*d2xdudv1+d2normvdu2_C[1]*d2xdudv2+d2normvdu2_C[2]*
     d2xdudv3)+2.0*(dnormvdu_C[0]*d3xdu2dv1+dnormvdu_C[1]*d3xdu2dv2+
     dnormvdu_C[2]*d3xdu2dv3)+(normv_C[0]*d4xdu3dv1+normv_C[1]*d4xdu3dv2+
     normv_C[2]*d4xdu3dv3)-Scal[3]*CoeM-2*Scal[1]*DCoeMDu)/Scal[0];
D2CoeMDuDv =
     ((d2normvdudv_C[0]*d2xdudv1+d2normvdudv_C[1]*d2xdudv2+d2normvdudv_C[2]*
     d2xdudv3)+(dnormvdu_C[0]*d3xdudv21+dnormvdu_C[1]*d3xdudv22+dnormvdu_C[2]
     *d3xdudv23)+(dnormvdv_C[0]*d3xdu2dv1+dnormvdv_C[1]*d3xdu2dv2+dnormvdv_C[
     2]*d3xdu2dv3)+(normv_C[0]*d4xdu2dv21+normv_C[1]*d4xdu2dv22+normv_C[2]*
     d4xdu2dv23)-Scal[4]*CoeM-Scal[1]*DCoeMDv-Scal[2]*DCoeMDu)/Scal[0];
```

```
D2CoeMDv2 =
    ((d2normvdv2_C[0]*d2xdudv1+d2normvdv2_C[1]*d2xdudv2+d2normvdv2_C[2]*
    d2xdudv3)+2.0*(dnormvdv_C[0]*d3xdudv21+dnormvdv_C[1]*d3xdudv22+
    dnormvdv_C[2]*d3xdudv23)+(normv_C[0]*d4xdudv31+normv_C[1]*d4xdudv32+
    normv_C[2]*d4xdudv33)-Scal[5]*CoeM-2*Scal[2]*DCoeMDv)/Scal[0];

coeff[0] = CoeM;
coeff[1] = DCoeMDu;
coeff[2] = DCoeMDv;
coeff[3] = D2CoeMDu2;
coeff[4] = D2CoeMDuDv;
coeff[5] = D2CoeMDv2;
}
__device__ void DcoefN_2 opt24
{
float CoeN,DCoeNDu,DCoeNDv,D2CoeNDu2,D2CoeNDuDv,D2CoeNDv2;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float d2normvdu2_C[3];
float d2normvdudv_C[3];
float d2normvdv2_C[3];
float Scal[6];
repSurNorvec;
repDSurNvDu;
repDSurNvDv;
repD2SurNvDu2;
repD2SurNvDuDv;
repD2SurNvDv2;
repScalarSurNor;

CoeN = (normv_C[0]*d2xdv21+normv_C[1]*d2xdv22+normv_C[2]*d2xdv23)/Scal[0];
DCoeNDu =
    ((dnormvdu_C[0]*d2xdv21+dnormvdu_C[1]*d2xdv22+dnormvdu_C[2]*d2xdv23)+(
    normv_C[0]*d3xdudv21+normv_C[1]*d3xdudv22+normv_C[2]*d3xdudv23)-Scal[1]*
    CoeN)/Scal[0];
DCoeNDv =
    ((dnormvdv_C[0]*d2xdv21+dnormvdv_C[1]*d2xdv22+dnormvdv_C[2]*d2xdv23)+(
    normv_C[0]*d3xdv31+normv_C[1]*d3xdv32+normv_C[2]*d3xdv33)-Scal[2]*CoeN)/
    Scal[0];
D2CoeNDu2 =
    ((d2normvdu2_C[0]*d2xdv21+d2normvdu2_C[1]*d2xdv22+d2normvdu2_C[2]*
    d2xdv23)+2.0*(dnormvdu_C[0]*d3xdudv21+dnormvdu_C[1]*d3xdudv22+dnormvdu_C
    [2]*d3xdudv23)+(normv_C[0]*d4xdu2dv21+normv_C[1]*d4xdu2dv22+normv_C[2]*
    d4xdu2dv23)-Scal[3]*CoeN-2.0*Scal[1]*DCoeNDu)/Scal[0];
D2CoeNDuDv =
    ((d2normvdudv_C[0]*d2xdv21+d2normvdudv_C[1]*d2xdv22+d2normvdudv_C[2]*
```

```
     d2xdv23)+(dnormvdu_C[0]*d3xdv31+dnormvdu_C[1]*d3xdv32+dnormvdu_C[2]*
     d3xdv33)+(dnormvdv_C[0]*d3xdudv21+dnormvdv_C[1]*d3xdudv22+dnormvdv_C[2]*
     d3xdudv23)+(normv_C[0]*d4xdudv31+normv_C[1]*d4xdudv32+normv_C[2]*
     d4xdudv33)-Scal[4]*CoeN-Scal[1]*DCoeNDv-Scal[2]*DCoeNDu)/Scal[0];
D2CoeNDv2 =
     ((d2normvdv2_C[0]*d2xdv21+d2normvdv2_C[1]*d2xdv22+d2normvdv2_C[2]*
     d2xdv23)+2.0*(dnormvdv_C[0]*d3xdv31+dnormvdv_C[1]*d3xdv32+dnormvdv_C[2]*
     d3xdv33)+(normv_C[0]*d4xdv41+normv_C[1]*d4xdv42+normv_C[2]*d4xdv43)-Scal
     [5]*CoeN-2.0*Scal[2]*DCoeNDv)/Scal[0];

coeff[0] = CoeN;
coeff[1] = DCoeNDu;
coeff[2] = DCoeNDv;
coeff[3] = D2CoeNDu2;
coeff[4] = D2CoeNDuDv;
coeff[5] = D2CoeNDv2;
}
__device__ void DGaussCurva_2 opt24
{
float GCurv,DGCurvDu,DGCurvDv,D2GCurvDu2,D2GCurvDuDv,D2GCurvDv2;
float CoeffL[6];
float CoeffM[6];
float CoeffN[6];
repDcoefL_2;
repDcoefM_2;
repDcoefN_2;

GCurv=
     (CoeffL[0]*CoeffN[0]-pow(CoeffM[0],2))/(repcoefE*repcoefG-pow(repcoefF,2));
DGCurvDu
     =
     ((pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repcoefG*repDEDu-2.0*repcoefF*
     repDFDu+repcoefE*repDGDu)+(-pow(repcoefF,2)+repcoefE*repcoefG)*(CoeffN[0
     ]*CoeffL[1]-2.0*CoeffM[0]*CoeffM[1]+CoeffL[0]*CoeffN[1]))/(pow(pow(
     repcoefF,2)-repcoefE*repcoefG,2));
DGCurvDv
     =((pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repcoefG*repDEDv-2.0*repcoefF*
     repDFDv+repcoefE*repDGDv)+(-pow(repcoefF,2)+repcoefE*repcoefG)*(CoeffN[0
     ]*CoeffL[2]-2.0*CoeffM[0]*CoeffM[2]+CoeffL[0]*CoeffN[2]))/(pow(pow(
     repcoefF,2)-repcoefE*repcoefG,2));
D2GCurvDu2 =
     ((-2.0*(pow(repcoefF,2)-repcoefE*repcoefG)*(repcoefG*repDEDu-2.0*
     repcoefF*repDFDu+repcoefE*repDGDu)*(CoeffN[0]*CoeffL[1]-2.0*CoeffM[0]*
     CoeffM[1]+CoeffL[0]*CoeffN[1])+(pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(
     2.0*pow(repcoefG*repDEDu-2.0*repcoefF*repDFDu+repcoefE*repDGDu,2)-(pow(
     repcoefF,2)-repcoefE*repcoefG)*(2.0*pow(repDFDu,2)-2.0*repDEDu*repDGDu-
     repcoefG*repD2EDu2+2.0*repcoefF*repD2FDu2-repcoefE*repD2GDu2))+pow(pow(
```

```
    repcoefF,2)-repcoefE*repcoefG,2)*(2.0*pow(CoeffM[1],2)-2.0*CoeffL[1]*
    CoeffN[1]-CoeffN[0]*CoeffL[3]+2.0*CoeffM[0]*CoeffM[3]-CoeffL[0]*CoeffN[3
    ]))/pow(pow(repcoefF,2)-repcoefE*repcoefG,3));
D2GCurvDuDv =
    -((2.0*(pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repcoefG*repDEDv-2.0*
    repcoefF*repDFDv+repcoefE*repDGDv)*(repcoefG*repDEDu-2.0*repcoefF*
    repDFDu+repcoefE*repDGDu)+(-pow(repcoefF,2)+repcoefE*repcoefG)*(CoeffN[0
    ]*CoeffL[2]-2.0*CoeffM[0]*CoeffM[2]+CoeffL[0]*CoeffN[2])*(repcoefG*
    repDEDu-2.0*repcoefF*repDFDu+repcoefE*repDGDu)+(-pow(repcoefF,2)+
    repcoefE*repcoefG)*(repcoefG*repDEDv-2.0*repcoefF*repDFDv+repcoefE*
    repDGDv)*(CoeffN[0]*CoeffL[1]-2.0*CoeffM[0]*CoeffM[1]+CoeffL[0]*CoeffN[1
    ])+(-pow(repcoefF,2)+repcoefE*repcoefG)*(-pow(CoeffM[0],2)+CoeffL[0]*
    CoeffN[0])*(repDGDv*repDEDu-2.0*repDFDv*repDFDu+repDEDv*repDGDu+repcoefG
    *repD2EDuDv-2.0*repcoefF*repD2FDuDv+repcoefE*repD2GDuDv)-pow(pow(
    repcoefF,2)-repcoefE*repcoefG,2)*(CoeffN[2]*CoeffL[1]-2.0*CoeffM[2]*
    CoeffM[1]+CoeffL[2]*CoeffN[1]+CoeffN[0]*CoeffL[4]-2.0*CoeffM[0]*CoeffM[4
    ]+CoeffL[0]*CoeffN[4]))/pow(-pow(repcoefF,2)+repcoefE*repcoefG,3));
D2GCurvDv2 =
    ((-2.0*(pow(repcoefF,2)-repcoefE*repcoefG)*(repcoefG*repDEDv-2.0*
    repcoefF*repDFDv+repcoefE*repDGDv)*(CoeffN[0]*CoeffL[2]-2.0*CoeffM[0]*
    CoeffM[2]+CoeffL[0]*CoeffN[2])+(pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(
    2.0*pow(repcoefG*repDEDv-2.0*repcoefF*repDFDv+repcoefE*repDGDv,2)-(pow(
    repcoefF,2)-repcoefE*repcoefG)*(2.0*pow(repDFDv,2)-2.0*repDEDv*repDGDv-
    repcoefG*repD2EDv2+2.0*repcoefF*repD2FDv2-repcoefE*repD2GDv2))+pow(pow(
    repcoefF,2)-repcoefE*repcoefG,2)*(2.0*pow(CoeffM[2],2)-2.0*CoeffL[2]*
    CoeffN[2]-CoeffN[0]*CoeffL[5]+2.0*CoeffM[0]*CoeffM[5]-CoeffL[0]*CoeffN[5
    ]))/pow(pow(repcoefF,2)-repcoefE*repcoefG,3));

coeff[0] = GCurv;
coeff[1] = DGCurvDu;
coeff[2] = DGCurvDv;
coeff[3] = D2GCurvDu2;
coeff[4] = D2GCurvDuDv;
coeff[5] = D2GCurvDv2;
}
__device__ void DMeanCurva_2 opt24
{
float MCurv,DMCurvDu,DMCurvDv,D2MCurvDu2,D2MCurvDuDv,D2MCurvDv2;
float CoeffL[6];
float CoeffM[6];
float CoeffN[6];
repDcoefL_2;
repDcoefM_2;
repDcoefN_2;

MCurv =
    0.5*(repcoefE*CoeffN[0]+repcoefG*CoeffL[0]-2.0*repcoefF*CoeffM[0])/(
```

```
        repcoefE*repcoefG-pow(repcoefF,2));
    DMCurvDu =
        0.5*(-(repcoefG*CoeffL[0]-2.0*repcoefF*CoeffM[0]+repcoefE*CoeffN[0])*(
        repcoefG*repDEDu-2.0*repcoefF*repDFDu+repcoefE*repDGDu)+(-pow(repcoefF,2
        )+repcoefE*repcoefG)*(CoeffN[0]*repDEDu-2.0*CoeffM[0]*repDFDu+CoeffL[0]*
        repDGDu+repcoefG*CoeffL[1]-2.0*repcoefF*CoeffM[1]+repcoefE*CoeffN[1]))/(
        pow(pow(repcoefF,2)-repcoefE*repcoefG,2));
    DMCurvDv =
        0.5*(-(repcoefG*CoeffL[0]-2.0*repcoefF*CoeffM[0]+repcoefE*CoeffN[0])*(
        repcoefG*repDEDv-2.0*repcoefF*repDFDv+repcoefE*repDGDv)+(-pow(repcoefF,2
        )+repcoefE*repcoefG)*(CoeffN[0]*repDEDv-2.0*CoeffM[0]*repDFDv+CoeffL[0]*
        repDGDv+repcoefG*CoeffL[2]-2.0*repcoefF*CoeffM[2]+repcoefE*CoeffN[2]))/(
        pow(pow(repcoefF,2)-repcoefE*repcoefG,2));
    D2MCurvDu2 =
        -((2.0*(pow(repcoefF,2)-repcoefE*repcoefG)*(repcoefG*repDEDu-2.0*
        repcoefF*repDFDu+repcoefE*repDGDu)*(CoeffN[0]*repDEDu-2.0*CoeffM[0]*
        repDFDu+CoeffL[0]*repDGDu+repcoefG*CoeffL[1]-2.0*repcoefF*CoeffM[1]+
        repcoefE*CoeffN[1])+(repcoefG*CoeffL[0]-2.0*repcoefF*CoeffM[0]+repcoefE*
        CoeffN[0])*(2.0*pow(repcoefG*repDEDu-2.0*repcoefF*repDFDu+repcoefE*
        repDGDu,2)-(pow(repcoefF,2)-repcoefE*repcoefG)*(2.0*pow(repDFDu,2)-2.0*
        repDEDu*repDGDu-repcoefG*repD2EDu2+2.0*repcoefF*repD2FDu2-repcoefE*
        repD2GDu2))+pow(pow(repcoefF,2)-repcoefE*repcoefG,2)*(2.0*repDGDu*CoeffL
        [1]-4*repDFDu*CoeffM[1]+2.0*repDEDu*CoeffN[1]+CoeffN[0]*repD2EDu2-2.0*
        CoeffM[0]*repD2FDu2+CoeffL[0]*repD2GDu2+repcoefG*CoeffL[3]-2.0*repcoefF*
        CoeffM[3]+repcoefE*CoeffN[3]))/(2.0*pow(pow(repcoefF,2)-repcoefE*
        repcoefG,3)));
    D2MCurvDuDv =
        -(0.5*(-2.0*(repcoefG*CoeffL[0]-2.0*repcoefF*CoeffM[0]+repcoefE*CoeffN[0
        ])*(repcoefG*repDEDv-2.0*repcoefF*repDFDv+repcoefE*repDGDv)*(repcoefG*
        repDEDu-2.0*repcoefF*repDFDu+repcoefE*repDGDu)+(-pow(repcoefF,2)+
        repcoefE*repcoefG)*(CoeffN[0]*repDEDv-2.0*CoeffM[0]*repDFDv+CoeffL[0]*
        repDGDv+repcoefG*CoeffL[2]-2.0*repcoefF*CoeffM[2]+repcoefE*CoeffN[2])*(
        repcoefG*repDEDu-2.0*repcoefF*repDFDu+repcoefE*repDGDu)+(-pow(repcoefF,2
        )+repcoefE*repcoefG)*(repcoefG*repDEDv-2.0*repcoefF*repDFDv+repcoefE*
        repDGDv)*(CoeffN[0]*repDEDu-2.0*CoeffM[0]*repDFDu+CoeffL[0]*repDGDu+
        repcoefG*CoeffL[1]-2.0*repcoefF*CoeffM[1]+repcoefE*CoeffN[1])+(-pow(
        repcoefF,2)+repcoefE*repcoefG)*(repcoefG*CoeffL[0]-2.0*repcoefF*CoeffM[0
        ]+repcoefE*CoeffN[0])*(repDGDv*repDEDu-2.0*repDFDv*repDFDu+repDEDv*
        repDGDu+repcoefG*repD2EDuDv-2.0*repcoefF*repD2FDuDv+repcoefE*repD2GDuDv)
        -pow(pow(repcoefF,2)-repcoefE*repcoefG,2)*(CoeffN[2]*repDEDu-2.0*CoeffM[
        2]*repDFDu+CoeffL[2]*repDGDu+repDGDv*CoeffL[1]-2.0*repDFDv*CoeffM[1]+
        repDEDv*CoeffN[1]+CoeffN[0]*repD2EDuDv-2.0*CoeffM[0]*repD2FDuDv+CoeffL[0
        ]*repD2GDuDv+repcoefG*CoeffL[4]-2.0*repcoefF*CoeffM[4]+repcoefE*CoeffN[4
        ]))/(pow(-pow(repcoefF,2)+repcoefE*repcoefG,3)));
    D2MCurvDv2 =
        -(0.5*(2.0*(pow(repcoefF,2)-repcoefE*repcoefG)*(repcoefG*repDEDv-2.0*
        repcoefF*repDFDv+repcoefE*repDGDv)*(CoeffN[0]*repDEDv-2.0*CoeffM[0]*
```

```
    repDFDv+CoeffL[0]*repDGDv+repcoefG*CoeffL[2]-2.0*repcoefF*CoeffM[2]+
    repcoefE*CoeffN[2])+(repcoefG*CoeffL[0]-2.0*repcoefF*CoeffM[0]+repcoefE*
    CoeffN[0])*(2.0*pow(repcoefG*repDEDv-2.0*repcoefF*repDFDv+repcoefE*
    repDGDv,2)-(pow(repcoefF,2)-repcoefE*repcoefG)*(2.0*pow(repDFDv,2)-2.0*
    repDEDv*repDGDv-repcoefG*repD2EDv2+2.0*repcoefF*repD2FDv2-repcoefE*
    repD2GDv2))+pow(pow(repcoefF,2)-repcoefE*repcoefG,2)*(2.0*repDGDv*CoeffL
    [2]-4*repDFDv*CoeffM[2]+2.0*repDEDv*CoeffN[2]+CoeffN[0]*repD2EDv2-2.0*
    CoeffM[0]*repD2FDv2+CoeffL[0]*repD2GDv2+repcoefG*CoeffL[5]-2.0*repcoefF*
    CoeffM[5]+repcoefE*CoeffN[5]))/(pow(pow(repcoefF,2)-repcoefE*repcoefG,3)
    ));


coeff[0] = MCurv;
coeff[1] = DMCurvDu;
coeff[2] = DMCurvDv;
coeff[3] = D2MCurvDu2;
coeff[4] = D2MCurvDuDv;
coeff[5] = D2MCurvDv2;
}
__device__ void DPrinci_2 opt25
{
float Princip,DPrincipDu,DPrincipDv,D2PrincipDu2,D2PrincipDuDv,D2PrincipDv2;
float GCurva[6];
float MCurva[6];
repDGaussCurva_2;
repDMeanCurva_2;

if(optimum==max)
{Princip = MCurva[0]+sqrt(pow(MCurva[0],2)-GCurva[0]);
DPrincipDu =
    MCurva[1]+(2.0*MCurva[0]*MCurva[1]-GCurva[1])/(2.0*sqrt(pow(MCurva[0],2)
    -GCurva[0]));
DPrincipDv =
    MCurva[2]+(2.0*MCurva[0]*MCurva[2]-GCurva[2])/(2.0*sqrt(pow(MCurva[0],2)
    -GCurva[0]));
D2PrincipDu2 =
    -(pow(-2.0*MCurva[0]*MCurva[1]+GCurva[1],2))/(4*sqrt(pow(pow(MCurva[0],2
    )-GCurva[0],3)))+MCurva[3]+(2.0*pow(MCurva[1],2)+2.0*MCurva[0]*MCurva[3]
    -GCurva[3])/(2.0*sqrt(pow(MCurva[0],2)-GCurva[0]));
D2PrincipDuDv =
    -(((2.0*MCurva[0]*MCurva[2]-GCurva[2])*(2.0*MCurva[0]*MCurva[1]-GCurva[1
    ]))/(4*sqrt(pow(pow(MCurva[0],2)-GCurva[0],3))))+MCurva[4]+(2.0*MCurva[2
    ]*MCurva[1]+2.0*MCurva[0]*MCurva[4]-GCurva[4])/(2.0*sqrt(pow(MCurva[0],2
    )-GCurva[0]));
D2PrincipDv2 =
    -(pow(-2.0*MCurva[0]*MCurva[2]+GCurva[2],2))/(4*sqrt(pow(pow(MCurva[0],2
    )-GCurva[0],3)))+MCurva[5]+(2.0*pow(MCurva[2],2)+2.0*MCurva[0]*MCurva[5]
```

```
    -GCurva[5])/(2.0*sqrt(pow(MCurva[0],2)-GCurva[0]));
}
else if(optimum==min)
{Princip = MCurva[0]-sqrt(pow(MCurva[0],2)-GCurva[0]);
DPrincipDu =
    MCurva[1]-(2.0*MCurva[0]*MCurva[1]-GCurva[1])/(2.0*sqrt(pow(MCurva[0],2)
    -GCurva[0]));
DPrincipDv =
    MCurva[2]-(2.0*MCurva[0]*MCurva[2]-GCurva[2])/(2.0*sqrt(pow(MCurva[0],2)
    -GCurva[0]));
D2PrincipDu2 =
    (pow(-2.0*MCurva[0]*MCurva[1]+GCurva[1],2))/(4*sqrt(pow(pow(MCurva[0],2)
    -GCurva[0],3)))+MCurva[3]+(-2.0*(pow(MCurva[1],2)+MCurva[0]*MCurva[3])+
    GCurva[3])/(2.0*sqrt(pow(MCurva[0],2)-GCurva[0]));
D2PrincipDuDv =
    ((2.0*MCurva[0]*MCurva[2]-GCurva[2])*(2.0*MCurva[0]*MCurva[1]-GCurva[1])
    )/(4*sqrt(pow(pow(MCurva[0],2)-GCurva[0],3)))+MCurva[4]+(-2.0*MCurva[2]*
    MCurva[1]-2.0*MCurva[0]*MCurva[4]+GCurva[4])/(2.0*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
D2PrincipDv2
    =(pow(-2.0*MCurva[0]*MCurva[2]+GCurva[2],2))/(4*sqrt(pow(pow(MCurva[0],2
    )-GCurva[0],3)))+MCurva[5]+(-2.0*(pow(MCurva[2],2)+MCurva[0]*MCurva[5])+
    GCurva[5])/(2.0*sqrt(pow(MCurva[0],2)-GCurva[0]));
}
else
{Princip = 0;
DPrincipDu = 0;
DPrincipDv = 0;
D2PrincipDu2 = 0;
D2PrincipDuDv = 0;
D2PrincipDv2 = 0;
}

coeff[0] = Princip;
coeff[1] = DPrincipDu;
coeff[2] = DPrincipDv;
coeff[3] = D2PrincipDu2;
coeff[4] = D2PrincipDuDv;
coeff[5] = D2PrincipDv2;
}

__device__ float Eta opt18
{
    return
    sign1/sqrt(repcoefE*pow(repcoefM-repPrinci*repcoefF,2)-2.0*repcoefF*(
    repcoefM-repPrinci*repcoefF)*(repcoefL-repPrinci*repcoefE)+repcoefG*pow(
    repcoefL-repPrinci*repcoefE,2));
```

```
}
__device__ float Miu opt18
{
    return
    sign1/sqrt(repcoefE*pow(repcoefN-repPrinci*repcoefG,2)-2.0*repcoefF*(
    repcoefM-repPrinci*repcoefF)*(repcoefN-repPrinci*repcoefG)+repcoefG*pow(
    repcoefM-repPrinci*repcoefF,2));
}
__device__ float DuDs opt18
{
if(abs(repcoefL-repPrinci*repcoefE)>=abs(repcoefN-repPrinci*repcoefG))
    {return repEta*(repcoefM-repPrinci*repcoefF);}
else
    {return repMiu*(repcoefN-repPrinci*repcoefG);}
}
__device__ float DvDs opt18
{
if(abs(repcoefL-repPrinci*repcoefE)>=abs(repcoefN-repPrinci*repcoefG))
    {return -repEta*(repcoefL-repPrinci*repcoefE);}
else
    {return -repMiu*(repcoefM-repPrinci*repcoefF);}
}
__device__ float DEDs opt18
{
    return repDEDu*repDuDs+repDEDv*repDvDs;
}
__device__ float DFDs opt18
{
    return repDFDu*repDuDs+repDFDv*repDvDs;
}
__device__ float DGDs opt18
{
    return repDGDu*repDuDs+repDGDv*repDvDs;
}
__device__ float DLDs opt26
{
float CoeffL[6];
repDcoefL_2;

    return CoeffL[1]*repDuDs+CoeffL[2]*repDvDs;
}
__device__ float DMDs opt26
{
float CoeffM[6];
repDcoefM_2;

    return CoeffM[1]*repDuDs+CoeffM[2]*repDvDs;
```

```
}
__device__ float DNDs opt26
{
float CoeffN[6];
repDcoefN_2;

    return CoeffN[1]*repDuDs+CoeffN[2]*repDvDs;
}
__device__ float DPrinciDs opt26
{
float PCurva[6];
repDPrinci_2;

    return PCurva[1]*repDuDs+PCurva[2]*repDvDs;
}
__device__ float DuDt opt18
{
if(abs(repcoefL-repPrinci*repcoefE)>=abs(repcoefN-repPrinci*repcoefG))
    {return (repcoefM-repPrinci*repcoefF);}
else
    {return (repcoefN-repPrinci*repcoefG);}
}
__device__ float DvDt opt18
{
if(abs(repcoefL-repPrinci*repcoefE)>=abs(repcoefN-repPrinci*repcoefG))
    {return (repcoefL-repPrinci*repcoefE);}
else
    {return (repcoefM-repPrinci*repcoefF);}
}
__device__ void DSurDs opt19
{
    tangv_C[0] =dxdu1*repDuDs+
                dxdv1*repDvDs;
    tangv_C[1] =dxdu2*repDuDs+
                dxdv2*repDvDs;
    tangv_C[2] =dxdu3*repDuDs+
                dxdv3*repDvDs;
}

__device__ float Curvecurvature opt27
{
    return sqrt(pow(repPrinci,2)+pow(GeoCur[i],2));
}
__device__ float D2EDs2 opt27
{
    return
    repD2EDu2*pow(repDuDs,2)+2*repD2EDuDv*repDuDs*repDvDs+repD2EDv2*pow(
```

```
    repDvDs,2)+repDEDu*D2uDs2[i]+repDEDv*D2vDs2[i];
}
__device__ float D2FDs2 opt27
{
    return
    repD2FDu2*pow(repDuDs,2)+2*repD2FDuDv*repDuDs*repDvDs+repD2FDv2*pow(
    repDvDs,2)+repDFDu*D2uDs2[i]+repDFDv*D2vDs2[i];
}
__device__ float D2GDs2 opt27
{
    return
    repD2GDu2*pow(repDuDs,2)+2*repD2GDuDv*repDuDs*repDvDs+repD2GDv2*pow(
    repDvDs,2)+repDGDu*D2uDs2[i]+repDGDv*D2vDs2[i];
}
__device__ float D2LDs2 opt27
{
float CoeffL[6];
repDcoefL_2;

    return
    CoeffL[3]*pow(repDuDs,2)+2*CoeffL[4]*repDuDs*repDvDs+CoeffL[5]*pow(
    repDvDs,2)+CoeffL[1]*D2uDs2[i]+CoeffL[2]*D2vDs2[i];
}

__device__ float D2MDs2 opt27
{
float CoeffM[6];
repDcoefM_2;

    return
    CoeffM[3]*pow(repDuDs,2)+2*CoeffM[4]*repDuDs*repDvDs+CoeffM[5]*pow(
    repDvDs,2)+CoeffM[1]*D2uDs2[i]+CoeffM[2]*D2vDs2[i];
}
__device__ float D2NDs2 opt27
{
float CoeffN[6];
repDcoefN_2;

    return
    CoeffN[3]*pow(repDuDs,2)+2*CoeffN[4]*repDuDs*repDvDs+CoeffN[5]*pow(
    repDvDs,2)+CoeffN[1]*D2uDs2[i]+CoeffN[2]*D2vDs2[i];
}
__device__ float D2PrinciDs2 opt27
{
float PCurva[6];
repDPrinci_2;
```

```
    return
    PCurva[3]*pow(repDuDs,2)+2*PCurva[4]*repDuDs*repDvDs+PCurva[5]*pow(
    repDvDs,2)+PCurva[1]*D2uDs2[i]+PCurva[2]*D2vDs2[i];
}

__device__ void D2SurDs2 opt28
{
    coeff[0] =
    d2xdu21*pow(repDuDs,2)+2*d2xdudv1*repDuDs*repDvDs+d2xdv21*pow(repDvDs,2)
    +dxdu1*D2uDs2[i]+dxdv1*D2vDs2[i];

    coeff[1] =
    d2xdu22*pow(repDuDs,2)+2*d2xdudv2*repDuDs*repDvDs+d2xdv22*pow(repDvDs,2)
    +dxdu2*D2uDs2[i]+dxdv2*D2vDs2[i];

    coeff[2] =
    d2xdu23*pow(repDuDs,2)+2*d2xdudv3*repDuDs*repDvDs+d2xdv23*pow(repDvDs,2)
    +dxdu3*D2uDs2[i]+dxdv3*D2vDs2[i];
}
__device__ void CurvenormaloS opt28
{
    float dtangds[3];
    repD2SurDs2;

    coeff[0] = dtangds[0]/repCurvecurvature;
    coeff[1] = dtangds[1]/repCurvecurvature;
    coeff[2] = dtangds[2]/repCurvecurvature;
}
__device__ void CurvebinormaloS opt28
{
    float tangv[3];
    float norv[3];
    repDSurDs;
    repCurvenormaloS;

    coeff[0] =tangv[1]*norv[2]-tangv[2]*norv[1];
    coeff[1] =tangv[2]*norv[0]-tangv[0]*norv[2];
    coeff[2] =tangv[0]*norv[1]-tangv[1]*norv[0];
}

__device__ void Alp2 opt28
{
    coeff[0] =
    (d3xdu31*pow(repDuDs,3)+3*d3xdu2dv1*pow(repDuDs,2)*repDvDs+3*d3xdudv21*
    repDuDs*pow(repDvDs,2)+d3xdv31*pow(repDvDs,3))+3*(d2xdu21*repDuDs*D2uDs2
    [i]+d2xdudv1*(repDvDs*D2uDs2[i]+repDuDs*D2vDs2[i])+d2xdv21*repDvDs*
    D2vDs2[i]);
```

```
    coeff[1] =
    (d3xdu32*pow(repDuDs,3)+3*d3xdu2dv2*pow(repDuDs,2)*repDvDs+3*d3xdudv22*
    repDuDs*pow(repDvDs,2)+d3xdv32*pow(repDvDs,3))+3*(d2xdu22*repDuDs*D2uDs2
    [i]+d2xdudv2*(repDvDs*D2uDs2[i]+repDuDs*D2vDs2[i])+d2xdv22*repDvDs*
    D2vDs2[i]);

    coeff[2] =
    (d3xdu33*pow(repDuDs,3)+3*d3xdu2dv3*pow(repDuDs,2)*repDvDs+3*d3xdudv23*
    repDuDs*pow(repDvDs,2)+d3xdv33*pow(repDvDs,3))+3*(d2xdu23*repDuDs*D2uDs2
    [i]+d2xdudv3*(repDvDs*D2uDs2[i]+repDuDs*D2vDs2[i])+d2xdv23*repDvDs*
    D2vDs2[i]);
}

__device__ float Beta2 opt27
{
if(abs(repcoefL-repPrinci*repcoefE)>=abs(repcoefN-repPrinci*repcoefG))
{return
    -(2.0*(repDLDs-repDPrinciDs*repcoefE-repPrinci*repDEDs)*D2uDs2[i]+2.0*(
    repDMDs-repDPrinciDs*repcoefF-repPrinci*repDFDs)*D2vDs2[i]+(repD2LDs2-
    repD2PrinciDs2*repcoefE-2*repDPrinciDs*repDEDs-repPrinci*repD2EDs2)*
    repDuDs+(repD2MDs2-repD2PrinciDs2*repcoefF-2*repDPrinciDs*repDFDs-
    repPrinci*repD2FDs2)*repDvDs);}
else
{return
    -(2.0*(repDMDs-repDPrinciDs*repcoefF-repPrinci*repDFDs)*D2uDs2[i]+2.0*(
    repDNDs-repDPrinciDs*repcoefG-repPrinci*repDGDs)*D2vDs2[i]+(repD2MDs2-
    repD2PrinciDs2*repcoefF-2*repDPrinciDs*repDFDs-repPrinci*repD2FDs2)*
    repDuDs+(repD2NDs2-repD2PrinciDs2*repcoefG-2*repDPrinciDs*repDGDs-
    repPrinci*repD2GDs2)*repDvDs);}
}

__device__ float Torsion opt27
{
    mint rc=4;
    float tangv[3];
    float norv[3];
    float binorv[3];
    float Alp2_C[3];
    repDSurDs;
    repCurvenormaloS;
    repCurvebinormaloS;
    repAlp2;
    float matrixA[4][4]={
    {repcoefE,repcoefF,-(norv[0]*dxdu1+norv[1]*dxdu2+norv[2]*dxdu3),-
    repCurvecurvature*(binorv[0]*dxdu1+binorv[1]*dxdu2+binorv[2]*dxdu3)},
```

```
{repcoefF,repcoefG,-(norv[0]*dxdv1+norv[1]*dxdv2+norv[2]*dxdv3),-
repCurvecurvature*(binorv[0]*dxdv1+binorv[1]*dxdv2+binorv[2]*dxdv3)},

{(norv[0]*dxdu1+norv[1]*dxdu2+norv[2]*dxdu3),(norv[0]*dxdv1+norv[1]*
dxdv2+norv[2]*dxdv3),-1,0},
{repDvDt,repDuDt,0,0}};

float
matrixB[4]={-(Alp2_C[0]*dxdu1+Alp2_C[1]*dxdu2+Alp2_C[2]*dxdu3)-(pow(
repCurvecurvature,2)*(tangv[0]*dxdu1+tangv[1]*dxdu2+tangv[2]*dxdu3)),-(
Alp2_C[0]*dxdv1+Alp2_C[1]*dxdv2+Alp2_C[2]*dxdv3)-(pow(repCurvecurvature,
2)*(tangv[0]*dxdv1+tangv[1]*dxdv2+tangv[2]*dxdv3)),-(Alp2_C[0]*norv[0]+
Alp2_C[1]*norv[1]+Alp2_C[2]*norv[2]),repBeta2};
float Lower[4][4];
float Upper[4][4];
float x[4];
float y[4];
float sum;

for(mint ii=0; ii<rc; ii++)
        {
            for(mint jj=0; jj<rc; jj++)
            {
                sum = 0;
                if(ii==jj)
                {
                Lower[ii][jj]=1;
                for (mint kk = 0; kk < ii; kk++)
                    {sum += Lower[ii][kk] * Upper[kk][jj];}
                Upper[ii][jj] = matrixA[ii][jj] - sum;
                }
                else if(ii < jj)
                {
                Lower[ii][jj]=0;
                for (mint kk = 0; kk < ii; kk++)
                    {sum += Lower[ii][kk] * Upper[kk][jj];}
                Upper[ii][jj] = matrixA[ii][jj] - sum;
                }
                else
                {
                Upper[ii][jj]=0;
                for (mint kk = 0; kk < jj; kk++)
                    {sum += Lower[ii][kk] * Upper[kk][jj];}
                Lower[ii][jj] =(matrixA[ii][jj] - sum)/Upper[jj][jj];
                }
            }
        }
```

```
    for (mint ii = 0; ii < rc; ii++)
    {
        sum = 0;
        for (mint jj = 0; jj < ii; jj++)
            {sum += Lower[ii][jj] * y[jj];}
        y[ii] = matrixB[ii] - sum;
    }

    for (mint ii = rc - 1; ii >= 0; ii--)
    {
        sum = 0;
        for (mint jj = ii + 1; jj < rc; jj++)
            {sum += Upper[ii][jj] * x[jj];}
        x[ii] = (y[ii] - sum)/Upper[ii][ii];
    }
return x[3];
}

__device__ float DCurvatureDs opt27
{
    mint rc=4;
    float tangv[3];
    float norv[3];
    float binorv[3];
    float Alp2_C[3];
    repDSurDs;
    repCurvenormaloS;
    repCurvebinormaloS;
    repAlp2;
    float matrixA[4][4]={
    {repcoefE,repcoefF,-(norv[0]*dxdu1+norv[1]*dxdu2+norv[2]*dxdu3),-
    repCurvecurvature*(binorv[0]*dxdu1+binorv[1]*dxdu2+binorv[2]*dxdu3)},

    {repcoefF,repcoefG,-(norv[0]*dxdv1+norv[1]*dxdv2+norv[2]*dxdv3),-
    repCurvecurvature*(binorv[0]*dxdv1+binorv[1]*dxdv2+binorv[2]*dxdv3)},

    {(norv[0]*dxdu1+norv[1]*dxdu2+norv[2]*dxdu3),(norv[0]*dxdv1+norv[1]*
    dxdv2+norv[2]*dxdv3),-1,0},
    {repDvDt,repDuDt,0,0}};

    float
    matrixB[4]={-(Alp2_C[0]*dxdu1+Alp2_C[1]*dxdu2+Alp2_C[2]*dxdu3)-(pow(
    repCurvecurvature,2)*(tangv[0]*dxdu1+tangv[1]*dxdu2+tangv[2]*dxdu3)),-(
    Alp2_C[0]*dxdv1+Alp2_C[1]*dxdv2+Alp2_C[2]*dxdv3)-(pow(repCurvecurvature,
    2)*(tangv[0]*dxdv1+tangv[1]*dxdv2+tangv[2]*dxdv3)),-(Alp2_C[0]*norv[0]+
    Alp2_C[1]*norv[1]+Alp2_C[2]*norv[2]),repBeta2};
    float Lower[4][4];
```

```
float Upper[4][4];
float x[4];
float y[4];
float sum;

for(mint ii=0; ii<rc; ii++)
        {
            for(mint jj=0; jj<rc; jj++)
            {
                sum = 0;
                if(ii==jj)
                {
                Lower[ii][jj]=1;
                for (mint kk = 0; kk < ii; kk++)
                    {sum += Lower[ii][kk] * Upper[kk][jj];}
                Upper[ii][jj] = matrixA[ii][jj] - sum;
                }
                else if(ii < jj)
                {
                Lower[ii][jj]=0;
                for (mint kk = 0; kk < ii; kk++)
                    {sum += Lower[ii][kk] * Upper[kk][jj];}
                Upper[ii][jj] = matrixA[ii][jj] - sum;
                }
                else
                {
                Upper[ii][jj]=0;
                for (mint kk = 0; kk < jj; kk++)
                    {sum += Lower[ii][kk] * Upper[kk][jj];}
                Lower[ii][jj] =(matrixA[ii][jj] - sum)/Upper[jj][jj];
                }
            }
        }
for (mint ii = 0; ii < rc; ii++)
{
    sum = 0;
    for (mint jj = 0; jj < ii; jj++)
        {sum += Lower[ii][jj] * y[jj];}
    y[ii] = matrixB[ii] - sum;
}

for (mint ii = rc - 1; ii >= 0; ii--)
{
    sum = 0;
    for (mint jj = ii + 1; jj < rc; jj++)
        {sum += Upper[ii][jj] * x[jj];}
    x[ii] = (y[ii] - sum)/Upper[ii][ii];
```

```
    }
return x[2];
}

__device__ float RK4_Init(float x1, float t1, float SSize)

{
float k1,k2,k3,k4;

k1=SSize*x1;
k2=SSize*(x1+0.5*SSize*t1);
k3=SSize*(x1+0.5*SSize*t1);
k4=SSize*(x1+SSize*t1);

return (1.0/6.0)*(k1+2*k2+2*k3+k4);
}

__device__ void RK4_LAC( float x1, float t1, float n1, float Alpha, float
    InitRC, float Lambda, float ArcLength, float initp, float stepsize,
    float& increment_x1, float& increment_t1, float& increment_n1)

{
float x_k1,x_k2,x_k3,x_k4;
float t_k1,t_k2,t_k3,t_k4;
float n_k1,n_k2,n_k3,n_k4;
float newx1,newt1,newn1;

x_k1=stepsize*t1;
t_k1=stepsize*ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
    n1);
n_k1=stepsize*ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,
    ArcLength)*t1);

newx1=x1+0.5*x_k1;
newt1=t1+0.5*t_k1;
newn1=n1+0.5*n_k1;
x_k2=stepsize*newt1;
t_k2=stepsize*ArcLength*(inicurvature_LAC(initp+0.5*stepsize,Alpha,Lambda,
    InitRC,ArcLength)*newn1);
n_k2=stepsize*ArcLength*(-inicurvature_LAC(initp+0.5*stepsize,Alpha,
    Lambda,InitRC,ArcLength)*newt1);

newx1=x1+0.5*x_k2;
newt1=t1+0.5*t_k2;
newn1=n1+0.5*n_k2;
x_k3=stepsize*newt1;
t_k3=stepsize*ArcLength*(inicurvature_LAC(initp+0.5*stepsize,Alpha,Lambda,
```

```
    InitRC,ArcLength)*newn1);
n_k3=stepsize*ArcLength*(-inicurvature_LAC(initp+0.5*stepsize,Alpha,
    Lambda,InitRC,ArcLength)*newt1);

newx1=x1+x_k3;
newt1=t1+t_k3;
newn1=n1+n_k3;
x_k4=stepsize*newt1;
t_k4=stepsize*ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*newn1);
n_k4=stepsize*ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,
    InitRC,ArcLength)*newt1);


increment_x1=(1.0/6.0)*(x_k1+2*x_k2+2*x_k3+x_k4);
increment_t1=(1.0/6.0)*(t_k1+2*t_k2+2*t_k3+t_k4);
increment_n1=(1.0/6.0)*(n_k1+2*n_k2+2*n_k3+n_k4);

}
__device__ void RK4_LASC( float x1, float t1, float n1, float b1, float Alpha,
    float Beta, float InitRC, float Omega, float Lambda, float InitRT,
    float ArcLength, float initp, float stepsize, float& increment_x1,
    float& increment_t1, float& increment_n1, float& increment_b1)

{
float x_k1,x_k2,x_k3,x_k4;
float t_k1,t_k2,t_k3,t_k4;
float n_k1,n_k2,n_k3,n_k4;
float b_k1,b_k2,b_k3,b_k4;
float newx1,newt1,newn1,newb1;

x_k1=stepsize*t1;
t_k1=stepsize*ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
    n1);
n_k1=stepsize*ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,
    ArcLength)*t1+iniTorsion_LASC(initp,Beta,Omega,InitRT,ArcLength)*b1);
b_k1=stepsize*ArcLength*(-iniTorsion_LASC(initp,Beta,Omega,InitRT,
    ArcLength)*n1);

newx1=x1+0.5*x_k1;
newt1=t1+0.5*t_k1;
newn1=n1+0.5*n_k1;
newb1=b1+0.5*b_k1;
x_k2=stepsize*newt1;
t_k2=stepsize*ArcLength*(inicurvature_LAC(initp+0.5*stepsize,Alpha,Lambda,
    InitRC,ArcLength)*newn1);
n_k2=stepsize*ArcLength*(-inicurvature_LAC(initp+0.5*stepsize,Alpha,
```

```
    Lambda,InitRC,ArcLength)*newt1+iniTorsion_LASC(initp+0.5*stepsize,Beta,
    Omega,InitRT,ArcLength)*newb1);
b_k2=stepsize*ArcLength*(-iniTorsion_LASC(initp+0.5*stepsize,Beta,Omega,
    InitRT,ArcLength)*newn1);

newx1=x1+0.5*x_k2;
newt1=t1+0.5*t_k2;
newn1=n1+0.5*n_k2;
newb1=b1+0.5*b_k2;
x_k3=stepsize*newt1;
t_k3=stepsize*ArcLength*(inicurvature_LAC(initp+0.5*stepsize,Alpha,Lambda,
    InitRC,ArcLength)*newn1);
n_k3=stepsize*ArcLength*(-inicurvature_LAC(initp+0.5*stepsize,Alpha,
    Lambda,InitRC,ArcLength)*newt1+iniTorsion_LASC(initp+0.5*stepsize,Beta,
    Omega,InitRT,ArcLength)*newb1);
b_k3=stepsize*ArcLength*(-iniTorsion_LASC(initp+0.5*stepsize,Beta,Omega,
    InitRT,ArcLength)*newn1);

newx1=x1+x_k3;
newt1=t1+t_k3;
newn1=n1+n_k3;
newb1=b1+b_k3;
x_k4=stepsize*newt1;
t_k4=stepsize*ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*newn1);
n_k4=stepsize*ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,
    InitRC,ArcLength)*newt1+iniTorsion_LASC(initp+stepsize,Beta,Omega,InitRT
    ,ArcLength)*newb1);
b_k4=stepsize*ArcLength*(-iniTorsion_LASC(initp+stepsize,Beta,Omega,
    InitRT,ArcLength)*newn1);

increment_x1=(1.0/6.0)*(x_k1+2*x_k2+2*x_k3+x_k4);
increment_t1=(1.0/6.0)*(t_k1+2*t_k2+2*t_k3+t_k4);
increment_n1=(1.0/6.0)*(n_k1+2*n_k2+2*n_k3+n_k4);
increment_b1=(1.0/6.0)*(b_k1+2*b_k2+2*b_k3+b_k4);

}

__device__ void First_Derivatives( float& dt1, float&
    dt2, float& dt3, float n1, float n2, float n3, float initp,
    float Alpha, float InitRC, float Lambda, float ArcLength)
{
dt1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n1);
dt2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n2);
dt3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n3);
}
```

```
__device__ void FS_2ndDerivatives_LAC( float& dt1, float& dt2,
     float& dt3, float& d2t1, float& d2t2, float& d2t3, float t1,
     float t2, float t3, float n1, float n2, float n3, float initp,
     float Alpha, float InitRC, float Lambda, float ArcLength)
{
float dn1, dn2, dn3;

dt1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n1);
dt2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n2);
dt3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n3);
dn1=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t1);
dn2=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t2);
dn3=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t3);

d2t1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*dn1+
     inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n1);
d2t2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
     dn2+inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n2);
d2t3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
     dn3+inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n3);

}

__device__ void FS_2ndDerivatives_LASC( float& dt1, float& dt2, float&
     dt3, float& d2t1, float& d2t2, float& d2t3, float t1, float t2,
     float t3, float n1, float n2, float n3, float b1, float b2,
     float b3, float initp, float Alpha, float Beta, float InitRC,
     float Omega, float Lambda, float InitRT, float ArcLength)
{
float dn1, dn2, dn3;

dt1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n1);
dt2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n2);
dt3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n3);
dn1=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t1+
     iniTorsion_LASC(initp,Beta,Omega,InitRT,ArcLength)*b1);
dn2=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t2
     +iniTorsion_LASC(initp,Beta,Omega,InitRT,ArcLength)*b2);
dn3=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t3
     +iniTorsion_LASC(initp,Beta,Omega,InitRT,ArcLength)*b3);

d2t1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*dn1+
     inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n1);
d2t2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
     dn2+inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n2);
d2t3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
```

```
    dn3+inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n3);
}

__device__ void FS_Derivatives_LAC( float& dt1, float& dt2, float& dt3, float&
    d2t1, float& d2t2, float& d2t3, float& d3t1, float& d3t2, float&
    d3t3, float t1, float t2, float t3, float n1, float n2, float n3,
    float initp, float Alpha, float InitRC, float Lambda, float ArcLength)
{
float dn1, dn2, dn3, d2n1, d2n2, d2n3;

dt1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n1);
dt2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n2);
dt3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n3);
dn1=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t1);
dn2=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t2);
dn3=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t3);

d2t1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*dn1+
    inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n1);
d2t2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
    dn2+inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n2);
d2t3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
    dn3+inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n3);

d2n1=-ArcLength*(inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*t1+
    inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*dt1);
d2n2=-ArcLength*(inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*t2+
    inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*dt2);
d2n3=-ArcLength*(inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*t3+
    inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*dt3);

d3t1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*d2n1+2.0*
    inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*dn1+inid2cLACdx2(initp,
    Alpha,Lambda,InitRC,ArcLength)*n1);
d3t2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
    d2n2+2.0*inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*dn2+
    inid2cLACdx2(initp,Alpha,Lambda,InitRC,ArcLength)*n2);
d3t3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
    d2n3+2.0*inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*dn3+
    inid2cLACdx2(initp,Alpha,Lambda,InitRC,ArcLength)*n3);
}

__device__ void FS_Derivatives_LASC( float& dt1, float& dt2, float& dt3,
    float& d2t1, float& d2t2, float& d2t3, float& d3t1, float& d3t2, float&
    d3t3, float t1, float t2, float t3, float n1, float n2, float n3,
    float b1, float b2, float b3, float initp, float Alpha, float Beta,
    float InitRC, float Omega, float Lambda, float InitRT, float ArcLength)
```

```
{
float dn1, dn2, dn3, d2n1, d2n2, d2n3;
float db1, db2, db3;

dt1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n1);
dt2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n2);
dt3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n3);
dn1=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t1+
    iniTorsion_LASC(initp,Beta,Omega,InitRT,ArcLength)*b1);
dn2=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t2
    +iniTorsion_LASC(initp,Beta,Omega,InitRT,ArcLength)*b2);
dn3=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t3
    +iniTorsion_LASC(initp,Beta,Omega,InitRT,ArcLength)*b3);
db1=ArcLength*(-iniTorsion_LASC(initp,Beta,Omega,InitRT,ArcLength)*n1);
db2=ArcLength*(-iniTorsion_LASC(initp,Beta,Omega,InitRT,ArcLength)*n2);
db3=ArcLength*(-iniTorsion_LASC(initp,Beta,Omega,InitRT,ArcLength)*n3);

d2t1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*dn1+
    inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n1);
d2t2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
    dn2+inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n2);
d2t3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
    dn3+inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n3);
d2n1=ArcLength*(-inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*t1-
    inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*dt1+inidtLASCdx(
    initp,Beta,Omega,InitRT,ArcLength)*b1+iniTorsion_LASC(initp,Beta,Omega,
    InitRT,ArcLength)*db1);
d2n2=ArcLength*(-inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*t2-
    inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*dt2+inidtLASCdx(
    initp,Beta,Omega,InitRT,ArcLength)*b2+iniTorsion_LASC(initp,Beta,Omega,
    InitRT,ArcLength)*db2);
d2n3=ArcLength*(-inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*t3-
    inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*dt3+inidtLASCdx(
    initp,Beta,Omega,InitRT,ArcLength)*b3+iniTorsion_LASC(initp,Beta,Omega,
    InitRT,ArcLength)*db3);

d3t1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*d2n1+2.0*
    inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*dn1+inid2cLACdx2(initp,
    Alpha,Lambda,InitRC,ArcLength)*n1);
d3t2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
    d2n2+2.0*inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*dn2+
    inid2cLACdx2(initp,Alpha,Lambda,InitRC,ArcLength)*n2);
d3t3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
    d2n3+2.0*inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*dn3+
    inid2cLACdx2(initp,Alpha,Lambda,InitRC,ArcLength)*n3);
}
```

```
__device__ void FS_LAC( float& x1, float& x2, float& x3, float& t1, float& t2,
     float& t3, float& n1, float& n2, float& n3, float Alpha, float
     InitRC, float Lambda, float ArcLength, float initp, float stepsize)
{
float increment_x1,increment_t1,increment_n1;
float increment_x2,increment_t2,increment_n2;
float increment_x3,increment_t3,increment_n3;

RK4_LAC(x1,t1,n1,Alpha,InitRC,Lambda,ArcLength,initp,stepsize,increment_x1,
     increment_t1,increment_n1);
RK4_LAC(x2,t2,n2,Alpha,InitRC,Lambda,ArcLength,initp,stepsize,
     increment_x2,increment_t2,increment_n2);
RK4_LAC(x3,t3,n3,Alpha,InitRC,Lambda,ArcLength,initp,stepsize,
     increment_x3,increment_t3,increment_n3);

x1+=increment_x1;
x2+=increment_x2;
x3+=increment_x3;
t1+=increment_t1;
t2+=increment_t2;
t3+=increment_t3;
n1+=increment_n1;
n2+=increment_n2;
n3+=increment_n3;
}

__device__ void FS_LASC( float& x1, float& x2, float& x3, float& t1, float&
     t2, float& t3, float& n1, float& n2, float& n3, float& b1, float& b2,
     float& b3, float Alpha, float Beta, float InitRC, float Omega, float
     Lambda, float InitRT, float ArcLength, float initp, float stepsize)
{
float increment_x1,increment_t1,increment_n1,increment_b1;
float increment_x2,increment_t2,increment_n2,increment_b2;
float increment_x3,increment_t3,increment_n3,increment_b3;

RK4_LASC(x1,t1,n1,b1,Alpha,Beta,InitRC,Omega,Lambda,InitRT,ArcLength,initp,
     stepsize,increment_x1,increment_t1,increment_n1,increment_b1);
RK4_LASC(x2,t2,n2,b2,Alpha,Beta,InitRC,Omega,Lambda,InitRT,ArcLength,
     initp,stepsize,increment_x2,increment_t2,increment_n2,increment_b2);
RK4_LASC(x3,t3,n3,b3,Alpha,Beta,InitRC,Omega,Lambda,InitRT,ArcLength,
     initp,stepsize,increment_x3,increment_t3,increment_n3,increment_b3);

x1+=increment_x1;
x2+=increment_x2;
x3+=increment_x3;
t1+=increment_t1;
t2+=increment_t2;
```

```
t3+=increment_t3;
n1+=increment_n1;
n2+=increment_n2;
n3+=increment_n3;
b1+=increment_b1;
b2+=increment_b2;
b3+=increment_b3;
}

__device__ void RK4_func_LoC( float pu, float pv, float u1_x1, float u1_x2,
    float u1_x3, float u1_t1, float u1_t2, float u1_t3, float u1_n1,
    float u1_n2, float u1_n3, float u2_x1, float u2_x2, float u2_x3,
    float u2_t1, float u2_t2, float u2_t3, float u2_n1, float u2_n2,
    float u2_n3, float u2_b1, float u2_b2, float u2_b3, float v1_x1,
    float v1_x2, float v1_x3, float v1_t1, float v1_t2, float v1_t3,
    float v1_n1, float v1_n2, float v1_n3, float v2_x1, float v2_x2,
    float v2_x3, float v2_t1, float v2_t2, float v2_t3, float v2_n1,
    float v2_n2, float v2_n3, float v2_b1, float v2_b2, float v2_b3, mint
    optimum, float stepsize, float& increment_u, float& increment_v)
{
float u1_dt1, u1_dt2, u1_dt3, u2_dt1, u2_dt2,
    u2_dt3, v1_dt1, v1_dt2, v1_dt3, v2_dt1, v2_dt2, v2_dt3;
float pu_k1,pu_k2,pu_k3,pu_k4;
float pv_k1,pv_k2,pv_k3,pv_k4;
float increment_u1_x1, increment_u1_x2, increment_u1_x3,
    increment_u1_t1, increment_u1_t2, increment_u1_t3,
    increment_u1_n1, increment_u1_n2, increment_u1_n3;
float increment_u2_x1, increment_u2_x2, increment_u2_x3, increment_u2_t1,
    increment_u2_t2, increment_u2_t3, increment_u2_n1, increment_u2_n2,
    increment_u2_n3, increment_u2_b1, increment_u2_b2, increment_u2_b3;
float increment_v1_x1, increment_v1_x2, increment_v1_x3,
    increment_v1_t1, increment_v1_t2, increment_v1_t3,
    increment_v1_n1, increment_v1_n2, increment_v1_n3;
float increment_v2_x1, increment_v2_x2, increment_v2_x3, increment_v2_t1,
    increment_v2_t2, increment_v2_t3, increment_v2_n1, increment_v2_n2,
    increment_v2_n3, increment_v2_b1, increment_v2_b2, increment_v2_b3;
float new_pu, new_pv;
float
    new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,new_u1_t2,new_u1_t3,new_u1_dt1,
    new_u1_dt2,new_u1_dt3,new_u1_n1,new_u1_n2,new_u1_n3;
float
    new_u2_x1,new_u2_x2,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,
    new_u2_dt2,new_u2_dt3,new_u2_n1,new_u2_n2,new_u2_n3;
float
    new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,new_v1_dt1,
    new_v1_dt2,new_v1_dt3,new_v1_n1,new_v1_n2,new_v1_n3;
float
```

```
        new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,new_v2_t2,new_v2_t3,new_v2_dt1,
        new_v2_dt2,new_v2_dt3,new_v2_n1,new_v2_n2,new_v2_n3;


    First_Derivatives(u1_dt1,u1_dt2,u1_dt3,u1_n1,u1_n2,u1_n3,pu,AlphaLAC,InitRCLAC,
        Lambda_Max_LAC,ArcLength_Max_LAC);
    First_Derivatives(u2_dt1,u2_dt2,u2_dt3,u2_n1,u2_n2,u2_n3,pu,AlphaLAC,
        InitRCLAC,Lambda_Max_LASC,ArcLength_Max_LASC);
    First_Derivatives(v1_dt1,v1_dt2,v1_dt3,v1_n1,v1_n2,v1_n3,pv,AlphaLAC,
        InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC);
    First_Derivatives(v2_dt1,v2_dt2,v2_dt3,v2_n1,v2_n2,v2_n3,pv,AlphaLAC,
        InitRCLAC,Lambda_Min_LASC,ArcLength_Min_LASC);

    pu_k1=stepsize*iniDuDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
        u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
        v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
        v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum);
    pv_k1=stepsize*iniDvDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,
        u1_dt2,u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,
        v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,
        v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum);

    RK4_LAC(u1_x1,u1_t1,u1_n1,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC,
        pu,0.5*pu_k1,increment_u1_x1,increment_u1_t1,increment_u1_n1);
    RK4_LAC(u1_x2,u1_t2,u1_n2,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
        ArcLength_Max_LAC,pu,0.5*pu_k1,increment_u1_x2,increment_u1_t2,
        increment_u1_n2);
    RK4_LAC(u1_x3,u1_t3,u1_n3,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
        ArcLength_Max_LAC,pu,0.5*pu_k1,increment_u1_x3,increment_u1_t3,
        increment_u1_n3);
    RK4_LASC(u2_x1,u2_t1,u2_n1,u2_b1,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
        Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,0.5*pu_k1,
        increment_u2_x1,increment_u2_t1,increment_u2_n1,increment_u2_b1);
    RK4_LASC(u2_x2,u2_t2,u2_n2,u2_b2,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
        Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,0.5*pu_k1,
        increment_u2_x2,increment_u2_t2,increment_u2_n2,increment_u2_b2);
    RK4_LASC(u2_x3,u2_t3,u2_n3,u2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
        Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,0.5*pu_k1,
        increment_u2_x3,increment_u2_t3,increment_u2_n3,increment_u2_b3);
    RK4_LAC(v1_x1,v1_t1,v1_n1,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
        ArcLength_Min_LAC,pv,0.5*pv_k1,increment_v1_x1,increment_v1_t1,
        increment_v1_n1);
    RK4_LAC(v1_x2,v1_t2,v1_n2,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
        ArcLength_Min_LAC,pv,0.5*pv_k1,increment_v1_x2,increment_v1_t2,
        increment_v1_n2);
    RK4_LAC(v1_x3,v1_t3,v1_n3,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
        ArcLength_Min_LAC,pv,0.5*pv_k1,increment_v1_x3,increment_v1_t3,
```

```
        increment_v1_n3);
    RK4_LASC(v2_x1,v2_t1,v2_n1,v2_b1,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
        Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,0.5*pv_k1,
        increment_v2_x1,increment_v2_t1,increment_v2_n1,increment_v2_b1);
    RK4_LASC(v2_x2,v2_t2,v2_n2,v2_b2,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
        Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,0.5*pv_k1,
        increment_v2_x2,increment_v2_t2,increment_v2_n2,increment_v2_b2);
    RK4_LASC(v2_x3,v2_t3,v2_n3,v2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
        Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,0.5*pv_k1,
        increment_v2_x3,increment_v2_t3,increment_v2_n3,increment_v2_b3);

    new_pu=pu+0.5*pu_k1;
    new_pv=pv+0.5*pv_k1;
    new_u1_x1=u1_x1+increment_u1_x1;
    new_u1_x2=u1_x2+increment_u1_x2;
    new_u1_x3=u1_x3+increment_u1_x3;
    new_u1_t1=u1_t1+increment_u1_t1;
    new_u1_t2=u1_t2+increment_u1_t2;
    new_u1_t3=u1_t3+increment_u1_t3;
    new_u1_n1=u1_n1+increment_u1_n1;
    new_u1_n2=u1_n2+increment_u1_n2;
    new_u1_n3=u1_n3+increment_u1_n3;
    new_u2_x1=u2_x1+increment_u2_x1;
    new_u2_x2=u2_x2+increment_u2_x2;
    new_u2_x3=u2_x3+increment_u2_x3;
    new_u2_t1=u2_t1+increment_u2_t1;
    new_u2_t2=u2_t2+increment_u2_t2;
    new_u2_t3=u2_t3+increment_u2_t3;
    new_u2_n1=u2_n1+increment_u2_n1;
    new_u2_n2=u2_n2+increment_u2_n2;
    new_u2_n3=u2_n3+increment_u2_n3;
    new_v1_x1=v1_x1+increment_v1_x1;
    new_v1_x2=v1_x2+increment_v1_x2;
    new_v1_x3=v1_x3+increment_v1_x3;
    new_v1_t1=v1_t1+increment_v1_t1;
    new_v1_t2=v1_t2+increment_v1_t2;
    new_v1_t3=v1_t3+increment_v1_t3;
    new_v1_n1=v1_n1+increment_v1_n1;
    new_v1_n2=v1_n2+increment_v1_n2;
    new_v1_n3=v1_n3+increment_v1_n3;
    new_v2_x1=v2_x1+increment_v2_x1;
    new_v2_x2=v2_x2+increment_v2_x2;
    new_v2_x3=v2_x3+increment_v2_x3;
    new_v2_t1=v2_t1+increment_v2_t1;
    new_v2_t2=v2_t2+increment_v2_t2;
    new_v2_t3=v2_t3+increment_v2_t3;
    new_v2_n1=v2_n1+increment_v2_n1;
```

```
new_v2_n2=v2_n2+increment_v2_n2;
new_v2_n3=v2_n3+increment_v2_n3;

First_Derivatives(new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u1_n1,new_u1_n2,new_u1
     _n3,new_pu,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC);
First_Derivatives(new_u2_dt1,new_u2_dt2,new_u2_dt3,new_u2_n1,new_u2_n2,
     new_u2_n3,new_pu,AlphaLAC,InitRCLAC,Lambda_Max_LASC,ArcLength_Max_LASC);
First_Derivatives(new_v1_dt1,new_v1_dt2,new_v1_dt3,new_v1_n1,new_v1_n2,
     new_v1_n3,new_pv,AlphaLAC,InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC);
First_Derivatives(new_v2_dt1,new_v2_dt2,new_v2_dt3,new_v2_n1,new_v2_n2,
     new_v2_n3,new_pv,AlphaLAC,InitRCLAC,Lambda_Min_LASC,ArcLength_Min_LASC);


pu_k2=stepsize*iniDuDs(new_pu,new_pv,new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,
     new_u1_t2,new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u2_x1,new_u2_x2
     ,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2,new_u2
     _dt3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,new_v1
     _dt1,new_v1_dt2,new_v1_dt3,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,
     new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,optimum);
pv_k2=stepsize*iniDvDs(new_pu,new_pv,new_u1_x1,new_u1_x2,new_u1_x3,
     new_u1_t1,new_u1_t2,new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u2_x1
     ,new_u2_x2,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2
     ,new_u2_dt3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,
     new_v1_dt1,new_v1_dt2,new_v1_dt3,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1
     ,new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,optimum);

RK4_LAC(u1_x1,u1_t1,u1_n1,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC,
     pu,0.5*pu_k2,increment_u1_x1,increment_u1_t1,increment_u1_n1);
RK4_LAC(u1_x2,u1_t2,u1_n2,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
     ArcLength_Max_LAC,pu,0.5*pu_k2,increment_u1_x2,increment_u1_t2,
     increment_u1_n2);
RK4_LAC(u1_x3,u1_t3,u1_n3,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
     ArcLength_Max_LAC,pu,0.5*pu_k2,increment_u1_x3,increment_u1_t3,
     increment_u1_n3);
RK4_LASC(u2_x1,u2_t1,u2_n1,u2_b1,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
     Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,0.5*pu_k2,
     increment_u2_x1,increment_u2_t1,increment_u2_n1,increment_u2_b1);
RK4_LASC(u2_x2,u2_t2,u2_n2,u2_b2,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
     Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,0.5*pu_k2,
     increment_u2_x2,increment_u2_t2,increment_u2_n2,increment_u2_b2);
RK4_LASC(u2_x3,u2_t3,u2_n3,u2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
     Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,0.5*pu_k2,
     increment_u2_x3,increment_u2_t3,increment_u2_n3,increment_u2_b3);
RK4_LAC(v1_x1,v1_t1,v1_n1,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
     ArcLength_Min_LAC,pv,0.5*pv_k2,increment_v1_x1,increment_v1_t1,
     increment_v1_n1);
RK4_LAC(v1_x2,v1_t2,v1_n2,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
```

```
    ArcLength_Min_LAC,pv,0.5*pv_k2,increment_v1_x2,increment_v1_t2,
    increment_v1_n2);
RK4_LAC(v1_x3,v1_t3,v1_n3,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
    ArcLength_Min_LAC,pv,0.5*pv_k2,increment_v1_x3,increment_v1_t3,
    increment_v1_n3);
RK4_LASC(v2_x1,v2_t1,v2_n1,v2_b1,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,0.5*pv_k2,
    increment_v2_x1,increment_v2_t1,increment_v2_n1,increment_v2_b1);
RK4_LASC(v2_x2,v2_t2,v2_n2,v2_b2,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,0.5*pv_k2,
    increment_v2_x2,increment_v2_t2,increment_v2_n2,increment_v2_b2);
RK4_LASC(v2_x3,v2_t3,v2_n3,v2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,0.5*pv_k2,
    increment_v2_x3,increment_v2_t3,increment_v2_n3,increment_v2_b3);

new_pu=pu+0.5*pu_k2;
new_pv=pv+0.5*pv_k2;
new_u1_x1=u1_x1+increment_u1_x1;
new_u1_x2=u1_x2+increment_u1_x2;
new_u1_x3=u1_x3+increment_u1_x3;
new_u1_t1=u1_t1+increment_u1_t1;
new_u1_t2=u1_t2+increment_u1_t2;
new_u1_t3=u1_t3+increment_u1_t3;
new_u1_n1=u1_n1+increment_u1_n1;
new_u1_n2=u1_n2+increment_u1_n2;
new_u1_n3=u1_n3+increment_u1_n3;
new_u2_x1=u2_x1+increment_u2_x1;
new_u2_x2=u2_x2+increment_u2_x2;
new_u2_x3=u2_x3+increment_u2_x3;
new_u2_t1=u2_t1+increment_u2_t1;
new_u2_t2=u2_t2+increment_u2_t2;
new_u2_t3=u2_t3+increment_u2_t3;
new_u2_n1=u2_n1+increment_u2_n1;
new_u2_n2=u2_n2+increment_u2_n2;
new_u2_n3=u2_n3+increment_u2_n3;
new_v1_x1=v1_x1+increment_v1_x1;
new_v1_x2=v1_x2+increment_v1_x2;
new_v1_x3=v1_x3+increment_v1_x3;
new_v1_t1=v1_t1+increment_v1_t1;
new_v1_t2=v1_t2+increment_v1_t2;
new_v1_t3=v1_t3+increment_v1_t3;
new_v1_n1=v1_n1+increment_v1_n1;
new_v1_n2=v1_n2+increment_v1_n2;
new_v1_n3=v1_n3+increment_v1_n3;
new_v2_x1=v2_x1+increment_v2_x1;
new_v2_x2=v2_x2+increment_v2_x2;
new_v2_x3=v2_x3+increment_v2_x3;
```

```
new_v2_t1=v2_t1+increment_v2_t1;
new_v2_t2=v2_t2+increment_v2_t2;
new_v2_t3=v2_t3+increment_v2_t3;
new_v2_n1=v2_n1+increment_v2_n1;
new_v2_n2=v2_n2+increment_v2_n2;
new_v2_n3=v2_n3+increment_v2_n3;

First_Derivatives(new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u1_n1,new_u1_n2,new_u1
    _n3,new_pu,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC);
First_Derivatives(new_u2_dt1,new_u2_dt2,new_u2_dt3,new_u2_n1,new_u2_n2,
    new_u2_n3,new_pu,AlphaLAC,InitRCLAC,Lambda_Max_LASC,ArcLength_Max_LASC);
First_Derivatives(new_v1_dt1,new_v1_dt2,new_v1_dt3,new_v1_n1,new_v1_n2,
    new_v1_n3,new_pv,AlphaLAC,InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC);
First_Derivatives(new_v2_dt1,new_v2_dt2,new_v2_dt3,new_v2_n1,new_v2_n2,
    new_v2_n3,new_pv,AlphaLAC,InitRCLAC,Lambda_Min_LASC,ArcLength_Min_LASC);


pu_k3=stepsize*iniDuDs(new_pu,new_pv,new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,
    new_u1_t2,new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u2_x1,new_u2_x2
    ,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2,new_u2
    _dt3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,new_v1
    _dt1,new_v1_dt2,new_v1_dt3,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,
    new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,optimum);
pv_k3=stepsize*iniDvDs(new_pu,new_pv,new_u1_x1,new_u1_x2,new_u1_x3,
    new_u1_t1,new_u1_t2,new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u2_x1
    ,new_u2_x2,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2
    ,new_u2_dt3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,
    new_v1_dt1,new_v1_dt2,new_v1_dt3,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1
    ,new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,optimum);

RK4_LAC(u1_x1,u1_t1,u1_n1,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC,
    pu,pu_k3,increment_u1_x1,increment_u1_t1,increment_u1_n1);
RK4_LAC(u1_x2,u1_t2,u1_n2,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
    ArcLength_Max_LAC,pu,pu_k3,increment_u1_x2,increment_u1_t2,increment_u1
    _n2);
RK4_LAC(u1_x3,u1_t3,u1_n3,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
    ArcLength_Max_LAC,pu,pu_k3,increment_u1_x3,increment_u1_t3,increment_u1
    _n3);
RK4_LASC(u2_x1,u2_t1,u2_n1,u2_b1,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,pu_k3,increment_u2
    _x1,increment_u2_t1,increment_u2_n1,increment_u2_b1);
RK4_LASC(u2_x2,u2_t2,u2_n2,u2_b2,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,pu_k3,increment_u2
    _x2,increment_u2_t2,increment_u2_n2,increment_u2_b2);
RK4_LASC(u2_x3,u2_t3,u2_n3,u2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,pu_k3,increment_u2
    _x3,increment_u2_t3,increment_u2_n3,increment_u2_b3);
```

```
RK4_LAC(v1_x1,v1_t1,v1_n1,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
     ArcLength_Min_LAC,pv,pv_k3,increment_v1_x1,increment_v1_t1,increment_v1
     _n1);
RK4_LAC(v1_x2,v1_t2,v1_n2,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
     ArcLength_Min_LAC,pv,pv_k3,increment_v1_x2,increment_v1_t2,increment_v1
     _n2);
RK4_LAC(v1_x3,v1_t3,v1_n3,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
     ArcLength_Min_LAC,pv,pv_k3,increment_v1_x3,increment_v1_t3,increment_v1
     _n3);
RK4_LASC(v2_x1,v2_t1,v2_n1,v2_b1,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
     Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,pv_k3,increment_v2
     _x1,increment_v2_t1,increment_v2_n1,increment_v2_b1);
RK4_LASC(v2_x2,v2_t2,v2_n2,v2_b2,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
     Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,pv_k3,increment_v2
     _x2,increment_v2_t2,increment_v2_n2,increment_v2_b2);
RK4_LASC(v2_x3,v2_t3,v2_n3,v2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
     Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,pv_k3,increment_v2
     _x3,increment_v2_t3,increment_v2_n3,increment_v2_b3);

new_pu=pu+pu_k3;
new_pv=pv+pv_k3;
new_u1_x1=u1_x1+increment_u1_x1;
new_u1_x2=u1_x2+increment_u1_x2;
new_u1_x3=u1_x3+increment_u1_x3;
new_u1_t1=u1_t1+increment_u1_t1;
new_u1_t2=u1_t2+increment_u1_t2;
new_u1_t3=u1_t3+increment_u1_t3;
new_u1_n1=u1_n1+increment_u1_n1;
new_u1_n2=u1_n2+increment_u1_n2;
new_u1_n3=u1_n3+increment_u1_n3;
new_u2_x1=u2_x1+increment_u2_x1;
new_u2_x2=u2_x2+increment_u2_x2;
new_u2_x3=u2_x3+increment_u2_x3;
new_u2_t1=u2_t1+increment_u2_t1;
new_u2_t2=u2_t2+increment_u2_t2;
new_u2_t3=u2_t3+increment_u2_t3;
new_u2_n1=u2_n1+increment_u2_n1;
new_u2_n2=u2_n2+increment_u2_n2;
new_u2_n3=u2_n3+increment_u2_n3;
new_v1_x1=v1_x1+increment_v1_x1;
new_v1_x2=v1_x2+increment_v1_x2;
new_v1_x3=v1_x3+increment_v1_x3;
new_v1_t1=v1_t1+increment_v1_t1;
new_v1_t2=v1_t2+increment_v1_t2;
new_v1_t3=v1_t3+increment_v1_t3;
new_v1_n1=v1_n1+increment_v1_n1;
new_v1_n2=v1_n2+increment_v1_n2;
```

```
new_v1_n3=v1_n3+increment_v1_n3;
new_v2_x1=v2_x1+increment_v2_x1;
new_v2_x2=v2_x2+increment_v2_x2;
new_v2_x3=v2_x3+increment_v2_x3;
new_v2_t1=v2_t1+increment_v2_t1;
new_v2_t2=v2_t2+increment_v2_t2;
new_v2_t3=v2_t3+increment_v2_t3;
new_v2_n1=v2_n1+increment_v2_n1;
new_v2_n2=v2_n2+increment_v2_n2;
new_v2_n3=v2_n3+increment_v2_n3;

First_Derivatives(new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u1_n1,new_u1_n2,new_u1
    _n3,new_pu,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC);
First_Derivatives(new_u2_dt1,new_u2_dt2,new_u2_dt3,new_u2_n1,new_u2_n2,
    new_u2_n3,new_pu,AlphaLAC,InitRCLAC,Lambda_Max_LASC,ArcLength_Max_LASC);
First_Derivatives(new_v1_dt1,new_v1_dt2,new_v1_dt3,new_v1_n1,new_v1_n2,
    new_v1_n3,new_pv,AlphaLAC,InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC);
First_Derivatives(new_v2_dt1,new_v2_dt2,new_v2_dt3,new_v2_n1,new_v2_n2,
    new_v2_n3,new_pv,AlphaLAC,InitRCLAC,Lambda_Min_LASC,ArcLength_Min_LASC);


pu_k4=stepsize*iniDuDs(new_pu,new_pv,new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,
    new_u1_t2,new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u2_x1,new_u2_x2
    ,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2,new_u2
    _dt3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,new_v1
    _dt1,new_v1_dt2,new_v1_dt3,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,
    new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,optimum);
pv_k4=stepsize*iniDvDs(new_pu,new_pv,new_u1_x1,new_u1_x2,new_u1_x3,
    new_u1_t1,new_u1_t2,new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u2_x1
    ,new_u2_x2,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2
    ,new_u2_dt3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,
    new_v1_dt1,new_v1_dt2,new_v1_dt3,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1
    ,new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,optimum);

increment_u=(1.0/6.0)*(pu_k1+2*pu_k2+2*pu_k3+pu_k4);
increment_v=(1.0/6.0)*(pv_k1+2*pv_k2+2*pv_k3+pv_k4);

}



__device__ void AllDetails(mint k,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u1_d2t1[],float
    u1_d2t2[],float u1_d2t3[],float u1_d3t1[],float u1_d3t2[],float
    u1_d3t3[],float u2_x1[],float u2_x2[],float u2_x3[],float
    u2_t1[],float u2_t2[],float u2_t3[],float u2_dt1[],float
```

```
    u2_dt2[],float u2_dt3[],float u2_d2t1[],float u2_d2t2[],float
    u2_d2t3[],float u2_d3t1[],float u2_d3t2[],float u2_d3t3[],float
    v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float
    v1_t3[],float v1_dt1[],float v1_dt2[],float v1_dt3[],float
    v1_d2t1[],float v1_d2t2[],float v1_d2t3[],float v1_d3t1[],float
    v1_d3t2[],float v1_d3t3[],float v2_x1[],float v2_x2[],float
    v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float
    v2_dt1[],float v2_dt2[],float v2_dt3[],float v2_d2t1[],float
    v2_d2t2[],float v2_d2t3[],float v2_d3t1[],float v2_d3t2[],float
    v2_d3t3[],float D2uDs2[],float D2vDs2[],float GeoCur[],mint optimum)
{
float D_u1_x1=Point00x, D_u1_x2=Point00y, D_u1_x3=Point00z;
float D_u2_x1=Point30x, D_u2_x2=Point30y, D_u2_x3=Point30z;
float D_v1_x1=Point00x, D_v1_x2=Point00y, D_v1_x3=Point00z;
float D_v2_x1=Point03x, D_v2_x2=Point03y, D_v2_x3=Point03z;

float D_u1_t1=-1*ScaleToOrigin_Max_LAC*ArcLength_Max_LAC,
    D_u1_t2=0, D_u1_t3=0;
float D_u1_n1=0, D_u1_n2=0, D_u1_n3=1*ScaleToOrigin_Max_LAC*ArcLength_Max_LAC;
float D_v1_t1=0,
    D_v1_t2=-1*ScaleToOrigin_Min_LAC*ArcLength_Min_LAC, D_v1_t3=0;
float D_v1_n1=0, D_v1_n2=0,
    D_v1_n3=-1*ScaleToOrigin_Min_LAC*ArcLength_Min_LAC;

float D_u2_t1=-1*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC,
    D_u2_t2=0, D_u2_t3=0;
float D_u2_n1=0,
    D_u2_n2=-0.5930588126117428*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC,
    D_u2_n3=0.8051591425200051*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC;
float D_u2_b1=0,
    D_u2_b2=0.8051591425200051*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC,
    D_u2_b3=0.5930588126117428*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC;

float D_v2_t1=0,
    D_v2_t2=-1*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC, D_v2_t3=0;
float D_v2_n1=-0.5930588126117428*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC,
    D_v2_n2=0,
    D_v2_n3=-0.8051591425200051*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC;
float D_v2_b1=0.8051591425200051*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC,
    D_v2_b2=0,
    D_v2_b3=-0.5930588126117428*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC;

float D_pu=0.0, D_pv=0.0;
float SSize_u, SSize_v, ratio;

float D_u1_dt1, D_u1_dt2, D_u1_dt3;
float D_u1_d2t1, D_u1_d2t2, D_u1_d2t3;
```

```
float D_u1_d3t1, D_u1_d3t2, D_u1_d3t3;
float D_u2_dt1, D_u2_dt2, D_u2_dt3;
float D_u2_d2t1, D_u2_d2t2, D_u2_d2t3;
float D_u2_d3t1, D_u2_d3t2, D_u2_d3t3;

float D_v1_dt1, D_v1_dt2, D_v1_dt3;
float D_v1_d2t1, D_v1_d2t2, D_v1_d2t3;
float D_v1_d3t1, D_v1_d3t2, D_v1_d3t3;
float D_v2_dt1, D_v2_dt2, D_v2_dt3;
float D_v2_d2t1, D_v2_d2t2, D_v2_d2t3;
float D_v2_d3t1, D_v2_d3t2, D_v2_d3t3;
float coef[3];


if(optimum==min)
{
    for(mint i=1;i<k+1;i++)
    {

    FS_LAC(D_u1_x1,D_u1_x2,D_u1_x3,D_u1_t1,D_u1_t2,D_u1_t3,D_u1_n1,D_u1_n2,
    D_u1_n3,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC,D_pu,h);
    //increase u by 0.001, then return the details

    FS_LASC(D_u2_x1,D_u2_x2,D_u2_x3,D_u2_t1,D_u2_t2,D_u2_t3,D_u2_n1,D_u2_n2,
    D_u2_n3,D_u2_b1,D_u2_b2,D_u2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,D_pu,h);
    //increase u by 0.001, then return the details

    D_pu+=h;
    }
}
else if(optimum==max)
{
    for(mint i=1;i<k+1;i++)
    {

    FS_LAC(D_v1_x1,D_v1_x2,D_v1_x3,D_v1_t1,D_v1_t2,D_v1_t3,D_v1_n1,D_v1_n2,
    D_v1_n3,AlphaLAC,InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC,D_pv,h);
    //increase v by 0.001, then return the details

    FS_LASC(D_v2_x1,D_v2_x2,D_v2_x3,D_v2_t1,D_v2_t2,D_v2_t3,D_v2_n1,D_v2_n2,
    D_v2_n3,D_v2_b1,D_v2_b2,D_v2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,D_pv,h);
    //increase v by 0.001, then return the details

    D_pv+=h;
    }
```

```
}

FS_Derivatives_LAC(D_u1_dt1,D_u1_dt2,D_u1_dt3,D_u1_d2t1,D_u1_d2t2,D_u1_d2t3,
    D_u1_d3t1,D_u1_d3t2,D_u1_d3t3,D_u1_t1,D_u1_t2,D_u1_t3,D_u1_n1,D_u1_n2,
    D_u1_n3,D_pu,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC);
FS_Derivatives_LASC(D_u2_dt1,D_u2_dt2,D_u2_dt3,D_u2_d2t1,D_u2_d2t2,D_u2
    _d2t3,D_u2_d3t1,D_u2_d3t2,D_u2_d3t3,D_u2_t1,D_u2_t2,D_u2_t3,D_u2_n1,D_u2
    _n2,D_u2_n3,D_u2_b1,D_u2_b2,D_u2_b3,D_pu,AlphaLAC,BetaLASC,InitRCLAC,
    OmegaLASC,Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC);
FS_Derivatives_LAC(D_v1_dt1,D_v1_dt2,D_v1_dt3,D_v1_d2t1,D_v1_d2t2,D_v1
    _d2t3,D_v1_d3t1,D_v1_d3t2,D_v1_d3t3,D_v1_t1,D_v1_t2,D_v1_t3,D_v1_n1,D_v1
    _n2,D_v1_n3,D_pv,AlphaLAC,InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC);
FS_Derivatives_LASC(D_v2_dt1,D_v2_dt2,D_v2_dt3,D_v2_d2t1,D_v2_d2t2,D_v2
    _d2t3,D_v2_d3t1,D_v2_d3t2,D_v2_d3t3,D_v2_t1,D_v2_t2,D_v2_t3,D_v2_n1,D_v2
    _n2,D_v2_n3,D_v2_b1,D_v2_b2,D_v2_b3,D_pv,AlphaLAC,BetaLASC,InitRCLAC,
    OmegaLASC,Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC);

LUDecomposition3x3(D_pu,D_pv,D_u1_x1,D_u1_x2,D_u1_x3,D_u1_t1,D_u1_t2,D_u1_t3,
    D_u1_dt1,D_u1_dt2,D_u1_dt3,D_u1_d2t1,D_u1_d2t2,D_u1_d2t3,D_u2_x1,D_u2_x2
    ,D_u2_x3,D_u2_t1,D_u2_t2,D_u2_t3,D_u2_dt1,D_u2_dt2,D_u2_dt3,D_u2_d2t1,
    D_u2_d2t2,D_u2_d2t3,D_v1_x1,D_v1_x2,D_v1_x3,D_v1_t1,D_v1_t2,D_v1_t3,D_v1
    _dt1,D_v1_dt2,D_v1_dt3,D_v1_d2t1,D_v1_d2t2,D_v1_d2t3,D_v2_x1,D_v2_x2,
    D_v2_x3,D_v2_t1,D_v2_t2,D_v2_t3,D_v2_dt1,D_v2_dt2,D_v2_dt3,D_v2_d2t1,
    D_v2_d2t2,D_v2_d2t3,optimum,coef);

pu[0]=D_pu;
pv[0]=D_pv;
u1_x1[0]=D_u1_x1;
u1_x2[0]=D_u1_x2;
u1_x3[0]=D_u1_x3;
u1_t1[0]=D_u1_t1;
u1_t2[0]=D_u1_t2;
u1_t3[0]=D_u1_t3;
u1_dt1[0]=D_u1_dt1;
u1_dt2[0]=D_u1_dt2;
u1_dt3[0]=D_u1_dt3;
u1_d2t1[0]=D_u1_d2t1;
u1_d2t2[0]=D_u1_d2t2;
u1_d2t3[0]=D_u1_d2t3;
u1_d3t1[0]=D_u1_d3t1;
u1_d3t2[0]=D_u1_d3t2;
u1_d3t3[0]=D_u1_d3t3;
u2_x1[0]=D_u2_x1;
u2_x2[0]=D_u2_x2;
u2_x3[0]=D_u2_x3;
u2_t1[0]=D_u2_t1;
u2_t2[0]=D_u2_t2;
```

```
u2_t3[0]=D_u2_t3;
u2_dt1[0]=D_u2_dt1;
u2_dt2[0]=D_u2_dt2;
u2_dt3[0]=D_u2_dt3;
u2_d2t1[0]=D_u2_d2t1;
u2_d2t2[0]=D_u2_d2t2;
u2_d2t3[0]=D_u2_d2t3;
u2_d3t1[0]=D_u2_d3t1;
u2_d3t2[0]=D_u2_d3t2;
u2_d3t3[0]=D_u2_d3t3;

v1_x1[0]=D_v1_x1;
v1_x2[0]=D_v1_x2;
v1_x3[0]=D_v1_x3;
v1_t1[0]=D_v1_t1;
v1_t2[0]=D_v1_t2;
v1_t3[0]=D_v1_t3;
v1_dt1[0]=D_v1_dt1;
v1_dt2[0]=D_v1_dt2;
v1_dt3[0]=D_v1_dt3;
v1_d2t1[0]=D_v1_d2t1;
v1_d2t2[0]=D_v1_d2t2;
v1_d2t3[0]=D_v1_d2t3;
v1_d3t1[0]=D_v1_d3t1;
v1_d3t2[0]=D_v1_d3t2;
v1_d3t3[0]=D_v1_d3t3;
v2_x1[0]=D_v2_x1;
v2_x2[0]=D_v2_x2;
v2_x3[0]=D_v2_x3;
v2_t1[0]=D_v2_t1;
v2_t2[0]=D_v2_t2;
v2_t3[0]=D_v2_t3;
v2_dt1[0]=D_v2_dt1;
v2_dt2[0]=D_v2_dt2;
v2_dt3[0]=D_v2_dt3;
v2_d2t1[0]=D_v2_d2t1;
v2_d2t2[0]=D_v2_d2t2;
v2_d2t3[0]=D_v2_d2t3;
v2_d3t1[0]=D_v2_d3t1;
v2_d3t2[0]=D_v2_d3t2;
v2_d3t3[0]=D_v2_d3t3;

D2uDs2[0]=coef[0];
D2vDs2[0]=coef[1];
GeoCur[0]=coef[2];
```

```
for(mint j=1;j<n;j++)
{

    RK4_func_LoC(D_pu,D_pv,D_u1_x1,D_u1_x2,D_u1_x3,D_u1_t1,D_u1_t2,D_u1_t3,
    D_u1_n1,D_u1_n2,D_u1_n3,D_u2_x1,D_u2_x2,D_u2_x3,D_u2_t1,D_u2_t2,D_u2_t3,
    D_u2_n1,D_u2_n2,D_u2_n3,D_u2_b1,D_u2_b2,D_u2_b3,D_v1_x1,D_v1_x2,D_v1_x3,
    D_v1_t1,D_v1_t2,D_v1_t3,D_v1_n1,D_v1_n2,D_v1_n3,D_v2_x1,D_v2_x2,D_v2_x3,
    D_v2_t1,D_v2_t2,D_v2_t3,D_v2_n1,D_v2_n2,D_v2_n3,D_v2_b1,D_v2_b2,D_v2_b3,
    optimum,h,SSize_u,SSize_v);
    //return increment of u and v from duds and dvds

    if((D_pu+SSize_u) > 1.0)
        {
        if(D_pu==1.0)
            {
            SSize_u=0.0;
            SSize_v=0.0;
            }
        else
            {
            ratio=(1.0-D_pu)/SSize_u;
            SSize_u*=ratio;
            SSize_v*=ratio;
            }
        }

    if((D_pv+SSize_v) > 1.0)
        {
        if(D_pv==1.0)
            {
            SSize_u=0.0;
            SSize_v=0.0;
            }
        else
            {
            ratio=(1.0-D_pv)/SSize_v;
            SSize_u*=ratio;
            SSize_v*=ratio;
            }
        }


    FS_LAC(D_u1_x1,D_u1_x2,D_u1_x3,D_u1_t1,D_u1_t2,D_u1_t3,D_u1_n1,D_u1_n2,
    D_u1_n3,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC,D_pu,SSize_u
    );
    //based on the increment
    of u as stepsize of LAC, then return the details
```

```
FS_LASC(D_u2_x1,D_u2_x2,D_u2_x3,D_u2_t1,D_u2_t2,D_u2_t3,D_u2_n1,D_u2_n2,
D_u2_n3,D_u2_b1,D_u2_b2,D_u2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,D_pu,SSize_u);
//based on the incremnt
of u as stepsize of LASC, then return the details

FS_LAC(D_v1_x1,D_v1_x2,D_v1_x3,D_v1_t1,D_v1_t2,D_v1_t3,D_v1_n1,D_v1_n2,
D_v1_n3,AlphaLAC,InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC,D_pv,SSize_v
);
//based on the increment
of v as stepsize of LAC, then return the details

FS_LASC(D_v2_x1,D_v2_x2,D_v2_x3,D_v2_t1,D_v2_t2,D_v2_t3,D_v2_n1,D_v2_n2,
D_v2_n3,D_v2_b1,D_v2_b2,D_v2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,D_pv,SSize_v);
//based on the increment
of v as stepsize of LASC, then return the details

D_pu+=SSize_u;
D_pv+=SSize_v;

FS_Derivatives_LAC(D_u1_dt1,D_u1_dt2,D_u1_dt3,D_u1_d2t1,D_u1_d2t2,D_u1
_d2t3,D_u1_d3t1,D_u1_d3t2,D_u1_d3t3,D_u1_t1,D_u1_t2,D_u1_t3,D_u1_n1,D_u1
_n2,D_u1_n3,D_pu,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC);

FS_Derivatives_LASC(D_u2_dt1,D_u2_dt2,D_u2_dt3,D_u2_d2t1,D_u2_d2t2,D_u2
_d2t3,D_u2_d3t1,D_u2_d3t2,D_u2_d3t3,D_u2_t1,D_u2_t2,D_u2_t3,D_u2_n1,D_u2
_n2,D_u2_n3,D_u2_b1,D_u2_b2,D_u2_b3,D_pu,AlphaLAC,BetaLASC,InitRCLAC,
OmegaLASC,Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC);

FS_Derivatives_LAC(D_v1_dt1,D_v1_dt2,D_v1_dt3,D_v1_d2t1,D_v1_d2t2,D_v1
_d2t3,D_v1_d3t1,D_v1_d3t2,D_v1_d3t3,D_v1_t1,D_v1_t2,D_v1_t3,D_v1_n1,D_v1
_n2,D_v1_n3,D_pv,AlphaLAC,InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC);

FS_Derivatives_LASC(D_v2_dt1,D_v2_dt2,D_v2_dt3,D_v2_d2t1,D_v2_d2t2,D_v2
_d2t3,D_v2_d3t1,D_v2_d3t2,D_v2_d3t3,D_v2_t1,D_v2_t2,D_v2_t3,D_v2_n1,D_v2
_n2,D_v2_n3,D_v2_b1,D_v2_b2,D_v2_b3,D_pv,AlphaLAC,BetaLASC,InitRCLAC,
OmegaLASC,Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC);

LUDecomposition3x3(D_pu,D_pv,D_u1_x1,D_u1_x2,D_u1_x3,D_u1_t1,D_u1_t2,
D_u1_t3,D_u1_dt1,D_u1_dt2,D_u1_dt3,D_u1_d2t1,D_u1_d2t2,D_u1_d2t3,D_u2_x1
,D_u2_x2,D_u2_x3,D_u2_t1,D_u2_t2,D_u2_t3,D_u2_dt1,D_u2_dt2,D_u2_dt3,D_u2
_d2t1,D_u2_d2t2,D_u2_d2t3,D_v1_x1,D_v1_x2,D_v1_x3,D_v1_t1,D_v1_t2,D_v1
_t3,D_v1_dt1,D_v1_dt2,D_v1_dt3,D_v1_d2t1,D_v1_d2t2,D_v1_d2t3,D_v2_x1,
D_v2_x2,D_v2_x3,D_v2_t1,D_v2_t2,D_v2_t3,D_v2_dt1,D_v2_dt2,D_v2_dt3,D_v2
_d2t1,D_v2_d2t2,D_v2_d2t3,optimum,coef);
```

```
pu[j]=D_pu;
pv[j]=D_pv;
u1_x1[j]=D_u1_x1;
u1_x2[j]=D_u1_x2;
u1_x3[j]=D_u1_x3;
u1_t1[j]=D_u1_t1;
u1_t2[j]=D_u1_t2;
u1_t3[j]=D_u1_t3;
u1_dt1[j]=D_u1_dt1;
u1_dt2[j]=D_u1_dt2;
u1_dt3[j]=D_u1_dt3;
u1_d2t1[j]=D_u1_d2t1;
u1_d2t2[j]=D_u1_d2t2;
u1_d2t3[j]=D_u1_d2t3;
u1_d3t1[j]=D_u1_d3t1;
u1_d3t2[j]=D_u1_d3t2;
u1_d3t3[j]=D_u1_d3t3;
u2_x1[j]=D_u2_x1;
u2_x2[j]=D_u2_x2;
u2_x3[j]=D_u2_x3;
u2_t1[j]=D_u2_t1;
u2_t2[j]=D_u2_t2;
u2_t3[j]=D_u2_t3;
u2_dt1[j]=D_u2_dt1;
u2_dt2[j]=D_u2_dt2;
u2_dt3[j]=D_u2_dt3;
u2_d2t1[j]=D_u2_d2t1;
u2_d2t2[j]=D_u2_d2t2;
u2_d2t3[j]=D_u2_d2t3;
u2_d3t1[j]=D_u2_d3t1;
u2_d3t2[j]=D_u2_d3t2;
u2_d3t3[j]=D_u2_d3t3;

v1_x1[j]=D_v1_x1;
v1_x2[j]=D_v1_x2;
v1_x3[j]=D_v1_x3;
v1_t1[j]=D_v1_t1;
v1_t2[j]=D_v1_t2;
v1_t3[j]=D_v1_t3;
v1_dt1[j]=D_v1_dt1;
v1_dt2[j]=D_v1_dt2;
v1_dt3[j]=D_v1_dt3;
v1_d2t1[j]=D_v1_d2t1;
v1_d2t2[j]=D_v1_d2t2;
v1_d2t3[j]=D_v1_d2t3;
```

```
    v1_d3t1[j]=D_v1_d3t1;
    v1_d3t2[j]=D_v1_d3t2;
    v1_d3t3[j]=D_v1_d3t3;
    v2_x1[j]=D_v2_x1;
    v2_x2[j]=D_v2_x2;
    v2_x3[j]=D_v2_x3;
    v2_t1[j]=D_v2_t1;
    v2_t2[j]=D_v2_t2;
    v2_t3[j]=D_v2_t3;
    v2_dt1[j]=D_v2_dt1;
    v2_dt2[j]=D_v2_dt2;
    v2_dt3[j]=D_v2_dt3;
    v2_d2t1[j]=D_v2_d2t1;
    v2_d2t2[j]=D_v2_d2t2;
    v2_d2t3[j]=D_v2_d2t3;
    v2_d3t1[j]=D_v2_d3t1;
    v2_d3t2[j]=D_v2_d3t2;
    v2_d3t3[j]=D_v2_d3t3;

    D2uDs2[j]=coef[0];
    D2vDs2[j]=coef[1];
    GeoCur[j]=coef[2];

}

}

__global__ void ReturnEveryDetails( float* pu, float* pv, float* u1_x1, float*
    u1_x2, float* u1_x3, float* u1_t1, float* u1_t2, float* u1_t3,
    float* u1_dt1, float* u1_dt2, float* u1_dt3, float* u1_d2t1, float*
    u1_d2t2, float* u1_d2t3, float* u1_d3t1, float* u1_d3t2, float*
    u1_d3t3, float* u2_x1, float* u2_x2, float* u2_x3, float* u2_t1,
    float* u2_t2, float* u2_t3, float* u2_dt1, float* u2_dt2, float*
    u2_dt3, float* u2_d2t1, float* u2_d2t2, float* u2_d2t3, float*
    u2_d3t1, float* u2_d3t2, float* u2_d3t3, float* v1_x1, float*
    v1_x2, float* v1_x3, float* v1_t1, float* v1_t2, float* v1_t3,
    float* v1_dt1, float* v1_dt2, float* v1_dt3, float* v1_d2t1, float*
    v1_d2t2, float* v1_d2t3, float* v1_d3t1, float* v1_d3t2, float*
    v1_d3t3, float* v2_x1, float* v2_x2, float* v2_x3, float* v2_t1,
    float* v2_t2, float* v2_t3, float* v2_dt1, float* v2_dt2, float*
    v2_dt3, float* v2_d2t1, float* v2_d2t2, float* v2_d2t3, float*
    v2_d3t1, float* v2_d3t2, float* v2_d3t3, float* D2uDs2, float*
    D2vDs2, float* GeoCur, mint init_s, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float D_pu[n], D_pv[n];
float D_u1_x1[n], D_u1_x2[n], D_u1_x3[n], D_u1_t1[n], D_u1_t2[n],
```

```
    D_u1_t3[n], D_u1_dt1[n], D_u1_dt2[n], D_u1_dt3[n], D_u1_d2t1[n],
    D_u1_d2t2[n], D_u1_d2t3[n], D_u1_d3t1[n], D_u1_d3t2[n], D_u1_d3t3[n];
float D_u2_x1[n], D_u2_x2[n], D_u2_x3[n], D_u2_t1[n], D_u2_t2[n],
    D_u2_t3[n], D_u2_dt1[n], D_u2_dt2[n], D_u2_dt3[n], D_u2_d2t1[n],
    D_u2_d2t2[n], D_u2_d2t3[n], D_u2_d3t1[n], D_u2_d3t2[n], D_u2_d3t3[n];
float D_v1_x1[n], D_v1_x2[n], D_v1_x3[n], D_v1_t1[n], D_v1_t2[n],
    D_v1_t3[n], D_v1_dt1[n], D_v1_dt2[n], D_v1_dt3[n], D_v1_d2t1[n],
    D_v1_d2t2[n], D_v1_d2t3[n], D_v1_d3t1[n], D_v1_d3t2[n], D_v1_d3t3[n];
float D_v2_x1[n], D_v2_x2[n], D_v2_x3[n], D_v2_t1[n], D_v2_t2[n],
    D_v2_t3[n], D_v2_dt1[n], D_v2_dt2[n], D_v2_dt3[n], D_v2_d2t1[n],
    D_v2_d2t2[n], D_v2_d2t3[n], D_v2_d3t1[n], D_v2_d3t2[n], D_v2_d3t3[n];
float D_D2uDs2[n], D_D2vDs2[n], D_GeoCur[n];

AllDetails( init_s, D_pu, D_pv, D_u1_x1, D_u1_x2, D_u1_x3, D_u1_t1, D_u1_t2,
    D_u1_t3, D_u1_dt1, D_u1_dt2, D_u1_dt3, D_u1_d2t1, D_u1_d2t2, D_u1_d2t3,
    D_u1_d3t1, D_u1_d3t2, D_u1_d3t3, D_u2_x1, D_u2_x2, D_u2_x3, D_u2_t1,
    D_u2_t2, D_u2_t3, D_u2_dt1, D_u2_dt2, D_u2_dt3, D_u2_d2t1, D_u2_d2t2,
    D_u2_d2t3, D_u2_d3t1, D_u2_d3t2, D_u2_d3t3, D_v1_x1, D_v1_x2,
    D_v1_x3, D_v1_t1, D_v1_t2, D_v1_t3, D_v1_dt1, D_v1_dt2, D_v1_dt3,
    D_v1_d2t1, D_v1_d2t2, D_v1_d2t3, D_v1_d3t1, D_v1_d3t2, D_v1_d3t3,
    D_v2_x1, D_v2_x2, D_v2_x3, D_v2_t1, D_v2_t2, D_v2_t3, D_v2_dt1,
    D_v2_dt2, D_v2_dt3, D_v2_d2t1, D_v2_d2t2, D_v2_d2t3, D_v2_d3t1,
    D_v2_d3t2, D_v2_d3t3, D_D2uDs2, D_D2vDs2, D_GeoCur, optimum);

if( index < ListSize)
{
    pu[index]=D_pu[index];
    pv[index]=D_pv[index];
    u1_x1[index]=D_u1_x1[index];
    u1_x2[index]=D_u1_x2[index];
    u1_x3[index]=D_u1_x3[index];
    u1_t1[index]=D_u1_t1[index];
    u1_t2[index]=D_u1_t2[index];
    u1_t3[index]=D_u1_t3[index];
    u1_dt1[index]=D_u1_dt1[index];
    u1_dt2[index]=D_u1_dt2[index];
    u1_dt3[index]=D_u1_dt3[index];
    u1_d2t1[index]=D_u1_d2t1[index];
    u1_d2t2[index]=D_u1_d2t2[index];
    u1_d2t3[index]=D_u1_d2t3[index];
    u1_d3t1[index]=D_u1_d3t1[index];
    u1_d3t2[index]=D_u1_d3t2[index];
    u1_d3t3[index]=D_u1_d3t3[index];
    u2_x1[index]=D_u2_x1[index];
    u2_x2[index]=D_u2_x2[index];
    u2_x3[index]=D_u2_x3[index];
    u2_t1[index]=D_u2_t1[index];
```

```
    u2_t2[index]=D_u2_t2[index];
    u2_t3[index]=D_u2_t3[index];
    u2_dt1[index]=D_u2_dt1[index];
    u2_dt2[index]=D_u2_dt2[index];
    u2_dt3[index]=D_u2_dt3[index];
    u2_d2t1[index]=D_u2_d2t1[index];
    u2_d2t2[index]=D_u2_d2t2[index];
    u2_d2t3[index]=D_u2_d2t3[index];
    u2_d3t1[index]=D_u2_d3t1[index];
    u2_d3t2[index]=D_u2_d3t2[index];
    u2_d3t3[index]=D_u2_d3t3[index];

    v1_x1[index]=D_v1_x1[index];
    v1_x2[index]=D_v1_x2[index];
    v1_x3[index]=D_v1_x3[index];
    v1_t1[index]=D_v1_t1[index];
    v1_t2[index]=D_v1_t2[index];
    v1_t3[index]=D_v1_t3[index];
    v1_dt1[index]=D_v1_dt1[index];
    v1_dt2[index]=D_v1_dt2[index];
    v1_dt3[index]=D_v1_dt3[index];
    v1_d2t1[index]=D_v1_d2t1[index];
    v1_d2t2[index]=D_v1_d2t2[index];
    v1_d2t3[index]=D_v1_d2t3[index];
    v1_d3t1[index]=D_v1_d3t1[index];
    v1_d3t2[index]=D_v1_d3t2[index];
    v1_d3t3[index]=D_v1_d3t3[index];
    v2_x1[index]=D_v2_x1[index];
    v2_x2[index]=D_v2_x2[index];
    v2_x3[index]=D_v2_x3[index];
    v2_t1[index]=D_v2_t1[index];
    v2_t2[index]=D_v2_t2[index];
    v2_t3[index]=D_v2_t3[index];
    v2_dt1[index]=D_v2_dt1[index];
    v2_dt2[index]=D_v2_dt2[index];
    v2_dt3[index]=D_v2_dt3[index];
    v2_d2t1[index]=D_v2_d2t1[index];
    v2_d2t2[index]=D_v2_d2t2[index];
    v2_d2t3[index]=D_v2_d2t3[index];
    v2_d3t1[index]=D_v2_d3t1[index];
    v2_d3t2[index]=D_v2_d3t2[index];
    v2_d3t3[index]=D_v2_d3t3[index];

    D2uDs2[index]=D_D2uDs2[index];
    D2vDs2[index]=D_D2vDs2[index];
    GeoCur[index]=D_GeoCur[index];
}
```

```
}
__global__ void CurvatureofLoC( float * pvalue, float pu[], float pv[], float
    u1_x1[], float u1_x2[], float u1_x3[], float u1_t1[], float u1_t2[],
    float u1_t3[], float u1_dt1[], float u1_dt2[], float u1_dt3[],
    float u1_d2t1[], float u1_d2t2[], float u1_d2t3[], float u1_d3t1[],
    float u1_d3t2[], float u1_d3t3[], float u2_x1[], float u2_x2[],
    float u2_x3[], float u2_t1[], float u2_t2[], float u2_t3[], float
    u2_dt1[], float u2_dt2[], float u2_dt3[], float u2_d2t1[], float
    u2_d2t2[], float u2_d2t3[], float u2_d3t1[], float u2_d3t2[],
    float u2_d3t3[], float v1_x1[], float v1_x2[], float v1_x3[],
    float v1_t1[], float v1_t2[], float v1_t3[], float v1_dt1[], float
    v1_dt2[], float v1_dt3[], float v1_d2t1[], float v1_d2t2[], float
    v1_d2t3[], float v1_d3t1[], float v1_d3t2[], float v1_d3t3[],
    float v2_x1[], float v2_x2[], float v2_x3[], float v2_t1[], float
    v2_t2[], float v2_t3[], float v2_dt1[], float v2_dt2[], float
    v2_dt3[], float v2_d2t1[], float v2_d2t2[], float v2_d2t3[], float
    v2_d3t1[], float v2_d3t2[], float v2_d3t3[], float D2uDs2[],
    float D2vDs2[], float GeoCur[], mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;

if( index < ListSize)
{

    pvalue[index]=Curvecurvature(index,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,
    u1_t3,u1_dt1,u1_dt2,u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,
    u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1
    ,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,
    v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1
    ,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2,v2_d3t3,D2uDs2,D2vDs2,GeoCur,optimum);

}
}

__global__ void TorsionofLoC( float * pvalue, float pu[], float pv[], float
    u1_x1[], float u1_x2[], float u1_x3[], float u1_t1[], float u1_t2[],
    float u1_t3[], float u1_dt1[], float u1_dt2[], float u1_dt3[],
    float u1_d2t1[], float u1_d2t2[], float u1_d2t3[], float u1_d3t1[],
    float u1_d3t2[], float u1_d3t3[], float u2_x1[], float u2_x2[],
    float u2_x3[], float u2_t1[], float u2_t2[], float u2_t3[], float
    u2_dt1[], float u2_dt2[], float u2_dt3[], float u2_d2t1[], float
    u2_d2t2[], float u2_d2t3[], float u2_d3t1[], float u2_d3t2[],
    float u2_d3t3[], float v1_x1[], float v1_x2[], float v1_x3[],
    float v1_t1[], float v1_t2[], float v1_t3[], float v1_dt1[], float
    v1_dt2[], float v1_dt3[], float v1_d2t1[], float v1_d2t2[], float
```

```
    v1_d2t3[], float v1_d3t1[], float v1_d3t2[], float v1_d3t3[],
    float v2_x1[], float v2_x2[], float v2_x3[], float v2_t1[], float
    v2_t2[], float v2_t3[], float v2_dt1[], float v2_dt2[], float
    v2_dt3[], float v2_d2t1[], float v2_d2t2[], float v2_d2t3[], float
    v2_d3t1[], float v2_d3t2[], float v2_d3t3[], float D2uDs2[],
    float D2vDs2[], float GeoCur[], mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;

if( index < ListSize)
{

    pvalue[index]=Torsion(index,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,
    u1_dt1,u1_dt2,u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2
    ,u2_d2t3,u2_d3t1,u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,
    v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,
    v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2
    ,v2_d2t3,v2_d3t1,v2_d3t2,v2_d3t3,D2uDs2,D2vDs2,GeoCur,optimum);

}
}

__global__ void DCurvatureDsofLoC( float * pvalue, float pu[], float pv[],
    float u1_x1[], float u1_x2[], float u1_x3[], float u1_t1[], float
    u1_t2[], float u1_t3[], float u1_dt1[], float u1_dt2[], float
    u1_dt3[], float u1_d2t1[], float u1_d2t2[], float u1_d2t3[], float
    u1_d3t1[], float u1_d3t2[], float u1_d3t3[], float u2_x1[], float
    u2_x2[], float u2_x3[], float u2_t1[], float u2_t2[], float u2_t3[],
    float u2_dt1[], float u2_dt2[], float u2_dt3[], float u2_d2t1[],
    float u2_d2t2[], float u2_d2t3[], float u2_d3t1[], float u2_d3t2[],
    float u2_d3t3[], float v1_x1[], float v1_x2[], float v1_x3[],
    float v1_t1[], float v1_t2[], float v1_t3[], float v1_dt1[], float
    v1_dt2[], float v1_dt3[], float v1_d2t1[], float v1_d2t2[], float
    v1_d2t3[], float v1_d3t1[], float v1_d3t2[], float v1_d3t3[],
    float v2_x1[], float v2_x2[], float v2_x3[], float v2_t1[], float
    v2_t2[], float v2_t3[], float v2_dt1[], float v2_dt2[], float
    v2_dt3[], float v2_d2t1[], float v2_d2t2[], float v2_d2t3[], float
    v2_d3t1[], float v2_d3t2[], float v2_d3t3[], float D2uDs2[],
    float D2vDs2[], float GeoCur[], mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;

if( index < ListSize)
{

    pvalue[index]=DCurvatureDs(index,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,
```

```
    u1_t3,u1_dt1,u1_dt2,u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,
    u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1
    ,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,
    v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1
    ,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2,v2_d3t3,D2uDs2,D2vDs2,GeoCur,optimum);

}
}

";
```

```
Needs["CUDALink`"]
```

```
AbsoluteTiming[
 CSAllDetails = CUDAFunctionLoad[kernelcodeCPLAS2, "ReturnEveryDetails",
   {{"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, _Integer, _Integer, _Integer}, 256]]
```

```
{655.985, CUDAFunction[<>, ReturnEveryDetails,
  {{Float, _, Output}, {Float, _, Output}, {Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, {Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, {Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, {Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, {Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, {Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, {Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, {Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, {Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, {Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, {Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, {Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, {Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, {Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, {Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, {Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, Integer64, Integer64, Integer64}]}
```

```
CSCurvature =
  CUDAFunctionLoad[kernelcodeCPLAS2, "CurvatureofLoC", {{"Float", _, "Output"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
```

```
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, _Integer, _Integer}, 160];
CSTorsion = CUDAFunctionLoad[kernelcodeCPLAS2, "TorsionofLoC",
    {{"Float", _, "Output"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
      {"Float", _, "Input"}, _Integer, _Integer}, 160];
CSDerivativeofCurvature = CUDAFunctionLoad[kernelcodeCPLAS2,
    "DCurvatureDsofLoC", {{"Float", _, "Output"}, {"Float", _, "Input"},
```

```
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, _Integer, _Integer}, 160];

CSReturnAll[v_, nn_, optimum_] :=
  Module[{opt1 = optimum, pu, pv, u1x1, u1x2, u1x3, u1t1, u1t2, u1t3, u1dt1,
    u1dt2, u1dt3, u1d2t1, u1d2t2, u1d2t3, u1d3t1, u1d3t2, u1d3t3, u2x1,
    u2x2, u2x3, u2t1, u2t2, u2t3, u2dt1, u2dt2, u2dt3, u2d2t1, u2d2t2,
    u2d2t3, u2d3t1, u2d3t2, u2d3t3, v1x1, v1x2, v1x3, v1t1, v1t2, v1t3,
    v1dt1, v1dt2, v1dt3, v1d2t1, v1d2t2, v1d2t3, v1d3t1, v1d3t2, v1d3t3,
    v2x1, v2x2, v2x3, v2t1, v2t2, v2t3, v2dt1, v2dt2, v2dt3, v2d2t1,
    v2d2t2, v2d2t3, v2d3t1, v2d3t2, v2d3t3, D2uDs2, D2vDs2, GeoCur},
   pu = CUDAMemoryAllocate["Float", nn];
   pv = CUDAMemoryAllocate["Float", nn];
   u1x1 = CUDAMemoryAllocate["Float", nn];
   u1x2 = CUDAMemoryAllocate["Float", nn];
   u1x3 = CUDAMemoryAllocate["Float", nn];
   u1t1 = CUDAMemoryAllocate["Float", nn];
   u1t2 = CUDAMemoryAllocate["Float", nn];
   u1t3 = CUDAMemoryAllocate["Float", nn];
   u1dt1 = CUDAMemoryAllocate["Float", nn];
   u1dt2 = CUDAMemoryAllocate["Float", nn];
   u1dt3 = CUDAMemoryAllocate["Float", nn];
   u1d2t1 = CUDAMemoryAllocate["Float", nn];
   u1d2t2 = CUDAMemoryAllocate["Float", nn];
   u1d2t3 = CUDAMemoryAllocate["Float", nn];
   u1d3t1 = CUDAMemoryAllocate["Float", nn];
   u1d3t2 = CUDAMemoryAllocate["Float", nn];
   u1d3t3 = CUDAMemoryAllocate["Float", nn];
```

```
u2x1 = CUDAMemoryAllocate["Float", nn];
u2x2 = CUDAMemoryAllocate["Float", nn];
u2x3 = CUDAMemoryAllocate["Float", nn];
u2t1 = CUDAMemoryAllocate["Float", nn];
u2t2 = CUDAMemoryAllocate["Float", nn];
u2t3 = CUDAMemoryAllocate["Float", nn];
u2dt1 = CUDAMemoryAllocate["Float", nn];
u2dt2 = CUDAMemoryAllocate["Float", nn];
u2dt3 = CUDAMemoryAllocate["Float", nn];
u2d2t1 = CUDAMemoryAllocate["Float", nn];
u2d2t2 = CUDAMemoryAllocate["Float", nn];
u2d2t3 = CUDAMemoryAllocate["Float", nn];
u2d3t1 = CUDAMemoryAllocate["Float", nn];
u2d3t2 = CUDAMemoryAllocate["Float", nn];
u2d3t3 = CUDAMemoryAllocate["Float", nn];
v1x1 = CUDAMemoryAllocate["Float", nn];
v1x2 = CUDAMemoryAllocate["Float", nn];
v1x3 = CUDAMemoryAllocate["Float", nn];
v1t1 = CUDAMemoryAllocate["Float", nn];
v1t2 = CUDAMemoryAllocate["Float", nn];
v1t3 = CUDAMemoryAllocate["Float", nn];
v1dt1 = CUDAMemoryAllocate["Float", nn];
v1dt2 = CUDAMemoryAllocate["Float", nn];
v1dt3 = CUDAMemoryAllocate["Float", nn];
v1d2t1 = CUDAMemoryAllocate["Float", nn];
v1d2t2 = CUDAMemoryAllocate["Float", nn];
v1d2t3 = CUDAMemoryAllocate["Float", nn];
v1d3t1 = CUDAMemoryAllocate["Float", nn];
v1d3t2 = CUDAMemoryAllocate["Float", nn];
v1d3t3 = CUDAMemoryAllocate["Float", nn];
v2x1 = CUDAMemoryAllocate["Float", nn];
v2x2 = CUDAMemoryAllocate["Float", nn];
v2x3 = CUDAMemoryAllocate["Float", nn];
v2t1 = CUDAMemoryAllocate["Float", nn];
v2t2 = CUDAMemoryAllocate["Float", nn];
v2t3 = CUDAMemoryAllocate["Float", nn];
v2dt1 = CUDAMemoryAllocate["Float", nn];
v2dt2 = CUDAMemoryAllocate["Float", nn];
v2dt3 = CUDAMemoryAllocate["Float", nn];
v2d2t1 = CUDAMemoryAllocate["Float", nn];
v2d2t2 = CUDAMemoryAllocate["Float", nn];
v2d2t3 = CUDAMemoryAllocate["Float", nn];
v2d3t1 = CUDAMemoryAllocate["Float", nn];
v2d3t2 = CUDAMemoryAllocate["Float", nn];
v2d3t3 = CUDAMemoryAllocate["Float", nn];
D2uDs2 = CUDAMemoryAllocate["Float", nn];
D2vDs2 = CUDAMemoryAllocate["Float", nn];
```

```
    GeoCur = CUDAMemoryAllocate["Float", nn];

    CSAllDetails[pu, pv, u1x1, u1x2, u1x3, u1t1, u1t2, u1t3, u1dt1, u1dt2, u1dt3,
     u1d2t1, u1d2t2, u1d2t3, u1d3t1, u1d3t2, u1d3t3, u2x1, u2x2, u2x3,
     u2t1, u2t2, u2t3, u2dt1, u2dt2, u2dt3, u2d2t1, u2d2t2, u2d2t3, u2d3t1,
     u2d3t2, u2d3t3, v1x1, v1x2, v1x3, v1t1, v1t2, v1t3, v1dt1, v1dt2,
     v1dt3, v1d2t1, v1d2t2, v1d2t3, v1d3t1, v1d3t2, v1d3t3, v2x1, v2x2,
     v2x3, v2t1, v2t2, v2t3, v2dt1, v2dt2, v2dt3, v2d2t1, v2d2t2, v2d2t3,
     v2d3t1, v2d3t2, v2d3t3, D2uDs2, D2vDs2, GeoCur, v, optimum, nn];
    Transpose[{CUDAMemoryGet[pu], CUDAMemoryGet[pv], CUDAMemoryGet[D2uDs2],
      CUDAMemoryGet[D2vDs2], CUDAMemoryGet[GeoCur]}]];

CSCurvatureofLoC[v_, nn_, optimum_] :=
  Module[{opt1 = optimum, pu, pv, u1x1, u1x2, u1x3, u1t1, u1t2, u1t3, u1dt1,
     u1dt2, u1dt3, u1d2t1, u1d2t2, u1d2t3, u1d3t1, u1d3t2, u1d3t3, u2x1,
     u2x2, u2x3, u2t1, u2t2, u2t3, u2dt1, u2dt2, u2dt3, u2d2t1, u2d2t2,
     u2d2t3, u2d3t1, u2d3t2, u2d3t3, v1x1, v1x2, v1x3, v1t1, v1t2, v1t3,
     v1dt1, v1dt2, v1dt3, v1d2t1, v1d2t2, v1d2t3, v1d3t1, v1d3t2, v1d3t3,
     v2x1, v2x2, v2x3, v2t1, v2t2, v2t3, v2dt1, v2dt2, v2dt3, v2d2t1,
     v2d2t2, v2d2t3, v2d3t1, v2d3t2, v2d3t3, D2uDs2, D2vDs2, GeoCur, Cur},
    pu = CUDAMemoryAllocate["Float", nn];
    pv = CUDAMemoryAllocate["Float", nn];
    u1x1 = CUDAMemoryAllocate["Float", nn];
    u1x2 = CUDAMemoryAllocate["Float", nn];
    u1x3 = CUDAMemoryAllocate["Float", nn];
    u1t1 = CUDAMemoryAllocate["Float", nn];
    u1t2 = CUDAMemoryAllocate["Float", nn];
    u1t3 = CUDAMemoryAllocate["Float", nn];
    u1dt1 = CUDAMemoryAllocate["Float", nn];
    u1dt2 = CUDAMemoryAllocate["Float", nn];
    u1dt3 = CUDAMemoryAllocate["Float", nn];
    u1d2t1 = CUDAMemoryAllocate["Float", nn];
    u1d2t2 = CUDAMemoryAllocate["Float", nn];
    u1d2t3 = CUDAMemoryAllocate["Float", nn];
    u1d3t1 = CUDAMemoryAllocate["Float", nn];
    u1d3t2 = CUDAMemoryAllocate["Float", nn];
    u1d3t3 = CUDAMemoryAllocate["Float", nn];
    u2x1 = CUDAMemoryAllocate["Float", nn];
    u2x2 = CUDAMemoryAllocate["Float", nn];
    u2x3 = CUDAMemoryAllocate["Float", nn];
    u2t1 = CUDAMemoryAllocate["Float", nn];
    u2t2 = CUDAMemoryAllocate["Float", nn];
    u2t3 = CUDAMemoryAllocate["Float", nn];
    u2dt1 = CUDAMemoryAllocate["Float", nn];
    u2dt2 = CUDAMemoryAllocate["Float", nn];
    u2dt3 = CUDAMemoryAllocate["Float", nn];
    u2d2t1 = CUDAMemoryAllocate["Float", nn];
    u2d2t2 = CUDAMemoryAllocate["Float", nn];
```

```
u2d2t3 = CUDAMemoryAllocate["Float", nn];
u2d3t1 = CUDAMemoryAllocate["Float", nn];
u2d3t2 = CUDAMemoryAllocate["Float", nn];
u2d3t3 = CUDAMemoryAllocate["Float", nn];
v1x1 = CUDAMemoryAllocate["Float", nn];
v1x2 = CUDAMemoryAllocate["Float", nn];
v1x3 = CUDAMemoryAllocate["Float", nn];
v1t1 = CUDAMemoryAllocate["Float", nn];
v1t2 = CUDAMemoryAllocate["Float", nn];
v1t3 = CUDAMemoryAllocate["Float", nn];
v1dt1 = CUDAMemoryAllocate["Float", nn];
v1dt2 = CUDAMemoryAllocate["Float", nn];
v1dt3 = CUDAMemoryAllocate["Float", nn];
v1d2t1 = CUDAMemoryAllocate["Float", nn];
v1d2t2 = CUDAMemoryAllocate["Float", nn];
v1d2t3 = CUDAMemoryAllocate["Float", nn];
v1d3t1 = CUDAMemoryAllocate["Float", nn];
v1d3t2 = CUDAMemoryAllocate["Float", nn];
v1d3t3 = CUDAMemoryAllocate["Float", nn];
v2x1 = CUDAMemoryAllocate["Float", nn];
v2x2 = CUDAMemoryAllocate["Float", nn];
v2x3 = CUDAMemoryAllocate["Float", nn];
v2t1 = CUDAMemoryAllocate["Float", nn];
v2t2 = CUDAMemoryAllocate["Float", nn];
v2t3 = CUDAMemoryAllocate["Float", nn];
v2dt1 = CUDAMemoryAllocate["Float", nn];
v2dt2 = CUDAMemoryAllocate["Float", nn];
v2dt3 = CUDAMemoryAllocate["Float", nn];
v2d2t1 = CUDAMemoryAllocate["Float", nn];
v2d2t2 = CUDAMemoryAllocate["Float", nn];
v2d2t3 = CUDAMemoryAllocate["Float", nn];
v2d3t1 = CUDAMemoryAllocate["Float", nn];
v2d3t2 = CUDAMemoryAllocate["Float", nn];
v2d3t3 = CUDAMemoryAllocate["Float", nn];
D2uDs2 = CUDAMemoryAllocate["Float", nn];
D2vDs2 = CUDAMemoryAllocate["Float", nn];
GeoCur = CUDAMemoryAllocate["Float", nn];
Cur = CUDAMemoryAllocate["Float", nn];

CSAllDetails[pu, pv, u1x1, u1x2, u1x3, u1t1, u1t2, u1t3, u1dt1, u1dt2, u1dt3,
 u1d2t1, u1d2t2, u1d2t3, u1d3t1, u1d3t2, u1d3t3, u2x1, u2x2, u2x3,
 u2t1, u2t2, u2t3, u2dt1, u2dt2, u2dt3, u2d2t1, u2d2t2, u2d2t3, u2d3t1,
 u2d3t2, u2d3t3, v1x1, v1x2, v1x3, v1t1, v1t2, v1t3, v1dt1, v1dt2,
 v1dt3, v1d2t1, v1d2t2, v1d2t3, v1d3t1, v1d3t2, v1d3t3, v2x1, v2x2,
 v2x3, v2t1, v2t2, v2t3, v2dt1, v2dt2, v2dt3, v2d2t1, v2d2t2, v2d2t3,
 v2d3t1, v2d3t2, v2d3t3, D2uDs2, D2vDs2, GeoCur, v, optimum, nn];
CSCurvature[Cur, pu, pv, u1x1, u1x2, u1x3, u1t1, u1t2, u1t3, u1dt1,
```

```
      u1dt2, u1dt3, u1d2t1, u1d2t2, u1d2t3, u1d3t1, u1d3t2, u1d3t3, u2x1,
      u2x2, u2x3, u2t1, u2t2, u2t3, u2dt1, u2dt2, u2dt3, u2d2t1, u2d2t2,
      u2d2t3, u2d3t1, u2d3t2, u2d3t3, v1x1, v1x2, v1x3, v1t1, v1t2, v1t3,
      v1dt1, v1dt2, v1dt3, v1d2t1, v1d2t2, v1d2t3, v1d3t1, v1d3t2, v1d3t3,
      v2x1, v2x2, v2x3, v2t1, v2t2, v2t3, v2dt1, v2dt2, v2dt3, v2d2t1, v2d2t2,
      v2d2t3, v2d3t1, v2d3t2, v2d3t3, D2uDs2, D2vDs2, GeoCur, optimum, nn];
    Transpose[{Table[i, {i, 0, (nn - 1) / 1000, 0.001}], CUDAMemoryGet[Cur]}]];

CSTorsionofLoC[v_, nn_, optimum_] :=
  Module[{opt1 = optimum, pu, pv, u1x1, u1x2, u1x3, u1t1, u1t2, u1t3, u1dt1,
      u1dt2, u1dt3, u1d2t1, u1d2t2, u1d2t3, u1d3t1, u1d3t2, u1d3t3, u2x1,
      u2x2, u2x3, u2t1, u2t2, u2t3, u2dt1, u2dt2, u2dt3, u2d2t1, u2d2t2,
      u2d2t3, u2d3t1, u2d3t2, u2d3t3, v1x1, v1x2, v1x3, v1t1, v1t2, v1t3,
      v1dt1, v1dt2, v1dt3, v1d2t1, v1d2t2, v1d2t3, v1d3t1, v1d3t2, v1d3t3,
      v2x1, v2x2, v2x3, v2t1, v2t2, v2t3, v2dt1, v2dt2, v2dt3, v2d2t1, v2d2t2,
      v2d2t3, v2d3t1, v2d3t2, v2d3t3, D2uDs2, D2vDs2, GeoCur, Torsion},
    pu = CUDAMemoryAllocate["Float", nn];
    pv = CUDAMemoryAllocate["Float", nn];
    u1x1 = CUDAMemoryAllocate["Float", nn];
    u1x2 = CUDAMemoryAllocate["Float", nn];
    u1x3 = CUDAMemoryAllocate["Float", nn];
    u1t1 = CUDAMemoryAllocate["Float", nn];
    u1t2 = CUDAMemoryAllocate["Float", nn];
    u1t3 = CUDAMemoryAllocate["Float", nn];
    u1dt1 = CUDAMemoryAllocate["Float", nn];
    u1dt2 = CUDAMemoryAllocate["Float", nn];
    u1dt3 = CUDAMemoryAllocate["Float", nn];
    u1d2t1 = CUDAMemoryAllocate["Float", nn];
    u1d2t2 = CUDAMemoryAllocate["Float", nn];
    u1d2t3 = CUDAMemoryAllocate["Float", nn];
    u1d3t1 = CUDAMemoryAllocate["Float", nn];
    u1d3t2 = CUDAMemoryAllocate["Float", nn];
    u1d3t3 = CUDAMemoryAllocate["Float", nn];
    u2x1 = CUDAMemoryAllocate["Float", nn];
    u2x2 = CUDAMemoryAllocate["Float", nn];
    u2x3 = CUDAMemoryAllocate["Float", nn];
    u2t1 = CUDAMemoryAllocate["Float", nn];
    u2t2 = CUDAMemoryAllocate["Float", nn];
    u2t3 = CUDAMemoryAllocate["Float", nn];
    u2dt1 = CUDAMemoryAllocate["Float", nn];
    u2dt2 = CUDAMemoryAllocate["Float", nn];
    u2dt3 = CUDAMemoryAllocate["Float", nn];
    u2d2t1 = CUDAMemoryAllocate["Float", nn];
    u2d2t2 = CUDAMemoryAllocate["Float", nn];
    u2d2t3 = CUDAMemoryAllocate["Float", nn];
    u2d3t1 = CUDAMemoryAllocate["Float", nn];
    u2d3t2 = CUDAMemoryAllocate["Float", nn];
    u2d3t3 = CUDAMemoryAllocate["Float", nn];
```

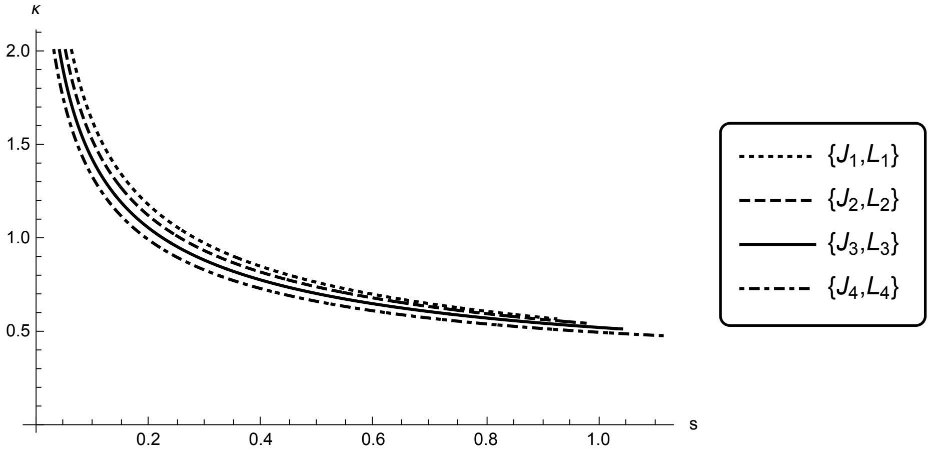
```
v1x1 = CUDAMemoryAllocate["Float", nn];
v1x2 = CUDAMemoryAllocate["Float", nn];
v1x3 = CUDAMemoryAllocate["Float", nn];
v1t1 = CUDAMemoryAllocate["Float", nn];
v1t2 = CUDAMemoryAllocate["Float", nn];
v1t3 = CUDAMemoryAllocate["Float", nn];
v1dt1 = CUDAMemoryAllocate["Float", nn];
v1dt2 = CUDAMemoryAllocate["Float", nn];
v1dt3 = CUDAMemoryAllocate["Float", nn];
v1d2t1 = CUDAMemoryAllocate["Float", nn];
v1d2t2 = CUDAMemoryAllocate["Float", nn];
v1d2t3 = CUDAMemoryAllocate["Float", nn];
v1d3t1 = CUDAMemoryAllocate["Float", nn];
v1d3t2 = CUDAMemoryAllocate["Float", nn];
v1d3t3 = CUDAMemoryAllocate["Float", nn];
v2x1 = CUDAMemoryAllocate["Float", nn];
v2x2 = CUDAMemoryAllocate["Float", nn];
v2x3 = CUDAMemoryAllocate["Float", nn];
v2t1 = CUDAMemoryAllocate["Float", nn];
v2t2 = CUDAMemoryAllocate["Float", nn];
v2t3 = CUDAMemoryAllocate["Float", nn];
v2dt1 = CUDAMemoryAllocate["Float", nn];
v2dt2 = CUDAMemoryAllocate["Float", nn];
v2dt3 = CUDAMemoryAllocate["Float", nn];
v2d2t1 = CUDAMemoryAllocate["Float", nn];
v2d2t2 = CUDAMemoryAllocate["Float", nn];
v2d2t3 = CUDAMemoryAllocate["Float", nn];
v2d3t1 = CUDAMemoryAllocate["Float", nn];
v2d3t2 = CUDAMemoryAllocate["Float", nn];
v2d3t3 = CUDAMemoryAllocate["Float", nn];
D2uDs2 = CUDAMemoryAllocate["Float", nn];
D2vDs2 = CUDAMemoryAllocate["Float", nn];
GeoCur = CUDAMemoryAllocate["Float", nn];
Torsion = CUDAMemoryAllocate["Float", nn];

CSAllDetails[pu, pv, u1x1, u1x2, u1x3, u1t1, u1t2, u1t3, u1dt1, u1dt2, u1dt3,
 u1d2t1, u1d2t2, u1d2t3, u1d3t1, u1d3t2, u1d3t3, u2x1, u2x2, u2x3,
 u2t1, u2t2, u2t3, u2dt1, u2dt2, u2dt3, u2d2t1, u2d2t2, u2d2t3, u2d3t1,
 u2d3t2, u2d3t3, v1x1, v1x2, v1x3, v1t1, v1t2, v1t3, v1dt1, v1dt2,
 v1dt3, v1d2t1, v1d2t2, v1d2t3, v1d3t1, v1d3t2, v1d3t3, v2x1, v2x2,
 v2x3, v2t1, v2t2, v2t3, v2dt1, v2dt2, v2dt3, v2d2t1, v2d2t2, v2d2t3,
 v2d3t1, v2d3t2, v2d3t3, D2uDs2, D2vDs2, GeoCur, v, optimum, nn];
CSTorsion[Torsion, pu, pv, u1x1, u1x2, u1x3, u1t1, u1t2, u1t3, u1dt1,
 u1dt2, u1dt3, u1d2t1, u1d2t2, u1d2t3, u1d3t1, u1d3t2, u1d3t3, u2x1,
 u2x2, u2x3, u2t1, u2t2, u2t3, u2dt1, u2dt2, u2dt3, u2d2t1, u2d2t2,
 u2d2t3, u2d3t1, u2d3t2, u2d3t3, v1x1, v1x2, v1x3, v1t1, v1t2, v1t3,
 v1dt1, v1dt2, v1dt3, v1d2t1, v1d2t2, v1d2t3, v1d3t1, v1d3t2, v1d3t3,
```

```
    v2x1, v2x2, v2x3, v2t1, v2t2, v2t3, v2dt1, v2dt2, v2dt3, v2d2t1, v2d2t2,
    v2d2t3, v2d3t1, v2d3t2, v2d3t3, D2uDs2, D2vDs2, GeoCur, optimum, nn];
   Transpose[{Table[i, {i, 0, (nn - 1) / 1000, 0.001}], CUDAMemoryGet[Torsion]}]];

CSDerivativeCurvatureofLoC[v_, nn_, optimum_] :=
  Module[{opt1 = optimum, pu, pv, u1x1, u1x2, u1x3, u1t1, u1t2, u1t3, u1dt1,
    u1dt2, u1dt3, u1d2t1, u1d2t2, u1d2t3, u1d3t1, u1d3t2, u1d3t3, u2x1,
    u2x2, u2x3, u2t1, u2t2, u2t3, u2dt1, u2dt2, u2dt3, u2d2t1, u2d2t2,
    u2d2t3, u2d3t1, u2d3t2, u2d3t3, v1x1, v1x2, v1x3, v1t1, v1t2, v1t3,
    v1dt1, v1dt2, v1dt3, v1d2t1, v1d2t2, v1d2t3, v1d3t1, v1d3t2, v1d3t3,
    v2x1, v2x2, v2x3, v2t1, v2t2, v2t3, v2dt1, v2dt2, v2dt3, v2d2t1, v2d2t2,
    v2d2t3, v2d3t1, v2d3t2, v2d3t3, D2uDs2, D2vDs2, GeoCur, DCurDs},
   pu = CUDAMemoryAllocate["Float", nn];
   pv = CUDAMemoryAllocate["Float", nn];
   u1x1 = CUDAMemoryAllocate["Float", nn];
   u1x2 = CUDAMemoryAllocate["Float", nn];
   u1x3 = CUDAMemoryAllocate["Float", nn];
   u1t1 = CUDAMemoryAllocate["Float", nn];
   u1t2 = CUDAMemoryAllocate["Float", nn];
   u1t3 = CUDAMemoryAllocate["Float", nn];
   u1dt1 = CUDAMemoryAllocate["Float", nn];
   u1dt2 = CUDAMemoryAllocate["Float", nn];
   u1dt3 = CUDAMemoryAllocate["Float", nn];
   u1d2t1 = CUDAMemoryAllocate["Float", nn];
   u1d2t2 = CUDAMemoryAllocate["Float", nn];
   u1d2t3 = CUDAMemoryAllocate["Float", nn];
   u1d3t1 = CUDAMemoryAllocate["Float", nn];
   u1d3t2 = CUDAMemoryAllocate["Float", nn];
   u1d3t3 = CUDAMemoryAllocate["Float", nn];
   u2x1 = CUDAMemoryAllocate["Float", nn];
   u2x2 = CUDAMemoryAllocate["Float", nn];
   u2x3 = CUDAMemoryAllocate["Float", nn];
   u2t1 = CUDAMemoryAllocate["Float", nn];
   u2t2 = CUDAMemoryAllocate["Float", nn];
   u2t3 = CUDAMemoryAllocate["Float", nn];
   u2dt1 = CUDAMemoryAllocate["Float", nn];
   u2dt2 = CUDAMemoryAllocate["Float", nn];
   u2dt3 = CUDAMemoryAllocate["Float", nn];
   u2d2t1 = CUDAMemoryAllocate["Float", nn];
   u2d2t2 = CUDAMemoryAllocate["Float", nn];
   u2d2t3 = CUDAMemoryAllocate["Float", nn];
   u2d3t1 = CUDAMemoryAllocate["Float", nn];
   u2d3t2 = CUDAMemoryAllocate["Float", nn];
   u2d3t3 = CUDAMemoryAllocate["Float", nn];
   v1x1 = CUDAMemoryAllocate["Float", nn];
   v1x2 = CUDAMemoryAllocate["Float", nn];
   v1x3 = CUDAMemoryAllocate["Float", nn];
   v1t1 = CUDAMemoryAllocate["Float", nn];
```

```
v1t2 = CUDAMemoryAllocate["Float", nn];
v1t3 = CUDAMemoryAllocate["Float", nn];
v1dt1 = CUDAMemoryAllocate["Float", nn];
v1dt2 = CUDAMemoryAllocate["Float", nn];
v1dt3 = CUDAMemoryAllocate["Float", nn];
v1d2t1 = CUDAMemoryAllocate["Float", nn];
v1d2t2 = CUDAMemoryAllocate["Float", nn];
v1d2t3 = CUDAMemoryAllocate["Float", nn];
v1d3t1 = CUDAMemoryAllocate["Float", nn];
v1d3t2 = CUDAMemoryAllocate["Float", nn];
v1d3t3 = CUDAMemoryAllocate["Float", nn];
v2x1 = CUDAMemoryAllocate["Float", nn];
v2x2 = CUDAMemoryAllocate["Float", nn];
v2x3 = CUDAMemoryAllocate["Float", nn];
v2t1 = CUDAMemoryAllocate["Float", nn];
v2t2 = CUDAMemoryAllocate["Float", nn];
v2t3 = CUDAMemoryAllocate["Float", nn];
v2dt1 = CUDAMemoryAllocate["Float", nn];
v2dt2 = CUDAMemoryAllocate["Float", nn];
v2dt3 = CUDAMemoryAllocate["Float", nn];
v2d2t1 = CUDAMemoryAllocate["Float", nn];
v2d2t2 = CUDAMemoryAllocate["Float", nn];
v2d2t3 = CUDAMemoryAllocate["Float", nn];
v2d3t1 = CUDAMemoryAllocate["Float", nn];
v2d3t2 = CUDAMemoryAllocate["Float", nn];
v2d3t3 = CUDAMemoryAllocate["Float", nn];
D2uDs2 = CUDAMemoryAllocate["Float", nn];
D2vDs2 = CUDAMemoryAllocate["Float", nn];
GeoCur = CUDAMemoryAllocate["Float", nn];
DCurDs = CUDAMemoryAllocate["Float", nn];

CSAllDetails[pu, pv, u1x1, u1x2, u1x3, u1t1, u1t2, u1t3, u1dt1, u1dt2, u1dt3,
 u1d2t1, u1d2t2, u1d2t3, u1d3t1, u1d3t2, u1d3t3, u2x1, u2x2, u2x3,
 u2t1, u2t2, u2t3, u2dt1, u2dt2, u2dt3, u2d2t1, u2d2t2, u2d2t3, u2d3t1,
 u2d3t2, u2d3t3, v1x1, v1x2, v1x3, v1t1, v1t2, v1t3, v1dt1, v1dt2,
 v1dt3, v1d2t1, v1d2t2, v1d2t3, v1d3t1, v1d3t2, v1d3t3, v2x1, v2x2,
 v2x3, v2t1, v2t2, v2t3, v2dt1, v2dt2, v2dt3, v2d2t1, v2d2t2, v2d2t3,
 v2d3t1, v2d3t2, v2d3t3, D2uDs2, D2vDs2, GeoCur, v, optimum, nn];
CSDerivativeofCurvature[DCurDs, pu, pv, u1x1, u1x2, u1x3, u1t1, u1t2,
 u1t3, u1dt1, u1dt2, u1dt3, u1d2t1, u1d2t2, u1d2t3, u1d3t1, u1d3t2, u1d3t3,
 u2x1, u2x2, u2x3, u2t1, u2t2, u2t3, u2dt1, u2dt2, u2dt3, u2d2t1, u2d2t2,
 u2d2t3, u2d3t1, u2d3t2, u2d3t3, v1x1, v1x2, v1x3, v1t1, v1t2, v1t3,
 v1dt1, v1dt2, v1dt3, v1d2t1, v1d2t2, v1d2t3, v1d3t1, v1d3t2, v1d3t3,
 v2x1, v2x2, v2x3, v2t1, v2t2, v2t3, v2dt1, v2dt2, v2dt3, v2d2t1, v2d2t2,
 v2d2t3, v2d3t1, v2d3t2, v2d3t3, D2uDs2, D2vDs2, GeoCur, optimum, nn];
Transpose[{Table[i, {i, 0, (nn - 1) / 1000, 0.001}], CUDAMemoryGet[DCurDs]}]];
```

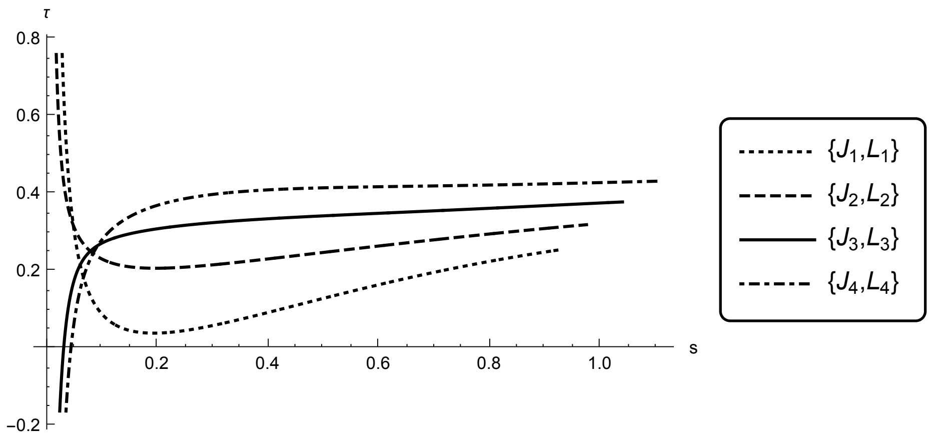
Curvature

```
ListLinePlot[{CSCurvatureofLoC[200, 923, 1],
   CSCurvatureofLoC[400, 975, 1], CSCurvatureofLoC[600, 1040, 1],
   CSCurvatureofLoC[800, 1112, 1]}, AxesLabel → {"s", "κ"},
 PlotStyle → {{Dotted, Black}, {Dashed, Black}, {Black}, {DotDashed, Black}},
 PlotLegends → LineLegend[{{{"J₁,L₁"}, {"J₂,L₂"}, {"J₃,L₃"}, {"J₄,L₄"}}},
   LegendFunction → (Framed[#, RoundingRadius → 5] &), LegendMargins → 5]]
```



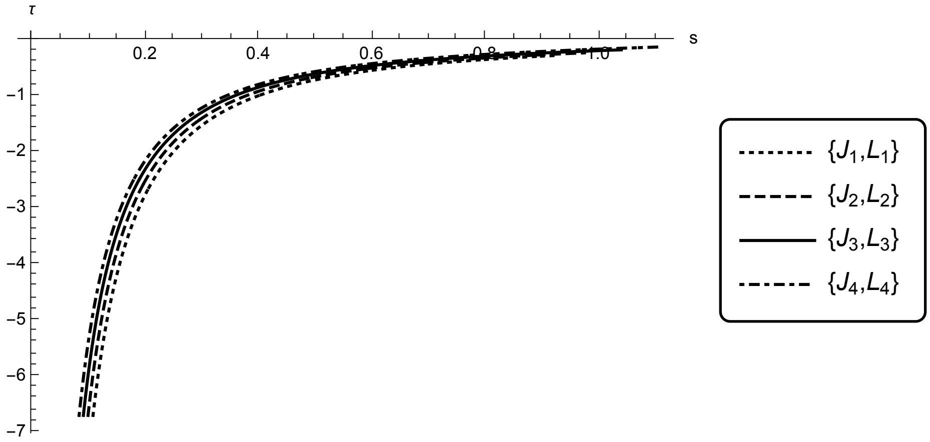
Torsion

```
ListLinePlot[{CSTorsionofLoC[200, 923, 1],
   CSTorsionofLoC[400, 975, 1], CSTorsionofLoC[600, 1040, 1],
   CSTorsionofLoC[800, 1112, 1]}, AxesLabel → {"s", "τ"},
 PlotStyle → {{Dotted, Black}, {Dashed, Black}, {Black}, {DotDashed, Black}},
 PlotLegends → LineLegend[{{{"J₁,L₁"}, {"J₂,L₂"}, {"J₃,L₃"}, {"J₄,L₄"}}},
   LegendFunction → (Framed[#, RoundingRadius → 5] &), LegendMargins → 5]]
```



Derivative of Curvature

```
ListLinePlot[{CSDerivativeCurvatureofLoC[200, 923, 1],
  CSDerivativeCurvatureofLoC[400, 975, 1],
  CSDerivativeCurvatureofLoC[600, 1040, 1],
  CSDerivativeCurvatureofLoC[800, 1112, 1]}, AxesLabel → {"s", "τ"},
 PlotStyle → {{Dotted, Black}, {Dashed, Black}, {Black}, {DotDashed, Black}},
 PlotLegends → LineLegend[{{{"J₁,L₁"}, {"J₂,L₂"}, {"J₃,L₃"}, {"J₄,L₄"}}},
   LegendFunction → (Framed[#, RoundingRadius → 5] &), LegendMargins → 5]]
```



LCG

Set up

```
CSCurvatureofLoC2[v_, nn_, optimum_] :=
  Module[{opt1 = optimum, pu, pv, u1x1, u1x2, u1x3, u1t1, u1t2, u1t3, u1dt1,
    u1dt2, u1dt3, u1d2t1, u1d2t2, u1d2t3, u1d3t1, u1d3t2, u1d3t3, u2x1,
    u2x2, u2x3, u2t1, u2t2, u2t3, u2dt1, u2dt2, u2dt3, u2d2t1, u2d2t2,
    u2d2t3, u2d3t1, u2d3t2, u2d3t3, v1x1, v1x2, v1x3, v1t1, v1t2, v1t3,
    v1dt1, v1dt2, v1dt3, v1d2t1, v1d2t2, v1d2t3, v1d3t1, v1d3t2, v1d3t3,
    v2x1, v2x2, v2x3, v2t1, v2t2, v2t3, v2dt1, v2dt2, v2dt3, v2d2t1,
    v2d2t2, v2d2t3, v2d3t1, v2d3t2, v2d3t3, D2uDs2, D2vDs2, GeoCur, Cur},
   pu = CUDAMemoryAllocate["Float", nn];
   pv = CUDAMemoryAllocate["Float", nn];
   u1x1 = CUDAMemoryAllocate["Float", nn];
   u1x2 = CUDAMemoryAllocate["Float", nn];
   u1x3 = CUDAMemoryAllocate["Float", nn];
   u1t1 = CUDAMemoryAllocate["Float", nn];
   u1t2 = CUDAMemoryAllocate["Float", nn];
   u1t3 = CUDAMemoryAllocate["Float", nn];
   u1dt1 = CUDAMemoryAllocate["Float", nn];
   u1dt2 = CUDAMemoryAllocate["Float", nn];
   u1dt3 = CUDAMemoryAllocate["Float", nn];
   u1d2t1 = CUDAMemoryAllocate["Float", nn];
```

```
u1d2t2 = CUDAMemoryAllocate["Float", nn];
u1d2t3 = CUDAMemoryAllocate["Float", nn];
u1d3t1 = CUDAMemoryAllocate["Float", nn];
u1d3t2 = CUDAMemoryAllocate["Float", nn];
u1d3t3 = CUDAMemoryAllocate["Float", nn];
u2x1 = CUDAMemoryAllocate["Float", nn];
u2x2 = CUDAMemoryAllocate["Float", nn];
u2x3 = CUDAMemoryAllocate["Float", nn];
u2t1 = CUDAMemoryAllocate["Float", nn];
u2t2 = CUDAMemoryAllocate["Float", nn];
u2t3 = CUDAMemoryAllocate["Float", nn];
u2dt1 = CUDAMemoryAllocate["Float", nn];
u2dt2 = CUDAMemoryAllocate["Float", nn];
u2dt3 = CUDAMemoryAllocate["Float", nn];
u2d2t1 = CUDAMemoryAllocate["Float", nn];
u2d2t2 = CUDAMemoryAllocate["Float", nn];
u2d2t3 = CUDAMemoryAllocate["Float", nn];
u2d3t1 = CUDAMemoryAllocate["Float", nn];
u2d3t2 = CUDAMemoryAllocate["Float", nn];
u2d3t3 = CUDAMemoryAllocate["Float", nn];
v1x1 = CUDAMemoryAllocate["Float", nn];
v1x2 = CUDAMemoryAllocate["Float", nn];
v1x3 = CUDAMemoryAllocate["Float", nn];
v1t1 = CUDAMemoryAllocate["Float", nn];
v1t2 = CUDAMemoryAllocate["Float", nn];
v1t3 = CUDAMemoryAllocate["Float", nn];
v1dt1 = CUDAMemoryAllocate["Float", nn];
v1dt2 = CUDAMemoryAllocate["Float", nn];
v1dt3 = CUDAMemoryAllocate["Float", nn];
v1d2t1 = CUDAMemoryAllocate["Float", nn];
v1d2t2 = CUDAMemoryAllocate["Float", nn];
v1d2t3 = CUDAMemoryAllocate["Float", nn];
v1d3t1 = CUDAMemoryAllocate["Float", nn];
v1d3t2 = CUDAMemoryAllocate["Float", nn];
v1d3t3 = CUDAMemoryAllocate["Float", nn];
v2x1 = CUDAMemoryAllocate["Float", nn];
v2x2 = CUDAMemoryAllocate["Float", nn];
v2x3 = CUDAMemoryAllocate["Float", nn];
v2t1 = CUDAMemoryAllocate["Float", nn];
v2t2 = CUDAMemoryAllocate["Float", nn];
v2t3 = CUDAMemoryAllocate["Float", nn];
v2dt1 = CUDAMemoryAllocate["Float", nn];
v2dt2 = CUDAMemoryAllocate["Float", nn];
v2dt3 = CUDAMemoryAllocate["Float", nn];
v2d2t1 = CUDAMemoryAllocate["Float", nn];
v2d2t2 = CUDAMemoryAllocate["Float", nn];
v2d2t3 = CUDAMemoryAllocate["Float", nn];
```

```
v2d3t1 = CUDAMemoryAllocate["Float", nn];
v2d3t2 = CUDAMemoryAllocate["Float", nn];
v2d3t3 = CUDAMemoryAllocate["Float", nn];
D2uDs2 = CUDAMemoryAllocate["Float", nn];
D2vDs2 = CUDAMemoryAllocate["Float", nn];
GeoCur = CUDAMemoryAllocate["Float", nn];
Cur = CUDAMemoryAllocate["Float", nn];

CSAllDetails[pu, pv, u1x1, u1x2, u1x3, u1t1, u1t2, u1t3, u1dt1, u1dt2, u1dt3,
 u1d2t1, u1d2t2, u1d2t3, u1d3t1, u1d3t2, u1d3t3, u2x1, u2x2, u2x3,
 u2t1, u2t2, u2t3, u2dt1, u2dt2, u2dt3, u2d2t1, u2d2t2, u2d2t3, u2d3t1,
 u2d3t2, u2d3t3, v1x1, v1x2, v1x3, v1t1, v1t2, v1t3, v1dt1, v1dt2,
 v1dt3, v1d2t1, v1d2t2, v1d2t3, v1d3t1, v1d3t2, v1d3t3, v2x1, v2x2,
 v2x3, v2t1, v2t2, v2t3, v2dt1, v2dt2, v2dt3, v2d2t1, v2d2t2, v2d2t3,
 v2d3t1, v2d3t2, v2d3t3, D2uDs2, D2vDs2, GeoCur, v, optimum, nn];
CSCurvature[Cur, pu, pv, u1x1, u1x2, u1x3, u1t1, u1t2, u1t3, u1dt1,
 u1dt2, u1dt3, u1d2t1, u1d2t2, u1d2t3, u1d3t1, u1d3t2, u1d3t3, u2x1,
 u2x2, u2x3, u2t1, u2t2, u2t3, u2dt1, u2dt2, u2dt3, u2d2t1, u2d2t2,
 u2d2t3, u2d3t1, u2d3t2, u2d3t3, v1x1, v1x2, v1x3, v1t1, v1t2, v1t3,
 v1dt1, v1dt2, v1dt3, v1d2t1, v1d2t2, v1d2t3, v1d3t1, v1d3t2, v1d3t3,
 v2x1, v2x2, v2x3, v2t1, v2t2, v2t3, v2dt1, v2dt2, v2dt3, v2d2t1, v2d2t2,
 v2d2t3, v2d3t1, v2d3t2, v2d3t3, D2uDs2, D2vDs2, GeoCur, optimum, nn];
CUDAMemoryGet[Cur]];
```

Four

```
1 / CSCurvatureofLoC2[200, 923, 1]
```

```
1 / CSCurvatureofLoC2[400, 975, 1]
```

```
1 / CSCurvatureofLoC2[600, 1040, 1]
```

```
1 / CSCurvatureofLoC2[800, 1112, 1]
```

```
CS2rho200[s_] :=
 {0.16116161978666116`, 0.1737116244827339`, 0.18516717962881307`,
  0.19576831299898864`, 0.20568010597691314`, 0.21502167696139488`,
  0.22388176654144287`, 0.2323288341394275`, 0.2404165358820607`,
  0.2481879274851462`, 0.2556782642350017`, 0.2629166627750215`,
  0.2699275811176882`, 0.2767317590148536`, 0.28334695921302`,
  0.2897885208397703`, 0.2960697437241152`, 0.30220253220738263`,
  0.30819701282187406`, 0.31406256610354405`, 0.3198072694685721`,
  0.3254385167907198`, 0.33096302391480364`, 0.3363866591053325`,
  0.3417149539642946`, 0.34695290881639895`, 0.35210491260724613`,
  0.357175405461356`, 0.3621679838557664`, 0.36708644285410463`,
  0.37193378604166455`, 0.37671322938214885`, 0.3814275870742705`,
  0.3860793592393996`, 0.39067125291782984`, 0.39520530294286005`,
  0.39968370672241377`, 0.4041085826769453`, 0.40848182090607243`,
  0.4128051254200829`, 0.4170802777589119`, 0.4213088053084678`,
  0.4254922174247636`, 0.429631924111646 2`, 0.4337293282563587`,
```

```
0.4377856581834882`, 0.4418021567104241`, 0.4457799632709885`,
0.4497201300898646`, 0.45362374107986636`, 0.45749189575961713`,
0.4613253883475497`, 0.46512507841624523`, 0.468892196708043`,
0.4726272961581927`, 0.476331211765096`, 0.48000452274769284`,
0.48364830270084275`, 0.48726301524706706`, 0.49084949192450783`,
0.49440825189370924`, 0.49794012092633416`, 0.5014455265669157`,
0.5049249487196433`, 0.5083791637344665`, 0.5118087589585971`,
0.5152138919309706`, 0.5185954426504563`, 0.5219536601214182`,
0.5252891025377302`, 0.5286023199851387`, 0.5318935458115595`,
0.5351634580026656`, 0.5384120972455299`, 0.5416401182378989`,
0.5448479745309178`, 0.5480359027655645`, 0.5512041966281732`,
0.5543532442612771`, 0.5574834940224397`, 0.5605952678576381`,
0.5636889031017912`, 0.5667645595674146`, 0.5698224012415087`,
0.5728632207221049`, 0.5758867312627262`, 0.5788936245506533`,
0.5818841274375384`, 0.5848583424888032`, 0.5878167756270204`,
0.5907593150702779`, 0.5936862490842869`, 0.5965982001080598`,
0.5994950256524968`, 0.6023771203477046`, 0.6052446113468357`,
0.6080978731745338`, 0.6109367819482285`, 0.6137618135829956`,
0.6165732098218256`, 0.6193710118605973`, 0.6221553733032513`,
0.6249266576415958`, 0.6276849300913078`, 0.6304305119479267`,
0.6331631308689112`, 0.6358836296014364`, 0.638591680061047`,
0.6412875460098874`, 0.6439714158214642`, 0.6466434483917567`,
0.6493038717703866`, 0.6519527827032086`, 0.6545901917065813`,
0.6572163795470722`, 0.659831492446704`, 0.6624356406870889`,
0.6650288439714248`, 0.6676113993785647`, 0.67018324923761`,
0.6727446682754268`, 0.6752959498623464`, 0.6778368584005438`,
0.680367822945891`, 0.6828884609250937`, 0.6853995075580082`,
0.6879007697156662`, 0.6903922844295644`, 0.6928742127159634`,
0.6953468437578203`, 0.6978100766520587`, 0.7002640491979144`,
0.7027088518311577`, 0.7051448220289837`, 0.7075717169972252`,
0.7099899552335873`, 0.7123991870677497`, 0.7147997908927576`,
0.7171919166667664`, 0.7195753543574558`, 0.7219505780263742`,
0.7243174549667886`, 0.7266761066497479`, 0.7290264100832653`,
0.7313690099526283`, 0.7337036104732535`, 0.7360300427301683`,
0.7383488537014271`, 0.7406596938334042`, 0.7429629994398954`,
0.7452586929884663`, 0.7475468996429537`, 0.7498276853179489`,
0.7521007849595049`, 0.7543666778266713`, 0.756625381498531`,
0.7588769182349484`, 0.7611213150148788`, 0.7633586035736025`,
0.7655888903105996`, 0.7678124286349114`, 0.7700287041635265`,
0.7722386162207147`, 0.7744417301148017`, 0.7766379642524309`,
0.7788276003255092`, 0.7810107844856023`, 0.7831872308280783`,
0.7853573121455768`, 0.7875210431625634`, 0.7896784421078593`,
0.791829381253977`, 0.7939740337819605`, 0.7961127308066084`,
0.798245281815924`, 0.8003714950624914`, 0.8024917917487667`,
0.8046060646287412`, 0.8067144396515454`, 0.8088172040729884`,
0.8109138698465437`, 0.8130045717047037`, 0.8150896873753235`,
0.8171692068383182`, 0.8192430422614001`, 0.8213111868222202`,
```

```
0.8233735551150633`, 0.8254307121411781`, 0.8274820937291307`,
0.8295281110340008`, 0.8315687734258289`, 0.8336040097540898`,
0.835633999311658`, 0.8376585923882045`, 0.8396779746048855`,
0.8416920838315927`, 0.8437012830008324`, 0.8457048355224324`,
0.8477038773072942`, 0.8496976741971759`, 0.8516865176059313`,
0.8536702718765138`, 0.8556493242614694`, 0.8576232818234358`,
0.8595921864757666`, 0.8615564365864214`, 0.8635158172629088`,
0.8654706470616214`, 0.8674203549609839`, 0.8693658027806082`,
0.8713061498870334`, 0.8732419965645394`, 0.8751734075069371`,
0.8770999914575732`, 0.8790222728183427`, 0.8809396764934169`,
0.882852919554964`, 0.8847614265227146`, 0.8866658316461162`,
0.8885656520119395`, 0.8904611539000868`, 0.8923523246922842`,
0.8942390575053818`, 0.8961217235654172`, 0.8980000286893853`,
0.899874061090026`, 0.9017441055493016`, 0.903610063426933`,
0.905471640110393`, 0.90732922425621`, 0.9091827193669738`,
0.9110320284697508`, 0.9128773521510861`, 0.914718695863575`,
0.9165559660649755`, 0.9183893704025698`, 0.9202187171075545`,
0.9220442177629694`, 0.9238659865855307`, 0.9256835276800475`,
0.9274975689229016`, 0.9293077181454897`, 0.9311138880200257`,
0.9329165095792727`, 0.934715292407181`, 0.9365102561540553`,
0.9383016315958785`, 0.9400892325223092`, 0.941873293764348`,
0.9436536299739726`, 0.9454302669486072`, 0.9472034456375342`,
0.9489729820230846`, 0.950739121155937`, 0.9525015717748412`,
0.9542604732961804`, 0.9560160769553118`, 0.9577677636316451`,
0.9595164424239783`, 0.961261604519023`, 0.9630031766010403`,
0.9647413069457145`, 0.9664762580677867`, 0.9682080734713711`,
0.9699363493608284`, 0.9716612395960554`, 0.9733832397829146`,
0.9751016064064694`, 0.9768168372944647`, 0.9785287531055071`,
0.980237287169077`, 0.9819427173704759`, 0.9836450954282345`,
0.9853442429080442`, 0.9870402117958523`, 0.988732938874614`,
0.9904227114811874`, 0.9921092348054664`, 0.9937930347360461`,
0.995473345427355`, 0.9971506950663637`, 0.9988251438505205`,
1.0004963356853669`, 1.0021648680656214`, 1.0038300256491812`,
1.005492530372052`, 1.0071516027893723`, 1.008807970159824`,
1.0104617612873998`, 1.0121121912854745`, 1.01375999690625`,
1.0154045730202181`, 1.0170468484774895`, 1.0186857235010327`,
1.02032213229437`, 1.0219552785652681`, 1.0235860424122911`,
1.0252138121049323`, 1.0268387227238192`, 1.0284609745861781`,
1.030080454774328`, 1.0316973031581695`, 1.0333112799529844`,
1.0349224620067072`, 1.0365311837290732`, 1.0381370775007586`,
1.039740287040728`, 1.0413409582733464`, 1.0429387854970402`,
1.044534109978058`, 1.046126625272752`, 1.047716806736176`,
1.049304217126686`, 1.0508892700055998`, 1.0524714599912928`,
1.0540512030962834`, 1.0556283229417227`, 1.057202841259858`,
1.0587750476874926`, 1.0603445655180792`, 1.0619115516349031`,
1.0634764347951926`, 1.065038635005859`, 1.0665979726316912`,
1.0681556264311156`, 1.069710333924237`, 1.0712625282849382`,
```

1.0728123736894315`, 1.0743599677510451`, 1.075905271534486`,
1.0774478999926467`, 1.0789884354791939`, 1.0805264234803993`,
1.0820619629062043`, 1.083595224104397`, 1.0851260989290847`,
1.0866546894598896`, 1.0881810992310186`, 1.0897052208984026`,
1.0912268752248833`, 1.0927463794815653`, 1.094263626543988`,
1.095778508374463`, 1.097291274857886`, 1.0988016032066754`,
1.1003098889742131`, 1.1018158087418966`, 1.1033198339458279`,
1.1048213501128974`, 1.1063207576719227`, 1.1078179496288392`,
1.1093129647837174`, 1.110805989707648`, 1.1122966232880767`,
1.1137851258191878`, 1.115271834786341`, 1.1167559764845918`,
1.118238483464987`, 1.1197185049103933`, 1.1211962302041443`,
1.122672226623096`, 1.124146239110204`, 1.1256179346193271`,
1.1270874311517796`, 1.1285550759334844`, 1.1300207633674575`,
1.1314843106616526`, 1.132945686363317`, 1.1344052424439657`,
1.1358627195183684`, 1.1373180870362185`, 1.1387714689460537`,
1.1402234556379474`, 1.1416727791681323`, 1.1431205733775913`,
1.1445660317478077`, 1.1460095922848597`, 1.1474514622038035`,
1.1488909855710134`, 1.1503289197620084`, 1.1517649229795872`,
1.1531989672551248`, 1.1546310245368838`, 1.156061146351102`,
1.1574892252159268`, 1.1589158732681677`, 1.1603401840959018`,
1.1617630123987885`, 1.163183689208974`, 1.1646028326480264`,
1.1660198514808546`, 1.167435123709163`, 1.1688484611695384`,
1.1702600005649988`, 1.1716699619241846`, 1.1730778293591932`,
1.174483904380306`, 1.175888244324802`, 1.1772909898576183`,
1.1786920347171868`, 1.1800907737460933`, 1.1814882602691394`,
1.1828835561772562`, 1.1842774701612084`, 1.1856692266436524`,
1.1870597226913624`, 1.188448096045941`, 1.1898349954072407`,
1.1912198070102493`, 1.1926033527098376`, 1.1939849326549785`,
1.195364947836604`, 1.196743290919648`, 1.1981198537194262`,
1.1994947844776491`, 1.2008678031710176`, 1.2022394888702734`,
1.2036091308686578`, 1.204977396819797`, 1.206343659493483`,
1.2077087640288968`, 1.2090719078277767`, 1.2104335911090083`,
1.2117936185373073`, 1.213151880870988`, 1.2145087076192447`,
1.2158636383401216`, 1.2172172683448141`, 1.21856931330868`,
1.219919752607502`, 1.221268387761588`, 1.2226155532448157`,
1.2239610508610295`, 1.2253052177231323`, 1.2266476765499599`,
1.2279885861551687`, 1.2293281068693476`, 1.230665858899011`,
1.232001911608801`, 1.233336788425588`, 1.2346698362275004`,
1.236001489046937`, 1.2373318197718168`, 1.2386599875227489`,
1.2399870688833254`, 1.2413125884803946`, 1.2426365271626316`,
1.2439591424333045`, 1.2452801395182713`, 1.2465995917489054`,
1.2479177588378843`, 1.2492344377117555`, 1.2505496101111266`,
1.2518628840901969`, 1.2531749877948288`, 1.2544859989814396`,
1.255795150164654`, 1.2571030804510888`, 1.2584094904921062`,
1.2597143623008094`, 1.2610174882842333`, 1.2623198939415143`,
1.263620709887055`, 1.2649198225923217`, 1.2662178822199321`,
1.2675143952484649`, 1.2688094399356507`, 1.2701029991290251`,

1.2713954410257666`, 1.2726861714797562`, 1.2739757524766158`,
1.2752638782451464`, 1.2765505320172892`, 1.2778360862906413`,
1.2791201364783868`, 1.2804026656942742`, 1.2816838528401113`,
1.282963976453475`, 1.2842427275534454`, 1.2855198932734162`,
1.2867958513198903`, 1.2880703891424212`, 1.2893439861988096`,
1.2906156365798702`, 1.2918861171271385`, 1.2931553138387497`,
1.2944233116333572`, 1.2956899960458679`, 1.296954951011727`,
1.2982190632313608`, 1.2994819179384287`, 1.3007432988974565`,
1.3020033924446284`, 1.3032621839695921`, 1.3045198616948386`,
1.3057761072277672`, 1.3070310070942739`, 1.308284750588475`,
1.309537221888886`, 1.3107884067699782`, 1.3120383935795485`,
1.3132871686129581`, 1.3145345121394059`, 1.3157808219865659`,
1.31702598213807`, 1.318269668598171`, 1.319512385222501`,
1.3207534960312959`, 1.3219936103545509`, 1.3232324028997953`,
1.3244698596792104`, 1.3257065952046498`, 1.326941759340614`,
1.3281756522737302`, 1.3294084711634913`, 1.3306400980934903`,
1.3318703086053894`, 1.3330994062829755`, 1.3343272725363957`,
1.3355541069173043`, 1.3367796844612696`, 1.3380038853212168`,
1.3392272302571895`, 1.3404493871220242`, 1.3416702359844168`,
1.3428898710811905`, 1.3441084950857392`, 1.3453255570555993`,
1.3465419070759197`, 1.347757318761429`, 1.3489709133550019`,
1.3501840876693332`, 1.3513956361365163`, 1.3526063074645702`,
1.3538154356212442`, 1.355023663139817`, 1.3562309794024818`,
1.3574370442769437`, 1.358641735313889`, 1.3598455905768947`,
1.3610482686872798`, 1.3622498683400057`, 1.363450267375681`,
1.3646495648298618`, 1.3658477492988081`, 1.367044920744275`,
1.368240956714085`, 1.369435510392418`, 1.3706293523477762`,
1.3718222490563272`, 1.3730135159520642`, 1.3742041521612742`,
1.3753938107501502`, 1.3765822553717975`, 1.377769248096856`,
1.3789557961466503`, 1.3801409836542968`, 1.3813249120555995`,
1.3825082531639394`, 1.3836899716634632`, 1.3848711971631018`,
1.386051120396863`, 1.3872299589525123`, 1.3884078170565983`,
1.389584569506826`, 1.3907602055881363`, 1.3919347145641356`,
1.3931083170327283`, 1.394281466874586`, 1.3954527639872856`,
1.3966231241416898`, 1.3977925376652207`, 1.3989609948646777`,
1.4001284860263519`, 1.4012952354988442`, 1.4024605312796834`,
1.4036247135498783`, 1.4047882424330014`, 1.4059506385860059`,
1.4071121276205836`, 1.4082722274278743`, 1.4094319920225344`,
1.4105902290243721`, 1.4117478765362144`, 1.412904332019906`,
1.4140600616743748`, 1.415214102257835`, 1.4163676365892428`,
1.4175200587447492`, 1.4186713586987865`, 1.4198218868772725`,
1.4209712739156415`, 1.4221201124627638`, 1.4232676708957512`,
1.4244143013192196`, 1.425560115895687`, 1.4267044996015534`,
1.427848048693783`, 1.4289911198761358`, 1.4301330969045267`,
1.4312738480364535`, 1.432413974518096`, 1.4335528565503473`,
1.4346910972906777`, 1.4358279520788642`, 1.436964025400284`,
1.43809943253363`, 1.4392337956602559`, 1.440367105587929`,

```
1.4414993531078917`, 1.4426306530431532`, 1.4437612452209425`,
1.4448908732593733`, 1.4460197777310821`, 1.447147576617529`,
1.4482743856329439`, 1.4494003212535402`, 1.4505252498983254`,
1.451649539608834`, 1.4527725543563912`, 1.4538950401300021`,
1.4550166118834569`, 1.4561372613671544`, 1.4572566005873249`,
1.4583752533673804`, 1.459492958648261`, 1.4606099624383349`,
1.4617257481631518`, 1.4628409440569288`, 1.4639550325435011`,
1.4650682605435397`, 1.4661806202553693`, 1.467292488839555`,
1.4684029605745401`, 1.469512926283114`, 1.4706218642199687`,
1.4717302823401481`, 1.4728373987201395`, 1.47394385116706`,
1.4750489845002719`, 1.4761535683223652`, 1.4772573362884556`,
1.478360281023181`, 1.4794620037479456`, 1.480563279255861`,
1.4816637093238783`, 1.4827630244299672`, 1.483861609685652`,
1.4849594582167245`, 1.4860561682494728`, 1.4871522584266297`,
1.488247194348193`, 1.4893421572413303`, 1.4904355547943222`,
1.4915281721569582`, 1.4926201351570283`, 1.4937109054449633`,
1.4948011401932926`, 1.4958904334326182`, 1.496978911133848`,
1.4980664325021138`, 1.4991532578306672`, 1.5002389781873944`,
1.5013245257716303`, 1.5024086855423078`, 1.5034925259630738`,
1.5045749628210565`, 1.5056572027857307`, 1.5067384297323974`,
1.5078187714504387`, 1.5088979493749985`, 1.5099766346791905`,
1.5110542775561207`, 1.5121314153897791`, 1.513207769762318`,
1.514283060691579`, 1.515357964726223`, 1.5164316545722172`,
1.5175046708548785`, 1.5185768700286189`, 1.519648383054069`,
1.5207189281340165`, 1.5217884979777172`, 1.5228577764275326`,
1.523926205389036`, 1.5249935008862427`, 1.5260600714347372`,
1.5271260498571972`, 1.528191291343728`, 1.5292556502976344`,
1.5303191199223147`, 1.5313818331911535`, 1.5324439238426188`,
1.5335052460432108`, 1.5345656531730534`, 1.535625560051733`,
1.5366843985630074`, 1.5377425840873693`, 1.5388001109073814`,
1.53985626663502`, 1.540912599409029`, 1.54196712289385`,
1.5430218134692104`, 1.544075105673494`, 1.545127844628284`,
1.546179740136031`, 1.5472309285055161`, 1.548281403928101`,
1.549331446738527`, 1.5503801926486847`, 1.5514286377444588`,
1.552476203290056`, 1.5535231705502395`, 1.5545692461478646`,
1.555614279385358`, 1.5566588409466242`, 1.5577024922176228`,
1.5587455163532384`, 1.5597880530345283`, 1.5608293714591182`,
1.5618703363614876`, 1.5629103611602073`, 1.5639495851117122`,
1.5649882942211006`, 1.5660263375038737`, 1.5670634168211728`,
1.5681001118601687`, 1.5691358313357358`, 1.5701707156122113`,
1.571204758674691`, 1.572238543854273`, 1.5732713297802259`,
1.5743034052822422`, 1.575334616971479`, 1.5763651069024303`,
1.5773953145089448`, 1.5784244935361353`, 1.57945293470665`,
1.580481079171031`, 1.5815080286634153`, 1.5825343730114816`,
1.583559957789897`, 1.584584927114471`, 1.5856091261560423`,
1.5866325493519402`, 1.5876553413736785`, 1.5886774972226674`,
1.5896987106332674`, 1.5907194278983015`, 1.591739644575195`,
```

```
1.5927590537934937`, 1.593777649933531`, 1.5947955789645587`,
1.5958128358314294`, 1.5968292634873325`, 1.5978447041039614`,
1.5988599132946297`, 1.59987427745775`, 1.600888096433349`,
1.6019013656995293`, 1.6029136212941997`, 1.6039251636146081`,
1.6049361410664935`, 1.6059462416488663`, 1.606955921393981`,
1.6079647139815798`, 1.6089729222847409`, 1.6099802327258503`,
1.6109875676993517`, 1.6119933760867509`, 1.612998890422571`,
1.614003796712985`, 1.6150077794107016`, 1.6160111440435243`,
1.617014041827664`, 1.6180163126580251`, 1.6190176394125406`,
1.6200184855663107`, 1.6210185337898833`, 1.62201793565031`,
1.6230162154423018`, 1.6240146243649394`, 1.6250119014583464`,
1.6260086706423587`, 1.6270047700097336`, 1.6280000368738183`,
1.6289949404420048`, 1.6299893187865193`, 1.630982533536446`,
1.6319753719536916`, 1.6329675125699992`, 1.633958791499628`,
1.6349496813852797`, 1.6359397000604559`, 1.6369290018871714`,
1.6379180618046663`, 1.6389063964685964`, 1.6398938407955996`,
1.6408807103786496`, 1.6418668403145784`, 1.6428523866635436`,
1.643837345187387`, 1.6448215503848913`, 1.6458054817752221`,
1.646788004760969`, 1.6477705687038409`, 1.6487520374507676`,
1.6497332160956508`, 1.650713614271459`, 1.6516932271112448`,
1.6526722125417264`, 1.6536505662623158`, 1.6546284471516388`,
1.6556056880972592`, 1.6565824483513232`, 1.6575582328761778`,
1.6585332001409692`, 1.659508001827542`, 1.6604823066441894`,
1.6614556172163286`, 1.6624279280332652`, 1.663399893259029`,
1.6643715096836762`, 1.665342443480631`, 1.666312524774124`,
1.6672820799843857`, 1.6682511053003206`, 1.669219264753054`,
1.6701870521702495`, 1.6711541313790934`, 1.6721203313141637`,
1.6730859806733485`, 1.6740512426504888`, 1.675015445046982`,
1.6759795868174618`, 1.6769426601343675`, 1.6779053306337044`,
1.6788672590175535`, 1.6798289454120194`, 1.6807898819294622`,
1.6817498955247505`, 1.682709487462737`, 1.6836683164938564`,
1.6846267164480488`, 1.6855841759471228`, 1.6865408594623004`,
1.6874977808940534`, 1.688453239947784`, 1.6894085907490841`,
1.6903636607826271`, 1.691317594748908`, 1.6922710697492764`,
1.6932233988612546`, 1.694175945030924`, 1.6951271657377958`,
1.6960779120262781`, 1.697028180501107`, 1.6979781396099185`,
1.6989274424438399`, 1.699875740551459`, 1.7008242357591639`,
1.7017712007866368`, 1.7027180112201146`, 1.703664145779863`,
1.7046090808762941`, 1.7055543712286432`, 1.706498627657921`,
1.7074423662931177`, 1.7083854097190472`, 1.709327579747811`,
1.710269917833277`, 1.7112110265155247`, 1.712151598933269`,
1.7130916316086608`, 1.7140307708339613`, 1.7149695378841676`,
1.7159082808358617`, 1.7168455924135988`, 1.7177828732791804`,
1.7187194173660238`, 1.719655396852242`, 1.720590102401329`,
1.721524588076392`, 1.722459028343157`, 1.7233923592153875`,
1.724325284324495`, 1.725257623294637`, 1.7261893725844901`,
1.7271207064457617`, 1.7280510879361402`, 1.7289808687128156`,
```

```
    1.7299102235893324`, 1.730839149577468`, 1.7317674649292443`,
    1.7326948081790983`, 1.7336217120174904`, 1.734547994109002`,
    1.7354741894337185`, 1.7363993975619307`, 1.7373239735609558`,
    1.7382479138340767`, 1.739171214780474`, 1.7400940532737608`,
    1.7410167876189222`, 1.741938511618519`, 1.7428592205072375`,
    1.7437799969795795`, 1.7447002954003608`, 1.7456193862078804`,
    1.7465383549542357`, 1.747456107669491`, 1.7483737321926516`,
    1.7492915913823397`, 1.7502078586315108`, 1.7511236240621766`,
    1.7520388845910284`, 1.752953453974933`, 1.753868061937497`,
    1.7547812387987314`, 1.755694632155996`, 1.7566067698378471`,
    1.7575187511123807`, 1.7579461769642335`}[[s]];
CS2rho400[s_] := {0.20324422894127608`, 0.2156094004231272`,
    0.22688585331296954`, 0.23731629547709282`,
    0.2470673336333584`, 0.2562582446213955`, 0.264977872097616`,
    0.2732943051653832`, 0.28126088013119727`, 0.288920183628051`,
    0.2963071390150572`, 0.30345023375333136`, 0.31037361568526695`,
    0.3170974666096527`, 0.32363916277823696`, 0.33001366066259485`,
    0.3362338953713392`, 0.342311293968592`, 0.34825585368269646`,
    0.3540763354860582`, 0.3597807949481311`, 0.3653762375090499`,
    0.37086899873954354`, 0.3762649466785812`, 0.38156919647933774`,
    0.3867866153127995`, 0.39192139788996366`, 0.3969777737899894`,
    0.4019590874790865`, 0.40686885309034465`, 0.411710296713241`,
    0.41648621728141594`, 0.4211992684477733`, 0.42585199253625117`,
    0.43044678143252446`, 0.4349856161284378`, 0.4394708416933091`,
    0.44390407456201253`, 0.44828742354201984`, 0.4526224244666883`,
    0.45691049364761793`, 0.4611534421388932`, 0.4653524500850255`,
    0.4695090840176766`, 0.4736243981318374`, 0.47769989933039164`,
    0.48173645940866255`, 0.4857352536358308`, 0.48969742651947856`,
    0.4936238402328173`, 0.4975155678822295`, 0.5013734473148073`,
    0.5051982931220503`, 0.5089910378441508`, 0.5127523852596452`,
    0.5164831646572815`, 0.5201841372948466`, 0.523855917091384`,
    0.527499151082213`, 0.5311145480649154`, 0.5347026712798534`,
    0.5382641635732195`, 0.5417996416682705`, 0.5453096884913412`,
    0.5487945581948297`, 0.5522551318051476`, 0.5556916640897903`,
    0.5591048199567025`, 0.5624948661311641`, 0.565862482447932`,
    0.5692077789619958`, 0.5725314697151235`, 0.5758338379866003`,
    0.5791151874514807`, 0.5823760024048692`, 0.5856167099494569`,
    0.5888375912279386`, 0.5920390632768935`, 0.5952214307325433`,
    0.5983849627655765`, 0.6015299322632307`, 0.6046567448542849`,
    0.6077657237296403`, 0.6108570144029615`, 0.6139310681787343`,
    0.6169881114831715`, 0.6200281786373644`, 0.6230518877408423`,
    0.6260594399905665`, 0.6290509750070845`, 0.6320267571752591`,
    0.634986892043929`, 0.6379319483628931`, 0.640861705528124`,
    0.6437767463020438`, 0.6466770470801965`, 0.6495630054065155`,
    0.6524346461553012`, 0.6552923717794437`, 0.6581361003382392`,
    0.6609662850469739`, 0.663782991295166`, 0.6665864087415343`,
    0.6693766422996544`, 0.6721540310253364`, 0.6749186141581355`,
```

```
0.6776707201708757`, 0.6804102605698064`, 0.6831374911631397`,
0.6858525769116305`, 0.6885556441587618`, 0.6912470063996692`,
0.6939265970198394`, 0.6965946480861769`, 0.6992513518046346`,
0.7018966822389752`, 0.7045309783032281`, 0.7071542417707181`,
0.7097666045680002`, 0.7123682727985908`, 0.7149591634322677`,
0.7175396912767167`, 0.7201099212589233`, 0.7226697420691925`,
0.7252192358327089`, 0.7277590004848772`, 0.7302887051730519`,
0.732808529734294`, 0.7353186022560934`, 0.7378189320572371`,
0.7403099933722876`, 0.7427915568749549`, 0.7452636587785616`,
0.7477268096253311`, 0.7501806679361341`, 0.7526256999279548`,
0.755061506769588`, 0.7574887798108655`, 0.7599069263014365`,
0.7623167989158947`, 0.7647176743547068`, 0.7671100792677831`,
0.7694942767278059`, 0.7718701154896423`, 0.7742377315334196`,
0.7765969816748246`, 0.7789483028644982`, 0.7812915653915903`,
0.7836268607220092`, 0.7859542871598356`, 0.7882740239590496`,
0.79058596099135`, 0.792890289631622`, 0.7951868330851477`,
0.7974760211264492`, 0.7997576113044047`, 0.8020319730181886`,
0.804298948178253`, 0.8065587666429627`, 0.808811277259366`,
0.8110567996401752`, 0.8132950332876107`, 0.8155264673989022`,
0.8177508082112686`, 0.819968400074914`, 0.8221791952186119`,
0.8243830670153826`, 0.8265804594303354`, 0.8287710920935258`,
0.8309550932428872`, 0.8331327623192266`, 0.8353036593665873`,
0.8374682577809618`, 0.839626455364647`, 0.8417783194774769`,
0.8439240913709986`, 0.8460633371894941`, 0.8481963887329503`,
0.8503235009218266`, 0.8524445031578198`, 0.8545592238540436`,
0.8566681903389374`, 0.8587712394319826`, 0.860868383708477`,
0.8629595494912271`, 0.8650449315371924`, 0.8671248202278075`,
0.86919870309369`, 0.8712669648715208`, 0.8733296360755194`,
0.8753867498300331`, 0.8774382501040758`, 0.8794841739969176`,
0.8815249318018386`, 0.8835600126857719`, 0.8855897396430301`,
0.8876141617315136`, 0.8896332363588852`, 0.8916471114821433`,
0.893655749082892`, 0.89565901653347`, 0.8976573568753344`,
0.8996504508597271`, 0.9016384597244678`, 0.9036214518489666`,
0.9055994006290587`, 0.9075723787362159`, 0.9095405602593629`,
0.9115037276257429`, 0.9134618599057094`, 0.9154154367709085`,
0.9173640446479384`, 0.9193078687729433`, 0.9212468965905024`,
0.9231814214950977`, 0.9251110289879094`, 0.9270362209722655`,
0.9289566860487972`, 0.9308725220977425`, 0.9327837261719831`,
0.9346906089678557`, 0.9365929646288723`, 0.938490690115449`,
0.9403841035934911`, 0.9422732120178978`, 0.9441577047780619`,
0.9460380142581098`, 0.9479139381839475`, 0.9497854872245675`,
0.9516527812725403`, 0.9535160515657024`, 0.9553748811562432`,
0.9572295029887861`, 0.9590800450236465`, 0.960926528126714`,
0.9627687534632143`, 0.9646068528808641`, 0.9664408498809548`,
0.9682711045697399`, 0.9700970859415042`, 0.971919269347647`,
0.9737372339741409`, 0.9755516881065865`, 0.977361871132503`,
0.9791683835111031`, 0.9809706883159984`, 0.9827693929120844`,
```

```
0.9845640744195742`, 0.9863551164286185`, 0.9881422105452623`,
0.9899255119702326`, 0.991705178955054`, 0.9934812551443984`,
0.9952533132367164`, 0.997021987091474`, 0.9987869689893647`,
1.0005483652483498`, 1.0023060448964238`, 1.0040601762854895`,
1.0058106896450014`, 1.007557575435826`, 1.0093008854024197`,
1.0110409773362687`, 1.0127774198395212`, 1.0145103283465122`,
1.0162398205451524`, 1.0179658928478033`, 1.019688542250707`,
1.021408139441085`, 1.0231240624253148`, 1.0248366204711183`,
1.026546002001 4239`, 1.0282518313642128`, 1.0299543607848622`,
1.0316537195597046`, 1.0333496573828178`, 1.0350424320599048`,
1.0367317928369966`, 1.0384182575062153`, 1.0401013839363595`,
1.041781306981869`, 1.0434580339916524`, 1.0451318333638888`,
1.0468022589972703`, 1.0484697113830086`, 1.0501338736014878`,
1.0517951505325989`, 1.0534531589216953`, 1.0551083737748603`,
1.0567605443686008`, 1.0584096176065816`, 1.060055540205297`,
1.061698729002816`, 1.0633387977397448`, 1.0649758290352713`,
1.0666099748014493`, 1.068241049179142`, 1.0698694102721318`,
1.0714946682378568`, 1.073117252589117`, 1.0747369788550447`,
1.0763536607390027`, 1.0779675949520642`, 1.0795788737814997`,
1.0811871032801044`, 1.0827925840279662`, 1.0843952701403452`,
1.0859952561707291`, 1.0875927086783945`, 1.0891870187354376`,
1.090778705324696`, 1.0923674404077512`, 1.0939538197940821`,
1.095537373524591`, 1.0971181287461391`, 1.0986962571920718`,
1.1002719329339212`, 1.1018448981993667`, 1.1034150377413463`,
1.1049828175638698`, 1.1065477605376262`, 1.1081101163398268`,
1.1096701378163862`, 1.1112274921706067`, 1.1127822122159854`,
1.1143341834364264`, 1.115883884122501`, 1.1174309795275053`,
1.1189755792743339`, 1.120517719673946`, 1.122057287653877`,
1.1235944701942226`, 1.1251290038991049`, 1.126661303728518`,
1.1281909549394484`, 1.1297184514174798`, 1.1312432248155306`,
1.1327656951892815`, 1.1342857506012443`, 1.1358033550753686`,
1.137318703821808`, 1.1388315305856194`, 1.1403422641208572`,
1.1418504054343674`, 1.143356385532874`, 1.1448598590019206`,
1.1463612603347644`, 1.1478600864349342`, 1.1493567740545896`,
1.1508512121578591`, 1.1523434470883218`, 1.1538330497903808`,
1.1553207008327402`, 1.1568058118521665`, 1.1582889881736882`,
1.1597697191667988`, 1.1612481322121115`, 1.1627245173979175`,
1.164198925636632`, 1.1656707606471783`, 1.1671405570591302`,
1.1686083667436882`, 1.1700739158892486`, 1.1715371735683613`,
1.1729982727915447`, 1.1744573481182707`, 1.1759140411384486`,
1.1773688163318468`, 1.1788215631983068`, 1.1802720873079153`,
1.1817206919326855`, 1.1831671828896073`, 1.1846114479286054`,
1.1860537931423545`, 1.1874940235002869`, 1.1889320265194263`,
1.1903682800347009`, 1.1918022507489248`, 1.1932339944624186`,
1.1946641631996906`, 1.1960922216080756`, 1.1975181422340875`,
1.1989420689113044`, 1.2003641470290851`, 1.2017842652742186`,
1.2032024840285658`, 1.2046186049393681`, 1.2060324279592343`,
```

```
1.2074444465395506`, 1.2088549847351329`, 1.2102633221210304`,
1.2116696067891335`, 1.213074076101288`, 1.2144767939615286`,
1.215877736903592`, 1.2172764397970337`, 1.2186732300955023`,
1.2200683497095421`, 1.2214613314943819`, 1.2228526846289172`,
1.2242419404258573`, 1.2256295214459394`, 1.2270152260704577`,
1.2283990308827417`, 1.2297810025507163`, 1.2311613891859674`,
1.2325397168816978`, 1.233916505886457`, 1.2352915536214333`,
1.2366644735203007`, 1.2380354238720588`, 1.2394050223292448`,
1.2407726074777607`, 1.2421383398679813`, 1.2435026581129924`,
1.2448649887583692`, 1.2462254940120738`, 1.247584337645501`,
1.2489414990440066`, 1.2502967711876072`, 1.2516503189029964`,
1.2530024021607198`, 1.2543526268282064`, 1.2557012533798415`,
1.2570479795482268`, 1.2583931613058947`, 1.2597364956814954`,
1.2610783407678896`, 1.2624184878833615`, 1.2637567266854877`,
1.2650934175421114`, 1.26642835041924`, 1.267761888411187`,
1.269093725793958`, 1.2704240353833187`, 1.2717526059525623`,
1.2730796110316465`, 1.2744048384982134`, 1.2757283655922613`,
1.277050270314501`, 1.2783708262419475`, 1.2796895284499763`,
1.281006748238648`, 1.282322566049042`, 1.283636473811766`,
1.2849491403504525`, 1.2862599580937808`, 1.287569203939142`,
1.2888769591222997`, 1.2901831072087833`, 1.2914879285884189`,
1.2927908094598675`, 1.2940924288044282`, 1.2953926710302444`,
1.2966910188241527`, 1.2979880554452015`, 1.2992834633400316`,
1.300577426617005`, 1.3018696258412634`, 1.3031605489127096`,
1.304450079166357`, 1.3057378958519132`, 1.3070244903647819`,
1.3083095418807675`, 1.3095928293592929`, 1.3108747446671243`,
1.3121551697168778`, 1.3134341912337753`, 1.314711587634269`,
1.3159880647215887`, 1.3172624722194437`, 1.3185359307614473`,
1.3198077009005131`, 1.3210779737789438`, 1.3223468378804102`,
1.3236143824547022`, 1.3248804882686767`, 1.3261451400809225`,
1.327408217593087`, 1.3286702310187009`, 1.3299307457211018`,
1.3311895350822682`, 1.3324473232623946`, 1.333703461685833`,
1.3349582529546622`, 1.3362115763477747`, 1.3374635235575736`,
1.338714293864631`, 1.3399637670539282`, 1.3412118220819924`,
1.3424581222391725`, 1.3437031893418165`, 1.3449470100746674`,
1.3461895710928509`, 1.3474306425893272`, 1.3486705352120103`,
1.349909236126315`, 1.3511461883985898`, 1.3523820302284164`,
1.353616640080656`, 1.3548496767727392`, 1.3560813447575606`,
1.3573118500528292`, 1.3585411802240346`, 1.3597691023945553`,
1.3609958236940505`, 1.3622211103960564`, 1.3634451703723773`,
1.3646676580450585`, 1.3658888926339017`, 1.3671091957179402`,
1.3683281100416227`, 1.3695457341326125`, 1.3707620552914026`,
1.3719771729881722`, 1.373190737887561`, 1.3744034115764614`,
1.37561473247438`, 1.3768248007458082`, 1.3780332645735593`,
1.379241016983346`, 1.380447480726068`, 1.3816524159088346`,
1.382855922972529`, 1.384058674026659`, 1.3852599733800999`,
1.3864602663222254`, 1.387658968320229`, 1.38885664058141`,
```

```
1.3900532723994585`, 1.3912485069376608`, 1.3924424472277996`,
1.3936350813065164`, 1.3948267450781382`, 1.3960170798331424`,
1.3972060735417116`, 1.3983944134958286`, 1.3995811571434864`,
1.4007669929208617`, 1.4019513253312472`, 1.4031347283877422`,
1.4043168396656938`, 1.4054975298344992`, 1.4066776122891917`,
1.4078560157021294`, 1.4090337916840445`, 1.4102096282363084`,
1.4113850542329314`, 1.412558754472126`, 1.4137315493428682`,
1.4149034294668716`, 1.4160739073531414`, 1.4172433303750438`,
1.418411688501967`, 1.419578731452618`, 1.4207449292244918`,
1.4219093088306236`, 1.4230730636866407`, 1.4242358233185923`,
1.4253973359677823`, 1.4265579547757106`, 1.4277170632586165`,
1.428874892827246`, 1.4300322859066281`, 1.4311878928369102`,
1.432342800971457`, 1.4334962675064236`, 1.4346490170424908`,
1.4358007958805705`, 1.436951348727252`, 1.4381004186955275`,
1.43924873504387`, 1.4403959187619262`, 1.4415419600422121`,
1.442687345291682`, 1.4438310729926944`, 1.444974002085483`,
1.4461158751689749`, 1.4472563084061159`, 1.4483959159162478`,
1.4495346895442003`, 1.450672370248119`, 1.4518089486596533`,
1.4529444153942055`, 1.4540786350262813`, 1.4552119762134843`,
1.456344177866666`, 1.4574752305373486`, 1.458605758814727`,
1.4597348671155472`, 1.4608630535798173`, 1.4619901824749746`,
1.463115989671373`, 1.4642412317935507`, 1.465365134296623`,
1.4664880715750948`, 1.467610035219738`, 1.4687311453836895`,
1.4698507503484028`, 1.4709696131242784`, 1.4720874680877798`,
1.4732043067555256`, 1.4743201206295917`, 1.4754350309514617`,
1.4765488998325362`, 1.4776618488775746`, 1.478773609435683`,
1.479884433425111`, 1.4809944435837714`, 1.482103239701516`,
1.4832113370805222`, 1.4843183348953113`, 1.485423829957152`,
1.4865288655521638`, 1.4876329083181434`, 1.4887356859556184`,
1.4898375860839221`, 1.4909388661616776`, 1.4920387236736639`,
1.4931376802597671`, 1.4942355952671362`, 1.4953329936337978`,
1.4964290686858075`, 1.497524212058195`, 1.4986186838551083`,
1.4997120754364144`, 1.5008041100588632`, 1.5018953161020585`,
1.502985821123429`, 1.5040752140883693`, 1.5051638918190726`,
1.506251712545837`, 1.507337992002236`, 1.5084238050793048`,
1.5095080598433017`, 1.5105921063057066`, 1.5116749867638912`,
1.5127572384551518`, 1.513838035762463`, 1.5149181898725994`,
1.5159974207502378`, 1.5170757211510204`, 1.5181529464429284`,
1.5192290887740132`, 1.520304966875775`, 1.5213796106562487`,
1.5224531498004013`, 1.5235257147615546`, 1.5245972982160978`,
1.5256684477876092`, 1.526738463795279`, 1.5278074774271788`,
1.5288756206392362`, 1.5299431656731024`, 1.5310092686146137`,
1.5320744799419723`, 1.5331392131333061`, 1.5342029019169836`,
1.5352655388614749`, 1.5363276792648433`, 1.5373883318520458`,
1.538448474927244`, 1.5395075382777466`, 1.5405660802698329`,
1.5416233874589904`, 1.5426798773117718`, 1.5437358275705075`,
1.5447906640856954`, 1.5458440944577176`, 1.5468973936262416`,
```

```
1.5479494155429023`, 1.5490005811100993`, 1.5500511703396256`,
1.551100604363957`, 1.552149162787423`, 1.5531969829650452`,
1.5542437710617376`, 1.555289808398772`, 1.5563353778389861`,
1.557379607250573`, 1.558423067448373`, 1.5594654626116067`,
1.5605072211014097`, 1.5615481922266485`, 1.5625880789765083`,
1.5636276028863876`, 1.5646657382810591`, 1.5657032074524908`,
1.566740005005435`, 1.567775979032049`, 1.5688106834473108`,
1.569844844959583`, 1.5708783116342`, 1.5719104889218605`,
1.5729421064064713`, 1.5739727162186339`, 1.5750027553040307`,
1.5760320707171525`, 1.5770603604867102`, 1.5780876179611656`,
1.5791142823715907`, 1.5801399023183407`, 1.5811647691582877`,
1.5821890265753786`, 1.5832122214067883`, 1.5842349453348248`,
1.5852564450233868`, 1.5862771631220693`, 1.5872969438895814`,
1.5883165330100886`, 1.589334572171647`, 1.590352258916669`,
1.5913689855976179`, 1.5923848971683021`, 1.5934001393459252`,
1.594414404037674`, 1.595427988469551`, 1.5964405837623805`,
1.5974526400000153`, 1.598463695800183`, 1.5994738974120928`,
1.6004830864876494`, 1.6014918682641224`, 1.602499626530794`,
1.6035069680498397`, 1.6045129681558437`, 1.6055188485505252`,
1.606523068060469`, 1.6075273123777147`, 1.6085303454829805`,
1.6095327781346582`, 1.610533832726467`, 1.611534585351928`,
1.6125345675979397`, 1.6135339294589783`, 1.6145323555740152`,
1.6155298401669644`, 1.6165265332114025`, 1.6175224294937731`,
1.618517992214582`, 1.6195124362126698`, 1.6205060681505934`,
1.6214987260714346`, 1.6224910317834715`, 1.6234821962604749`,
1.6244726849649955`, 1.6254624932047204`, 1.6264517739554882`,
1.6274402073536032`, 1.628427630053416`, 1.6294141944361324`,
1.630399736760449`, 1.6313850443094693`, 1.632369478771297`,
1.6333527167951611`, 1.6343360255645556`, 1.6353181280044697`,
1.6362991772184676`, 1.6372798064136977`, 1.6382595320440299`,
1.6392386690807923`, 1.6402170529539757`, 1.641194518322409`,
1.6421712205490655`, 1.6431474766840992`, 1.6441226385824566`,
1.6450971842107611`, 1.6460707863136819`, 1.6470434394874194`,
1.6480159477403848`, 1.6489876602230809`, 1.6499582475558519`,
1.6509281911080502`, 1.6518974865921057`, 1.6528659668767047`,
1.6538334640617458`, 1.6548004623557715`, 1.655766958020132`,
1.6567322929079202`, 1.6576974437284817`, 1.658661259996872`,
1.6596245559545053`, 1.6605871634711555`, 1.6615487490958012`,
1.6625101310538961`, 1.6634704820333719`, 1.66442996165521`,
1.665388730296661`, 1.666346783585886`, 1.6673041171456622`,
1.668260892478753`, 1.6692169396922678`, 1.6701720881272473`,
1.6711263328149126`, 1.6720800020730058`, 1.6730330920628032`,
1.67398543191451`, 1.6749368499932014`, 1.6758876761160262`,
1.676837906423315`, 1.6777870336943141`, 1.6787357242510201`,
1.6796836384580829`, 1.6806307718661002`, 1.6815769514766505`,
1.682522678462994`, 1.683467780575438`, 1.6844120848008288`,
1.6853554173629455`, 1.6862987901663018`, 1.687240674437039`,
```

```
1.688182082638654`, 1.6891230114147484`, 1.6900629466553614`,
1.6910020537291675`, 1.6919403281295349`, 1.6928782777967701`,
1.6938152158922788`, 1.694751650914545`, 1.6956874080886997`,
1.6966224834524049`, 1.697556873038708`, 1.6984902289728108`,
1.6994232347072111`, 1.700355025747728`, 1.7012866315191613`,
1.7022171861962383`, 1.7031472033397783`, 1.7040765064331944`,
1.7050050914737631`, 1.7059329544544033`, 1.7068600913637064`,
1.7077864981859647`, 1.7087125189563106`, 1.7096372797011898`,
1.710562169931871`, 1.7114856171056996`, 1.712408488831468`,
1.7133309566027444`, 1.7142524920304174`, 1.7151732658463887`,
1.716093449531525`, 1.7170130395928265`, 1.7179316807119112`,
1.7188497204561588`, 1.7197671553201286`, 1.720683275895914`,
1.7215994897145068`, 1.7225145575606295`, 1.7234295364509848`,
1.724343183967895`, 1.7252562039833912`, 1.7261689481668294`,
1.72708052516078`, 1.7279918196188044`, 1.7289021164634562`,
1.7298119460757633`, 1.7307207698467078`, 1.73162947672889`,
1.7325368124479985`, 1.7334434875975198`, 1.7343502157649857`,
1.7352557392919643`, 1.7361604122440524`, 1.7370649498828477`,
1.7379680900445866`, 1.7388709082464868`, 1.7397732216084996`,
1.74067448530048`, 1.7415750561745966`, 1.7424754735479904`,
1.743374829699193`, 1.7442733012450231`, 1.7451714286130366`,
1.7460688458682436`, 1.7469651855781692`, 1.747861171373826`,
1.7487566184692047`, 1.749651158861733`, 1.7505449710120915`,
1.7514380512843104`, 1.7523309451164104`, 1.7532227344822924`,
1.7541141482116933`, 1.7550042658492304`, 1.7558943684628392`,
1.756783534845479`, 1.7576717607795216`, 1.7585597793616061`,
1.759446850539447`, 1.7603333394888772`, 1.761219058208711`,
1.7621038179509823`, 1.7629883555156802`, 1.7638722980046853`,
1.7647550853997087`, 1.7656374561635129`, 1.7665190357164555`,
1.7674000065524744`, 1.7682805519046951`, 1.7691604826286116`,
1.770038861859725`, 1.7709175529584824`, 1.7717950586384061`,
1.7726719359754262`, 1.7735483692962846`, 1.7744237929853575`,
1.775298578477885`, 1.7761729106469593`, 1.7770465986524617`,
1.7779194509111071`, 1.7787916522871285`, 1.7796633883815793`,
1.7805342786376632`, 1.781404508465783`, 1.7822740746784151`,
1.783143542641952`, 1.7840113931963868`, 1.7848789495935908`,
1.7857458295968898`, 1.7866122202617343`, 1.787477928495624`,
1.7883423792140947`, 1.789206712408853`, 1.7900707355317371`,
1.7909329170449102`, 1.7917955467666822`, 1.7926570924175118`,
1.7935181248617624`, 1.7943780656817125`, 1.7952378710381578`,
1.7960967702255803`, 1.7969549519250851`, 1.7978124129006623`,
1.7986695355810784`, 1.7995253527408273`, 1.8003808254881795`,
1.801235951678498`, 1.802090342028627`, 1.802943993287379`,
1.803796708266598`, 1.8046488713963555`, 1.8055004799683179`,
1.8063515312715401`, 1.807201827924975`, 1.808051561513752`,
1.8089007293192865`, 1.8097487429705537`, 1.8105959888967658`,
1.811443441734938`, 1.8122893390646322`, 1.8131350470482872`,
```

```
    1.8139799756574957`, 1.8148241216086336`, 1.815667874607221`,
    1.8165110357823273`, 1.8173536023910724`, 1.8181949805589916`,
    1.8190357575735892`, 1.8198761280928113`, 1.820716089911418`,
    1.8215552452776107`, 1.822393194980092`, 1.8232309253019616`,
    1.8240680379382133`, 1.8249045301254934`, 1.825740001734632`,
    1.8265748466342495`, 1.8274092610998407`, 1.828243442131506`,
    1.8290763910019365`, 1.8299085024807118`, 1.8307403725596536`,
    1.8315709997823146`, 1.832366757841494`}[[s]];
CS2rho600[s_] := {0.2528654957362938`, 0.2645524357716837`,
    0.275249795365888`, 0.28517383702687366`, 0.2944755159670478`,
    0.30326389233335904`, 0.31162095492343994`, 0.3196092911017362`,
    0.32727821485404557`, 0.33466719743558504`, 0.3418081596097443`,
    0.3487274739335961`, 0.35544705159227297`, 0.36198563295642894`,
    0.36835881344364796`, 0.3745802576039461`, 0.38066175163956756`,
    0.3866136655352077`, 0.39244483363800026`, 0.39816337906659194`,
    0.40377643188971096`, 0.4092903143207659`, 0.4147107839235579`,
    0.42004286450072853`, 0.42529129683251926`, 0.4304602111277907`,
    0.4355536826696986`, 0.4405753282463211`, 0.4455280048335329`,
    0.45041503585608683`, 0.4552392459422715`, 0.4600029853060865`,
    0.4647088412370951`, 0.4693589250064904`, 0.47395530882436526`,
    0.4784999236212495`, 0.482994839216062`, 0.48744144792854727`,
    0.4918415221825674`, 0.4961965751904552`, 0.5005079298027051`,
    0.5047770119017768`, 0.5090051830058958`, 0.5131934315757121`,
    0.5173429617440027`, 0.5214550001659732`, 0.525530452526733`,
    0.5295703985290272`, 0.5335757869815545`, 0.5375474132694099`,
    0.541486141334939`, 0.5453927921826978`, 0.5492683467900218`,
    0.5531131094328292`, 0.5569283894317684`, 0.5607144272768734`,
    0.5644719464385364`, 0.5682016799072628`, 0.5719042082158571`,
    0.575580219726987`, 0.5792300320466102`, 0.5828544136175161`,
    0.5864537465337384`, 0.590028495579292`, 0.5935794613182722`,
    0.597106733563209`, 0.600611106862938`, 0.6040927268416939`,
    0.6075521046084281`, 0.6109898235468666`, 0.6144060450594765`,
    0.6178011643396959`, 0.6211756370476421`, 0.6245299334605183`,
    0.6278643016230703`, 0.631178942964761`, 0.6344744397867419`,
    0.6377509974210513`, 0.6410088632079404`, 0.6442485743730993`,
    0.6474700713446719`, 0.6506737762956742`, 0.6538601571816398`,
    0.6570293164374893`, 0.6601814911055884`, 0.6633169007981367`,
    0.6664360102951657`, 0.6695390588023079`, 0.6726258861051402`,
    0.6756971105167999`, 0.6787528456317353`, 0.6817933963001003`,
    0.6848188749440584`, 0.687829473947805`, 0.6908253545823321`,
    0.6938068743106269`, 0.6967742471367503`, 0.6997272446015926`,
    0.702666647791757`, 0.7055923307639006`, 0.7085045379264433`,
    0.7114032956413593`, 0.7142889499810882`, 0.717161566038073`,
    0.7200214102435792`, 0.7228684595400519`, 0.7257029543174206`,
    0.7285252185373823`, 0.73133515223929`, 0.7341333710233495`,
    0.7369192262241302`, 0.7396933651809913`, 0.7424559402304332`,
    0.7452069202922691`, 0.747946483796247`, 0.7506750261057225`,
```

```
0.7533921519794734`, 0.756098282047122`, 0.7587935839371749`,
0.7614782391229655`, 0.7641522341816558`, 0.766815633970615`,
0.7694685841446658`, 0.7721112425994111`, 0.7747435648901838`,
0.7773661614832421`, 0.779978506585171`, 0.7825810896474582`,
0.7851735385743769`, 0.7877568100152431`, 0.7903301801243728`,
0.7928941117890215`, 0.7954488602711575`, 0.7979941623556892`,
0.800530289019349`, 0.8030571398648381`, 0.8055752367910314`,
0.8080841864820628`, 0.8105844529999935`, 0.8130758870104161`,
0.8155585787325944`, 0.8180328652186675`, 0.8204983745763316`,
0.822955538311062`, 0.8254043159770577`, 0.8278445892661761`,
0.8302770636056197`, 0.8327012257629092`, 0.8351173031767798`,
0.8375254494234393`, 0.8399254895505542`, 0.8423179254614716`,
0.8447025091424766`, 0.8470794186435049`, 0.8494483239092043`,
0.851809929365217`, 0.8541640892392187`, 0.8565107456331125`,
0.8588500188794068`, 0.8611817703045229`, 0.8635065728489688`,
0.8658238535348557`, 0.8681342881980378`, 0.8704373962941437`,
0.8727335074373603`, 0.8750226876061623`, 0.8773050990133543`,
0.8795806336002711`, 0.8818489986146681`, 0.884110923216611`,
0.8863659369364857`, 0.888614503790109`, 0.8908564377606971`,
0.8930917419168972`, 0.8953206131882652`, 0.8975429669691107`,
0.8997590090582904`, 0.9019687578861441`, 0.9041722346978216`,
0.9063694635703808`, 0.9085604714294255`, 0.9107452880652919`,
0.9129238467961884`, 0.9150965810725752`, 0.9172631324343173`,
0.9194238431805463`, 0.9215785579988824`, 0.9237273241818866`,
0.9258702941787089`, 0.9280074199812023`, 0.9301390672228469`,
0.932264677742416`, 0.9343847258551512`, 0.9364991737009724`,
0.9386078795500246`, 0.9407111229233496`, 0.9428088722809106`,
0.9449013101388966`, 0.9469878756406493`, 0.9490692882371795`,
0.9511450954664803`, 0.9532159217337818`, 0.9552812073205131`,
0.9573411491506553`, 0.9593959491675945`, 0.9614455939516298`,
0.9634900713617821`, 0.9655293705453395`, 0.967563705149758`,
0.9695929576683372`, 0.9716170100341026`, 0.9736364216696167`,
0.975650627945247`, 0.9776599677493493`, 0.9796644437011721`,
0.9816639448959925`, 0.9836587059563793`, 0.9856486191755861`,
0.987633808912596`, 0.9896140525391764`, 0.9915898283591158`,
0.9935607997845309`, 0.9955269804691241`, 0.9974887413111032`,
0.9994458648303358`, 1.001398310555302`, 1.0033462185242825`,
1.0052897320715952`, 1.007228695484231`, 1.0091628910012245`,
1.0110932563473203`, 1.0130189096441273`, 1.0149400617159339`,
1.0168569892729242`, 1.018769479305407`, 1.0206778133282977`,
1.0225818417597372`, 1.0244816639561434`, 1.026377193320002`,
1.028268531961934`, 1.030155721165114`, 1.0320389306868398`,
1.0339176964780386`, 1.035792443868854`, 1.0376632843620954`,
1.0395300098542484`, 1.0413926686328956`, 1.0432511807701503`,
1.045105791538309`, 1.0469563585133799`, 1.0488031314281498`,
1.0506461011461645`, 1.052484532873392`, 1.0543198024940121`,
1.056150912371827`, 1.0579781192989448`, 1.059801684391496`,
```

```
1.0616212684986406`, 1.0634373373956028`, 1.0652493493948119`,
1.0670577741030187`, 1.068862407254588`, 1.0706631791375698`,
1.072460568228664`, 1.0742539595356533`, 1.0760440417482011`,
1.0778304055017276`, 1.0796129839663096`, 1.0813921979881993`,
1.0831681249359628`, 1.0849398615301653`, 1.0867085345583698`,
1.0884734500758035`, 1.0902349683766397`, 1.0919930996453493`,
1.0937476408485918`, 1.0954986730382423`, 1.0972464224027143`,
1.0989908306159692`, 1.1007316947073968`, 1.1024693892410047`,
1.1042035673141524`, 1.1059342426834233`, 1.1076617223393155`,
1.1093860242039695`, 1.1111068725747626`, 1.1128244317037257`,
1.114538720305203`, 1.1162499063538467`, 1.117957713395769`,
1.119662310404143`, 1.12136371900031`, 1.1230619615369786`,
1.124757362718283`, 1.1264491927732807`, 1.1281380031831143`,
1.129823667753556`, 1.1315062118398929`, 1.133185814617186`,
1.1348625042860345`, 1.1365357708461226`, 1.138206025324741`,
1.1398737614724774`, 1.1415380034657348`, 1.1431992445183596`,
1.1448576715326415`, 1.1465130822244378`, 1.1481653509277696`,
1.14981490253873`, 1.151461535008798`, 1.1531049650849508`,
1.1547459398920388`, 1.1563838604609968`, 1.1580188407384637`,
1.159651076494444`, 1.1612802034003469`, 1.1629065788459503`,
1.164530322099512`, 1.1661509056441963`, 1.1677690156593323`,
1.169384042187337`, 1.170996431358067`, 1.1726060608771598`,
1.1742128896078423`, 1.175816876290687`, 1.1774182274374707`,
1.1790169864244375`, 1.180612782007797`, 1.1822060722767684`,
1.1837966523618475`, 1.185384399236452`, 1.1869699446552409`,
1.1885524960166611`, 1.1901323500650816`, 1.1917099763174044`,
1.1932846613364887`, 1.1948567906689418`, 1.1964264129498494`,
1.197993235427668`, 1.1995578202376678`, 1.2011194458068857`,
1.2026786765242468`, 1.204235391431757`, 1.2057893819167413`,
1.2073411325544474`, 1.2088902621061124`, 1.2104369969774456`,
1.2119811283885207`, 1.2135227967850342`, 1.2150620563029202`,
1.2165990500976087`, 1.218133303903538`, 1.2196652255799114`,
1.2211948717900192`, 1.2227218544378466`, 1.224246764456773`,
1.2257688555702393`, 1.2272886312049167`, 1.2288059700290048`,
1.2303210204060382`, 1.2318339323917857`, 1.2333441323987473`,
1.2348522229850756`, 1.2363581750653856`, 1.23786132014657`,
1.239362631418801`, 1.2408612562351051`, 1.2423578978422778`,
1.2438521598762051`, 1.2453440121116812`, 1.246833702230932`,
1.2483212947314333`, 1.2498064825063986`, 1.2512890490459911`,
1.2527696179090657`, 1.2542480686383408`, 1.2557243738075001`,
1.257198600126595`, 1.2586701541166507`, 1.260139856673236`,
1.261607398126345`, 1.2630728464516332`, 1.264536079816773`,
1.2659968799090988`, 1.2674556968480217`, 1.2689124092620565`,
1.270366894833042`, 1.271819223121538`, 1.273269754400302`,
1.274717883878548`, 1.276163583855084`, 1.2776077021915522`,
1.2790493393963318`, 1.2804892494324613`, 1.2819271157752408`,
1.283362815544118`, 1.28479632331054`, 1.286227810805125`,
```

1.2876576491113996`, 1.2890847270513295`, 1.290510206875884`,
1.2919336695279764`, 1.2933549915019877`, 1.2947744480551853`,
1.2961918163039317`, 1.2976071726426444`, 1.2990204937066028`,
1.3004318568760505`, 1.301841542387289`, 1.3032488206864725`,
1.304654477865368`, 1.3060579852074898`, 1.307459420923804`,
1.3088592724858261`, 1.3102568037471127`, 1.31165270804286`,
1.3130467598912385`, 1.3144387320158195`, 1.3158286017471739`,
1.3172168634562957`, 1.3186031867392212`, 1.3199875500899598`,
1.3213698278846793`, 1.3227505193130364`, 1.32412918731621`,
1.3255063338193014`, 1.3268812057609802`, 1.3282545161615942`,
1.3296259303279276`, 1.330995533285051`, 1.3323633050729613`,
1.3337292256878173`, 1.3350933813261538`, 1.3364555396273043`,
1.337815893163017`, 1.339174636197703`, 1.3405314291325683`,
1.3418864663650094`, 1.3432398364402176`, 1.3445914132069283`,
1.3459411775842909`, 1.3472892186449514`, 1.3486355178771687`,
1.349979948103245`, 1.3513224900838918`, 1.352663451716938`,
1.354002487807315`, 1.3553401264536982`, 1.3566755835582773`,
1.3580099376199282`, 1.3593424033855996`, 1.3606731827071752`,
1.3620021474171164`, 1.3633292792341467`, 1.3646544488406769`,
1.3659784157508628`, 1.3673006082970942`, 1.3686210087023825`,
1.369939711017237`, 1.3712571463506573`, 1.3725727385872868`,
1.3738865823199442`, 1.375198772968319`, 1.3765090679659524`,
1.3778180150609494`, 1.3791254860621094`, 1.3804307839753607`,
1.3817347997745701`, 1.3830370631354845`, 1.3843382425261386`,
1.3856374093662993`, 1.386934660073528`, 1.3882306662387096`,
1.3895248386479884`, 1.3908173908545296`, 1.3921087691185343`,
1.3933980343847705`, 1.3946859798164262`, 1.3959727076163364`,
1.39725762260555`, 1.3985408247875228`, 1.3998224149925667`,
1.4011024948773607`, 1.4023810497017606`, 1.403658182133369`,
1.4049336427156631`, 1.406207769572979`, 1.4074800768252584`,
1.4087507848917222`, 1.410020234374501`, 1.4112880561075887`,
1.4125543540583325`, 1.4138191138770833`, 1.4150824405433908`,
1.416344320293487`, 1.4176047393371336`, 1.4188636838578659`,
1.42012077939139`, 1.4213766122551128`, 1.4226308079340007`,
1.4238835935619598`, 1.4251351979482803`, 1.4263851239562622`,
1.4276335999202163`, 1.4288804907647747`, 1.4301261481692882`,
1.4313700715010353`, 1.432612735359351`, 1.4338540049965882`,
1.4350936220374577`, 1.436332187158943`, 1.4375689502814222`,
1.4388041433458245`, 1.4400383707733913`, 1.4412708792377968`,
1.4425020265443782`, 1.443731924540356`, 1.4449601880999239`,
1.4461871775008341`, 1.4474127560482215`, 1.4486367861797496`,
1.4498596306444373`, 1.4510809016703605`, 1.4523008375141522`,
1.4535195521723803`, 1.4547366559089625`, 1.455952514533954`,
1.4571669901251434`, 1.4583800706675447`, 1.4595917441257342`,
1.4608019984440486`, 1.4620108215467835`, 1.4632184565664232`,
1.464424764826117`, 1.4656296067870436`, 1.4668330985398916`,
1.4680352285418006`, 1.4692361138964285`, 1.4704357436529827`,

```
1.4716338486690073`, 1.4728305459911029`, 1.4740262124998835`,
1.4752205789961788`, 1.4764131147304391`, 1.4776047170714475`,
1.4787948554432075`, 1.4799836486715605`, 1.481171216403153`,
1.4823579410536825`, 1.4835428956966137`, 1.4847264614548663`,
1.4859090219475422`, 1.4870903042286818`, 1.4882697696178833`,
1.4894483306044444`, 1.4906258460016832`, 1.4918017758232347`,
1.4929765067237806`, 1.4941496294508534`, 1.4953217984308638`,
1.4964926043782794`, 1.4976624376420917`, 1.498830619847864`,
1.4999978095325033`, 1.5011637291587256`, 1.50232836855884`,
1.5034915828138347`, 1.5046533610996016`, 1.5058142331853956`,
1.5069740553553659`, 1.5081325474760792`, 1.509289563509938`,
1.5104456364960452`, 1.511599940859425`, 1.5127532828271215`,
1.5139055173416966`, 1.5150564984990957`, 1.5162064906213664`,
1.5173550735189179`, 1.5185025116922826`, 1.519648383054069`,
1.520793503701952`, 1.5219370379569495`, 1.523079804377956`,
1.5242211032449584`, 1.5253609240966102`, 1.5264998120219342`,
1.5276376199207864`, 1.5287739212404987`, 1.5299098215626692`,
1.5310434989802453`, 1.532177038736356`, 1.5333083344932539`,
1.5344390566554398`, 1.5355683557090931`, 1.536696784693206`,
1.5378237722660533`, 1.538950014259206`, 1.5400745141557528`,
1.5411982517449756`, 1.5423207947340585`, 1.5434418503326863`,
1.544562261665935`, 1.5456811679549787`, 1.546799272164573`,
1.5479157103994663`, 1.5490313300640701`, 1.550145694448846`,
1.5512590815262743`, 1.552371196348602`, 1.5534820299990306`,
1.5545918616491285`, 1.555700539337195`, 1.5568080547285639`,
1.5579148334760267`, 1.5590199998401693`, 1.5601241239118933`,
1.561227052644695`, 1.5623293595966175`, 1.5634297275947246`,
1.5645297500216349`, 1.5656281072213283`, 1.5667256667579963`,
1.5678222755175046`, 1.5689174861210056`, 1.570012024095355`,
1.5711045592816744`, 1.5721968480937418`, 1.5732880011163008`,
1.5743774194937827`, 1.5754664243664245`, 1.5765539738430914`,
1.5776405038304095`, 1.578725858447599`, 1.5798101785095662`,
1.5808933077069276`, 1.5819759839203835`, 1.5830568578731037`,
1.5841366670182309`, 1.5852157035339924`, 1.5862935112949508`,
1.58737053285177`, 1.5884464612288474`, 1.5895212890030266`,
1.5905947071431166`, 1.5916670089919442`, 1.5927386407166044`,
1.5938088396400245`, 1.5948786583608943`, 1.595947181553801`,
1.597014249101697`, 1.598080461177013`, 1.5991455066419873`,
1.6002096832375305`, 1.6012726789957763`, 1.6023349454848297`,
1.603396017279411`, 1.6044558868294105`, 1.6055147002179861`,
1.6065726043301156`, 1.6076292847123965`, 1.6086855050260107`,
1.60974002510383`, 1.6107939180999078`, 1.6118465594095244`,
1.6128982515319787`, 1.6139489880553728`, 1.6149986070972773`,
1.6160469459591111`, 1.6170947763219583`, 1.6181413132190132`,
1.6191868615250657`, 1.6202314147788273`, 1.6212751231837232`,
1.6223173533698598`, 1.6233588824173832`, 1.6243995477649327`,
1.6254393435158139`, 1.6264774753656353`, 1.6275150395509097`,
```

```
1.628551557111899`, 1.629587179800489`, 1.6306215847120673`,
1.631655082067975`, 1.6326876659173029`, 1.6337191712144277`,
1.634749591390943`, 1.6357790793598577`, 1.6368073097622586`,
1.637835234744738`, 1.6388617300870412`, 1.6398872688540866`,
1.6409115240543461`, 1.6419354526166743`, 1.6429580855715824`,
1.6439795767784529`, 1.645000564766124`, 1.6460200763397348`,
1.6470389121063704`, 1.6480564196408232`, 1.6490732400446484`,
1.6500892061537493`, 1.651104149997884`, 1.6521180654918466`,
1.653130620763889`, 1.654142623966672`, 1.6551539075019919`,
1.6561638125018077`, 1.6571726590280522`, 1.6581807687308907`,
1.6591878085038883`, 1.6601941007810046`, 1.661199147162408`,
1.6622037643174614`, 1.6632066294243508`, 1.6642093847424606`,
1.6652107054488918`, 1.6662110806129098`, 1.6672106703289113`,
1.6682093037292056`, 1.6692069752194059`, 1.6702038454711499`,
1.6711994100623497`, 1.6721939955283813`, 1.6731879299849128`,
1.6741810419325212`, 1.6751729918102598`, 1.6761639408497684`,
1.6771538834104656`, 1.6781434852721975`, 1.6791313987223673`,
1.6801189617465162`, 1.6811053288058204`, 1.6820906622862222`,
1.6830749565141847`, 1.684058712936781`, 1.685041419928168`,
1.6860230717986149`, 1.6870040021208697`, 1.6879838667160871`,
1.6889626598790912`, 1.6899407163495654`, 1.690918031593969`,
1.6918942598357924`, 1.6928693953495881`, 1.693843603417326`,
1.6948170501026807`, 1.6957897308452368`, 1.6967616410792061`,
1.6977319172425798`, 1.698701755775051`, 1.6996708088918677`,
1.7006388996196038`, 1.7016058501950935`, 1.7025720004034892`,
1.7035373456433485`, 1.7045015349663162`, 1.705464909319371`,
1.7064276376488197`, 1.707389194646172`, 1.7083500963728397`,
1.7093101646355138`, 1.7102688717641068`, 1.7112272586006754`,
1.7121840991100676`, 1.7131404358838107`, 1.7140957400615502`,
1.7150501817201946`, 1.7160037561966146`, 1.7169564588231514`,
1.7179082849276557`, 1.7188592298335288`, 1.7198092888597627`,
1.7207582808310693`, 1.7217067305274776`, 1.722653926906282`,
1.7236003953226997`, 1.7245459543898056`, 1.725490599400997`,
1.7264343256455497`, 1.7273776619595194`, 1.7283195371043598`,
1.7292606575641063`, 1.7302010191935677`, 1.7311402605911927`,
1.7320789128998626`, 1.7330162564893015`, 1.7339530022828313`,
1.7348889672947083`, 1.735823967760876`, 1.7367579989269237`,
1.737691415995711`, 1.7386234946701`, 1.7395549504927397`,
1.740485599293233`, 1.7414154368776948`, 1.7423444590483186`,
1.7432724804649387`, 1.7441994963437986`, 1.745125864945411`,
1.7460514009184005`, 1.7469759181371605`, 1.74789959390534`,
1.748822423999076`, 1.74974458667679`, 1.7506655302495915`,
1.7515856150703957`, 1.7525052030255892`, 1.7534235580087585`,
1.7543412249613286`, 1.755258016572469`, 1.756173928590158`,
1.7570889567590133`, 1.75800346524564`, 1.7589165289161757`,
1.7598292493554788`, 1.7607408848697874`, 1.7616518005693453`,
1.762561622414091`, 1.763470716272913 7`, 1.7643788928862172`,
```

```
1.7652863337115265`, 1.7661928491007293`, 1.7670984347629721`,
1.7680032727187656`, 1.7689071727381986`, 1.7698105039121623`,
1.770712702422986`, 1.7716139504697879`, 1.7725146182766651`,
1.7734141403037185`, 1.774313262252434`, 1.7752114180060645`,
1.7761086032441276`, 1.7770050018599157`, 1.7779002332848057`,
1.778794669812906`, 1.779688685252684`, 1.780581899026588`,
1.7814741182018397`, 1.782365338438954`, 1.78325555539612`,
1.7841453339250277`, 1.7850339116965934`, 1.7859218541243065`,
1.7868089676909862`, 1.7876950581209188`, 1.7885805024101338`,
1.7894652973314518`, 1.7903490575470773`, 1.7912315874504152`,
1.792113456401529`, 1.7929952360183814`, 1.7938751984897299`,
1.7947550651407362`, 1.7956338735006832`, 1.7965116191911272`,
1.7973888755092018`, 1.7982648687750873`, 1.7991401728447753`,
1.8000147844575225`, 1.800888700349804`, 1.8017611432664649`,
1.8026332697973366`, 1.8035048839233752`, 1.804375400755038`,
1.805244815892578`, 1.8061133193679018`, 1.80698149119882`,
1.8078485499952532`, 1.8087146863434582`, 1.8095800915747695`,
1.8104445670312328`, 1.8113083044129086`, 1.8121713004238247`,
1.8130335517655642`, 1.8138948590251973`, 1.814755022042981`,
1.8156146258498984`, 1.8164736676964612`, 1.817331751118554`,
1.8181886752048295`, 1.8190450272088856`, 1.8199006069583807`,
1.820755213536338`, 1.8216090408494667`, 1.8224622835439759`,
1.8233145425421722`, 1.824166210640046`, 1.8250164909660835`,
1.8258661733615682`, 1.8267154539422554`, 1.8275633350874898`,
1.8284108081151318`, 1.8292572724497362`, 1.8301031234615939`,
1.8309477589103813`, 1.831791574347649`, 1.8326349667973683`,
1.8334773329078553`, 1.8343188693297787`, 1.8351597734481384`,
1.8359992387765225`, 1.8368390702475426`, 1.8376770536381857`,
1.8385145931381344`, 1.8393510815216612`, 1.8401871203915707`,
1.8410218993961478`, 1.8418562228897284`, 1.8426896838274294`,
1.8435224814106277`, 1.8443538018021821`, 1.8451848576029888`,
1.8460152408453865`, 1.8468443387965185`, 1.8476727574362999`,
1.8485002902671108`, 1.8493273416060556`, 1.8501535011032737`,
1.8509785611626515`, 1.8518029266431904`, 1.8526267992813972`,
1.8534493577472457`, 1.8542716222617601`, 1.8550925654614672`,
1.855913004077934`, 1.856732935807027`, 1.8575513300137656`,
1.8583692109060732`, 1.8591863701463083`, 1.860003217297397`,
1.860818305859711`, 1.861633282827134`, 1.8624471132108378`,
1.8632602069295017`, 1.864072561120727`, 1.8648839656279754`,
1.8656948319906552`, 1.8665045349235`, 1.8673136939768584`,
1.8681214747692159`, 1.868929330160602`, 1.8697358008282812`,
1.870541507937853`, 1.871346866078117`, 1.8721510378046446`,
1.8729546464351021`, 1.873756643279167`, 1.8745582800006748`,
1.8753597644210929`, 1.8761594168134685`, 1.8769589114964351`,
1.8777574065366873`, 1.87855489851209`, 1.8793515945214196`,
1.8801477023763498`, 1.8809432197225517`, 1.881737511035171`,
1.8825312060566972`, 1.8833238796039784`, 1.8841161630199375`,
```

```
    1.8849078427087858`, 1.885698068524691`, 1.8864878965880163`,
    1.8872764758633753`, 1.8880644398214077`, 1.888851360784091`,
    1.889637873815894`, 1.8904233380500348`, 1.8912083896111918`,
    1.8919921732053286`, 1.8927751119466676`, 1.8935578440783485`,
    1.8943390849275576`, 1.8951199003439796`, 1.8958998599875017`,
    1.896679175370643`, 1.8974574149174854`, 1.898235004736192`,
    1.8990115125454032`, 1.8997877954048759`, 1.9005625601812517`,
    1.9013366642845018`, 1.9021105366292999`, 1.9028830967472652`,
    1.9036547726746396`, 1.9044259938485233`, 1.905196325699549`,
    1.9059655487842284`, 1.9067343097629972`, 1.907502173014598`,
    1.9082691356199206`, 1.9090354118835668`, 1.909800782009924`,
    1.910565025505948`, 1.9113292276606804`, 1.912091862197586`,
    1.9128542331944174`, 1.9136156844889056`, 1.9143759947164496`,
    1.9151358162771779`, 1.9158949284994524`, 1.9166528910560738`,
    1.9174101387713007`, 1.9181668885463004`, 1.9189226995327369`,
    1.9196775688043424`, 1.9204317132601258`, 1.9211851304910725`,
    1.9219375979170896`, 1.9226891126108079`, 1.9234401126740541`,
    1.92419015483814`, 1.924939236174853`, 1.9256877958165928`,
    1.926435389461155`, 1.9271824569267`, 1.9279283316780047`,
    1.9286734537042136`, 1.9294180424772411`, 1.930161429918624`,
    1.9309042792739655`, 1.9316461438382582`, 1.9323874658234315`,
    1.9331277978400914`, 1.9338673598699045`, 1.9346063725761518`,
    1.9353443875446958`, 1.9360814018447525`, 1.9368176361385638`,
    1.9375530880054916`, 1.93828775502447`, 1.9390218588758834`,
    1.9397547248322753`, 1.9404872472176915`, 1.9412189754054876`,
    1.9419492326349297`, 1.942679139701924`, 1.9430224711124329`}[[s]];
CS2rho800[s_] := {0.30941848369226793`, 0.31993804910517665`,
    0.3295881350584772`, 0.338571965948296`, 0.3470280615643063`,
    0.35505486910099493`, 0.3627243588939767`, 0.37009085010077497`,
    0.3771963303993171`, 0.3840738657684959`, 0.39074979488054257`,
    0.3972459233389879`, 0.4035799197267094`, 0.4097668279204688`,
    0.415819218001268`, 0.42174790473652723`, 0.42756229275843116`,
    0.43327013116472934`, 0.43887892455041977`, 0.4443946705747395`,
    0.44982300629652944`, 0.45516894562015203`, 0.4604369113223703`,
    0.4656308001885923`, 0.47075439827155646`, 0.4758110606595847`,
    0.4808037483925651`, 0.4857352536358308`, 0.4906081204848083`,
    0.49542479892691726`, 0.5001872584921473`, 0.5048979313986856`,
    0.5095582629866284`, 0.5141704117233145`, 0.5187357764586469`,
    0.5232559756984303`, 0.5277325100933795`, 0.5321667959346622`,
    0.536559964769217`, 0.5409133086109762`, 0.5452279567535835`,
    0.5495052780257185`, 0.5537458973756406`, 0.5579511083220012`,
    0.5621216315436471`, 0.5662584001902512`, 0.5703626041135476`,
    0.5744345571436046`, 0.5784752599415816`, 0.5824855512321336`,
    0.5864660466153581`, 0.5904174479500071`, 0.5943402897092634`,
    0.5982351772446349`, 0.6021031313266311`, 0.6059441208032671`,
    0.6097592054573128`, 0.6135487161355786`, 0.6173131120297202`,
    0.6210530305901975`, 0.6247688753590038`, 0.6284610331081611`,
```

```
0.6321302503869709`, 0.6357765908989281`, 0.6394007081832862`,
0.6430029402911409`, 0.6465835875712334`, 0.6501432102470273`,
0.6536820288732276`, 0.6572005723529459`, 0.6606990678737964`,
0.6641777378026162`, 0.6676371163414228`, 0.6710774195855446`,
0.6744990149393759`, 0.6779020436279322`, 0.6812868501107823`,
0.684653879007869`, 0.6880032268522926`, 0.691335420979817`,
0.6946503461613786`, 0.6979483735353964`, 0.7012300311677`,
0.7044952998838776`, 0.7077444222130259`, 0.7109776123545867`,
0.7141951753601793`, 0.7173974488055402`, 0.7205843078798876`,
0.723756142918191`, 0.7269133145910512`, 0.730055770199139`,
0.7331839219560083`, 0.7362975676468246`, 0.7393974928943481`,
0.7424832778561739`, 0.7455556311011391`, 0.7486142314744124`,
0.7516595032113015`, 0.7546915597267905`, 0.7577106008205949`,
0.7607167086097586`, 0.7637103289524835`, 0.766691234104283`,
0.7696597667773302`, 0.7726162215910504`, 0.7755603405635441`,
0.7784925919567164`, 0.7814127417681415`, 0.7843215133577907`,
0.7872184804631377`, 0.7901038099307004`, 0.7929782085757762`,
0.7958412057050127`, 0.798693087006868`, 0.8015340793073291`,
0.8043641165691121`, 0.8071835281832701`, 0.8099922704838214`,
0.8127904645698748`, 0.8155784018058094`, 0.818355910632022`,
0.8211232245525065`, 0.8238805102346082`, 0.8266278648788046`,
0.8293652315904744`, 0.8320928892757865`, 0.834810715812237`,
0.8375189271748804`, 0.8402179199843587`, 0.8429074299898955`,
0.8455872776764607`, 0.848258143011539`, 0.8509199524744729`,
0.8535725496800098`, 0.8562163044920539`, 0.8588510740593764`,
0.861476982937666`, 0.8640941641921558`, 0.8667027594443917`,
0.8693026486627253`, 0.8718943483659766`, 0.8744773896010617`,
0.8770523974687351`, 0.8796187252933186`, 0.8821770146866359`,
0.8847270870657908`, 0.887269045682688`, 0.8898032837174195`,
0.892329448237208`, 0.8948476597077427`, 0.8973581414693613`,
0.8998608363314899`, 0.9023560780216814`, 0.9048435283190212`,
0.9073233359778788`, 0.9097959528077014`, 0.9122608520233747`,
0.9147185961199463`, 0.9171688607659707`, 0.9196121245819675`,
0.9220480689889939`, 0.92447698320545`, 0.9268987561363367`,
0.9293134834189167`, 0.9317213693045606`, 0.9341220012102173`,
0.9365159020328894`, 0.9389029778796254`, 0.9412834526665128`,
0.9436572392151834`, 0.9460243580232908`, 0.9483848327309694`,
0.9507387978944725`, 0.9530861768096673`, 0.9554271114798297`,
0.9577616399052127`, 0.9600898033217523`, 0.9624116462184312`,
0.9647272163541842`, 0.9670364532946485`, 0.9693395218040518`,
0.9716365920915117`, 0.9739273875281313`, 0.976212423695292`,
0.9784913149442868`, 0.9807642438432509`, 0.9830310528814328`,
0.9852920466220311`, 0.9875473047728229`, 0.9897966768935097`,
0.9920403637028475`, 0.9942782179482894`, 0.9965105651211894`,
0.9987370249794743`, 1.0009579289261603`, 1.003173076250397`,
1.0053828066951869`, 1.007587104684652`, 1.009786077614516`,
1.011979226011434`, 1.0141670242732617`, 1.0163495257167285`,
```

```
1.018526600674051`, 1.0206985535189708`, 1.0228650717688181`,
1.025026528031735`, 1.0271826738548733`, 1.029333636990107`,
1.031479358566677`, 1.0336202259444887`, 1.0357559949077786`,
1.0378866111876388`, 1.0400120854828467`, 1.0421325595177866`,
1.0442482436850777`, 1.0463590920711177`, 1.0484647316803504`,
1.0505661004962463`, 1.0526619071927688`, 1.05475329451348`,
1.0568397603485908`, 1.0589215309926063`, 1.0609987033177963`,
1.0630711074337476`, 1.0651387748058847`, 1.0672016706576752`,
1.0692601696951465`, 1.0713140378389863`, 1.0733629698061773`,
1.0754078286015443`, 1.0774481075767062`, 1.0794837101204364`,
1.0815146786475573`, 1.0835416870779389`, 1.0855640156176338`,
1.0875822742124124`, 1.0895958091269895`, 1.0916050942823778`,
1.093610041292911`, 1.0956107044953223`, 1.0976072117563687`,
1.0995990450907798`, 1.1015870541648574`, 1.10357072263317`,
1.1055502560584058`, 1.1075254982727856`, 1.1094965118466011`,
1.1114637292655756`, 1.1134267771544517`, 1.115385722568443`,
1.1173404854990692`, 1.1192913589911524`, 1.1212379665788907`,
1.1231809800382708`, 1.1251199494403343`, 1.1270546464995652`,
1.1289856002859675`, 1.1309124331021363`, 1.1328359101861818`,
1.1347549658840654`, 1.1366704460342556`, 1.1385821268728051`,
1.140489937346938`, 1.1423939617116037`, 1.1442941299360392`,
1.1461904500343367`, 1.1480830879280586`, 1.1499720547433663`,
1.1518573623897632`, 1.1537391029035795`, 1.1556169721489358`,
1.1574913015065613`, 1.159361866476015`, 1.161228842125022`,
1.163092083858851`, 1.1649522547233182`, 1.166808485009413`,
1.1686612783695542`, 1.1705104921445921`, 1.1723560643471733`,
1.1741982614922728`, 1.1760371066929578`, 1.1778722104087824`,
1.1797037594355047`, 1.1815321930805938`, 1.1833571219590504`,
1.1851786547106236`, 1.1869969858150133`, 1.188811638739129`,
1.1906233145413565`, 1.1924311978118172`, 1.1942359088454606`,
1.196037393848325`, 1.1978355133202063`, 1.1996302123304552`,
1.2014221240428562`, 1.203210508998037`, 1.2049957444669093`,
1.2067776050509507`, 1.2085561243083056`, 1.2103314239675722`,
1.212103715208639`, 1.2138727729823777`, 1.2156387223589218`,
1.217401601954203`, 1.2191612740590863`, 1.2209180440296223`,
1.2226715974209736`, 1.2244219740081703`, 1.2261691248201734`,
1.2279132703461042`, 1.2296543636116706`, 1.2313925382551016`,
1.233127657902731`, 1.23485976679653`, 1.2365890011994987`,
1.2383150422746618`, 1.2400382095666418`, 1.2417584589218755`,
1.243475653887303`, 1.2451899341082797`, 1.2469013484599587`,
1.2486097608515532`, 1.2503154067623776`, 1.2520178705582277`,
1.2537176692853886`, 1.2554144804886842`, 1.2571087320988694`,
1.258799912514645`, 1.2604885466589633`, 1.2621737410906253`,
1.2638564972966084`, 1.2655363004045939`, 1.2672135875089714`,
1.2688879369389647`, 1.2705594039648334`, 1.2722281412301408`,
1.2738943032896983`, 1.2755576586911064`, 1.2772182655466795`,
1.2788761828709176`, 1.2805311773690304`, 1.2821838955282339`,
```

```
1.2838336154078251`, 1.2854809868186452`, 1.2871254815726005`,
1.288767061329374`, 1.2904063824030123`, 1.2920429140325436`,
1.293676918354284`, 1.2953083604102593`, 1.2969370046297048`,
1.2985630153704786`, 1.300186458147663`, 1.3018071973639849`,
1.3034257039045243`, 1.3050414384409836`, 1.306654365358661`,
1.3082650610545727`, 1.3098732897188177`, 1.3114790184819465`,
1.3130821116173517`, 1.3146829474095978`, 1.3162810819021173`,
1.317876791692762`, 1.319470252530158`, 1.3210609139274185`,
1.322649367305257`, 1.3242349557377184`, 1.3258184830325157`,
1.327399605669997`, 1.3289780823228625`, 1.3305543027218316`,
1.3321279201111487`, 1.3336989027274058`, 1.3352680689042618`,
1.336834647069679`, 1.3383986059836983`, 1.3399603424126851`,
1.341519827962712`, 1.3430767116122324`, 1.344631393649177`,
1.3461836301781005`, 1.347733716566661`, 1.349281082988203`,
1.3508266772474826`, 1.3523697118547298`, 1.3539104848877146`,
1.3554491878316373`, 1.3569854653093159`, 1.3585195089213995`,
1.3600512916835115`, 1.3615808970476466`, 1.363108187895427`,
1.3646331370500406`, 1.3661558285172013`, 1.3676763472318993`,
1.3691945555773455`, 1.370710650849189`, 1.3722246077918976`,
1.3737359511576301`, 1.3752453289844981`, 1.376752604344679`,
1.3782576392624457`, 1.3797604081873667`, 1.3812612266680016`,
1.382759615400465`, 1.3842561189859128`, 1.3857504856919822`,
1.3872424617568546`, 1.3887325960571826`, 1.3902201751603493`,
1.3917063267862233`, 1.3931897589373463`, 1.394671255602783`,
1.3961505619606338`, 1.3976280033162103`, 1.399103324497561`,
1.4005761513928445`, 1.4020471613945513`, 1.4035164506791662`,
1.404983528063592`, 1.4064482520974688`, 1.4079109528976477`,
1.4093717266887313`, 1.410830433349022`, 1.4122874072674834`,
1.4137420326883083`, 1.4151947632816304`, 1.4166454584925776`,
1.418093856876555`, 1.4195408960769327`, 1.4209857162156039`,
1.4224280535115004`, 1.4238687297687764`, 1.4253076048955802`,
1.426744416447191`, 1.4281792647078353`, 1.4296122508958076`,
1.4310427447826084`, 1.432471701214345`, 1.433898612186571`,
1.4353235797263775`, 1.4367465837565716`, 1.4381679740051292`,
1.4395871148787225`, 1.4410044797939452`, 1.4424200499737905`,
1.443833309581002`, 1.4452448593335798`, 1.446654681066632`,
1.4480626315923029`, 1.4494684414859706`, 1.45087259216148`,
1.4522750660297525`, 1.4536750897347097`, 1.4550733984316424`,
1.456470227300014`, 1.4578647999656589`, 1.4592579842180893`,
1.460649128597369`, 1.4620383412156783`, 1.4634262417199768`,
1.4648116640708073`, 1.4661953554968903`, 1.4675775554935653`,
1.4689574765181717`, 1.4703358715004164`, 1.471712466332719`,
1.4730873729668372`, 1.4744605747044108`, 1.4758316653433585`,
1.4772011462095094`, 1.4785686103141016`, 1.4799341699318684`,
1.4812980690574264`, 1.4826606846080115`, 1.4840212149410361`,
1.4853799046808653`, 1.4867364733491053`, 1.4880918264213834`,
1.4894452893137204`, 1.4907969774273833`, 1.49214687438521`,
```

1.4934949637811883`, 1.4948417619392902`, 1.496186721556164`,
1.4975296924707904`, 1.4988709252623638`, 1.5002108065471909`,
1.5015487850740974`, 1.5028852481220223`, 1.5042197766176766`,
1.5055527592217173`, 1.5068839107162344`, 1.5082136218604645`,
1.509541063855995`, 1.5108677151799321`, 1.5121922024993708`,
1.5135151909790352`, 1.5148363930943058`, 1.516155519387509`,
1.517473513748484`, 1.5187894018344372`, 1.520104130820958`,
1.5214168609485332`, 1.5227281287748347`, 1.52403764398096`,
1.5253455303641863`, 1.526652051567544`, 1.527956777330466`,
1.529259971486145`, 1.5305612015895664`, 1.5318615708751915`,
1.5331593880381296`, 1.5344560379020487`, 1.5357513683774036`,
1.5370448039539573`, 1.5383367526814185`, 1.5396272015596313`,
1.5409159960396113`, 1.5422029806041768`, 1.543488566767372`,
1.5447724576795159`, 1.5460546395233454`, 1.5473358120023644`,
1.5486146783003727`, 1.5498925103288654`, 1.5511684372306327`,
1.5524434499426019`, 1.553715956004576`, 1.5549875232788655`,
1.5562575645926233`, 1.5575259227667588`, 1.558792584490729`,
1.5600578265643825`, 1.561322072949664`, 1.5625844405960116`,
1.5638449154115925`, 1.565104213310407`, 1.5663618849690666`,
1.5676182105985772`, 1.5688727387254096`, 1.570125896701889`,
1.5713772316112034`, 1.5726276144117088`, 1.5738762975611376`,
1.5751235639751338`, 1.5763694021965908`, 1.5776132073591036`,
1.5788558566544948`, 1.5800976376069134`, 1.5813369019542272`,
1.5825754246378079`, 1.5838121509878331`, 1.5850475172742635`,
1.5862815126030556`, 1.587513825631967`, 1.5887444440351448`,
1.589973807516043`, 1.5912017548992818`, 1.5924285775032043`,
1.5936535088327561`, 1.594876990623647`, 1.596099315542213`,
1.5973200177124391`, 1.598539542311159`, 1.599757269495414`,
1.600973950656055`, 1.602188811753117`, 1.6034022999930233`,
1.604614711880825`, 1.6058254235540939`, 1.607034731024624`,
1.6082424696504098`, 1.6094489370358336`, 1.6106544326219516`,
1.6118580188207485`, 1.6130599931947955`, 1.6142608104626077`,
1.6154603058964745`, 1.6166586253943669`, 1.6178552917130946`,
1.6190502935570503`, 1.620244089038978`, 1.62143651220807`,
1.6226278669812393`, 1.6238172017877341`, 1.625005605649914`,
1.6261929130229167`, 1.627378326009239`, 1.628562622945091`,
1.6297454785514967`, 1.630927199873937`, 1.6321069843687663`,
1.6332857739204116`, 1.6344629235866028`, 1.6356389009724404`,
1.6368136973503413`, 1.6379869841413035`, 1.6391590715153443`,
1.6403297903304797`, 1.6414991311232427`, 1.6426672452530249`,
1.6438336407296306`, 1.6449987905585717`, 1.6461628474355288`,
1.6473254797134238`, 1.6484863539079835`, 1.64964610792462`,
1.6508048960095845`, 1.6519622226694703`, 1.6531182412159826`,
1.6542731058609736`, 1.6554263183445357`, 1.6565780319470471`,
1.657728564646086`, 1.6588782362268752`, 1.6600262185184589`,
1.661172830148775`, 1.6623182268463959`, 1.6634625652370445`,
1.6646055073652661`, 1.6657468788374137`, 1.6668873324055635`,

```
1.6680260318870532`, 1.6691637972116957`, 1.6702999562045961`,
1.6714351652001656`, 1.672568750194401`, 1.6737012023418547`,
1.6748323467186756`, 1.6759621749087807`, 1.677091013778819`,
1.678218856247106`, 1.6793448547365883`, 1.6804695037325035`,
1.6815931318674986`, 1.6827152257037985`, 1.683836452075713`,
1.6849557891852882`, 1.6860745820871177`, 1.6871918076161494`,
1.6883077963756141`, 1.6894225405156111`, 1.69053620252236`,
1.6916484340631113`, 1.6927590557962604`, 1.693868742562223`,
1.694977487812423`, 1.696084427664553`, 1.6971904106385165`,
1.6982952582469815`, 1.699398963235131`, 1.7005010012626136`,
1.7016020533773024`, 1.702701940164148`, 1.7038001352501524`,
1.7048974955899543`, 1.7059936682692924`, 1.707088819673729`,
1.708182247473395`, 1.7092742906226062`, 1.7103658129424`,
1.711455587651523`, 1.7125443046360387`, 1.7136316071434956`,
1.714718012902788`, 1.7158033403991573`, 1.716887231525561`,
1.7179698541490327`, 1.7190513770165063`, 1.7201314406697117`,
1.7212105667856907`, 1.7222880420628304`, 1.7233643888288146`,
1.7244401320577074`, 1.7255142021659104`, 1.7265869457628977`,
1.7276590670349243`, 1.7287298486605478`, 1.7297991048299202`,
1.7308673631245297`, 1.73193443860373`, 1.7330003244491827`,
1.7340648346060188`, 1.7351281410332717`, 1.7361907759028474`,
1.7372518349013821`, 1.7383116700787644`, 1.739370454907373`,
1.740427822087557`, 1.7414844871613753`, 1.742539721282402`,
1.7435936981180509`, 1.744646592206354`, 1.745698397311969`,
1.7467483797469752`, 1.7477978052017593`, 1.7488461224941643`,
1.7498929603308118`, 1.7509388593422557`, 1.7519838139122035`,
1.753027268903941`, 1.7540697668947929`, 1.7551107514292195`,
1.7561507663813618`, 1.7571898061051845`, 1.7582274964292521`,
1.7592636459145017`, 1.7602991704131692`, 1.7613336957369596`,
1.7623670310826827`, 1.7633986140834423`, 1.764429549823328`,
1.7654590907976433`, 1.7664877879955554`, 1.7675148915189454`,
1.7685415122801211`, 1.769566153361621`, 1.7705904875440768`,
1.7716132021657394`, 1.7726346641393451`, 1.773655242100242`,
1.7746745555119983`, 1.7756927859241882`, 1.7767103038975294`,
1.7777261631257062`, 1.7787412988365652`, 1.779754951361558`,
1.7807673026589315`, 1.781779103229814`, 1.7827894020129311`,
1.7837983815988634`, 1.7848066051781384`, 1.785813307900608`,
1.786819434193556`, 1.7878240276108002`, 1.7888276532540262`,
1.7898301150162008`, 1.7908315982440532`, 1.7918317150237126`,
1.7928310336477684`, 1.793828590588728`, 1.7948259141838847`,
1.7958216562363993`, 1.7968163869429667`, 1.7978099084550736`,
1.7988024077931304`, 1.7997938797497421`, 1.80078412582593`,
1.8017729466689807`, 1.8027611106335426`, 1.8037478380731142`,
1.8047340931685611`, 1.8057185121596253`, 1.8067022546281275`,
1.8076849270626785`, 1.8086661342541885`, 1.8096466504655921`,
1.8106256900918274`, 1.8116038334628732`, 1.8125808801270358`,
1.8135568248236056`, 1.8145312697888922`, 1.8155051907933202`,
```

```
1.8164777977777478`, 1.8174494785990145`, 1.8184198344269689`,
1.8193896485576992`, 1.820357732525545`, 1.8213250677627986`,
1.8222908585109348`, 1.8232558907838465`, 1.824219763840683`,
1.825182472381647`, 1.8261444086435592`, 1.8271049716392351`,
1.8280647529965133`, 1.829022951069878`, 1.8299805573204553`,
1.8309367690520777`, 1.8318919803647118`, 1.8328463867925766`,
1.833799583380789`, 1.8347517654487566`, 1.8357023257218554`,
1.8366524640022612`, 1.8376009698184745`, 1.83854884398223`,
1.8394952762737877`, 1.8404410683043093`, 1.8413856102895878`,
1.8423290991898666`, 1.8432715302758147`, 1.8442131015380299`,
1.8451532000738624`, 1.8460922261778823`, 1.8470301751105622`,
1.8479672456786684`, 1.8489032300012838`, 1.8498379193698482`,
1.8507721250853009`, 1.8517050267565007`, 1.8526364144322018`,
1.853567306043588`, 1.854496673732167`, 1.8554251270895106`,
1.8563528673682441`, 1.8572790686115022`, 1.8582043422520897`,
1.8591288901496914`, 1.8600518838811289`, 1.860973936717274`,
1.861895251124009`, 1.8628154098829541`, 1.8637339941392614`,
1.8646522413327073`, 1.8655689044569033`, 1.8664850157459352`,
1.8673995329836035`, 1.8683136984929962`, 1.8692260521099242`,
1.870137629522669`, 1.8710484271551797`, 1.8719576059558214`,
1.8728664143347398`, 1.8737738035566063`, 1.874680187163908`,
1.8755853512823173`, 1.8764897108686134`, 1.8773928421640644`,
1.878295160930441`, 1.8791960321102381`, 1.880096293040674`,
1.8809953080551238`, 1.8818930723401734`, 1.882790426251217`,
1.883686309918715`, 1.8845813530057265`, 1.8854747043173858`,
1.8863678425028496`, 1.8872592796740058`, 1.8881500722916842`,
1.8890397919414468`, 1.8899282215020978`, 1.8908157823438376`,
1.8917024708403405`, 1.8925878563686787`, 1.8934723614942857`,
1.894355768689007`, 1.895237859625424`, 1.8961194866517823`,
1.8969995748568533`, 1.8978785481964673`, 1.898757261999438`,
1.899634423871973`, 1.9005104589599469`, 1.9013855785098257`,
1.9022597788749271`, 1.9031326246409908`, 1.9040045431321782`,
1.9048757469759905`, 1.9057455836842925`, 1.9066144817648316`,
1.9074822206913848`, 1.9083490132835397`, 1.9092150731482718`,
1.910079744831113`, 1.9109434588075511`, 1.9118062114156733`,
1.9126675628890952`, 1.9135283813767965`, 1.914388227502662`,
1.9152468789594854`, 1.916104331491261`, 1.9169607998732174`,
1.9178164996624403`, 1.918670988929101`, 1.919524263414787`,
1.9203767584865148`, 1.9212280310397378`, 1.9220782970172483`,
1.922927552743148`, 1.9237757945405924`, 1.9246234603038472`,
1.9254694426183752`, 1.9263146207544437`, 1.9271589916138592`,
1.928002330534891`, 1.928844190324126`, 1.9296854539337667`,
1.9305252302717464`, 1.93136440383634`, 1.9322025270446712`,
1.9330395962111206`, 1.9338756076495052`, 1.9347105576731094`,
1.93554444259472`, 1.9363774822176887`, 1.9372092260638751`,
1.9380398937435974`, 1.938869929701413`, 1.9396991071394507`,
1.940526749591878`, 1.9413533015020445`, 1.9421794336762424`,
```

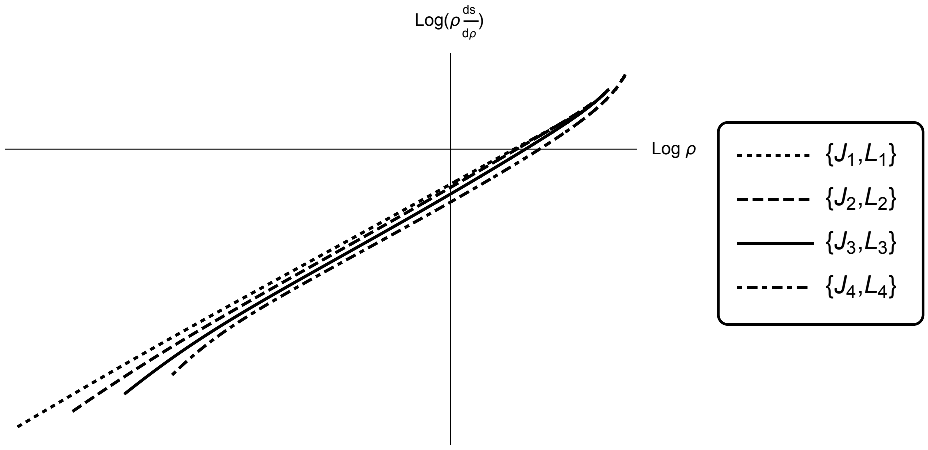
```
1.9430042440494018`, 1.9438277283565557`, 1.9446501077385112`,
1.9454720552893325`, 1.9462924401021335`, 1.947112161245783`,
1.9479309900022512`, 1.9487484705361435`, 1.949565278221125`,
1.9503805035606745`, 1.9511950494223889`, 1.9520086861348434`,
1.9528211832684594`, 1.9536323096371278`, 1.954442971702938`,
1.9552522559931491`, 1.9560608424209651`, 1.956868500176474`,
1.9576747692611796`, 1.9584803312832415`, 1.9592844972456858`,
1.9600879498844548`, 1.9608902282607197`, 1.9616915580531973`,
1.9624917065702678`, 1.9632913593590229`, 1.9640895946736476`,
1.9648866383807209`, 1.9656831777085002`, 1.9664782886690135`,
1.9672721976904284`, 1.9680653628122902`, 1.9688573193561167`,
1.96964852610505`, 1.970438749059394`, 1.9712277534766867`,
1.9720157674600547`, 1.9728030198537132`, 1.9735885794250658`,
1.9743740678125272`, 1.9751578563697167`, 1.9759408711694189`,
1.9767226438280792`, 1.9775038699147587`, 1.978283614322556`,
1.9790625731728824`, 1.9798405102463716`, 1.9806174224090098`,
1.98139330652694`, 1.9821679252804265`, 1.9829419780941924`,
1.9837147592768243`, 1.98448626514723`, 1.9852571967753032`,
1.986026846826115`, 1.9867956821826263`, 1.987563229333435`,
1.9883299558903857`, 1.9890956234413153`, 1.9898604648625198`,
1.9906240051902446`, 1.9913864771204188`, 1.9921481140737984`,
1.9929086767325652`, 1.993668161967856`, 1.9944265666514107`,
1.9951841249273385`, 1.995940596756784`, 1.9966959790128853`,
1.9974505063805799`, 1.9982039382821741`, 1.9989567479321424`,
1.9997079798847008`, 2.000458226261349`, 2.0012076035605935`,
2.0019558703106797`, 2.0027030233896754`, 2.003449657780765`,
2.0041948140200563`, 2.0049393267783167`, 2.005682474676896`,
2.0064244940172795`, 2.007165381682223`, 2.007906215937797`,
2.0086451924264384`, 2.009383269084479`, 2.0101205637466717`,
2.0108567125726666`, 2.011591591854436`, 2.012326042998357`,
2.013059219079385`, 2.0137909958642615`, 2.0145220953978065`,
2.015252031487112`, 2.015981164391072`, 2.0167090069533913`,
2.0174357984004296`, 2.018161900043028`, 2.0188869454111527`,
2.0196105669884052`, 2.020333612670533`, 2.021055472000183`,
2.021776263697834`, 2.022496106832518`, 2.0232149988281374`,
2.023932937109057`, 2.0246495526024986`, 2.0253652087130267`,
2.026079902867282`, 2.026793632492453`, 2.027506272505444`,
2.028217697479697`, 2.028928763108293`, 2.029637994885746`,
2.0303468626934045`, 2.0310546276082593`, 2.031760917728427`,
2.0324663450160787`, 2.03317103035799`, 2.0338743552924363`,
2.0345766865883324`, 2.0352782685860658`, 2.035978728622468`,
2.0366779402588637`, 2.0373760241319756`, 2.038073348801613`,
2.038769788341065`, 2.0394652164854064`, 2.040159010453537`,
2.040852531843434`, 2.0415447859110034`, 2.0422355210019374`,
2.042925977368344`, 2.043614909316898`, 2.0443029360354776`,
2.044990179858575`, 2.0456760151568534`, 2.046361062630381`,
2.047044820699814`, 2.047727911153407`, 2.0484097069463907`,
```

```
    2.0490905804358763`, 2.0497704041130738`, 2.050449050138071`,
    2.0511272680148323`, 2.051803676334308`, 2.0524796520922344`,
    2.053154188952124`, 2.05382803791129`, 2.0545008197936854`,
    2.0551724061897905`, 2.055842794322114`, 2.0565124855802663`,
    2.057180847558142`, 2.057847877237714`, 2.058514834471563`,
    2.0591802022749985`, 2.059844609395776`, 2.060507673962787`,
    2.0611705324839034`, 2.0618311570420547`, 2.062491571332314`,
    2.0631506328362854`, 2.0638089732402696`, 2.0644662097056963`,
    2.0651222126225015`, 2.065777615136807`, 2.0664313976271758`,
    2.0670845753453837`, 2.0677362548056193`, 2.0683877077931117`,
    2.069037402515272`, 2.069686101238138`, 2.070333929444935`,
    2.070980885118453`, 2.071626582539896`, 2.072271146957113`,
    2.072914960058259`, 2.0735576356506917`, 2.0741990430273`,
    2.07483969265511`, 2.0754790692541967`, 2.076117812383833`,
    2.076754892231586`, 2.0773912056669444`, 2.078026622228988`,
    2.078661397455274`, 2.079294499018802`, 2.079926568144271`,
    2.08055760259654`, 2.0811879873941943`, 2.081816687710609`,
    2.0824447340656613`, 2.0830716076599978`, 2.083697823615145`,
    2.0843228625110655`, 2.084946592339284`, 2.0855695289249243`,
    2.086191411100007`, 2.0868119770793205`, 2.087431743621343`,
    2.088050579035468`, 2.088668091296664`, 2.0888661208691746`}[[s]];
```

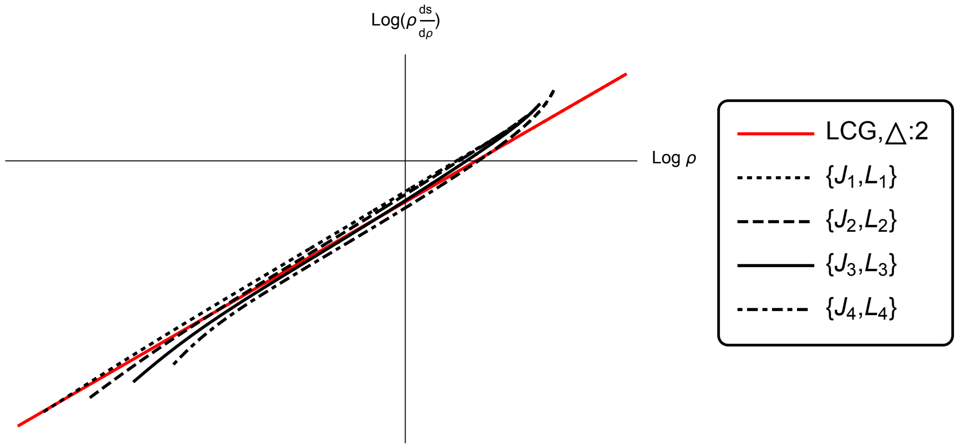
LCG

```
{CSDerivativeCurvatureofLoC[200, 923, 1],
 CSDerivativeCurvatureofLoC[400, 973, 1],
 CSDerivativeCurvatureofLoC[600, 1040, 1],
 CSDerivativeCurvatureofLoC[800, 1112, 1]}
```

```
ListLinePlot[{Table[{Log[CS2rho200[s]],
    Log[.001 * CS2rho200[s] / (CS2rho200[s + 1] - CS2rho200[s])]},
   {s, 1, 921, 1}], Table[{Log[CS2rho400[s]],
    Log[.001 * CS2rho400[s] / (CS2rho400[s + 1] - CS2rho400[s])]},
   {s, 1, 973, 1}], Table[{Log[CS2rho600[s]],
    Log[.001 * CS2rho600[s] / (CS2rho600[s + 1] - CS2rho600[s])]},
   {s, 1, 1038, 1}], Table[{Log[CS2rho800[s]],
    Log[.001 * CS2rho800[s] / (CS2rho800[s + 1] - CS2rho800[s])]}, {s, 1, 1110, 1}]},
 AxesLabel → {"Log ρ", "Log(ρ ds/dρ)"}, PlotStyle →
  {{Dotted, Black}, {Dashed, Black}, {Black}, {DotDashed, Black}}, Ticks → None,
 PlotLegends → LineLegend[{{{"J1,L1"}, {"J2,L2"}, {"J3,L3"}, {"J4,L4"}}},
   LegendFunction → (Framed[#, RoundingRadius → 5] &), LegendMargins → 5]]
```



```
ListLinePlot[
 {Transpose[{LogRhoalpha2, LogrhoDsDrhoalpha2}], Table[{Log[CS2rho200[s]],
    Log[.001 * CS2rho200[s] / (CS2rho200[s + 1] - CS2rho200[s])]},
   {s, 1, 921, 1}], Table[{Log[CS2rho400[s]],
    Log[.001 * CS2rho400[s] / (CS2rho400[s + 1] - CS2rho400[s])]},
   {s, 1, 973, 1}], Table[{Log[CS2rho600[s]],
    Log[.001 * CS2rho600[s] / (CS2rho600[s + 1] - CS2rho600[s])]},
   {s, 1, 1038, 1}], Table[{Log[CS2rho800[s]],
    Log[.001 * CS2rho800[s] / (CS2rho800[s + 1] - CS2rho800[s])]}, {s, 1, 1110, 1}]},
 AxesLabel → {"Log ρ", "Log(ρ ds/dρ)"}, PlotStyle →
  {Red, {Dotted, Black}, {Dashed, Black}, {Black}, {DotDashed, Black}},
 Ticks → None, PlotLegends →
  LineLegend[{{"LCG,△:2", {"J1,L1"}, {"J2,L2"}, {"J3,L3"}, {"J4,L4"}}},
   LegendFunction → (Framed[#, RoundingRadius → 5] &), LegendMargins → 5]]
```




```
LogRhoalpha2 = Table[Log[Rho[s, 2, 2, 1]], {s, -0.245, 2, 0.001}]
LogrhoDsDrhoalpha2 =
 Table[Log[Rho[s, 2, 2, 1] * DsDrho[s, 2, 2, 1]], {s, -0.245, 2, 0.001}]
```

Figure 7

(a)

Set up

```
kernelcodeHP = "
#define min 0
#define max 1
#define uuu 0
#define vvv 1
```

```
#define sign1 1
#define h 0.001
#define n 1201
#define n_j 302


#define inidxdu1 (tu1)
#define inidxdu2 (0)
#define inidxdu3 (tu2)
#define inidxdv1 (0)
#define inidxdv2 (tv1)
#define inidxdv3 (tv2)

#define inid2xdu21 (dtu1)
#define inid2xdu22 (0)
#define inid2xdu23 (dtu2)
#define inid2xdv21 (0)
#define inid2xdv22 (dtv1)
#define inid2xdv23 (dtv2)
#define inid2xdudv1 (0)
#define inid2xdudv2 (0)
#define inid2xdudv3 (0)

#define inid3xdu31 (d2tu1)
#define inid3xdu32 (0)
#define inid3xdu33 (d2tu2)
#define inid3xdv31 (0)
#define inid3xdv32 (d2tv1)
#define inid3xdv33 (d2tv2)
#define inid3xdu2dv1 (0)
#define inid3xdu2dv2 (0)
#define inid3xdu2dv3 (0)
#define inid3xdudv21 (0)
#define inid3xdudv22 (0)
#define inid3xdudv23 (0)



#define repiniSurNorvec iniSurNorvec(xu1,xu2,tu1,tu2,xv1,xv2,tv1,tv2,normv_C)
#define repiniDSurNvDu
    iniDSurNvDu(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,dnormvdu_C)
#define repiniDSurNvDv
    iniDSurNvDv(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,dnormvdv_C)
#define repiniScalarSurNor
    iniScalarSurNor(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,Scal)
#define repinicoefE inicoefE(xu1,xu2,tu1,tu2,xv1,xv2,tv1,tv2)
#define repiniDEDu
```

```
    iniDEDu(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniDEDv
    iniDEDv(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repinicoefF inicoefF(xu1,xu2,tu1,tu2,xv1,xv2,tv1,tv2)
#define repiniDFDu
    iniDFDu(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniDFDv
    iniDFDv(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repinicoefG inicoefG(xu1,xu2,tu1,tu2,xv1,xv2,tv1,tv2)
#define repiniDGDu
    iniDGDu(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniDGDv
    iniDGDv(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repinicoefL
    inicoefL(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repinicoefM
    inicoefM(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repinicoefN
    inicoefN(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniGaussCurva
    iniGaussCurva(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniMeanCurva
    iniMeanCurva(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniMaxPrinci
    iniMaxPrinci(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniMinPrinci
    iniMinPrinci(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2)
#define repiniPrinci
    iniPrinci(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)

#define repiniDcoefL_2
    iniDcoefL_2(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
    dtv2,d2tv1,d2tv2,CoeffL)
#define repiniDcoefM_2
    iniDcoefM_2(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
    dtv2,d2tv1,d2tv2,CoeffM)
#define repiniDcoefN_2
    iniDcoefN_2(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
    dtv2,d2tv1,d2tv2,CoeffN)
#define repiniDGaussCurva_2
    iniDGaussCurva_2(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,
    dtv1,dtv2,d2tv1,d2tv2,GCurva)
#define repiniDMeanCurva_2
    iniDMeanCurva_2(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,
    dtv1,dtv2,d2tv1,d2tv2,MCurva)
#define repiniDPrinci_2
    iniDPrinci_2(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
```

```
    dtv2,d2tv1,d2tv2,optimum,PCurva)
#define repiniEta
    iniEta(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniMiu
    iniMiu(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDuDs
    iniDuDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDvDs
    iniDvDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDEDs
    iniDEDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDFDs
    iniDFDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDGDs
    iniDGDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDLDs
    iniDLDs(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2,optimum)
#define repiniDMDs
    iniDMDs(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2,optimum)
#define repiniDNDs
    iniDNDs(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    d2tv1,d2tv2,optimum)
#define repiniDPrinciDs
    iniDPrinciDs(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
    dtv2,d2tv1,d2tv2,optimum)
#define repiniDvDt
    iniDvDt(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDuDt
    iniDuDt(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum)
#define repiniDSurDs
    iniDSurDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum,tangv)
#define repiniSurNor iniSurNor(xu1,xu2,tu1,tu2,xv1,xv2,tv1,tv2,Snor_C)
#define repiniBinormalSur
    iniBinormalSur(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,
    optimum,binormv_C)
#define repiniAlp1
    iniAlp1(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum,Alp1_C)
#define repiniBeta1
    iniBeta1(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,dtv2
    ,d2tv1,d2tv2,optimum)


#define opt1 (float xu1,float xu2,float tu1,float
    tu2,float xv1,float xv2,float tv1,float tv2,float normv_C[])
#define opt2 (float xu1,float xu2,float tu1,float tu2,float
```

```
    xv1,float xv2,float tv1,float tv2)
#define opt3 (float xu1,float xu2,float tu1,float tu2,float dtu1,float
    dtu2,float xv1,float xv2,float tv1,float tv2,float dtv1,float dtv2)
#define opt4 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float xv1,float xv2,float
    tv1,float tv2,float dtv1,float dtv2,mint optimum)
 #define opt5 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float xv1,float xv2,float tv1,float
    tv2,float dtv1,float dtv2,mint optimum,float tangv_C[])
#define opt6 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float xv1,float xv2,float tv1,float
    tv2,float dtv1,float dtv2,float dnormv_C[])
#define opt7 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float xv1,float xv2,float
    tv1,float tv2,float dtv1,float dtv2,float Scal[])
#define opt8 (float xu1,float xu2,float tu1,float tu2,float dtu1,float
    dtu2,float d2tu1,float d2tu2,float xv1,float xv2,float tv1,float
    tv2,float dtv1,float dtv2,float d2tv1,float d2tv2,float coeff[])
#define opt9 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float d2tu1,float d2tu2,float xv1,float
    xv2,float tv1,float tv2,float dtv1,float dtv2,float
    d2tv1,float d2tv2,mint optimum,float coeff[])
#define opt10 (float xu1,float xu2,float tu1,float tu2,float dtu1,float
    dtu2,float d2tu1,float d2tu2,float xv1,float xv2,float tv1,float
    tv2,float dtv1,float dtv2,float d2tv1,float d2tv2,mint optimum)
#define opt11 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float xv1,float xv2,float tv1,float
    tv2,float dtv1,float dtv2,mint optimum,float Alph_C[])
#define opt12 (float xu1,float xu2,float tu1,float tu2,float
    xv1,float xv2,float tv1,float tv2,float Snor_C[])
#define opt13 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float xv1,float xv2,float tv1,float
    tv2,float dtv1,float dtv2,mint optimum,float binormv_C[])
#define opt14 (float xu1,float xu2,float tu1,float tu2,float
    dtu1,float dtu2,float d2tu1,float d2tu2,float
    xv1,float xv2,float tv1,float tv2,float dtv1,float
    dtv2,float d2tv1,float d2tv2,mint optimum,float x[])
```

```
__device__ void iniSurNorvec opt1
{
    normv_C[0] =inidxdu2*inidxdv3-inidxdu3*inidxdv2;
    normv_C[1] =inidxdu3*inidxdv1-inidxdu1*inidxdv3;
    normv_C[2] =inidxdu1*inidxdv2-inidxdu2*inidxdv1;
}
```

```
__device__ void iniDSurNvDu opt6
{
    dnormv_C[0]
    =(inid2xdu22*inidxdv3-inid2xdu23*inidxdv2)+(inidxdu2*inid2xdudv3-
    inidxdu3*inid2xdudv2);
    dnormv_C[1]
    =(inid2xdu23*inidxdv1-inid2xdu21*inidxdv3)+(inidxdu3*inid2xdudv1-
    inidxdu1*inid2xdudv3);
    dnormv_C[2]
    =(inid2xdu21*inidxdv2-inid2xdu22*inidxdv1)+(inidxdu1*inid2xdudv2-
    inidxdu2*inid2xdudv1);
}
__device__ void iniDSurNvDv opt6
{
    dnormv_C[0]
    =(inid2xdudv2*inidxdv3-inid2xdudv3*inidxdv2)+(inidxdu2*inid2xdv23-
    inidxdu3*inid2xdv22);
    dnormv_C[1]
    =(inid2xdudv3*inidxdv1-inid2xdudv1*inidxdv3)+(inidxdu3*inid2xdv21-
    inidxdu1*inid2xdv23);
    dnormv_C[2]
    =(inid2xdudv1*inidxdv2-inid2xdudv2*inidxdv1)+(inidxdu1*inid2xdv22-
    inidxdu2*inid2xdv21);
}

__device__ void iniScalarSurNor opt7
{
float ScalSNor,DScalSNorDu,DScalSNorDv;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];

repiniSurNorvec;
repiniDSurNvDu;
repiniDSurNvDv;



ScalSNor =
    sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
DScalSNorDu
    =
    (normv_C[0]*dnormvdu_C[0]+normv_C[1]*dnormvdu_C[1]+normv_C[2]*dnormvdu_C
    [2])/ScalSNor;
DScalSNorDv =
    (normv_C[0]*dnormvdv_C[0]+normv_C[1]*dnormvdv_C[1]+normv_C[2]*dnormvdv_C
    [2])/ScalSNor;
```

```
Scal[0] = ScalSNor;
Scal[1] = DScalSNorDu;
Scal[2] = DScalSNorDv;
}

__device__ float inicoefE opt2
{
    return inidxdu1*inidxdu1+inidxdu2*inidxdu2+inidxdu3*inidxdu3;
}
__device__ float iniDEDu opt3
{
    return 2*(inidxdu1*inid2xdu21+inidxdu2*inid2xdu22+inidxdu3*inid2xdu23);
}
__device__ float iniDEDv opt3
{
    return 2*(inidxdu1*inid2xdudv1+inidxdu2*inid2xdudv2+inidxdu3*inid2xdudv3);
}


__device__ float inicoefF opt2
{
    return (inidxdu1*inidxdv1+inidxdu2*inidxdv2+inidxdu3*inidxdv3);
}
__device__ float iniDFDu opt3
{
    return inidxdv1*inid2xdu21+inidxdv2*inid2xdu22+inidxdv3*inid2xdu23
     + inidxdu1*inid2xdudv1+inidxdu2*inid2xdudv2+inidxdu3*inid2xdudv3;
}
__device__ float iniDFDv opt3
{
    return
     inidxdv1*inid2xdudv1+inidxdv2*inid2xdudv2+inidxdv3*inid2xdudv3
     + inidxdu1*inid2xdv21+inidxdu2*inid2xdv22+inidxdu3*inid2xdv23;
}

__device__ float inicoefG opt2
{
    return inidxdv1*inidxdv1+inidxdv2*inidxdv2+inidxdv3*inidxdv3;
}
__device__ float iniDGDu opt3
{
    return
     2*(inidxdv1*inid2xdudv1+inidxdv2*inid2xdudv2+inidxdv3*inid2xdudv3);
}
__device__ float iniDGDv opt3
{
```

```
    return
    2*(inidxdv1*inid2xdv21+inidxdv2*inid2xdv22+inidxdv3*inid2xdv23);
}

__device__ float inicoefL opt3
{
float normv_C[3];
repiniSurNorvec;

    return
    (normv_C[0]*inid2xdu21+normv_C[1]*inid2xdu22+normv_C[2]*inid2xdu23)/sqrt
    (normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float inicoefM opt3
{
float normv_C[3];
repiniSurNorvec;

    return
    (normv_C[0]*inid2xdudv1+normv_C[1]*inid2xdudv2+normv_C[2]*inid2xdudv3)/
    sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float inicoefN opt3
{
float normv_C[3];
repiniSurNorvec;

    return
    (normv_C[0]*inid2xdv21+normv_C[1]*inid2xdv22+normv_C[2]*inid2xdv23)/sqrt
    (normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float iniGaussCurva opt3
{
    return
    (repinicoefL*repinicoefN-pow(repinicoefM,2))/(repinicoefE*repinicoefG-
    pow(repinicoefF,2));
}
__device__ float iniMeanCurva opt3
{
    return
    0.5*(repinicoefE*repinicoefN+repinicoefG*repinicoefL-2*repinicoefF*
    repinicoefM)/(repinicoefE*repinicoefG-pow(repinicoefF,2));
}
__device__ float iniMaxPrinci opt3
{
    return repiniMeanCurva+sqrt(pow(repiniMeanCurva,2)-repiniGaussCurva);
}
```

```
__device__ float iniMinPrinci opt3
{
    return repiniMeanCurva-sqrt(pow(repiniMeanCurva,2)-repiniGaussCurva);
}
__device__ float iniPrinci opt4
{
    if(optimum==max)
    {return repiniMaxPrinci;}
    else if(optimum==min)
    {return repiniMinPrinci;}
    else
    {return 0;}
}
__device__ void iniDcoefL_2 opt8
{

float CoeL,DCoeLDu,DCoeLDv;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float Scal[3];

repiniSurNorvec;
repiniDSurNvDu;
repiniDSurNvDv;
repiniScalarSurNor;

CoeL =
    (normv_C[0]*inid2xdu21+normv_C[1]*inid2xdu22+normv_C[2]*inid2xdu23)/Scal[0];
DCoeLDu
    =
    ((dnormvdu_C[0]*inid2xdu21+dnormvdu_C[1]*inid2xdu22+dnormvdu_C[2]*
    inid2xdu23)+(normv_C[0]*inid3xdu31+normv_C[1]*inid3xdu32+normv_C[2]*
    inid3xdu33)-Scal[1]*CoeL)/Scal[0];
DCoeLDv =
    ((dnormvdv_C[0]*inid2xdu21+dnormvdv_C[1]*inid2xdu22+dnormvdv_C[2]*
    inid2xdu23)+(normv_C[0]*inid3xdu2dv1+normv_C[1]*inid3xdu2dv2+normv_C[2]*
    inid3xdu2dv3)-Scal[2]*CoeL)/Scal[0];

coeff[0] = CoeL;
coeff[1] = DCoeLDu;
coeff[2] = DCoeLDv;

}
__device__ void iniDcoefM_2 opt8
{
```

```
float CoeM,DCoeMDu,DCoeMDv;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float Scal[3];
repiniSurNorvec;
repiniDSurNvDu;
repiniDSurNvDv;
repiniScalarSurNor;

CoeM =
    (normv_C[0]*inid2xdudv1+normv_C[1]*inid2xdudv2+normv_C[2]*inid2xdudv3)/
    Scal[0];
DCoeMDu =
    ((dnormvdu_C[0]*inid2xdudv1+dnormvdu_C[1]*inid2xdudv2+dnormvdu_C[2]*
    inid2xdudv3)+(normv_C[0]*inid3xdu2dv1+normv_C[1]*inid3xdu2dv2+normv_C[2]
    *inid3xdu2dv3)-Scal[1]*CoeM)/Scal[0];
DCoeMDv =
    ((dnormvdv_C[0]*inid2xdudv1+dnormvdv_C[1]*inid2xdudv2+dnormvdv_C[2]*
    inid2xdudv3)+(normv_C[0]*inid3xdudv21+normv_C[1]*inid3xdudv22+normv_C[2]
    *inid3xdudv23)-Scal[2]*CoeM)/Scal[0];


coeff[0] = CoeM;
coeff[1] = DCoeMDu;
coeff[2] = DCoeMDv;
}
__device__ void iniDcoefN_2 opt8
{
float CoeN,DCoeNDu,DCoeNDv;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float Scal[3];
repiniSurNorvec;
repiniDSurNvDu;
repiniDSurNvDv;
repiniScalarSurNor;

CoeN =
    (normv_C[0]*inid2xdv21+normv_C[1]*inid2xdv22+normv_C[2]*inid2xdv23)/Scal[0];
DCoeNDu
    =
    ((dnormvdu_C[0]*inid2xdv21+dnormvdu_C[1]*inid2xdv22+dnormvdu_C[2]*
    inid2xdv23)+(normv_C[0]*inid3xdudv21+normv_C[1]*inid3xdudv22+normv_C[2]*
    inid3xdudv23)-Scal[1]*CoeN)/Scal[0];
DCoeNDv =
```

```
    ((dnormvdv_C[0]*inid2xdv21+dnormvdv_C[1]*inid2xdv22+dnormvdv_C[2]*
    inid2xdv23)+(normv_C[0]*inid3xdv31+normv_C[1]*inid3xdv32+normv_C[2]*
    inid3xdv33)-Scal[2]*CoeN)/Scal[0];

coeff[0] = CoeN;
coeff[1] = DCoeNDu;
coeff[2] = DCoeNDv;

}
__device__ void iniDGaussCurva_2 opt8
{
float GCurv,DGCurvDu,DGCurvDv;
float CoeffL[3];
float CoeffM[3];
float CoeffN[3];
repiniDcoefL_2;
repiniDcoefM_2;
repiniDcoefN_2;

GCurv=
    (CoeffL[0]*CoeffN[0]-pow(CoeffM[0],2))/(repinicoefE*repinicoefG-pow(
    repinicoefF,2));
DGCurvDu =
    ((pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repinicoefG*repiniDEDu-2*
    repinicoefF*repiniDFDu+repinicoefE*repiniDGDu)+(-pow(repinicoefF,2)+
    repinicoefE*repinicoefG)*(CoeffN[0]*CoeffL[1]-2*CoeffM[0]*CoeffM[1]+
    CoeffL[0]*CoeffN[1]))/(pow(pow(repinicoefF,2)-repinicoefE*repinicoefG,2)
    );
DGCurvDv
    =((pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repinicoefG*repiniDEDv-2*
    repinicoefF*repiniDFDv+repinicoefE*repiniDGDv)+(-pow(repinicoefF,2)+
    repinicoefE*repinicoefG)*(CoeffN[0]*CoeffL[2]-2*CoeffM[0]*CoeffM[2]+
    CoeffL[0]*CoeffN[2]))/(pow(pow(repinicoefF,2)-repinicoefE*repinicoefG,2)
    );


coeff[0] = GCurv;
coeff[1] = DGCurvDu;
coeff[2] = DGCurvDv;

}
__device__ void iniDMeanCurva_2 opt8
{
float MCurv,DMCurvDu,DMCurvDv;
float CoeffL[3];
float CoeffM[3];
float CoeffN[3];
```

```
repiniDcoefL_2;
repiniDcoefM_2;
repiniDcoefN_2;

MCurv =
    0.5*(repinicoefE*CoeffN[0]+repinicoefG*CoeffL[0]-2*repinicoefF*CoeffM[0]
    )/(repinicoefE*repinicoefG-pow(repinicoefF,2));
DMCurvDu =
    0.5*(-(repinicoefG*CoeffL[0]-2*repinicoefF*CoeffM[0]+repinicoefE*CoeffN[
    0])*(repinicoefG*repiniDEDu-2*repinicoefF*repiniDFDu+repinicoefE*
    repiniDGDu)+(-pow(repinicoefF,2)+repinicoefE*repinicoefG)*(CoeffN[0]*
    repiniDEDu-2*CoeffM[0]*repiniDFDu+CoeffL[0]*repiniDGDu+repinicoefG*
    CoeffL[1]-2*repinicoefF*CoeffM[1]+repinicoefE*CoeffN[1]))/(pow(pow(
    repinicoefF,2)-repinicoefE*repinicoefG,2));
DMCurvDv =
    0.5*(-(repinicoefG*CoeffL[0]-2*repinicoefF*CoeffM[0]+repinicoefE*CoeffN[
    0])*(repinicoefG*repiniDEDv-2*repinicoefF*repiniDFDv+repinicoefE*
    repiniDGDv)+(-pow(repinicoefF,2)+repinicoefE*repinicoefG)*(CoeffN[0]*
    repiniDEDv-2*CoeffM[0]*repiniDFDv+CoeffL[0]*repiniDGDv+repinicoefG*
    CoeffL[2]-2*repinicoefF*CoeffM[2]+repinicoefE*CoeffN[2]))/(pow(pow(
    repinicoefF,2)-repinicoefE*repinicoefG,2));



coeff[0] = MCurv;
coeff[1] = DMCurvDu;
coeff[2] = DMCurvDv;
}
__device__ void iniDPrinci_2 opt9
{
float Princip,DPrincipDu,DPrincipDv;
float GCurva[3];
float MCurva[3];
repiniDGaussCurva_2;
repiniDMeanCurva_2;

if(optimum==max)
{Princip = MCurva[0]+sqrt(pow(MCurva[0],2)-GCurva[0]);
DPrincipDu =
    MCurva[1]+(2*MCurva[0]*MCurva[1]-GCurva[1])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
DPrincipDv =
    MCurva[2]+(2*MCurva[0]*MCurva[2]-GCurva[2])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));

}
else if(optimum==min)
```

```
{Princip = MCurva[0]-sqrt(pow(MCurva[0],2)-GCurva[0]);
DPrincipDu =
    MCurva[1]-(2*MCurva[0]*MCurva[1]-GCurva[1])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
DPrincipDv =
    MCurva[2]-(2*MCurva[0]*MCurva[2]-GCurva[2])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));

}
else
{Princip = 0;
DPrincipDu = 0;
DPrincipDv = 0;
}

coeff[0] = Princip;
coeff[1] = DPrincipDu;
coeff[2] = DPrincipDv;

}

__device__ float iniEta opt4
{
    return
    sign1/sqrt(repinicoefE*pow(repinicoefM-repiniPrinci*repinicoefF,2)-2*
    repinicoefF*(repinicoefM-repiniPrinci*repinicoefF)*(repinicoefL-
    repiniPrinci*repinicoefE)+repinicoefG*pow(repinicoefL-repiniPrinci*
    repinicoefE,2));
}
__device__ float iniMiu opt4
{
    return
    sign1/sqrt(repinicoefE*pow(repinicoefN-repiniPrinci*repinicoefG,2)-2*
    repinicoefF*(repinicoefM-repiniPrinci*repinicoefF)*(repinicoefN-
    repiniPrinci*repinicoefG)+repinicoefG*pow(repinicoefM-repiniPrinci*
    repinicoefF,2));
}
__device__ float iniDuDs opt4
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-
    repiniPrinci*repinicoefG))
    {return repiniEta*(repinicoefM-repiniPrinci*repinicoefF);}
else
    {return repiniMiu*(repinicoefN-repiniPrinci*repinicoefG);}
}
__device__ float iniDvDs opt4
{
```

```
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
    repinicoefG))
    {return -repiniEta*(repinicoefL-repiniPrinci*repinicoefE);}
else
    {return -repiniMiu*(repinicoefM-repiniPrinci*repinicoefF);}
}


__device__ float iniDEDs opt4
{
    return repiniDEDu*repiniDuDs+repiniDEDv*repiniDvDs;
}
__device__ float iniDFDs opt4
{
    return repiniDFDu*repiniDuDs+repiniDFDv*repiniDvDs;
}
__device__ float iniDGDs opt4
{
    return repiniDGDu*repiniDuDs+repiniDGDv*repiniDvDs;
}
__device__ float iniDLDs opt10
{
float CoeffL[3];
repiniDcoefL_2;

    return CoeffL[1]*repiniDuDs+CoeffL[2]*repiniDvDs;
}
__device__ float iniDMDs opt10
{
float CoeffM[3];
repiniDcoefM_2;

    return CoeffM[1]*repiniDuDs+CoeffM[2]*repiniDvDs;
}
__device__ float iniDNDs opt10
{
float CoeffN[3];
repiniDcoefN_2;

    return CoeffN[1]*repiniDuDs+CoeffN[2]*repiniDvDs;
}
__device__ float iniDPrinciDs opt10
{
float PCurva[3];
repiniDPrinci_2;

    return PCurva[1]*repiniDuDs+PCurva[2]*repiniDvDs;
```

```
}
__device__ float iniDuDt opt4
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
    repinicoefG))
    {return (repinicoefM-repiniPrinci*repinicoefF);}
else
    {return (repinicoefN-repiniPrinci*repinicoefG);}
}
__device__ float iniDvDt opt4
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
    repinicoefG))
    {return (repinicoefL-repiniPrinci*repinicoefE);}
else
    {return (repinicoefM-repiniPrinci*repinicoefF);}
}
__device__ void iniDSurDs opt5
{
    tangv_C[0] =inidxdu1*repiniDuDs+inidxdv1*repiniDvDs;
    tangv_C[1] =inidxdu2*repiniDuDs+inidxdv2*repiniDvDs;
    tangv_C[2] =inidxdu3*repiniDuDs+inidxdv3*repiniDvDs;
}

__device__ void iniAlp1 opt11
{
    Alph_C[0]=(inid2xdu21*pow(repiniDuDs,2)+2*inid2xdudv1*repiniDuDs*
    repiniDvDs+inid2xdv21*pow(repiniDvDs,2));

    Alph_C[1]=(inid2xdu22*pow(repiniDuDs,2)+2*inid2xdudv2*repiniDuDs*
    repiniDvDs+inid2xdv22*pow(repiniDvDs,2));

    Alph_C[2]=(inid2xdu23*pow(repiniDuDs,2)+2*inid2xdudv3*repiniDuDs*
    repiniDvDs+inid2xdv23*pow(repiniDvDs,2));
}
__device__ void iniSurNor opt12
{
    float normv_C[3];
    repiniSurNorvec;
    Snor_C[0]
    =normv_C[0]/sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*
    normv_C[2]);
    Snor_C[1]
    =normv_C[1]/sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*
    normv_C[2]);
    Snor_C[2]
    =normv_C[2]/sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*
```

```
    normv_C[2]);
}

__device__ void iniBinormalSur opt13
{
    float Snor_C[3];
    float tangv[3];
    repiniSurNor;
    repiniDSurDs;

    binormv_C[0] =Snor_C[1]*tangv[2] -Snor_C[2]*tangv[1];
    binormv_C[1] =Snor_C[2]*tangv[0] -Snor_C[0]*tangv[2];
    binormv_C[2] =Snor_C[0]*tangv[1] -Snor_C[1]*tangv[0];
}
__device__ float iniBeta1 opt10
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
    repinicoefG))
    {return
    (-(repiniDLDs-repiniDPrinciDs*repinicoefE-repiniPrinci*repiniDEDs)*
    repiniDuDs-(repiniDMDs-repiniDPrinciDs*repinicoefF-repiniPrinci*
    repiniDFDs)*repiniDvDs);}
else
    {return
    (-(repiniDMDs-repiniDPrinciDs*repinicoefF-repiniPrinci*repiniDFDs)*
    repiniDuDs-(repiniDNDs-repiniDPrinciDs*repinicoefG-repiniPrinci*
    repiniDGDs)*repiniDvDs);}
}

__device__ void LUDecomposition3x3 opt14
{
    mint rc=3;
    float binormv_C[3];
    float Alp1_C[3];
    repiniBinormalSur;
    repiniAlp1;
    float matrixA[3][3]={

    {repinicoefE,repinicoefF,-(binormv_C[0]*inidxdu1+binormv_C[1]*inidxdu2+
    binormv_C[2]*inidxdu3)},

    {repinicoefF,repinicoefG,-(binormv_C[0]*inidxdv1+binormv_C[1]*inidxdv2+
    binormv_C[2]*inidxdv3)},
    {repiniDvDt,repiniDuDt,0}};

    float
    matrixB[3]={-(Alp1_C[0]*inidxdu1+Alp1_C[1]*inidxdu2+Alp1_C[2]*inidxdu3),
```

```
-(Alp1_C[0]*inidxdv1+Alp1_C[1]*inidxdv2+Alp1_C[2]*inidxdv3),repiniBeta1}
;
float Lower[3][3];
float Upper[3][3];
float y[3];
float sum;
for(mint ii=0; ii<rc; ii++)
        {
            for(mint jj=0; jj<rc; jj++)
            {
                sum = 0;
                if(ii==jj)
                {
                Lower[ii][jj]=1;
                for (mint kk = 0; kk < ii; kk++)
                    {sum += Lower[ii][kk] * Upper[kk][jj];}
                Upper[ii][jj] = matrixA[ii][jj] - sum;
                }
                else if(ii < jj)
                {
                Lower[ii][jj]=0;
                for (mint kk = 0; kk < ii; kk++)
                    {sum += Lower[ii][kk] * Upper[kk][jj];}
                Upper[ii][jj] = matrixA[ii][jj] - sum;
                }
                else
                {
                Upper[ii][jj]=0;
                for (mint kk = 0; kk < jj; kk++)
                    {sum += Lower[ii][kk] * Upper[kk][jj];}
                Lower[ii][jj] =(matrixA[ii][jj] - sum)/Upper[jj][jj];
                }
            }
        }
for (mint ii = 0; ii < rc; ii++)
{
    sum = 0;
    for (mint jj = 0; jj < ii; jj++)
        {sum += Lower[ii][jj] * y[jj];}
    y[ii] = matrixB[ii] - sum;
}
for (mint ii = rc - 1; ii >= 0; ii--)
{
    sum = 0;
    for (mint jj = ii + 1; jj < rc; jj++)
        {sum += Upper[ii][jj] * x[jj];}
    x[ii] = (y[ii] - sum)/Upper[ii][ii];
```

```
    }
}


__device__ float RK4_Init(float x1, float t1, float SSize)

{
float k1,k2,k3,k4;

k1=SSize*x1;
k2=SSize*(x1+0.5*SSize*t1);
k3=SSize*(x1+0.5*SSize*t1);
k4=SSize*(x1+SSize*t1);

return (1.0/6.0)*(k1+2*k2+2*k3+k4);
}


__device__ void RK4_func_LoC( float xu1, float xu2,
    float tu1, float tu2, float dtu1, float dtu2, float xv1,
    float xv2, float tv1, float tv2, float dtv1, float dtv2,
    mint optimum, float& increment_u, float& increment_v)
{

float u_k1,u_k2,u_k3,u_k4;
float v_k1,v_k2,v_k3,v_k4;
float newxu1,newxu2,newtu1,newtu2,newdtu1,newdtu2;
float newxv1,newxv2,newtv1,newtv2,newdtv1,newdtv2;


u_k1=h*iniDuDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum);
v_k1=h*iniDvDs(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum);

newxu1=xu1+0.5*u_k1*tu1;
newxu2=xu2+0.5*u_k1*tu2;
newtu1=tu1+0.5*u_k1*dtu1;
newtu2=tu2+0.5*u_k1*dtu2;
newdtu1=0;
newdtu2=2;
newxv1=xv1+0.5*v_k1*tv1;
newxv2=xv2+0.5*v_k1*tv2;
newtv1=tv1+0.5*v_k1*dtv1;
newtv2=tv2+0.5*v_k1*dtv2;
newdtv1=0;
newdtv2=-2;
u_k2=h*iniDuDs(newxu1,newxu2,newtu1,newtu2,newdtu1,newdtu2,newxv1,newxv2,newtv1
    ,newtv2,newdtv1,newdtv2,optimum);
```

```
v_k2=h*iniDvDs(newxu1,newxu2,newtu1,newtu2,newdtu1,newdtu2,newxv1,newxv2
     ,newtv1,newtv2,newdtv1,newdtv2,optimum);

newxu1=xu1+0.5*u_k2*tu1;
newxu2=xu2+0.5*u_k2*tu2;
newtu1=tu1+0.5*u_k2*dtu1;
newtu2=tu2+0.5*u_k2*dtu2;
newdtu1=0;
newdtu2=2;
newxv1=xv1+0.5*v_k2*tv1;
newxv2=xv2+0.5*v_k2*tv2;
newtv1=tv1+0.5*v_k2*dtv1;
newtv2=tv2+0.5*v_k2*dtv2;
newdtv1=0;
newdtv2=-2;

u_k3=h*iniDuDs(newxu1,newxu2,newtu1,newtu2,newdtu1,newdtu2,newxv1,newxv2,newtv1
     ,newtv2,newdtv1,newdtv2,optimum);
v_k3=h*iniDvDs(newxu1,newxu2,newtu1,newtu2,newdtu1,newdtu2,newxv1,newxv2
     ,newtv1,newtv2,newdtv1,newdtv2,optimum);

newxu1=xu1+u_k3*tu1;
newxu2=xu2+u_k3*tu2;
newtu1=tu1+u_k3*dtu1;
newtu2=tu2+u_k3*dtu2;
newdtu1=0;
newdtu2=2;
newxv1=xv1+v_k3*tv1;
newxv2=xv2+v_k3*tv2;
newtv1=tv1+v_k3*dtv1;
newtv2=tv2+v_k3*dtv2;
newdtv1=0;
newdtv2=-2;

u_k4=h*iniDuDs(newxu1,newxu2,newtu1,newtu2,newdtu1,newdtu2,newxv1,newxv2,newtv1
     ,newtv2,newdtv1,newdtv2,optimum);
v_k4=h*iniDvDs(newxu1,newxu2,newtu1,newtu2,newdtu1,newdtu2,newxv1,newxv2
     ,newtv1,newtv2,newdtv1,newdtv2,optimum);

increment_u=(1.0/6.0)*(u_k1+2*u_k2+2*u_k3+u_k4);
increment_v=(1.0/6.0)*(v_k1+2*v_k2+2*v_k3+v_k4);

}

__device__ void ReturnDetail( mint k, float
     x[], float y[], float z[], float GeoCur[], mint optimum)
{
```

```
float iniSSize_u,iniSSize_v,SSize_u,SSize_v;

float xu1, xu2, tu1, tu2, dtu1, dtu2, d2tu1, d2tu2;
float xv1, xv2, tv1, tv2, dtv1, dtv2, d2tv1, d2tv2;
float u_s,v_s;
float coef[3];

u_s=0;
v_s=0;
xu1=0;
xu2=0;
tu1=1;
tu2=0;
dtu1=0;
dtu2=2;
d2tu1=0;
d2tu2=0;
xv1=0;
xv2=0;
tv1=1;
tv2=0;
dtv1=0;
dtv2=-2;
d2tv1=0;
d2tv2=0;

if(optimum==min)
{
    for(mint i=1;i<=k;i++)
    {

    RK4_func_LoC(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,max,
    SSize_u,SSize_v);
        u_s+=SSize_u;
        v_s+=SSize_v;
        xu1+=RK4_Init(tu1,dtu1,SSize_u);
        xu2+=RK4_Init(tu2,dtu2,SSize_u);
        tu1+=RK4_Init(dtu1,d2tu1,SSize_u);
        tu2+=RK4_Init(dtu2,d2tu2,SSize_u);
        dtu1=0;
        dtu2=2;
        d2tu1=0;
        d2tu2=0;
        xv1+=RK4_Init(tv1,dtv1,SSize_v);
        xv2+=RK4_Init(tv2,dtv2,SSize_v);
        tv1+=RK4_Init(dtv1,d2tv1,SSize_v);
        tv2+=RK4_Init(dtv2,d2tv2,SSize_v);
```

```
            dtv1=0;
            dtv2=-2;
            d2tv1=0;
            d2tv2=0;
        }
    }
    else if(optimum==max)
    {
        for(mint i=1;i<=k;i++)
        {


        RK4_func_LoC(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,min,
        SSize_u,SSize_v);
            u_s+=SSize_u;
            v_s+=SSize_v;
            xu1+=RK4_Init(tu1,dtu1,SSize_u);
            xu2+=RK4_Init(tu2,dtu2,SSize_u);
            tu1+=RK4_Init(dtu1,d2tu1,SSize_u);
            tu2+=RK4_Init(dtu2,d2tu2,SSize_u);
            dtu1=0;
            dtu2=2;
            d2tu1=0;
            d2tu2=0;
            xv1+=RK4_Init(tv1,dtv1,SSize_v);
            xv2+=RK4_Init(tv2,dtv2,SSize_v);
            tv1+=RK4_Init(dtv1,d2tv1,SSize_v);
            tv2+=RK4_Init(dtv2,d2tv2,SSize_v);
            dtv1=0;
            dtv2=-2;
            d2tv1=0;
            d2tv2=0;
        }
    }
    x[0]=xu1;
    y[0]=xv1;
    z[0]=xu2+xv2;
    LUDecomposition3x3(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2,dtv1,
        dtv2,d2tv1,d2tv2,optimum,coef);
    GeoCur[0]=coef[2];

    for(mint j=1;j<n;j++)
    {

        RK4_func_LoC(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum
        ,SSize_u,SSize_v);
        u_s+=SSize_u;
```

```
        v_s+=SSize_v;
        xu1+=RK4_Init(tu1,dtu1,SSize_u);
        xu2+=RK4_Init(tu2,dtu2,SSize_u);
        tu1+=RK4_Init(dtu1,d2tu1,SSize_u);
        tu2+=RK4_Init(dtu2,d2tu2,SSize_u);
        dtu1=0;
        dtu2=2;
        d2tu1=0;
        d2tu2=0;

        xv1+=RK4_Init(tv1,dtv1,SSize_v);
        xv2+=RK4_Init(tv2,dtv2,SSize_v);
        tv1+=RK4_Init(dtv1,d2tv1,SSize_v);
        tv2+=RK4_Init(dtv2,d2tv2,SSize_v);
        dtv1=0;
        dtv2=-2;
        d2tv1=0;
        d2tv2=0;

        x[j]=xu1;
        y[j]=xv1;
        z[j]=xu2+xv2;

        LUDecomposition3x3(xu1,xu2,tu1,tu2,dtu1,dtu2,d2tu1,d2tu2,xv1,xv2,tv1,tv2
        ,dtv1,dtv2,d2tv1,d2tv2,optimum,coef);
        GeoCur[j]=coef[2];

    }

    }

__device__ void ReturnXYZCoordinate( mint
    k, float x[], float y[], float z[], mint optimum)
{
float iniSSize_u,iniSSize_v,SSize_u,SSize_v;

float xu1, xu2, tu1, tu2, dtu1, dtu2, d2tu1, d2tu2;
float xv1, xv2, tv1, tv2, dtv1, dtv2, d2tv1, d2tv2;
float u_s,v_s;
float coef[3];

u_s=0;
v_s=0;
xu1=0;
xu2=0;
tu1=1;
tu2=0;
```

```
dtu1=0;
dtu2=2;
d2tu1=0;
d2tu2=0;
xv1=0;
xv2=0;
tv1=1;
tv2=0;
dtv1=0;
dtv2=-2;
d2tv1=0;
d2tv2=0;

if(optimum==min)
{
    for(mint i=1;i<=k;i++)
    {

    RK4_func_LoC(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,max,
    SSize_u,SSize_v);
        u_s+=SSize_u;
        v_s+=SSize_v;
        xu1+=RK4_Init(tu1,dtu1,SSize_u);
        xu2+=RK4_Init(tu2,dtu2,SSize_u);
        tu1+=RK4_Init(dtu1,d2tu1,SSize_u);
        tu2+=RK4_Init(dtu2,d2tu2,SSize_u);
        dtu1=0;
        dtu2=2;
        d2tu1=0;
        d2tu2=0;
        xv1+=RK4_Init(tv1,dtv1,SSize_v);
        xv2+=RK4_Init(tv2,dtv2,SSize_v);
        tv1+=RK4_Init(dtv1,d2tv1,SSize_v);
        tv2+=RK4_Init(dtv2,d2tv2,SSize_v);
        dtv1=0;
        dtv2=-2;
        d2tv1=0;
        d2tv2=0;
    }
}
else if(optimum==max)
{
    for(mint i=1;i<=k;i++)
    {


    RK4_func_LoC(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,min,
```

```
    SSize_u,SSize_v);
        u_s+=SSize_u;
        v_s+=SSize_v;
        xu1+=RK4_Init(tu1,dtu1,SSize_u);
        xu2+=RK4_Init(tu2,dtu2,SSize_u);
        tu1+=RK4_Init(dtu1,d2tu1,SSize_u);
        tu2+=RK4_Init(dtu2,d2tu2,SSize_u);
        dtu1=0;
        dtu2=2;
        d2tu1=0;
        d2tu2=0;
        xv1+=RK4_Init(tv1,dtv1,SSize_v);
        xv2+=RK4_Init(tv2,dtv2,SSize_v);
        tv1+=RK4_Init(dtv1,d2tv1,SSize_v);
        tv2+=RK4_Init(dtv2,d2tv2,SSize_v);
        dtv1=0;
        dtv2=-2;
        d2tv1=0;
        d2tv2=0;
    }
}
x[0]=xu1;
y[0]=xv1;
z[0]=xu2+xv2;

for(mint j=1;j<n;j++)
{

    RK4_func_LoC(xu1,xu2,tu1,tu2,dtu1,dtu2,xv1,xv2,tv1,tv2,dtv1,dtv2,optimum
    ,SSize_u,SSize_v);
    u_s+=SSize_u;
    v_s+=SSize_v;
    xu1+=RK4_Init(tu1,dtu1,SSize_u);
    xu2+=RK4_Init(tu2,dtu2,SSize_u);
    tu1+=RK4_Init(dtu1,d2tu1,SSize_u);
    tu2+=RK4_Init(dtu2,d2tu2,SSize_u);
    dtu1=0;
    dtu2=2;
    d2tu1=0;
    d2tu2=0;

    xv1+=RK4_Init(tv1,dtv1,SSize_v);
    xv2+=RK4_Init(tv2,dtv2,SSize_v);
    tv1+=RK4_Init(dtv1,d2tv1,SSize_v);
    tv2+=RK4_Init(dtv2,d2tv2,SSize_v);
    dtv1=0;
    dtv2=-2;
```

```
    d2tv1=0;
    d2tv2=0;

    x[j]=xu1;
    y[j]=xv1;
    z[j]=xu2+xv2;

}

}


__device__ void Init_Curve_2D_Boundaries( float GeoCur[],  float init_x,
     float init_y, float x[], float y[], float tangent1[], float
     tangent2[], float normal1[], float normal2[], mint optimum)
{//use to compute plane curve coordinates from given geodesic curvature
float DummyGeoCur1,DummyGeoCur2,DGCDs;
float x1, x2;
float t1, t2, dt1, dt2, d2t1, d2t2;
float n1, n2, dn1, dn2, d2n1, d2n2;

x1=init_x;
x2=init_y;

if (optimum==max)//when curve render in maximum principle direction, the
     tangent is {-1,0,0} at (-x)(-y) plane. We cross {-1,0,0}x{0,0,1}
     then we get normal is {0,1,0}. Then we flip the curve from (-x)(-y)
     plane to (-x)(y) plane we get tangent is {-1,0,0} and normal
     is {0,-1,0}. Finally we flip the curve from (-x)(y) plane to
     (x)(y) plane then we have tangent {1,0,0} and normal is {0,-1,0}.
{
    t1=1;
    t2=0;
    n1=0;
    n2=-1;
}
else if(optimum==min)//when curve render in minimum principle direction, the
     tangent is {0,-1,0} at (-x)(-y) plane. We cross {0,-1,0}x{0,0,1}
     then we get normal is {-1,0,0}. Then we flip the curve from
     (-x)(-y) plane to (-x)(y) plane we get tangent is {0,1,0} and
     normal is {-1,0,0}. Finally we flip the curve from (-x)(y) plane
     to (x)(y) plane then we have tangent {0,1,0} and normal is {1,0,0}.
{
    t1=0;
    t2=1;
    n1=1;
    n2=0;
```

```
}

x[0]=x1;
y[0]=x2;
tangent1[0]=t1;
tangent2[0]=t2;
normal1[0]=n1;
normal2[0]=n2;

for(mint j=1;j<n;j++)
{

DummyGeoCur1=GeoCur[j-1];
DummyGeoCur2=GeoCur[j];
DGCDs=(DummyGeoCur2-DummyGeoCur1)/h;
dt1=DummyGeoCur1*n1;
dt2=DummyGeoCur1*n2;
dn1=-DummyGeoCur1*t1;
dn2=-DummyGeoCur1*t2;
d2t1=DummyGeoCur1*dn1+DGCDs*n1;
d2t2=DummyGeoCur1*dn2+DGCDs*n2;
d2n1=-DummyGeoCur1*dt1-DGCDs*t1;
d2n2=-DummyGeoCur1*dt2-DGCDs*t2;

x1=x1+RK4_Init(t1,dt1,h);
x2=x2+RK4_Init(t2,dt2,h);
t1=t1+RK4_Init(dt1,d2t1,h);
t2=t2+RK4_Init(dt2,d2t2,h);
n1=n1+RK4_Init(dn1,d2n1,h);
n2=n2+RK4_Init(dn2,d2n2,h);

x[j]=x1;
y[j]=x2;
tangent1[j]=t1;
tangent2[j]=t2;
normal1[j]=n1;
normal2[j]=n2;
}

}


__device__ void CutCurve_2D( float GeoCur[], mint max_n, float init_x,
    float init_y, float init_t1, float init_t2, float init_n1,
    float init_n2, float x[], float y[], float tangent1[], float
    tangent2[], float normal1[], float normal2[], mint optimum)
{//use to compute plane curve coordinates from given geodesic curvature
```

```
float DummyGeoCur1,DummyGeoCur2,DGCDs;
float x1, x2;
float t1, t2, dt1, dt2, d2t1, d2t2;
float n1, n2, dn1, dn2, d2n1, d2n2;

x1=init_x;
x2=init_y;

if (optimum==max)//when curve render in maximum principle direction, the
    tangent is {-1,0,0} at (-x)(-y) plane. We cross {-1,0,0}x{0,0,1}
    then we get normal is {0,1,0}. Then we flip the curve from (-x)(-y)
    plane to (-x)(y) plane we get tangent is {-1,0,0} and normal is
    {0,-1,0}. Finally we flip the curve from (-x)(y) plane to (x)(y)
    plane then we have tangent {1,0,0} and normal is {0,-1,0}. Then
    we reverse the normal={0,1,0} so that we can cut out the pieces
{
    t1=-init_n1;
    t2=-init_n2;
    n1=init_t1;
    n2=init_t2;
}
else if(optimum==min)//when curve render in minimum principle direction, the
    tangent is {0,-1,0} at (-x)(-y) plane. We cross {0,-1,0}x{0,0,1}
    then we get normal is {-1,0,0}. Then we flip the curve from (-x)(-y)
    plane to (-x)(y) plane we get tangent is {0,1,0} and normal is
    {-1,0,0}. Finally we flip the curve from (-x)(y) plane to (x)(y)
    plane then we have tangent {0,1,0} and normal is {1,0,0}. Then
    we reverse the normal={-1,0,0} so that we can cut out the pieces
{
    t1=init_n1;
    t2=init_n2;
    n1=-init_t1;
    n2=-init_t2;
}

x[max_n]=x1;
y[max_n]=x2;
tangent1[max_n]=t1;
tangent2[max_n]=t2;
normal1[max_n]=n1;
normal2[max_n]=n2;

for(mint j=max_n-1;j>=0;j--)
{

DummyGeoCur1=GeoCur[j+1];
DummyGeoCur2=GeoCur[j];
```

```
DGCDs=(DummyGeoCur2-DummyGeoCur1)/h;
dt1=DummyGeoCur1*n1;
dt2=DummyGeoCur1*n2;
dn1=-DummyGeoCur1*t1;
dn2=-DummyGeoCur1*t2;
d2t1=DummyGeoCur1*dn1+DGCDs*n1;
d2t2=DummyGeoCur1*dn2+DGCDs*n2;
d2n1=-DummyGeoCur1*dt1-DGCDs*t1;
d2n2=-DummyGeoCur1*dt2-DGCDs*t2;

x1=x1+RK4_Init(t1,dt1,h);
x2=x2+RK4_Init(t2,dt2,h);
t1=t1+RK4_Init(dt1,d2t1,h);
t2=t2+RK4_Init(dt2,d2t2,h);
n1=n1+RK4_Init(dn1,d2n1,h);
n2=n2+RK4_Init(dn2,d2n2,h);


x[j]=x1;
y[j]=x2;
tangent1[j]=t1;
tangent2[j]=t2;
normal1[j]=n1;
normal2[j]=n2;
}

}

__device__ void ReturnInteger( mint& min_i, float x1[],
    float y1[], float z1[], float x2[], float y2[], float z2[])
{
float dif1;
float min_dif1=5;

for(mint i=0; i<n; i++)
{
    for(mint j=0; j<n; j++)
    {
        dif1=pow(x1[i]-x2[j],2)+pow(y1[i]-y2[j],2)+pow(z1[i]-z2[j],2);

        if (dif1<min_dif1)
        {
            min_dif1=dif1;
            min_i=i;
        }
    }
}
```

```
}


__device__ void ReturnPointsPosition( mint& res_i, mint& res_min_j, mint&
    res_max_j, float x1[], float y1[], float z1[], float x2[],
    float y2[], float z2[], float x3[], float y3[], float z3[])
{
mint min_i;
mint min_j, max_j;
float dif1, dif2;
float min_dif1=5;
float min_dif2=5;

for(mint i=0; i<n; i++)
{
    for(mint j=0; j<n; j++)
    {
        dif1=pow(x1[i]-x3[j],2)+pow(y1[i]-y3[j],2)+pow(z1[i]-z3[j],2);
        dif2=pow(x2[i]-x3[j],2)+pow(y2[i]-y3[j],2)+pow(z2[i]-z3[j],2);

        if (dif1<min_dif1)
        {
            min_dif1=dif1;
            min_j=j;
        }

        if (dif2<min_dif2)
        {
            min_dif2=dif2;
            min_i=i;
            max_j=j;
        }
    }
}

res_i=min_i;
res_min_j=min_j;
res_max_j=max_j;

}

__device__ void ObtainSmallerPiece( mint res_i, mint min_j, mint max_j, float
    GeoCur[], float x_2d[], float y_2d[], float t1_2d[], float t2_2d[],
    float n1_2d[], float n2_2d[], float& resx, float& resy, mint optimum)
{
float DummyGeoCur1,DummyGeoCur2,DGCDs;
float x1, x2;
```

```
float t1, t2, dt1, dt2, d2t1, d2t2;
float n1, n2, dn1, dn2, d2n1, d2n2;


x1=x_2d[res_i];
x2=y_2d[res_i];

if (optimum==max)//Since the 1st curve is in minimum principle
    direction (from curve to cut) and its tangent is tangent1={0,1}
    and normal is normal1={-1,0}. The orthogonal curve is
    rendered in maximum principle direction , hence tangent is
    tangento={1,0}=-normal1 and normal is normalo={0,-1}=-tangent1.
{
    t1=n1_2d[res_i];
    t2=n2_2d[res_i];
    n1=t1_2d[res_i];
    n2=t2_2d[res_i];

}
else if(optimum==min)//Since the 1st curve is in maximum
    principle direction (from curve to cut) and its tangent is
    tangent1={1,0} and normal is normal1={0,1}. The orthogonal curve
    is rendered in minimum principle direction , hence tangent is
    tangento={0,1}=normal1 and normal is normalo={1,0}=tangent1.
{
    t1=-n1_2d[res_i];
    t2=-n2_2d[res_i];
    n1=-t1_2d[res_i];
    n2=-t2_2d[res_i];
}

for(mint j=max_j-1;j>=min_j;j--)
{

DummyGeoCur1=GeoCur[j+1];
DummyGeoCur2=GeoCur[j];
DGCDs=(DummyGeoCur2-DummyGeoCur1)/h;
dt1=DummyGeoCur1*n1;
dt2=DummyGeoCur1*n2;
dn1=-DummyGeoCur1*t1;
dn2=-DummyGeoCur1*t2;
d2t1=DummyGeoCur1*dn1+DGCDs*n1;
d2t2=DummyGeoCur1*dn2+DGCDs*n2;
d2n1=-DummyGeoCur1*dt1-DGCDs*t1;
d2n2=-DummyGeoCur1*dt2-DGCDs*t2;

x1=x1+RK4_Init(t1,dt1,h);
```

```
x2=x2+RK4_Init(t2,dt2,h);
t1=t1+RK4_Init(dt1,d2t1,h);
t2=t2+RK4_Init(dt2,d2t2,h);
n1=n1+RK4_Init(dn1,d2n1,h);
n2=n2+RK4_Init(dn2,d2n2,h);
}

resx=x1;
resy=x2;

}

__device__ void SmallerPieceBoundary( float resx[], float resy[], float
    GeoCur[], mint min_j, mint max_j, float x_2d, float y_2d, float
    t1_2d, float t2_2d, float n1_2d, float n2_2d, mint optimum)
{
float DummyGeoCur1,DummyGeoCur2,DGCDs;
float x1, x2;
float t1, t2, dt1, dt2, d2t1, d2t2;
float n1, n2, dn1, dn2, d2n1, d2n2;


x1=x_2d;
x2=y_2d;

if (optimum==max)//Since the 1st curve is in minimum principle
    direction (from curve to cut) and its tangent is tangent1={0,1}
    and normal is normal1={-1,0}. The orthogonal curve is
    rendered in maximum principle direction , hence tangent is
    tangento={1,0}=-normal1 and normal is normalo={0,-1}=-tangent1.
{
    t1=n1_2d;
    t2=n2_2d;
    n1=t1_2d;
    n2=t2_2d;

}
else if(optimum==min)//Since the 1st curve is in maximum
    principle direction (from curve to cut) and its tangent is
    tangent1={1,0} and normal is normal1={0,1}. The orthogonal curve
    is rendered in minimum principle direction , hence tangent is
    tangento={0,1}=normal1 and normal is normalo={1,0}=tangent1.
{
    t1=-n1_2d;
    t2=-n2_2d;
    n1=-t1_2d;
    n2=-t2_2d;
```

```
}

resx[0]=x1;
resy[0]=x2;


for(mint i=1, mint j=max_j-1;j>=min_j;j--,i++)
{
DummyGeoCur1=GeoCur[j+1];
DummyGeoCur2=GeoCur[j];
DGCDs=(DummyGeoCur2-DummyGeoCur1)/h;
dt1=DummyGeoCur1*n1;
dt2=DummyGeoCur1*n2;
dn1=-DummyGeoCur1*t1;
dn2=-DummyGeoCur1*t2;
d2t1=DummyGeoCur1*dn1+DGCDs*n1;
d2t2=DummyGeoCur1*dn2+DGCDs*n2;
d2n1=-DummyGeoCur1*dt1-DGCDs*t1;
d2n2=-DummyGeoCur1*dt2-DGCDs*t2;

x1=x1+RK4_Init(t1,dt1,h);
x2=x2+RK4_Init(t2,dt2,h);
t1=t1+RK4_Init(dt1,d2t1,h);
t2=t2+RK4_Init(dt2,d2t2,h);
n1=n1+RK4_Init(dn1,d2n1,h);
n2=n2+RK4_Init(dn2,d2n2,h);
resx[i]=x1;
resy[i]=x2;

}
}

__global__ void ReturnSmallPieceBoundary( float *
    xvalue, float * yvalue, float GeoCur[], mint min_j, mint
    max_j, float x_2d, float y_2d, float t1_2d, float t2_2d,
    float n1_2d, float n2_2d, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float res_x[181], res_y[181];

SmallerPieceBoundary(res_x,res_y,GeoCur,min_j,max_j,x_2d,y_2d,t1_2d,t2_2d,n1_2d
    ,n2_2d,optimum);
if( index < ListSize)
{
    xvalue[index]= res_x[index];
    yvalue[index]= res_y[index];
}
```

```
}

__global__ void ReturnEndPoints( float * xvalue, float * yvalue, mint
    res_i[], mint res_min_j[], mint res_max_j[], float x_2d[],
    float y_2d[], float t1_2d[], float t2_2d[], float n1_2d[],
    float n2_2d[], mint SPoint, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float res_x, res_y;
mint min_i, min_j, max_j;
float x3[n], y3[n], z3[n], GeoCur3[n];

if( index < ListSize)
{
    ReturnDetail(index+SPoint,x3,y3,z3,GeoCur3,optimum);
    min_i=res_i[index];
    min_j=res_min_j[index];
    max_j=res_max_j[index];

    ObtainSmallerPiece(min_i,min_j,max_j,GeoCur3,x_2d,y_2d,t1_2d,t2_2d,n1_2d
    ,n2_2d,res_x,res_y,optimum);
    xvalue[index]= res_x;
    yvalue[index]= res_y;
}

}

__global__ void ReturnIntersectionPoint( mint * xvalue, float x1[], float
    y1[], float z1[], float x2[], float y2[], float z2[], mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
mint mint_i1;

ReturnInteger( mint_i1, x1, y1, z1, x2, y2, z2);//compute intersection point

if( index < ListSize)
{
    xvalue[index]= mint_i1;
}
}


__global__ void ReturnPositionofPoints( mint * ivalue, mint * min_jvalue, mint
    * max_jvalue, float x1[], float y1[], float z1[], float x2[],
    float y2[], float z2[], mint SPoint, mint optimum, mint ListSize)
{
```

```
mint index = threadIdx.x + blockIdx.x * blockDim.x;
mint res_i, res_min_j, res_max_j;
float x3[n], y3[n], z3[n];

if( index < ListSize)
{
    ReturnXYZCoordinate(index+SPoint,x3,y3,z3,optimum);

     ReturnPointsPosition(res_i,res_min_j,res_max_j,x1,y1,z1,x2,y2,z2,x3,y3,z3);
    ivalue[index]= res_i;
    min_jvalue[index]= res_min_j;
    max_jvalue[index]= res_max_j;
}

}

__global__ void BoundaryPositionofPoints( mint * min_jvalue, mint *
     max_jvalue, float x1[], float y1[], float z1[], float x2[], float
     y2[], float z2[], float x3[], float y3[], float z3[], mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
mint res_i, res_min_j, res_max_j;

ReturnPointsPosition(res_i,res_min_j,res_max_j,x1,y1,z1,x2,y2,z2,x3,y3,z3);

if( index < ListSize)
{
    min_jvalue[index]= res_min_j;
    max_jvalue[index]= res_max_j;
}

}

__global__ void Boundariesin2D( float * xvalue, float * yvalue, float *
     t1value, float * t2value, float * n1value, float * n2value, float
     GeoCur[], float init_x, float init_y, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float x_2d[n], y_2d[n], t1_2d[n], t2_2d[n], n1_2d[n], n2_2d[n];

Init_Curve_2D_Boundaries( GeoCur, init_x,
     init_y, x_2d, y_2d, t1_2d, t2_2d, n1_2d, n2_2d, optimum);

if( index < ListSize)
{
    xvalue[index]= x_2d[index];
    yvalue[index]= y_2d[index];
```

```
    t1value[index]= t1_2d[index];
    t2value[index]= t2_2d[index];
    n1value[index]= n1_2d[index];
    n2value[index]= n2_2d[index];
}
}

__global__ void CutCurvein2D( float * xvalue, float * yvalue, float
    * t1value, float * t2value, float * n1value, float * n2value,
    float GeoCur[], float init_x, float init_y, float init_t1, float
    init_t2, float init_n1, float init_n2, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float x_2d[n], y_2d[n], t1_2d[n], t2_2d[n], n1_2d[n], n2_2d[n];

CutCurve_2D( GeoCur, ListSize-1, init_x, init_y, init_t1, init_t2, init_n1,
    init_n2, x_2d, y_2d, t1_2d, t2_2d, n1_2d, n2_2d, optimum);

if( index < ListSize)
{
    xvalue[index]= x_2d[index];
    yvalue[index]= y_2d[index];
    t1value[index]= t1_2d[index];
    t2value[index]= t2_2d[index];
    n1value[index]= n1_2d[index];
    n2value[index]= n2_2d[index];
}

}

__global__ void ReturnAllDetails( float* xvalue, float* yvalue, float*
    zvalue, float* GCvalue, mint init_s, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float x[n], y[n], z[n], GeoCur[n];

ReturnDetail(init_s,x,y,z,GeoCur,optimum);

if( index < ListSize)
{
    xvalue[index]= x[index];
    yvalue[index]= y[index];
    zvalue[index]= z[index];
    GCvalue[index]= GeoCur[index];

}
```

```
}


";
Needs["CUDALink`"]
HPDetails = CUDAFunctionLoad[kernelcodeHP, "ReturnAllDetails",
   {{"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, _Integer, _Integer, _Integer}, 160];
HPBoundary2D = CUDAFunctionLoad[kernelcodeHP, "Boundariesin2D",
   {{"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Input"}, "Float", "Float", _Integer, _Integer}, 160];
HPCutCurve2D = CUDAFunctionLoad[kernelcodeHP, "CutCurvein2D",
   {{"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Input"}, "Float", "Float",
    "Float", "Float", "Float", "Float", _Integer, _Integer}, 160];
HPReturnIntersection = CUDAFunctionLoad[kernelcodeHP,
   "ReturnIntersectionPoint", {{_Integer, _, "Output"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, _Integer}, 160];
HPPointsPosition3D = CUDAFunctionLoad[kernelcodeHP, "ReturnPositionofPoints",
   {{_Integer, _, "Output"}, {_Integer, _, "Output"},
    {_Integer, _, "Output"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, _Integer, _Integer, _Integer}, 160];
HPSmallPieceBoundary2D = CUDAFunctionLoad[kernelcodeHP,
   "ReturnSmallPieceBoundary", {{"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Input"}, _Integer, _Integer, "Float", "Float",
    "Float", "Float", "Float", "Float", _Integer, _Integer}, 160];
HPBoundaryPosition3D = CUDAFunctionLoad[kernelcodeHP,
   "BoundaryPositionofPoints",
   {{_Integer, _, "Output"}, {_Integer, _, "Output"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, _Integer}, 160];
```

```
AbsoluteTiming[HPEndPoints2D =
  CUDAFunctionLoad[kernelcodeHP, "ReturnEndPoints", {{"Float", _, "Output"},
    {"Float", _, "Output"}, {_Integer, _, "Input"}, {_Integer, _, "Input"},
    {_Integer, _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, _Integer, _Integer, _Integer}, 160]]
```

```
{131.829,
 CUDAFunction[<>, ReturnEndPoints, {{Float, _, Output}, {Float, _, Output},
   {_Integer, _, Input}, {_Integer, _, Input}, {_Integer, _, Input},
   {Float, _, Input}, {Float, _, Input}, {Float, _, Input}, {Float, _, Input},
   {Float, _, Input}, {Float, _, Input}, Integer64, Integer64, Integer64}]}
```

```
HPDetail[v1_, n1_, optimum_] :=
  Module[{v11 = v1, nn = n1, opt1 = optimum, Tx, Ty, Tz, TGC},
   Tx = CUDAMemoryAllocate["Float", nn];
   Ty = CUDAMemoryAllocate["Float", nn];
   Tz = CUDAMemoryAllocate["Float", nn];
   TGC = CUDAMemoryAllocate["Float", nn];
   HPDetails[Tx, Ty, Tz, TGC, v11, opt1, nn];
   Transpose[{CUDAMemoryGet[Tx],
     CUDAMemoryGet[Ty], CUDAMemoryGet[Tz], CUDAMemoryGet[TGC]}]];
```

```
HPPointsPosition[v1_, v2_, {SPoint_, n3_}, optimum_] :=
  Module[{opt1, opt2 = optimum, n1 = 1201, n2 = 1201, Tx1, Ty1,
    Tz1, Tx2, Ty2, Tz2, TGC1, TGC2, resi, resminj, resmaxj},
   opt1 = Piecewise[{{0, optimum == 1}}, 1];
   Tx1 = CUDAMemoryAllocate["Float", n1];
   Ty1 = CUDAMemoryAllocate["Float", n1];
   Tz1 = CUDAMemoryAllocate["Float", n1];
   TGC1 = CUDAMemoryAllocate["Float", n1];
   Tx2 = CUDAMemoryAllocate["Float", n2];
   Ty2 = CUDAMemoryAllocate["Float", n2];
   Tz2 = CUDAMemoryAllocate["Float", n2];
   TGC2 = CUDAMemoryAllocate["Float", n2];
   resi = CUDAMemoryAllocate[_Integer, n3];
   resminj = CUDAMemoryAllocate[_Integer, n3];
   resmaxj = CUDAMemoryAllocate[_Integer, n3];

   HPDetails[Tx1, Ty1, Tz1, TGC1, v1, opt1, n1];
   HPDetails[Tx2, Ty2, Tz2, TGC2, v2, opt1, n2];
   HPPointsPosition3D[resi, resminj,
    resmaxj, Tx1, Ty1, Tz1, Tx2, Ty2, Tz2, SPoint, opt2, n3];
   Transpose[{CUDAMemoryGet[resi], CUDAMemoryGet[resminj],
     CUDAMemoryGet[resmaxj]}]];
```

```
HPIntersection[v1_, v2_, optimum_] := Module[{opt1, opt2 = optimum,
    n1 = 1201, n2 = 1201, Tx1, Ty1, Tz1, Tx2, Ty2, Tz2, TGC1, TGC2, resi},
   opt1 = Piecewise[{{0, optimum == 1}}, 1];
   Tx1 = CUDAMemoryAllocate["Float", n1];
   Ty1 = CUDAMemoryAllocate["Float", n1];
   Tz1 = CUDAMemoryAllocate["Float", n1];
   TGC1 = CUDAMemoryAllocate["Float", n1];
   Tx2 = CUDAMemoryAllocate["Float", n2];
   Ty2 = CUDAMemoryAllocate["Float", n2];
   Tz2 = CUDAMemoryAllocate["Float", n2];
   TGC2 = CUDAMemoryAllocate["Float", n2];
   resi = CUDAMemoryAllocate[_Integer, 1];

   HPDetails[Tx1, Ty1, Tz1, TGC1, v1, opt1, n1];
   HPDetails[Tx2, Ty2, Tz2, TGC2, v2, opt2, n2];
   HPReturnIntersection[resi, Tx1, Ty1, Tz1, Tx2, Ty2, Tz2, 1];
   Transpose[{CUDAMemoryGet[resi]}]];

HPInitialBoundary2D[{Initx_, Inity_}, {n1_, optimum_}] :=
  Module[{v1 = 900, opt1 = optimum, InitTx, InitTy, InitTz, InitTGC,
    InitTx2D, InitTy2D, InitTt12D, InitTt22D, InitTn12D, InitTn22D},
   InitTx = CUDAMemoryAllocate["Float", n1];
   InitTy = CUDAMemoryAllocate["Float", n1];
   InitTz = CUDAMemoryAllocate["Float", n1];
   InitTGC = CUDAMemoryAllocate["Float", n1];
   InitTx2D = CUDAMemoryAllocate["Float", n1];
   InitTy2D = CUDAMemoryAllocate["Float", n1];
   InitTt12D = CUDAMemoryAllocate["Float", n1];
   InitTt22D = CUDAMemoryAllocate["Float", n1];
   InitTn12D = CUDAMemoryAllocate["Float", n1];
   InitTn22D = CUDAMemoryAllocate["Float", n1];
   HPDetails[InitTx, InitTy, InitTz, InitTGC, v1, opt1, n1];
   HPBoundary2D[InitTx2D, InitTy2D, InitTt12D,
    InitTt22D, InitTn12D, InitTn22D, InitTGC, Initx, Inity, opt1, n1];
   Transpose[{CUDAMemoryGet[InitTx2D], CUDAMemoryGet[InitTy2D]}]];
```

```
HPCutCurvein2D[{Initx_, Inity_}, {n1_}, {v2_, n2_, optimum_}] :=
  Module[{nn = 1201, v1 = 900, opt1, opt2 = optimum, InitTx2D,
    InitTy2D, InitTt12D, InitTt22D, InitTn12D, InitTn22D, TBx, TBy,
    TBz, TBGC, Tx1, Ty1, Tz1, TGC1, resi, ivalue, Tx2D, Ty2D, Tt12D,
    Tt22D, Tn12D, Tn22D, Inix, Iniy, Init1, Init2, Inin1, Inin2},
   opt1 = Piecewise[{{0, optimum == 1}}, 1];
   InitTx2D = CUDAMemoryAllocate["Float", nn];
   InitTy2D = CUDAMemoryAllocate["Float", nn];
   InitTt12D = CUDAMemoryAllocate["Float", nn];
   InitTt22D = CUDAMemoryAllocate["Float", nn];
   InitTn12D = CUDAMemoryAllocate["Float", nn];
   InitTn22D = CUDAMemoryAllocate["Float", nn];
   TBx = CUDAMemoryAllocate["Float", nn];
   TBy = CUDAMemoryAllocate["Float", nn];
   TBz = CUDAMemoryAllocate["Float", nn];
   TBGC = CUDAMemoryAllocate["Float", nn];
   Tx1 = CUDAMemoryAllocate["Float", nn];
   Ty1 = CUDAMemoryAllocate["Float", nn];
   Tz1 = CUDAMemoryAllocate["Float", nn];
   TGC1 = CUDAMemoryAllocate["Float", nn];
   resi = CUDAMemoryAllocate[_Integer, 1];

   Tx2D = CUDAMemoryAllocate["Float", nn];
   Ty2D = CUDAMemoryAllocate["Float", nn];
   Tt12D = CUDAMemoryAllocate["Float", nn];
   Tt22D = CUDAMemoryAllocate["Float", nn];
   Tn12D = CUDAMemoryAllocate["Float", nn];
   Tn22D = CUDAMemoryAllocate["Float", nn];

   HPDetails[TBx, TBy, TBz, TBGC, v1, opt1, nn];
   HPDetails[Tx1, Ty1, Tz1, TGC1, v2, opt2, nn];
   (*Detail of upper curve*)
   HPReturnIntersection[resi, TBx, TBy, TBz, Tx1, Ty1, Tz1, 1];
   HPBoundary2D[InitTx2D, InitTy2D, InitTt12D,
    InitTt22D, InitTn12D, InitTn22D, TBGC, Initx, Inity, opt1, n1];
   ivalue = First[CUDAMemoryGet[resi]] + 1;
   Inix = CUDAMemoryGet[InitTx2D][[ivalue]];
   Iniy = CUDAMemoryGet[InitTy2D][[ivalue]];
   Init1 = CUDAMemoryGet[InitTt12D][[ivalue]];
   Init2 = CUDAMemoryGet[InitTt22D][[ivalue]];
   Inin1 = CUDAMemoryGet[InitTn12D][[ivalue]];
   Inin2 = CUDAMemoryGet[InitTn22D][[ivalue]];
   HPCutCurve2D[Tx2D, Ty2D, Tt12D, Tt22D, Tn12D,
    Tn22D, TGC1, Inix, Iniy, Init1, Init2, Inin1, Inin2, opt2, n2];
   Transpose[{CUDAMemoryGet[Tx2D][[1 ;; n2]], CUDAMemoryGet[Ty2D][[1 ;; n2]]}]];

HPEndPointin2D[{Initx_, Inity_},
   {v1_, n1_}, {v2_, n2_}, {SPoint_, n3_}, optimum_] :=
```

```
Module[{n0 = 1193, nn = 1201, v0 = 900, opt1, opt2 = optimum, InitTx2D,
   InitTy2D, InitTt12D, InitTt22D, InitTn12D, InitTn22D, TBx,
   TBy, TBz, TBGC, Tx1, Ty1, Tz1, TGC1, Tx2, Ty2, Tz2, TGC2, BPosi,
   ivalue, Tx2D, Ty2D, Tt12D, Tt22D, Tn12D, Tn22D, Inix, Iniy, Init1,
   Init2, Inin1, Inin2, resi, resminj, resmaxj, Endx2D, Endy2D},
  opt1 = Piecewise[{{0, optimum == 1}}, 1];
  InitTx2D = CUDAMemoryAllocate["Float", nn];
  InitTy2D = CUDAMemoryAllocate["Float", nn];
  InitTt12D = CUDAMemoryAllocate["Float", nn];
  InitTt22D = CUDAMemoryAllocate["Float", nn];
  InitTn12D = CUDAMemoryAllocate["Float", nn];
  InitTn22D = CUDAMemoryAllocate["Float", nn];
  TBx = CUDAMemoryAllocate["Float", nn];
  TBy = CUDAMemoryAllocate["Float", nn];
  TBz = CUDAMemoryAllocate["Float", nn];
  TBGC = CUDAMemoryAllocate["Float", nn];
  Tx1 = CUDAMemoryAllocate["Float", nn];
  Ty1 = CUDAMemoryAllocate["Float", nn];
  Tz1 = CUDAMemoryAllocate["Float", nn];
  TGC1 = CUDAMemoryAllocate["Float", nn];
  Tx2 = CUDAMemoryAllocate["Float", nn];
  Ty2 = CUDAMemoryAllocate["Float", nn];
  Tz2 = CUDAMemoryAllocate["Float", nn];
  TGC2 = CUDAMemoryAllocate["Float", nn];

  BPosi = CUDAMemoryAllocate[_Integer, 1];

  Tx2D = CUDAMemoryAllocate["Float", nn];
  Ty2D = CUDAMemoryAllocate["Float", nn];
  Tt12D = CUDAMemoryAllocate["Float", nn];
  Tt22D = CUDAMemoryAllocate["Float", nn];
  Tn12D = CUDAMemoryAllocate["Float", nn];
  Tn22D = CUDAMemoryAllocate["Float", nn];


  resi = CUDAMemoryAllocate[_Integer, n3];
  resminj = CUDAMemoryAllocate[_Integer, n3];
  resmaxj = CUDAMemoryAllocate[_Integer, n3];
  Endx2D = CUDAMemoryAllocate["Float", n3];
  Endy2D = CUDAMemoryAllocate["Float", n3];


  HPDetails[TBx, TBy, TBz, TBGC, v0, opt1, nn];
  (*Detail of boundary curve*)
  HPDetails[Tx2, Ty2, Tz2, TGC2, v2, opt2, nn];
  (*Detail of upper curve*)
  HPReturnIntersection[BPosi, TBx, TBy, TBz, Tx2, Ty2, Tz2, 1];
```

```
    (*compute intersection point of boundary curve and upper curve*)
    HPBoundary2D[InitTx2D, InitTy2D, InitTt12D,
     InitTt22D, InitTn12D, InitTn22D, TBGC, Initx, Inity, opt1, n0];
    (*Draw of 2D boundary curve*)
    ivalue = First[CUDAMemoryGet[BPosi]] + 1;
    Inix = CUDAMemoryGet[InitTx2D][[ivalue]];
    Iniy = CUDAMemoryGet[InitTy2D][[ivalue]];
    Init1 = CUDAMemoryGet[InitTt12D][[ivalue]];
    Init2 = CUDAMemoryGet[InitTt22D][[ivalue]];
    Inin1 = CUDAMemoryGet[InitTn12D][[ivalue]];
    Inin2 = CUDAMemoryGet[InitTn22D][[ivalue]];
    HPCutCurve2D[Tx2D, Ty2D, Tt12D, Tt22D, Tn12D,
     Tn22D, TGC2, Inix, Iniy, Init1, Init2, Inin1, Inin2, opt2, n2];
    (*Draw of 2D upper curve*)
    HPDetails[Tx1, Ty1, Tz1, TGC1, v1, opt2, nn];
    (*Detail of lower curve*)

    HPPointsPosition3D[resi, resminj,
     resmaxj, Tx1, Ty1, Tz1, Tx2, Ty2, Tz2, SPoint, opt1, n3];
    HPEndPoints2D[Endx2D, Endy2D, resi, resminj, resmaxj, Tx2D,
     Ty2D, Tt12D, Tt22D, Tn12D, Tn22D, SPoint, opt1, n3];
    Transpose[{CUDAMemoryGet[Endx2D], CUDAMemoryGet[Endy2D]}]];

HPLastBoundary2D[{Initx_, Inity_}, {v1_, n1_}, {v2_, n2_}, {n3_}, optimum_] :=
  Module[{n0 = 1193, nn = 1201, v0 = 900, opt1, opt2 = optimum, InitTx2D, InitTy2D,
     InitTt12D, InitTt22D, InitTn12D, InitTn22D, TBx, TBy, TBz, TBGC, Tx1, Ty1,
     Tz1, TGC1, Tx2, Ty2, Tz2, TGC2, Tx3, Ty3, Tz3, TGC3, BPosi, ivalue, Tx2D, Ty2D,
     Tt12D, Tt22D, Tn12D, Tn22D, Inix, Iniy, Init1, Init2, Inin1, Inin2, resminj,
     resmaxj, minj, maxj, Dx2D, Dy2D, Dt12D, Dt22D, Dn12D, Dn22D, Endx2D, Endy2D},
    opt1 = Piecewise[{{0, optimum == 1}}, 1];
    InitTx2D = CUDAMemoryAllocate["Float", nn];
    InitTy2D = CUDAMemoryAllocate["Float", nn];
    InitTt12D = CUDAMemoryAllocate["Float", nn];
    InitTt22D = CUDAMemoryAllocate["Float", nn];
    InitTn12D = CUDAMemoryAllocate["Float", nn];
    InitTn22D = CUDAMemoryAllocate["Float", nn];
    TBx = CUDAMemoryAllocate["Float", nn];
    TBy = CUDAMemoryAllocate["Float", nn];
    TBz = CUDAMemoryAllocate["Float", nn];
    TBGC = CUDAMemoryAllocate["Float", nn];
    Tx1 = CUDAMemoryAllocate["Float", nn];
    Ty1 = CUDAMemoryAllocate["Float", nn];
    Tz1 = CUDAMemoryAllocate["Float", nn];
    TGC1 = CUDAMemoryAllocate["Float", nn];
    Tx2 = CUDAMemoryAllocate["Float", nn];
    Ty2 = CUDAMemoryAllocate["Float", nn];
    Tz2 = CUDAMemoryAllocate["Float", nn];
    TGC2 = CUDAMemoryAllocate["Float", nn];
```

```
Tx3 = CUDAMemoryAllocate["Float", nn];
Ty3 = CUDAMemoryAllocate["Float", nn];
Tz3 = CUDAMemoryAllocate["Float", nn];
TGC3 = CUDAMemoryAllocate["Float", nn];

BPosi = CUDAMemoryAllocate[_Integer, 1];

Tx2D = CUDAMemoryAllocate["Float", nn];
Ty2D = CUDAMemoryAllocate["Float", nn];
Tt12D = CUDAMemoryAllocate["Float", nn];
Tt22D = CUDAMemoryAllocate["Float", nn];
Tn12D = CUDAMemoryAllocate["Float", nn];
Tn22D = CUDAMemoryAllocate["Float", nn];

resminj = CUDAMemoryAllocate[_Integer, 1];
resmaxj = CUDAMemoryAllocate[_Integer, 1];
Endx2D = CUDAMemoryAllocate["Float", n3];
Endy2D = CUDAMemoryAllocate["Float", n3];

HPDetails[TBx, TBy, TBz, TBGC, v0, opt1, nn];
(*Detail of boundary curve*)
HPDetails[Tx2, Ty2, Tz2, TGC2, v2, opt2, nn];
(*Detail of upper curve*)
HPReturnIntersection[BPosi, TBx, TBy, TBz, Tx2, Ty2, Tz2, 1];
(*compute intersection point of boundary curve and upper curve*)
HPBoundary2D[InitTx2D, InitTy2D, InitTt12D,
 InitTt22D, InitTn12D, InitTn22D, TBGC, Initx, Inity, opt1, n0];
(*Draw of 2D boundary curve*)
ivalue = First[CUDAMemoryGet[BPosi]] + 1;
Inix = CUDAMemoryGet[InitTx2D][[ivalue]];
Iniy = CUDAMemoryGet[InitTy2D][[ivalue]];
Init1 = CUDAMemoryGet[InitTt12D][[ivalue]];
Init2 = CUDAMemoryGet[InitTt22D][[ivalue]];
Inin1 = CUDAMemoryGet[InitTn12D][[ivalue]];
Inin2 = CUDAMemoryGet[InitTn22D][[ivalue]];
HPCutCurve2D[Tx2D, Ty2D, Tt12D, Tt22D, Tn12D,
 Tn22D, TGC2, Inix, Iniy, Init1, Init2, Inin1, Inin2, opt2, n2];
(*Draw of 2D upper curve*)
HPDetails[Tx1, Ty1, Tz1, TGC1, v1, opt2, nn];
(*Detail of lower curve*)
HPDetails[Tx3, Ty3, Tz3, TGC3, 0, opt1, nn];
HPBoundaryPosition3D[resminj,
 resmaxj, Tx1, Ty1, Tz1, Tx2, Ty2, Tz2, Tx3, Ty3, Tz3, 1];
minj = First[CUDAMemoryGet[resminj]];
maxj = First[CUDAMemoryGet[resmaxj]];
Dx2D = First[CUDAMemoryGet[Tx2D]];
Dy2D = First[CUDAMemoryGet[Ty2D]];
```

```
    Dt12D = First[CUDAMemoryGet[Tt12D]];
    Dt22D = First[CUDAMemoryGet[Tt22D]];
    Dn12D = First[CUDAMemoryGet[Tn12D]];
    Dn22D = First[CUDAMemoryGet[Tn22D]];

    HPSmallPieceBoundary2D[Endx2D, Endy2D, TGC3,
     minj, maxj, Dx2D, Dy2D, Dt12D, Dt22D, Dn12D, Dn22D, opt1, n3];

    Transpose[{CUDAMemoryGet[Endx2D], CUDAMemoryGet[Endy2D]}]];
HPIntersection[0, 900, 1]
HPIntersection[180, 900, 1]
HPIntersection[360, 900, 1]
HPIntersection[540, 900, 1]
HPIntersection[720, 900, 1]
HPIntersection[900, 900, 1]
```

```
{{900}}

{{921}}

{{974}}

{{1042}}

{{1117}}

{{1192}}
```

First Boundary in 2D

Upper Cut Curve (LoC) in 2D

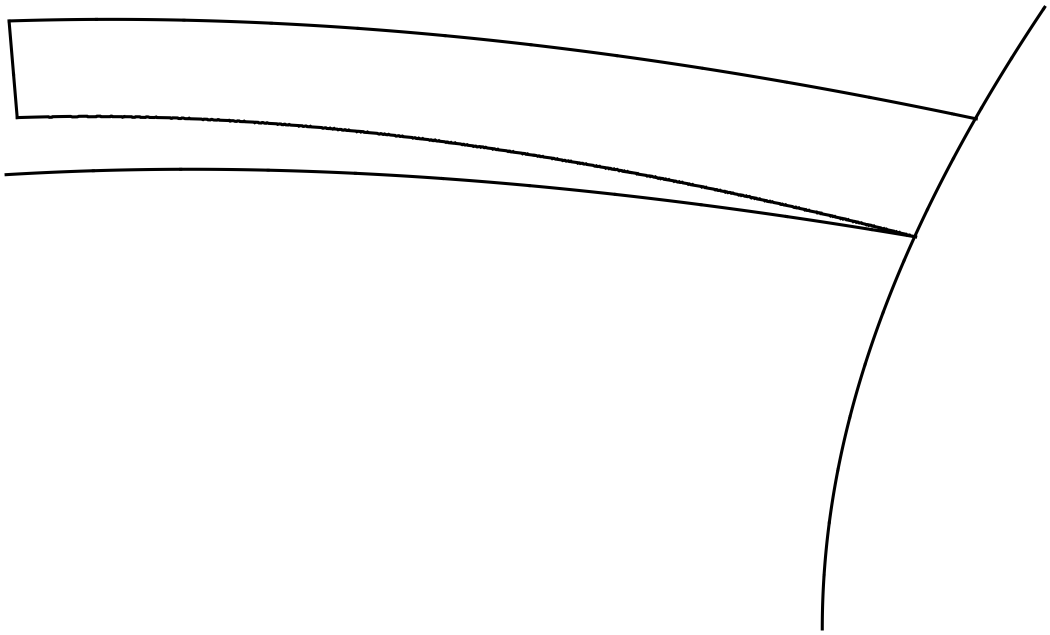
Lower Cut LoC in 2D

Last Boundary in 2D

(a)

Final in 2D

```
ListLinePlot[{PPHPBoundary, PPHPUpperCut540, PPHPUpperCut720, PPHPLowerCut540,
  PPHPLastBoundary720}, PlotStyle → Black, PlotRange → All, Axes → None]
```



(b)

Set up

```
kernelcodeCPLAS2 = "
#define min 0
#define max 1
#define vvv 0
#define uuu 1
#define sign1 -1
#define h 0.001
#define n 1201
#define AlphaLAC (2.0)
#define BetaLASC (2.0)
#define InitRCLAC (1.0)
#define OmegaLASC (1.0)
#define Lambda_Max_LAC (12.718433357775211)
#define ArcLength_Max_LAC (6.764357890136049)
#define Lambda_Min_LAC (12.718433357775211)
#define ArcLength_Min_LAC (6.764357890136049)
```

```
#define Lambda_Max_LASC (6.285247558593747)
#define InitRT_Max_LASC (7.303131835937503)
#define ArcLength_Max_LASC (3.1084204465150833)
#define Lambda_Min_LASC (6.285247558593747)
#define InitRT_Min_LASC (7.303131835937503)
#define ArcLength_Min_LASC (3.1084204465150833)
#define Point00x (0.0)
#define Point00y (0.0)
#define Point00z (0.0)
#define Point03x (-0.7077264466498845)
#define Point03y (0.0)
#define Point03z (0.5008767232876717)
#define Point30x (0.0)
#define Point30y (-0.7077264466498845)
#define Point30z (-0.5008767232876717)
#define Point33x (-1.0013149889663)
#define Point33y (-1.0013149889663)
#define Point33z (0.0)

#define maxPoint00x (0.0)
#define maxPoint00y (0.0)
#define maxPoint00z (0.0)
#define maxPoint03x (-0.7077264357685561)
#define maxPoint03y (0.0)
#define maxPoint03z (0.5008767386038725)
#define maxPoint30x (0.0)
#define maxPoint30y (-0.7077264466498845)
#define maxPoint30z (-0.5008767232876717)
#define maxPoint33x (-1.0013149894892335)
#define maxPoint33y (-1.0013149882505217)
#define maxPoint33z (-0.0000000012420347421705752)


#define ScaleToOrigin_Max_LAC (0.13192213273168868)
#define ScaleToOrigin_Min_LAC (0.13192213273168868)
#define ScaleToOrigin_Max_LASC (0.38164258171833326)
#define ScaleToOrigin_Min_LASC (0.38164258171833326)

#define inidxdu1
    ((1-pv)*(maxPoint00x-maxPoint03x)+pv*(maxPoint30x-maxPoint33x)-v1_x1+
    v2_x1+(1-pv)*u1_t1+pv*u2_t1)
#define inidxdu2
    ((1-pv)*(maxPoint00y-maxPoint03y)+pv*(maxPoint30y-maxPoint33y)-v1_x2+
    v2_x2+(1-pv)*u1_t2+pv*u2_t2)
#define inidxdu3
    ((1-pv)*(maxPoint00z-maxPoint03z)+pv*(maxPoint30z-maxPoint33z)-v1_x3+
    v2_x3+(1-pv)*u1_t3+pv*u2_t3)
```

```
#define inidxdv1
    ((1-pu)*(maxPoint00x-maxPoint30x)+pu*(maxPoint03x-maxPoint33x)-u1_x1+
    u2_x1+(1-pu)*v1_t1+pu*v2_t1)
#define inidxdv2
    ((1-pu)*(maxPoint00y-maxPoint30y)+pu*(maxPoint03y-maxPoint33y)-u1_x2+
    u2_x2+(1-pu)*v1_t2+pu*v2_t2)
#define inidxdv3
    ((1-pu)*(maxPoint00z-maxPoint30z)+pu*(maxPoint03z-maxPoint33z)-u1_x3+
    u2_x3+(1-pu)*v1_t3+pu*v2_t3)

#define inid2xdu21 ((1-pv)*u1_dt1+pv*u2_dt1)
#define inid2xdu22 ((1-pv)*u1_dt2+pv*u2_dt2)
#define inid2xdu23 ((1-pv)*u1_dt3+pv*u2_dt3)
#define inid2xdv21 ((1-pu)*v1_dt1+pu*v2_dt1)
#define inid2xdv22 ((1-pu)*v1_dt2+pu*v2_dt2)
#define inid2xdv23 ((1-pu)*v1_dt3+pu*v2_dt3)
#define inid2xdudv1
    (-maxPoint00x+maxPoint03x+maxPoint30x-maxPoint33x-u1_t1+u2_t1-v1_t1+v2_t1)
#define inid2xdudv2
    (-maxPoint00y+maxPoint03y+maxPoint30y-maxPoint33y-u1_t2+u2_t2-v1_t2+v2_t2)
#define inid2xdudv3
    (-maxPoint00z+maxPoint03z+maxPoint30z-maxPoint33z-u1_t3+u2_t3-v1_t3+v2_t3)

#define inid3xdu31 ((1-pv)*u1_d2t1+pv*u2_d2t1)
#define inid3xdu32 ((1-pv)*u1_d2t2+pv*u2_d2t2)
#define inid3xdu33 ((1-pv)*u1_d2t3+pv*u2_d2t3)
#define inid3xdv31 ((1-pu)*v1_d2t1+pu*v2_d2t1)
#define inid3xdv32 ((1-pu)*v1_d2t2+pu*v2_d2t2)
#define inid3xdv33 ((1-pu)*v1_d2t3+pu*v2_d2t3)
#define inid3xdu2dv1 (-u1_dt1+u2_dt1)
#define inid3xdu2dv2 (-u1_dt2+u2_dt2)
#define inid3xdu2dv3 (-u1_dt3+u2_dt3)
#define inid3xdudv21 (-v1_dt1+v2_dt1)
#define inid3xdudv22 (-v1_dt2+v2_dt2)
#define inid3xdudv23 (-v1_dt3+v2_dt3)

#define Coons_x1
    ((1-pv)*u1_x1+pv*u2_x1+(1-pu)*v1_x1+pu*v2_x1-((1-pu)*(1-pv)*maxPoint00x+
    pu*(1-pv)*maxPoint03x+(1-pu)*pv*maxPoint30x+pu*pv*maxPoint33x))
#define Coons_x2
    ((1-pv)*u1_x2+pv*u2_x2+(1-pu)*v1_x2+pu*v2_x2-((1-pu)*(1-pv)*maxPoint00y+
    pu*(1-pv)*maxPoint03y+(1-pu)*pv*maxPoint30y+pu*pv*maxPoint33y))
#define Coons_x3
    ((1-pv)*u1_x3+pv*u2_x3+(1-pu)*v1_x3+pu*v2_x3-((1-pu)*(1-pv)*maxPoint00z+
    pu*(1-pv)*maxPoint03z+(1-pu)*pv*maxPoint30z+pu*pv*maxPoint33z))
```

```
#define repiniSurNorvec
    iniSurNorvec(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3
    ,v2_t1,v2_t2,v2_t3,normv_C)
#define repiniDSurNvDu
    iniDSurNvDu(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,dnormvdu_C)
#define repiniDSurNvDv
    iniDSurNvDv(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,dnormvdv_C)
#define repiniScalarSurNor
    iniScalarSurNor(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,Scal)
#define repinicoefE
    inicoefE(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3)
#define repiniDEDu
    iniDEDu(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniDEDv
    iniDEDv(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repinicoefF
    inicoefF(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3)
#define repiniDFDu
    iniDFDu(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniDFDv
    iniDFDv(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
```

```
#define repinicoefG
    inicoefG(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3)
#define repiniDGDu
    iniDGDu(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniDGDv
    iniDGDv(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repinicoefL
    inicoefL(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repinicoefM
    inicoefM(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repinicoefN
    inicoefN(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniGaussCurva
    iniGaussCurva(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniMeanCurva
    iniMeanCurva(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniMaxPrinci
    iniMaxPrinci(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniMinPrinci
    iniMinPrinci(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
```

```
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniPrinci
    iniPrinci(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3
    ,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)

#define repiniDcoefL_2
    iniDcoefL_2(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    CoeffL)
#define repiniDcoefM_2
    iniDcoefM_2(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    CoeffM)
#define repiniDcoefN_2
    iniDcoefN_2(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    CoeffN)
#define repiniDGaussCurva_2
    iniDGaussCurva_2(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2
    ,u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    GCurva)
#define repiniDMeanCurva_2
    iniDMeanCurva_2(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    MCurva)
#define repiniDPrinci_2
    iniDPrinci_2(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
```

```
      v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
      v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
      optimum,PCurva)
#define repiniEta
      iniEta(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
      u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
      v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
      v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniMiu
      iniMiu(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
      u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
      v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
      v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniDuDs
      iniDuDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
      u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
      v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
      v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniDvDs
      iniDvDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
      u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
      v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
      v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniDEDs
      iniDEDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
      u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
      v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
      v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniDFDs
      iniDFDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
      u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
      v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
      v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniDGDs
      iniDGDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
      u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
      v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
      v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniDLDs
      iniDLDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
      u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
      u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
      v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
      v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,optimum)
#define repiniDMDs
      iniDMDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
      u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
```

```
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,optimum)
#define repiniDNDs
    iniDNDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,optimum)
#define repiniDPrinciDs
    iniDPrinciDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
    u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    optimum)
#define repiniDvDt
    iniDvDt(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniDuDt
    iniDuDt(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniDSurDs
    iniDSurDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3
    ,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum,tangv)
#define repiniSurNor
    iniSurNor(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,Snor_C)
#define repiniBinormalSur
    iniBinormalSur(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum,binormv_C)
#define repiniAlp1
    iniAlp1(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum,Alp1_C)
#define repiniBeta1
    iniBeta1(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
```

```
    u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,optimum)


#define opt1 (float pu,float pv,float u1_x1,float u1_x2,float u1_x3,float
    u1_t1,float u1_t2,float u1_t3,float u2_x1,float u2_x2,float u2_x3,float
    u2_t1,float u2_t2,float u2_t3,float v1_x1,float v1_x2,float
    v1_x3,float v1_t1,float v1_t2,float v1_t3,float v2_x1,float
    v2_x2,float v2_x3,float v2_t1,float v2_t2,float v2_t3,float normv_C[])
#define opt2 (float pu,float pv,float u1_x1,float u1_x2,float u1_x3,float
    u1_t1,float u1_t2,float u1_t3,float u2_x1,float u2_x2,float
    u2_x3,float u2_t1,float u2_t2,float u2_t3,float v1_x1,float
    v1_x2,float v1_x3,float v1_t1,float v1_t2,float v1_t3,float
    v2_x1,float v2_x2,float v2_x3,float v2_t1,float v2_t2,float v2_t3)
#define opt3 (float pu,float pv,float u1_x1,float u1_x2,float
    u1_x3,float u1_t1,float u1_t2,float u1_t3,float u1_dt1,float
    u1_dt2,float u1_dt3,float u2_x1,float u2_x2,float
    u2_x3,float u2_t1,float u2_t2,float u2_t3,float u2_dt1,float
    u2_dt2,float u2_dt3,float v1_x1,float v1_x2,float v1_x3,float
    v1_t1,float v1_t2,float v1_t3,float v1_dt1,float v1_dt2,float
    v1_dt3,float v2_x1,float v2_x2,float v2_x3,float v2_t1,float
    v2_t2,float v2_t3,float v2_dt1,float v2_dt2,float v2_dt3)
#define opt4 (float pu,float pv,float u1_x1,float u1_x2,float
    u1_x3,float u1_t1,float u1_t2,float u1_t3,float u1_dt1,float
    u1_dt2,float u1_dt3,float u2_x1,float u2_x2,float u2_x3,float
    u2_t1,float u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float
    u2_dt3,float v1_x1,float v1_x2,float v1_x3,float v1_t1,float
    v1_t2,float v1_t3,float v1_dt1,float v1_dt2,float v1_dt3,float
    v2_x1,float v2_x2,float v2_x3,float v2_t1,float v2_t2,float
    v2_t3,float v2_dt1,float v2_dt2,float v2_dt3,mint optimum)
 #define opt5 (float pu,float pv,float u1_x1,float u1_x2,float u1_x3,float
    u1_t1,float u1_t2,float u1_t3,float u1_dt1,float u1_dt2,float
    u1_dt3,float u2_x1,float u2_x2,float u2_x3,float u2_t1,float
    u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float u2_dt3,float
    v1_x1,float v1_x2,float v1_x3,float v1_t1,float v1_t2,float
    v1_t3,float v1_dt1,float v1_dt2,float v1_dt3,float v2_x1,float
    v2_x2,float v2_x3,float v2_t1,float v2_t2,float v2_t3,float
    v2_dt1,float v2_dt2,float v2_dt3,mint optimum,float tangv_C[])
#define opt6 (float pu,float pv,float u1_x1,float u1_x2,float
    u1_x3,float u1_t1,float u1_t2,float u1_t3,float u1_dt1,float
    u1_dt2,float u1_dt3,float u2_x1,float u2_x2,float u2_x3,float
    u2_t1,float u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float
    u2_dt3,float v1_x1,float v1_x2,float v1_x3,float v1_t1,float
    v1_t2,float v1_t3,float v1_dt1,float v1_dt2,float v1_dt3,float
    v2_x1,float v2_x2,float v2_x3,float v2_t1,float v2_t2,float
```

```
    v2_t3,float v2_dt1,float v2_dt2,float v2_dt3,float dnormv_C[])
#define opt7 (float pu,float pv,float u1_x1,float u1_x2,float
    u1_x3,float u1_t1,float u1_t2,float u1_t3,float u1_dt1,float
    u1_dt2,float u1_dt3,float u2_x1,float u2_x2,float u2_x3,float
    u2_t1,float u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float
    u2_dt3,float v1_x1,float v1_x2,float v1_x3,float v1_t1,float
    v1_t2,float v1_t3,float v1_dt1,float v1_dt2,float v1_dt3,float
    v2_x1,float v2_x2,float v2_x3,float v2_t1,float v2_t2,float
    v2_t3,float v2_dt1,float v2_dt2,float v2_dt3,float Scal[])
#define opt8 (float pu,float pv,float u1_x1,float u1_x2,float
    u1_x3,float u1_t1,float u1_t2,float u1_t3,float u1_dt1,float
    u1_dt2,float u1_dt3,float u1_d2t1,float u1_d2t2,float
    u1_d2t3,float u2_x1,float u2_x2,float u2_x3,float u2_t1,float
    u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float u2_dt3,float
    u2_d2t1,float u2_d2t2,float u2_d2t3,float v1_x1,float
    v1_x2,float v1_x3,float v1_t1,float v1_t2,float v1_t3,float
    v1_dt1,float v1_dt2,float v1_dt3,float v1_d2t1,float
    v1_d2t2,float v1_d2t3,float v2_x1,float v2_x2,float v2_x3,float
    v2_t1,float v2_t2,float v2_t3,float v2_dt1,float v2_dt2,float
    v2_dt3,float v2_d2t1,float v2_d2t2,float v2_d2t3,float coeff[])
#define opt9 (float pu,float pv,float u1_x1,float u1_x2,float
    u1_x3,float u1_t1,float u1_t2,float u1_t3,float u1_dt1,float
    u1_dt2,float u1_dt3,float u1_d2t1,float u1_d2t2,float
    u1_d2t3,float u2_x1,float u2_x2,float u2_x3,float u2_t1,float
    u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float u2_dt3,float
    u2_d2t1,float u2_d2t2,float u2_d2t3,float v1_x1,float v1_x2,float
    v1_x3,float v1_t1,float v1_t2,float v1_t3,float v1_dt1,float
    v1_dt2,float v1_dt3,float v1_d2t1,float v1_d2t2,float
    v1_d2t3,float v2_x1,float v2_x2,float v2_x3,float v2_t1,float
    v2_t2,float v2_t3,float v2_dt1,float v2_dt2,float v2_dt3,float
    v2_d2t1,float v2_d2t2,float v2_d2t3,mint optimum,float coeff[])
#define opt10 (float pu,float pv,float u1_x1,float u1_x2,float
    u1_x3,float u1_t1,float u1_t2,float u1_t3,float u1_dt1,float
    u1_dt2,float u1_dt3,float u1_d2t1,float u1_d2t2,float
    u1_d2t3,float u2_x1,float u2_x2,float u2_x3,float u2_t1,float
    u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float u2_dt3,float
    u2_d2t1,float u2_d2t2,float u2_d2t3,float v1_x1,float
    v1_x2,float v1_x3,float v1_t1,float v1_t2,float v1_t3,float
    v1_dt1,float v1_dt2,float v1_dt3,float v1_d2t1,float
    v1_d2t2,float v1_d2t3,float v2_x1,float v2_x2,float v2_x3,float
    v2_t1,float v2_t2,float v2_t3,float v2_dt1,float v2_dt2,float
    v2_dt3,float v2_d2t1,float v2_d2t2,float v2_d2t3,mint optimum)
#define opt11 (float pu,float pv,float u1_x1,float u1_x2,float u1_x3,float
    u1_t1,float u1_t2,float u1_t3,float u1_dt1,float u1_dt2,float
    u1_dt3,float u2_x1,float u2_x2,float u2_x3,float u2_t1,float
    u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float u2_dt3,float
    v1_x1,float v1_x2,float v1_x3,float v1_t1,float v1_t2,float
```

```
    v1_t3,float v1_dt1,float v1_dt2,float v1_dt3,float v2_x1,float
    v2_x2,float v2_x3,float v2_t1,float v2_t2,float v2_t3,float
    v2_dt1,float v2_dt2,float v2_dt3,mint optimum,float Alph_C[])
#define opt12 (float pu,float pv,float u1_x1,float u1_x2,float u1_x3,float
    u1_t1,float u1_t2,float u1_t3,float u2_x1,float u2_x2,float u2_x3,float
    u2_t1,float u2_t2,float u2_t3,float v1_x1,float v1_x2,float
    v1_x3,float v1_t1,float v1_t2,float v1_t3,float v2_x1,float
    v2_x2,float v2_x3,float v2_t1,float v2_t2,float v2_t3,float Snor_C[])
#define opt13 (float pu,float pv,float u1_x1,float u1_x2,float u1_x3,float
    u1_t1,float u1_t2,float u1_t3,float u1_dt1,float u1_dt2,float
    u1_dt3,float u2_x1,float u2_x2,float u2_x3,float u2_t1,float
    u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float u2_dt3,float
    v1_x1,float v1_x2,float v1_x3,float v1_t1,float v1_t2,float
    v1_t3,float v1_dt1,float v1_dt2,float v1_dt3,float v2_x1,float
    v2_x2,float v2_x3,float v2_t1,float v2_t2,float v2_t3,float
    v2_dt1,float v2_dt2,float v2_dt3,mint optimum,float binormv_C[])
#define opt14 (float pu,float pv,float u1_x1,float u1_x2,float
    u1_x3,float u1_t1,float u1_t2,float u1_t3,float u1_dt1,float
    u1_dt2,float u1_dt3,float u1_d2t1,float u1_d2t2,float
    u1_d2t3,float u2_x1,float u2_x2,float u2_x3,float u2_t1,float
    u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float u2_dt3,float
    u2_d2t1,float u2_d2t2,float u2_d2t3,float v1_x1,float v1_x2,float
    v1_x3,float v1_t1,float v1_t2,float v1_t3,float v1_dt1,float
    v1_dt2,float v1_dt3,float v1_d2t1,float v1_d2t2,float
    v1_d2t3,float v2_x1,float v2_x2,float v2_x3,float v2_t1,float
    v2_t2,float v2_t3,float v2_dt1,float v2_dt2,float v2_dt3,float
    v2_d2t1,float v2_d2t2,float v2_d2t3,mint optimum,float x[])

__device__ float inicurvature_LAC(float
    t,float Alpha,float Lambda,float InitRC,float Total_s)
{
if(Alpha==-1)
    {return ((float) 1)/InitRC-Lambda*t*Total_s;}
else if(Alpha==0)
    {return ((float) 1)/exp(Lambda*t*Total_s+log(InitRC));}
else if(Alpha==1)
    {return ((float) 1)/(InitRC+Lambda*t*Total_s);}
else if(Alpha==2)
    {return ((float)
    1)/sqrt(pow(InitRC,2)+((float) 2)*Lambda*t*Total_s);}
else
    {return pow(pow(InitRC,Alpha)+Lambda*Alpha*t*Total_s,-((float) 1)/Alpha);}
}

__device__ float inidcLACdx(float
    t,float Alpha,float Lambda,float InitRC,float Total_s)
{
```

```
if(Alpha==-1)
    {return -Total_s*Lambda;}
else if(Alpha==0)
    {return -Total_s*(Lambda*exp(-Lambda*t*Total_s))/InitRC;}
else if(Alpha==1)
    {return -Total_s*Lambda/pow(InitRC+Lambda*t*Total_s,2);}
else if(Alpha==2)
    {return -Total_s*Lambda/sqrt(pow(pow(InitRC,2)+((float)
    2)*Lambda*t*Total_s,3));}
else
    {return
    -Total_s*Lambda*pow(pow(InitRC,Alpha)+Lambda*Alpha*t*Total_s,-((float)
    1)-((float) 1)/Alpha);}
}
__device__ float inid2cLACdx2(float
    t,float Alpha,float Lambda,float InitRC,float Total_s)
{
if(Alpha==-1)
    {return 0;}
else if(Alpha==0)
    {return
    pow(Total_s,2)*(pow(Lambda,2)*exp(-Lambda*t*Total_s))/InitRC;}
else if(Alpha==1)
    {return pow(Total_s,2)*((float)
    2)*pow(Lambda,2)/pow(InitRC+Lambda*t*Total_s,3);}
else if(Alpha==2)
    {return pow(Total_s,2)*(((float)
    3)*pow(Lambda,2))/sqrt(pow(pow(InitRC,2)+((float)
    2)*Lambda*t*Total_s,5));}
else
    {return (((float)
    1)+Alpha)*pow(Total_s,2)*pow(Lambda,2)*pow(pow(InitRC,Alpha)+Lambda*
    Alpha*t*Total_s,-((float) 2)-((float) 1)/Alpha);}
}

__device__ float iniTorsion_LASC(float
    t,float Beta,float Omega,float InitRT,float Total_s)
{
if(Beta==-1)
    {return ((float) 1)/InitRT-Omega*t*Total_s;}
else if(Beta==0)
    {return ((float) 1)/exp(Omega*t*Total_s+log(InitRT));}
else if(Beta==1)
    {return ((float) 1)/(InitRT+Omega*t*Total_s);}
else if(Beta==2)
    {return ((float)
    1)/sqrt(pow(InitRT,2)+((float) 2)*Omega*t*Total_s);}
```

```
else
    {return pow(pow(InitRT,Beta)+Omega*Beta*t*Total_s,-((float) 1)/Beta);}
}

__device__ float inidtLASCdx(float
    t,float Beta,float Omega,float InitRT,float Total_s)
{
if(Beta==-1)
    {return -Total_s*Omega;}
else if(Beta==0)
    {return -Total_s*(Omega*exp(-Omega*t*Total_s))/InitRT;}
else if(Beta==1)
    {return -Total_s*Omega/pow(InitRT+Omega*t*Total_s,2);}
else if(Beta==2)
    {return
    -Total_s*Omega/sqrt(pow(pow(InitRT,2)+((float) 2)*Omega*t*Total_s,3));}
else
    {return -Total_s*Omega*pow(pow(InitRT,Beta)+Omega*Beta*t*Total_s,-((float)
    1)-((float) 1)/Beta);}
}

__device__ float inid2tLASCdx2(float
    t,float Beta,float Omega,float InitRT,float Total_s)
{
if(Beta==-1)
    {return 0;}
else if(Beta==0)
    {return
    pow(Total_s,2)*(pow(Omega,2)*exp(-Omega*t*Total_s))/InitRT;}
else if(Beta==1)
    {return pow(Total_s,2)*((float)
    2)*pow(Omega,2)/pow(InitRT+Omega*t*Total_s,3);}
else if(Beta==2)
    {return pow(Total_s,2)*(((float)
    3)*pow(Omega,2))/sqrt(pow(pow(InitRT,2)+((float)
    2)*Omega*t*Total_s,5));}
else
    {return (((float)
    1)+Beta)*pow(Total_s,2)*pow(Omega,2)*pow(pow(InitRT,Beta)+Omega*Beta*t*
    Total_s,-((float) 2)-((float) 1)/Beta);}
}

__device__ void iniSurNorvec opt1
{
    normv_C[0] =inidxdu2*inidxdv3-inidxdu3*inidxdv2;
    normv_C[1] =inidxdu3*inidxdv1-inidxdu1*inidxdv3;
    normv_C[2] =inidxdu1*inidxdv2-inidxdu2*inidxdv1;
```

```
}
__device__ void iniDSurNvDu opt6
{
    dnormv_C[0]
     =(inid2xdu22*inidxdv3-inid2xdu23*inidxdv2)+(inidxdu2*inid2xdudv3-
     inidxdu3*inid2xdudv2);
    dnormv_C[1]
     =(inid2xdu23*inidxdv1-inid2xdu21*inidxdv3)+(inidxdu3*inid2xdudv1-
     inidxdu1*inid2xdudv3);
    dnormv_C[2]
     =(inid2xdu21*inidxdv2-inid2xdu22*inidxdv1)+(inidxdu1*inid2xdudv2-
     inidxdu2*inid2xdudv1);
}
__device__ void iniDSurNvDv opt6
{
    dnormv_C[0]
     =(inid2xdudv2*inidxdv3-inid2xdudv3*inidxdv2)+(inidxdu2*inid2xdv23-
     inidxdu3*inid2xdv22);
    dnormv_C[1]
     =(inid2xdudv3*inidxdv1-inid2xdudv1*inidxdv3)+(inidxdu3*inid2xdv21-
     inidxdu1*inid2xdv23);
    dnormv_C[2]
     =(inid2xdudv1*inidxdv2-inid2xdudv2*inidxdv1)+(inidxdu1*inid2xdv22-
     inidxdu2*inid2xdv21);
}

__device__ void iniScalarSurNor opt7
{
float ScalSNor,DScalSNorDu,DScalSNorDv;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];

repiniSurNorvec;
repiniDSurNvDu;
repiniDSurNvDv;


ScalSNor =
     sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
DScalSNorDu
     =
     (normv_C[0]*dnormvdu_C[0]+normv_C[1]*dnormvdu_C[1]+normv_C[2]*dnormvdu_C
     [2])/ScalSNor;
DScalSNorDv =
     (normv_C[0]*dnormvdv_C[0]+normv_C[1]*dnormvdv_C[1]+normv_C[2]*dnormvdv_C
     [2])/ScalSNor;
```

```
Scal[0] = ScalSNor;
Scal[1] = DScalSNorDu;
Scal[2] = DScalSNorDv;
}

__device__ float inicoefE opt2
{
    return inidxdu1*inidxdu1+inidxdu2*inidxdu2+inidxdu3*inidxdu3;
}
__device__ float iniDEDu opt3
{
    return 2*(inidxdu1*inid2xdu21+inidxdu2*inid2xdu22+inidxdu3*inid2xdu23);
}
__device__ float iniDEDv opt3
{
    return 2*(inidxdu1*inid2xdudv1+inidxdu2*inid2xdudv2+inidxdu3*inid2xdudv3);
}


__device__ float inicoefF opt2
{
    return (inidxdu1*inidxdv1+inidxdu2*inidxdv2+inidxdu3*inidxdv3);
}
__device__ float iniDFDu opt3
{
    return inidxdv1*inid2xdu21+inidxdv2*inid2xdu22+inidxdv3*inid2xdu23
     + inidxdu1*inid2xdudv1+inidxdu2*inid2xdudv2+inidxdu3*inid2xdudv3;
}
__device__ float iniDFDv opt3
{
    return
     inidxdv1*inid2xdudv1+inidxdv2*inid2xdudv2+inidxdv3*inid2xdudv3
     + inidxdu1*inid2xdv21+inidxdu2*inid2xdv22+inidxdu3*inid2xdv23;
}

__device__ float inicoefG opt2
{
    return inidxdv1*inidxdv1+inidxdv2*inidxdv2+inidxdv3*inidxdv3;
}
__device__ float iniDGDu opt3
{
    return
     2*(inidxdv1*inid2xdudv1+inidxdv2*inid2xdudv2+inidxdv3*inid2xdudv3);
}
__device__ float iniDGDv opt3
```

```
{
    return
    2*(inidxdv1*inid2xdv21+inidxdv2*inid2xdv22+inidxdv3*inid2xdv23);
}

__device__ float inicoefL opt3
{
float normv_C[3];
repiniSurNorvec;

    return
    (normv_C[0]*inid2xdu21+normv_C[1]*inid2xdu22+normv_C[2]*inid2xdu23)/sqrt
    (normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float inicoefM opt3
{
float normv_C[3];
repiniSurNorvec;

    return
    (normv_C[0]*inid2xdudv1+normv_C[1]*inid2xdudv2+normv_C[2]*inid2xdudv3)/
    sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float inicoefN opt3
{
float normv_C[3];
repiniSurNorvec;

    return
    (normv_C[0]*inid2xdv21+normv_C[1]*inid2xdv22+normv_C[2]*inid2xdv23)/sqrt
    (normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float iniGaussCurva opt3
{
    return
    (repinicoefL*repinicoefN-pow(repinicoefM,2))/(repinicoefE*repinicoefG-
    pow(repinicoefF,2));
}
__device__ float iniMeanCurva opt3
{
    return
    0.5*(repinicoefE*repinicoefN+repinicoefG*repinicoefL-2*repinicoefF*
    repinicoefM)/(repinicoefE*repinicoefG-pow(repinicoefF,2));
}
__device__ float iniMaxPrinci opt3
{
    return repiniMeanCurva+sqrt(pow(repiniMeanCurva,2)-repiniGaussCurva);
```

```
}

__device__ float iniMinPrinci opt3
{
    return repiniMeanCurva-sqrt(pow(repiniMeanCurva,2)-repiniGaussCurva);
}
__device__ float iniPrinci opt4
{
    if(optimum==max)
    {return repiniMaxPrinci;}
    else if(optimum==min)
    {return repiniMinPrinci;}
    else
    {return 0;}
}
__device__ void iniDcoefL_2 opt8
{

float CoeL,DCoeLDu,DCoeLDv;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float Scal[3];

repiniSurNorvec;
repiniDSurNvDu;
repiniDSurNvDv;
repiniScalarSurNor;

CoeL =
    (normv_C[0]*inid2xdu21+normv_C[1]*inid2xdu22+normv_C[2]*inid2xdu23)/Scal[0];
DCoeLDu
    =
    ((dnormvdu_C[0]*inid2xdu21+dnormvdu_C[1]*inid2xdu22+dnormvdu_C[2]*
    inid2xdu23)+(normv_C[0]*inid3xdu31+normv_C[1]*inid3xdu32+normv_C[2]*
    inid3xdu33)-Scal[1]*CoeL)/Scal[0];
DCoeLDv =
    ((dnormvdv_C[0]*inid2xdu21+dnormvdv_C[1]*inid2xdu22+dnormvdv_C[2]*
    inid2xdu23)+(normv_C[0]*inid3xdu2dv1+normv_C[1]*inid3xdu2dv2+normv_C[2]*
    inid3xdu2dv3)-Scal[2]*CoeL)/Scal[0];

coeff[0] = CoeL;
coeff[1] = DCoeLDu;
coeff[2] = DCoeLDv;

}
__device__ void iniDcoefM_2 opt8
```

```
{
float CoeM,DCoeMDu,DCoeMDv;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float Scal[3];
repiniSurNorvec;
repiniDSurNvDu;
repiniDSurNvDv;
repiniScalarSurNor;

CoeM =
    (normv_C[0]*inid2xdudv1+normv_C[1]*inid2xdudv2+normv_C[2]*inid2xdudv3)/
    Scal[0];
DCoeMDu =
    ((dnormvdu_C[0]*inid2xdudv1+dnormvdu_C[1]*inid2xdudv2+dnormvdu_C[2]*
    inid2xdudv3)+(normv_C[0]*inid3xdu2dv1+normv_C[1]*inid3xdu2dv2+normv_C[2]
    *inid3xdu2dv3)-Scal[1]*CoeM)/Scal[0];
DCoeMDv =
    ((dnormvdv_C[0]*inid2xdudv1+dnormvdv_C[1]*inid2xdudv2+dnormvdv_C[2]*
    inid2xdudv3)+(normv_C[0]*inid3xdudv21+normv_C[1]*inid3xdudv22+normv_C[2]
    *inid3xdudv23)-Scal[2]*CoeM)/Scal[0];


coeff[0] = CoeM;
coeff[1] = DCoeMDu;
coeff[2] = DCoeMDv;
}
__device__ void iniDcoefN_2 opt8
{
float CoeN,DCoeNDu,DCoeNDv;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float Scal[3];
repiniSurNorvec;
repiniDSurNvDu;
repiniDSurNvDv;
repiniScalarSurNor;

CoeN =
    (normv_C[0]*inid2xdv21+normv_C[1]*inid2xdv22+normv_C[2]*inid2xdv23)/Scal[0];
DCoeNDu
    =
    ((dnormvdu_C[0]*inid2xdv21+dnormvdu_C[1]*inid2xdv22+dnormvdu_C[2]*
    inid2xdv23)+(normv_C[0]*inid3xdudv21+normv_C[1]*inid3xdudv22+normv_C[2]*
    inid3xdudv23)-Scal[1]*CoeN)/Scal[0];
```

```
DCoeNDv =
    ((dnormvdv_C[0]*inid2xdv21+dnormvdv_C[1]*inid2xdv22+dnormvdv_C[2]*
    inid2xdv23)+(normv_C[0]*inid3xdv31+normv_C[1]*inid3xdv32+normv_C[2]*
    inid3xdv33)-Scal[2]*CoeN)/Scal[0];

coeff[0] = CoeN;
coeff[1] = DCoeNDu;
coeff[2] = DCoeNDv;

}
__device__ void iniDGaussCurva_2 opt8
{
float GCurv,DGCurvDu,DGCurvDv;
float CoeffL[3];
float CoeffM[3];
float CoeffN[3];
repiniDcoefL_2;
repiniDcoefM_2;
repiniDcoefN_2;

GCurv=
    (CoeffL[0]*CoeffN[0]-pow(CoeffM[0],2))/(repinicoefE*repinicoefG-pow(
    repinicoefF,2));
DGCurvDu =
    ((pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repinicoefG*repiniDEDu-2*
    repinicoefF*repiniDFDu+repinicoefE*repiniDGDu)+(-pow(repinicoefF,2)+
    repinicoefE*repinicoefG)*(CoeffN[0]*CoeffL[1]-2*CoeffM[0]*CoeffM[1]+
    CoeffL[0]*CoeffN[1]))/(pow(pow(repinicoefF,2)-repinicoefE*repinicoefG,2)
    );
DGCurvDv
    =((pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repinicoefG*repiniDEDv-2*
    repinicoefF*repiniDFDv+repinicoefE*repiniDGDv)+(-pow(repinicoefF,2)+
    repinicoefE*repinicoefG)*(CoeffN[0]*CoeffL[2]-2*CoeffM[0]*CoeffM[2]+
    CoeffL[0]*CoeffN[2]))/(pow(pow(repinicoefF,2)-repinicoefE*repinicoefG,2)
    );


coeff[0] = GCurv;
coeff[1] = DGCurvDu;
coeff[2] = DGCurvDv;

}
__device__ void iniDMeanCurva_2 opt8
{
float MCurv,DMCurvDu,DMCurvDv;
float CoeffL[3];
float CoeffM[3];
```

```
float CoeffN[3];
repiniDcoefL_2;
repiniDcoefM_2;
repiniDcoefN_2;

MCurv =
    0.5*(repinicoefE*CoeffN[0]+repinicoefG*CoeffL[0]-2*repinicoefF*CoeffM[0]
    )/(repinicoefE*repinicoefG-pow(repinicoefF,2));
DMCurvDu =
    0.5*(-(repinicoefG*CoeffL[0]-2*repinicoefF*CoeffM[0]+repinicoefE*CoeffN[
    0])*(repinicoefG*repiniDEDu-2*repinicoefF*repiniDFDu+repinicoefE*
    repiniDGDu)+(-pow(repinicoefF,2)+repinicoefE*repinicoefG)*(CoeffN[0]*
    repiniDEDu-2*CoeffM[0]*repiniDFDu+CoeffL[0]*repiniDGDu+repinicoefG*
    CoeffL[1]-2*repinicoefF*CoeffM[1]+repinicoefE*CoeffN[1]))/(pow(pow(
    repinicoefF,2)-repinicoefE*repinicoefG,2));
DMCurvDv =
    0.5*(-(repinicoefG*CoeffL[0]-2*repinicoefF*CoeffM[0]+repinicoefE*CoeffN[
    0])*(repinicoefG*repiniDEDv-2*repinicoefF*repiniDFDv+repinicoefE*
    repiniDGDv)+(-pow(repinicoefF,2)+repinicoefE*repinicoefG)*(CoeffN[0]*
    repiniDEDv-2*CoeffM[0]*repiniDFDv+CoeffL[0]*repiniDGDv+repinicoefG*
    CoeffL[2]-2*repinicoefF*CoeffM[2]+repinicoefE*CoeffN[2]))/(pow(pow(
    repinicoefF,2)-repinicoefE*repinicoefG,2));



coeff[0] = MCurv;
coeff[1] = DMCurvDu;
coeff[2] = DMCurvDv;
}
__device__ void iniDPrinci_2 opt9
{
float Princip,DPrincipDu,DPrincipDv;
float GCurva[3];
float MCurva[3];
repiniDGaussCurva_2;
repiniDMeanCurva_2;

if(optimum==max)
{Princip = MCurva[0]+sqrt(pow(MCurva[0],2)-GCurva[0]);
DPrincipDu =
    MCurva[1]+(2*MCurva[0]*MCurva[1]-GCurva[1])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
DPrincipDv =
    MCurva[2]+(2*MCurva[0]*MCurva[2]-GCurva[2])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));

}
```

```
else if(optimum==min)
{Princip = MCurva[0]-sqrt(pow(MCurva[0],2)-GCurva[0]);
DPrincipDu =
    MCurva[1]-(2*MCurva[0]*MCurva[1]-GCurva[1])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
DPrincipDv =
    MCurva[2]-(2*MCurva[0]*MCurva[2]-GCurva[2])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));

}
else
{Princip = 0;
DPrincipDu = 0;
DPrincipDv = 0;
}

coeff[0] = Princip;
coeff[1] = DPrincipDu;
coeff[2] = DPrincipDv;

}

__device__ float iniEta opt4
{
    return
    sign1/sqrt(repinicoefE*pow(repinicoefM-repiniPrinci*repinicoefF,2)-2*
    repinicoefF*(repinicoefM-repiniPrinci*repinicoefF)*(repinicoefL-
    repiniPrinci*repinicoefE)+repinicoefG*pow(repinicoefL-repiniPrinci*
    repinicoefE,2));
}
__device__ float iniMiu opt4
{
    return
    sign1/sqrt(repinicoefE*pow(repinicoefN-repiniPrinci*repinicoefG,2)-2*
    repinicoefF*(repinicoefM-repiniPrinci*repinicoefF)*(repinicoefN-
    repiniPrinci*repinicoefG)+repinicoefG*pow(repinicoefM-repiniPrinci*
    repinicoefF,2));
}
__device__ float iniDuDs opt4
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-
    repiniPrinci*repinicoefG))
    {return repiniEta*(repinicoefM-repiniPrinci*repinicoefF);}
else
    {return repiniMiu*(repinicoefN-repiniPrinci*repinicoefG);}
}
__device__ float iniDvDs opt4
```

```
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
    repinicoefG))
    {return -repiniEta*(repinicoefL-repiniPrinci*repinicoefE);}
else
    {return -repiniMiu*(repinicoefM-repiniPrinci*repinicoefF);}
}


__device__ float iniDEDs opt4
{
    return repiniDEDu*repiniDuDs+repiniDEDv*repiniDvDs;
}
__device__ float iniDFDs opt4
{
    return repiniDFDu*repiniDuDs+repiniDFDv*repiniDvDs;
}
__device__ float iniDGDs opt4
{
    return repiniDGDu*repiniDuDs+repiniDGDv*repiniDvDs;
}
__device__ float iniDLDs opt10
{
float CoeffL[3];
repiniDcoefL_2;

    return CoeffL[1]*repiniDuDs+CoeffL[2]*repiniDvDs;
}
__device__ float iniDMDs opt10
{
float CoeffM[3];
repiniDcoefM_2;

    return CoeffM[1]*repiniDuDs+CoeffM[2]*repiniDvDs;
}
__device__ float iniDNDs opt10
{
float CoeffN[3];
repiniDcoefN_2;

    return CoeffN[1]*repiniDuDs+CoeffN[2]*repiniDvDs;
}
__device__ float iniDPrinciDs opt10
{
float PCurva[3];
repiniDPrinci_2;
```

```
    return PCurva[1]*repiniDuDs+PCurva[2]*repiniDvDs;
}
__device__ float iniDuDt opt4
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
    repinicoefG))
    {return (repinicoefM-repiniPrinci*repinicoefF);}
else
    {return (repinicoefN-repiniPrinci*repinicoefG);}
}
__device__ float iniDvDt opt4
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
    repinicoefG))
    {return (repinicoefL-repiniPrinci*repinicoefE);}
else
    {return (repinicoefM-repiniPrinci*repinicoefF);}
}
__device__ void iniDSurDs opt5
{
    tangv_C[0] =inidxdu1*repiniDuDs+inidxdv1*repiniDvDs;
    tangv_C[1] =inidxdu2*repiniDuDs+inidxdv2*repiniDvDs;
    tangv_C[2] =inidxdu3*repiniDuDs+inidxdv3*repiniDvDs;
}

__device__ void iniAlp1 opt11
{
    Alph_C[0]=(inid2xdu21*pow(repiniDuDs,2)+2*inid2xdudv1*repiniDuDs*
    repiniDvDs+inid2xdv21*pow(repiniDvDs,2));

    Alph_C[1]=(inid2xdu22*pow(repiniDuDs,2)+2*inid2xdudv2*repiniDuDs*
    repiniDvDs+inid2xdv22*pow(repiniDvDs,2));

    Alph_C[2]=(inid2xdu23*pow(repiniDuDs,2)+2*inid2xdudv3*repiniDuDs*
    repiniDvDs+inid2xdv23*pow(repiniDvDs,2));
}
__device__ void iniSurNor opt12
{
    float normv_C[3];
    repiniSurNorvec;
    Snor_C[0]
    =normv_C[0]/sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*
    normv_C[2]);
    Snor_C[1]
    =normv_C[1]/sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*
    normv_C[2]);
    Snor_C[2]
```

```
    =normv_C[2]/sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*
    normv_C[2]);
}

__device__ void iniBinormalSur opt13
{
    float Snor_C[3];
    float tangv[3];
    repiniSurNor;
    repiniDSurDs;

    binormv_C[0] =Snor_C[1]*tangv[2] -Snor_C[2]*tangv[1];
    binormv_C[1] =Snor_C[2]*tangv[0] -Snor_C[0]*tangv[2];
    binormv_C[2] =Snor_C[0]*tangv[1] -Snor_C[1]*tangv[0];
}
__device__ float iniBeta1 opt10
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
    repinicoefG))
    {return
    (-(repiniDLDs-repiniDPrinciDs*repinicoefE-repiniPrinci*repiniDEDs)*
    repiniDuDs-(repiniDMDs-repiniDPrinciDs*repinicoefF-repiniPrinci*
    repiniDFDs)*repiniDvDs);}
else
    {return
    (-(repiniDMDs-repiniDPrinciDs*repinicoefF-repiniPrinci*repiniDFDs)*
    repiniDuDs-(repiniDNDs-repiniDPrinciDs*repinicoefG-repiniPrinci*
    repiniDGDs)*repiniDvDs);}
}

__device__ void LUDecomposition3x3 opt14
{
    mint rc=3;
    float binormv_C[3];
    float Alp1_C[3];
    repiniBinormalSur;
    repiniAlp1;
    float matrixA[3][3]={

    {repinicoefE,repinicoefF,-(binormv_C[0]*inidxdu1+binormv_C[1]*inidxdu2+
    binormv_C[2]*inidxdu3)},

    {repinicoefF,repinicoefG,-(binormv_C[0]*inidxdv1+binormv_C[1]*inidxdv2+
    binormv_C[2]*inidxdv3)},
    {repiniDvDt,repiniDuDt,0}};

    float
```

```
matrixB[3]={-(Alp1_C[0]*inidxdu1+Alp1_C[1]*inidxdu2+Alp1_C[2]*inidxdu3),
-(Alp1_C[0]*inidxdv1+Alp1_C[1]*inidxdv2+Alp1_C[2]*inidxdv3),repiniBeta1}
;
float Lower[3][3];
float Upper[3][3];
float y[3];
float sum;
for(mint ii=0; ii<rc; ii++)
        {
            for(mint jj=0; jj<rc; jj++)
            {
                sum = 0;
                if(ii==jj)
                {
                Lower[ii][jj]=1;
                for (mint kk = 0; kk < ii; kk++)
                    {sum += Lower[ii][kk] * Upper[kk][jj];}
                Upper[ii][jj] = matrixA[ii][jj] - sum;
                }
                else if(ii < jj)
                {
                Lower[ii][jj]=0;
                for (mint kk = 0; kk < ii; kk++)
                    {sum += Lower[ii][kk] * Upper[kk][jj];}
                Upper[ii][jj] = matrixA[ii][jj] - sum;
                }
                else
                {
                Upper[ii][jj]=0;
                for (mint kk = 0; kk < jj; kk++)
                    {sum += Lower[ii][kk] * Upper[kk][jj];}
                Lower[ii][jj] =(matrixA[ii][jj] - sum)/Upper[jj][jj];
                }
            }
        }
for (mint ii = 0; ii < rc; ii++)
{
    sum = 0;
    for (mint jj = 0; jj < ii; jj++)
        {sum += Lower[ii][jj] * y[jj];}
    y[ii] = matrixB[ii] - sum;
}
for (mint ii = rc - 1; ii >= 0; ii--)
{
    sum = 0;
    for (mint jj = ii + 1; jj < rc; jj++)
        {sum += Upper[ii][jj] * x[jj];}
```

```
        x[ii] = (y[ii] - sum)/Upper[ii][ii];
    }
}


__device__ void RK4_LAC( float x1, float t1, float n1, float Alpha, float
    InitRC, float Lambda, float ArcLength, float initp, float stepsize,
    float& increment_x1, float& increment_t1, float& increment_n1)

{
float x_k1,x_k2,x_k3,x_k4;
float t_k1,t_k2,t_k3,t_k4;
float n_k1,n_k2,n_k3,n_k4;
float newx1,newt1,newn1;

x_k1=stepsize*t1;
t_k1=stepsize*ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
    n1);
n_k1=stepsize*ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,
    ArcLength)*t1);

newx1=x1+0.5*x_k1;
newt1=t1+0.5*t_k1;
newn1=n1+0.5*n_k1;
x_k2=stepsize*newt1;
t_k2=stepsize*ArcLength*(inicurvature_LAC(initp+0.5*stepsize,Alpha,Lambda,
    InitRC,ArcLength)*newn1);
n_k2=stepsize*ArcLength*(-inicurvature_LAC(initp+0.5*stepsize,Alpha,
    Lambda,InitRC,ArcLength)*newt1);

newx1=x1+0.5*x_k2;
newt1=t1+0.5*t_k2;
newn1=n1+0.5*n_k2;
x_k3=stepsize*newt1;
t_k3=stepsize*ArcLength*(inicurvature_LAC(initp+0.5*stepsize,Alpha,Lambda,
    InitRC,ArcLength)*newn1);
n_k3=stepsize*ArcLength*(-inicurvature_LAC(initp+0.5*stepsize,Alpha,
    Lambda,InitRC,ArcLength)*newt1);

newx1=x1+x_k3;
newt1=t1+t_k3;
newn1=n1+n_k3;
x_k4=stepsize*newt1;
t_k4=stepsize*ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*newn1);
n_k4=stepsize*ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,
    InitRC,ArcLength)*newt1);
```

```
increment_x1=(1.0/6.0)*(x_k1+2*x_k2+2*x_k3+x_k4);
increment_t1=(1.0/6.0)*(t_k1+2*t_k2+2*t_k3+t_k4);
increment_n1=(1.0/6.0)*(n_k1+2*n_k2+2*n_k3+n_k4);

}
__device__ void RK4_LASC( float x1, float t1, float n1, float b1, float Alpha,
     float Beta, float InitRC, float Omega, float Lambda, float InitRT,
     float ArcLength, float initp, float stepsize, float& increment_x1,
     float& increment_t1, float& increment_n1, float& increment_b1)

{
float x_k1,x_k2,x_k3,x_k4;
float t_k1,t_k2,t_k3,t_k4;
float n_k1,n_k2,n_k3,n_k4;
float b_k1,b_k2,b_k3,b_k4;
float newx1,newt1,newn1,newb1;

x_k1=stepsize*t1;
t_k1=stepsize*ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
     n1);
n_k1=stepsize*ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,
     ArcLength)*t1+iniTorsion_LASC(initp,Beta,Omega,InitRT,ArcLength)*b1);
b_k1=stepsize*ArcLength*(-iniTorsion_LASC(initp,Beta,Omega,InitRT,
     ArcLength)*n1);

newx1=x1+0.5*x_k1;
newt1=t1+0.5*t_k1;
newn1=n1+0.5*n_k1;
newb1=b1+0.5*b_k1;
x_k2=stepsize*newt1;
t_k2=stepsize*ArcLength*(inicurvature_LAC(initp+0.5*stepsize,Alpha,Lambda,
     InitRC,ArcLength)*newn1);
n_k2=stepsize*ArcLength*(-inicurvature_LAC(initp+0.5*stepsize,Alpha,
     Lambda,InitRC,ArcLength)*newt1+iniTorsion_LASC(initp+0.5*stepsize,Beta,
     Omega,InitRT,ArcLength)*newb1);
b_k2=stepsize*ArcLength*(-iniTorsion_LASC(initp+0.5*stepsize,Beta,Omega,
     InitRT,ArcLength)*newn1);

newx1=x1+0.5*x_k2;
newt1=t1+0.5*t_k2;
newn1=n1+0.5*n_k2;
newb1=b1+0.5*b_k2;
x_k3=stepsize*newt1;
t_k3=stepsize*ArcLength*(inicurvature_LAC(initp+0.5*stepsize,Alpha,Lambda,
     InitRC,ArcLength)*newn1);
```

```
n_k3=stepsize*ArcLength*(-inicurvature_LAC(initp+0.5*stepsize,Alpha,
     Lambda,InitRC,ArcLength)*newt1+iniTorsion_LASC(initp+0.5*stepsize,Beta,
     Omega,InitRT,ArcLength)*newb1);
b_k3=stepsize*ArcLength*(-iniTorsion_LASC(initp+0.5*stepsize,Beta,Omega,
     InitRT,ArcLength)*newn1);

newx1=x1+x_k3;
newt1=t1+t_k3;
newn1=n1+n_k3;
newb1=b1+b_k3;
x_k4=stepsize*newt1;
t_k4=stepsize*ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
     ArcLength)*newn1);
n_k4=stepsize*ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,
     InitRC,ArcLength)*newt1+iniTorsion_LASC(initp+stepsize,Beta,Omega,InitRT
     ,ArcLength)*newb1);
b_k4=stepsize*ArcLength*(-iniTorsion_LASC(initp+stepsize,Beta,Omega,
     InitRT,ArcLength)*newn1);

increment_x1=(1.0/6.0)*(x_k1+2*x_k2+2*x_k3+x_k4);
increment_t1=(1.0/6.0)*(t_k1+2*t_k2+2*t_k3+t_k4);
increment_n1=(1.0/6.0)*(n_k1+2*n_k2+2*n_k3+n_k4);
increment_b1=(1.0/6.0)*(b_k1+2*b_k2+2*b_k3+b_k4);

}

__device__ void First_Derivatives( float& dt1, float&
     dt2, float& dt3, float n1, float n2, float n3, float initp,
     float Alpha, float InitRC, float Lambda, float ArcLength)
{
dt1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n1);
dt2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n2);
dt3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n3);
}


__device__ void FS_2ndDerivatives_LAC( float& dt1, float& dt2,
     float& dt3, float& d2t1, float& d2t2, float& d2t3, float t1,
     float t2, float t3, float n1, float n2, float n3, float initp,
     float Alpha, float InitRC, float Lambda, float ArcLength)
{
float dn1, dn2, dn3;

dt1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n1);
dt2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n2);
dt3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n3);
dn1=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t1);
```

```
dn2=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t2);
dn3=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t3);

d2t1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*dn1+
     inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n1);
d2t2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
     dn2+inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n2);
d2t3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
     dn3+inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n3);

}

__device__ void FS_2ndDerivatives_LASC( float& dt1, float& dt2, float&
     dt3, float& d2t1, float& d2t2, float& d2t3, float t1, float t2,
     float t3, float n1, float n2, float n3, float b1, float b2,
     float b3, float initp, float Alpha, float Beta, float InitRC,
     float Omega, float Lambda, float InitRT, float ArcLength)
{
float dn1, dn2, dn3;

dt1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n1);
dt2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n2);
dt3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n3);
dn1=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t1+
     iniTorsion_LASC(initp,Beta,Omega,InitRT,ArcLength)*b1);
dn2=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t2
     +iniTorsion_LASC(initp,Beta,Omega,InitRT,ArcLength)*b2);
dn3=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t3
     +iniTorsion_LASC(initp,Beta,Omega,InitRT,ArcLength)*b3);

d2t1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*dn1+
     inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n1);
d2t2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
     dn2+inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n2);
d2t3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
     dn3+inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n3);
}


__device__ void FS_Derivatives_LAC( float& dt1, float& dt2, float& dt3, float&
     d2t1, float& d2t2, float& d2t3, float& d3t1, float& d3t2, float&
     d3t3, float t1, float t2, float t3, float n1, float n2, float n3,
     float initp, float Alpha, float InitRC, float Lambda, float ArcLength)
{
float dn1, dn2, dn3, d2n1, d2n2, d2n3;

dt1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n1);
```

```
dt2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n2);
dt3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n3);
dn1=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t1);
dn2=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t2);
dn3=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t3);

d2t1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*dn1+
     inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n1);
d2t2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
     dn2+inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n2);
d2t3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
     dn3+inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n3);
d2n1=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
     dt1-inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*t1);
d2n2=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
     dt2-inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*t2);
d2n3=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
     dt3-inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*t3);

d3t1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*d2n1+2.0*
     inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*dn1+inid2cLACdx2(initp,Alpha
     ,Lambda,InitRC,ArcLength)*n1);
d3t2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
     d2n2+2.0*inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*dn2+inid2cLACdx2(
     initp,Alpha,Lambda,InitRC,ArcLength)*n2);
d3t3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
     d2n3+2.0*inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*dn3+inid2cLACdx2(
     initp,Alpha,Lambda,InitRC,ArcLength)*n3);
}


__device__ void FS_Derivatives_LASC( float& dt1, float& dt2, float& dt3,
     float& d2t1, float& d2t2, float& d2t3, float& d3t1, float& d3t2, float&
     d3t3, float t1, float t2, float t3, float n1, float n2, float n3,
     float b1, float b2, float b3, float initp, float Alpha, float Beta,
     float InitRC, float Omega, float Lambda, float InitRT, float ArcLength)
{
float dn1, dn2, dn3, d2n1, d2n2, d2n3;
float db1, db2, db3;

dt1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n1);
dt2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n2);
dt3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*n3);
dn1=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t1+
     iniTorsion_LASC(initp,Beta,Omega,InitRT,ArcLength)*b1);
dn2=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t2
     +iniTorsion_LASC(initp,Beta,Omega,InitRT,ArcLength)*b2);
```

```
dn3=ArcLength*(-inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*t3
     +iniTorsion_LASC(initp,Beta,Omega,InitRT,ArcLength)*b3);
db1=ArcLength*(-iniTorsion_LASC(initp,Beta,Omega,InitRT,ArcLength)*n1);
db2=ArcLength*(-iniTorsion_LASC(initp,Beta,Omega,InitRT,ArcLength)*n2);
db3=ArcLength*(-iniTorsion_LASC(initp,Beta,Omega,InitRT,ArcLength)*n3);

d2t1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*dn1+
     inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n1);
d2t2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
     dn2+inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n2);
d2t3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
     dn3+inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*n3);
d2n1=ArcLength*(-inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*t1-
     inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*dt1+inidtLASCdx(
     initp,Beta,Omega,InitRT,ArcLength)*b1+iniTorsion_LASC(initp,Beta,Omega,
     InitRT,ArcLength)*db1);
d2n2=ArcLength*(-inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*t2-
     inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*dt2+inidtLASCdx(
     initp,Beta,Omega,InitRT,ArcLength)*b2+iniTorsion_LASC(initp,Beta,Omega,
     InitRT,ArcLength)*db2);
d2n3=ArcLength*(-inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*t3-
     inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*dt3+inidtLASCdx(
     initp,Beta,Omega,InitRT,ArcLength)*b3+iniTorsion_LASC(initp,Beta,Omega,
     InitRT,ArcLength)*db3);

d3t1=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*d2n1+2.0*
     inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*dn1+inid2cLACdx2(initp,
     Alpha,Lambda,InitRC,ArcLength)*n1);
d3t2=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
     d2n2+2.0*inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*dn2+
     inid2cLACdx2(initp,Alpha,Lambda,InitRC,ArcLength)*n2);
d3t3=ArcLength*(inicurvature_LAC(initp,Alpha,Lambda,InitRC,ArcLength)*
     d2n3+2.0*inidcLACdx(initp,Alpha,Lambda,InitRC,ArcLength)*dn3+
     inid2cLACdx2(initp,Alpha,Lambda,InitRC,ArcLength)*n3);
}

__device__ void FS_LAC( float& x1, float& x2, float& x3, float& t1, float& t2,
     float& t3, float& n1, float& n2, float& n3, float Alpha, float
     InitRC, float Lambda, float ArcLength, float initp, float stepsize)
{
float increment_x1,increment_t1,increment_n1;
float increment_x2,increment_t2,increment_n2;
float increment_x3,increment_t3,increment_n3;

RK4_LAC(x1,t1,n1,Alpha,InitRC,Lambda,ArcLength,initp,stepsize,increment_x1,
     increment_t1,increment_n1);
RK4_LAC(x2,t2,n2,Alpha,InitRC,Lambda,ArcLength,initp,stepsize,
```

```
    increment_x2,increment_t2,increment_n2);
RK4_LAC(x3,t3,n3,Alpha,InitRC,Lambda,ArcLength,initp,stepsize,
    increment_x3,increment_t3,increment_n3);

x1+=increment_x1;
x2+=increment_x2;
x3+=increment_x3;
t1+=increment_t1;
t2+=increment_t2;
t3+=increment_t3;
n1+=increment_n1;
n2+=increment_n2;
n3+=increment_n3;
}

__device__ void FS_LASC( float& x1, float& x2, float& x3, float& t1, float&
    t2, float& t3, float& n1, float& n2, float& n3, float& b1, float& b2,
    float& b3, float Alpha, float Beta, float InitRC, float Omega, float
    Lambda, float InitRT, float ArcLength, float initp, float stepsize)
{
float increment_x1,increment_t1,increment_n1,increment_b1;
float increment_x2,increment_t2,increment_n2,increment_b2;
float increment_x3,increment_t3,increment_n3,increment_b3;

RK4_LASC(x1,t1,n1,b1,Alpha,Beta,InitRC,Omega,Lambda,InitRT,ArcLength,initp,
    stepsize,increment_x1,increment_t1,increment_n1,increment_b1);
RK4_LASC(x2,t2,n2,b2,Alpha,Beta,InitRC,Omega,Lambda,InitRT,ArcLength,
    initp,stepsize,increment_x2,increment_t2,increment_n2,increment_b2);
RK4_LASC(x3,t3,n3,b3,Alpha,Beta,InitRC,Omega,Lambda,InitRT,ArcLength,
    initp,stepsize,increment_x3,increment_t3,increment_n3,increment_b3);

x1+=increment_x1;
x2+=increment_x2;
x3+=increment_x3;
t1+=increment_t1;
t2+=increment_t2;
t3+=increment_t3;
n1+=increment_n1;
n2+=increment_n2;
n3+=increment_n3;
b1+=increment_b1;
b2+=increment_b2;
b3+=increment_b3;
}

__device__ void RK4_func_LoC( float pu, float pv, float u1_x1, float u1_x2,
    float u1_x3, float u1_t1, float u1_t2, float u1_t3, float u1_n1,
```

```
    float u1_n2, float u1_n3, float u2_x1, float u2_x2, float u2_x3,
    float u2_t1, float u2_t2, float u2_t3, float u2_n1, float u2_n2,
    float u2_n3, float u2_b1, float u2_b2, float u2_b3, float v1_x1,
    float v1_x2, float v1_x3, float v1_t1, float v1_t2, float v1_t3,
    float v1_n1, float v1_n2, float v1_n3, float v2_x1, float v2_x2,
    float v2_x3, float v2_t1, float v2_t2, float v2_t3, float v2_n1,
    float v2_n2, float v2_n3, float v2_b1, float v2_b2, float v2_b3, mint
    optimum, float stepsize, float& increment_u, float& increment_v)
{
float u1_dt1, u1_dt2, u1_dt3, u2_dt1, u2_dt2,
    u2_dt3, v1_dt1, v1_dt2, v1_dt3, v2_dt1, v2_dt2, v2_dt3;
float pu_k1,pu_k2,pu_k3,pu_k4;
float pv_k1,pv_k2,pv_k3,pv_k4;
float increment_u1_x1, increment_u1_x2, increment_u1_x3,
    increment_u1_t1, increment_u1_t2, increment_u1_t3,
    increment_u1_n1, increment_u1_n2, increment_u1_n3;
float increment_u2_x1, increment_u2_x2, increment_u2_x3, increment_u2_t1,
    increment_u2_t2, increment_u2_t3, increment_u2_n1, increment_u2_n2,
    increment_u2_n3, increment_u2_b1, increment_u2_b2, increment_u2_b3;
float increment_v1_x1, increment_v1_x2, increment_v1_x3,
    increment_v1_t1, increment_v1_t2, increment_v1_t3,
    increment_v1_n1, increment_v1_n2, increment_v1_n3;
float increment_v2_x1, increment_v2_x2, increment_v2_x3, increment_v2_t1,
    increment_v2_t2, increment_v2_t3, increment_v2_n1, increment_v2_n2,
    increment_v2_n3, increment_v2_b1, increment_v2_b2, increment_v2_b3;
float new_pu, new_pv;
float
    new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,new_u1_t2,new_u1_t3,new_u1_dt1,
    new_u1_dt2,new_u1_dt3,new_u1_n1,new_u1_n2,new_u1_n3;
float
    new_u2_x1,new_u2_x2,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,
    new_u2_dt2,new_u2_dt3,new_u2_n1,new_u2_n2,new_u2_n3;
float
    new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,new_v1_dt1,
    new_v1_dt2,new_v1_dt3,new_v1_n1,new_v1_n2,new_v1_n3;
float
    new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,new_v2_t2,new_v2_t3,new_v2_dt1,
    new_v2_dt2,new_v2_dt3,new_v2_n1,new_v2_n2,new_v2_n3;


First_Derivatives(u1_dt1,u1_dt2,u1_dt3,u1_n1,u1_n2,u1_n3,pu,AlphaLAC,InitRCLAC,
    Lambda_Max_LAC,ArcLength_Max_LAC);
First_Derivatives(u2_dt1,u2_dt2,u2_dt3,u2_n1,u2_n2,u2_n3,pu,AlphaLAC,
    InitRCLAC,Lambda_Max_LASC,ArcLength_Max_LASC);
First_Derivatives(v1_dt1,v1_dt2,v1_dt3,v1_n1,v1_n2,v1_n3,pv,AlphaLAC,
    InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC);
First_Derivatives(v2_dt1,v2_dt2,v2_dt3,v2_n1,v2_n2,v2_n3,pv,AlphaLAC,
```

```
        InitRCLAC,Lambda_Min_LASC,ArcLength_Min_LASC);

    pu_k1=stepsize*iniDuDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
        u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
        v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
        v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum);
    pv_k1=stepsize*iniDvDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,
        u1_dt2,u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,
        v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,
        v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum);

    RK4_LAC(u1_x1,u1_t1,u1_n1,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC,
        pu,0.5*pu_k1,increment_u1_x1,increment_u1_t1,increment_u1_n1);
    RK4_LAC(u1_x2,u1_t2,u1_n2,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
        ArcLength_Max_LAC,pu,0.5*pu_k1,increment_u1_x2,increment_u1_t2,
        increment_u1_n2);
    RK4_LAC(u1_x3,u1_t3,u1_n3,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
        ArcLength_Max_LAC,pu,0.5*pu_k1,increment_u1_x3,increment_u1_t3,
        increment_u1_n3);
    RK4_LASC(u2_x1,u2_t1,u2_n1,u2_b1,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
        Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,0.5*pu_k1,
        increment_u2_x1,increment_u2_t1,increment_u2_n1,increment_u2_b1);
    RK4_LASC(u2_x2,u2_t2,u2_n2,u2_b2,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
        Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,0.5*pu_k1,
        increment_u2_x2,increment_u2_t2,increment_u2_n2,increment_u2_b2);
    RK4_LASC(u2_x3,u2_t3,u2_n3,u2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
        Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,0.5*pu_k1,
        increment_u2_x3,increment_u2_t3,increment_u2_n3,increment_u2_b3);
    RK4_LAC(v1_x1,v1_t1,v1_n1,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
        ArcLength_Min_LAC,pv,0.5*pv_k1,increment_v1_x1,increment_v1_t1,
        increment_v1_n1);
    RK4_LAC(v1_x2,v1_t2,v1_n2,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
        ArcLength_Min_LAC,pv,0.5*pv_k1,increment_v1_x2,increment_v1_t2,
        increment_v1_n2);
    RK4_LAC(v1_x3,v1_t3,v1_n3,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
        ArcLength_Min_LAC,pv,0.5*pv_k1,increment_v1_x3,increment_v1_t3,
        increment_v1_n3);
    RK4_LASC(v2_x1,v2_t1,v2_n1,v2_b1,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
        Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,0.5*pv_k1,
        increment_v2_x1,increment_v2_t1,increment_v2_n1,increment_v2_b1);
    RK4_LASC(v2_x2,v2_t2,v2_n2,v2_b2,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
        Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,0.5*pv_k1,
        increment_v2_x2,increment_v2_t2,increment_v2_n2,increment_v2_b2);
    RK4_LASC(v2_x3,v2_t3,v2_n3,v2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
        Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,0.5*pv_k1,
        increment_v2_x3,increment_v2_t3,increment_v2_n3,increment_v2_b3);
```

```
new_pu=pu+0.5*pu_k1;
new_pv=pv+0.5*pv_k1;
new_u1_x1=u1_x1+increment_u1_x1;
new_u1_x2=u1_x2+increment_u1_x2;
new_u1_x3=u1_x3+increment_u1_x3;
new_u1_t1=u1_t1+increment_u1_t1;
new_u1_t2=u1_t2+increment_u1_t2;
new_u1_t3=u1_t3+increment_u1_t3;
new_u1_n1=u1_n1+increment_u1_n1;
new_u1_n2=u1_n2+increment_u1_n2;
new_u1_n3=u1_n3+increment_u1_n3;
new_u2_x1=u2_x1+increment_u2_x1;
new_u2_x2=u2_x2+increment_u2_x2;
new_u2_x3=u2_x3+increment_u2_x3;
new_u2_t1=u2_t1+increment_u2_t1;
new_u2_t2=u2_t2+increment_u2_t2;
new_u2_t3=u2_t3+increment_u2_t3;
new_u2_n1=u2_n1+increment_u2_n1;
new_u2_n2=u2_n2+increment_u2_n2;
new_u2_n3=u2_n3+increment_u2_n3;
new_v1_x1=v1_x1+increment_v1_x1;
new_v1_x2=v1_x2+increment_v1_x2;
new_v1_x3=v1_x3+increment_v1_x3;
new_v1_t1=v1_t1+increment_v1_t1;
new_v1_t2=v1_t2+increment_v1_t2;
new_v1_t3=v1_t3+increment_v1_t3;
new_v1_n1=v1_n1+increment_v1_n1;
new_v1_n2=v1_n2+increment_v1_n2;
new_v1_n3=v1_n3+increment_v1_n3;
new_v2_x1=v2_x1+increment_v2_x1;
new_v2_x2=v2_x2+increment_v2_x2;
new_v2_x3=v2_x3+increment_v2_x3;
new_v2_t1=v2_t1+increment_v2_t1;
new_v2_t2=v2_t2+increment_v2_t2;
new_v2_t3=v2_t3+increment_v2_t3;
new_v2_n1=v2_n1+increment_v2_n1;
new_v2_n2=v2_n2+increment_v2_n2;
new_v2_n3=v2_n3+increment_v2_n3;

First_Derivatives(new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u1_n1,new_u1_n2,new_u1
     _n3,new_pu,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC);
First_Derivatives(new_u2_dt1,new_u2_dt2,new_u2_dt3,new_u2_n1,new_u2_n2,
     new_u2_n3,new_pu,AlphaLAC,InitRCLAC,Lambda_Max_LASC,ArcLength_Max_LASC);
First_Derivatives(new_v1_dt1,new_v1_dt2,new_v1_dt3,new_v1_n1,new_v1_n2,
     new_v1_n3,new_pv,AlphaLAC,InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC);
First_Derivatives(new_v2_dt1,new_v2_dt2,new_v2_dt3,new_v2_n1,new_v2_n2,
     new_v2_n3,new_pv,AlphaLAC,InitRCLAC,Lambda_Min_LASC,ArcLength_Min_LASC);
```

```
pu_k2=stepsize*iniDuDs(new_pu,new_pv,new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,
     new_u1_t2,new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u2_x1,new_u2_x2
     ,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2,new_u2
     _dt3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,new_v1
     _dt1,new_v1_dt2,new_v1_dt3,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,
     new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,optimum);
pv_k2=stepsize*iniDvDs(new_pu,new_pv,new_u1_x1,new_u1_x2,new_u1_x3,
     new_u1_t1,new_u1_t2,new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u2_x1
     ,new_u2_x2,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2
     ,new_u2_dt3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,
     new_v1_dt1,new_v1_dt2,new_v1_dt3,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1
     ,new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,optimum);

RK4_LAC(u1_x1,u1_t1,u1_n1,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC,
     pu,0.5*pu_k2,increment_u1_x1,increment_u1_t1,increment_u1_n1);
RK4_LAC(u1_x2,u1_t2,u1_n2,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
     ArcLength_Max_LAC,pu,0.5*pu_k2,increment_u1_x2,increment_u1_t2,
     increment_u1_n2);
RK4_LAC(u1_x3,u1_t3,u1_n3,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
     ArcLength_Max_LAC,pu,0.5*pu_k2,increment_u1_x3,increment_u1_t3,
     increment_u1_n3);
RK4_LASC(u2_x1,u2_t1,u2_n1,u2_b1,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
     Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,0.5*pu_k2,
     increment_u2_x1,increment_u2_t1,increment_u2_n1,increment_u2_b1);
RK4_LASC(u2_x2,u2_t2,u2_n2,u2_b2,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
     Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,0.5*pu_k2,
     increment_u2_x2,increment_u2_t2,increment_u2_n2,increment_u2_b2);
RK4_LASC(u2_x3,u2_t3,u2_n3,u2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
     Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,0.5*pu_k2,
     increment_u2_x3,increment_u2_t3,increment_u2_n3,increment_u2_b3);
RK4_LAC(v1_x1,v1_t1,v1_n1,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
     ArcLength_Min_LAC,pv,0.5*pv_k2,increment_v1_x1,increment_v1_t1,
     increment_v1_n1);
RK4_LAC(v1_x2,v1_t2,v1_n2,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
     ArcLength_Min_LAC,pv,0.5*pv_k2,increment_v1_x2,increment_v1_t2,
     increment_v1_n2);
RK4_LAC(v1_x3,v1_t3,v1_n3,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
     ArcLength_Min_LAC,pv,0.5*pv_k2,increment_v1_x3,increment_v1_t3,
     increment_v1_n3);
RK4_LASC(v2_x1,v2_t1,v2_n1,v2_b1,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
     Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,0.5*pv_k2,
     increment_v2_x1,increment_v2_t1,increment_v2_n1,increment_v2_b1);
RK4_LASC(v2_x2,v2_t2,v2_n2,v2_b2,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
     Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,0.5*pv_k2,
     increment_v2_x2,increment_v2_t2,increment_v2_n2,increment_v2_b2);
```

```
RK4_LASC(v2_x3,v2_t3,v2_n3,v2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
     Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,0.5*pv_k2,
     increment_v2_x3,increment_v2_t3,increment_v2_n3,increment_v2_b3);

new_pu=pu+0.5*pu_k2;
new_pv=pv+0.5*pv_k2;
new_u1_x1=u1_x1+increment_u1_x1;
new_u1_x2=u1_x2+increment_u1_x2;
new_u1_x3=u1_x3+increment_u1_x3;
new_u1_t1=u1_t1+increment_u1_t1;
new_u1_t2=u1_t2+increment_u1_t2;
new_u1_t3=u1_t3+increment_u1_t3;
new_u1_n1=u1_n1+increment_u1_n1;
new_u1_n2=u1_n2+increment_u1_n2;
new_u1_n3=u1_n3+increment_u1_n3;
new_u2_x1=u2_x1+increment_u2_x1;
new_u2_x2=u2_x2+increment_u2_x2;
new_u2_x3=u2_x3+increment_u2_x3;
new_u2_t1=u2_t1+increment_u2_t1;
new_u2_t2=u2_t2+increment_u2_t2;
new_u2_t3=u2_t3+increment_u2_t3;
new_u2_n1=u2_n1+increment_u2_n1;
new_u2_n2=u2_n2+increment_u2_n2;
new_u2_n3=u2_n3+increment_u2_n3;
new_v1_x1=v1_x1+increment_v1_x1;
new_v1_x2=v1_x2+increment_v1_x2;
new_v1_x3=v1_x3+increment_v1_x3;
new_v1_t1=v1_t1+increment_v1_t1;
new_v1_t2=v1_t2+increment_v1_t2;
new_v1_t3=v1_t3+increment_v1_t3;
new_v1_n1=v1_n1+increment_v1_n1;
new_v1_n2=v1_n2+increment_v1_n2;
new_v1_n3=v1_n3+increment_v1_n3;
new_v2_x1=v2_x1+increment_v2_x1;
new_v2_x2=v2_x2+increment_v2_x2;
new_v2_x3=v2_x3+increment_v2_x3;
new_v2_t1=v2_t1+increment_v2_t1;
new_v2_t2=v2_t2+increment_v2_t2;
new_v2_t3=v2_t3+increment_v2_t3;
new_v2_n1=v2_n1+increment_v2_n1;
new_v2_n2=v2_n2+increment_v2_n2;
new_v2_n3=v2_n3+increment_v2_n3;

First_Derivatives(new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u1_n1,new_u1_n2,new_u1
    _n3,new_pu,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC);
First_Derivatives(new_u2_dt1,new_u2_dt2,new_u2_dt3,new_u2_n1,new_u2_n2,
     new_u2_n3,new_pu,AlphaLAC,InitRCLAC,Lambda_Max_LASC,ArcLength_Max_LASC);
```

```
First_Derivatives(new_v1_dt1,new_v1_dt2,new_v1_dt3,new_v1_n1,new_v1_n2,
    new_v1_n3,new_pv,AlphaLAC,InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC);
First_Derivatives(new_v2_dt1,new_v2_dt2,new_v2_dt3,new_v2_n1,new_v2_n2,
    new_v2_n3,new_pv,AlphaLAC,InitRCLAC,Lambda_Min_LASC,ArcLength_Min_LASC);


pu_k3=stepsize*iniDuDs(new_pu,new_pv,new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,
    new_u1_t2,new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u2_x1,new_u2_x2
    ,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2,new_u2
    _dt3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,new_v1
    _dt1,new_v1_dt2,new_v1_dt3,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,
    new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,optimum);
pv_k3=stepsize*iniDvDs(new_pu,new_pv,new_u1_x1,new_u1_x2,new_u1_x3,
    new_u1_t1,new_u1_t2,new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u2_x1
    ,new_u2_x2,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2
    ,new_u2_dt3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,
    new_v1_dt1,new_v1_dt2,new_v1_dt3,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1
    ,new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,optimum);

RK4_LAC(u1_x1,u1_t1,u1_n1,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC,
    pu,pu_k3,increment_u1_x1,increment_u1_t1,increment_u1_n1);
RK4_LAC(u1_x2,u1_t2,u1_n2,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
    ArcLength_Max_LAC,pu,pu_k3,increment_u1_x2,increment_u1_t2,increment_u1
    _n2);
RK4_LAC(u1_x3,u1_t3,u1_n3,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
    ArcLength_Max_LAC,pu,pu_k3,increment_u1_x3,increment_u1_t3,increment_u1
    _n3);
RK4_LASC(u2_x1,u2_t1,u2_n1,u2_b1,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,pu_k3,increment_u2
    _x1,increment_u2_t1,increment_u2_n1,increment_u2_b1);
RK4_LASC(u2_x2,u2_t2,u2_n2,u2_b2,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,pu_k3,increment_u2
    _x2,increment_u2_t2,increment_u2_n2,increment_u2_b2);
RK4_LASC(u2_x3,u2_t3,u2_n3,u2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,pu_k3,increment_u2
    _x3,increment_u2_t3,increment_u2_n3,increment_u2_b3);
RK4_LAC(v1_x1,v1_t1,v1_n1,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
    ArcLength_Min_LAC,pv,pv_k3,increment_v1_x1,increment_v1_t1,increment_v1
    _n1);
RK4_LAC(v1_x2,v1_t2,v1_n2,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
    ArcLength_Min_LAC,pv,pv_k3,increment_v1_x2,increment_v1_t2,increment_v1
    _n2);
RK4_LAC(v1_x3,v1_t3,v1_n3,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
    ArcLength_Min_LAC,pv,pv_k3,increment_v1_x3,increment_v1_t3,increment_v1
    _n3);
RK4_LASC(v2_x1,v2_t1,v2_n1,v2_b1,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,pv_k3,increment_v2
```

```
    _x1,increment_v2_t1,increment_v2_n1,increment_v2_b1);
RK4_LASC(v2_x2,v2_t2,v2_n2,v2_b2,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,pv_k3,increment_v2
    _x2,increment_v2_t2,increment_v2_n2,increment_v2_b2);
RK4_LASC(v2_x3,v2_t3,v2_n3,v2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,pv_k3,increment_v2
    _x3,increment_v2_t3,increment_v2_n3,increment_v2_b3);

new_pu=pu+pu_k3;
new_pv=pv+pv_k3;
new_u1_x1=u1_x1+increment_u1_x1;
new_u1_x2=u1_x2+increment_u1_x2;
new_u1_x3=u1_x3+increment_u1_x3;
new_u1_t1=u1_t1+increment_u1_t1;
new_u1_t2=u1_t2+increment_u1_t2;
new_u1_t3=u1_t3+increment_u1_t3;
new_u1_n1=u1_n1+increment_u1_n1;
new_u1_n2=u1_n2+increment_u1_n2;
new_u1_n3=u1_n3+increment_u1_n3;
new_u2_x1=u2_x1+increment_u2_x1;
new_u2_x2=u2_x2+increment_u2_x2;
new_u2_x3=u2_x3+increment_u2_x3;
new_u2_t1=u2_t1+increment_u2_t1;
new_u2_t2=u2_t2+increment_u2_t2;
new_u2_t3=u2_t3+increment_u2_t3;
new_u2_n1=u2_n1+increment_u2_n1;
new_u2_n2=u2_n2+increment_u2_n2;
new_u2_n3=u2_n3+increment_u2_n3;
new_v1_x1=v1_x1+increment_v1_x1;
new_v1_x2=v1_x2+increment_v1_x2;
new_v1_x3=v1_x3+increment_v1_x3;
new_v1_t1=v1_t1+increment_v1_t1;
new_v1_t2=v1_t2+increment_v1_t2;
new_v1_t3=v1_t3+increment_v1_t3;
new_v1_n1=v1_n1+increment_v1_n1;
new_v1_n2=v1_n2+increment_v1_n2;
new_v1_n3=v1_n3+increment_v1_n3;
new_v2_x1=v2_x1+increment_v2_x1;
new_v2_x2=v2_x2+increment_v2_x2;
new_v2_x3=v2_x3+increment_v2_x3;
new_v2_t1=v2_t1+increment_v2_t1;
new_v2_t2=v2_t2+increment_v2_t2;
new_v2_t3=v2_t3+increment_v2_t3;
new_v2_n1=v2_n1+increment_v2_n1;
new_v2_n2=v2_n2+increment_v2_n2;
new_v2_n3=v2_n3+increment_v2_n3;
```

```
First_Derivatives(new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u1_n1,new_u1_n2,new_u1
    _n3,new_pu,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC);
First_Derivatives(new_u2_dt1,new_u2_dt2,new_u2_dt3,new_u2_n1,new_u2_n2,
    new_u2_n3,new_pu,AlphaLAC,InitRCLAC,Lambda_Max_LASC,ArcLength_Max_LASC);
First_Derivatives(new_v1_dt1,new_v1_dt2,new_v1_dt3,new_v1_n1,new_v1_n2,
    new_v1_n3,new_pv,AlphaLAC,InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC);
First_Derivatives(new_v2_dt1,new_v2_dt2,new_v2_dt3,new_v2_n1,new_v2_n2,
    new_v2_n3,new_pv,AlphaLAC,InitRCLAC,Lambda_Min_LASC,ArcLength_Min_LASC);


pu_k4=stepsize*iniDuDs(new_pu,new_pv,new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,
    new_u1_t2,new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u2_x1,new_u2_x2
    ,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2,new_u2
    _dt3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,new_v1
    _dt1,new_v1_dt2,new_v1_dt3,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,
    new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,optimum);
pv_k4=stepsize*iniDvDs(new_pu,new_pv,new_u1_x1,new_u1_x2,new_u1_x3,
    new_u1_t1,new_u1_t2,new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u2_x1
    ,new_u2_x2,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2
    ,new_u2_dt3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,
    new_v1_dt1,new_v1_dt2,new_v1_dt3,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1
    ,new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,optimum);

increment_u=(1.0/6.0)*(pu_k1+2*pu_k2+2*pu_k3+pu_k4);
increment_v=(1.0/6.0)*(pv_k1+2*pv_k2+2*pv_k3+pv_k4);

}


__device__ void ReturnDetail( mint k, float
    x[], float y[], float z[], float GeoCur[], mint optimum)
{
float u1_x1=Point00x, u1_x2=Point00y, u1_x3=Point00z;
float u2_x1=Point30x, u2_x2=Point30y, u2_x3=Point30z;
float v1_x1=Point00x, v1_x2=Point00y, v1_x3=Point00z;
float v2_x1=Point03x, v2_x2=Point03y, v2_x3=Point03z;

float u1_t1=-1*ScaleToOrigin_Max_LAC*ArcLength_Max_LAC, u1_t2=0, u1_t3=0;
float u1_n1=0, u1_n2=0, u1_n3=1*ScaleToOrigin_Max_LAC*ArcLength_Max_LAC;
float v1_t1=0, v1_t2=-1*ScaleToOrigin_Min_LAC*ArcLength_Min_LAC, v1_t3=0;
float v1_n1=0, v1_n2=0, v1_n3=-1*ScaleToOrigin_Min_LAC*ArcLength_Min_LAC;

float u2_t1=-1*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC, u2_t2=0, u2_t3=0;
float u2_n1=0,
    u2_n2=-0.5930588126117428*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC,
    u2_n3=0.8051591425200051*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC;
float u2_b1=0,
```

```
    u2_b2=0.8051591425200051*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC,
    u2_b3=0.5930588126117428*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC;

float v2_t1=0, v2_t2=-1*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC, v2_t3=0;
float v2_n1=-0.5930588126117428*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC,
    v2_n2=0,
    v2_n3=-0.8051591425200051*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC;
float v2_b1=0.8051591425200051*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC,
    v2_b2=0,
    v2_b3=-0.5930588126117428*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC;

float pu=0.0, pv=0.0;
float SSize_u, SSize_v, ratio;

float u1_dt1, u1_dt2, u1_dt3;
float u1_d2t1, u1_d2t2, u1_d2t3;
float u1_d3t1, u1_d3t2, u1_d3t3;
float u2_dt1, u2_dt2, u2_dt3;
float u2_d2t1, u2_d2t2, u2_d2t3;
float u2_d3t1, u2_d3t2, u2_d3t3;

float v1_dt1, v1_dt2, v1_dt3;
float v1_d2t1, v1_d2t2, v1_d2t3;
float v1_d3t1, v1_d3t2, v1_d3t3;
float v2_dt1, v2_dt2, v2_dt3;
float v2_d2t1, v2_d2t2, v2_d2t3;
float v2_d3t1, v2_d3t2, v2_d3t3;
float coef[3];


if(optimum==min)
{
    for(mint i=1;i<k+1;i++)
    {

    FS_LAC(u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_n1,u1_n2,u1_n3,AlphaLAC,
    InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC,pu,h);
    //increase u by 0.001, then return the details

    FS_LASC(u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_n1,u2_n2,u2_n3,u2_b1,
    u2_b2,u2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,Lambda_Max_LASC,
    InitRT_Max_LASC,ArcLength_Max_LASC,pu,h);
    //increase u by 0.001, then return the details

    pu+=h;
    }
}
```

```
else if(optimum==max)
{
    for(mint i=1;i<k+1;i++)
    {

    FS_LAC(v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_n1,v1_n2,v1_n3,AlphaLAC,
    InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC,pv,h);
    //increase v by 0.001, then return the details

    FS_LASC(v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_n1,v2_n2,v2_n3,v2_b1,
    v2_b2,v2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,Lambda_Min_LASC,
    InitRT_Min_LASC,ArcLength_Min_LASC,pv,h);
    //increase v by 0.001, then return the details

    pv+=h;
    }
}

FS_2ndDerivatives_LAC(u1_dt1,u1_dt2,u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_t1,u1_t2,
    u1_t3,u1_n1,u1_n2,u1_n3,pu,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
    ArcLength_Max_LAC);
FS_2ndDerivatives_LASC(u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,
    u2_t1,u2_t2,u2_t3,u2_n1,u2_n2,u2_n3,u2_b1,u2_b2,u2_b3,pu,AlphaLAC,
    BetaLASC,InitRCLAC,OmegaLASC,Lambda_Max_LASC,InitRT_Max_LASC,
    ArcLength_Max_LASC);
FS_2ndDerivatives_LAC(v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v1_t1
    ,v1_t2,v1_t3,v1_n1,v1_n2,v1_n3,pv,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
    ArcLength_Min_LAC);
FS_2ndDerivatives_LASC(v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    v2_t1,v2_t2,v2_t3,v2_n1,v2_n2,v2_n3,v2_b1,v2_b2,v2_b3,pv,AlphaLAC,
    BetaLASC,InitRCLAC,OmegaLASC,Lambda_Min_LASC,InitRT_Min_LASC,
    ArcLength_Min_LASC);
LUDecomposition3x3(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,
    u1_dt2,u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,
    u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,
    v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,
    v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,
    v2_d2t3,optimum,coef);


x[0]=((1-pv)*u1_x1+pv*u2_x1+(1-pu)*v1_x1+pu*v2_x1-((1-pu)*(1-pv)*maxPoint00x+pu
    *(1-pv)*maxPoint03x+(1-pu)*pv*maxPoint30x+pu*pv*maxPoint33x));
y[0]=((1-pv)*u1_x2+pv*u2_x2+(1-pu)*v1_x2+pu*v2_x2-((1-pu)*(1-pv)*
    maxPoint00y+pu*(1-pv)*maxPoint03y+(1-pu)*pv*maxPoint30y+pu*pv*
    maxPoint33y));
z[0]=((1-pv)*u1_x3+pv*u2_x3+(1-pu)*v1_x3+pu*v2_x3-((1-pu)*(1-pv)*
    maxPoint00z+pu*(1-pv)*maxPoint03z+(1-pu)*pv*maxPoint30z+pu*pv*
```

```
        maxPoint33z));
    GeoCur[0]=coef[2];


    for(mint j=1;j<n;j++)
    {

        RK4_func_LoC(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_n1,u1_n2,u1_n3
        ,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_n1,u2_n2,u2_n3,u2_b1,u2_b2,u2_b3
        ,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_n1,v1_n2,v1_n3,v2_x1,v2_x2,v2_x3
        ,v2_t1,v2_t2,v2_t3,v2_n1,v2_n2,v2_n3,v2_b1,v2_b2,v2_b3,optimum,h,SSize_u
        ,SSize_v);
        //return increment of u and v from duds and dvds

        if((pu+SSize_u) > 1.0)
            {
            if(pu==1.0)
                {
                SSize_u=0.0;
                SSize_v=0.0;
                }
            else
                {
                ratio=(1.0-pu)/SSize_u;
                SSize_u*=ratio;
                SSize_v*=ratio;
                }
            }

        if((pv+SSize_v) > 1.0)
            {
            if(pv==1.0)
                {
                SSize_u=0.0;
                SSize_v=0.0;
                }
            else
                {
                ratio=(1.0-pv)/SSize_v;
                SSize_u*=ratio;
                SSize_v*=ratio;
                }
            }

        FS_LAC(u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_n1,u1_n2,u1_n3,AlphaLAC,
        InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC,pu,SSize_u);
        //based on the increment
```

```
of u as stepsize of LAC, then return the details
FS_LASC(u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_n1,u2_n2,u2_n3,u2_b1,
u2_b2,u2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,Lambda_Max_LASC,
InitRT_Max_LASC,ArcLength_Max_LASC,pu,SSize_u);
//based on the incremnt
of u as stepsize of LASC, then return the details

FS_LAC(v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_n1,v1_n2,v1_n3,AlphaLAC,
InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC,pv,SSize_v);
//based on the increment
of v as stepsize of LAC, then return the details
FS_LASC(v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_n1,v2_n2,v2_n3,v2_b1,
v2_b2,v2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,Lambda_Min_LASC,
InitRT_Min_LASC,ArcLength_Min_LASC,pv,SSize_v);
//based on the increment
of v as stepsize of LASC, then return the details

pu+=SSize_u;
pv+=SSize_v;


FS_2ndDerivatives_LAC(u1_dt1,u1_dt2,u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_t1
,u1_t2,u1_t3,u1_n1,u1_n2,u1_n3,pu,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
ArcLength_Max_LAC);

FS_2ndDerivatives_LASC(u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,
u2_t1,u2_t2,u2_t3,u2_n1,u2_n2,u2_n3,u2_b1,u2_b2,u2_b3,pu,AlphaLAC,
BetaLASC,InitRCLAC,OmegaLASC,Lambda_Max_LASC,InitRT_Max_LASC,
ArcLength_Max_LASC);

FS_2ndDerivatives_LAC(v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v1_t1
,v1_t2,v1_t3,v1_n1,v1_n2,v1_n3,pv,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
ArcLength_Min_LAC);

FS_2ndDerivatives_LASC(v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
v2_t1,v2_t2,v2_t3,v2_n1,v2_n2,v2_n3,v2_b1,v2_b2,v2_b3,pv,AlphaLAC,
BetaLASC,InitRCLAC,OmegaLASC,Lambda_Min_LASC,InitRT_Min_LASC,
ArcLength_Min_LASC);

LUDecomposition3x3(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,
u1_dt2,u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,
u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,
v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,
v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,
v2_d2t3,optimum,coef);
```

```
        x[j]=((1-pv)*u1_x1+pv*u2_x1+(1-pu)*v1_x1+pu*v2_x1-((1-pu)*(1-pv)*
        maxPoint00x+pu*(1-pv)*maxPoint03x+(1-pu)*pv*maxPoint30x+pu*pv*
        maxPoint33x));

        y[j]=((1-pv)*u1_x2+pv*u2_x2+(1-pu)*v1_x2+pu*v2_x2-((1-pu)*(1-pv)*
        maxPoint00y+pu*(1-pv)*maxPoint03y+(1-pu)*pv*maxPoint30y+pu*pv*
        maxPoint33y));

        z[j]=((1-pv)*u1_x3+pv*u2_x3+(1-pu)*v1_x3+pu*v2_x3-((1-pu)*(1-pv)*
        maxPoint00z+pu*(1-pv)*maxPoint03z+(1-pu)*pv*maxPoint30z+pu*pv*
        maxPoint33z));
        GeoCur[j]=coef[2];

    }

    }

    __device__ void ReturnXYZCoordinate(
        mint k, float x[], float y[], float z[], mint optimum)
    {
    float u1_x1=Point00x, u1_x2=Point00y, u1_x3=Point00z;
    float u2_x1=Point30x, u2_x2=Point30y, u2_x3=Point30z;
    float v1_x1=Point00x, v1_x2=Point00y, v1_x3=Point00z;
    float v2_x1=Point03x, v2_x2=Point03y, v2_x3=Point03z;

    float u1_t1=-1*ScaleToOrigin_Max_LAC*ArcLength_Max_LAC, u1_t2=0, u1_t3=0;
    float u1_n1=0, u1_n2=0, u1_n3=1*ScaleToOrigin_Max_LAC*ArcLength_Max_LAC;
    float v1_t1=0, v1_t2=-1*ScaleToOrigin_Min_LAC*ArcLength_Min_LAC, v1_t3=0;
    float v1_n1=0, v1_n2=0, v1_n3=-1*ScaleToOrigin_Min_LAC*ArcLength_Min_LAC;

    float u2_t1=-1*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC, u2_t2=0, u2_t3=0;
    float u2_n1=0,
        u2_n2=-0.5930588126117428*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC,
        u2_n3=0.8051591425200051*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC;
    float u2_b1=0,
        u2_b2=0.8051591425200051*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC,
        u2_b3=0.5930588126117428*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC;

    float v2_t1=0, v2_t2=-1*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC, v2_t3=0;
    float v2_n1=-0.5930588126117428*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC,
        v2_n2=0,
        v2_n3=-0.8051591425200051*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC;
    float v2_b1=0.8051591425200051*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC,
        v2_b2=0,
        v2_b3=-0.5930588126117428*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC;

    float pu=0.0, pv=0.0;
```

```
float SSize_u, SSize_v, ratio;

if(optimum==min)
{
    for(mint i=1;i<k+1;i++)
    {

    FS_LAC(u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_n1,u1_n2,u1_n3,AlphaLAC,
    InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC,pu,h);
    //increase u by 0.001, then return the details

    FS_LASC(u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_n1,u2_n2,u2_n3,u2_b1,
    u2_b2,u2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,Lambda_Max_LASC,
    InitRT_Max_LASC,ArcLength_Max_LASC,pu,h);
    //increase u by 0.001, then return the details

    pu+=h;
    }
}
else if(optimum==max)
{
    for(mint i=1;i<k+1;i++)
    {

    FS_LAC(v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_n1,v1_n2,v1_n3,AlphaLAC,
    InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC,pv,h);
    //increase v by 0.001, then return the details

    FS_LASC(v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_n1,v2_n2,v2_n3,v2_b1,
    v2_b2,v2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,Lambda_Min_LASC,
    InitRT_Min_LASC,ArcLength_Min_LASC,pv,h);
    //increase v by 0.001, then return the details

    pv+=h;
    }
}


x[0]=((1-pv)*u1_x1+pv*u2_x1+(1-pu)*v1_x1+pu*v2_x1-((1-pu)*(1-pv)*maxPoint00x+pu
    *(1-pv)*maxPoint03x+(1-pu)*pv*maxPoint30x+pu*pv*maxPoint33x));
y[0]=((1-pv)*u1_x2+pv*u2_x2+(1-pu)*v1_x2+pu*v2_x2-((1-pu)*(1-pv)*
    maxPoint00y+pu*(1-pv)*maxPoint03y+(1-pu)*pv*maxPoint30y+pu*pv*
    maxPoint33y));
z[0]=((1-pv)*u1_x3+pv*u2_x3+(1-pu)*v1_x3+pu*v2_x3-((1-pu)*(1-pv)*
    maxPoint00z+pu*(1-pv)*maxPoint03z+(1-pu)*pv*maxPoint30z+pu*pv*
    maxPoint33z));
```

```
for(mint j=1;j<n;j++)
{

    RK4_func_LoC(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_n1,u1_n2,u1_n3
    ,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_n1,u2_n2,u2_n3,u2_b1,u2_b2,u2_b3
    ,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_n1,v1_n2,v1_n3,v2_x1,v2_x2,v2_x3
    ,v2_t1,v2_t2,v2_t3,v2_n1,v2_n2,v2_n3,v2_b1,v2_b2,v2_b3,optimum,h,SSize_u
    ,SSize_v);
    //return increment of u and v from duds and dvds

    if((pu+SSize_u) > 1.0)
        {
        if(pu==1.0)
            {
            SSize_u=0.0;
            SSize_v=0.0;
            }
        else
            {
            ratio=(1.0-pu)/SSize_u;
            SSize_u*=ratio;
            SSize_v*=ratio;
            }
        }

    if((pv+SSize_v) > 1.0)
        {
        if(pv==1.0)
            {
            SSize_u=0.0;
            SSize_v=0.0;
            }
        else
            {
            ratio=(1.0-pv)/SSize_v;
            SSize_u*=ratio;
            SSize_v*=ratio;
            }
        }


    FS_LAC(u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_n1,u1_n2,u1_n3,AlphaLAC,
    InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC,pu,SSize_u);
    //based on the increment
    of u as stepsize of LAC, then return the details
```

```
    FS_LASC(u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_n1,u2_n2,u2_n3,u2_b1,
    u2_b2,u2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,Lambda_Max_LASC,
    InitRT_Max_LASC,ArcLength_Max_LASC,pu,SSize_u);
    //based on the incremnt
    of u as stepsize of LASC, then return the details

    FS_LAC(v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_n1,v1_n2,v1_n3,AlphaLAC,
    InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC,pv,SSize_v);
    //based on the increment
    of v as stepsize of LAC, then return the details
    FS_LASC(v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_n1,v2_n2,v2_n3,v2_b1,
    v2_b2,v2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,Lambda_Min_LASC,
    InitRT_Min_LASC,ArcLength_Min_LASC,pv,SSize_v);
    //based on the increment
    of v as stepsize of LASC, then return the details

    pu+=SSize_u;
    pv+=SSize_v;

    x[j]=((1-pv)*u1_x1+pv*u2_x1+(1-pu)*v1_x1+pu*v2_x1-((1-pu)*(1-pv)*
    maxPoint00x+pu*(1-pv)*maxPoint03x+(1-pu)*pv*maxPoint30x+pu*pv*
    maxPoint33x));

    y[j]=((1-pv)*u1_x2+pv*u2_x2+(1-pu)*v1_x2+pu*v2_x2-((1-pu)*(1-pv)*
    maxPoint00y+pu*(1-pv)*maxPoint03y+(1-pu)*pv*maxPoint30y+pu*pv*
    maxPoint33y));

    z[j]=((1-pv)*u1_x3+pv*u2_x3+(1-pu)*v1_x3+pu*v2_x3-((1-pu)*(1-pv)*
    maxPoint00z+pu*(1-pv)*maxPoint03z+(1-pu)*pv*maxPoint30z+pu*pv*
    maxPoint33z));
}

}


__device__ float DuDsforBoundaries opt4
{
if(optimum == max)
    {return 1.0/(sqrt(repinicoefE));}
else if(optimum == min)
    {return 0.0;}
}
__device__ float DvDsforBoundaries opt4
{
if(optimum == max)
    {return 0.0;}
else if(optimum == min)
```

```
    {return 1.0/(sqrt(repinicoefG));}
}

__device__ float GeoCurforBoundaries opt4
{
if(optimum == max)
    {return
    ((2.0*repinicoefE*repiniDFDu-repinicoefE*repiniDEDv-repinicoefF*
    repiniDEDu)*pow(DuDsforBoundaries(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,
    u1_t3,u1_dt1,u1_dt2,u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum),3))/(
    2.0*sqrt(repinicoefE*repinicoefG-pow(repinicoefF,2)));}
else if(optimum == min)
    {return
    (-(2.0*repinicoefG*repiniDFDv-repinicoefG*repiniDGDu-repinicoefF*
    repiniDGDv)*pow(DvDsforBoundaries(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,
    u1_t3,u1_dt1,u1_dt2,u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum),3))/(
    2.0*sqrt(repinicoefE*repinicoefG-pow(repinicoefF,2)));}
}

__device__ void RK4_func_Boundaries( float pu, float pv, float u1_x1, float
    u1_x2, float u1_x3, float u1_t1, float u1_t2, float u1_t3, float u1_n1,
    float u1_n2, float u1_n3, float u2_x1, float u2_x2, float u2_x3,
    float u2_t1, float u2_t2, float u2_t3, float u2_n1, float u2_n2,
    float u2_n3, float u2_b1, float u2_b2, float u2_b3, float v1_x1,
    float v1_x2, float v1_x3, float v1_t1, float v1_t2, float v1_t3,
    float v1_n1, float v1_n2, float v1_n3, float v2_x1, float v2_x2,
    float v2_x3, float v2_t1, float v2_t2, float v2_t3, float v2_n1,
    float v2_n2, float v2_n3, float v2_b1, float v2_b2, float v2_b3, mint
    optimum, float stepsize, float& increment_u, float& increment_v)
{
float u1_dt1, u1_dt2, u1_dt3, u2_dt1, u2_dt2,
    u2_dt3, v1_dt1, v1_dt2, v1_dt3, v2_dt1, v2_dt2, v2_dt3;
float pu_k1,pu_k2,pu_k3,pu_k4;
float pv_k1,pv_k2,pv_k3,pv_k4;
float increment_u1_x1, increment_u1_x2, increment_u1_x3,
    increment_u1_t1, increment_u1_t2, increment_u1_t3,
    increment_u1_n1, increment_u1_n2, increment_u1_n3;
float increment_u2_x1, increment_u2_x2, increment_u2_x3, increment_u2_t1,
    increment_u2_t2, increment_u2_t3, increment_u2_n1, increment_u2_n2,
    increment_u2_n3, increment_u2_b1, increment_u2_b2, increment_u2_b3;
float increment_v1_x1, increment_v1_x2, increment_v1_x3,
    increment_v1_t1, increment_v1_t2, increment_v1_t3,
    increment_v1_n1, increment_v1_n2, increment_v1_n3;
```

```
float increment_v2_x1, increment_v2_x2, increment_v2_x3, increment_v2_t1,
    increment_v2_t2, increment_v2_t3, increment_v2_n1, increment_v2_n2,
    increment_v2_n3, increment_v2_b1, increment_v2_b2, increment_v2_b3;
float new_pu, new_pv;
float
    new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,new_u1_t2,new_u1_t3,new_u1_dt1,
    new_u1_dt2,new_u1_dt3,new_u1_n1,new_u1_n2,new_u1_n3;
float
    new_u2_x1,new_u2_x2,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,
    new_u2_dt2,new_u2_dt3,new_u2_n1,new_u2_n2,new_u2_n3;
float
    new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,new_v1_dt1,
    new_v1_dt2,new_v1_dt3,new_v1_n1,new_v1_n2,new_v1_n3;
float
    new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,new_v2_t2,new_v2_t3,new_v2_dt1,
    new_v2_dt2,new_v2_dt3,new_v2_n1,new_v2_n2,new_v2_n3;


First_Derivatives(u1_dt1,u1_dt2,u1_dt3,u1_n1,u1_n2,u1_n3,pu,AlphaLAC,InitRCLAC,
    Lambda_Max_LAC,ArcLength_Max_LAC);
First_Derivatives(u2_dt1,u2_dt2,u2_dt3,u2_n1,u2_n2,u2_n3,pu,AlphaLAC,
    InitRCLAC,Lambda_Max_LASC,ArcLength_Max_LASC);
First_Derivatives(v1_dt1,v1_dt2,v1_dt3,v1_n1,v1_n2,v1_n3,pv,AlphaLAC,
    InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC);
First_Derivatives(v2_dt1,v2_dt2,v2_dt3,v2_n1,v2_n2,v2_n3,pv,AlphaLAC,
    InitRCLAC,Lambda_Min_LASC,ArcLength_Min_LASC);

pu_k1=stepsize*DuDsforBoundaries(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,
    u1_dt1,u1_dt2,u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,
    u2_dt3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,
    v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum);
pv_k1=stepsize*DvDsforBoundaries(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,
    u1_t3,u1_dt1,u1_dt2,u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum);

RK4_LAC(u1_x1,u1_t1,u1_n1,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC,
    pu,0.5*pu_k1,increment_u1_x1,increment_u1_t1,increment_u1_n1);
RK4_LAC(u1_x2,u1_t2,u1_n2,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
    ArcLength_Max_LAC,pu,0.5*pu_k1,increment_u1_x2,increment_u1_t2,
    increment_u1_n2);
RK4_LAC(u1_x3,u1_t3,u1_n3,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
    ArcLength_Max_LAC,pu,0.5*pu_k1,increment_u1_x3,increment_u1_t3,
    increment_u1_n3);
RK4_LASC(u2_x1,u2_t1,u2_n1,u2_b1,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,0.5*pu_k1,
    increment_u2_x1,increment_u2_t1,increment_u2_n1,increment_u2_b1);
```

```
RK4_LASC(u2_x2,u2_t2,u2_n2,u2_b2,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
     Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,0.5*pu_k1,
     increment_u2_x2,increment_u2_t2,increment_u2_n2,increment_u2_b2);
RK4_LASC(u2_x3,u2_t3,u2_n3,u2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
     Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,0.5*pu_k1,
     increment_u2_x3,increment_u2_t3,increment_u2_n3,increment_u2_b3);
RK4_LAC(v1_x1,v1_t1,v1_n1,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
     ArcLength_Min_LAC,pv,0.5*pv_k1,increment_v1_x1,increment_v1_t1,
     increment_v1_n1);
RK4_LAC(v1_x2,v1_t2,v1_n2,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
     ArcLength_Min_LAC,pv,0.5*pv_k1,increment_v1_x2,increment_v1_t2,
     increment_v1_n2);
RK4_LAC(v1_x3,v1_t3,v1_n3,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
     ArcLength_Min_LAC,pv,0.5*pv_k1,increment_v1_x3,increment_v1_t3,
     increment_v1_n3);
RK4_LASC(v2_x1,v2_t1,v2_n1,v2_b1,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
     Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,0.5*pv_k1,
     increment_v2_x1,increment_v2_t1,increment_v2_n1,increment_v2_b1);
RK4_LASC(v2_x2,v2_t2,v2_n2,v2_b2,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
     Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,0.5*pv_k1,
     increment_v2_x2,increment_v2_t2,increment_v2_n2,increment_v2_b2);
RK4_LASC(v2_x3,v2_t3,v2_n3,v2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
     Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,0.5*pv_k1,
     increment_v2_x3,increment_v2_t3,increment_v2_n3,increment_v2_b3);

new_pu=pu+0.5*pu_k1;
new_pv=pv+0.5*pv_k1;
new_u1_x1=u1_x1+increment_u1_x1;
new_u1_x2=u1_x2+increment_u1_x2;
new_u1_x3=u1_x3+increment_u1_x3;
new_u1_t1=u1_t1+increment_u1_t1;
new_u1_t2=u1_t2+increment_u1_t2;
new_u1_t3=u1_t3+increment_u1_t3;
new_u1_n1=u1_n1+increment_u1_n1;
new_u1_n2=u1_n2+increment_u1_n2;
new_u1_n3=u1_n3+increment_u1_n3;
new_u2_x1=u2_x1+increment_u2_x1;
new_u2_x2=u2_x2+increment_u2_x2;
new_u2_x3=u2_x3+increment_u2_x3;
new_u2_t1=u2_t1+increment_u2_t1;
new_u2_t2=u2_t2+increment_u2_t2;
new_u2_t3=u2_t3+increment_u2_t3;
new_u2_n1=u2_n1+increment_u2_n1;
new_u2_n2=u2_n2+increment_u2_n2;
new_u2_n3=u2_n3+increment_u2_n3;
new_v1_x1=v1_x1+increment_v1_x1;
new_v1_x2=v1_x2+increment_v1_x2;
```

```
new_v1_x3=v1_x3+increment_v1_x3;
new_v1_t1=v1_t1+increment_v1_t1;
new_v1_t2=v1_t2+increment_v1_t2;
new_v1_t3=v1_t3+increment_v1_t3;
new_v1_n1=v1_n1+increment_v1_n1;
new_v1_n2=v1_n2+increment_v1_n2;
new_v1_n3=v1_n3+increment_v1_n3;
new_v2_x1=v2_x1+increment_v2_x1;
new_v2_x2=v2_x2+increment_v2_x2;
new_v2_x3=v2_x3+increment_v2_x3;
new_v2_t1=v2_t1+increment_v2_t1;
new_v2_t2=v2_t2+increment_v2_t2;
new_v2_t3=v2_t3+increment_v2_t3;
new_v2_n1=v2_n1+increment_v2_n1;
new_v2_n2=v2_n2+increment_v2_n2;
new_v2_n3=v2_n3+increment_v2_n3;

First_Derivatives(new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u1_n1,new_u1_n2,new_u1
    _n3,new_pu,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC);
First_Derivatives(new_u2_dt1,new_u2_dt2,new_u2_dt3,new_u2_n1,new_u2_n2,
    new_u2_n3,new_pu,AlphaLAC,InitRCLAC,Lambda_Max_LASC,ArcLength_Max_LASC);
First_Derivatives(new_v1_dt1,new_v1_dt2,new_v1_dt3,new_v1_n1,new_v1_n2,
    new_v1_n3,new_pv,AlphaLAC,InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC);
First_Derivatives(new_v2_dt1,new_v2_dt2,new_v2_dt3,new_v2_n1,new_v2_n2,
    new_v2_n3,new_pv,AlphaLAC,InitRCLAC,Lambda_Min_LASC,ArcLength_Min_LASC);


pu_k2=stepsize*DuDsforBoundaries(new_pu,new_pv,new_u1_x1,new_u1_x2,new_u1_x3,
    new_u1_t1,new_u1_t2,new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u2_x1
    ,new_u2_x2,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2
    ,new_u2_dt3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,
    new_v1_dt1,new_v1_dt2,new_v1_dt3,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1
    ,new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,optimum);
pv_k2=stepsize*DvDsforBoundaries(new_pu,new_pv,new_u1_x1,new_u1_x2,
    new_u1_x3,new_u1_t1,new_u1_t2,new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3
    ,new_u2_x1,new_u2_x2,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,
    new_u2_dt2,new_u2_dt3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,
    new_v1_t3,new_v1_dt1,new_v1_dt2,new_v1_dt3,new_v2_x1,new_v2_x2,new_v2_x3
    ,new_v2_t1,new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,optimum)
    ;

RK4_LAC(u1_x1,u1_t1,u1_n1,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC,
    pu,0.5*pu_k2,increment_u1_x1,increment_u1_t1,increment_u1_n1);
RK4_LAC(u1_x2,u1_t2,u1_n2,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
    ArcLength_Max_LAC,pu,0.5*pu_k2,increment_u1_x2,increment_u1_t2,
    increment_u1_n2);
RK4_LAC(u1_x3,u1_t3,u1_n3,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
```

```
    ArcLength_Max_LAC,pu,0.5*pu_k2,increment_u1_x3,increment_u1_t3,
    increment_u1_n3);
RK4_LASC(u2_x1,u2_t1,u2_n1,u2_b1,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,0.5*pu_k2,
    increment_u2_x1,increment_u2_t1,increment_u2_n1,increment_u2_b1);
RK4_LASC(u2_x2,u2_t2,u2_n2,u2_b2,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,0.5*pu_k2,
    increment_u2_x2,increment_u2_t2,increment_u2_n2,increment_u2_b2);
RK4_LASC(u2_x3,u2_t3,u2_n3,u2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,0.5*pu_k2,
    increment_u2_x3,increment_u2_t3,increment_u2_n3,increment_u2_b3);
RK4_LAC(v1_x1,v1_t1,v1_n1,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
    ArcLength_Min_LAC,pv,0.5*pv_k2,increment_v1_x1,increment_v1_t1,
    increment_v1_n1);
RK4_LAC(v1_x2,v1_t2,v1_n2,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
    ArcLength_Min_LAC,pv,0.5*pv_k2,increment_v1_x2,increment_v1_t2,
    increment_v1_n2);
RK4_LAC(v1_x3,v1_t3,v1_n3,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
    ArcLength_Min_LAC,pv,0.5*pv_k2,increment_v1_x3,increment_v1_t3,
    increment_v1_n3);
RK4_LASC(v2_x1,v2_t1,v2_n1,v2_b1,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,0.5*pv_k2,
    increment_v2_x1,increment_v2_t1,increment_v2_n1,increment_v2_b1);
RK4_LASC(v2_x2,v2_t2,v2_n2,v2_b2,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,0.5*pv_k2,
    increment_v2_x2,increment_v2_t2,increment_v2_n2,increment_v2_b2);
RK4_LASC(v2_x3,v2_t3,v2_n3,v2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,0.5*pv_k2,
    increment_v2_x3,increment_v2_t3,increment_v2_n3,increment_v2_b3);

new_pu=pu+0.5*pu_k2;
new_pv=pv+0.5*pv_k2;
new_u1_x1=u1_x1+increment_u1_x1;
new_u1_x2=u1_x2+increment_u1_x2;
new_u1_x3=u1_x3+increment_u1_x3;
new_u1_t1=u1_t1+increment_u1_t1;
new_u1_t2=u1_t2+increment_u1_t2;
new_u1_t3=u1_t3+increment_u1_t3;
new_u1_n1=u1_n1+increment_u1_n1;
new_u1_n2=u1_n2+increment_u1_n2;
new_u1_n3=u1_n3+increment_u1_n3;
new_u2_x1=u2_x1+increment_u2_x1;
new_u2_x2=u2_x2+increment_u2_x2;
new_u2_x3=u2_x3+increment_u2_x3;
new_u2_t1=u2_t1+increment_u2_t1;
new_u2_t2=u2_t2+increment_u2_t2;
new_u2_t3=u2_t3+increment_u2_t3;
```

```
new_u2_n1=u2_n1+increment_u2_n1;
new_u2_n2=u2_n2+increment_u2_n2;
new_u2_n3=u2_n3+increment_u2_n3;
new_v1_x1=v1_x1+increment_v1_x1;
new_v1_x2=v1_x2+increment_v1_x2;
new_v1_x3=v1_x3+increment_v1_x3;
new_v1_t1=v1_t1+increment_v1_t1;
new_v1_t2=v1_t2+increment_v1_t2;
new_v1_t3=v1_t3+increment_v1_t3;
new_v1_n1=v1_n1+increment_v1_n1;
new_v1_n2=v1_n2+increment_v1_n2;
new_v1_n3=v1_n3+increment_v1_n3;
new_v2_x1=v2_x1+increment_v2_x1;
new_v2_x2=v2_x2+increment_v2_x2;
new_v2_x3=v2_x3+increment_v2_x3;
new_v2_t1=v2_t1+increment_v2_t1;
new_v2_t2=v2_t2+increment_v2_t2;
new_v2_t3=v2_t3+increment_v2_t3;
new_v2_n1=v2_n1+increment_v2_n1;
new_v2_n2=v2_n2+increment_v2_n2;
new_v2_n3=v2_n3+increment_v2_n3;

First_Derivatives(new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u1_n1,new_u1_n2,new_u1
    _n3,new_pu,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC);
First_Derivatives(new_u2_dt1,new_u2_dt2,new_u2_dt3,new_u2_n1,new_u2_n2,
    new_u2_n3,new_pu,AlphaLAC,InitRCLAC,Lambda_Max_LASC,ArcLength_Max_LASC);
First_Derivatives(new_v1_dt1,new_v1_dt2,new_v1_dt3,new_v1_n1,new_v1_n2,
    new_v1_n3,new_pv,AlphaLAC,InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC);
First_Derivatives(new_v2_dt1,new_v2_dt2,new_v2_dt3,new_v2_n1,new_v2_n2,
    new_v2_n3,new_pv,AlphaLAC,InitRCLAC,Lambda_Min_LASC,ArcLength_Min_LASC);


pu_k3=stepsize*DuDsforBoundaries(new_pu,new_pv,new_u1_x1,new_u1_x2,new_u1_x3,
    new_u1_t1,new_u1_t2,new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u2_x1
    ,new_u2_x2,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2
    ,new_u2_dt3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,
    new_v1_dt1,new_v1_dt2,new_v1_dt3,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1
    ,new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,optimum);
pv_k3=stepsize*DvDsforBoundaries(new_pu,new_pv,new_u1_x1,new_u1_x2,
    new_u1_x3,new_u1_t1,new_u1_t2,new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3
    ,new_u2_x1,new_u2_x2,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,
    new_u2_dt2,new_u2_dt3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,
    new_v1_t3,new_v1_dt1,new_v1_dt2,new_v1_dt3,new_v2_x1,new_v2_x2,new_v2_x3
    ,new_v2_t1,new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,optimum)
    ;

RK4_LAC(u1_x1,u1_t1,u1_n1,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC,
```

```
        pu,pu_k3,increment_u1_x1,increment_u1_t1,increment_u1_n1);
    RK4_LAC(u1_x2,u1_t2,u1_n2,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
        ArcLength_Max_LAC,pu,pu_k3,increment_u1_x2,increment_u1_t2,increment_u1
        _n2);
    RK4_LAC(u1_x3,u1_t3,u1_n3,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
        ArcLength_Max_LAC,pu,pu_k3,increment_u1_x3,increment_u1_t3,increment_u1
        _n3);
    RK4_LASC(u2_x1,u2_t1,u2_n1,u2_b1,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
        Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,pu_k3,increment_u2
        _x1,increment_u2_t1,increment_u2_n1,increment_u2_b1);
    RK4_LASC(u2_x2,u2_t2,u2_n2,u2_b2,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
        Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,pu_k3,increment_u2
        _x2,increment_u2_t2,increment_u2_n2,increment_u2_b2);
    RK4_LASC(u2_x3,u2_t3,u2_n3,u2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
        Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,pu,pu_k3,increment_u2
        _x3,increment_u2_t3,increment_u2_n3,increment_u2_b3);
    RK4_LAC(v1_x1,v1_t1,v1_n1,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
        ArcLength_Min_LAC,pv,pv_k3,increment_v1_x1,increment_v1_t1,increment_v1
        _n1);
    RK4_LAC(v1_x2,v1_t2,v1_n2,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
        ArcLength_Min_LAC,pv,pv_k3,increment_v1_x2,increment_v1_t2,increment_v1
        _n2);
    RK4_LAC(v1_x3,v1_t3,v1_n3,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
        ArcLength_Min_LAC,pv,pv_k3,increment_v1_x3,increment_v1_t3,increment_v1
        _n3);
    RK4_LASC(v2_x1,v2_t1,v2_n1,v2_b1,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
        Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,pv_k3,increment_v2
        _x1,increment_v2_t1,increment_v2_n1,increment_v2_b1);
    RK4_LASC(v2_x2,v2_t2,v2_n2,v2_b2,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
        Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,pv_k3,increment_v2
        _x2,increment_v2_t2,increment_v2_n2,increment_v2_b2);
    RK4_LASC(v2_x3,v2_t3,v2_n3,v2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
        Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,pv,pv_k3,increment_v2
        _x3,increment_v2_t3,increment_v2_n3,increment_v2_b3);

    new_pu=pu+pu_k3;
    new_pv=pv+pv_k3;
    new_u1_x1=u1_x1+increment_u1_x1;
    new_u1_x2=u1_x2+increment_u1_x2;
    new_u1_x3=u1_x3+increment_u1_x3;
    new_u1_t1=u1_t1+increment_u1_t1;
    new_u1_t2=u1_t2+increment_u1_t2;
    new_u1_t3=u1_t3+increment_u1_t3;
    new_u1_n1=u1_n1+increment_u1_n1;
    new_u1_n2=u1_n2+increment_u1_n2;
    new_u1_n3=u1_n3+increment_u1_n3;
    new_u2_x1=u2_x1+increment_u2_x1;
```

```
new_u2_x2=u2_x2+increment_u2_x2;
new_u2_x3=u2_x3+increment_u2_x3;
new_u2_t1=u2_t1+increment_u2_t1;
new_u2_t2=u2_t2+increment_u2_t2;
new_u2_t3=u2_t3+increment_u2_t3;
new_u2_n1=u2_n1+increment_u2_n1;
new_u2_n2=u2_n2+increment_u2_n2;
new_u2_n3=u2_n3+increment_u2_n3;
new_v1_x1=v1_x1+increment_v1_x1;
new_v1_x2=v1_x2+increment_v1_x2;
new_v1_x3=v1_x3+increment_v1_x3;
new_v1_t1=v1_t1+increment_v1_t1;
new_v1_t2=v1_t2+increment_v1_t2;
new_v1_t3=v1_t3+increment_v1_t3;
new_v1_n1=v1_n1+increment_v1_n1;
new_v1_n2=v1_n2+increment_v1_n2;
new_v1_n3=v1_n3+increment_v1_n3;
new_v2_x1=v2_x1+increment_v2_x1;
new_v2_x2=v2_x2+increment_v2_x2;
new_v2_x3=v2_x3+increment_v2_x3;
new_v2_t1=v2_t1+increment_v2_t1;
new_v2_t2=v2_t2+increment_v2_t2;
new_v2_t3=v2_t3+increment_v2_t3;
new_v2_n1=v2_n1+increment_v2_n1;
new_v2_n2=v2_n2+increment_v2_n2;
new_v2_n3=v2_n3+increment_v2_n3;

First_Derivatives(new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u1_n1,new_u1_n2,new_u1
    _n3,new_pu,AlphaLAC,InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC);
First_Derivatives(new_u2_dt1,new_u2_dt2,new_u2_dt3,new_u2_n1,new_u2_n2,
    new_u2_n3,new_pu,AlphaLAC,InitRCLAC,Lambda_Max_LASC,ArcLength_Max_LASC);
First_Derivatives(new_v1_dt1,new_v1_dt2,new_v1_dt3,new_v1_n1,new_v1_n2,
    new_v1_n3,new_pv,AlphaLAC,InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC);
First_Derivatives(new_v2_dt1,new_v2_dt2,new_v2_dt3,new_v2_n1,new_v2_n2,
    new_v2_n3,new_pv,AlphaLAC,InitRCLAC,Lambda_Min_LASC,ArcLength_Min_LASC);


pu_k4=stepsize*DuDsforBoundaries(new_pu,new_pv,new_u1_x1,new_u1_x2,new_u1_x3,
    new_u1_t1,new_u1_t2,new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u2_x1
    ,new_u2_x2,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2
    ,new_u2_dt3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,
    new_v1_dt1,new_v1_dt2,new_v1_dt3,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1
    ,new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,optimum);
pv_k4=stepsize*DvDsforBoundaries(new_pu,new_pv,new_u1_x1,new_u1_x2,
    new_u1_x3,new_u1_t1,new_u1_t2,new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3
    ,new_u2_x1,new_u2_x2,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,
    new_u2_dt2,new_u2_dt3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,
```

```
    new_v1_t3,new_v1_dt1,new_v1_dt2,new_v1_dt3,new_v2_x1,new_v2_x2,new_v2_x3
    ,new_v2_t1,new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,optimum)
    ;

increment_u=(1.0/6.0)*(pu_k1+2*pu_k2+2*pu_k3+pu_k4);
increment_v=(1.0/6.0)*(pv_k1+2*pv_k2+2*pv_k3+pv_k4);

}

__device__ void ReturnDetailforBoundaries( mint k, float
    x[], float y[], float z[], float GeoCur[], mint optimum)
{
float u1_x1=Point00x, u1_x2=Point00y, u1_x3=Point00z;
float u2_x1=Point30x, u2_x2=Point30y, u2_x3=Point30z;
float v1_x1=Point00x, v1_x2=Point00y, v1_x3=Point00z;
float v2_x1=Point03x, v2_x2=Point03y, v2_x3=Point03z;

float u1_t1=-1*ScaleToOrigin_Max_LAC*ArcLength_Max_LAC, u1_t2=0, u1_t3=0;
float u1_n1=0, u1_n2=0, u1_n3=1*ScaleToOrigin_Max_LAC*ArcLength_Max_LAC;
float v1_t1=0, v1_t2=-1*ScaleToOrigin_Min_LAC*ArcLength_Min_LAC, v1_t3=0;
float v1_n1=0, v1_n2=0, v1_n3=-1*ScaleToOrigin_Min_LAC*ArcLength_Min_LAC;

float u2_t1=-1*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC, u2_t2=0, u2_t3=0;
float u2_n1=0,
    u2_n2=-0.5930588126117428*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC,
    u2_n3=0.8051591425200051*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC;
float u2_b1=0,
    u2_b2=0.8051591425200051*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC,
    u2_b3=0.5930588126117428*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC;

float v2_t1=0, v2_t2=-1*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC, v2_t3=0;
float v2_n1=-0.5930588126117428*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC,
    v2_n2=0,
    v2_n3=-0.8051591425200051*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC;
float v2_b1=0.8051591425200051*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC,
    v2_b2=0,
    v2_b3=-0.5930588126117428*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC;

float u1_dt1,u1_dt2,u1_dt3;
float u2_dt1,u2_dt2,u2_dt3;
float v1_dt1,v1_dt2,v1_dt3;
float v2_dt1,v2_dt2,v2_dt3;

float pu=0.0, pv=0.0;
float SSize_u, SSize_v, ratio;
```

```
if(optimum==min)
{
     for(mint i=1;i<k+1;i++)
     {

     FS_LAC(u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_n1,u1_n2,u1_n3,AlphaLAC,
     InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC,pu,h);
     //increase u by 0.001, then return the details

     FS_LASC(u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_n1,u2_n2,u2_n3,u2_b1,
     u2_b2,u2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,Lambda_Max_LASC,
     InitRT_Max_LASC,ArcLength_Max_LASC,pu,h);
     //increase u by 0.001, then return the details

     pu+=h;
     }
}
else if(optimum==max)
{
     for(mint i=1;i<k+1;i++)
     {

     FS_LAC(v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_n1,v1_n2,v1_n3,AlphaLAC,
     InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC,pv,h);
     //increase v by 0.001, then return the details

     FS_LASC(v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_n1,v2_n2,v2_n3,v2_b1,
     v2_b2,v2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,Lambda_Min_LASC,
     InitRT_Min_LASC,ArcLength_Min_LASC,pv,h);
     //increase v by 0.001, then return the details

     pv+=h;
     }
}
First_Derivatives(u1_dt1,u1_dt2,u1_dt3,u1_n1,u1_n2,u1_n3,pu,AlphaLAC,InitRCLAC,
     Lambda_Max_LAC,ArcLength_Max_LAC);
First_Derivatives(u2_dt1,u2_dt2,u2_dt3,u2_n1,u2_n2,u2_n3,pu,AlphaLAC,
     InitRCLAC,Lambda_Max_LASC,ArcLength_Max_LASC);
First_Derivatives(v1_dt1,v1_dt2,v1_dt3,v1_n1,v1_n2,v1_n3,pv,AlphaLAC,
     InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC);
First_Derivatives(v2_dt1,v2_dt2,v2_dt3,v2_n1,v2_n2,v2_n3,pv,AlphaLAC,
     InitRCLAC,Lambda_Min_LASC,ArcLength_Min_LASC);


x[0]=((1-pv)*u1_x1+pv*u2_x1+(1-pu)*v1_x1+pu*v2_x1-((1-pu)*(1-pv)*maxPoint00x+pu
     *(1-pv)*maxPoint03x+(1-pu)*pv*maxPoint30x+pu*pv*maxPoint33x));
y[0]=((1-pv)*u1_x2+pv*u2_x2+(1-pu)*v1_x2+pu*v2_x2-((1-pu)*(1-pv)*
```

```
    maxPoint00y+pu*(1-pv)*maxPoint03y+(1-pu)*pv*maxPoint30y+pu*pv*
    maxPoint33y));
z[0]=((1-pv)*u1_x3+pv*u2_x3+(1-pu)*v1_x3+pu*v2_x3-((1-pu)*(1-pv)*
    maxPoint00z+pu*(1-pv)*maxPoint03z+(1-pu)*pv*maxPoint30z+pu*pv*
    maxPoint33z));
GeoCur[0]=GeoCurforBoundaries(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,
    u1_dt1,u1_dt2,u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,
    u2_dt3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,
    v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum);

for(mint j=1;j<n;j++)
{
    RK4_func_Boundaries(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_n1,
    u1_n2,u1_n3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_n1,u2_n2,u2_n3,u2_b1,
    u2_b2,u2_b3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_n1,v1_n2,v1_n3,v2_x1,
    v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_n1,v2_n2,v2_n3,v2_b1,v2_b2,v2_b3,
    optimum,h,SSize_u,SSize_v);
    //return increment of u and v from duds and dvds

    if((pu+SSize_u) > 1.0)
        {
        if(pu==1.0)
            {
            SSize_u=0.0;
            SSize_v=0.0;
            }
        else
            {
            ratio=(1.0-pu)/SSize_u;
            SSize_u*=ratio;
            SSize_v*=ratio;
            }
        }

    if((pv+SSize_v) > 1.0)
        {
        if(pv==1.0)
            {
            SSize_u=0.0;
            SSize_v=0.0;
            }
        else
            {
            ratio=(1.0-pv)/SSize_v;
            SSize_u*=ratio;
            SSize_v*=ratio;
            }
```

```
    }

FS_LAC(u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_n1,u1_n2,u1_n3,AlphaLAC,
InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC,pu,SSize_u);
//based on the increment
of u as stepsize of LAC, then return the details
FS_LASC(u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_n1,u2_n2,u2_n3,u2_b1,
u2_b2,u2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,Lambda_Max_LASC,
InitRT_Max_LASC,ArcLength_Max_LASC,pu,SSize_u);
//based on the incremnt
of u as stepsize of LASC, then return the details

FS_LAC(v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_n1,v1_n2,v1_n3,AlphaLAC,
InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC,pv,SSize_v);
//based on the increment
of v as stepsize of LAC, then return the details
FS_LASC(v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_n1,v2_n2,v2_n3,v2_b1,
v2_b2,v2_b3,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,Lambda_Min_LASC,
InitRT_Min_LASC,ArcLength_Min_LASC,pv,SSize_v);
//based on the increment
of v as stepsize of LASC, then return the details

pu+=SSize_u;
pv+=SSize_v;


First_Derivatives(u1_dt1,u1_dt2,u1_dt3,u1_n1,u1_n2,u1_n3,pu,AlphaLAC,
InitRCLAC,Lambda_Max_LAC,ArcLength_Max_LAC);

First_Derivatives(u2_dt1,u2_dt2,u2_dt3,u2_n1,u2_n2,u2_n3,pu,AlphaLAC,
InitRCLAC,Lambda_Max_LASC,ArcLength_Max_LASC);

First_Derivatives(v1_dt1,v1_dt2,v1_dt3,v1_n1,v1_n2,v1_n3,pv,AlphaLAC,
InitRCLAC,Lambda_Min_LAC,ArcLength_Min_LAC);

First_Derivatives(v2_dt1,v2_dt2,v2_dt3,v2_n1,v2_n2,v2_n3,pv,AlphaLAC,
InitRCLAC,Lambda_Min_LASC,ArcLength_Min_LASC);


x[j]=((1-pv)*u1_x1+pv*u2_x1+(1-pu)*v1_x1+pu*v2_x1-((1-pu)*(1-pv)*
maxPoint00x+pu*(1-pv)*maxPoint03x+(1-pu)*pv*maxPoint30x+pu*pv*
maxPoint33x));

y[j]=((1-pv)*u1_x2+pv*u2_x2+(1-pu)*v1_x2+pu*v2_x2-((1-pu)*(1-pv)*
maxPoint00y+pu*(1-pv)*maxPoint03y+(1-pu)*pv*maxPoint30y+pu*pv*
maxPoint33y));
```

```
    z[j]=((1-pv)*u1_x3+pv*u2_x3+(1-pu)*v1_x3+pu*v2_x3-((1-pu)*(1-pv)*
    maxPoint00z+pu*(1-pv)*maxPoint03z+(1-pu)*pv*maxPoint30z+pu*pv*
    maxPoint33z));

    GeoCur[j]=GeoCurforBoundaries(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,
    u1_dt1,u1_dt2,u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,
    u2_dt3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,
    v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum);

}

}
__device__ float RK4_Init(float x1, float t1, float SSize)

{
float k1,k2,k3,k4;

k1=SSize*x1;
k2=SSize*(x1+0.5*SSize*t1);
k3=SSize*(x1+0.5*SSize*t1);
k4=SSize*(x1+SSize*t1);

return (1.0/6.0)*(k1+2*k2+2*k3+k4);
}

__device__ void Init_Curve_2D_Boundaries( float GeoCur[],  float init_x,
    float init_y, float x[], float y[], float tangent1[], float
    tangent2[], float normal1[], float normal2[], mint optimum)
{//use to compute plane curve coordinates from given geodesic curvature
float DummyGeoCur1,DummyGeoCur2,DGCDs;
float x1, x2;
float t1, t2, dt1, dt2, d2t1, d2t2;
float n1, n2, dn1, dn2, d2n1, d2n2;

x1=init_x;
x2=init_y;

if (optimum==max)//when curve render in maximum principle direction, the
    tangent is {-1,0,0} at (-x)(-y) plane. We cross {-1,0,0}x{0,0,1}
    then we get normal is {0,1,0}. Then we flip the curve from (-x)(-y)
    plane to (-x)(y) plane we get tangent is {-1,0,0} and normal
    is {0,-1,0}. Finally we flip the curve from (-x)(y) plane to
    (x)(y) plane then we have tangent {1,0,0} and normal is {0,-1,0}.
{
    t1=1;
    t2=0;
    n1=0;
```

```
    n2=-1;
}
else if(optimum==min)//when curve render in minimum principle direction, the
     tangent is {0,-1,0} at (-x)(-y) plane. We cross {0,-1,0}x{0,0,1}
     then we get normal is {-1,0,0}. Then we flip the curve from
     (-x)(-y) plane to (-x)(y) plane we get tangent is {0,1,0} and
     normal is {-1,0,0}. Finally we flip the curve from (-x)(y) plane
     to (x)(y) plane then we have tangent {0,1,0} and normal is {1,0,0}.
{
    t1=0;
    t2=1;
    n1=1;
    n2=0;
}

x[0]=x1;
y[0]=x2;
tangent1[0]=t1;
tangent2[0]=t2;
normal1[0]=n1;
normal2[0]=n2;

for(mint j=1;j<n;j++)
{

DummyGeoCur1=GeoCur[j-1];
DummyGeoCur2=GeoCur[j];
DGCDs=(DummyGeoCur2-DummyGeoCur1)/h;
dt1=DummyGeoCur1*n1;
dt2=DummyGeoCur1*n2;
dn1=-DummyGeoCur1*t1;
dn2=-DummyGeoCur1*t2;
d2t1=DummyGeoCur1*dn1+DGCDs*n1;
d2t2=DummyGeoCur1*dn2+DGCDs*n2;
d2n1=-DummyGeoCur1*dt1-DGCDs*t1;
d2n2=-DummyGeoCur1*dt2-DGCDs*t2;

x1=x1+RK4_Init(t1,dt1,h);
x2=x2+RK4_Init(t2,dt2,h);
t1=t1+RK4_Init(dt1,d2t1,h);
t2=t2+RK4_Init(dt2,d2t2,h);
n1=n1+RK4_Init(dn1,d2n1,h);
n2=n2+RK4_Init(dn2,d2n2,h);

x[j]=x1;
y[j]=x2;
tangent1[j]=t1;
```

```
tangent2[j]=t2;
normal1[j]=n1;
normal2[j]=n2;
}

}


__device__ void CutCurve_2D( float GeoCur[], mint max_n, float init_x,
     float init_y, float init_t1, float init_t2, float init_n1,
     float init_n2, float x[], float y[], float tangent1[], float
     tangent2[], float normal1[], float normal2[], mint optimum)
{//use to compute plane curve coordinates from given geodesic curvature
float DummyGeoCur1,DummyGeoCur2,DGCDs;
float x1, x2;
float t1, t2, dt1, dt2, d2t1, d2t2;
float n1, n2, dn1, dn2, d2n1, d2n2;

x1=init_x;
x2=init_y;

if (optimum==max)//when curve render in maximum principle direction, the
     tangent is {-1,0,0} at (-x)(-y) plane. We cross {-1,0,0}x{0,0,1}
     then we get normal is {0,1,0}. Then we flip the curve from (-x)(-y)
     plane to (-x)(y) plane we get tangent is {-1,0,0} and normal is
     {0,-1,0}. Finally we flip the curve from (-x)(y) plane to (x)(y)
     plane then we have tangent {1,0,0} and normal is {0,-1,0}. Then
     we reverse the normal={0,1,0} so that we can cut out the pieces
{
    t1=-init_n1;
    t2=-init_n2;
    n1=init_t1;
    n2=init_t2;
}
else if(optimum==min)//when curve render in minimum principle direction, the
     tangent is {0,-1,0} at (-x)(-y) plane. We cross {0,-1,0}x{0,0,1}
     then we get normal is {-1,0,0}. Then we flip the curve from (-x)(-y)
     plane to (-x)(y) plane we get tangent is {0,1,0} and normal is
     {-1,0,0}. Finally we flip the curve from (-x)(y) plane to (x)(y)
     plane then we have tangent {0,1,0} and normal is {1,0,0}. Then
     we reverse the normal={-1,0,0} so that we can cut out the pieces
{
    t1=init_n1;
    t2=init_n2;
    n1=-init_t1;
    n2=-init_t2;
}
```

```
x[max_n]=x1;
y[max_n]=x2;
tangent1[max_n]=t1;
tangent2[max_n]=t2;
normal1[max_n]=n1;
normal2[max_n]=n2;

for(mint j=max_n-1;j>=0;j--)
{

DummyGeoCur1=GeoCur[j+1];
DummyGeoCur2=GeoCur[j];
DGCDs=(DummyGeoCur2-DummyGeoCur1)/h;
dt1=DummyGeoCur1*n1;
dt2=DummyGeoCur1*n2;
dn1=-DummyGeoCur1*t1;
dn2=-DummyGeoCur1*t2;
d2t1=DummyGeoCur1*dn1+DGCDs*n1;
d2t2=DummyGeoCur1*dn2+DGCDs*n2;
d2n1=-DummyGeoCur1*dt1-DGCDs*t1;
d2n2=-DummyGeoCur1*dt2-DGCDs*t2;

x1=x1+RK4_Init(t1,dt1,h);
x2=x2+RK4_Init(t2,dt2,h);
t1=t1+RK4_Init(dt1,d2t1,h);
t2=t2+RK4_Init(dt2,d2t2,h);
n1=n1+RK4_Init(dn1,d2n1,h);
n2=n2+RK4_Init(dn2,d2n2,h);


x[j]=x1;
y[j]=x2;
tangent1[j]=t1;
tangent2[j]=t2;
normal1[j]=n1;
normal2[j]=n2;
}

}

__device__ void ReturnInteger( mint& min_i, float x1[],
    float y1[], float z1[], float x2[], float y2[], float z2[])
{
float dif1;
float min_dif1=5;
```

```
for(mint i=0; i<n; i++)
{
    for(mint j=0; j<n; j++)
    {
        dif1=pow(x1[i]-x2[j],2)+pow(y1[i]-y2[j],2)+pow(z1[i]-z2[j],2);

        if (dif1<min_dif1)
        {
            min_dif1=dif1;
            min_i=i;
        }
    }
}
}

__device__ void ReturnPointsPosition( mint& res_i, mint& res_min_j, mint&
    res_max_j, float x1[], float y1[], float z1[], float x2[],
    float y2[], float z2[], float x3[], float y3[], float z3[])
{
mint min_i;
mint min_j, max_j;
float dif1, dif2;
float min_dif1=5;
float min_dif2=5;

for(mint i=0; i<n; i++)
{
    for(mint j=0; j<n; j++)
    {
        dif1=pow(x1[i]-x3[j],2)+pow(y1[i]-y3[j],2)+pow(z1[i]-z3[j],2);
        dif2=pow(x2[i]-x3[j],2)+pow(y2[i]-y3[j],2)+pow(z2[i]-z3[j],2);

        if (dif1<min_dif1)
        {
            min_dif1=dif1;
            min_j=j;
        }

        if (dif2<min_dif2)
        {
            min_dif2=dif2;
            min_i=i;
            max_j=j;
        }
    }
}
```

```
res_i=min_i;
res_min_j=min_j;
res_max_j=max_j;

}



__device__ void ObtainSmallerPiece( mint res_i, mint min_j, mint max_j, float
    GeoCur[], float x_2d[], float y_2d[], float t1_2d[], float t2_2d[],
    float n1_2d[], float n2_2d[], float& resx, float& resy, mint optimum)
{
float DummyGeoCur1,DummyGeoCur2,DGCDs;
float x1, x2;
float t1, t2, dt1, dt2, d2t1, d2t2;
float n1, n2, dn1, dn2, d2n1, d2n2;



x1=x_2d[res_i];
x2=y_2d[res_i];

if (optimum==max)//Since the 1st curve is in minimum principle
    direction (from curve to cut) and its tangent is tangent1={0,1}
    and normal is normal1={-1,0}. The orthogonal curve is
    rendered in maximum principle direction , hence tangent is
    tangento={1,0}=-normal1 and normal is normalo={0,-1}=-tangent1.
{
    t1=n1_2d[res_i];
    t2=n2_2d[res_i];
    n1=t1_2d[res_i];
    n2=t2_2d[res_i];

}
else if(optimum==min)//Since the 1st curve is in maximum
    principle direction (from curve to cut) and its tangent is
    tangent1={1,0} and normal is normal1={0,1}. The orthogonal curve
    is rendered in minimum principle direction , hence tangent is
    tangento={0,1}=normal1 and normal is normalo={1,0}=tangent1.
{
    t1=-n1_2d[res_i];
    t2=-n2_2d[res_i];
    n1=-t1_2d[res_i];
    n2=-t2_2d[res_i];
}

for(mint j=max_j-1;j>=min_j;j--)
{
```

```
DummyGeoCur1=GeoCur[j+1];
DummyGeoCur2=GeoCur[j];
DGCDs=(DummyGeoCur2-DummyGeoCur1)/h;
dt1=DummyGeoCur1*n1;
dt2=DummyGeoCur1*n2;
dn1=-DummyGeoCur1*t1;
dn2=-DummyGeoCur1*t2;
d2t1=DummyGeoCur1*dn1+DGCDs*n1;
d2t2=DummyGeoCur1*dn2+DGCDs*n2;
d2n1=-DummyGeoCur1*dt1-DGCDs*t1;
d2n2=-DummyGeoCur1*dt2-DGCDs*t2;

x1=x1+RK4_Init(t1,dt1,h);
x2=x2+RK4_Init(t2,dt2,h);
t1=t1+RK4_Init(dt1,d2t1,h);
t2=t2+RK4_Init(dt2,d2t2,h);
n1=n1+RK4_Init(dn1,d2n1,h);
n2=n2+RK4_Init(dn2,d2n2,h);
}

resx=x1;
resy=x2;

}

__device__ void SmallerPieceBoundary( float resx[], float resy[], float
    GeoCur[], mint min_j, mint max_j, float x_2d, float y_2d, float
    t1_2d, float t2_2d, float n1_2d, float n2_2d, mint optimum)
{
float DummyGeoCur1,DummyGeoCur2,DGCDs;
float x1, x2;
float t1, t2, dt1, dt2, d2t1, d2t2;
float n1, n2, dn1, dn2, d2n1, d2n2;


x1=x_2d;
x2=y_2d;

if (optimum==max)//Since the 1st curve is in minimum principle
    direction (from curve to cut) and its tangent is tangent1={0,1}
    and normal is normal1={-1,0}. The orthogonal curve is
    rendered in maximum principle direction , hence tangent is
    tangento={1,0}=-normal1 and normal is normalo={0,-1}=-tangent1.
{
    t1=n1_2d;
    t2=n2_2d;
    n1=t1_2d;
```

```
    n2=t2_2d;

}
else if(optimum==min)//Since the 1st curve is in maximum
     principle direction (from curve to cut) and its tangent is
     tangent1={1,0} and normal is normal1={0,1}. The orthogonal curve
     is rendered in minimum principle direction , hence tangent is
     tangento={0,1}=normal1 and normal is normalo={1,0}=tangent1.
{
    t1=-n1_2d;
    t2=-n2_2d;
    n1=-t1_2d;
    n2=-t2_2d;
}

resx[0]=x1;
resy[0]=x2;


for(mint i=1,mint j=max_j-1;j>=min_j;i++,j--)
{

DummyGeoCur1=GeoCur[j+1];
DummyGeoCur2=GeoCur[j];
DGCDs=(DummyGeoCur2-DummyGeoCur1)/h;
dt1=DummyGeoCur1*n1;
dt2=DummyGeoCur1*n2;
dn1=-DummyGeoCur1*t1;
dn2=-DummyGeoCur1*t2;
d2t1=DummyGeoCur1*dn1+DGCDs*n1;
d2t2=DummyGeoCur1*dn2+DGCDs*n2;
d2n1=-DummyGeoCur1*dt1-DGCDs*t1;
d2n2=-DummyGeoCur1*dt2-DGCDs*t2;

x1=x1+RK4_Init(t1,dt1,h);
x2=x2+RK4_Init(t2,dt2,h);
t1=t1+RK4_Init(dt1,d2t1,h);
t2=t2+RK4_Init(dt2,d2t2,h);
n1=n1+RK4_Init(dn1,d2n1,h);
n2=n2+RK4_Init(dn2,d2n2,h);
resx[i]=x1;
resy[i]=x2;

}
}

__global__ void ReturnSmallPieceBoundary( float *
```

```
    xvalue, float * yvalue, float GeoCur[], mint min_j, mint
    max_j, float x_2d, float y_2d, float t1_2d, float t2_2d,
    float n1_2d, float n2_2d, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float res_x[180], res_y[180];

SmallerPieceBoundary(res_x,res_y,GeoCur,min_j,max_j,x_2d,y_2d,t1_2d,t2_2d,n1_2d
    ,n2_2d,optimum);
if( index < ListSize)
{
    xvalue[index]= res_x[index];
    yvalue[index]= res_y[index];
}

}


__global__ void ReturnEndPoints( float * xvalue, float * yvalue, mint
    res_i[], mint res_min_j[], mint res_max_j[], float x_2d[],
    float y_2d[], float t1_2d[], float t2_2d[], float n1_2d[],
    float n2_2d[], mint SPoint, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float res_x, res_y;
mint min_i, min_j, max_j;
float x3[n], y3[n], z3[n], GeoCur3[n];

if( index < ListSize)
{
    ReturnDetail(index+SPoint,x3,y3,z3,GeoCur3,optimum);
    min_i=res_i[index];
    min_j=res_min_j[index];
    max_j=res_max_j[index];

    ObtainSmallerPiece(min_i,min_j,max_j,GeoCur3,x_2d,y_2d,t1_2d,t2_2d,n1_2d
    ,n2_2d,res_x,res_y,optimum);
    xvalue[index]= res_x;
    yvalue[index]= res_y;
}

}



__global__ void ReturnIntersectionPoint( mint * xvalue, float x1[], float
    y1[], float z1[], float x2[], float y2[], float z2[], mint ListSize)
```

```
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
mint mint_i1;

ReturnInteger( mint_i1, x1, y1, z1, x2, y2, z2);//compute intersection point

if( index < ListSize)
{
    xvalue[index]= mint_i1;
}
}

__global__ void ReturnPositionofPoints( mint * ivalue, mint * min_jvalue, mint
     * max_jvalue, float x1[], float y1[], float z1[], float x2[],
     float y2[], float z2[], mint SPoint, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
mint res_i, res_min_j, res_max_j;
float x3[n], y3[n], z3[n];


if( index < ListSize)
{
    ReturnXYZCoordinate(index+SPoint,x3,y3,z3,optimum);

    ReturnPointsPosition(res_i,res_min_j,res_max_j,x1,y1,z1,x2,y2,z2,x3,y3,z3);
    ivalue[index]= res_i;
    min_jvalue[index]= res_min_j;
    max_jvalue[index]= res_max_j;
}

}
__global__ void BoundaryPositionofPoints( mint * min_jvalue, mint *
     max_jvalue, float x1[], float y1[], float z1[], float x2[], float
     y2[], float z2[], float x3[], float y3[], float z3[], mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
mint res_i, res_min_j, res_max_j;

ReturnPointsPosition(res_i,res_min_j,res_max_j,x1,y1,z1,x2,y2,z2,x3,y3,z3);

if( index < ListSize)
{
    min_jvalue[index]= res_min_j;
    max_jvalue[index]= res_max_j;
}
```

```
}


__global__ void ReturnAllDetailsforBoundaries(
    float * xvalue, float * yvalue, float * zvalue, float
    * GCvalue, mint init_s, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float x[n], y[n], z[n], GeoCur[n];

ReturnDetailforBoundaries(init_s,x,y,z,GeoCur,optimum);

if( index < ListSize)
{
    xvalue[index]= x[index];
    yvalue[index]= y[index];
    zvalue[index]= z[index];
    GCvalue[index]= GeoCur[index];

}

}

__global__ void Boundariesin2D( float * xvalue, float * yvalue, float *
    t1value, float * t2value, float * n1value, float * n2value, float
    GeoCur[], float init_x, float init_y, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float x_2d[n], y_2d[n], t1_2d[n], t2_2d[n], n1_2d[n], n2_2d[n];

Init_Curve_2D_Boundaries( GeoCur, init_x,
    init_y, x_2d, y_2d, t1_2d, t2_2d, n1_2d, n2_2d, optimum);

if( index < ListSize)
{
    xvalue[index]= x_2d[index];
    yvalue[index]= y_2d[index];
    t1value[index]= t1_2d[index];
    t2value[index]= t2_2d[index];
    n1value[index]= n1_2d[index];
    n2value[index]= n2_2d[index];
}
}
__global__ void CutCurvein2D( float * xvalue, float * yvalue, float
    * t1value, float * t2value, float * n1value, float * n2value,
    float GeoCur[], float init_x, float init_y, float init_t1, float
    init_t2, float init_n1, float init_n2, mint optimum, mint ListSize)
```

```
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float x_2d[n], y_2d[n], t1_2d[n], t2_2d[n], n1_2d[n], n2_2d[n];

CutCurve_2D( GeoCur, ListSize-1, init_x, init_y, init_t1, init_t2, init_n1,
    init_n2, x_2d, y_2d, t1_2d, t2_2d, n1_2d, n2_2d, optimum);

if( index < ListSize)
{
    xvalue[index]= x_2d[index];
    yvalue[index]= y_2d[index];
    t1value[index]= t1_2d[index];
    t2value[index]= t2_2d[index];
    n1value[index]= n1_2d[index];
    n2value[index]= n2_2d[index];
}

}

__global__ void ReturnAllDetails( float* xvalue, float* yvalue, float*
    zvalue, float* GCvalue, mint init_s, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float x[n], y[n], z[n], GeoCur[n];

ReturnDetail(init_s,x,y,z,GeoCur,optimum);

if( index < ListSize)
{
    xvalue[index]= x[index];
    yvalue[index]= y[index];
    zvalue[index]= z[index];
    GCvalue[index]= GeoCur[index];

}

}


";

Clear["Global`*"]

Needs["CUDALink`"]
```

```
CSDetails = CUDAFunctionLoad[kernelcodeCPLAS2, "ReturnAllDetails",
   {{"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, _Integer, _Integer, _Integer}, 160];
CSBoundaryDetails = CUDAFunctionLoad[kernelcodeCPLAS2,
   "ReturnAllDetailsforBoundaries",
   {{"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, _Integer, _Integer, _Integer}, 160];
CSBoundary2D = CUDAFunctionLoad[kernelcodeCPLAS2, "Boundariesin2D",
   {{"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Input"}, "Float", "Float", _Integer, _Integer}, 160];
CSCutCurve2D = CUDAFunctionLoad[kernelcodeCPLAS2, "CutCurvein2D",
   {{"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, {"Float", _, "Input"}, "Float", "Float",
    "Float", "Float", "Float", "Float", _Integer, _Integer}, 160];
CSReturnIntersection = CUDAFunctionLoad[kernelcodeCPLAS2,
   "ReturnIntersectionPoint", {{_Integer, _, "Output"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, _Integer}, 160];
CSPointsPosition3D = CUDAFunctionLoad[kernelcodeCPLAS2,
   "ReturnPositionofPoints", {{_Integer, _, "Output"}, {_Integer, _, "Output"},
    {_Integer, _, "Output"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, _Integer, _Integer, _Integer}, 160];
CSSmallPieceBoundary2D = CUDAFunctionLoad[kernelcodeCPLAS2,
   "ReturnSmallPieceBoundary", {{"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Input"}, _Integer, _Integer, "Float", "Float",
    "Float", "Float", "Float", "Float", _Integer, _Integer}, 160];
CSBoundaryPosition3D = CUDAFunctionLoad[kernelcodeCPLAS2,
   "BoundaryPositionofPoints",
   {{_Integer, _, "Output"}, {_Integer, _, "Output"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, _Integer}, 160];
```

```
AbsoluteTiming[
 CSEndPoints2D = CUDAFunctionLoad[kernelcodeCPLAS2, "ReturnEndPoints",
   {{"Float", _, "Output"}, {"Float", _, "Output"}, {_Integer, _, "Input"},
    {_Integer, _, "Input"}, {_Integer, _, "Input"}, {"Float", _, "Input"},
    {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float", _, "Input"}, {"Float",
      _, "Input"}, {"Float", _, "Input"}, _Integer, _Integer, _Integer}, 160]]
```

```
{118.321,
 CUDAFunction[<>, ReturnEndPoints, {{Float, _, Output}, {Float, _, Output},
   {_Integer, _, Input}, {_Integer, _, Input}, {_Integer, _, Input},
   {Float, _, Input}, {Float, _, Input}, {Float, _, Input}, {Float, _, Input},
   {Float, _, Input}, {Float, _, Input}, Integer64, Integer64, Integer64}]}
```

```
CSDetail[v1_, n1_, optimum_] :=
  Module[{v11 = v1, nn = n1, opt1 = optimum, Tx, Ty, Tz, TGC},
   Tx = CUDAMemoryAllocate["Float", nn];
   Ty = CUDAMemoryAllocate["Float", nn];
   Tz = CUDAMemoryAllocate["Float", nn];
   TGC = CUDAMemoryAllocate["Float", nn];
   CSDetails[Tx, Ty, Tz, TGC, v11, opt1, nn];
   Transpose[{CUDAMemoryGet[Tx],
     CUDAMemoryGet[Ty], CUDAMemoryGet[Tz], CUDAMemoryGet[TGC]}]];

CSBoundaryDetail[v1_, n1_, optimum_] :=
  Module[{v11 = v1, nn = n1, opt1 = optimum, Tx, Ty, Tz, TGC},
   Tx = CUDAMemoryAllocate["Float", nn];
   Ty = CUDAMemoryAllocate["Float", nn];
   Tz = CUDAMemoryAllocate["Float", nn];
   TGC = CUDAMemoryAllocate["Float", nn];
   CSBoundaryDetails[Tx, Ty, Tz, TGC, v11, opt1, nn];
   Transpose[{CUDAMemoryGet[Tx],
     CUDAMemoryGet[Ty], CUDAMemoryGet[Tz], CUDAMemoryGet[TGC]}]];

CSPointsPosition[v1_, v2_, {SPoint_, n3_}, optimum_] :=
  Module[{opt1, opt2 = optimum, n1 = 1201, n2 = 1201, Tx1, Ty1,
    Tz1, Tx2, Ty2, Tz2, TGC1, TGC2, resi, resminj, resmaxj},
   opt1 = Piecewise[{{0, optimum == 1}}, 1];
   Tx1 = CUDAMemoryAllocate["Float", n1];
   Ty1 = CUDAMemoryAllocate["Float", n1];
   Tz1 = CUDAMemoryAllocate["Float", n1];
   TGC1 = CUDAMemoryAllocate["Float", n1];
   Tx2 = CUDAMemoryAllocate["Float", n2];
   Ty2 = CUDAMemoryAllocate["Float", n2];
   Tz2 = CUDAMemoryAllocate["Float", n2];
   TGC2 = CUDAMemoryAllocate["Float", n2];
   resi = CUDAMemoryAllocate[_Integer, n3];
   resminj = CUDAMemoryAllocate[_Integer, n3];
   resmaxj = CUDAMemoryAllocate[_Integer, n3];

   Piecewise[{{CSBoundaryDetails[Tx1, Ty1, Tz1, TGC1, v1, opt1, n1], v1 == 0}},
    CSDetails[Tx1, Ty1, Tz1, TGC1, v1, opt1, n1]];
   Piecewise[{{CSBoundaryDetails[Tx2, Ty2, Tz2, TGC2, v2, opt1, n2], v2 == 1000}},
    CSDetails[Tx2, Ty2, Tz2, TGC2, v2, opt1, n2]];
   CSPointsPosition3D[resi, resminj, resmaxj, Tx1, Ty1,
    Tz1, Tx2, Ty2, Tz2, SPoint, opt2, n3];
   Transpose[{CUDAMemoryGet[resi], CUDAMemoryGet[resminj],
     CUDAMemoryGet[resmaxj]}]];
```

```
CSPointsPosition[{v1_, v2_}, {n3_}, optimum_] :=
  Module[{opt1, opt2 = optimum, nn = 1201, Tx1, Ty1, Tz1, TGC1,
    Tx2, Ty2, Tz2, TGC2, Tx3, Ty3, Tz3, TGC3, resminj, resmaxj},
   opt1 = Piecewise[{{0, optimum == 1}}, 1];
   Tx1 = CUDAMemoryAllocate["Float", nn];
   Ty1 = CUDAMemoryAllocate["Float", nn];
   Tz1 = CUDAMemoryAllocate["Float", nn];
   TGC1 = CUDAMemoryAllocate["Float", nn];
   Tx2 = CUDAMemoryAllocate["Float", nn];
   Ty2 = CUDAMemoryAllocate["Float", nn];
   Tz2 = CUDAMemoryAllocate["Float", nn];
   TGC2 = CUDAMemoryAllocate["Float", nn];
   Tx3 = CUDAMemoryAllocate["Float", nn];
   Ty3 = CUDAMemoryAllocate["Float", nn];
   Tz3 = CUDAMemoryAllocate["Float", nn];
   TGC3 = CUDAMemoryAllocate["Float", nn];

   resminj = CUDAMemoryAllocate[_Integer, n3];
   resmaxj = CUDAMemoryAllocate[_Integer, n3];

   Piecewise[{{CSBoundaryDetails[Tx1, Ty1, Tz1, TGC1, v1, opt1, nn], v1 == 0}},
    CSDetails[Tx1, Ty1, Tz1, TGC1, v1, opt1, nn]];
   Piecewise[{{CSBoundaryDetails[Tx2, Ty2, Tz2, TGC2, v2, opt1, nn], v2 == 1000}},
    CSDetails[Tx2, Ty2, Tz2, TGC2, v2, opt1, nn]];
   CSBoundaryDetails[Tx3, Ty3, Tz3, TGC3, 0, opt2, nn];
   CSBoundaryPosition3D[resminj,
    resmaxj, Tx1, Ty1, Tz1, Tx2, Ty2, Tz2, Tx3, Ty3, Tz3, n3];
   Transpose[{CUDAMemoryGet[resminj], CUDAMemoryGet[resmaxj]}]];

CSIntersection[v1_, v2_, optimum_] := Module[{opt1, opt2 = optimum,
    n1 = 1201, n2 = 1201, Tx1, Ty1, Tz1, Tx2, Ty2, Tz2, TGC1, TGC2, resi},
   opt1 = Piecewise[{{0, optimum == 1}}, 1];
   Tx1 = CUDAMemoryAllocate["Float", n1];
   Ty1 = CUDAMemoryAllocate["Float", n1];
   Tz1 = CUDAMemoryAllocate["Float", n1];
   TGC1 = CUDAMemoryAllocate["Float", n1];
   Tx2 = CUDAMemoryAllocate["Float", n2];
   Ty2 = CUDAMemoryAllocate["Float", n2];
   Tz2 = CUDAMemoryAllocate["Float", n2];
   TGC2 = CUDAMemoryAllocate["Float", n2];
   resi = CUDAMemoryAllocate[_Integer, 1];

   CSDetails[Tx1, Ty1, Tz1, TGC1, v1, opt1, n1];
   CSDetails[Tx2, Ty2, Tz2, TGC2, v2, opt2, n2];
   CSReturnIntersection[resi, Tx1, Ty1, Tz1, Tx2, Ty2, Tz2, 1];
   Transpose[{CUDAMemoryGet[resi]}]];
```

```
CSInitialBoundary2D[{Initx_, Inity_}, {n1_, optimum_}] :=
  Module[{v1 = 1000, opt1 = optimum, InitTx, InitTy, InitTz, InitTGC,
     InitTx2D, InitTy2D, InitTt12D, InitTt22D, InitTn12D, InitTn22D},
   InitTx = CUDAMemoryAllocate["Float", n1];
   InitTy = CUDAMemoryAllocate["Float", n1];
   InitTz = CUDAMemoryAllocate["Float", n1];
   InitTGC = CUDAMemoryAllocate["Float", n1];
   InitTx2D = CUDAMemoryAllocate["Float", n1];
   InitTy2D = CUDAMemoryAllocate["Float", n1];
   InitTt12D = CUDAMemoryAllocate["Float", n1];
   InitTt22D = CUDAMemoryAllocate["Float", n1];
   InitTn12D = CUDAMemoryAllocate["Float", n1];
   InitTn22D = CUDAMemoryAllocate["Float", n1];
   CSBoundaryDetails[InitTx, InitTy, InitTz, InitTGC, v1, opt1, n1];
   CSBoundary2D[InitTx2D, InitTy2D, InitTt12D, InitTt22D,
    InitTn12D, InitTn22D, InitTGC, Initx, Inity, opt1, n1];
   Transpose[{CUDAMemoryGet[InitTx2D], CUDAMemoryGet[InitTy2D]}]];
```

```
CSCutCurvein2D[{Initx_, Inity_}, {n1_}, {v2_, n2_, optimum_}] :=
  Module[{nn = 1201, v1 = 1000, opt1, opt2 = optimum, InitTx2D,
    InitTy2D, InitTt12D, InitTt22D, InitTn12D, InitTn22D, TBx, TBy,
    TBz, TBGC, Tx1, Ty1, Tz1, TGC1, resi, ivalue, Tx2D, Ty2D, Tt12D,
    Tt22D, Tn12D, Tn22D, Inix, Iniy, Init1, Init2, Inin1, Inin2},
   opt1 = Piecewise[{{0, optimum == 1}}, 1];
   InitTx2D = CUDAMemoryAllocate["Float", nn];
   InitTy2D = CUDAMemoryAllocate["Float", nn];
   InitTt12D = CUDAMemoryAllocate["Float", nn];
   InitTt22D = CUDAMemoryAllocate["Float", nn];
   InitTn12D = CUDAMemoryAllocate["Float", nn];
   InitTn22D = CUDAMemoryAllocate["Float", nn];
   TBx = CUDAMemoryAllocate["Float", nn];
   TBy = CUDAMemoryAllocate["Float", nn];
   TBz = CUDAMemoryAllocate["Float", nn];
   TBGC = CUDAMemoryAllocate["Float", nn];
   Tx1 = CUDAMemoryAllocate["Float", nn];
   Ty1 = CUDAMemoryAllocate["Float", nn];
   Tz1 = CUDAMemoryAllocate["Float", nn];
   TGC1 = CUDAMemoryAllocate["Float", nn];
   resi = CUDAMemoryAllocate[_Integer, 1];

   Tx2D = CUDAMemoryAllocate["Float", nn];
   Ty2D = CUDAMemoryAllocate["Float", nn];
   Tt12D = CUDAMemoryAllocate["Float", nn];
   Tt22D = CUDAMemoryAllocate["Float", nn];
   Tn12D = CUDAMemoryAllocate["Float", nn];
   Tn22D = CUDAMemoryAllocate["Float", nn];

   CSBoundaryDetails[TBx, TBy, TBz, TBGC, v1, opt1, nn];
   Piecewise[{{CSBoundaryDetails[Tx1, Ty1, Tz1, TGC1, 0, opt2, nn], v2 == 0},
     {CSBoundaryDetails[Tx1, Ty1, Tz1, TGC1, v2, opt2, nn], v2 == 1000}},
    CSDetails[Tx1, Ty1, Tz1, TGC1, v2, opt2, nn]];
   (*Detail of upper curve*)
   CSReturnIntersection[resi, TBx, TBy, TBz, Tx1, Ty1, Tz1, 1];
   CSBoundary2D[InitTx2D, InitTy2D, InitTt12D,
    InitTt22D, InitTn12D, InitTn22D, TBGC, Initx, Inity, opt1, n1];
   ivalue = First[CUDAMemoryGet[resi]] + 1;
   Inix = CUDAMemoryGet[InitTx2D][[ivalue]];
   Iniy = CUDAMemoryGet[InitTy2D][[ivalue]];
   Init1 = CUDAMemoryGet[InitTt12D][[ivalue]];
   Init2 = CUDAMemoryGet[InitTt22D][[ivalue]];
   Inin1 = CUDAMemoryGet[InitTn12D][[ivalue]];
   Inin2 = CUDAMemoryGet[InitTn22D][[ivalue]];
   CSCutCurve2D[Tx2D, Ty2D, Tt12D, Tt22D, Tn12D,
    Tn22D, TGC1, Inix, Iniy, Init1, Init2, Inin1, Inin2, opt2, n2];
   Transpose[{CUDAMemoryGet[Tx2D][[1 ;; n2]], CUDAMemoryGet[Ty2D][[1 ;; n2]]}]];
```

```
CSEndPointin2D[{Initx_, Inity_},
   {v1_, n1_}, {v2_, n2_}, {SPoint_, n3_}, optimum_] :=
  Module[{n0 = 1188, nn = 1201, v0 = 1000, opt1, opt2 = optimum, InitTx2D,
    InitTy2D, InitTt12D, InitTt22D, InitTn12D, InitTn22D, TBx,
    TBy, TBz, TBGC, Tx1, Ty1, Tz1, TGC1, Tx2, Ty2, Tz2, TGC2, BPosi,
    ivalue, Tx2D, Ty2D, Tt12D, Tt22D, Tn12D, Tn22D, Inix, Iniy, Init1,
    Init2, Inin1, Inin2, resi, resminj, resmaxj, Endx2D, Endy2D},
   opt1 = Piecewise[{{0, optimum == 1}}, 1];
   InitTx2D = CUDAMemoryAllocate["Float", nn];
   InitTy2D = CUDAMemoryAllocate["Float", nn];
   InitTt12D = CUDAMemoryAllocate["Float", nn];
   InitTt22D = CUDAMemoryAllocate["Float", nn];
   InitTn12D = CUDAMemoryAllocate["Float", nn];
   InitTn22D = CUDAMemoryAllocate["Float", nn];
   TBx = CUDAMemoryAllocate["Float", nn];
   TBy = CUDAMemoryAllocate["Float", nn];
   TBz = CUDAMemoryAllocate["Float", nn];
   TBGC = CUDAMemoryAllocate["Float", nn];
   Tx1 = CUDAMemoryAllocate["Float", nn];
   Ty1 = CUDAMemoryAllocate["Float", nn];
   Tz1 = CUDAMemoryAllocate["Float", nn];
   TGC1 = CUDAMemoryAllocate["Float", nn];
   Tx2 = CUDAMemoryAllocate["Float", nn];
   Ty2 = CUDAMemoryAllocate["Float", nn];
   Tz2 = CUDAMemoryAllocate["Float", nn];
   TGC2 = CUDAMemoryAllocate["Float", nn];

   BPosi = CUDAMemoryAllocate[_Integer, 1];

   Tx2D = CUDAMemoryAllocate["Float", nn];
   Ty2D = CUDAMemoryAllocate["Float", nn];
   Tt12D = CUDAMemoryAllocate["Float", nn];
   Tt22D = CUDAMemoryAllocate["Float", nn];
   Tn12D = CUDAMemoryAllocate["Float", nn];
   Tn22D = CUDAMemoryAllocate["Float", nn];


   resi = CUDAMemoryAllocate[_Integer, n3];
   resminj = CUDAMemoryAllocate[_Integer, n3];
   resmaxj = CUDAMemoryAllocate[_Integer, n3];
   Endx2D = CUDAMemoryAllocate["Float", n3];
   Endy2D = CUDAMemoryAllocate["Float", n3];


   CSBoundaryDetails[TBx, TBy, TBz, TBGC, v0, opt1, nn];
   (*Detail of boundary curve*)
   Piecewise[{{CSBoundaryDetails[Tx2, Ty2, Tz2, TGC2, v2, opt2, nn], v2 == 1000}},
```

```
      CSDetails[Tx2, Ty2, Tz2, TGC2, v2, opt2, nn]];
     (*Detail of upper curve*)
     CSReturnIntersection[BPosi, TBx, TBy, TBz, Tx2, Ty2, Tz2, 1];
     (*compute intersection point of boundary curve and upper curve*)
     CSBoundary2D[InitTx2D, InitTy2D, InitTt12D,
      InitTt22D, InitTn12D, InitTn22D, TBGC, Initx, Inity, opt1, n0];
     (*Draw of 2D boundary curve*)
     ivalue = First[CUDAMemoryGet[BPosi]] + 1;
     Inix = CUDAMemoryGet[InitTx2D][[ivalue]];
     Iniy = CUDAMemoryGet[InitTy2D][[ivalue]];
     Init1 = CUDAMemoryGet[InitTt12D][[ivalue]];
     Init2 = CUDAMemoryGet[InitTt22D][[ivalue]];
     Inin1 = CUDAMemoryGet[InitTn12D][[ivalue]];
     Inin2 = CUDAMemoryGet[InitTn22D][[ivalue]];
     CSCutCurve2D[Tx2D, Ty2D, Tt12D, Tt22D, Tn12D,
      Tn22D, TGC2, Inix, Iniy, Init1, Init2, Inin1, Inin2, opt2, n2];
     (*Draw of 2D upper curve*)
     Piecewise[{{CSBoundaryDetails[Tx1, Ty1, Tz1, TGC1, v1, opt2, nn], v1 == 0}},
      CSDetails[Tx1, Ty1, Tz1, TGC1, v1, opt2, nn]];
     (*Detail of lower curve*)

     CSPointsPosition3D[resi, resminj,
      resmaxj, Tx1, Ty1, Tz1, Tx2, Ty2, Tz2, SPoint, opt1, n3];
     CSEndPoints2D[Endx2D, Endy2D, resi, resminj, resmaxj, Tx2D,
      Ty2D, Tt12D, Tt22D, Tn12D, Tn22D, SPoint, opt1, n3];
     Transpose[{CUDAMemoryGet[Endx2D], CUDAMemoryGet[Endy2D]}]];

CSLastBoundary2D[{Initx_, Inity_}, {v1_, n1_}, {v2_, n2_}, {n3_}, optimum_] :=
  Module[{n0 = 1188, nn = 1201, v0 = 1000, opt1, opt2 = optimum, InitTx2D, InitTy2D,
    InitTt12D, InitTt22D, InitTn12D, InitTn22D, TBx, TBy, TBz, TBGC, Tx1, Ty1,
    Tz1, TGC1, Tx2, Ty2, Tz2, TGC2, Tx3, Ty3, Tz3, TGC3, BPosi, ivalue, Tx2D, Ty2D,
    Tt12D, Tt22D, Tn12D, Tn22D, Inix, Iniy, Init1, Init2, Inin1, Inin2, resminj,
    resmaxj, minj, maxj, Dx2D, Dy2D, Dt12D, Dt22D, Dn12D, Dn22D, Endx2D, Endy2D},
   opt1 = Piecewise[{{0, optimum == 1}}, 1];
   InitTx2D = CUDAMemoryAllocate["Float", nn];
   InitTy2D = CUDAMemoryAllocate["Float", nn];
   InitTt12D = CUDAMemoryAllocate["Float", nn];
   InitTt22D = CUDAMemoryAllocate["Float", nn];
   InitTn12D = CUDAMemoryAllocate["Float", nn];
   InitTn22D = CUDAMemoryAllocate["Float", nn];
   TBx = CUDAMemoryAllocate["Float", nn];
   TBy = CUDAMemoryAllocate["Float", nn];
   TBz = CUDAMemoryAllocate["Float", nn];
   TBGC = CUDAMemoryAllocate["Float", nn];
   Tx1 = CUDAMemoryAllocate["Float", nn];
   Ty1 = CUDAMemoryAllocate["Float", nn];
   Tz1 = CUDAMemoryAllocate["Float", nn];
   TGC1 = CUDAMemoryAllocate["Float", nn];
```

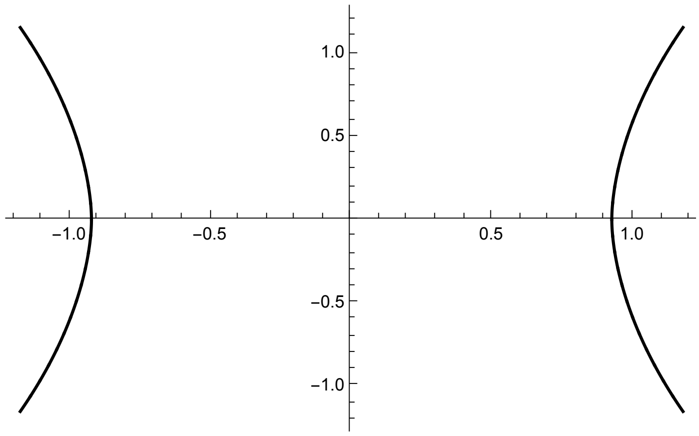
```
Tx2 = CUDAMemoryAllocate["Float", nn];
Ty2 = CUDAMemoryAllocate["Float", nn];
Tz2 = CUDAMemoryAllocate["Float", nn];
TGC2 = CUDAMemoryAllocate["Float", nn];
Tx3 = CUDAMemoryAllocate["Float", nn];
Ty3 = CUDAMemoryAllocate["Float", nn];
Tz3 = CUDAMemoryAllocate["Float", nn];
TGC3 = CUDAMemoryAllocate["Float", nn];

BPosi = CUDAMemoryAllocate[_Integer, 1];

Tx2D = CUDAMemoryAllocate["Float", nn];
Ty2D = CUDAMemoryAllocate["Float", nn];
Tt12D = CUDAMemoryAllocate["Float", nn];
Tt22D = CUDAMemoryAllocate["Float", nn];
Tn12D = CUDAMemoryAllocate["Float", nn];
Tn22D = CUDAMemoryAllocate["Float", nn];

resminj = CUDAMemoryAllocate[_Integer, 1];
resmaxj = CUDAMemoryAllocate[_Integer, 1];
Endx2D = CUDAMemoryAllocate["Float", n3];
Endy2D = CUDAMemoryAllocate["Float", n3];

CSBoundaryDetails[TBx, TBy, TBz, TBGC, v0, opt1, nn];
(*Detail of boundary curve*)
Piecewise[{{CSBoundaryDetails[Tx2, Ty2, Tz2, TGC2, v2, opt2, nn], v2 == 1000}},
 CSDetails[Tx2, Ty2, Tz2, TGC2, v2, opt2, nn]];
(*Detail of upper curve*)
CSReturnIntersection[BPosi, TBx, TBy, TBz, Tx2, Ty2, Tz2, 1];
(*compute intersection point of boundary curve and upper curve*)
CSBoundary2D[InitTx2D, InitTy2D, InitTt12D,
 InitTt22D, InitTn12D, InitTn22D, TBGC, Initx, Inity, opt1, n0];
(*Draw of 2D boundary curve*)
ivalue = First[CUDAMemoryGet[BPosi]] + 1;
Inix = CUDAMemoryGet[InitTx2D][[ivalue]];
Iniy = CUDAMemoryGet[InitTy2D][[ivalue]];
Init1 = CUDAMemoryGet[InitTt12D][[ivalue]];
Init2 = CUDAMemoryGet[InitTt22D][[ivalue]];
Inin1 = CUDAMemoryGet[InitTn12D][[ivalue]];
Inin2 = CUDAMemoryGet[InitTn22D][[ivalue]];
CSCutCurve2D[Tx2D, Ty2D, Tt12D, Tt22D, Tn12D,
 Tn22D, TGC2, Inix, Iniy, Init1, Init2, Inin1, Inin2, opt2, n2];
(*Draw of 2D upper curve*)
Piecewise[{{CSBoundaryDetails[Tx1, Ty1, Tz1, TGC1, v1, opt2, nn], v1 == 0}},
 CSDetails[Tx1, Ty1, Tz1, TGC1, v1, opt2, nn]];
(*Detail of lower curve*)
CSBoundaryDetails[Tx3, Ty3, Tz3, TGC3, 0, opt1, nn];
```

```
CSBoundaryPosition3D[resminj,
 resmaxj, Tx1, Ty1, Tz1, Tx2, Ty2, Tz2, Tx3, Ty3, Tz3, 1];
minj = First[CUDAMemoryGet[resminj]];
maxj = First[CUDAMemoryGet[resmaxj]];
Dx2D = First[CUDAMemoryGet[Tx2D]];
Dy2D = First[CUDAMemoryGet[Ty2D]];
Dt12D = First[CUDAMemoryGet[Tt12D]];
Dt22D = First[CUDAMemoryGet[Tt22D]];
Dn12D = First[CUDAMemoryGet[Tn12D]];
Dn22D = First[CUDAMemoryGet[Tn22D]];

CSSmallPieceBoundary2D[Endx2D, Endy2D, TGC3,
 minj, maxj, Dx2D, Dy2D, Dt12D, Dt22D, Dn12D, Dn22D, opt1, n3];

Transpose[{CUDAMemoryGet[Endx2D], CUDAMemoryGet[Endy2D]}]];
```

First Boundary in 2D

```
PPBoundary = CSInitialBoundary2D[{0.9239453620743007`, 0}, {1188, 0}];
NPBoundary = Table[{-1, 1}, {i, 1, 1188}] * PPBoundary;
NNBoundary = Table[{-1, -1}, {i, 1, 1188}] * PPBoundary;
PNBoundary = Table[{1, -1}, {i, 1, 1188}] * PPBoundary;
```

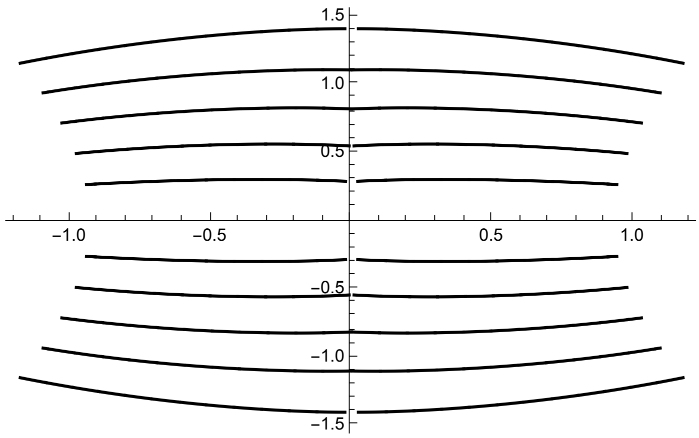
```
ListLinePlot[{PPBoundary, NPBoundary, NNBoundary, PNBoundary}, PlotStyle → Black]
```



Upper Cut Curve (LoC) in 2D

```
PPUpperCut200 = CSCutCurvein2D[{0.9239453620743007`, 0}, {1188}, {200, 923, 1}];
NPUpperCut200 = Table[{-1, 1}, {i, 1, 923}] * PPUpperCut200;
NNUpperCut200 = Table[{-1, -1}, {i, 1, 923}] * PPUpperCut200;
PNUpperCut200 = Table[{1, -1}, {i, 1, 923}] * PPUpperCut200;
```

```
PPUpperCut400 = CSCutCurvein2D[{0.9239453620743007`, 0}, {1188}, {400, 975, 1}];
NPUpperCut400 = Table[{-1, 1}, {i, 1, 975}] * PPUpperCut400;
NNUpperCut400 = Table[{-1, -1}, {i, 1, 975}] * PPUpperCut400;
PNUpperCut400 = Table[{1, -1}, {i, 1, 975}] * PPUpperCut400;

PPUpperCut600 = CSCutCurvein2D[{0.9239453620743007`, 0}, {1188}, {600, 1040, 1}];
NPUpperCut600 = Table[{-1, 1}, {i, 1, 1040}] * PPUpperCut600;
NNUpperCut600 = Table[{-1, -1}, {i, 1, 1040}] * PPUpperCut600;
PNUpperCut600 = Table[{1, -1}, {i, 1, 1040}] * PPUpperCut600;

PPUpperCut800 = CSCutCurvein2D[{0.9239453620743007`, 0}, {1188}, {800, 1112, 1}];
NPUpperCut800 = Table[{-1, 1}, {i, 1, 1112}] * PPUpperCut800;
NNUpperCut800 = Table[{-1, -1}, {i, 1, 1112}] * PPUpperCut800;
PNUpperCut800 = Table[{1, -1}, {i, 1, 1112}] * PPUpperCut800;

PPUpperCut1000 =
  CSCutCurvein2D[{0.9239453620743007`, 0}, {1188}, {1000, 1188, 1}];
NPUpperCut1000 = Table[{-1, 1}, {i, 1, 1188}] * PPUpperCut1000;
NNUpperCut1000 = Table[{-1, -1}, {i, 1, 1188}] * PPUpperCut1000;
PNUpperCut1000 = Table[{1, -1}, {i, 1, 1188}] * PPUpperCut1000;

ListLinePlot[{
  PPUpperCut200, NPUpperCut200, NNUpperCut200, PNUpperCut200,
  PPUpperCut400, NPUpperCut400, NNUpperCut400, PNUpperCut400,
  PPUpperCut600, NPUpperCut600, NNUpperCut600, PNUpperCut600,
  PPUpperCut800, NPUpperCut800, NNUpperCut800, PNUpperCut800,
  PPUpperCut1000, NPUpperCut1000, NNUpperCut1000, PNUpperCut1000},
 PlotStyle → Black]
```



Lower Cut LoC in 2D

When u (or v) start from 0 to 200 and v (or u) is 0

When u (or v) start from 200 to 400 and v (or u) is 0

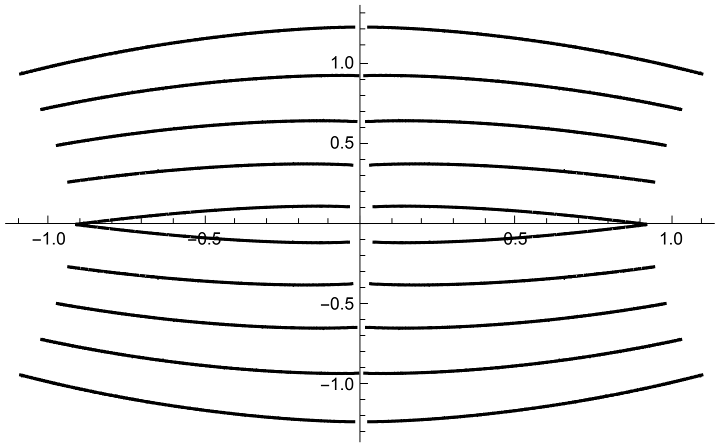
When u (or v) start from 400 to 600 and v (or u) is 0

When u (or v) start from 600 to 800 and v (or u) is 0

When u (or v) start from 800 to 1000 and v (or u) is 0

Draw the line

```
ListLinePlot[{PPLowerCut0, NPLowerCut0, NNLowerCut0, PNLowerCut0,
  PPLowerCut200, NPLowerCut200, NNLowerCut200, PNLowerCut200,
  PPLowerCut400, NPLowerCut400, NNLowerCut400, PNLowerCut400,
  PPLowerCut600, NPLowerCut600, NNLowerCut600, PNLowerCut600,
  PPLowerCut800, NPLowerCut800, NNLowerCut800, PNLowerCut800}, PlotStyle → Black]
```



Last Boundary in 2D

```
PPLastBoundary200 =
  CSLastBoundary2D[{0.9239453620743007`, 0}, {0, 894}, {200, 923}, {179}, 1];
NPLastBoundary200 = Table[{-1, 1}, {i, 1, 179}] * PPLastBoundary200;
NNLastBoundary200 = Table[{-1, -1}, {i, 1, 179}] * PPLastBoundary200;
PNLastBoundary200 = Table[{1, -1}, {i, 1, 179}] * PPLastBoundary200;
```

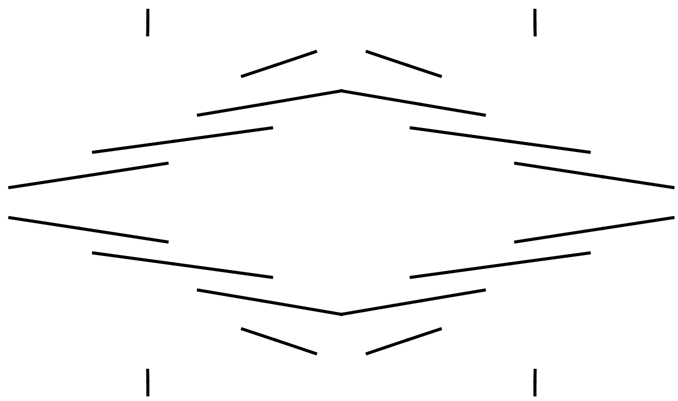
```
PPLastBoundary400 =
  CSLastBoundary2D[{0.9239453620743007`, 0}, {200, 923}, {400, 975}, {180}, 1];
NPLastBoundary400 = Table[{-1, 1}, {i, 1, 180}] * PPLastBoundary400;
NNLastBoundary400 = Table[{-1, -1}, {i, 1, 180}] * PPLastBoundary400;
PNLastBoundary400 = Table[{1, -1}, {i, 1, 180}] * PPLastBoundary400;
```

```
PPLastBoundary600 =
  CSLastBoundary2D[{0.9239453620743007`, 0}, {400, 975}, {600, 1040}, {179}, 1];
NPLastBoundary600 = Table[{-1, 1}, {i, 1, 179}] * PPLastBoundary600;
NNLastBoundary600 = Table[{-1, -1}, {i, 1, 179}] * PPLastBoundary600;
PNLastBoundary600 = Table[{1, -1}, {i, 1, 179}] * PPLastBoundary600;
```

```
PPLastBoundary800 =
  CSLastBoundary2D[{0.9239453620743007`, 0}, {600, 1040}, {800, 1112}, {180}, 1];
NPLastBoundary800 = Table[{-1, 1}, {i, 1, 180}] * PPLastBoundary800;
NNLastBoundary800 = Table[{-1, -1}, {i, 1, 180}] * PPLastBoundary800;
PNLastBoundary800 = Table[{1, -1}, {i, 1, 180}] * PPLastBoundary800;
```

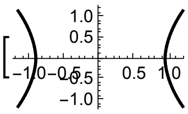
```
PPLastBoundary1000 = CSLastBoundary2D[
   {0.9239453620743007`, 0}, {800, 1112}, {1000, 1188}, {180}, 1];
NPLastBoundary1000 = Table[{-1, 1}, {i, 1, 180}] * PPLastBoundary1000;
NNLastBoundary1000 = Table[{-1, -1}, {i, 1, 180}] * PPLastBoundary1000;
PNLastBoundary1000 = Table[{1, -1}, {i, 1, 180}] * PPLastBoundary1000;
```

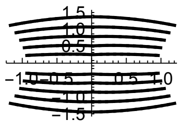
```
ListLinePlot[
 {PPLastBoundary200, NPLastBoundary200, NNLastBoundary200, PNLastBoundary200,
  PPLastBoundary400, NPLastBoundary400, NNLastBoundary400, PNLastBoundary400,
  PPLastBoundary600, NPLastBoundary600, NNLastBoundary600, PNLastBoundary600,
  PPLastBoundary800, NPLastBoundary800, NNLastBoundary800, PNLastBoundary800,
  PPLastBoundary1000, NPLastBoundary1000, NNLastBoundary1000,
  PNLastBoundary1000}, PlotStyle → Black, Axes → None]
```

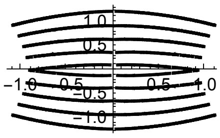


(b)

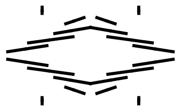
Draw the line

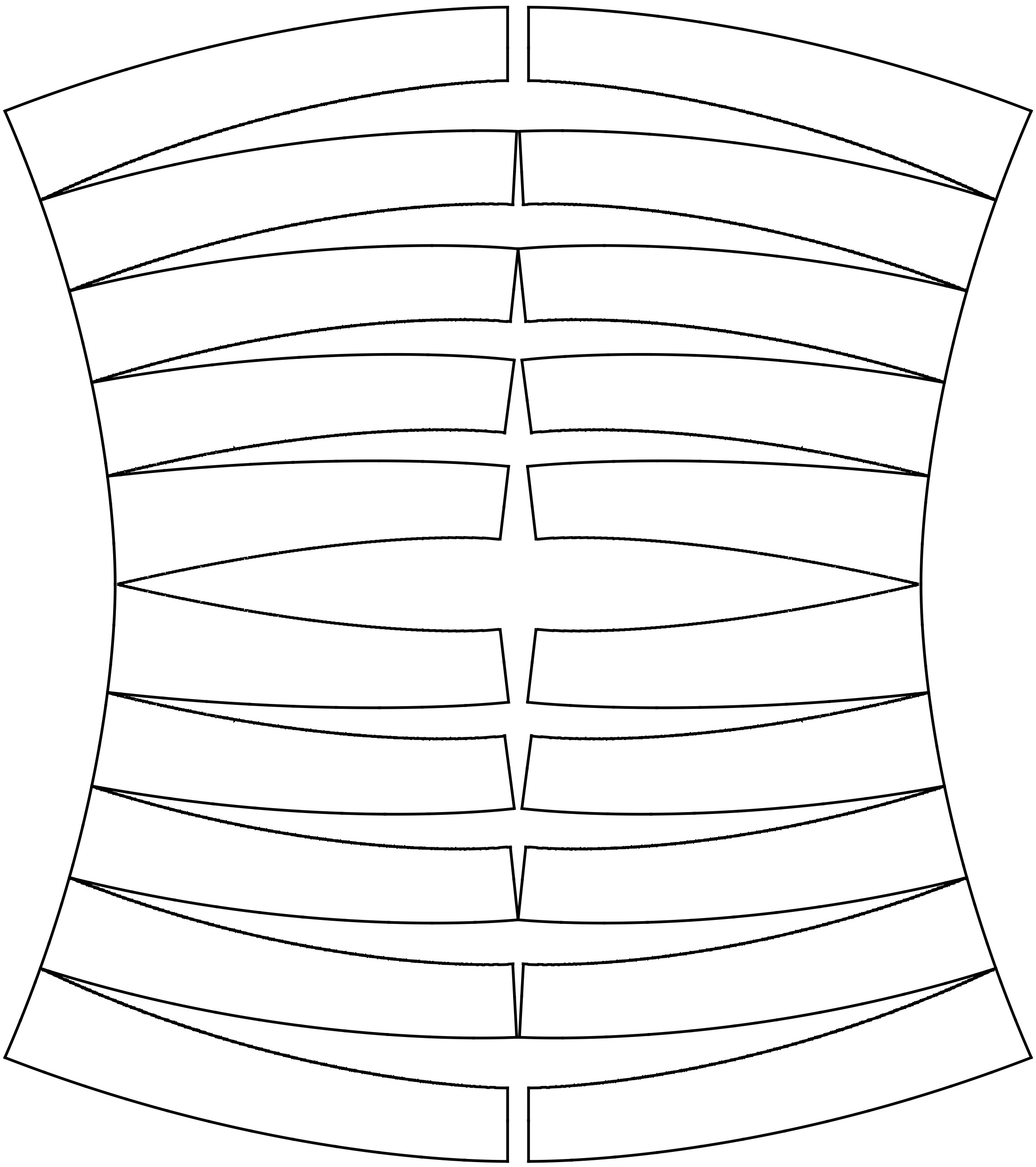
Show[

, 

, 

,

, PlotRange → All, AspectRatio → 1.2, Axes → None]



■ (c),(e)

Set up

```
kernelcodeCS = "
#define min 0
#define max 1
```

```
#define vvv 0
#define uuu 1
#define sign1 -1
#define h 0.001
#define n 1201
#define AlphaLAC (2.0)
#define BetaLASC (2.0)
#define InitRCLAC (1.0)
#define OmegaLASC (1.0)
#define Lambda_Max_LAC (12.718433357775211)
#define ArcLength_Max_LAC (6.764357890136049)
#define Lambda_Min_LAC (12.718433357775211)
#define ArcLength_Min_LAC (6.764357890136049)
#define Lambda_Max_LASC (6.285247558593747)
#define InitRT_Max_LASC (7.303131835937503)
#define ArcLength_Max_LASC (3.1084204465150833)
#define Lambda_Min_LASC (6.285247558593747)
#define InitRT_Min_LASC (7.303131835937503)
#define ArcLength_Min_LASC (3.1084204465150833)
#define Point00x (0.0)
#define Point00y (0.0)
#define Point00z (0.0)
#define Point03x (-0.7077264466498845)
#define Point03y (0.0)
#define Point03z (0.5008767232876717)
#define Point30x (0.0)
#define Point30y (-0.7077264466498845)
#define Point30z (-0.5008767232876717)
#define Point33x (-1.0013149889663)
#define Point33y (-1.0013149889663)
#define Point33z (0.0)

#define maxPoint00x (0.0)
#define maxPoint00y (0.0)
#define maxPoint00z (0.0)
#define maxPoint03x (-0.7077264466498845)
#define maxPoint03y (0.0)
#define maxPoint03z (0.5008767232876717)
#define maxPoint30x (0.0)
#define maxPoint30y (-0.7077264466498845)
#define maxPoint30z (-0.5008767232876717)
#define maxPoint33x (-1.0013149889663)
#define maxPoint33y (-1.0013149889663)
#define maxPoint33z (0.0)


#define ScaleToOrigin_Max_LAC (0.13192365851313362)
```

```
#define ScaleToOrigin_Min_LAC (0.13192365851313362)
#define ScaleToOrigin_Max_LASC (0.38164320308766897)
#define ScaleToOrigin_Min_LASC (0.38164320308766897)

#define inidxdu1
    ((1-pv)*(maxPoint00x-maxPoint03x)+pv*(maxPoint30x-maxPoint33x)-v1_x1+
    v2_x1+(1-pv)*u1_t1+pv*u2_t1)
#define inidxdu2
    ((1-pv)*(maxPoint00y-maxPoint03y)+pv*(maxPoint30y-maxPoint33y)-v1_x2+
    v2_x2+(1-pv)*u1_t2+pv*u2_t2)
#define inidxdu3
    ((1-pv)*(maxPoint00z-maxPoint03z)+pv*(maxPoint30z-maxPoint33z)-v1_x3+
    v2_x3+(1-pv)*u1_t3+pv*u2_t3)
#define inidxdv1
    ((1-pu)*(maxPoint00x-maxPoint30x)+pu*(maxPoint03x-maxPoint33x)-u1_x1+
    u2_x1+(1-pu)*v1_t1+pu*v2_t1)
#define inidxdv2
    ((1-pu)*(maxPoint00y-maxPoint30y)+pu*(maxPoint03y-maxPoint33y)-u1_x2+
    u2_x2+(1-pu)*v1_t2+pu*v2_t2)
#define inidxdv3
    ((1-pu)*(maxPoint00z-maxPoint30z)+pu*(maxPoint03z-maxPoint33z)-u1_x3+
    u2_x3+(1-pu)*v1_t3+pu*v2_t3)

#define inid2xdu21 ((1-pv)*u1_dt1+pv*u2_dt1)
#define inid2xdu22 ((1-pv)*u1_dt2+pv*u2_dt2)
#define inid2xdu23 ((1-pv)*u1_dt3+pv*u2_dt3)
#define inid2xdv21 ((1-pu)*v1_dt1+pu*v2_dt1)
#define inid2xdv22 ((1-pu)*v1_dt2+pu*v2_dt2)
#define inid2xdv23 ((1-pu)*v1_dt3+pu*v2_dt3)
#define inid2xdudv1
    (-maxPoint00x+maxPoint03x+maxPoint30x-maxPoint33x-u1_t1+u2_t1-v1_t1+v2_t1)
#define inid2xdudv2
    (-maxPoint00y+maxPoint03y+maxPoint30y-maxPoint33y-u1_t2+u2_t2-v1_t2+v2_t2)
#define inid2xdudv3
    (-maxPoint00z+maxPoint03z+maxPoint30z-maxPoint33z-u1_t3+u2_t3-v1_t3+v2_t3)


#define dxdu1
    ((1-pv[i])*(maxPoint00x-maxPoint03x)+pv[i]*(maxPoint30x-maxPoint33x)-
    v1_x1[i]+v2_x1[i]+(1-pv[i])*u1_t1[i]+pv[i]*u2_t1[i])
#define dxdu2
    ((1-pv[i])*(maxPoint00y-maxPoint03y)+pv[i]*(maxPoint30y-maxPoint33y)-
    v1_x2[i]+v2_x2[i]+(1-pv[i])*u1_t2[i]+pv[i]*u2_t2[i])
#define dxdu3
    ((1-pv[i])*(maxPoint00z-maxPoint03z)+pv[i]*(maxPoint30z-maxPoint33z)-
    v1_x3[i]+v2_x3[i]+(1-pv[i])*u1_t3[i]+pv[i]*u2_t3[i])
#define dxdv1
```

```
    ((1-pu[i])*(maxPoint00x-maxPoint30x)+pu[i]*(maxPoint03x-maxPoint33x)-
    u1_x1[i]+u2_x1[i]+(1-pu[i])*v1_t1[i]+pu[i]*v2_t1[i])
#define dxdv2
    ((1-pu[i])*(maxPoint00y-maxPoint30y)+pu[i]*(maxPoint03y-maxPoint33y)-
    u1_x2[i]+u2_x2[i]+(1-pu[i])*v1_t2[i]+pu[i]*v2_t2[i])
#define dxdv3
    ((1-pu[i])*(maxPoint00z-maxPoint30z)+pu[i]*(maxPoint03z-maxPoint33z)-
    u1_x3[i]+u2_x3[i]+(1-pu[i])*v1_t3[i]+pu[i]*v2_t3[i])


#define d2xdu21 ((1-pv[i])*u1_dt1[i]+pv[i]*u2_dt1[i])
#define d2xdu22 ((1-pv[i])*u1_dt2[i]+pv[i]*u2_dt2[i])
#define d2xdu23 ((1-pv[i])*u1_dt3[i]+pv[i]*u2_dt3[i])
#define d2xdv21 ((1-pu[i])*v1_dt1[i]+pu[i]*v2_dt1[i])
#define d2xdv22 ((1-pu[i])*v1_dt2[i]+pu[i]*v2_dt2[i])
#define d2xdv23 ((1-pu[i])*v1_dt3[i]+pu[i]*v2_dt3[i])
#define d2xdudv1
    (-maxPoint00x+maxPoint03x+maxPoint30x-maxPoint33x-u1_t1[i]+u2_t1[i]-
    v1_t1[i]+v2_t1[i])
#define d2xdudv2
    (-maxPoint00y+maxPoint03y+maxPoint30y-maxPoint33y-u1_t2[i]+u2_t2[i]-
    v1_t2[i]+v2_t2[i])
#define d2xdudv3
    (-maxPoint00z+maxPoint03z+maxPoint30z-maxPoint33z-u1_t3[i]+u2_t3[i]-
    v1_t3[i]+v2_t3[i])


#define d3xdu31 ((1-pv[i])*u1_d2t1[i]+pv[i]*u2_d2t1[i])
#define d3xdu32 ((1-pv[i])*u1_d2t2[i]+pv[i]*u2_d2t2[i])
#define d3xdu33 ((1-pv[i])*u1_d2t3[i]+pv[i]*u2_d2t3[i])
#define d3xdv31 ((1-pu[i])*v1_d2t1[i]+pu[i]*v2_d2t1[i])
#define d3xdv32 ((1-pu[i])*v1_d2t2[i]+pu[i]*v2_d2t2[i])
#define d3xdv33 ((1-pu[i])*v1_d2t3[i]+pu[i]*v2_d2t3[i])
#define d3xdu2dv1 (-u1_dt1[i]+u2_dt1[i])
#define d3xdu2dv2 (-u1_dt2[i]+u2_dt2[i])
#define d3xdu2dv3 (-u1_dt3[i]+u2_dt3[i])
#define d3xdudv21 (-v1_dt1[i]+v2_dt1[i])
#define d3xdudv22 (-v1_dt2[i]+v2_dt2[i])
#define d3xdudv23 (-v1_dt3[i]+v2_dt3[i])

#define d4xdu41 ((1-pv[i])*u1_d3t1[i]+pv[i]*u2_d3t1[i])
#define d4xdu42 ((1-pv[i])*u1_d3t2[i]+pv[i]*u2_d3t2[i])
#define d4xdu43 ((1-pv[i])*u1_d3t3[i]+pv[i]*u2_d3t3[i])
#define d4xdv41 ((1-pu[i])*v1_d3t1[i]+pu[i]*v2_d3t1[i])
#define d4xdv42 ((1-pu[i])*v1_d3t2[i]+pu[i]*v2_d3t2[i])
#define d4xdv43 ((1-pu[i])*v1_d3t3[i]+pu[i]*v2_d3t3[i])
#define d4xdu3dv1 (-u1_d2t1[i]+u2_d2t1[i])
```

```
#define d4xdu3dv2 (-u1_d2t2[i]+u2_d2t2[i])
#define d4xdu3dv3 (-u1_d2t3[i]+u2_d2t3[i])
#define d4xdudv31 (-v1_d2t1[i]+v2_d2t1[i])
#define d4xdudv32 (-v1_d2t2[i]+v2_d2t2[i])
#define d4xdudv33 (-v1_d2t3[i]+v2_d2t3[i])
#define d4xdu2dv21 (0.0)
#define d4xdu2dv22 (0.0)
#define d4xdu2dv23 (0.0)

#define Coons_x1
    ((1-pv[index])*u1_x1[index]+pv[index]*u2_x1[index]+(1-pu[index])*v1_x1[
    index]+pu[index]*v2_x1[index]-((1-pu[index])*(1-pv[index])*maxPoint00x+
    pu[index]*(1-pv[index])*maxPoint03x+(1-pu[index])*pv[index]*maxPoint30x+
    pu[index]*pv[index]*maxPoint33x))
#define Coons_x2
    ((1-pv[index])*u1_x2[index]+pv[index]*u2_x2[index]+(1-pu[index])*v1_x2[
    index]+pu[index]*v2_x2[index]-((1-pu[index])*(1-pv[index])*maxPoint00y+
    pu[index]*(1-pv[index])*maxPoint03y+(1-pu[index])*pv[index]*maxPoint30y+
    pu[index]*pv[index]*maxPoint33y))
#define Coons_x3
    ((1-pv[index])*u1_x3[index]+pv[index]*u2_x3[index]+(1-pu[index])*v1_x3[
    index]+pu[index]*v2_x3[index]-((1-pu[index])*(1-pv[index])*maxPoint00z+
    pu[index]*(1-pv[index])*maxPoint03z+(1-pu[index])*pv[index]*maxPoint30z+
    pu[index]*pv[index]*maxPoint33z))

#define repiniSurNorvec
    iniSurNorvec(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3
    ,v2_t1,v2_t2,v2_t3,normv_C)
#define repinicoefE
    inicoefE(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3)
#define repinicoefF
    inicoefF(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3)
#define repinicoefG
    inicoefG(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3)
#define repinicoefL
    inicoefL(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repinicoefM
```

```
    inicoefM(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repinicoefN
    inicoefN(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniMeanCurva
    iniMeanCurva(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniGaussCurva
    iniGaussCurva(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniMaxPrinci
    iniMaxPrinci(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniMinPrinci
    iniMinPrinci(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repiniPrinci
    iniPrinci(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3
    ,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniEta
    iniEta(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniMiu
    iniMiu(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repiniDuDs
    iniDuDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
```

```
        v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
        v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
    #define repiniDvDs
        iniDvDs(pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
        u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
        v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
        v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)

    #define repSurNorvec
        SurNorvec(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
        u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
        v2_t1,v2_t2,v2_t3,normv_C)
    #define repDSurNvDu
        DSurNvDu(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
        u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
        v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
        v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,dnormvdu_C)
    #define repDSurNvDv
        DSurNvDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
        u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
        v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
        v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,dnormvdv_C)
    #define repD2SurNvDu2
        D2SurNvDu2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
        u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
        u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
        v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
        v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
        d2normvdu2_C)
    #define repD2SurNvDuDv
        D2SurNvDuDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
        u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
        u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
        v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
        v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
        d2normvdudv_C)
    #define repD2SurNvDv2
        D2SurNvDv2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
        u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
        u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
        v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
        v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
        d2normvdv2_C)
    #define repScalarSurNor
        ScalarSurNor(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
        u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,
        u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,
```

```
    v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,
    v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,
    Scal)
#define repcoefE
    coefE(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3)
#define repDEDu
    DEDu(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repDEDv
    DEDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repD2EDu2
    D2EDu2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repD2EDuDv
    D2EDuDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3
    ,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repD2EDv2
    D2EDv2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repcoefF
    coefF(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3)
#define repDFDu
    DFDu(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repDFDv
    DFDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
```

```
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repD2FDu2
    D2FDu2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repD2FDuDv
    D2FDuDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3
    ,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repD2FDv2
    D2FDv2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repcoefG
    coefG(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u2_x1,u2_x2,u2_x3,
    u2_t1,u2_t2,u2_t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3)
#define repDGDu
    DGDu(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repDGDv
    DGDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repD2GDu2
    D2GDu2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repD2GDuDv
    D2GDuDv(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3
    ,u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repD2GDv2
```

```
    D2GDv2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,
    u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,
    v1_t3,v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3)
#define repcoefL
    coefL(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repcoefM
    coefM(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repcoefN
    coefN(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repGaussCurva
    GaussCurva(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repMeanCurva
    MeanCurva(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repMaxPrinci
    MaxPrinci(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repMinPrinci
    MinPrinci(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3)
#define repPrinci
    Princi(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDcoefL_2
    DcoefL_2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
```

```
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,CoeffL)
#define repDcoefM_2
    DcoefM_2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,CoeffM)
#define repDcoefN_2
    DcoefN_2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,CoeffN)
#define repDGaussCurva_2
    DGaussCurva_2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,GCurva)
#define repDMeanCurva_2
    DMeanCurva_2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,MCurva)
#define repDPrinci_2
    DPrinci_2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,optimum,PCurva)
#define repEta
```

```
    Eta(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repMiu
    Miu(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDuDs
    DuDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDuDs1
    DuDs(i+1,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDvDs
    DvDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDvDs1
    DvDs(i+1,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDEDs
    DEDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDFDs
    DFDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDGDs
    DGDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDLDs
    DLDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
```

```
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repDMDs
    DMDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repDNDs
    DNDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repDPrinciDs
    DPrinciDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,optimum)
#define repDvDt
    DvDt(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDuDt
    DuDt(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum)
#define repDSurDs
    DSurDs(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,
    v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum,tangv)
#define repD2uDs2
    D2uDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
```

```
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repD2vDs2
    D2vDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repGeoCur
    GeoCur(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repCurvecurvature
    Curvecurvature(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2
    ,u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,
    u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,
    u2_d3t1,u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,
    v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,
    v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,
    v2_d2t3,v2_d3t1,v2_d3t2,v2_d3t3,optimum)
#define repD2EDs2
    D2EDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repD2FDs2
    D2FDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repD2GDs2
    D2GDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
```

```
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repD2LDs2
    D2LDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repD2MDs2
    D2MDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repD2NDs2
    D2NDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)
#define repD2PrinciDs2
    D2PrinciDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,optimum)
#define repD2SurDs2
    D2SurDs2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,optimum,dtangds)
#define repCurvenormaloS
```

```
    CurvenormaloS(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3
    ,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,
    u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3
    ,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1
    ,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,
    v2_d3t2,v2_d3t3,optimum,norv)
#define repCurvebinormaloS
    CurvebinormaloS(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,
    u1_dt2,u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,
    u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,
    u2_d2t3,u2_d3t1,u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,
    v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,
    v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2
    ,v2_d2t3,v2_d3t1,v2_d3t2,v2_d3t3,optimum,binorv)
#define repAlp2
    Alp2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum,Alp2_C)
#define repBeta2
    Beta2(i,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,
    u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,u2_x2,u2_x3,u2_t1,
    u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,u2_d2t3,u2_d3t1,u2_d3t2
    ,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,
    v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,v2_x1,v2_x2,v2_x3,v2_t1,
    v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2,v2_d2t3,v2_d3t1,v2_d3t2
    ,v2_d3t3,optimum)


#define opt1 (float pu,float pv,float u1_x1,float u1_x2,float u1_x3,float
    u1_t1,float u1_t2,float u1_t3,float u2_x1,float u2_x2,float u2_x3,float
    u2_t1,float u2_t2,float u2_t3,float v1_x1,float v1_x2,float
    v1_x3,float v1_t1,float v1_t2,float v1_t3,float v2_x1,float
    v2_x2,float v2_x3,float v2_t1,float v2_t2,float v2_t3,float normv_C[])
#define opt2 (float pu,float pv,float u1_x1,float u1_x2,float u1_x3,float
    u1_t1,float u1_t2,float u1_t3,float u2_x1,float u2_x2,float
    u2_x3,float u2_t1,float u2_t2,float u2_t3,float v1_x1,float
    v1_x2,float v1_x3,float v1_t1,float v1_t2,float v1_t3,float
    v2_x1,float v2_x2,float v2_x3,float v2_t1,float v2_t2,float v2_t3)
#define opt3 (float pu,float pv,float u1_x1,float u1_x2,float
    u1_x3,float u1_t1,float u1_t2,float u1_t3,float u1_dt1,float
    u1_dt2,float u1_dt3,float u2_x1,float u2_x2,float
    u2_x3,float u2_t1,float u2_t2,float u2_t3,float u2_dt1,float
```

```
    u2_dt2,float u2_dt3,float v1_x1,float v1_x2,float v1_x3,float
    v1_t1,float v1_t2,float v1_t3,float v1_dt1,float v1_dt2,float
    v1_dt3,float v2_x1,float v2_x2,float v2_x3,float v2_t1,float
    v2_t2,float v2_t3,float v2_dt1,float v2_dt2,float v2_dt3)
#define opt4 (float pu,float pv,float u1_x1,float u1_x2,float
    u1_x3,float u1_t1,float u1_t2,float u1_t3,float u1_dt1,float
    u1_dt2,float u1_dt3,float u2_x1,float u2_x2,float u2_x3,float
    u2_t1,float u2_t2,float u2_t3,float u2_dt1,float u2_dt2,float
    u2_dt3,float v1_x1,float v1_x2,float v1_x3,float v1_t1,float
    v1_t2,float v1_t3,float v1_dt1,float v1_dt2,float v1_dt3,float
    v2_x1,float v2_x2,float v2_x3,float v2_t1,float v2_t2,float
    v2_t3,float v2_dt1,float v2_dt2,float v2_dt3,mint optimum)
#define opt5 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u2_x1[],float u2_x2[],float u2_x3[],float u2_t1[],float u2_t2[],float
    u2_t3[],float v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float
    v1_t2[],float v1_t3[],float v2_x1[],float v2_x2[],float
    v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float normv_C[])
#define opt6 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u2_x1[],float u2_x2[],float u2_x3[],float u2_t1[],float u2_t2[],float
    u2_t3[],float v1_x1[],float v1_x2[],float v1_x3[],float
    v1_t1[],float v1_t2[],float v1_t3[],float v2_x1[],float
    v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[])
#define opt7 (int i,float pu[],float pv[],float u1_x1[],float u1_x2[],float
    u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float u1_dt1[],float
    u1_dt2[],float u1_dt3[],float u2_x1[],float u2_x2[],float u2_x3[],float
    u2_t1[],float u2_t2[],float u2_t3[],float u2_dt1[],float u2_dt2[],float
    u2_dt3[],float v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float
    v1_t2[],float v1_t3[],float v1_dt1[],float v1_dt2[],float
    v1_dt3[],float v2_x1[],float v2_x2[],float v2_x3[],float v2_t1[],float
    v2_t2[],float v2_t3[],float v2_dt1[],float v2_dt2[],float v2_dt3[])
#define opt8 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u2_x1[],float
    u2_x2[],float u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float
    u2_dt1[],float u2_dt2[],float u2_dt3[],float v1_x1[],float
    v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float v1_t3[],float
    v1_dt1[],float v1_dt2[],float v1_dt3[],float v2_x1[],float
    v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float
    v2_t3[],float v2_dt1[],float v2_dt2[],float v2_dt3[],mint optimum)
#define opt9 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u2_x1[],float
    u2_x2[],float u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float
    u2_dt1[],float u2_dt2[],float u2_dt3[],float v1_x1[],float
    v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float v1_t3[],float
```

```
    v1_dt1[],float v1_dt2[],float v1_dt3[],float v2_x1[],float
    v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float
    v2_dt1[],float v2_dt2[],float v2_dt3[],mint optimum,float tangv_C[])
#define opt10 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u2_x1[],float
    u2_x2[],float u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float
    u2_dt1[],float u2_dt2[],float u2_dt3[],float v1_x1[],float
    v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float v1_t3[],float
    v1_dt1[],float v1_dt2[],float v1_dt3[],float v2_x1[],float
    v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float
    v2_dt1[],float v2_dt2[],float v2_dt3[],float dnormv_C[])
#define opt11 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u1_d2t1[],float
    u1_d2t2[],float u1_d2t3[],float u2_x1[],float u2_x2[],float
    u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float u2_dt1[],float
    u2_dt2[],float u2_dt3[],float u2_d2t1[],float u2_d2t2[],float
    u2_d2t3[],float v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float
    v1_t2[],float v1_t3[],float v1_dt1[],float v1_dt2[],float
    v1_dt3[],float v1_d2t1[],float v1_d2t2[],float v1_d2t3[],float
    v2_x1[],float v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float
    v2_t3[],float v2_dt1[],float v2_dt2[],float v2_dt3[],float
    v2_d2t1[],float v2_d2t2[],float v2_d2t3[],float d2normv_C[])
#define opt12 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u1_d2t1[],float
    u1_d2t2[],float u1_d2t3[],float u2_x1[],float u2_x2[],float
    u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float u2_dt1[],float
    u2_dt2[],float u2_dt3[],float u2_d2t1[],float u2_d2t2[],float
    u2_d2t3[],float v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float
    v1_t2[],float v1_t3[],float v1_dt1[],float v1_dt2[],float
    v1_dt3[],float v1_d2t1[],float v1_d2t2[],float v1_d2t3[],float
    v2_x1[],float v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float
    v2_t3[],float v2_dt1[],float v2_dt2[],float v2_dt3[],float
    v2_d2t1[],float v2_d2t2[],float v2_d2t3[],float Scal[])
#define opt13 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u1_d2t1[],float
    u1_d2t2[],float u1_d2t3[],float u2_x1[],float u2_x2[],float
    u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float u2_dt1[],float
    u2_dt2[],float u2_dt3[],float u2_d2t1[],float u2_d2t2[],float
    u2_d2t3[],float v1_x1[],float v1_x2[],float v1_x3[],float
    v1_t1[],float v1_t2[],float v1_t3[],float v1_dt1[],float
    v1_dt2[],float v1_dt3[],float v1_d2t1[],float v1_d2t2[],float
    v1_d2t3[],float v2_x1[],float v2_x2[],float v2_x3[],float v2_t1[],float
    v2_t2[],float v2_t3[],float v2_dt1[],float v2_dt2[],float
```

```
    v2_dt3[],float v2_d2t1[],float v2_d2t2[],float v2_d2t3[])
#define opt14 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u1_d2t1[],float
    u1_d2t2[],float u1_d2t3[],float u1_d3t1[],float u1_d3t2[],float
    u1_d3t3[],float u2_x1[],float u2_x2[],float u2_x3[],float
    u2_t1[],float u2_t2[],float u2_t3[],float u2_dt1[],float
    u2_dt2[],float u2_dt3[],float u2_d2t1[],float u2_d2t2[],float
    u2_d2t3[],float u2_d3t1[],float u2_d3t2[],float u2_d3t3[],float
    v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float
    v1_t3[],float v1_dt1[],float v1_dt2[],float v1_dt3[],float
    v1_d2t1[],float v1_d2t2[],float v1_d2t3[],float v1_d3t1[],float
    v1_d3t2[],float v1_d3t3[],float v2_x1[],float v2_x2[],float
    v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float v2_dt1[],float
    v2_dt2[],float v2_dt3[],float v2_d2t1[],float v2_d2t2[],float
    v2_d2t3[],float v2_d3t1[],float v2_d3t2[],float v2_d3t3[],float coeff[])
#define opt15 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u1_d2t1[],float
    u1_d2t2[],float u1_d2t3[],float u1_d3t1[],float u1_d3t2[],float
    u1_d3t3[],float u2_x1[],float u2_x2[],float u2_x3[],float u2_t1[],float
    u2_t2[],float u2_t3[],float u2_dt1[],float u2_dt2[],float
    u2_dt3[],float u2_d2t1[],float u2_d2t2[],float u2_d2t3[],float
    u2_d3t1[],float u2_d3t2[],float u2_d3t3[],float v1_x1[],float
    v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float v1_t3[],float
    v1_dt1[],float v1_dt2[],float v1_dt3[],float v1_d2t1[],float
    v1_d2t2[],float v1_d2t3[],float v1_d3t1[],float v1_d3t2[],float
    v1_d3t3[],float v2_x1[],float v2_x2[],float v2_x3[],float v2_t1[],float
    v2_t2[],float v2_t3[],float v2_dt1[],float v2_dt2[],float
    v2_dt3[],float v2_d2t1[],float v2_d2t2[],float v2_d2t3[],float
    v2_d3t1[],float v2_d3t2[],float v2_d3t3[],mint optimum,float coeff[])
#define opt16 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u1_d2t1[],float
    u1_d2t2[],float u1_d2t3[],float u1_d3t1[],float u1_d3t2[],float
    u1_d3t3[],float u2_x1[],float u2_x2[],float u2_x3[],float
    u2_t1[],float u2_t2[],float u2_t3[],float u2_dt1[],float
    u2_dt2[],float u2_dt3[],float u2_d2t1[],float u2_d2t2[],float
    u2_d2t3[],float u2_d3t1[],float u2_d3t2[],float u2_d3t3[],float
    v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float
    v1_t3[],float v1_dt1[],float v1_dt2[],float v1_dt3[],float
    v1_d2t1[],float v1_d2t2[],float v1_d2t3[],float v1_d3t1[],float
    v1_d3t2[],float v1_d3t3[],float v2_x1[],float v2_x2[],float
    v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float v2_dt1[],float
    v2_dt2[],float v2_dt3[],float v2_d2t1[],float v2_d2t2[],float
    v2_d2t3[],float v2_d3t1[],float v2_d3t2[],float v2_d3t3[],mint optimum)
#define opt17 (int i,float pu[],float pv[],float u1_x1[],float
```

```
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u2_x1[],float
    u2_x2[],float u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float
    u2_dt1[],float u2_dt2[],float u2_dt3[],float v1_x1[],float
    v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float v1_t3[],float
    v1_dt1[],float v1_dt2[],float v1_dt3[],float v2_x1[],float
    v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float
    v2_dt1[],float v2_dt2[],float v2_dt3[],mint optimum,float Alph_C[])
#define opt18 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u2_x1[],float u2_x2[],float u2_x3[],float u2_t1[],float u2_t2[],float
    u2_t3[],float v1_x1[],float v1_x2[],float v1_x3[],float v1_t1[],float
    v1_t2[],float v1_t3[],float v2_x1[],float v2_x2[],float
    v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float Snor_C[])
#define opt19 (int i,float pu[],float pv[],float u1_x1[],float
    u1_x2[],float u1_x3[],float u1_t1[],float u1_t2[],float u1_t3[],float
    u1_dt1[],float u1_dt2[],float u1_dt3[],float u2_x1[],float
    u2_x2[],float u2_x3[],float u2_t1[],float u2_t2[],float u2_t3[],float
    u2_dt1[],float u2_dt2[],float u2_dt3[],float v1_x1[],float
    v1_x2[],float v1_x3[],float v1_t1[],float v1_t2[],float v1_t3[],float
    v1_dt1[],float v1_dt2[],float v1_dt3[],float v2_x1[],float
    v2_x2[],float v2_x3[],float v2_t1[],float v2_t2[],float v2_t3[],float
    v2_dt1[],float v2_dt2[],float v2_dt3[],mint optimum,float binormv_C[])

__device__ float inicurvature_LAC(float
    t,float Alpha,float Lambda,float InitRC,float Total_s)
{
if(Alpha==-1)
    {return ((float) 1)/InitRC-Lambda*t*Total_s;}
else if(Alpha==0)
    {return ((float) 1)/exp(Lambda*t*Total_s+log(InitRC));}
else if(Alpha==1)
    {return ((float) 1)/(InitRC+Lambda*t*Total_s);}
else if(Alpha==2)
    {return ((float)
    1)/sqrt(pow(InitRC,2)+((float) 2)*Lambda*t*Total_s);}
else
    {return pow(pow(InitRC,Alpha)+Lambda*Alpha*t*Total_s,-((float) 1)/Alpha);}
}

__device__ float inidcLACdx(float
    t,float Alpha,float Lambda,float InitRC,float Total_s)
{
if(Alpha==-1)
    {return -Total_s*Lambda;}
else if(Alpha==0)
    {return -Total_s*(Lambda*exp(-Lambda*t*Total_s))/InitRC;}
```

```
else if(Alpha==1)
    {return -Total_s*Lambda/pow(InitRC+Lambda*t*Total_s,2);}
else if(Alpha==2)
    {return -Total_s*Lambda/sqrt(pow(pow(InitRC,2)+((float)
    2)*Lambda*t*Total_s,3));}
else
    {return
    -Total_s*Lambda*pow(pow(InitRC,Alpha)+Lambda*Alpha*t*Total_s,-((float)
    1)-((float) 1)/Alpha);}
}
__device__ float inid2cLACdx2(float
    t,float Alpha,float Lambda,float InitRC,float Total_s)
{
if(Alpha==-1)
    {return 0;}
else if(Alpha==0)
    {return
    pow(Total_s,2)*(pow(Lambda,2)*exp(-Lambda*t*Total_s))/InitRC;}
else if(Alpha==1)
    {return pow(Total_s,2)*((float)
    2)*pow(Lambda,2)/pow(InitRC+Lambda*t*Total_s,3);}
else if(Alpha==2)
    {return pow(Total_s,2)*(((float)
    3)*pow(Lambda,2))/sqrt(pow(pow(InitRC,2)+((float)
    2)*Lambda*t*Total_s,5));}
else
    {return (((float)
    1)+Alpha)*pow(Total_s,2)*pow(Lambda,2)*pow(pow(InitRC,Alpha)+Lambda*
    Alpha*t*Total_s,-((float) 2)-((float) 1)/Alpha);}
}

__device__ float iniTorsion_LAC(float
    t,float Beta,float Omega,float InitRT,float Total_s)
{
if(Beta==-1)
    {return ((float) 1)/InitRT-Omega*t*Total_s;}
else if(Beta==0)
    {return ((float) 1)/exp(Omega*t*Total_s+log(InitRT));}
else if(Beta==1)
    {return ((float) 1)/(InitRT+Omega*t*Total_s);}
else if(Beta==2)
    {return ((float)
    1)/sqrt(pow(InitRT,2)+((float) 2)*Omega*t*Total_s);}
else
    {return pow(pow(InitRT,Beta)+Omega*Beta*t*Total_s,-((float) 1)/Beta);}
}
```

```
__device__ float inidtLACdx(float
    t,float Beta,float Omega,float InitRT,float Total_s)
{
if(Beta==-1)
    {return -Total_s*Omega;}
else if(Beta==0)
    {return -Total_s*(Omega*exp(-Omega*t*Total_s))/InitRT;}
else if(Beta==1)
    {return -Total_s*Omega/pow(InitRT+Omega*t*Total_s,2);}
else if(Beta==2)
    {return
    -Total_s*Omega/sqrt(pow(pow(InitRT,2)+((float) 2)*Omega*t*Total_s,3));}
else
    {return -Total_s*Omega*pow(pow(InitRT,Beta)+Omega*Beta*t*Total_s,-((float)
    1)-((float) 1)/Beta);}
}


__device__ void iniSurNorvec opt1
{
    normv_C[0] =inidxdu2*inidxdv3-inidxdu3*inidxdv2;
    normv_C[1] =inidxdu3*inidxdv1-inidxdu1*inidxdv3;
    normv_C[2] =inidxdu1*inidxdv2-inidxdu2*inidxdv1;
}
__device__ float inicoefE opt2
{
    return inidxdu1*inidxdu1+inidxdu2*inidxdu2+inidxdu3*inidxdu3;
}
__device__ float inicoefF opt2
{
    return (inidxdu1*inidxdv1+inidxdu2*inidxdv2+inidxdu3*inidxdv3);
}

__device__ float inicoefG opt2
{
    return inidxdv1*inidxdv1+inidxdv2*inidxdv2+inidxdv3*inidxdv3;
}
__device__ float inicoefL opt3
{
float normv_C[3];
repiniSurNorvec;

    return
    (normv_C[0]*inid2xdu21+normv_C[1]*inid2xdu22+normv_C[2]*inid2xdu23)/sqrt
    (normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float inicoefM opt3
```

```
{
float normv_C[3];
repiniSurNorvec;

    return
     (normv_C[0]*inid2xdudv1+normv_C[1]*inid2xdudv2+normv_C[2]*inid2xdudv3)/
     sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float inicoefN opt3
{
float normv_C[3];
repiniSurNorvec;

    return
     (normv_C[0]*inid2xdv21+normv_C[1]*inid2xdv22+normv_C[2]*inid2xdv23)/sqrt
     (normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float iniGaussCurva opt3
{
    return
     (repinicoefL*repinicoefN-pow(repinicoefM,2))/(repinicoefE*repinicoefG-
     pow(repinicoefF,2));
}
__device__ float iniMeanCurva opt3
{
    return
     0.5*(repinicoefE*repinicoefN+repinicoefG*repinicoefL-2*repinicoefF*
     repinicoefM)/(repinicoefE*repinicoefG-pow(repinicoefF,2));
}
__device__ float iniMaxPrinci opt3
{
    return repiniMeanCurva+sqrt(pow(repiniMeanCurva,2)-repiniGaussCurva);
}

__device__ float iniMinPrinci opt3
{
    return repiniMeanCurva-sqrt(pow(repiniMeanCurva,2)-repiniGaussCurva);
}
__device__ float iniPrinci opt4
{
    if(optimum==max)
    {return repiniMaxPrinci;}
    else if(optimum==min)
    {return repiniMinPrinci;}
    else
    {return 0;}
}
```

```
__device__ float iniEta opt4
{
    return
     sign1/sqrt(repinicoefE*pow(repinicoefM-repiniPrinci*repinicoefF,2)-2*
     repinicoefF*(repinicoefM-repiniPrinci*repinicoefF)*(repinicoefL-
     repiniPrinci*repinicoefE)+repinicoefG*pow(repinicoefL-repiniPrinci*
     repinicoefE,2));
}
__device__ float iniMiu opt4
{
    return
     sign1/sqrt(repinicoefE*pow(repinicoefN-repiniPrinci*repinicoefG,2)-2*
     repinicoefF*(repinicoefM-repiniPrinci*repinicoefF)*(repinicoefN-
     repiniPrinci*repinicoefG)+repinicoefG*pow(repinicoefM-repiniPrinci*
     repinicoefF,2));
}

__device__ float iniDuDs opt4
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-
     repiniPrinci*repinicoefG))
    {return repiniEta*(repinicoefM-repiniPrinci*repinicoefF);}
else
    {return repiniMiu*(repinicoefN-repiniPrinci*repinicoefG);}
}
__device__ float iniDvDs opt4
{
if(abs(repinicoefL-repiniPrinci*repinicoefE)>=abs(repinicoefN-repiniPrinci*
     repinicoefG))
    {return -repiniEta*(repinicoefL-repiniPrinci*repinicoefE);}
else
    {return -repiniMiu*(repinicoefM-repiniPrinci*repinicoefF);}
}

__device__ void SurNorvec opt5
{
    normv_C[0] =dxdu2*dxdv3-dxdu3*dxdv2;
    normv_C[1] =dxdu3*dxdv1-dxdu1*dxdv3;
    normv_C[2] =dxdu1*dxdv2-dxdu2*dxdv1;
}
__device__ void DSurNvDu opt10
{
    dnormv_C[0]
     =(d2xdu22*dxdv3-d2xdu23*dxdv2)+(dxdu2*d2xdudv3-dxdu3*d2xdudv2);
    dnormv_C[1]
     =(d2xdu23*dxdv1-d2xdu21*dxdv3)+(dxdu3*d2xdudv1-dxdu1*d2xdudv3);
    dnormv_C[2]
```

```
    =(d2xdu21*dxdv2-d2xdu22*dxdv1)+(dxdu1*d2xdudv2-dxdu2*d2xdudv1);

}
__device__ void DSurNvDv opt10
{
    dnormv_C[0]
    =(d2xdudv2*dxdv3-d2xdudv3*dxdv2)+(dxdu2*d2xdv23-dxdu3*d2xdv22);
    dnormv_C[1]
    =(d2xdudv3*dxdv1-d2xdudv1*dxdv3)+(dxdu3*d2xdv21-dxdu1*d2xdv23);

    dnormv_C[2]
    =(d2xdudv1*dxdv2-d2xdudv2*dxdv1)+(dxdu1*d2xdv22-dxdu2*d2xdv21);

}
__device__ void D2SurNvDu2 opt11
{
    d2normv_C[0]
    =(d3xdu32*dxdv3-d3xdu33*dxdv2)+2*(d2xdu22*d2xdudv3-d2xdu23*d2xdudv2)+(
    dxdu2*d3xdu2dv3-dxdu3*d3xdu2dv2);
    d2normv_C[1]
    =(d3xdu33*dxdv1-d3xdu31*dxdv3)+2*(d2xdu23*d2xdudv1-d2xdu21*d2xdudv3)+(
    dxdu3*d3xdu2dv1-dxdu1*d3xdu2dv3);
    d2normv_C[2]
    =(d3xdu31*dxdv2-d3xdu32*dxdv1)+2*(d2xdu21*d2xdudv2-d2xdu22*d2xdudv1)+(
    dxdu1*d3xdu2dv2-dxdu2*d3xdu2dv1);
}
__device__ void D2SurNvDuDv opt11
{
    d2normv_C[0]
    =(d3xdu2dv2*dxdv3-d3xdu2dv3*dxdv2)+(d2xdu22*d2xdv23-d2xdu23*d2xdv22)+(
    dxdu2*d3xdudv23-dxdu3*d3xdudv22);
    d2normv_C[1]
    =(d3xdu2dv3*dxdv1-d3xdu2dv1*dxdv3)+(d2xdu23*d2xdv21-d2xdu21*d2xdv23)+(
    dxdu3*d3xdudv21-dxdu1*d3xdudv23);
    d2normv_C[2]
    =(d3xdu2dv1*dxdv2-d3xdu2dv2*dxdv1)+(d2xdu21*d2xdv22-d2xdu22*d2xdv21)+(
    dxdu1*d3xdudv22-dxdu2*d3xdudv21);
}
__device__ void D2SurNvDv2 opt11
{
    d2normv_C[0]
    =(d3xdudv22*dxdv3-d3xdudv23*dxdv2)+2*(d2xdudv2*d2xdv23-d2xdudv3*d2xdv22)
    +(dxdu2*d3xdv33-dxdu3*d3xdv32);
    d2normv_C[1]
    =(d3xdudv23*dxdv1-d3xdudv21*dxdv3)+2*(d2xdudv3*d2xdv21-d2xdudv1*d2xdv23)
    +(dxdu3*d3xdv31-dxdu1*d3xdv33);
    d2normv_C[2]
```

```
     =(d3xdudv21*dxdv2-d3xdudv22*dxdv1)+2*(d2xdudv1*d2xdv22-d2xdudv2*d2xdv21)
     +(dxdu1*d3xdv32-dxdu2*d3xdv31);
}
__device__ void ScalarSurNor opt12
{
float
     ScalSNor,DScalSNorDu,DScalSNorDv,D2ScalSNorDu2,D2ScalSNorDuDv,D2ScalSNorDv2;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float d2normvdu2_C[3];
float d2normvdudv_C[3];
float d2normvdv2_C[3];
repSurNorvec;
repDSurNvDu;
repDSurNvDv;
repD2SurNvDu2;
repD2SurNvDuDv;
repD2SurNvDv2;

ScalSNor =
     sqrt(normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
DScalSNorDu
     =
     (normv_C[0]*dnormvdu_C[0]+normv_C[1]*dnormvdu_C[1]+normv_C[2]*dnormvdu_C
     [2])/ScalSNor;
DScalSNorDv =
     (normv_C[0]*dnormvdv_C[0]+normv_C[1]*dnormvdv_C[1]+normv_C[2]*dnormvdv_C
     [2])/ScalSNor;
D2ScalSNorDu2 =
     ((normv_C[0]*d2normvdu2_C[0]+normv_C[1]*d2normvdu2_C[1]+normv_C[2]*
     d2normvdu2_C[2])+(dnormvdu_C[0]*dnormvdu_C[0]+dnormvdu_C[1]*dnormvdu_C[1
     ]+dnormvdu_C[2]*dnormvdu_C[2])-pow(DScalSNorDu,2))/ScalSNor;
D2ScalSNorDuDv =
     ((normv_C[0]*d2normvdudv_C[0]+normv_C[1]*d2normvdudv_C[1]+normv_C[2]*
     d2normvdudv_C[2])+(dnormvdu_C[0]*dnormvdv_C[0]+dnormvdu_C[1]*dnormvdv_C[
     1]+dnormvdu_C[2]*dnormvdv_C[2])-DScalSNorDu*DScalSNorDv)/ScalSNor;
D2ScalSNorDv2 =
     ((normv_C[0]*d2normvdv2_C[0]+normv_C[1]*d2normvdv2_C[1]+normv_C[2]*
     d2normvdv2_C[2])+(dnormvdv_C[0]*dnormvdv_C[0]+dnormvdv_C[1]*dnormvdv_C[1
     ]+dnormvdv_C[2]*dnormvdv_C[2])-pow(DScalSNorDv,2))/ScalSNor;

Scal[0] = ScalSNor;
Scal[1] = DScalSNorDu;
Scal[2] = DScalSNorDv;
Scal[3] = D2ScalSNorDu2;
Scal[4] = D2ScalSNorDuDv;
```

```
Scal[5] = D2ScalSNorDv2;
}

__device__ float coefE opt6
{
    return dxdu1*dxdu1+dxdu2*dxdu2+dxdu3*dxdu3;
}
__device__ float DEDu opt7
{
    return 2*(dxdu1*d2xdu21+dxdu2*d2xdu22+dxdu3*d2xdu23);
}
__device__ float DEDv opt7
{
    return 2*(dxdu1*d2xdudv1+dxdu2*d2xdudv2+dxdu3*d2xdudv3);
}
__device__ float D2EDu2 opt13
{
    return
    2*((d2xdu21*d2xdu21+d2xdu22*d2xdu22+d2xdu23*d2xdu23)+(dxdu1*d3xdu31+
    dxdu2*d3xdu32+dxdu3*d3xdu33));
}
__device__ float D2EDuDv opt13
{
    return
    2*((d2xdudv1*d2xdu21+d2xdudv2*d2xdu22+d2xdudv3*d2xdu23)+(dxdu1*d3xdu2dv1
    +dxdu2*d3xdu2dv2+dxdu3*d3xdu2dv3));
}
__device__ float D2EDv2 opt13
{
    return
    2*((d2xdudv1*d2xdudv1+d2xdudv2*d2xdudv2+d2xdudv3*d2xdudv3)+(dxdu1*
    d3xdudv21+dxdu2*d3xdudv22+dxdu3*d3xdudv23));
}

__device__ float coefF opt6
{
    return (dxdu1*dxdv1+dxdu2*dxdv2+dxdu3*dxdv3);
}
__device__ float DFDu opt7
{
    return dxdv1*d2xdu21+dxdv2*d2xdu22+dxdv3*d2xdu23
    + dxdu1*d2xdudv1+dxdu2*d2xdudv2+dxdu3*d2xdudv3;
}
__device__ float DFDv opt7
{
    return dxdv1*d2xdudv1+dxdv2*d2xdudv2+dxdv3*d2xdudv3
    + dxdu1*d2xdv21+dxdu2*d2xdv22+dxdu3*d2xdv23;
```

```
}
__device__ float D2FDu2 opt13
{
    return
     (dxdv1*d3xdu31+dxdv2*d3xdu32+dxdv3*d3xdu33)+2*(d2xdu21*d2xdudv1+d2xdu22*
     d2xdudv2+d2xdu23*d2xdudv3)+(dxdu1*d3xdu2dv1+dxdu2*d3xdu2dv2+dxdu3*
     d3xdu2dv3);
}
__device__ float D2FDuDv opt13
{
    return
     (dxdv1*d3xdu2dv1+dxdv2*d3xdu2dv2+dxdv3*d3xdu2dv3)+(d2xdu21*d2xdv21+
     d2xdu22*d2xdv22+d2xdu23*d2xdv23)+(d2xdudv1*d2xdudv1+d2xdudv2*d2xdudv2+
     d2xdudv3*d2xdudv3)+(dxdu1*d3xdudv21+dxdu2*d3xdudv22+dxdu3*d3xdudv23);
}
__device__ float D2FDv2 opt13
{
    return
     (dxdv1*d3xdudv21+dxdv2*d3xdudv22+dxdv3*d3xdudv23)+2*(d2xdudv1*d2xdv21+
     d2xdudv2*d2xdv22+d2xdudv3*d2xdv23)+(dxdu1*d3xdv31+dxdu2*d3xdv32+dxdu3*
     d3xdv33);
}
__device__ float coefG opt6
{
    return dxdv1*dxdv1+dxdv2*dxdv2+dxdv3*dxdv3;
}
__device__ float DGDu opt7
{
    return 2*(dxdv1*d2xdudv1+dxdv2*d2xdudv2+dxdv3*d2xdudv3);
}
__device__ float DGDv opt7
{
    return 2*(dxdv1*d2xdv21+dxdv2*d2xdv22+dxdv3*d2xdv23);
}
__device__ float D2GDu2 opt13
{
    return
     2*((d2xdudv1*d2xdudv1+d2xdudv2*d2xdudv2+d2xdudv3*d2xdudv3)+(dxdv1*
     d3xdu2dv1+dxdv2*d3xdu2dv2+dxdv3*d3xdu2dv3));
}
__device__ float D2GDuDv opt13
{
    return
     2*((d2xdv21*d2xdudv1+d2xdv22*d2xdudv2+d2xdv23*d2xdudv3)+(dxdv1*d3xdudv21
     +dxdv2*d3xdudv22+dxdv3*d3xdudv23));
}
__device__ float D2GDv2 opt13
```

```
{
    return
    2*((d2xdv21*d2xdv21+d2xdv22*d2xdv22+d2xdv23*d2xdv23)+(dxdv1*d3xdv31+
    dxdv2*d3xdv32+dxdv3*d3xdv33));
}
__device__ float coefL opt7
{
float normv_C[3];
repSurNorvec;

    return
    (normv_C[0]*d2xdu21+normv_C[1]*d2xdu22+normv_C[2]*d2xdu23)/sqrt(normv_C[
    0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float coefM opt7
{
float normv_C[3];
repSurNorvec;

    return
    (normv_C[0]*d2xdudv1+normv_C[1]*d2xdudv2+normv_C[2]*d2xdudv3)/sqrt(
    normv_C[0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float coefN opt7
{
float normv_C[3];
repSurNorvec;

    return
    (normv_C[0]*d2xdv21+normv_C[1]*d2xdv22+normv_C[2]*d2xdv23)/sqrt(normv_C[
    0]*normv_C[0]+normv_C[1]*normv_C[1]+normv_C[2]*normv_C[2]);
}
__device__ float GaussCurva opt7
{
    return
    (repcoefL*repcoefN-pow(repcoefM,2))/(repcoefE*repcoefG-pow(repcoefF,2));

}
__device__ float MeanCurva opt7
{
    return
    0.5*(repcoefE*repcoefN+repcoefG*repcoefL-2*repcoefF*repcoefM)/(repcoefE*
    repcoefG-pow(repcoefF,2));
}
__device__ float MaxPrinci opt7
{
    return repMeanCurva+sqrt(pow(repMeanCurva,2)-repGaussCurva);
```

```
}

__device__ float MinPrinci opt7
{
    return repMeanCurva-sqrt(pow(repMeanCurva,2)-repGaussCurva);
}
__device__ float Princi opt8
{
    if(optimum==max)
    {return repMaxPrinci;}
    else if(optimum==min)
    {return repMinPrinci;}
    else
    {return 0;}
}
__device__ void DcoefL_2 opt14
{
float CoeL,DCoeLDu,DCoeLDv,D2CoeLDu2,D2CoeLDuDv,D2CoeLDv2;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float d2normvdu2_C[3];
float d2normvdudv_C[3];
float d2normvdv2_C[3];
float Scal[6];
repSurNorvec;
repDSurNvDu;
repDSurNvDv;
repD2SurNvDu2;
repD2SurNvDuDv;
repD2SurNvDv2;
repScalarSurNor;

CoeL = (normv_C[0]*d2xdu21+normv_C[1]*d2xdu22+normv_C[2]*d2xdu23)/Scal[0];
DCoeLDu =
     ((dnormvdu_C[0]*d2xdu21+dnormvdu_C[1]*d2xdu22+dnormvdu_C[2]*d2xdu23)+(
     normv_C[0]*d3xdu31+normv_C[1]*d3xdu32+normv_C[2]*d3xdu33)-Scal[1]*CoeL)/
     Scal[0];
DCoeLDv =
     ((dnormvdv_C[0]*d2xdu21+dnormvdv_C[1]*d2xdu22+dnormvdv_C[2]*d2xdu23)+(
     normv_C[0]*d3xdu2dv1+normv_C[1]*d3xdu2dv2+normv_C[2]*d3xdu2dv3)-Scal[2]*
     CoeL)/Scal[0];
D2CoeLDu2 =
     ((d2normvdu2_C[0]*d2xdu21+d2normvdu2_C[1]*d2xdu22+d2normvdu2_C[2]*
     d2xdu23)+2*(dnormvdu_C[0]*d3xdu31+dnormvdu_C[1]*d3xdu32+dnormvdu_C[2]*
     d3xdu33)+(normv_C[0]*d4xdu41+normv_C[1]*d4xdu42+normv_C[2]*d4xdu43)-Scal
     [3]*CoeL-2*Scal[1]*DCoeLDu)/Scal[0];
```

```
D2CoeLDuDv =
    ((d2normvdudv_C[0]*d2xdu21+d2normvdudv_C[1]*d2xdu22+d2normvdudv_C[2]*
    d2xdu23)+(dnormvdu_C[0]*d3xdu2dv1+dnormvdu_C[1]*d3xdu2dv2+dnormvdu_C[2]*
    d3xdu2dv3)+(dnormvdv_C[0]*d3xdu31+dnormvdv_C[1]*d3xdu32+dnormvdv_C[2]*
    d3xdu33)+(normv_C[0]*d4xdu3dv1+normv_C[1]*d4xdu3dv2+normv_C[2]*d4xdu3dv3
    )-Scal[4]*CoeL-Scal[1]*DCoeLDv-Scal[2]*DCoeLDu)/Scal[0];
D2CoeLDv2 =
    ((d2normvdv2_C[0]*d2xdu21+d2normvdv2_C[1]*d2xdu22+d2normvdv2_C[2]*
    d2xdu23)+2*(dnormvdv_C[0]*d3xdu2dv1+dnormvdv_C[1]*d3xdu2dv2+dnormvdv_C[2
    ]*d3xdu2dv3)+(normv_C[0]*d4xdu2dv21+normv_C[1]*d4xdu2dv22+normv_C[2]*
    d4xdu2dv23)-Scal[5]*CoeL-2*Scal[2]*DCoeLDv)/Scal[0];

coeff[0] = CoeL;
coeff[1] = DCoeLDu;
coeff[2] = DCoeLDv;
coeff[3] = D2CoeLDu2;
coeff[4] = D2CoeLDuDv;
coeff[5] = D2CoeLDv2;
}
__device__ void DcoefM_2 opt14
{
float CoeM,DCoeMDu,DCoeMDv,D2CoeMDu2,D2CoeMDuDv,D2CoeMDv2;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float d2normvdu2_C[3];
float d2normvdudv_C[3];
float d2normvdv2_C[3];
float Scal[6];
repSurNorvec;
repDSurNvDu;
repDSurNvDv;
repD2SurNvDu2;
repD2SurNvDuDv;
repD2SurNvDv2;
repScalarSurNor;

CoeM = (normv_C[0]*d2xdudv1+normv_C[1]*d2xdudv2+normv_C[2]*d2xdudv3)/Scal[0];
DCoeMDu =
    ((dnormvdu_C[0]*d2xdudv1+dnormvdu_C[1]*d2xdudv2+dnormvdu_C[2]*d2xdudv3)+
    (normv_C[0]*d3xdu2dv1+normv_C[1]*d3xdu2dv2+normv_C[2]*d3xdu2dv3)-Scal[1]
    *CoeM)/Scal[0];
DCoeMDv =
    ((dnormvdv_C[0]*d2xdudv1+dnormvdv_C[1]*d2xdudv2+dnormvdv_C[2]*d2xdudv3)+
    (normv_C[0]*d3xdudv21+normv_C[1]*d3xdudv22+normv_C[2]*d3xdudv23)-Scal[2]
    *CoeM)/Scal[0];
D2CoeMDu2 =
```

```
     ((d2normvdu2_C[0]*d2xdudv1+d2normvdu2_C[1]*d2xdudv2+d2normvdu2_C[2]*
     d2xdudv3)+2*(dnormvdu_C[0]*d3xdu2dv1+dnormvdu_C[1]*d3xdu2dv2+dnormvdu_C[
     2]*d3xdu2dv3)+(normv_C[0]*d4xdu3dv1+normv_C[1]*d4xdu3dv2+normv_C[2]*
     d4xdu3dv3)-Scal[3]*CoeM-2*Scal[1]*DCoeMDu)/Scal[0];
D2CoeMDuDv =
     ((d2normvdudv_C[0]*d2xdudv1+d2normvdudv_C[1]*d2xdudv2+d2normvdudv_C[2]*
     d2xdudv3)+(dnormvdu_C[0]*d3xdudv21+dnormvdu_C[1]*d3xdudv22+dnormvdu_C[2]
     *d3xdudv23)+(dnormvdv_C[0]*d3xdu2dv1+dnormvdv_C[1]*d3xdu2dv2+dnormvdv_C[
     2]*d3xdu2dv3)+(normv_C[0]*d4xdu2dv21+normv_C[1]*d4xdu2dv22+normv_C[2]*
     d4xdu2dv23)-Scal[4]*CoeM-Scal[1]*DCoeMDv-Scal[2]*DCoeMDu)/Scal[0];
D2CoeMDv2 =
     ((d2normvdv2_C[0]*d2xdudv1+d2normvdv2_C[1]*d2xdudv2+d2normvdv2_C[2]*
     d2xdudv3)+2*(dnormvdv_C[0]*d3xdudv21+dnormvdv_C[1]*d3xdudv22+dnormvdv_C[
     2]*d3xdudv23)+(normv_C[0]*d4xdudv31+normv_C[1]*d4xdudv32+normv_C[2]*
     d4xdudv33)-Scal[5]*CoeM-2*Scal[2]*DCoeMDv)/Scal[0];

coeff[0] = CoeM;
coeff[1] = DCoeMDu;
coeff[2] = DCoeMDv;
coeff[3] = D2CoeMDu2;
coeff[4] = D2CoeMDuDv;
coeff[5] = D2CoeMDv2;
}
__device__ void DcoefN_2 opt14
{
float CoeN,DCoeNDu,DCoeNDv,D2CoeNDu2,D2CoeNDuDv,D2CoeNDv2;
float normv_C[3];
float dnormvdu_C[3];
float dnormvdv_C[3];
float d2normvdu2_C[3];
float d2normvdudv_C[3];
float d2normvdv2_C[3];
float Scal[6];
repSurNorvec;
repDSurNvDu;
repDSurNvDv;
repD2SurNvDu2;
repD2SurNvDuDv;
repD2SurNvDv2;
repScalarSurNor;

CoeN = (normv_C[0]*d2xdv21+normv_C[1]*d2xdv22+normv_C[2]*d2xdv23)/Scal[0];
DCoeNDu =
     ((dnormvdu_C[0]*d2xdv21+dnormvdu_C[1]*d2xdv22+dnormvdu_C[2]*d2xdv23)+(
     normv_C[0]*d3xdudv21+normv_C[1]*d3xdudv22+normv_C[2]*d3xdudv23)-Scal[1]*
     CoeN)/Scal[0];
DCoeNDv =
```

```
        ((dnormvdv_C[0]*d2xdv21+dnormvdv_C[1]*d2xdv22+dnormvdv_C[2]*d2xdv23)+(
        normv_C[0]*d3xdv31+normv_C[1]*d3xdv32+normv_C[2]*d3xdv33)-Scal[2]*CoeN)/
        Scal[0];
    D2CoeNDu2 =
        ((d2normvdu2_C[0]*d2xdv21+d2normvdu2_C[1]*d2xdv22+d2normvdu2_C[2]*
        d2xdv23)+2*(dnormvdu_C[0]*d3xdudv21+dnormvdu_C[1]*d3xdudv22+dnormvdu_C[2
        ]*d3xdudv23)+(normv_C[0]*d4xdu2dv21+normv_C[1]*d4xdu2dv22+normv_C[2]*
        d4xdu2dv23)-Scal[3]*CoeN-2*Scal[1]*DCoeNDu)/Scal[0];
    D2CoeNDuDv =
        ((d2normvdudv_C[0]*d2xdv21+d2normvdudv_C[1]*d2xdv22+d2normvdudv_C[2]*
        d2xdv23)+(dnormvdu_C[0]*d3xdv31+dnormvdu_C[1]*d3xdv32+dnormvdu_C[2]*
        d3xdv33)+(dnormvdv_C[0]*d3xdudv21+dnormvdv_C[1]*d3xdudv22+dnormvdv_C[2]*
        d3xdudv23)+(normv_C[0]*d4xdudv31+normv_C[1]*d4xdudv32+normv_C[2]*
        d4xdudv33)-Scal[4]*CoeN-Scal[1]*DCoeNDv-Scal[2]*DCoeNDu)/Scal[0];
    D2CoeNDv2 =
        ((d2normvdv2_C[0]*d2xdv21+d2normvdv2_C[1]*d2xdv22+d2normvdv2_C[2]*
        d2xdv23)+2*(dnormvdv_C[0]*d3xdv31+dnormvdv_C[1]*d3xdv32+dnormvdv_C[2]*
        d3xdv33)+(normv_C[0]*d4xdv41+normv_C[1]*d4xdv42+normv_C[2]*d4xdv43)-Scal
        [5]*CoeN-2*Scal[2]*DCoeNDv)/Scal[0];

    coeff[0] = CoeN;
    coeff[1] = DCoeNDu;
    coeff[2] = DCoeNDv;
    coeff[3] = D2CoeNDu2;
    coeff[4] = D2CoeNDuDv;
    coeff[5] = D2CoeNDv2;
    }
    __device__ void DGaussCurva_2 opt14
    {
    float GCurv,DGCurvDu,DGCurvDv,D2GCurvDu2,D2GCurvDuDv,D2GCurvDv2;
    float CoeffL[6];
    float CoeffM[6];
    float CoeffN[6];
    repDcoefL_2;
    repDcoefM_2;
    repDcoefN_2;

    GCurv=
        (CoeffL[0]*CoeffN[0]-pow(CoeffM[0],2))/(repcoefE*repcoefG-pow(repcoefF,2));
    DGCurvDu
        =
        ((pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repcoefG*repDEDu-2*repcoefF*
        repDFDu+repcoefE*repDGDu)+(-pow(repcoefF,2)+repcoefE*repcoefG)*(CoeffN[0
        ]*CoeffL[1]-2*CoeffM[0]*CoeffM[1]+CoeffL[0]*CoeffN[1]))/(pow(pow(
        repcoefF,2)-repcoefE*repcoefG,2));
    DGCurvDv
        =((pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repcoefG*repDEDv-2*repcoefF*
```

```
        repDFDv+repcoefE*repDGDv)+(-pow(repcoefF,2)+repcoefE*repcoefG)*(CoeffN[0
        ]*CoeffL[2]-2*CoeffM[0]*CoeffM[2]+CoeffL[0]*CoeffN[2]))/(pow(pow(
        repcoefF,2)-repcoefE*repcoefG,2));
D2GCurvDu2 =
        ((-2*(pow(repcoefF,2)-repcoefE*repcoefG)*(repcoefG*repDEDu-2*repcoefF*
        repDFDu+repcoefE*repDGDu)*(CoeffN[0]*CoeffL[1]-2*CoeffM[0]*CoeffM[1]+
        CoeffL[0]*CoeffN[1])+(pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(2*pow(
        repcoefG*repDEDu-2*repcoefF*repDFDu+repcoefE*repDGDu,2)-(pow(repcoefF,2)
        -repcoefE*repcoefG)*(2*pow(repDFDu,2)-2*repDEDu*repDGDu-repcoefG*
        repD2EDu2+2*repcoefF*repD2FDu2-repcoefE*repD2GDu2))+pow(pow(repcoefF,2)-
        repcoefE*repcoefG,2)*(2*pow(CoeffM[1],2)-2*CoeffL[1]*CoeffN[1]-CoeffN[0]
        *CoeffL[3]+2*CoeffM[0]*CoeffM[3]-CoeffL[0]*CoeffN[3]))/pow(pow(repcoefF,
        2)-repcoefE*repcoefG,3));
D2GCurvDuDv =
        -((2*(pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(repcoefG*repDEDv-2*repcoefF
        *repDFDv+repcoefE*repDGDv)*(repcoefG*repDEDu-2*repcoefF*repDFDu+repcoefE
        *repDGDu)+(-pow(repcoefF,2)+repcoefE*repcoefG)*(CoeffN[0]*CoeffL[2]-2*
        CoeffM[0]*CoeffM[2]+CoeffL[0]*CoeffN[2])*(repcoefG*repDEDu-2*repcoefF*
        repDFDu+repcoefE*repDGDu)+(-pow(repcoefF,2)+repcoefE*repcoefG)*(repcoefG
        *repDEDv-2*repcoefF*repDFDv+repcoefE*repDGDv)*(CoeffN[0]*CoeffL[1]-2*
        CoeffM[0]*CoeffM[1]+CoeffL[0]*CoeffN[1])+(-pow(repcoefF,2)+repcoefE*
        repcoefG)*(-pow(CoeffM[0],2)+CoeffL[0]*CoeffN[0])*(repDGDv*repDEDu-2*
        repDFDv*repDFDu+repDEDv*repDGDu+repcoefG*repD2EDuDv-2*repcoefF*
        repD2FDuDv+repcoefE*repD2GDuDv)-pow(pow(repcoefF,2)-repcoefE*repcoefG,2)
        *(CoeffN[2]*CoeffL[1]-2*CoeffM[2]*CoeffM[1]+CoeffL[2]*CoeffN[1]+CoeffN[0
        ]*CoeffL[4]-2*CoeffM[0]*CoeffM[4]+CoeffL[0]*CoeffN[4]))/pow(-pow(
        repcoefF,2)+repcoefE*repcoefG,3));
D2GCurvDv2 =
        ((-2*(pow(repcoefF,2)-repcoefE*repcoefG)*(repcoefG*repDEDv-2*repcoefF*
        repDFDv+repcoefE*repDGDv)*(CoeffN[0]*CoeffL[2]-2*CoeffM[0]*CoeffM[2]+
        CoeffL[0]*CoeffN[2])+(pow(CoeffM[0],2)-CoeffL[0]*CoeffN[0])*(2*pow(
        repcoefG*repDEDv-2*repcoefF*repDFDv+repcoefE*repDGDv,2)-(pow(repcoefF,2)
        -repcoefE*repcoefG)*(2*pow(repDFDv,2)-2*repDEDv*repDGDv-repcoefG*
        repD2EDv2+2*repcoefF*repD2FDv2-repcoefE*repD2GDv2))+pow(pow(repcoefF,2)-
        repcoefE*repcoefG,2)*(2*pow(CoeffM[2],2)-2*CoeffL[2]*CoeffN[2]-CoeffN[0]
        *CoeffL[5]+2*CoeffM[0]*CoeffM[5]-CoeffL[0]*CoeffN[5]))/pow(pow(repcoefF,
        2)-repcoefE*repcoefG,3));

coeff[0] = GCurv;
coeff[1] = DGCurvDu;
coeff[2] = DGCurvDv;
coeff[3] = D2GCurvDu2;
coeff[4] = D2GCurvDuDv;
coeff[5] = D2GCurvDv2;
}
__device__ void DMeanCurva_2 opt14
{
```

```
float MCurv,DMCurvDu,DMCurvDv,D2MCurvDu2,D2MCurvDuDv,D2MCurvDv2;
float CoeffL[6];
float CoeffM[6];
float CoeffN[6];
repDcoefL_2;
repDcoefM_2;
repDcoefN_2;

MCurv =
    0.5*(repcoefE*CoeffN[0]+repcoefG*CoeffL[0]-2*repcoefF*CoeffM[0])/(
    repcoefE*repcoefG-pow(repcoefF,2));
DMCurvDu =
    0.5*(-(repcoefG*CoeffL[0]-2*repcoefF*CoeffM[0]+repcoefE*CoeffN[0])*(
    repcoefG*repDEDu-2*repcoefF*repDFDu+repcoefE*repDGDu)+(-pow(repcoefF,2)+
    repcoefE*repcoefG)*(CoeffN[0]*repDEDu-2*CoeffM[0]*repDFDu+CoeffL[0]*
    repDGDu+repcoefG*CoeffL[1]-2*repcoefF*CoeffM[1]+repcoefE*CoeffN[1]))/(
    pow(pow(repcoefF,2)-repcoefE*repcoefG,2));
DMCurvDv =
    0.5*(-(repcoefG*CoeffL[0]-2*repcoefF*CoeffM[0]+repcoefE*CoeffN[0])*(
    repcoefG*repDEDv-2*repcoefF*repDFDv+repcoefE*repDGDv)+(-pow(repcoefF,2)+
    repcoefE*repcoefG)*(CoeffN[0]*repDEDv-2*CoeffM[0]*repDFDv+CoeffL[0]*
    repDGDv+repcoefG*CoeffL[2]-2*repcoefF*CoeffM[2]+repcoefE*CoeffN[2]))/(
    pow(pow(repcoefF,2)-repcoefE*repcoefG,2));
D2MCurvDu2 =
    -((2*(pow(repcoefF,2)-repcoefE*repcoefG)*(repcoefG*repDEDu-2*repcoefF*
    repDFDu+repcoefE*repDGDu)*(CoeffN[0]*repDEDu-2*CoeffM[0]*repDFDu+CoeffL[
    0]*repDGDu+repcoefG*CoeffL[1]-2*repcoefF*CoeffM[1]+repcoefE*CoeffN[1])+(
    repcoefG*CoeffL[0]-2*repcoefF*CoeffM[0]+repcoefE*CoeffN[0])*(2*pow(
    repcoefG*repDEDu-2*repcoefF*repDFDu+repcoefE*repDGDu,2)-(pow(repcoefF,2)
    -repcoefE*repcoefG)*(2*pow(repDFDu,2)-2*repDEDu*repDGDu-repcoefG*
    repD2EDu2+2*repcoefF*repD2FDu2-repcoefE*repD2GDu2))+pow(pow(repcoefF,2)-
    repcoefE*repcoefG,2)*(2*repDGDu*CoeffL[1]-4*repDFDu*CoeffM[1]+2*repDEDu*
    CoeffN[1]+CoeffN[0]*repD2EDu2-2*CoeffM[0]*repD2FDu2+CoeffL[0]*repD2GDu2+
    repcoefG*CoeffL[3]-2*repcoefF*CoeffM[3]+repcoefE*CoeffN[3]))/(2*pow(pow(
    repcoefF,2)-repcoefE*repcoefG,3)));
D2MCurvDuDv =
    -(0.5*(-2*(repcoefG*CoeffL[0]-2*repcoefF*CoeffM[0]+repcoefE*CoeffN[0])*(
    repcoefG*repDEDv-2*repcoefF*repDFDv+repcoefE*repDGDv)*(repcoefG*repDEDu-
    2*repcoefF*repDFDu+repcoefE*repDGDu)+(-pow(repcoefF,2)+repcoefE*repcoefG
    )*(CoeffN[0]*repDEDv-2*CoeffM[0]*repDFDv+CoeffL[0]*repDGDv+repcoefG*
    CoeffL[2]-2*repcoefF*CoeffM[2]+repcoefE*CoeffN[2])*(repcoefG*repDEDu-2*
    repcoefF*repDFDu+repcoefE*repDGDu)+(-pow(repcoefF,2)+repcoefE*repcoefG)*
    (repcoefG*repDEDv-2*repcoefF*repDFDv+repcoefE*repDGDv)*(CoeffN[0]*
    repDEDu-2*CoeffM[0]*repDFDu+CoeffL[0]*repDGDu+repcoefG*CoeffL[1]-2*
    repcoefF*CoeffM[1]+repcoefE*CoeffN[1])+(-pow(repcoefF,2)+repcoefE*
    repcoefG)*(repcoefG*CoeffL[0]-2*repcoefF*CoeffM[0]+repcoefE*CoeffN[0])*(
    repDGDv*repDEDu-2*repDFDv*repDFDu+repDEDv*repDGDu+repcoefG*repD2EDuDv-2*
```

```
     repcoefF*repD2FDuDv+repcoefE*repD2GDuDv)-pow(pow(repcoefF,2)-repcoefE*
     repcoefG,2)*(CoeffN[2]*repDEDu-2*CoeffM[2]*repDFDu+CoeffL[2]*repDGDu+
     repDGDv*CoeffL[1]-2*repDFDv*CoeffM[1]+repDEDv*CoeffN[1]+CoeffN[0]*
     repD2EDuDv-2*CoeffM[0]*repD2FDuDv+CoeffL[0]*repD2GDuDv+repcoefG*CoeffL[4
     ]-2*repcoefF*CoeffM[4]+repcoefE*CoeffN[4]))/(pow(-pow(repcoefF,2)+
     repcoefE*repcoefG,3)));
D2MCurvDv2 =
     -(0.5*(2*(pow(repcoefF,2)-repcoefE*repcoefG)*(repcoefG*repDEDv-2*
     repcoefF*repDFDv+repcoefE*repDGDv)*(CoeffN[0]*repDEDv-2*CoeffM[0]*
     repDFDv+CoeffL[0]*repDGDv+repcoefG*CoeffL[2]-2*repcoefF*CoeffM[2]+
     repcoefE*CoeffN[2])+(repcoefG*CoeffL[0]-2*repcoefF*CoeffM[0]+repcoefE*
     CoeffN[0])*(2*pow(repcoefG*repDEDv-2*repcoefF*repDFDv+repcoefE*repDGDv,2
     )-(pow(repcoefF,2)-repcoefE*repcoefG)*(2*pow(repDFDv,2)-2*repDEDv*
     repDGDv-repcoefG*repD2EDv2+2*repcoefF*repD2FDv2-repcoefE*repD2GDv2))+pow
     (pow(repcoefF,2)-repcoefE*repcoefG,2)*(2*repDGDv*CoeffL[2]-4*repDFDv*
     CoeffM[2]+2*repDEDv*CoeffN[2]+CoeffN[0]*repD2EDv2-2*CoeffM[0]*repD2FDv2+
     CoeffL[0]*repD2GDv2+repcoefG*CoeffL[5]-2*repcoefF*CoeffM[5]+repcoefE*
     CoeffN[5]))/(pow(pow(repcoefF,2)-repcoefE*repcoefG,3)));


coeff[0] = MCurv;
coeff[1] = DMCurvDu;
coeff[2] = DMCurvDv;
coeff[3] = D2MCurvDu2;
coeff[4] = D2MCurvDuDv;
coeff[5] = D2MCurvDv2;
}
__device__ void DPrinci_2 opt15
{
float Princip,DPrincipDu,DPrincipDv,D2PrincipDu2,D2PrincipDuDv,D2PrincipDv2;
float GCurva[6];
float MCurva[6];
repDGaussCurva_2;
repDMeanCurva_2;

if(optimum==max)
{Princip = MCurva[0]+sqrt(pow(MCurva[0],2)-GCurva[0]);
DPrincipDu =
     MCurva[1]+(2*MCurva[0]*MCurva[1]-GCurva[1])/(2*sqrt(pow(MCurva[0],2)-
     GCurva[0]));
DPrincipDv =
     MCurva[2]+(2*MCurva[0]*MCurva[2]-GCurva[2])/(2*sqrt(pow(MCurva[0],2)-
     GCurva[0]));
D2PrincipDu2 =
     -(pow(-2*MCurva[0]*MCurva[1]+GCurva[1],2))/(4*sqrt(pow(pow(MCurva[0],2)-
     GCurva[0],3)))+MCurva[3]+(2*pow(MCurva[1],2)+2*MCurva[0]*MCurva[3]-
     GCurva[3])/(2*sqrt(pow(MCurva[0],2)-GCurva[0]));
```

```
D2PrincipDuDv =
    -(((2*MCurva[0]*MCurva[2]-GCurva[2])*(2*MCurva[0]*MCurva[1]-GCurva[1]))/
    (4*sqrt(pow(pow(MCurva[0],2)-GCurva[0],3))))+MCurva[4]+(2*MCurva[2]*
    MCurva[1]+2*MCurva[0]*MCurva[4]-GCurva[4])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
D2PrincipDv2 =
    -(pow(-2*MCurva[0]*MCurva[2]+GCurva[2],2))/(4*sqrt(pow(pow(MCurva[0],2)-
    GCurva[0],3)))+MCurva[5]+(2*pow(MCurva[2],2)+2*MCurva[0]*MCurva[5]-
    GCurva[5])/(2*sqrt(pow(MCurva[0],2)-GCurva[0]));
}
else if(optimum==min)
{Princip = MCurva[0]-sqrt(pow(MCurva[0],2)-GCurva[0]);
DPrincipDu =
    MCurva[1]-(2*MCurva[0]*MCurva[1]-GCurva[1])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
DPrincipDv =
    MCurva[2]-(2*MCurva[0]*MCurva[2]-GCurva[2])/(2*sqrt(pow(MCurva[0],2)-
    GCurva[0]));
D2PrincipDu2 =
    (pow(-2*MCurva[0]*MCurva[1]+GCurva[1],2))/(4*sqrt(pow(pow(MCurva[0],2)-
    GCurva[0],3)))+MCurva[3]+(-2*(pow(MCurva[1],2)+MCurva[0]*MCurva[3])+
    GCurva[3])/(2*sqrt(pow(MCurva[0],2)-GCurva[0]));
D2PrincipDuDv =
    ((2*MCurva[0]*MCurva[2]-GCurva[2])*(2*MCurva[0]*MCurva[1]-GCurva[1]))/(4
    *sqrt(pow(pow(MCurva[0],2)-GCurva[0],3)))+MCurva[4]+(-2*MCurva[2]*MCurva
    [1]-2*MCurva[0]*MCurva[4]+GCurva[4])/(2*sqrt(pow(MCurva[0],2)-GCurva[0])
    );
D2PrincipDv2
    =(pow(-2*MCurva[0]*MCurva[2]+GCurva[2],2))/(4*sqrt(pow(pow(MCurva[0],2)-
    GCurva[0],3)))+MCurva[5]+(-2*(pow(MCurva[2],2)+MCurva[0]*MCurva[5])+
    GCurva[5])/(2*sqrt(pow(MCurva[0],2)-GCurva[0]));
}
else
{Princip = 0;
DPrincipDu = 0;
DPrincipDv = 0;
D2PrincipDu2 = 0;
D2PrincipDuDv = 0;
D2PrincipDv2 = 0;
}

coeff[0] = Princip;
coeff[1] = DPrincipDu;
coeff[2] = DPrincipDv;
coeff[3] = D2PrincipDu2;
coeff[4] = D2PrincipDuDv;
coeff[5] = D2PrincipDv2;
```

```
}

__device__ float Eta opt8
{
    return
    sign1/sqrt(repcoefE*pow(repcoefM-repPrinci*repcoefF,2)-2*repcoefF*(
    repcoefM-repPrinci*repcoefF)*(repcoefL-repPrinci*repcoefE)+repcoefG*pow(
    repcoefL-repPrinci*repcoefE,2));
}
__device__ float Miu opt8
{
    return
    sign1/sqrt(repcoefE*pow(repcoefN-repPrinci*repcoefG,2)-2*repcoefF*(
    repcoefM-repPrinci*repcoefF)*(repcoefN-repPrinci*repcoefG)+repcoefG*pow(
    repcoefM-repPrinci*repcoefF,2));
}
__device__ float DuDs opt8
{
if(abs(repcoefL-repPrinci*repcoefE)>=abs(repcoefN-repPrinci*repcoefG))
    {return repEta*(repcoefM-repPrinci*repcoefF);}
else
    {return repMiu*(repcoefN-repPrinci*repcoefG);}
}
__device__ float DvDs opt8
{
if(abs(repcoefL-repPrinci*repcoefE)>=abs(repcoefN-repPrinci*repcoefG))
    {return -repEta*(repcoefL-repPrinci*repcoefE);}
else
    {return -repMiu*(repcoefM-repPrinci*repcoefF);}
}
__device__ float DEDs opt8
{
    return repDEDu*repDuDs+repDEDv*repDvDs;
}
__device__ float DFDs opt8
{
    return repDFDu*repDuDs+repDFDv*repDvDs;
}
__device__ float DGDs opt8
{
    return repDGDu*repDuDs+repDGDv*repDvDs;
}
__device__ float DLDs opt16
{
float CoeffL[6];
repDcoefL_2;
```

```
    return CoeffL[1]*repDuDs+CoeffL[2]*repDvDs;
}
__device__ float DMDs opt16
{
float CoeffM[6];
repDcoefM_2;

    return CoeffM[1]*repDuDs+CoeffM[2]*repDvDs;
}
__device__ float DNDs opt16
{
float CoeffN[6];
repDcoefN_2;

    return CoeffN[1]*repDuDs+CoeffN[2]*repDvDs;
}
__device__ float DPrinciDs opt16
{
float PCurva[6];
repDPrinci_2;

    return PCurva[1]*repDuDs+PCurva[2]*repDvDs;
}
__device__ float DuDt opt8
{
if(abs(repcoefL-repPrinci*repcoefE)>=abs(repcoefN-repPrinci*repcoefG))
    {return (repcoefM-repPrinci*repcoefF);}
else
    {return (repcoefN-repPrinci*repcoefG);}
}
__device__ float DvDt opt8
{
if(abs(repcoefL-repPrinci*repcoefE)>=abs(repcoefN-repPrinci*repcoefG))
    {return (repcoefL-repPrinci*repcoefE);}
else
    {return (repcoefM-repPrinci*repcoefF);}
}
__device__ void DSurDs opt9
{
    tangv_C[0] =dxdu1*repDuDs+
                dxdv1*repDvDs;
    tangv_C[1] =dxdu2*repDuDs+
                dxdv2*repDvDs;
    tangv_C[2] =dxdu3*repDuDs+
                dxdv3*repDvDs;
}
```

```
__device__ float D2uDs2 opt16
{
    return (repDuDs1-repDuDs)/h;
}
__device__ float D2vDs2 opt16
{
    return (repDvDs1-repDvDs)/h;
}
__device__ float GeoCur opt16
{
    return
     (((2*repcoefE*repDFDu-repcoefE*repDEDv-repcoefF*repDEDu)/(2*(repcoefE*
     repcoefG-pow(repcoefF,2))))*pow(repDuDs,3)+(2*(repcoefE*repDGDu-repcoefF
     *repDEDv)/(2*(repcoefE*repcoefG-pow(repcoefF,2)))-(repcoefG*repDEDu-2*
     repcoefF*repDFDu+repcoefF*repDEDv)/(2*(repcoefE*repcoefG-pow(repcoefF,2)
     )))*pow(repDuDs,2)*repDvDs+((repcoefE*repDGDv-2*repcoefF*repDFDv+
     repcoefF*repDGDu)/(2*(repcoefE*repcoefG-pow(repcoefF,2)))-2*(repcoefG*
     repDEDv-repcoefF*repDGDu)/(2*(repcoefE*repcoefG-pow(repcoefF,2))))*
     repDuDs*pow(repDvDs,2)-((2*repcoefG*repDFDv-repcoefG*repDGDu-repcoefF*
     repDGDv)/(2*(repcoefE*repcoefG-pow(repcoefF,2))))*pow(repDvDs,3)+repDuDs
     *repD2vDs2-repD2uDs2*repDvDs)*sqrt(repcoefE*repcoefG-pow(repcoefF,2));
}

__device__ float Curvecurvature opt16
{
    return sqrt(pow(repPrinci,2)+pow(repGeoCur,2));
}
__device__ float D2EDs2 opt16
{
    return
     repD2EDu2*pow(repDuDs,2)+2*repD2EDuDv*repDuDs*repDvDs+repD2EDv2*pow(
     repDvDs,2)+repDEDu*repD2uDs2+repDEDv*repD2vDs2;
}
__device__ float D2FDs2 opt16
{
    return
     repD2FDu2*pow(repDuDs,2)+2*repD2FDuDv*repDuDs*repDvDs+repD2FDv2*pow(
     repDvDs,2)+repDFDu*repD2uDs2+repDFDv*repD2vDs2;
}
__device__ float D2GDs2 opt16
{
    return
     repD2GDu2*pow(repDuDs,2)+2*repD2GDuDv*repDuDs*repDvDs+repD2GDv2*pow(
     repDvDs,2)+repDGDu*repD2uDs2+repDGDv*repD2vDs2;
}
__device__ float D2LDs2 opt16
{
```

```
float CoeffL[6];
repDcoefL_2;

    return
    CoeffL[3]*pow(repDuDs,2)+2*CoeffL[4]*repDuDs*repDvDs+CoeffL[5]*pow(
    repDvDs,2)+CoeffL[1]*repD2uDs2+CoeffL[2]*repD2vDs2;
}

__device__ float D2MDs2 opt16
{
float CoeffM[6];
repDcoefM_2;

    return
    CoeffM[3]*pow(repDuDs,2)+2*CoeffM[4]*repDuDs*repDvDs+CoeffM[5]*pow(
    repDvDs,2)+CoeffM[1]*repD2uDs2+CoeffM[2]*repD2vDs2;
}
__device__ float D2NDs2 opt16
{
float CoeffN[6];
repDcoefN_2;

    return
    CoeffN[3]*pow(repDuDs,2)+2*CoeffN[4]*repDuDs*repDvDs+CoeffN[5]*pow(
    repDvDs,2)+CoeffN[1]*repD2uDs2+CoeffN[2]*repD2vDs2;
}
__device__ float D2PrinciDs2 opt16
{
float PCurva[6];
repDPrinci_2;

    return
    PCurva[3]*pow(repDuDs,2)+2*PCurva[4]*repDuDs*repDvDs+PCurva[5]*pow(
    repDvDs,2)+PCurva[1]*repD2uDs2+PCurva[2]*repD2vDs2;
}
__device__ void D2SurDs2 opt15
{
    coeff[0] =
    d2xdu21*pow(repDuDs,2)+2*d2xdudv1*repDuDs*repDvDs+d2xdv21*pow(repDvDs,2)
    +dxdu1*repD2uDs2+dxdv1*repD2vDs2;

    coeff[1] =
    d2xdu22*pow(repDuDs,2)+2*d2xdudv2*repDuDs*repDvDs+d2xdv22*pow(repDvDs,2)
    +dxdu2*repD2uDs2+dxdv2*repD2vDs2;

    coeff[2] =
    d2xdu23*pow(repDuDs,2)+2*d2xdudv3*repDuDs*repDvDs+d2xdv23*pow(repDvDs,2)
```

```
    +dxdu3*repD2uDs2+dxdv3*repD2vDs2;
}
__device__ void CurvenormaloS opt15
{
    float dtangds[3];
    repD2SurDs2;

    coeff[0] = dtangds[0]/repCurvecurvature;
    coeff[1] = dtangds[1]/repCurvecurvature;
    coeff[2] = dtangds[2]/repCurvecurvature;
}
__device__ void CurvebinormaloS opt15
{
    float tangv[3];
    float norv[3];
    repDSurDs;
    repCurvenormaloS;

    coeff[0] =tangv[1]*norv[2]-tangv[2]*norv[1];
    coeff[1] =tangv[2]*norv[0]-tangv[0]*norv[2];
    coeff[2] =tangv[0]*norv[1]-tangv[1]*norv[0];
}
__device__ void Alp2 opt15
{
    coeff[0] =
     (d3xdu31*pow(repDuDs,3)+3*d3xdu2dv1*pow(repDuDs,2)*repDvDs+3*d3xdudv21*
     repDuDs*pow(repDvDs,2)+d3xdv31*pow(repDvDs,3))+3*(d2xdu21*repDuDs*
     repD2uDs2+d2xdudv1*(repDvDs*repD2uDs2+repDuDs*repD2vDs2)+d2xdv21*repDvDs
     *repD2vDs2);

    coeff[1] =
     (d3xdu32*pow(repDuDs,3)+3*d3xdu2dv2*pow(repDuDs,2)*repDvDs+3*d3xdudv22*
     repDuDs*pow(repDvDs,2)+d3xdv32*pow(repDvDs,3))+3*(d2xdu22*repDuDs*
     repD2uDs2+d2xdudv2*(repDvDs*repD2uDs2+repDuDs*repD2vDs2)+d2xdv22*repDvDs
     *repD2vDs2);

    coeff[2] =
     (d3xdu33*pow(repDuDs,3)+3*d3xdu2dv3*pow(repDuDs,2)*repDvDs+3*d3xdudv23*
     repDuDs*pow(repDvDs,2)+d3xdv33*pow(repDvDs,3))+3*(d2xdu23*repDuDs*
     repD2uDs2+d2xdudv3*(repDvDs*repD2uDs2+repDuDs*repD2vDs2)+d2xdv23*repDvDs
     *repD2vDs2);
}

__device__ float Beta2 opt16
{
if(abs(repcoefL-repPrinci*repcoefE)>=abs(repcoefN-repPrinci*repcoefG))
{return
```

```
    -(2*(repDLDs-repDPrinciDs*repcoefE-repPrinci*repDEDs)*repD2uDs2+2*(
    repDMDs-repDPrinciDs*repcoefF-repPrinci*repDFDs)*repD2vDs2+(repD2LDs2-
    repD2PrinciDs2*repcoefE-2*repDPrinciDs*repDEDs-repPrinci*repD2EDs2)*
    repDuDs+(repD2MDs2-repD2PrinciDs2*repcoefF-2*repDPrinciDs*repDFDs-
    repPrinci*repD2FDs2)*repDvDs);}
else
{return
    -(2*(repDMDs-repDPrinciDs*repcoefF-repPrinci*repDFDs)*repD2uDs2+2*(
    repDNDs-repDPrinciDs*repcoefG-repPrinci*repDGDs)*repD2vDs2+(repD2MDs2-
    repD2PrinciDs2*repcoefF-2*repDPrinciDs*repDFDs-repPrinci*repD2FDs2)*
    repDuDs+(repD2NDs2-repD2PrinciDs2*repcoefG-2*repDPrinciDs*repDGDs-
    repPrinci*repD2GDs2)*repDvDs);}
}

__device__ float Torsion opt16
{
    int rc=4;
    float tangv[3];
    float norv[3];
    float binorv[3];
    float Alp2_C[3];
    repDSurDs;
    repCurvenormaloS;
    repCurvebinormaloS;
    repAlp2;
    float matrixA[4][4]={
    {repcoefE,repcoefF,-(norv[0]*dxdu1+norv[1]*dxdu2+norv[2]*dxdu3),-
    repCurvecurvature*(binorv[0]*dxdu1+binorv[1]*dxdu2+binorv[2]*dxdu3)},

    {repcoefF,repcoefG,-(norv[0]*dxdv1+norv[1]*dxdv2+norv[2]*dxdv3),-
    repCurvecurvature*(binorv[0]*dxdv1+binorv[1]*dxdv2+binorv[2]*dxdv3)},

    {(norv[0]*dxdu1+norv[1]*dxdu2+norv[2]*dxdu3),(norv[0]*dxdv1+norv[1]*
    dxdv2+norv[2]*dxdv3),-1,0},
    {repDvDt,repDuDt,0,0}};

    float
    matrixB[4]={-(Alp2_C[0]*dxdu1+Alp2_C[1]*dxdu2+Alp2_C[2]*dxdu3)-(pow(
    repCurvecurvature,2)*(tangv[0]*dxdu1+tangv[1]*dxdu2+tangv[2]*dxdu3)),-(
    Alp2_C[0]*dxdv1+Alp2_C[1]*dxdv2+Alp2_C[2]*dxdv3)-(pow(repCurvecurvature,
    2)*(tangv[0]*dxdv1+tangv[1]*dxdv2+tangv[2]*dxdv3)),-(Alp2_C[0]*norv[0]+
    Alp2_C[1]*norv[1]+Alp2_C[2]*norv[2]),repBeta2};
    float Lower[4][4];
    float Upper[4][4];
    float x[4];
    float y[4];
    float sum;
```

```
    for(int ii=0; ii<rc; ii++)
            {
                for(int jj=0; jj<rc; jj++)
                {
                    sum = 0;
                    if(ii==jj)
                    {
                    Lower[ii][jj]=1;
                    for (int kk = 0; kk < ii; kk++)
                        {sum += Lower[ii][kk] * Upper[kk][jj];}
                    Upper[ii][jj] = matrixA[ii][jj] - sum;
                    }
                    else if(ii < jj)
                    {
                    Lower[ii][jj]=0;
                    for (int kk = 0; kk < ii; kk++)
                        {sum += Lower[ii][kk] * Upper[kk][jj];}
                    Upper[ii][jj] = matrixA[ii][jj] - sum;
                    }
                    else
                    {
                    Upper[ii][jj]=0;
                    for (int kk = 0; kk < jj; kk++)
                        {sum += Lower[ii][kk] * Upper[kk][jj];}
                    Lower[ii][jj] =(matrixA[ii][jj] - sum)/Upper[jj][jj];
                    }
                }
            }
    for (int ii = 0; ii < rc; ii++)
    {
        sum = 0;
        for (int jj = 0; jj < ii; jj++)
            {sum += Lower[ii][jj] * y[jj];}
        y[ii] = matrixB[ii] - sum;
    }

    for (int ii = rc - 1; ii >= 0; ii--)
    {
        sum = 0;
        for (int jj = ii + 1; jj < rc; jj++)
            {sum += Upper[ii][jj] * x[jj];}
        x[ii] = (y[ii] - sum)/Upper[ii][ii];
    }
return x[3];
}
```

```
__device__ float DerivativeofCurvature opt16
{
    int rc=4;
    float tangv[3];
    float norv[3];
    float binorv[3];
    float Alp2_C[3];
    repDSurDs;
    repCurvenormaloS;
    repCurvebinormaloS;
    repAlp2;
    float matrixA[4][4]={
    {repcoefE,repcoefF,-(norv[0]*dxdu1+norv[1]*dxdu2+norv[2]*dxdu3),-
    repCurvecurvature*(binorv[0]*dxdu1+binorv[1]*dxdu2+binorv[2]*dxdu3)},

    {repcoefF,repcoefG,-(norv[0]*dxdv1+norv[1]*dxdv2+norv[2]*dxdv3),-
    repCurvecurvature*(binorv[0]*dxdv1+binorv[1]*dxdv2+binorv[2]*dxdv3)},

    {(norv[0]*dxdu1+norv[1]*dxdu2+norv[2]*dxdu3),(norv[0]*dxdv1+norv[1]*
    dxdv2+norv[2]*dxdv3),-1,0},
    {repDvDt,repDuDt,0,0}};

    float
    matrixB[4]={-(Alp2_C[0]*dxdu1+Alp2_C[1]*dxdu2+Alp2_C[2]*dxdu3)-(pow(
    repCurvecurvature,2)*(tangv[0]*dxdu1+tangv[1]*dxdu2+tangv[2]*dxdu3)),-(
    Alp2_C[0]*dxdv1+Alp2_C[1]*dxdv2+Alp2_C[2]*dxdv3)-(pow(repCurvecurvature,
    2)*(tangv[0]*dxdv1+tangv[1]*dxdv2+tangv[2]*dxdv3)),-(Alp2_C[0]*norv[0]+
    Alp2_C[1]*norv[1]+Alp2_C[2]*norv[2]),repBeta2};
    float Lower[4][4];
    float Upper[4][4];
    float x[4];
    float y[4];
    float sum;

    for(int ii=0; ii<rc; ii++)
            {
                for(int jj=0; jj<rc; jj++)
                {
                    sum = 0;
                    if(ii==jj)
                    {
                    Lower[ii][jj]=1;
                    for (int kk = 0; kk < ii; kk++)
                        {sum += Lower[ii][kk] * Upper[kk][jj];}
                    Upper[ii][jj] = matrixA[ii][jj] - sum;
                    }
                    else if(ii < jj)
```

```
                    {
                    Lower[ii][jj]=0;
                    for (int kk = 0; kk < ii; kk++)
                        {sum += Lower[ii][kk] * Upper[kk][jj];}
                    Upper[ii][jj] = matrixA[ii][jj] - sum;
                    }
                    else
                    {
                    Upper[ii][jj]=0;
                    for (int kk = 0; kk < jj; kk++)
                        {sum += Lower[ii][kk] * Upper[kk][jj];}
                    Lower[ii][jj] =(matrixA[ii][jj] - sum)/Upper[jj][jj];
                    }
                }
            }
    for (int ii = 0; ii < rc; ii++)
    {
        sum = 0;
        for (int jj = 0; jj < ii; jj++)
            {sum += Lower[ii][jj] * y[jj];}
        y[ii] = matrixB[ii] - sum;
    }

    for (int ii = rc - 1; ii >= 0; ii--)
    {
        sum = 0;
        for (int jj = ii + 1; jj < rc; jj++)
            {sum += Upper[ii][jj] * x[jj];}
        x[ii] = (y[ii] - sum)/Upper[ii][ii];
    }
return x[2];
}

__device__ float RK4_Init(float x1,float t1,float SSize)

{
float k1,k2,k3,k4;

k1=SSize*x1;
k2=SSize*(x1+0.5*SSize*t1);
k3=SSize*(x1+0.5*SSize*t1);
k4=SSize*(x1+SSize*t1);

return (1.0/6.0)*(k1+2*k2+2*k3+k4);
}

__device__ void InitialFS_LAC( float t1, float t2, float t3, float& dt1,
```

```
    float& dt2, float& dt3, float& d2t1, float& d2t2, float& d2t3,
    float& d3t1, float& d3t2, float& d3t3, float n1, float n2, float n3,
    float& dn1, float& dn2, float& dn3, float& d2n1, float& d2n2, float&
    d2n3, float Alpha, float InitRC, float Lambda, float ArcLength)
{
dt1=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*n1);
dt2=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*n2);
dt3=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*n3);
dn1=ArcLength*(-inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*t1);
dn2=ArcLength*(-inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*t2);
dn3=ArcLength*(-inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*t3);

d2t1=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dn1+
    inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*n1);
d2t2=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dn2+
    inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*n2);
d2t3=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dn3+
    inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*n3);
d2n1=ArcLength*(-inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dt1-
    inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*t1);
d2n2=ArcLength*(-inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dt2-
    inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*t2);
d2n3=ArcLength*(-inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dt3-
    inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*t3);

d3t1=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*d2n1+2.0*
    inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*dn1+inid2cLACdx2(0,Alpha,
    Lambda,InitRC,ArcLength)*n1);
d3t2=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*d2n2+
    2.0*inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*dn2+inid2cLACdx2(0,Alpha
    ,Lambda,InitRC,ArcLength)*n2);
d3t3=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*d2n3+
    2.0*inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*dn3+inid2cLACdx2(0,Alpha
    ,Lambda,InitRC,ArcLength)*n3);
}

__device__ void InitialFS_LASC( float t1, float t2, float t3, float& dt1,
    float& dt2, float& dt3, float& d2t1, float& d2t2, float& d2t3, float&
    d3t1, float& d3t2, float& d3t3, float n1, float n2, float n3, float&
    dn1, float& dn2, float& dn3, float& d2n1, float& d2n2, float& d2n3,
    float b1, float b2, float b3, float& db1, float& db2, float& db3,
    float& d2b1, float& d2b2, float& d2b3, float Alpha, float Beta, float
    InitRC, float Omega, float Lambda, float InitRT, float ArcLength)
{
dt1=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*n1);
dt2=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*n2);
dt3=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*n3);
```

```
dn1=ArcLength*(-inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*t1+
     iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)*b1);
dn2=ArcLength*(-inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*t2+
     iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)*b2);
dn3=ArcLength*(-inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*t3+
     iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)*b3);
db1=ArcLength*(-iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)*n1);
db2=ArcLength*(-iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)*n2);
db3=ArcLength*(-iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)*n3);

d2t1=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dn1+
     inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*n1);
d2t2=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dn2+
     inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*n2);
d2t3=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dn3+
     inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*n3);
d2n1=ArcLength*(-inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*t1-
     inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dt1+inidtLACdx(0,Beta,
     Omega,InitRT,ArcLength)*b1+iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)
     *db1);
d2n2=ArcLength*(-inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*t2-
     inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dt2+inidtLACdx(0,Beta,
     Omega,InitRT,ArcLength)*b2+iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)
     *db2);
d2n3=ArcLength*(-inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*t3-
     inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*dt3+inidtLACdx(0,Beta,
     Omega,InitRT,ArcLength)*b3+iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)
     *db3);
d2b1=ArcLength*(-iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)*dn1-
     inidtLACdx(0,Beta,Omega,InitRT,ArcLength)*n1);
d2b2=ArcLength*(-iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)*dn2-
     inidtLACdx(0,Beta,Omega,InitRT,ArcLength)*n2);
d2b3=ArcLength*(-iniTorsion_LAC(0,Beta,Omega,InitRT,ArcLength)*dn3-
     inidtLACdx(0,Beta,Omega,InitRT,ArcLength)*n3);
d3t1=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*d2n1+
     2.0*inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*dn1+inid2cLACdx2(0,Alpha
     ,Lambda,InitRC,ArcLength)*n1);
d3t2=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*d2n2+
     2.0*inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*dn2+inid2cLACdx2(0,Alpha
     ,Lambda,InitRC,ArcLength)*n2);
d3t3=ArcLength*(inicurvature_LAC(0,Alpha,Lambda,InitRC,ArcLength)*d2n3+
     2.0*inidcLACdx(0,Alpha,Lambda,InitRC,ArcLength)*dn3+inid2cLACdx2(0,Alpha
     ,Lambda,InitRC,ArcLength)*n3);
}
__device__ void FS_LAC( float& x1, float& x2, float& x3, float& t1, float&
     t2, float& t3, float& dt1, float& dt2, float& dt3, float& d2t1,
     float& d2t2, float& d2t3, float& d3t1, float& d3t2, float& d3t3,
```

```
    float& n1, float& n2, float& n3, float& dn1, float& dn2, float&
    dn3, float& d2n1, float& d2n2, float& d2n3, float Alpha, float
    InitRC, float Lambda, float ArcLength, float initp, float stepsize)
{
x1=x1+RK4_Init(t1,dt1,stepsize);
x2=x2+RK4_Init(t2,dt2,stepsize);
x3=x3+RK4_Init(t3,dt3,stepsize);
t1=t1+RK4_Init(dt1,d2t1,stepsize);
t2=t2+RK4_Init(dt2,d2t2,stepsize);
t3=t3+RK4_Init(dt3,d2t3,stepsize);
n1=n1+RK4_Init(dn1,d2n1,stepsize);
n2=n2+RK4_Init(dn2,d2n2,stepsize);
n3=n3+RK4_Init(dn3,d2n3,stepsize);

dt1=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
    n1);
dt2=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*n2);
dt3=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*n3);
dn1=ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*t1);
dn2=ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*t2);
dn3=ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*t3);

d2t1=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
    dn1+inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*n1);
d2t2=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*dn2+inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
    n2);
d2t3=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*dn3+inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
    n3);
d2n1=ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*dt1-inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
    t1);
d2n2=ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*dt2-inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
    t2);
d2n3=ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*dt3-inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
    t3);

d3t1=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
    d2n1+2.0*inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*dn1+
```

```
     inid2cLACdx2(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*n1);
d3t2=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
     ArcLength)*d2n2+2.0*inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,
     ArcLength)*dn2+inid2cLACdx2(initp+stepsize,Alpha,Lambda,InitRC,ArcLength
     )*n2);
d3t3=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
     ArcLength)*d2n3+2.0*inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,
     ArcLength)*dn3+inid2cLACdx2(initp+stepsize,Alpha,Lambda,InitRC,ArcLength
     )*n3);

}


__device__ void FS_LASC( float& x1, float& x2, float& x3, float& t1, float&
     t2, float& t3, float& dt1, float& dt2, float& dt3, float& d2t1,
     float& d2t2, float& d2t3, float& d3t1, float& d3t2, float& d3t3,
     float& n1, float& n2, float& n3, float& dn1, float& dn2, float& dn3,
     float& d2n1, float& d2n2, float& d2n3, float& b1, float& b2, float&
     b3, float& db1, float& db2, float& db3, float& d2b1, float& d2b2,
     float& d2b3, float Alpha, float Beta, float InitRC, float Omega, float
     Lambda, float InitRT, float ArcLength, float initp, float stepsize)
{
x1=x1+RK4_Init(t1,dt1,stepsize);
x2=x2+RK4_Init(t2,dt2,stepsize);
x3=x3+RK4_Init(t3,dt3,stepsize);
t1=t1+RK4_Init(dt1,d2t1,stepsize);
t2=t2+RK4_Init(dt2,d2t2,stepsize);
t3=t3+RK4_Init(dt3,d2t3,stepsize);
n1=n1+RK4_Init(dn1,d2n1,stepsize);
n2=n2+RK4_Init(dn2,d2n2,stepsize);
n3=n3+RK4_Init(dn3,d2n3,stepsize);
b1=b1+RK4_Init(db1,d2b1,stepsize);
b2=b2+RK4_Init(db2,d2b2,stepsize);
b3=b3+RK4_Init(db3,d2b3,stepsize);

dt1=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
     n1);
dt2=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
     ArcLength)*n2);
dt3=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
     ArcLength)*n3);
dn1=ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
     ArcLength)*t1+iniTorsion_LAC(initp+stepsize,Beta,Omega,InitRT,ArcLength)
     *b1);
dn2=ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
     ArcLength)*t2+iniTorsion_LAC(initp+stepsize,Beta,Omega,InitRT,ArcLength)
     *b2);
```

```
dn3=ArcLength*(-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*t3+iniTorsion_LAC(initp+stepsize,Beta,Omega,InitRT,ArcLength)
    *b3);
db1=ArcLength*(-iniTorsion_LAC(initp+stepsize,Beta,Omega,InitRT,
    ArcLength)*n1);
db2=ArcLength*(-iniTorsion_LAC(initp+stepsize,Beta,Omega,InitRT,
    ArcLength)*n2);
db3=ArcLength*(-iniTorsion_LAC(initp+stepsize,Beta,Omega,InitRT,
    ArcLength)*n3);

d2t1=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
    dn1+inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*n1);
d2t2=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*dn2+inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
    n2);
d2t3=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*dn3+inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*
    n3);
d2n1=ArcLength*(-inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength
    )*t1-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*dt1+
    inidtLACdx(initp+stepsize,Beta,Omega,InitRT,ArcLength)*b1+iniTorsion_LAC
    (initp+stepsize,Beta,Omega,InitRT,ArcLength)*db1);
d2n2=ArcLength*(-inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength
    )*t2-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*dt2+
    inidtLACdx(initp+stepsize,Beta,Omega,InitRT,ArcLength)*b2+iniTorsion_LAC
    (initp+stepsize,Beta,Omega,InitRT,ArcLength)*db2);
d2n3=ArcLength*(-inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,ArcLength
    )*t3-inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*dt3+
    inidtLACdx(initp+stepsize,Beta,Omega,InitRT,ArcLength)*b3+iniTorsion_LAC
    (initp+stepsize,Beta,Omega,InitRT,ArcLength)*db3);
d2b1=ArcLength*(-iniTorsion_LAC(initp+stepsize,Beta,Omega,InitRT,
    ArcLength)*dn1-inidtLACdx(initp+stepsize,Beta,Omega,InitRT,ArcLength)*n1
    );
d2b2=ArcLength*(-iniTorsion_LAC(initp+stepsize,Beta,Omega,InitRT,
    ArcLength)*dn2-inidtLACdx(initp+stepsize,Beta,Omega,InitRT,ArcLength)*n2
    );
d2b3=ArcLength*(-iniTorsion_LAC(initp+stepsize,Beta,Omega,InitRT,
    ArcLength)*dn3-inidtLACdx(initp+stepsize,Beta,Omega,InitRT,ArcLength)*n3
    );
d3t1=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*d2n1+2.0*inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*dn1+inid2cLACdx2(initp+stepsize,Alpha,Lambda,InitRC,ArcLength
    )*n1);
d3t2=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*d2n2+2.0*inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,
    ArcLength)*dn2+inid2cLACdx2(initp+stepsize,Alpha,Lambda,InitRC,ArcLength
    )*n2);
```

```
d3t3=ArcLength*(inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
     ArcLength)*d2n3+2.0*inidcLACdx(initp+stepsize,Alpha,Lambda,InitRC,
     ArcLength)*dn3+inid2cLACdx2(initp+stepsize,Alpha,Lambda,InitRC,ArcLength
     )*n3);
}
__device__ void RK4_part( float old_x1, float old_x2, float old_x3, float
     old_t1, float old_t2, float old_t3, float old_dt1, float old_dt2,
     float old_dt3, float old_d2t1, float old_d2t2, float old_d2t3,
     float old_n1, float old_n2, float old_n3, float old_dn1, float
     old_dn2, float old_dn3, float old_d2n1, float old_d2n2, float
     old_d2n3, float& x1, float& x2, float& x3, float& t1, float& t2,
     float& t3, float& dt1, float& dt2, float& dt3, float Alpha, float
     InitRC, float Lambda, float ArcLength, float initp, float stepsize)
{
float n1,n2,n3;

x1=old_x1+RK4_Init(old_t1,old_dt1,stepsize);
x2=old_x2+RK4_Init(old_t2,old_dt2,stepsize);
x3=old_x3+RK4_Init(old_t3,old_dt3,stepsize);
t1=old_t1+RK4_Init(old_dt1,old_d2t1,stepsize);
t2=old_t2+RK4_Init(old_dt2,old_d2t2,stepsize);
t3=old_t3+RK4_Init(old_dt3,old_d2t3,stepsize);
n1=old_n1+RK4_Init(old_dn1,old_d2n1,stepsize);
n2=old_n2+RK4_Init(old_dn2,old_d2n2,stepsize);
n3=old_n3+RK4_Init(old_dn3,old_d2n3,stepsize);

dt1=ArcLength*inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,ArcLength)*n1
     ;
dt2=ArcLength*inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
     ArcLength)*n2;
dt3=ArcLength*inicurvature_LAC(initp+stepsize,Alpha,Lambda,InitRC,
     ArcLength)*n3;
}


__device__ float RK4_func(float (f)(float, float, float, float, float, float,
     float, float, float, float, float, float, float, float, float, float,
     float, float, float, float, float, float, float, float, float, float,
     float, float, float, float, float, float, float, float, float, float,
     float, float, mint), float u_s, float v_s, float u1_x1, float u1_x2,
     float u1_x3, float u1_t1, float u1_t2, float u1_t3, float u1_dt1,
     float u1_dt2, float u1_dt3, float i1_d2tu1, float i1_d2tu2, float
     i1_d2tu3, float i1_nu1, float i1_nu2, float i1_nu3, float i1_dnu1,
     float i1_dnu2, float i1_dnu3, float i1_d2nu1, float i1_d2nu2, float
     i1_d2nu3, float u2_x1, float u2_x2, float u2_x3, float u2_t1, float
     u2_t2, float u2_t3, float u2_dt1, float u2_dt2, float u2_dt3, float
     i2_d2tu1, float i2_d2tu2, float i2_d2tu3, float i2_nu1, float i2_nu2,
```

```
    float i2_nu3, float i2_dnu1, float i2_dnu2, float i2_dnu3, float
    i2_d2nu1, float i2_d2nu2, float i2_d2nu3, float i2_bu1, float i2_bu2,
    float i2_bu3, float i2_dbu1, float i2_dbu2, float i2_dbu3, float
    i2_d2bu1, float i2_d2bu2, float i2_d2bu3, float v1_x1, float v1_x2,
    float v1_x3, float v1_t1, float v1_t2, float v1_t3, float v1_dt1,
    float v1_dt2, float v1_dt3, float j1_d2tv1, float j1_d2tv2, float
    j1_d2tv3, float j1_nv1, float j1_nv2, float j1_nv3, float j1_dnv1,
    float j1_dnv2, float j1_dnv3, float j1_d2nv1, float j1_d2nv2, float
    j1_d2nv3, float v2_x1, float v2_x2, float v2_x3, float v2_t1, float
    v2_t2, float v2_t3, float v2_dt1, float v2_dt2, float v2_dt3, float
    j2_d2tv1, float j2_d2tv2, float j2_d2tv3, float j2_nv1, float j2_nv2,
    float j2_nv3, float j2_dnv1, float j2_dnv2, float j2_dnv3, float
    j2_d2nv1, float j2_d2nv2, float j2_d2nv3, float j2_bv1, float j2_bv2,
    float j2_bv3, float j2_dbv1, float j2_dbv2, float j2_dbv3, float
    j2_d2bv1, float j2_d2bv2, float j2_d2bv3, mint optimum, int u_or_v)

{

float k1,k2,k3,k4;
float
    new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,new_u1_t2,new_u1_t3,new_u1_dt1,
    new_u1_dt2,new_u1_dt3,new_i1_d2tu1,new_i1_d2tu2,new_i1_d2tu3;
float
    new_u2_x1,new_u2_x2,new_u2_x3,new_u2_t1,new_u2_t2,new_u2_t3,new_u2_dt1,
    new_u2_dt2,new_u2_dt3,new_i2_d2tu1,new_i2_d2tu2,new_i2_d2tu3;
float
    new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,new_v1_dt1,
    new_v1_dt2,new_v1_dt3,new_j1_d2tv1,new_j1_d2tv2,new_j1_d2tv3;
float
    new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,new_v2_t2,new_v2_t3,new_v2_dt1,
    new_v2_dt2,new_v2_dt3,new_j2_d2tv1,new_j2_d2tv2,new_j2_d2tv3;


if(u_or_v==uuu)
{
k1=h*f(u_s,v_s,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,u2_x1,
    u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,v1_x3,
    v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,
    v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum);

RK4_part(u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,i1_d2tu1,
    i1_d2tu2,i1_d2tu3,i1_nu1,i1_nu2,i1_nu3,i1_dnu1,i1_dnu2,i1_dnu3,i1_d2nu1,
    i1_d2nu2,i1_d2nu3,new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,new_u1_t2,
    new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3,AlphaLAC,InitRCLAC,Lambda_Max
    _LAC,ArcLength_Max_LAC,u_s,0.5*k1);
RK4_part(u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,
    i2_d2tu1,i2_d2tu2,i2_d2tu3,i2_nu1,i2_nu2,i2_nu3,i2_dnu1,i2_dnu2,i2_dnu3,
```

```
    i2_d2nu1,i2_d2nu2,i2_d2nu3,new_u2_x1,new_u2_x2,new_u2_x3,new_u2_t1,
    new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2,new_u2_dt3,AlphaLAC,InitRCLAC,
    Lambda_Max_LASC,ArcLength_Max_LASC,u_s,0.5*k1);


k2=h*f(u_s,v_s,new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,new_u1_t2,new_u1_t3,
    new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u2_x1,new_u2_x2,new_u2_x3,new_u2_t1
    ,new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2,new_u2_dt3,v1_x1,v1_x2,v1_x3,
    v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,
    v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum);

RK4_part(u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,i1_d2tu1,
    i1_d2tu2,i1_d2tu3,i1_nu1,i1_nu2,i1_nu3,i1_dnu1,i1_dnu2,i1_dnu3,i1_d2nu1,
    i1_d2nu2,i1_d2nu3,new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,new_u1_t2,
    new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3,AlphaLAC,InitRCLAC,Lambda_Max
    _LAC,ArcLength_Max_LAC,u_s,0.5*k2);
RK4_part(u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,
    i2_d2tu1,i2_d2tu2,i2_d2tu3,i2_nu1,i2_nu2,i2_nu3,i2_dnu1,i2_dnu2,i2_dnu3,
    i2_d2nu1,i2_d2nu2,i2_d2nu3,new_u2_x1,new_u2_x2,new_u2_x3,new_u2_t1,
    new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2,new_u2_dt3,AlphaLAC,InitRCLAC,
    Lambda_Max_LASC,ArcLength_Max_LASC,u_s,0.5*k2);

k3=h*f(u_s,v_s,new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,new_u1_t2,new_u1_t3,
    new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u2_x1,new_u2_x2,new_u2_x3,new_u2_t1
    ,new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2,new_u2_dt3,v1_x1,v1_x2,v1_x3,
    v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,
    v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum);

RK4_part(u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,i1_d2tu1,
    i1_d2tu2,i1_d2tu3,i1_nu1,i1_nu2,i1_nu3,i1_dnu1,i1_dnu2,i1_dnu3,i1_d2nu1,
    i1_d2nu2,i1_d2nu3,new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,new_u1_t2,
    new_u1_t3,new_u1_dt1,new_u1_dt2,new_u1_dt3,AlphaLAC,InitRCLAC,Lambda_Max
    _LAC,ArcLength_Max_LAC,u_s,k3);

RK4_part(u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,i2_d2tu1,
    i2_d2tu2,i2_d2tu3,i2_nu1,i2_nu2,i2_nu3,i2_dnu1,i2_dnu2,i2_dnu3,i2_d2nu1,
    i2_d2nu2,i2_d2nu3,new_u2_x1,new_u2_x2,new_u2_x3,new_u2_t1,new_u2_t2,
    new_u2_t3,new_u2_dt1,new_u2_dt2,new_u2_dt3,AlphaLAC,InitRCLAC,Lambda_Max
    _LASC,ArcLength_Max_LASC,u_s,k3);

k4=h*f(u_s,v_s,new_u1_x1,new_u1_x2,new_u1_x3,new_u1_t1,new_u1_t2,new_u1_t3,
    new_u1_dt1,new_u1_dt2,new_u1_dt3,new_u2_x1,new_u2_x2,new_u2_x3,new_u2_t1
    ,new_u2_t2,new_u2_t3,new_u2_dt1,new_u2_dt2,new_u2_dt3,v1_x1,v1_x2,v1_x3,
    v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,
    v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum);

return (1.0/6.0)*(k1+2*k2+2*k3+k4);
```

```
}
else if (u_or_v==vvv)
{

k1=h*f(u_s,v_s,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,u2_x1,
     u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,v1_x2,v1_x3,
     v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,
     v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum);

RK4_part(v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,j1_d2tv1,
     j1_d2tv2,j1_d2tv3,j1_nv1,j1_nv2,j1_nv3,j1_dnv1,j1_dnv2,j1_dnv3,j1_d2nv1,
     j1_d2nv2,j1_d2nv3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,
     new_v1_t3,new_v1_dt1,new_v1_dt2,new_v1_dt3,AlphaLAC,InitRCLAC,Lambda_Min
     _LAC,ArcLength_Min_LAC,v_s,0.5*k1);

RK4_part(v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,j2_d2tv1,
     j2_d2tv2,j2_d2tv3,j2_nv1,j2_nv2,j2_nv3,j2_dnv1,j2_dnv2,j2_dnv3,j2_d2nv1,
     j2_d2nv2,j2_d2nv3,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,new_v2_t2,
     new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,AlphaLAC,InitRCLAC,Lambda_Min
     _LASC,ArcLength_Min_LASC,v_s,0.5*k1);



k2=h*f(u_s,v_s,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,u2_x1,
     u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,new_v1_x1,new_v1_x2,
     new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,new_v1_dt1,new_v1_dt2,new_v1_dt3
     ,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,new_v2_t2,new_v2_t3,new_v2_dt1,
     new_v2_dt2,new_v2_dt3,optimum);

RK4_part(v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,j1_d2tv1,
     j1_d2tv2,j1_d2tv3,j1_nv1,j1_nv2,j1_nv3,j1_dnv1,j1_dnv2,j1_dnv3,j1_d2nv1,
     j1_d2nv2,j1_d2nv3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,
     new_v1_t3,new_v1_dt1,new_v1_dt2,new_v1_dt3,AlphaLAC,InitRCLAC,Lambda_Min
     _LAC,ArcLength_Min_LAC,v_s,0.5*k2);
RK4_part(v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,
     j2_d2tv1,j2_d2tv2,j2_d2tv3,j2_nv1,j2_nv2,j2_nv3,j2_dnv1,j2_dnv2,j2_dnv3,
     j2_d2nv1,j2_d2nv2,j2_d2nv3,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,
     new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,AlphaLAC,InitRCLAC,
     Lambda_Min_LASC,ArcLength_Min_LASC,v_s,0.5*k2);


k3=h*f(u_s,v_s,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,u2_x1,
     u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,new_v1_x1,new_v1_x2,
     new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,new_v1_dt1,new_v1_dt2,new_v1_dt3
     ,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,new_v2_t2,new_v2_t3,new_v2_dt1,
     new_v2_dt2,new_v2_dt3,optimum);
```

```
RK4_part(v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,j1_d2tv1,
    j1_d2tv2,j1_d2tv3,j1_nv1,j1_nv2,j1_nv3,j1_dnv1,j1_dnv2,j1_dnv3,j1_d2nv1,
    j1_d2nv2,j1_d2nv3,new_v1_x1,new_v1_x2,new_v1_x3,new_v1_t1,new_v1_t2,
    new_v1_t3,new_v1_dt1,new_v1_dt2,new_v1_dt3,AlphaLAC,InitRCLAC,Lambda_Min
    _LAC,ArcLength_Min_LAC,v_s,k3);
RK4_part(v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,
    j2_d2tv1,j2_d2tv2,j2_d2tv3,j2_nv1,j2_nv2,j2_nv3,j2_dnv1,j2_dnv2,j2_dnv3,
    j2_d2nv1,j2_d2nv2,j2_d2nv3,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,
    new_v2_t2,new_v2_t3,new_v2_dt1,new_v2_dt2,new_v2_dt3,AlphaLAC,InitRCLAC,
    Lambda_Min_LASC,ArcLength_Min_LASC,v_s,k3);

k4=h*f(u_s,v_s,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,u1_dt3,u2_x1,
    u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,new_v1_x1,new_v1_x2,
    new_v1_x3,new_v1_t1,new_v1_t2,new_v1_t3,new_v1_dt1,new_v1_dt2,new_v1_dt3
    ,new_v2_x1,new_v2_x2,new_v2_x3,new_v2_t1,new_v2_t2,new_v2_t3,new_v2_dt1,
    new_v2_dt2,new_v2_dt3,optimum);

return (1.0/6.0)*(k1+2*k2+2*k3+k4);
}
}


__device__ void Integration( float pu[], float pv[], float u1_x1[], float
    u1_x2[], float u1_x3[], float u1_t1[], float u1_t2[], float u1_t3[],
    float u1_dt1[], float u1_dt2[], float u1_dt3[], float u1_d2t1[], float
    u1_d2t2[], float u1_d2t3[], float u1_d3t1[], float u1_d3t2[], float
    u1_d3t3[], float u2_x1[], float u2_x2[], float u2_x3[], float u2_t1[],
    float u2_t2[], float u2_t3[], float u2_dt1[], float u2_dt2[], float
    u2_dt3[], float u2_d2t1[], float u2_d2t2[], float u2_d2t3[], float
    u2_d3t1[], float u2_d3t2[], float u2_d3t3[], float v1_x1[], float
    v1_x2[], float v1_x3[], float v1_t1[], float v1_t2[], float v1_t3[],
    float v1_dt1[], float v1_dt2[], float v1_dt3[], float v1_d2t1[], float
    v1_d2t2[], float v1_d2t3[], float v1_d3t1[], float v1_d3t2[], float
    v1_d3t3[], float v2_x1[], float v2_x2[], float v2_x3[], float v2_t1[],
    float v2_t2[], float v2_t3[], float v2_dt1[], float v2_dt2[], float
    v2_dt3[], float v2_d2t1[], float v2_d2t2[], float v2_d2t3[], float
    v2_d3t1[], float v2_d3t2[], float v2_d3t3[], mint idx, mint optimum)
{
float t1_Max_LAC=-1*ScaleToOrigin_Max_LAC*ArcLength_Max_LAC,
    t2_Max_LAC=0, t3_Max_LAC=0;
float n1_Max_LAC=0, n2_Max_LAC=0,
    n3_Max_LAC=1*ScaleToOrigin_Max_LAC*ArcLength_Max_LAC;
float t1_Min_LAC=0, t2_Min_LAC=-1*ScaleToOrigin_Min_LAC*ArcLength_Min_LAC,
    t3_Min_LAC=0;
float n1_Min_LAC=0, n2_Min_LAC=0,
    n3_Min_LAC=-1*ScaleToOrigin_Min_LAC*ArcLength_Min_LAC;
```

```
float t1_Max_LASC=-1*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC,
     t2_Max_LASC=0, t3_Max_LASC=0;
float n1_Max_LASC=0,
     n2_Max_LASC=-0.5930588126117428*ScaleToOrigin_Max_LASC*ArcLength_Max
     _LASC,
     n3_Max_LASC=0.8051591425200051*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC;
float b1_Max_LASC=0,
     b2_Max_LASC=0.8051591425200051*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC
     ,
     b3_Max_LASC=0.5930588126117428*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC;

float t1_Min_LASC=0,
     t2_Min_LASC=-1*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC, t3_Min_LASC=0;
float
     n1_Min_LASC=-0.5930588126117428*ScaleToOrigin_Min_LASC*ArcLength_Min
     _LASC, n2_Min_LASC=0,
     n3_Min_LASC=-0.8051591425200051*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC;
float
     b1_Min_LASC=0.8051591425200051*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC
     , b2_Min_LASC=0,
     b3_Min_LASC=-0.5930588126117428*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC;

float Init_u=0.0, Init_v=0.0;

float dt1_Max_LAC, dt2_Max_LAC, dt3_Max_LAC;
float d2t1_Max_LAC, d2t2_Max_LAC, d2t3_Max_LAC;
float d3t1_Max_LAC, d3t2_Max_LAC, d3t3_Max_LAC;
float dt1_Min_LAC, dt2_Min_LAC, dt3_Min_LAC;
float d2t1_Min_LAC, d2t2_Min_LAC, d2t3_Min_LAC;
float d3t1_Min_LAC, d3t2_Min_LAC, d3t3_Min_LAC;
float dn1_Max_LAC, dn2_Max_LAC, dn3_Max_LAC;
float d2n1_Max_LAC, d2n2_Max_LAC, d2n3_Max_LAC;
float dn1_Min_LAC, dn2_Min_LAC, dn3_Min_LAC;
float d2n1_Min_LAC, d2n2_Min_LAC, d2n3_Min_LAC;

float dt1_Max_LASC, dt2_Max_LASC, dt3_Max_LASC;
float d2t1_Max_LASC, d2t2_Max_LASC, d2t3_Max_LASC;
float d3t1_Max_LASC, d3t2_Max_LASC, d3t3_Max_LASC;
float dt1_Min_LASC, dt2_Min_LASC, dt3_Min_LASC;
float d2t1_Min_LASC, d2t2_Min_LASC, d2t3_Min_LASC;
float d3t1_Min_LASC, d3t2_Min_LASC, d3t3_Min_LASC;
float dn1_Max_LASC, dn2_Max_LASC, dn3_Max_LASC;
float d2n1_Max_LASC, d2n2_Max_LASC, d2n3_Max_LASC;
float dn1_Min_LASC, dn2_Min_LASC, dn3_Min_LASC;
float d2n1_Min_LASC, d2n2_Min_LASC, d2n3_Min_LASC;
float db1_Max_LASC, db2_Max_LASC, db3_Max_LASC;
float d2b1_Max_LASC, d2b2_Max_LASC, d2b3_Max_LASC;
```

```
float db1_Min_LASC, db2_Min_LASC, db3_Min_LASC;
float d2b1_Min_LASC, d2b2_Min_LASC, d2b3_Min_LASC;

float dummy_u1_x1=Point00x, dummy_u1_x2=Point00y, dummy_u1_x3=Point00z;
float dummy_u2_x1=Point30x, dummy_u2_x2=Point30y, dummy_u2_x3=Point30z;
float dummy_v1_x1=Point00x, dummy_v1_x2=Point00y, dummy_v1_x3=Point00z;
float dummy_v2_x1=Point03x, dummy_v2_x2=Point03y, dummy_v2_x3=Point03z;
float SSize_u=0, SSize_v=0, ratio;

InitialFS_LAC(t1_Max_LAC,t2_Max_LAC,t3_Max_LAC,dt1_Max_LAC,dt2_Max_LAC,dt3_Max
    _LAC,d2t1_Max_LAC,d2t2_Max_LAC,d2t3_Max_LAC,d3t1_Max_LAC,d3t2_Max_LAC,
    d3t3_Max_LAC,n1_Max_LAC,n2_Max_LAC,n3_Max_LAC,dn1_Max_LAC,dn2_Max_LAC,
    dn3_Max_LAC,d2n1_Max_LAC,d2n2_Max_LAC,d2n3_Max_LAC,AlphaLAC,InitRCLAC,
    Lambda_Max_LAC,ArcLength_Max_LAC);

InitialFS_LASC(t1_Max_LASC,t2_Max_LASC,t3_Max_LASC,dt1_Max_LASC,dt2_Max_LASC,
    dt3_Max_LASC,d2t1_Max_LASC,d2t2_Max_LASC,d2t3_Max_LASC,d3t1_Max_LASC,
    d3t2_Max_LASC,d3t3_Max_LASC,n1_Max_LASC,n2_Max_LASC,n3_Max_LASC,dn1_Max
    _LASC,dn2_Max_LASC,dn3_Max_LASC,d2n1_Max_LASC,d2n2_Max_LASC,d2n3_Max
    _LASC,b1_Max_LASC,b2_Max_LASC,b3_Max_LASC,db1_Max_LASC,db2_Max_LASC,
    db3_Max_LASC,d2b1_Max_LASC,d2b2_Max_LASC,d2b3_Max_LASC,AlphaLAC,BetaLASC
    ,InitRCLAC,OmegaLASC,Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC)
    ;

InitialFS_LAC(t1_Min_LAC,t2_Min_LAC,t3_Min_LAC,dt1_Min_LAC,dt2_Min_LAC,dt3_Min
    _LAC,d2t1_Min_LAC,d2t2_Min_LAC,d2t3_Min_LAC,d3t1_Min_LAC,d3t2_Min_LAC,
    d3t3_Min_LAC,n1_Min_LAC,n2_Min_LAC,n3_Min_LAC,dn1_Min_LAC,dn2_Min_LAC,
    dn3_Min_LAC,d2n1_Min_LAC,d2n2_Min_LAC,d2n3_Min_LAC,AlphaLAC,InitRCLAC,
    Lambda_Min_LAC,ArcLength_Min_LAC);

InitialFS_LASC(t1_Min_LASC,t2_Min_LASC,t3_Min_LASC,dt1_Min_LASC,dt2_Min_LASC,
    dt3_Min_LASC,d2t1_Min_LASC,d2t2_Min_LASC,d2t3_Min_LASC,d3t1_Min_LASC,
    d3t2_Min_LASC,d3t3_Min_LASC,n1_Min_LASC,n2_Min_LASC,n3_Min_LASC,dn1_Min
    _LASC,dn2_Min_LASC,dn3_Min_LASC,d2n1_Min_LASC,d2n2_Min_LASC,d2n3_Min
    _LASC,b1_Min_LASC,b2_Min_LASC,b3_Min_LASC,db1_Min_LASC,db2_Min_LASC,
    db3_Min_LASC,d2b1_Min_LASC,d2b2_Min_LASC,d2b3_Min_LASC,AlphaLAC,BetaLASC
    ,InitRCLAC,OmegaLASC,Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC)
    ;

if(idx==0 || idx==1000)
{
pu[0]=Init_u;
pv[0]=Init_v;
u1_x1[0]=dummy_u1_x1;
u1_x2[0]=dummy_u1_x2;
u1_x3[0]=dummy_u1_x3;
u1_t1[0]=t1_Max_LAC;
```

```
u1_t2[0]=t2_Max_LAC;
u1_t3[0]=t3_Max_LAC;
u1_dt1[0]=dt1_Max_LAC;
u1_dt2[0]=dt2_Max_LAC;
u1_dt3[0]=dt3_Max_LAC;
u1_d2t1[0]=d2t1_Max_LAC;
u1_d2t2[0]=d2t2_Max_LAC;
u1_d2t3[0]=d2t3_Max_LAC;
u1_d3t1[0]=d3t1_Max_LAC;
u1_d3t2[0]=d3t2_Max_LAC;
u1_d3t3[0]=d3t3_Max_LAC;
u2_x1[0]=dummy_u2_x1;
u2_x2[0]=dummy_u2_x2;
u2_x3[0]=dummy_u2_x3;
u2_t1[0]=t1_Max_LASC;
u2_t2[0]=t2_Max_LASC;
u2_t3[0]=t3_Max_LASC;
u2_dt1[0]=dt1_Max_LASC;
u2_dt2[0]=dt2_Max_LASC;
u2_dt3[0]=dt3_Max_LASC;
u2_d2t1[0]=d2t1_Max_LASC;
u2_d2t2[0]=d2t2_Max_LASC;
u2_d2t3[0]=d2t3_Max_LASC;
u2_d3t1[0]=d3t1_Max_LASC;
u2_d3t2[0]=d3t2_Max_LASC;
u2_d3t3[0]=d3t3_Max_LASC;
v1_x1[0]=dummy_v1_x1;
v1_x2[0]=dummy_v1_x2;
v1_x3[0]=dummy_v1_x3;
v1_t1[0]=t1_Min_LAC;
v1_t2[0]=t2_Min_LAC;
v1_t3[0]=t3_Min_LAC;
v1_dt1[0]=dt1_Min_LAC;
v1_dt2[0]=dt2_Min_LAC;
v1_dt3[0]=dt3_Min_LAC;
v1_d2t1[0]=d2t1_Min_LAC;
v1_d2t2[0]=d2t2_Min_LAC;
v1_d2t3[0]=d2t3_Min_LAC;
v1_d3t1[0]=d3t1_Min_LAC;
v1_d3t2[0]=d3t2_Min_LAC;
v1_d3t3[0]=d3t3_Min_LAC;
v2_x1[0]=dummy_v2_x1;
v2_x2[0]=dummy_v2_x2;
v2_x3[0]=dummy_v2_x3;
v2_t1[0]=t1_Min_LASC;
v2_t2[0]=t2_Min_LASC;
v2_t3[0]=t3_Min_LASC;
```

```
v2_dt1[0]=dt1_Min_LASC;
v2_dt2[0]=dt2_Min_LASC;
v2_dt3[0]=dt3_Min_LASC;
v2_d2t1[0]=d2t1_Min_LASC;
v2_d2t2[0]=d2t2_Min_LASC;
v2_d2t3[0]=d2t3_Min_LASC;
v2_d3t1[0]=d3t1_Min_LASC;
v2_d3t2[0]=d3t2_Min_LASC;
v2_d3t3[0]=d3t3_Min_LASC;

for(int j=1;j<n;j++)
    {
    SSize_u=h;
    SSize_v=h;
    if((Init_u+SSize_u) > 1.0)
        {
        if(Init_u==1.0)
            {
            SSize_u=0.0;
            SSize_v=0.0;
            }
        else
            {
            ratio=(1.0-Init_u)/SSize_u;
            SSize_u*=ratio;
            SSize_v*=ratio;
            }
        }

    if((Init_v+SSize_v) > 1.0)
        {
        if(Init_v==1.0)
            {
            SSize_u=0.0;
            SSize_v=0.0;
            }
        else
            {
            ratio=(1.0-Init_v)/SSize_v;
            SSize_u*=ratio;
            SSize_v*=ratio;
            }
        }

    FS_LAC(dummy_u1_x1,dummy_u1_x2,dummy_u1_x3,t1_Max_LAC,t2_Max_LAC,t3_Max
    _LAC,dt1_Max_LAC,dt2_Max_LAC,dt3_Max_LAC,d2t1_Max_LAC,d2t2_Max_LAC,
    d2t3_Max_LAC,d3t1_Max_LAC,d3t2_Max_LAC,d3t3_Max_LAC,n1_Max_LAC,n2_Max
```

```
_LAC,n3_Max_LAC,dn1_Max_LAC,dn2_Max_LAC,dn3_Max_LAC,d2n1_Max_LAC,
d2n2_Max_LAC,d2n3_Max_LAC,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
ArcLength_Max_LAC,Init_u,SSize_u);

FS_LASC(dummy_u2_x1,dummy_u2_x2,dummy_u2_x3,t1_Max_LASC,t2_Max_LASC,
t3_Max_LASC,dt1_Max_LASC,dt2_Max_LASC,dt3_Max_LASC,d2t1_Max_LASC,
d2t2_Max_LASC,d2t3_Max_LASC,d3t1_Max_LASC,d3t2_Max_LASC,d3t3_Max_LASC,
n1_Max_LASC,n2_Max_LASC,n3_Max_LASC,dn1_Max_LASC,dn2_Max_LASC,dn3_Max
_LASC,d2n1_Max_LASC,d2n2_Max_LASC,d2n3_Max_LASC,b1_Max_LASC,b2_Max_LASC,
b3_Max_LASC,db1_Max_LASC,db2_Max_LASC,db3_Max_LASC,d2b1_Max_LASC,
d2b2_Max_LASC,d2b3_Max_LASC,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,Init_u,SSize_u);


FS_LAC(dummy_v1_x1,dummy_v1_x2,dummy_v1_x3,t1_Min_LAC,t2_Min_LAC,t3_Min
_LAC,dt1_Min_LAC,dt2_Min_LAC,dt3_Min_LAC,d2t1_Min_LAC,d2t2_Min_LAC,
d2t3_Min_LAC,d3t1_Min_LAC,d3t2_Min_LAC,d3t3_Min_LAC,n1_Min_LAC,n2_Min
_LAC,n3_Min_LAC,dn1_Min_LAC,dn2_Min_LAC,dn3_Min_LAC,d2n1_Min_LAC,
d2n2_Min_LAC,d2n3_Min_LAC,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
ArcLength_Min_LAC,Init_v,SSize_v);

FS_LASC(dummy_v2_x1,dummy_v2_x2,dummy_v2_x3,t1_Min_LASC,t2_Min_LASC,
t3_Min_LASC,dt1_Min_LASC,dt2_Min_LASC,dt3_Min_LASC,d2t1_Min_LASC,
d2t2_Min_LASC,d2t3_Min_LASC,d3t1_Min_LASC,d3t2_Min_LASC,d3t3_Min_LASC,
n1_Min_LASC,n2_Min_LASC,n3_Min_LASC,dn1_Min_LASC,dn2_Min_LASC,dn3_Min
_LASC,d2n1_Min_LASC,d2n2_Min_LASC,d2n3_Min_LASC,b1_Min_LASC,b2_Min_LASC,
b3_Min_LASC,db1_Min_LASC,db2_Min_LASC,db3_Min_LASC,d2b1_Min_LASC,
d2b2_Min_LASC,d2b3_Min_LASC,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,Init_v,SSize_v);

Init_u+=h;
Init_v+=h;
pu[j]=Init_u;
pv[j]=Init_v;
u1_x1[j]=dummy_u1_x1;
u1_x2[j]=dummy_u1_x2;
u1_x3[j]=dummy_u1_x3;
u1_t1[j]=t1_Max_LAC;
u1_t2[j]=t2_Max_LAC;
u1_t3[j]=t3_Max_LAC;
u1_dt1[j]=dt1_Max_LAC;
u1_dt2[j]=dt2_Max_LAC;
u1_dt3[j]=dt3_Max_LAC;
u1_d2t1[j]=d2t1_Max_LAC;
u1_d2t2[j]=d2t2_Max_LAC;
u1_d2t3[j]=d2t3_Max_LAC;
u1_d3t1[j]=d3t1_Max_LAC;
```

```
u1_d3t2[j]=d3t2_Max_LAC;
u1_d3t3[j]=d3t3_Max_LAC;
u2_x1[j]=dummy_u2_x1;
u2_x2[j]=dummy_u2_x2;
u2_x3[j]=dummy_u2_x3;
u2_t1[j]=t1_Max_LASC;
u2_t2[j]=t2_Max_LASC;
u2_t3[j]=t3_Max_LASC;
u2_dt1[j]=dt1_Max_LASC;
u2_dt2[j]=dt2_Max_LASC;
u2_dt3[j]=dt3_Max_LASC;
u2_d2t1[j]=d2t1_Max_LASC;
u2_d2t2[j]=d2t2_Max_LASC;
u2_d2t3[j]=d2t3_Max_LASC;
u2_d3t1[j]=d3t1_Max_LASC;
u2_d3t2[j]=d3t2_Max_LASC;
u2_d3t3[j]=d3t3_Max_LASC;
v1_x1[j]=dummy_v1_x1;
v1_x2[j]=dummy_v1_x2;
v1_x3[j]=dummy_v1_x3;
v1_t1[j]=t1_Min_LAC;
v1_t2[j]=t2_Min_LAC;
v1_t3[j]=t3_Min_LAC;
v1_dt1[j]=dt1_Min_LAC;
v1_dt2[j]=dt2_Min_LAC;
v1_dt3[j]=dt3_Min_LAC;
v1_d2t1[j]=d2t1_Min_LAC;
v1_d2t2[j]=d2t2_Min_LAC;
v1_d2t3[j]=d2t3_Min_LAC;
v1_d3t1[j]=d3t1_Min_LAC;
v1_d3t2[j]=d3t2_Min_LAC;
v1_d3t3[j]=d3t3_Min_LAC;
v2_x1[j]=dummy_v2_x1;
v2_x2[j]=dummy_v2_x2;
v2_x3[j]=dummy_v2_x3;
v2_t1[j]=t1_Min_LASC;
v2_t2[j]=t2_Min_LASC;
v2_t3[j]=t3_Min_LASC;
v2_dt1[j]=dt1_Min_LASC;
v2_dt2[j]=dt2_Min_LASC;
v2_dt3[j]=dt3_Min_LASC;
v2_d2t1[j]=d2t1_Min_LASC;
v2_d2t2[j]=d2t2_Min_LASC;
v2_d2t3[j]=d2t3_Min_LASC;
v2_d3t1[j]=d3t1_Min_LASC;
v2_d3t2[j]=d3t2_Min_LASC;
v2_d3t3[j]=d3t3_Min_LASC;
```

```
        }

    }
    else
    {
    if(optimum==min)
        {

        for(int i=1;i<idx+1;i++)
        {

        FS_LAC(dummy_u1_x1,dummy_u1_x2,dummy_u1_x3,t1_Max_LAC,t2_Max_LAC,t3_Max
        _LAC,dt1_Max_LAC,dt2_Max_LAC,dt3_Max_LAC,d2t1_Max_LAC,d2t2_Max_LAC,
        d2t3_Max_LAC,d3t1_Max_LAC,d3t2_Max_LAC,d3t3_Max_LAC,n1_Max_LAC,n2_Max
        _LAC,n3_Max_LAC,dn1_Max_LAC,dn2_Max_LAC,dn3_Max_LAC,d2n1_Max_LAC,
        d2n2_Max_LAC,d2n3_Max_LAC,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
        ArcLength_Max_LAC,Init_u,h);

        FS_LASC(dummy_u2_x1,dummy_u2_x2,dummy_u2_x3,t1_Max_LASC,t2_Max_LASC,
        t3_Max_LASC,dt1_Max_LASC,dt2_Max_LASC,dt3_Max_LASC,d2t1_Max_LASC,
        d2t2_Max_LASC,d2t3_Max_LASC,d3t1_Max_LASC,d3t2_Max_LASC,d3t3_Max_LASC,
        n1_Max_LASC,n2_Max_LASC,n3_Max_LASC,dn1_Max_LASC,dn2_Max_LASC,dn3_Max
        _LASC,d2n1_Max_LASC,d2n2_Max_LASC,d2n3_Max_LASC,b1_Max_LASC,b2_Max_LASC,
        b3_Max_LASC,db1_Max_LASC,db2_Max_LASC,db3_Max_LASC,d2b1_Max_LASC,
        d2b2_Max_LASC,d2b3_Max_LASC,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
        Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,Init_u,h);

        Init_u+=h;
        }
        }
    else if(optimum==max)
        {
        for(int i=1;i<idx+1;i++)
        {

        FS_LAC(dummy_v1_x1,dummy_v1_x2,dummy_v1_x3,t1_Min_LAC,t2_Min_LAC,t3_Min
        _LAC,dt1_Min_LAC,dt2_Min_LAC,dt3_Min_LAC,d2t1_Min_LAC,d2t2_Min_LAC,
        d2t3_Min_LAC,d3t1_Min_LAC,d3t2_Min_LAC,d3t3_Min_LAC,n1_Min_LAC,n2_Min
        _LAC,n3_Min_LAC,dn1_Min_LAC,dn2_Min_LAC,dn3_Min_LAC,d2n1_Min_LAC,
        d2n2_Min_LAC,d2n3_Min_LAC,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
        ArcLength_Min_LAC,Init_v,h);

        FS_LASC(dummy_v2_x1,dummy_v2_x2,dummy_v2_x3,t1_Min_LASC,t2_Min_LASC,
        t3_Min_LASC,dt1_Min_LASC,dt2_Min_LASC,dt3_Min_LASC,d2t1_Min_LASC,
        d2t2_Min_LASC,d2t3_Min_LASC,d3t1_Min_LASC,d3t2_Min_LASC,d3t3_Min_LASC,
        n1_Min_LASC,n2_Min_LASC,n3_Min_LASC,dn1_Min_LASC,dn2_Min_LASC,dn3_Min
```

```
    _LASC,d2n1_Min_LASC,d2n2_Min_LASC,d2n3_Min_LASC,b1_Min_LASC,b2_Min_LASC,
    b3_Min_LASC,db1_Min_LASC,db2_Min_LASC,db3_Min_LASC,d2b1_Min_LASC,
    d2b2_Min_LASC,d2b3_Min_LASC,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
    Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,Init_v,h);
    Init_v+=h;
    }
    }

pu[0]=Init_u;
pv[0]=Init_v;
u1_x1[0]=dummy_u1_x1;
u1_x2[0]=dummy_u1_x2;
u1_x3[0]=dummy_u1_x3;
u1_t1[0]=t1_Max_LAC;
u1_t2[0]=t2_Max_LAC;
u1_t3[0]=t3_Max_LAC;
u1_dt1[0]=dt1_Max_LAC;
u1_dt2[0]=dt2_Max_LAC;
u1_dt3[0]=dt3_Max_LAC;
u1_d2t1[0]=d2t1_Max_LAC;
u1_d2t2[0]=d2t2_Max_LAC;
u1_d2t3[0]=d2t3_Max_LAC;
u1_d3t1[0]=d3t1_Max_LAC;
u1_d3t2[0]=d3t2_Max_LAC;
u1_d3t3[0]=d3t3_Max_LAC;
u2_x1[0]=dummy_u2_x1;
u2_x2[0]=dummy_u2_x2;
u2_x3[0]=dummy_u2_x3;
u2_t1[0]=t1_Max_LASC;
u2_t2[0]=t2_Max_LASC;
u2_t3[0]=t3_Max_LASC;
u2_dt1[0]=dt1_Max_LASC;
u2_dt2[0]=dt2_Max_LASC;
u2_dt3[0]=dt3_Max_LASC;
u2_d2t1[0]=d2t1_Max_LASC;
u2_d2t2[0]=d2t2_Max_LASC;
u2_d2t3[0]=d2t3_Max_LASC;
u2_d3t1[0]=d3t1_Max_LASC;
u2_d3t2[0]=d3t2_Max_LASC;
u2_d3t3[0]=d3t3_Max_LASC;
v1_x1[0]=dummy_v1_x1;
v1_x2[0]=dummy_v1_x2;
v1_x3[0]=dummy_v1_x3;
v1_t1[0]=t1_Min_LAC;
v1_t2[0]=t2_Min_LAC;
v1_t3[0]=t3_Min_LAC;
v1_dt1[0]=dt1_Min_LAC;
```

```
v1_dt2[0]=dt2_Min_LAC;
v1_dt3[0]=dt3_Min_LAC;
v1_d2t1[0]=d2t1_Min_LAC;
v1_d2t2[0]=d2t2_Min_LAC;
v1_d2t3[0]=d2t3_Min_LAC;
v1_d3t1[0]=d3t1_Min_LAC;
v1_d3t2[0]=d3t2_Min_LAC;
v1_d3t3[0]=d3t3_Min_LAC;
v2_x1[0]=dummy_v2_x1;
v2_x2[0]=dummy_v2_x2;
v2_x3[0]=dummy_v2_x3;
v2_t1[0]=t1_Min_LASC;
v2_t2[0]=t2_Min_LASC;
v2_t3[0]=t3_Min_LASC;
v2_dt1[0]=dt1_Min_LASC;
v2_dt2[0]=dt2_Min_LASC;
v2_dt3[0]=dt3_Min_LASC;
v2_d2t1[0]=d2t1_Min_LASC;
v2_d2t2[0]=d2t2_Min_LASC;
v2_d2t3[0]=d2t3_Min_LASC;
v2_d3t1[0]=d3t1_Min_LASC;
v2_d3t2[0]=d3t2_Min_LASC;
v2_d3t3[0]=d3t3_Min_LASC;



for(int j=1;j<n;j++)
    {

    SSize_u=RK4_func(iniDuDs,Init_u,Init_v,dummy_u1_x1,dummy_u1_x2,dummy_u1
    _x3,t1_Max_LAC,t2_Max_LAC,t3_Max_LAC,dt1_Max_LAC,dt2_Max_LAC,dt3_Max_LAC
    ,d2t1_Max_LAC,d2t2_Max_LAC,d2t3_Max_LAC,n1_Max_LAC,n2_Max_LAC,n3_Max_LAC
    ,dn1_Max_LAC,dn2_Max_LAC,dn3_Max_LAC,d2n1_Max_LAC,d2n2_Max_LAC,d2n3_Max
    _LAC,dummy_u2_x1,dummy_u2_x2,dummy_u2_x3,t1_Max_LASC,t2_Max_LASC,t3_Max
    _LASC,dt1_Max_LASC,dt2_Max_LASC,dt3_Max_LASC,d2t1_Max_LASC,d2t2_Max_LASC
    ,d2t3_Max_LASC,n1_Max_LASC,n2_Max_LASC,n3_Max_LASC,dn1_Max_LASC,dn2_Max
    _LASC,dn3_Max_LASC,d2n1_Max_LASC,d2n2_Max_LASC,d2n3_Max_LASC,b1_Max_LASC
    ,b2_Max_LASC,b3_Max_LASC,db1_Max_LASC,db2_Max_LASC,db3_Max_LASC,d2b1_Max
    _LASC,d2b2_Max_LASC,d2b3_Max_LASC,dummy_v1_x1,dummy_v1_x2,dummy_v1_x3,
    t1_Min_LAC,t2_Min_LAC,t3_Min_LAC,dt1_Min_LAC,dt2_Min_LAC,dt3_Min_LAC,
    d2t1_Min_LAC,d2t2_Min_LAC,d2t3_Min_LAC,n1_Min_LAC,n2_Min_LAC,n3_Min_LAC,
    dn1_Min_LAC,dn2_Min_LAC,dn3_Min_LAC,d2n1_Min_LAC,d2n2_Min_LAC,d2n3_Min
    _LAC,dummy_v2_x1,dummy_v2_x2,dummy_v2_x3,t1_Min_LASC,t2_Min_LASC,t3_Min
    _LASC,dt1_Min_LASC,dt2_Min_LASC,dt3_Min_LASC,d2t1_Min_LASC,d2t2_Min_LASC
    ,d2t3_Min_LASC,n1_Min_LASC,n2_Min_LASC,n3_Min_LASC,dn1_Min_LASC,dn2_Min
    _LASC,dn3_Min_LASC,d2n1_Min_LASC,d2n2_Min_LASC,d2n3_Min_LASC,b1_Min_LASC
    ,b2_Min_LASC,b3_Min_LASC,db1_Min_LASC,db2_Min_LASC,db3_Min_LASC,d2b1_Min
```

```
_LASC,d2b2_Min_LASC,d2b3_Min_LASC,optimum,uuu);

SSize_v=RK4_func(iniDvDs,Init_u,Init_v,dummy_u1_x1,dummy_u1_x2,dummy_u1
_x3,t1_Max_LAC,t2_Max_LAC,t3_Max_LAC,dt1_Max_LAC,dt2_Max_LAC,dt3_Max_LAC
,d2t1_Max_LAC,d2t2_Max_LAC,d2t3_Max_LAC,n1_Max_LAC,n2_Max_LAC,n3_Max_LAC
,dn1_Max_LAC,dn2_Max_LAC,dn3_Max_LAC,d2n1_Max_LAC,d2n2_Max_LAC,d2n3_Max
_LAC,dummy_u2_x1,dummy_u2_x2,dummy_u2_x3,t1_Max_LASC,t2_Max_LASC,t3_Max
_LASC,dt1_Max_LASC,dt2_Max_LASC,dt3_Max_LASC,d2t1_Max_LASC,d2t2_Max_LASC
,d2t3_Max_LASC,n1_Max_LASC,n2_Max_LASC,n3_Max_LASC,dn1_Max_LASC,dn2_Max
_LASC,dn3_Max_LASC,d2n1_Max_LASC,d2n2_Max_LASC,d2n3_Max_LASC,b1_Max_LASC
,b2_Max_LASC,b3_Max_LASC,db1_Max_LASC,db2_Max_LASC,db3_Max_LASC,d2b1_Max
_LASC,d2b2_Max_LASC,d2b3_Max_LASC,dummy_v1_x1,dummy_v1_x2,dummy_v1_x3,
t1_Min_LAC,t2_Min_LAC,t3_Min_LAC,dt1_Min_LAC,dt2_Min_LAC,dt3_Min_LAC,
d2t1_Min_LAC,d2t2_Min_LAC,d2t3_Min_LAC,n1_Min_LAC,n2_Min_LAC,n3_Min_LAC,
dn1_Min_LAC,dn2_Min_LAC,dn3_Min_LAC,d2n1_Min_LAC,d2n2_Min_LAC,d2n3_Min
_LAC,dummy_v2_x1,dummy_v2_x2,dummy_v2_x3,t1_Min_LASC,t2_Min_LASC,t3_Min
_LASC,dt1_Min_LASC,dt2_Min_LASC,dt3_Min_LASC,d2t1_Min_LASC,d2t2_Min_LASC
,d2t3_Min_LASC,n1_Min_LASC,n2_Min_LASC,n3_Min_LASC,dn1_Min_LASC,dn2_Min
_LASC,dn3_Min_LASC,d2n1_Min_LASC,d2n2_Min_LASC,d2n3_Min_LASC,b1_Min_LASC
,b2_Min_LASC,b3_Min_LASC,db1_Min_LASC,db2_Min_LASC,db3_Min_LASC,d2b1_Min
_LASC,d2b2_Min_LASC,d2b3_Min_LASC,optimum,vvv);

if((Init_u+SSize_u) > 1.0)
    {
    if(Init_u==1.0)
        {
        SSize_u=0.0;
        SSize_v=0.0;
        }
    else
        {
        ratio=(1.0-Init_u)/SSize_u;
        SSize_u*=ratio;
        SSize_v*=ratio;
        }
    }

if((Init_v+SSize_v) > 1.0)
    {
    if(Init_v==1.0)
        {
        SSize_u=0.0;
        SSize_v=0.0;
        }
    else
        {
        ratio=(1.0-Init_v)/SSize_v;
```

```
        SSize_u*=ratio;
        SSize_v*=ratio;
        }
    }

FS_LAC(dummy_u1_x1,dummy_u1_x2,dummy_u1_x3,t1_Max_LAC,t2_Max_LAC,t3_Max
_LAC,dt1_Max_LAC,dt2_Max_LAC,dt3_Max_LAC,d2t1_Max_LAC,d2t2_Max_LAC,
d2t3_Max_LAC,d3t1_Max_LAC,d3t2_Max_LAC,d3t3_Max_LAC,n1_Max_LAC,n2_Max
_LAC,n3_Max_LAC,dn1_Max_LAC,dn2_Max_LAC,dn3_Max_LAC,d2n1_Max_LAC,
d2n2_Max_LAC,d2n3_Max_LAC,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
ArcLength_Max_LAC,Init_u,SSize_u);

FS_LASC(dummy_u2_x1,dummy_u2_x2,dummy_u2_x3,t1_Max_LASC,t2_Max_LASC,
t3_Max_LASC,dt1_Max_LASC,dt2_Max_LASC,dt3_Max_LASC,d2t1_Max_LASC,
d2t2_Max_LASC,d2t3_Max_LASC,d3t1_Max_LASC,d3t2_Max_LASC,d3t3_Max_LASC,
n1_Max_LASC,n2_Max_LASC,n3_Max_LASC,dn1_Max_LASC,dn2_Max_LASC,dn3_Max
_LASC,d2n1_Max_LASC,d2n2_Max_LASC,d2n3_Max_LASC,b1_Max_LASC,b2_Max_LASC,
b3_Max_LASC,db1_Max_LASC,db2_Max_LASC,db3_Max_LASC,d2b1_Max_LASC,
d2b2_Max_LASC,d2b3_Max_LASC,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,Init_u,SSize_u);


FS_LAC(dummy_v1_x1,dummy_v1_x2,dummy_v1_x3,t1_Min_LAC,t2_Min_LAC,t3_Min
_LAC,dt1_Min_LAC,dt2_Min_LAC,dt3_Min_LAC,d2t1_Min_LAC,d2t2_Min_LAC,
d2t3_Min_LAC,d3t1_Min_LAC,d3t2_Min_LAC,d3t3_Min_LAC,n1_Min_LAC,n2_Min
_LAC,n3_Min_LAC,dn1_Min_LAC,dn2_Min_LAC,dn3_Min_LAC,d2n1_Min_LAC,
d2n2_Min_LAC,d2n3_Min_LAC,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
ArcLength_Min_LAC,Init_v,SSize_v);

FS_LASC(dummy_v2_x1,dummy_v2_x2,dummy_v2_x3,t1_Min_LASC,t2_Min_LASC,
t3_Min_LASC,dt1_Min_LASC,dt2_Min_LASC,dt3_Min_LASC,d2t1_Min_LASC,
d2t2_Min_LASC,d2t3_Min_LASC,d3t1_Min_LASC,d3t2_Min_LASC,d3t3_Min_LASC,
n1_Min_LASC,n2_Min_LASC,n3_Min_LASC,dn1_Min_LASC,dn2_Min_LASC,dn3_Min
_LASC,d2n1_Min_LASC,d2n2_Min_LASC,d2n3_Min_LASC,b1_Min_LASC,b2_Min_LASC,
b3_Min_LASC,db1_Min_LASC,db2_Min_LASC,db3_Min_LASC,d2b1_Min_LASC,
d2b2_Min_LASC,d2b3_Min_LASC,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,Init_v,SSize_v);

Init_u+=SSize_u;
Init_v+=SSize_v;
pu[j]=Init_u;
pv[j]=Init_v;
u1_x1[j]=dummy_u1_x1;
u1_x2[j]=dummy_u1_x2;
u1_x3[j]=dummy_u1_x3;
u1_t1[j]=t1_Max_LAC;
u1_t2[j]=t2_Max_LAC;
```

```
u1_t3[j]=t3_Max_LAC;
u1_dt1[j]=dt1_Max_LAC;
u1_dt2[j]=dt2_Max_LAC;
u1_dt3[j]=dt3_Max_LAC;
u1_d2t1[j]=d2t1_Max_LAC;
u1_d2t2[j]=d2t2_Max_LAC;
u1_d2t3[j]=d2t3_Max_LAC;
u1_d3t1[j]=d3t1_Max_LAC;
u1_d3t2[j]=d3t2_Max_LAC;
u1_d3t3[j]=d3t3_Max_LAC;
u2_x1[j]=dummy_u2_x1;
u2_x2[j]=dummy_u2_x2;
u2_x3[j]=dummy_u2_x3;
u2_t1[j]=t1_Max_LASC;
u2_t2[j]=t2_Max_LASC;
u2_t3[j]=t3_Max_LASC;
u2_dt1[j]=dt1_Max_LASC;
u2_dt2[j]=dt2_Max_LASC;
u2_dt3[j]=dt3_Max_LASC;
u2_d2t1[j]=d2t1_Max_LASC;
u2_d2t2[j]=d2t2_Max_LASC;
u2_d2t3[j]=d2t3_Max_LASC;
u2_d3t1[j]=d3t1_Max_LASC;
u2_d3t2[j]=d3t2_Max_LASC;
u2_d3t3[j]=d3t3_Max_LASC;
v1_x1[j]=dummy_v1_x1;
v1_x2[j]=dummy_v1_x2;
v1_x3[j]=dummy_v1_x3;
v1_t1[j]=t1_Min_LAC;
v1_t2[j]=t2_Min_LAC;
v1_t3[j]=t3_Min_LAC;
v1_dt1[j]=dt1_Min_LAC;
v1_dt2[j]=dt2_Min_LAC;
v1_dt3[j]=dt3_Min_LAC;
v1_d2t1[j]=d2t1_Min_LAC;
v1_d2t2[j]=d2t2_Min_LAC;
v1_d2t3[j]=d2t3_Min_LAC;
v1_d3t1[j]=d3t1_Min_LAC;
v1_d3t2[j]=d3t2_Min_LAC;
v1_d3t3[j]=d3t3_Min_LAC;
v2_x1[j]=dummy_v2_x1;
v2_x2[j]=dummy_v2_x2;
v2_x3[j]=dummy_v2_x3;
v2_t1[j]=t1_Min_LASC;
v2_t2[j]=t2_Min_LASC;
v2_t3[j]=t3_Min_LASC;
v2_dt1[j]=dt1_Min_LASC;
```

```
    v2_dt2[j]=dt2_Min_LASC;
    v2_dt3[j]=dt3_Min_LASC;
    v2_d2t1[j]=d2t1_Min_LASC;
    v2_d2t2[j]=d2t2_Min_LASC;
    v2_d2t3[j]=d2t3_Min_LASC;
    v2_d3t1[j]=d3t1_Min_LASC;
    v2_d3t2[j]=d3t2_Min_LASC;
    v2_d3t3[j]=d3t3_Min_LASC;

    }
}

}

__device__ void Integration2( float pu[], float pv[], float u1_x1[], float
    u1_x2[], float u1_x3[], float u1_t1[], float u1_t2[], float u1_t3[],
    float u1_n1[], float u1_n2[], float u1_n3[], float u1_b1[], float
    u1_b2[], float u1_b3[], float u2_x1[], float u2_x2[], float u2_x3[],
    float u2_t1[], float u2_t2[], float u2_t3[], float u2_n1[], float
    u2_n2[], float u2_n3[], float u2_b1[], float u2_b2[], float
    u2_b3[], float v1_x1[], float v1_x2[], float v1_x3[], float
    v1_t1[], float v1_t2[], float v1_t3[], float v1_n1[], float
    v1_n2[], float v1_n3[], float v1_b1[], float v1_b2[], float
    v1_b3[], float v2_x1[], float v2_x2[], float v2_x3[], float
    v2_t1[], float v2_t2[], float v2_t3[], float v2_n1[], float
    v2_n2[], float v2_n3[], float v2_b1[], float v2_b2[], float v2_b3[])
{
float t1_Max_LAC=-1*ScaleToOrigin_Max_LAC*ArcLength_Max_LAC,
    t2_Max_LAC=0, t3_Max_LAC=0;
float n1_Max_LAC=0, n2_Max_LAC=0,
    n3_Max_LAC=1*ScaleToOrigin_Max_LAC*ArcLength_Max_LAC;
float t1_Min_LAC=0, t2_Min_LAC=-1*ScaleToOrigin_Min_LAC*ArcLength_Min_LAC,
    t3_Min_LAC=0;
float n1_Min_LAC=0, n2_Min_LAC=0,
    n3_Min_LAC=-1*ScaleToOrigin_Min_LAC*ArcLength_Min_LAC;

float t1_Max_LASC=-1*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC,
    t2_Max_LASC=0, t3_Max_LASC=0;
float n1_Max_LASC=0,
    n2_Max_LASC=-0.5930588126117428*ScaleToOrigin_Max_LASC*ArcLength_Max
    _LASC,
    n3_Max_LASC=0.8051591425200051*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC;
float b1_Max_LASC=0,
    b2_Max_LASC=0.8051591425200051*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC
    ,
    b3_Max_LASC=0.5930588126117428*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC;
```

```
float t1_Min_LASC=0,
    t2_Min_LASC=-1*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC, t3_Min_LASC=0;
float
    n1_Min_LASC=-0.5930588126117428*ScaleToOrigin_Min_LASC*ArcLength_Min
    _LASC, n2_Min_LASC=0,
    n3_Min_LASC=-0.8051591425200051*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC;
float
    b1_Min_LASC=0.8051591425200051*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC
    , b2_Min_LASC=0,
    b3_Min_LASC=-0.5930588126117428*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC;

float Init_u=0.0, Init_v=0.0;

float dt1_Max_LAC, dt2_Max_LAC, dt3_Max_LAC;
float d2t1_Max_LAC, d2t2_Max_LAC, d2t3_Max_LAC;
float d3t1_Max_LAC, d3t2_Max_LAC, d3t3_Max_LAC;
float dt1_Min_LAC, dt2_Min_LAC, dt3_Min_LAC;
float d2t1_Min_LAC, d2t2_Min_LAC, d2t3_Min_LAC;
float d3t1_Min_LAC, d3t2_Min_LAC, d3t3_Min_LAC;
float dn1_Max_LAC, dn2_Max_LAC, dn3_Max_LAC;
float d2n1_Max_LAC, d2n2_Max_LAC, d2n3_Max_LAC;
float dn1_Min_LAC, dn2_Min_LAC, dn3_Min_LAC;
float d2n1_Min_LAC, d2n2_Min_LAC, d2n3_Min_LAC;

float dt1_Max_LASC, dt2_Max_LASC, dt3_Max_LASC;
float d2t1_Max_LASC, d2t2_Max_LASC, d2t3_Max_LASC;
float d3t1_Max_LASC, d3t2_Max_LASC, d3t3_Max_LASC;
float dt1_Min_LASC, dt2_Min_LASC, dt3_Min_LASC;
float d2t1_Min_LASC, d2t2_Min_LASC, d2t3_Min_LASC;
float d3t1_Min_LASC, d3t2_Min_LASC, d3t3_Min_LASC;
float dn1_Max_LASC, dn2_Max_LASC, dn3_Max_LASC;
float d2n1_Max_LASC, d2n2_Max_LASC, d2n3_Max_LASC;
float dn1_Min_LASC, dn2_Min_LASC, dn3_Min_LASC;
float d2n1_Min_LASC, d2n2_Min_LASC, d2n3_Min_LASC;
float db1_Max_LASC, db2_Max_LASC, db3_Max_LASC;
float d2b1_Max_LASC, d2b2_Max_LASC, d2b3_Max_LASC;
float db1_Min_LASC, db2_Min_LASC, db3_Min_LASC;
float d2b1_Min_LASC, d2b2_Min_LASC, d2b3_Min_LASC;

float dummy_u1_x1=Point00x, dummy_u1_x2=Point00y, dummy_u1_x3=Point00z;
float dummy_u2_x1=Point30x, dummy_u2_x2=Point30y, dummy_u2_x3=Point30z;
float dummy_v1_x1=Point00x, dummy_v1_x2=Point00y, dummy_v1_x3=Point00z;
float dummy_v2_x1=Point03x, dummy_v2_x2=Point03y, dummy_v2_x3=Point03z;
float SSize_u=0, SSize_v=0, ratio;

InitialFS_LAC(t1_Max_LAC,t2_Max_LAC,t3_Max_LAC,dt1_Max_LAC,dt2_Max_LAC,dt3_Max
    _LAC,d2t1_Max_LAC,d2t2_Max_LAC,d2t3_Max_LAC,d3t1_Max_LAC,d3t2_Max_LAC,
```

```
    d3t3_Max_LAC,n1_Max_LAC,n2_Max_LAC,n3_Max_LAC,dn1_Max_LAC,dn2_Max_LAC,
    dn3_Max_LAC,d2n1_Max_LAC,d2n2_Max_LAC,d2n3_Max_LAC,AlphaLAC,InitRCLAC,
    Lambda_Max_LAC,ArcLength_Max_LAC);

InitialFS_LASC(t1_Max_LASC,t2_Max_LASC,t3_Max_LASC,dt1_Max_LASC,dt2_Max_LASC,
    dt3_Max_LASC,d2t1_Max_LASC,d2t2_Max_LASC,d2t3_Max_LASC,d3t1_Max_LASC,
    d3t2_Max_LASC,d3t3_Max_LASC,n1_Max_LASC,n2_Max_LASC,n3_Max_LASC,dn1_Max
    _LASC,dn2_Max_LASC,dn3_Max_LASC,d2n1_Max_LASC,d2n2_Max_LASC,d2n3_Max
    _LASC,b1_Max_LASC,b2_Max_LASC,b3_Max_LASC,db1_Max_LASC,db2_Max_LASC,
    db3_Max_LASC,d2b1_Max_LASC,d2b2_Max_LASC,d2b3_Max_LASC,AlphaLAC,BetaLASC
    ,InitRCLAC,OmegaLASC,Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC)
    ;

InitialFS_LAC(t1_Min_LAC,t2_Min_LAC,t3_Min_LAC,dt1_Min_LAC,dt2_Min_LAC,dt3_Min
    _LAC,d2t1_Min_LAC,d2t2_Min_LAC,d2t3_Min_LAC,d3t1_Min_LAC,d3t2_Min_LAC,
    d3t3_Min_LAC,n1_Min_LAC,n2_Min_LAC,n3_Min_LAC,dn1_Min_LAC,dn2_Min_LAC,
    dn3_Min_LAC,d2n1_Min_LAC,d2n2_Min_LAC,d2n3_Min_LAC,AlphaLAC,InitRCLAC,
    Lambda_Min_LAC,ArcLength_Min_LAC);

InitialFS_LASC(t1_Min_LASC,t2_Min_LASC,t3_Min_LASC,dt1_Min_LASC,dt2_Min_LASC,
    dt3_Min_LASC,d2t1_Min_LASC,d2t2_Min_LASC,d2t3_Min_LASC,d3t1_Min_LASC,
    d3t2_Min_LASC,d3t3_Min_LASC,n1_Min_LASC,n2_Min_LASC,n3_Min_LASC,dn1_Min
    _LASC,dn2_Min_LASC,dn3_Min_LASC,d2n1_Min_LASC,d2n2_Min_LASC,d2n3_Min
    _LASC,b1_Min_LASC,b2_Min_LASC,b3_Min_LASC,db1_Min_LASC,db2_Min_LASC,
    db3_Min_LASC,d2b1_Min_LASC,d2b2_Min_LASC,d2b3_Min_LASC,AlphaLAC,BetaLASC
    ,InitRCLAC,OmegaLASC,Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC)
    ;


pu[0]=Init_u;
pv[0]=Init_v;
u1_x1[0]=dummy_u1_x1;
u1_x2[0]=dummy_u1_x2;
u1_x3[0]=dummy_u1_x3;
u1_t1[0]=t1_Max_LAC/sqrt(pow(t1_Max_LAC,2)+pow(t2_Max_LAC,2)+pow(t3_Max_LAC,2))
    ;
u1_t2[0]=t2_Max_LAC/sqrt(pow(t1_Max_LAC,2)+pow(t2_Max_LAC,2)+pow(t3_Max
    _LAC,2));
u1_t3[0]=t3_Max_LAC/sqrt(pow(t1_Max_LAC,2)+pow(t2_Max_LAC,2)+pow(t3_Max
    _LAC,2));
u1_n1[0]=n1_Max_LAC/sqrt(pow(n1_Max_LAC,2)+pow(n2_Max_LAC,2)+pow(n3_Max
    _LAC,2));
u1_n2[0]=n2_Max_LAC/sqrt(pow(n1_Max_LAC,2)+pow(n2_Max_LAC,2)+pow(n3_Max
    _LAC,2));
u1_n3[0]=n3_Max_LAC/sqrt(pow(n1_Max_LAC,2)+pow(n2_Max_LAC,2)+pow(n3_Max
    _LAC,2));
u1_b1[0]=0;
```

```
u1_b2[0]=1.0;
u1_b3[0]=0;
u2_x1[0]=dummy_u2_x1;
u2_x2[0]=dummy_u2_x2;
u2_x3[0]=dummy_u2_x3;
u2_t1[0]=t1_Max_LASC/sqrt(pow(t1_Max_LASC,2)+pow(t2_Max_LASC,2)+pow(t3_Max_LASC
     ,2));
u2_t2[0]=t2_Max_LASC/sqrt(pow(t1_Max_LASC,2)+pow(t2_Max_LASC,2)+pow(
     t3_Max_LASC,2));
u2_t3[0]=t3_Max_LASC/sqrt(pow(t1_Max_LASC,2)+pow(t2_Max_LASC,2)+pow(
     t3_Max_LASC,2));
u2_n1[0]=n1_Max_LASC/sqrt(pow(n1_Max_LASC,2)+pow(n2_Max_LASC,2)+pow(
     n3_Max_LASC,2));
u2_n2[0]=n2_Max_LASC/sqrt(pow(n1_Max_LASC,2)+pow(n2_Max_LASC,2)+pow(
     n3_Max_LASC,2));
u2_n3[0]=n3_Max_LASC/sqrt(pow(n1_Max_LASC,2)+pow(n2_Max_LASC,2)+pow(
     n3_Max_LASC,2));
u2_b1[0]=b1_Max_LASC/sqrt(pow(b1_Max_LASC,2)+pow(b2_Max_LASC,2)+pow(
     b3_Max_LASC,2));
u2_b2[0]=b2_Max_LASC/sqrt(pow(b1_Max_LASC,2)+pow(b2_Max_LASC,2)+pow(
     b3_Max_LASC,2));
u2_b3[0]=b3_Max_LASC/sqrt(pow(b1_Max_LASC,2)+pow(b2_Max_LASC,2)+pow(
     b3_Max_LASC,2));
v1_x1[0]=dummy_v1_x1;
v1_x2[0]=dummy_v1_x2;
v1_x3[0]=dummy_v1_x3;
v1_t1[0]=t1_Min_LAC/sqrt(pow(t1_Min_LAC,2)+pow(t2_Min_LAC,2)+pow(t3_Min_LAC,2))
     ;
v1_t2[0]=t2_Min_LAC/sqrt(pow(t1_Min_LAC,2)+pow(t2_Min_LAC,2)+pow(t3_Min
     _LAC,2));
v1_t3[0]=t3_Min_LAC/sqrt(pow(t1_Min_LAC,2)+pow(t2_Min_LAC,2)+pow(t3_Min
     _LAC,2));
v1_n1[0]=n1_Min_LAC/sqrt(pow(n1_Min_LAC,2)+pow(n2_Min_LAC,2)+pow(n3_Min
     _LAC,2));
v1_n2[0]=n2_Min_LAC/sqrt(pow(n1_Min_LAC,2)+pow(n2_Min_LAC,2)+pow(n3_Min
     _LAC,2));
v1_n3[0]=n3_Min_LAC/sqrt(pow(n1_Min_LAC,2)+pow(n2_Min_LAC,2)+pow(n3_Min
     _LAC,2));
v1_b1[0]=1.0;
v1_b2[0]=0;
v1_b3[0]=0;
v2_x1[0]=dummy_v2_x1;
v2_x2[0]=dummy_v2_x2;
v2_x3[0]=dummy_v2_x3;
v2_t1[0]=t1_Min_LASC/sqrt(pow(t1_Min_LASC,2)+pow(t2_Min_LASC,2)+pow(t3_Min_LASC
     ,2));
v2_t2[0]=t2_Min_LASC/sqrt(pow(t1_Min_LASC,2)+pow(t2_Min_LASC,2)+pow(
```

```
    t3_Min_LASC,2));
v2_t3[0]=t3_Min_LASC/sqrt(pow(t1_Min_LASC,2)+pow(t2_Min_LASC,2)+pow(
    t3_Min_LASC,2));
v2_n1[0]=n1_Min_LASC/sqrt(pow(n1_Min_LASC,2)+pow(n2_Min_LASC,2)+pow(
    n3_Min_LASC,2));
v2_n2[0]=n2_Min_LASC/sqrt(pow(n1_Min_LASC,2)+pow(n2_Min_LASC,2)+pow(
    n3_Min_LASC,2));
v2_n3[0]=n3_Min_LASC/sqrt(pow(n1_Min_LASC,2)+pow(n2_Min_LASC,2)+pow(
    n3_Min_LASC,2));
v2_b1[0]=b1_Min_LASC/sqrt(pow(b1_Min_LASC,2)+pow(b2_Min_LASC,2)+pow(
    b3_Min_LASC,2));
v2_b2[0]=b2_Min_LASC/sqrt(pow(b1_Min_LASC,2)+pow(b2_Min_LASC,2)+pow(
    b3_Min_LASC,2));
v2_b3[0]=b3_Min_LASC/sqrt(pow(b1_Min_LASC,2)+pow(b2_Min_LASC,2)+pow(
    b3_Min_LASC,2));


for(int j=1;j<n;j++)
    {
    SSize_u=h;
    SSize_v=h;
    if((Init_u+SSize_u) > 1.0)
        {
        if(Init_u==1.0)
            {
            SSize_u=0.0;
            SSize_v=0.0;
            }
        else
            {
            ratio=(1.0-Init_u)/SSize_u;
            SSize_u*=ratio;
            SSize_v*=ratio;
            }
        }

    if((Init_v+SSize_v) > 1.0)
        {
        if(Init_v==1.0)
            {
            SSize_u=0.0;
            SSize_v=0.0;
            }
        else
            {
            ratio=(1.0-Init_v)/SSize_v;
            SSize_u*=ratio;
```

```
        SSize_v*=ratio;
        }
    }

FS_LAC(dummy_u1_x1,dummy_u1_x2,dummy_u1_x3,t1_Max_LAC,t2_Max_LAC,t3_Max
_LAC,dt1_Max_LAC,dt2_Max_LAC,dt3_Max_LAC,d2t1_Max_LAC,d2t2_Max_LAC,
d2t3_Max_LAC,d3t1_Max_LAC,d3t2_Max_LAC,d3t3_Max_LAC,n1_Max_LAC,n2_Max
_LAC,n3_Max_LAC,dn1_Max_LAC,dn2_Max_LAC,dn3_Max_LAC,d2n1_Max_LAC,
d2n2_Max_LAC,d2n3_Max_LAC,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
ArcLength_Max_LAC,Init_u,SSize_u);

FS_LASC(dummy_u2_x1,dummy_u2_x2,dummy_u2_x3,t1_Max_LASC,t2_Max_LASC,
t3_Max_LASC,dt1_Max_LASC,dt2_Max_LASC,dt3_Max_LASC,d2t1_Max_LASC,
d2t2_Max_LASC,d2t3_Max_LASC,d3t1_Max_LASC,d3t2_Max_LASC,d3t3_Max_LASC,
n1_Max_LASC,n2_Max_LASC,n3_Max_LASC,dn1_Max_LASC,dn2_Max_LASC,dn3_Max
_LASC,d2n1_Max_LASC,d2n2_Max_LASC,d2n3_Max_LASC,b1_Max_LASC,b2_Max_LASC,
b3_Max_LASC,db1_Max_LASC,db2_Max_LASC,db3_Max_LASC,d2b1_Max_LASC,
d2b2_Max_LASC,d2b3_Max_LASC,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,Init_u,SSize_u);


FS_LAC(dummy_v1_x1,dummy_v1_x2,dummy_v1_x3,t1_Min_LAC,t2_Min_LAC,t3_Min
_LAC,dt1_Min_LAC,dt2_Min_LAC,dt3_Min_LAC,d2t1_Min_LAC,d2t2_Min_LAC,
d2t3_Min_LAC,d3t1_Min_LAC,d3t2_Min_LAC,d3t3_Min_LAC,n1_Min_LAC,n2_Min
_LAC,n3_Min_LAC,dn1_Min_LAC,dn2_Min_LAC,dn3_Min_LAC,d2n1_Min_LAC,
d2n2_Min_LAC,d2n3_Min_LAC,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
ArcLength_Min_LAC,Init_v,SSize_v);

FS_LASC(dummy_v2_x1,dummy_v2_x2,dummy_v2_x3,t1_Min_LASC,t2_Min_LASC,
t3_Min_LASC,dt1_Min_LASC,dt2_Min_LASC,dt3_Min_LASC,d2t1_Min_LASC,
d2t2_Min_LASC,d2t3_Min_LASC,d3t1_Min_LASC,d3t2_Min_LASC,d3t3_Min_LASC,
n1_Min_LASC,n2_Min_LASC,n3_Min_LASC,dn1_Min_LASC,dn2_Min_LASC,dn3_Min
_LASC,d2n1_Min_LASC,d2n2_Min_LASC,d2n3_Min_LASC,b1_Min_LASC,b2_Min_LASC,
b3_Min_LASC,db1_Min_LASC,db2_Min_LASC,db3_Min_LASC,d2b1_Min_LASC,
d2b2_Min_LASC,d2b3_Min_LASC,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,Init_v,SSize_v);

Init_u+=h;
Init_v+=h;
pu[j]=Init_u;
pv[j]=Init_v;
u1_x1[j]=dummy_u1_x1;
u1_x2[j]=dummy_u1_x2;
u1_x3[j]=dummy_u1_x3;

u1_t1[j]=t1_Max_LAC/sqrt(pow(t1_Max_LAC,2)+pow(t2_Max_LAC,2)+pow(t3_Max
_LAC,2));
```

```
u1_t2[j]=t2_Max_LAC/sqrt(pow(t1_Max_LAC,2)+pow(t2_Max_LAC,2)+pow(t3_Max
_LAC,2));

u1_t3[j]=t3_Max_LAC/sqrt(pow(t1_Max_LAC,2)+pow(t2_Max_LAC,2)+pow(t3_Max
_LAC,2));

u1_n1[j]=n1_Max_LAC/sqrt(pow(n1_Max_LAC,2)+pow(n2_Max_LAC,2)+pow(n3_Max
_LAC,2));

u1_n2[j]=n2_Max_LAC/sqrt(pow(n1_Max_LAC,2)+pow(n2_Max_LAC,2)+pow(n3_Max
_LAC,2));

u1_n3[j]=n3_Max_LAC/sqrt(pow(n1_Max_LAC,2)+pow(n2_Max_LAC,2)+pow(n3_Max
_LAC,2));
u1_b1[j]=0;
u1_b2[j]=1.0;
u1_b3[j]=0;
u2_x1[j]=dummy_u2_x1;
u2_x2[j]=dummy_u2_x2;
u2_x3[j]=dummy_u2_x3;

u2_t1[j]=t1_Max_LASC/sqrt(pow(t1_Max_LASC,2)+pow(t2_Max_LASC,2)+pow(
t3_Max_LASC,2));

u2_t2[j]=t2_Max_LASC/sqrt(pow(t1_Max_LASC,2)+pow(t2_Max_LASC,2)+pow(
t3_Max_LASC,2));

u2_t3[j]=t3_Max_LASC/sqrt(pow(t1_Max_LASC,2)+pow(t2_Max_LASC,2)+pow(
t3_Max_LASC,2));

u2_n1[j]=n1_Max_LASC/sqrt(pow(n1_Max_LASC,2)+pow(n2_Max_LASC,2)+pow(
n3_Max_LASC,2));

u2_n2[j]=n2_Max_LASC/sqrt(pow(n1_Max_LASC,2)+pow(n2_Max_LASC,2)+pow(
n3_Max_LASC,2));

u2_n3[j]=n3_Max_LASC/sqrt(pow(n1_Max_LASC,2)+pow(n2_Max_LASC,2)+pow(
n3_Max_LASC,2));

u2_b1[j]=b1_Max_LASC/sqrt(pow(b1_Max_LASC,2)+pow(b2_Max_LASC,2)+pow(
b3_Max_LASC,2));

u2_b2[j]=b2_Max_LASC/sqrt(pow(b1_Max_LASC,2)+pow(b2_Max_LASC,2)+pow(
b3_Max_LASC,2));

u2_b3[j]=b3_Max_LASC/sqrt(pow(b1_Max_LASC,2)+pow(b2_Max_LASC,2)+pow(
```

```
b3_Max_LASC,2));
v1_x1[j]=dummy_v1_x1;
v1_x2[j]=dummy_v1_x2;
v1_x3[j]=dummy_v1_x3;

v1_t1[j]=t1_Min_LAC/sqrt(pow(t1_Min_LAC,2)+pow(t2_Min_LAC,2)+pow(t3_Min
_LAC,2));

v1_t2[j]=t2_Min_LAC/sqrt(pow(t1_Min_LAC,2)+pow(t2_Min_LAC,2)+pow(t3_Min
_LAC,2));

v1_t3[j]=t3_Min_LAC/sqrt(pow(t1_Min_LAC,2)+pow(t2_Min_LAC,2)+pow(t3_Min
_LAC,2));

v1_n1[j]=n1_Min_LAC/sqrt(pow(n1_Min_LAC,2)+pow(n2_Min_LAC,2)+pow(n3_Min
_LAC,2));

v1_n2[j]=n2_Min_LAC/sqrt(pow(n1_Min_LAC,2)+pow(n2_Min_LAC,2)+pow(n3_Min
_LAC,2));

v1_n3[j]=n3_Min_LAC/sqrt(pow(n1_Min_LAC,2)+pow(n2_Min_LAC,2)+pow(n3_Min
_LAC,2));
v1_b1[j]=1.0;
v1_b2[j]=0;
v1_b3[j]=0;
v2_x1[j]=dummy_v2_x1;
v2_x2[j]=dummy_v2_x2;
v2_x3[j]=dummy_v2_x3;

v2_t1[j]=t1_Min_LASC/sqrt(pow(t1_Min_LASC,2)+pow(t2_Min_LASC,2)+pow(
t3_Min_LASC,2));

v2_t2[j]=t2_Min_LASC/sqrt(pow(t1_Min_LASC,2)+pow(t2_Min_LASC,2)+pow(
t3_Min_LASC,2));

v2_t3[j]=t3_Min_LASC/sqrt(pow(t1_Min_LASC,2)+pow(t2_Min_LASC,2)+pow(
t3_Min_LASC,2));

v2_n1[j]=n1_Min_LASC/sqrt(pow(n1_Min_LASC,2)+pow(n2_Min_LASC,2)+pow(
n3_Min_LASC,2));

v2_n2[j]=n2_Min_LASC/sqrt(pow(n1_Min_LASC,2)+pow(n2_Min_LASC,2)+pow(
n3_Min_LASC,2));

v2_n3[j]=n3_Min_LASC/sqrt(pow(n1_Min_LASC,2)+pow(n2_Min_LASC,2)+pow(
n3_Min_LASC,2));
```

```
    v2_b1[j]=b1_Min_LASC/sqrt(pow(b1_Min_LASC,2)+pow(b2_Min_LASC,2)+pow(
    b3_Min_LASC,2));

    v2_b2[j]=b2_Min_LASC/sqrt(pow(b1_Min_LASC,2)+pow(b2_Min_LASC,2)+pow(
    b3_Min_LASC,2));

    v2_b3[j]=b3_Min_LASC/sqrt(pow(b1_Min_LASC,2)+pow(b2_Min_LASC,2)+pow(
    b3_Min_LASC,2));

    }



}

__device__ void Integration3( float pu[], float pv[], float u1_x1[], float
    u1_x2[], float u1_x3[], float u1_t1[], float u1_t2[], float u1_t3[],
    float u1_n1[], float u1_n2[], float u1_n3[], float u1_b1[], float
    u1_b2[], float u1_b3[], float u2_x1[], float u2_x2[], float u2_x3[],
    float u2_t1[], float u2_t2[], float u2_t3[], float u2_n1[], float
    u2_n2[], float u2_n3[], float u2_b1[], float u2_b2[], float
    u2_b3[], float v1_x1[], float v1_x2[], float v1_x3[], float
    v1_t1[], float v1_t2[], float v1_t3[], float v1_n1[], float
    v1_n2[], float v1_n3[], float v1_b1[], float v1_b2[], float
    v1_b3[], float v2_x1[], float v2_x2[], float v2_x3[], float
    v2_t1[], float v2_t2[], float v2_t3[], float v2_n1[], float
    v2_n2[], float v2_n3[], float v2_b1[], float v2_b2[], float v2_b3[])
{
float t1_Max_LAC=-1*ScaleToOrigin_Max_LAC*ArcLength_Max_LAC,
    t2_Max_LAC=0, t3_Max_LAC=0;
float n1_Max_LAC=0, n2_Max_LAC=0,
    n3_Max_LAC=1*ScaleToOrigin_Max_LAC*ArcLength_Max_LAC;
float t1_Min_LAC=0, t2_Min_LAC=-1*ScaleToOrigin_Min_LAC*ArcLength_Min_LAC,
    t3_Min_LAC=0;
float n1_Min_LAC=0, n2_Min_LAC=0,
    n3_Min_LAC=-1*ScaleToOrigin_Min_LAC*ArcLength_Min_LAC;

float t1_Max_LASC=-1*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC,
    t2_Max_LASC=0, t3_Max_LASC=0;
float n1_Max_LASC=0,
    n2_Max_LASC=-0.5930588126117428*ScaleToOrigin_Max_LASC*ArcLength_Max
    _LASC,
    n3_Max_LASC=0.8051591425200051*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC;
float b1_Max_LASC=0,
    b2_Max_LASC=0.8051591425200051*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC
    ,
    b3_Max_LASC=0.5930588126117428*ScaleToOrigin_Max_LASC*ArcLength_Max_LASC;
```

```
float t1_Min_LASC=0,
     t2_Min_LASC=-1*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC, t3_Min_LASC=0;
float
     n1_Min_LASC=-0.5930588126117428*ScaleToOrigin_Min_LASC*ArcLength_Min
     _LASC, n2_Min_LASC=0,
     n3_Min_LASC=-0.8051591425200051*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC;
float
     b1_Min_LASC=0.8051591425200051*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC
     , b2_Min_LASC=0,
     b3_Min_LASC=-0.5930588126117428*ScaleToOrigin_Min_LASC*ArcLength_Min_LASC;

float Init_u=0.0, Init_v=0.0;

float dt1_Max_LAC, dt2_Max_LAC, dt3_Max_LAC;
float d2t1_Max_LAC, d2t2_Max_LAC, d2t3_Max_LAC;
float d3t1_Max_LAC, d3t2_Max_LAC, d3t3_Max_LAC;
float dt1_Min_LAC, dt2_Min_LAC, dt3_Min_LAC;
float d2t1_Min_LAC, d2t2_Min_LAC, d2t3_Min_LAC;
float d3t1_Min_LAC, d3t2_Min_LAC, d3t3_Min_LAC;
float dn1_Max_LAC, dn2_Max_LAC, dn3_Max_LAC;
float d2n1_Max_LAC, d2n2_Max_LAC, d2n3_Max_LAC;
float dn1_Min_LAC, dn2_Min_LAC, dn3_Min_LAC;
float d2n1_Min_LAC, d2n2_Min_LAC, d2n3_Min_LAC;

float dt1_Max_LASC, dt2_Max_LASC, dt3_Max_LASC;
float d2t1_Max_LASC, d2t2_Max_LASC, d2t3_Max_LASC;
float d3t1_Max_LASC, d3t2_Max_LASC, d3t3_Max_LASC;
float dt1_Min_LASC, dt2_Min_LASC, dt3_Min_LASC;
float d2t1_Min_LASC, d2t2_Min_LASC, d2t3_Min_LASC;
float d3t1_Min_LASC, d3t2_Min_LASC, d3t3_Min_LASC;
float dn1_Max_LASC, dn2_Max_LASC, dn3_Max_LASC;
float d2n1_Max_LASC, d2n2_Max_LASC, d2n3_Max_LASC;
float dn1_Min_LASC, dn2_Min_LASC, dn3_Min_LASC;
float d2n1_Min_LASC, d2n2_Min_LASC, d2n3_Min_LASC;
float db1_Max_LASC, db2_Max_LASC, db3_Max_LASC;
float d2b1_Max_LASC, d2b2_Max_LASC, d2b3_Max_LASC;
float db1_Min_LASC, db2_Min_LASC, db3_Min_LASC;
float d2b1_Min_LASC, d2b2_Min_LASC, d2b3_Min_LASC;

float dummy_u1_x1=Point00x, dummy_u1_x2=Point00y, dummy_u1_x3=Point00z;
float dummy_u2_x1=Point30x, dummy_u2_x2=Point30y, dummy_u2_x3=Point30z;
float dummy_v1_x1=Point00x, dummy_v1_x2=Point00y, dummy_v1_x3=Point00z;
float dummy_v2_x1=Point03x, dummy_v2_x2=Point03y, dummy_v2_x3=Point03z;
float SSize_u=0, SSize_v=0, ratio;

InitialFS_LAC(t1_Max_LAC,t2_Max_LAC,t3_Max_LAC,dt1_Max_LAC,dt2_Max_LAC,dt3_Max
```

```
_LAC,d2t1_Max_LAC,d2t2_Max_LAC,d2t3_Max_LAC,d3t1_Max_LAC,d3t2_Max_LAC,
    d3t3_Max_LAC,n1_Max_LAC,n2_Max_LAC,n3_Max_LAC,dn1_Max_LAC,dn2_Max_LAC,
    dn3_Max_LAC,d2n1_Max_LAC,d2n2_Max_LAC,d2n3_Max_LAC,AlphaLAC,InitRCLAC,
    Lambda_Max_LAC,ArcLength_Max_LAC);

InitialFS_LASC(t1_Max_LASC,t2_Max_LASC,t3_Max_LASC,dt1_Max_LASC,dt2_Max_LASC,
    dt3_Max_LASC,d2t1_Max_LASC,d2t2_Max_LASC,d2t3_Max_LASC,d3t1_Max_LASC,
    d3t2_Max_LASC,d3t3_Max_LASC,n1_Max_LASC,n2_Max_LASC,n3_Max_LASC,dn1_Max
    _LASC,dn2_Max_LASC,dn3_Max_LASC,d2n1_Max_LASC,d2n2_Max_LASC,d2n3_Max
    _LASC,b1_Max_LASC,b2_Max_LASC,b3_Max_LASC,db1_Max_LASC,db2_Max_LASC,
    db3_Max_LASC,d2b1_Max_LASC,d2b2_Max_LASC,d2b3_Max_LASC,AlphaLAC,BetaLASC
    ,InitRCLAC,OmegaLASC,Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC)
    ;

InitialFS_LAC(t1_Min_LAC,t2_Min_LAC,t3_Min_LAC,dt1_Min_LAC,dt2_Min_LAC,dt3_Min
    _LAC,d2t1_Min_LAC,d2t2_Min_LAC,d2t3_Min_LAC,d3t1_Min_LAC,d3t2_Min_LAC,
    d3t3_Min_LAC,n1_Min_LAC,n2_Min_LAC,n3_Min_LAC,dn1_Min_LAC,dn2_Min_LAC,
    dn3_Min_LAC,d2n1_Min_LAC,d2n2_Min_LAC,d2n3_Min_LAC,AlphaLAC,InitRCLAC,
    Lambda_Min_LAC,ArcLength_Min_LAC);

InitialFS_LASC(t1_Min_LASC,t2_Min_LASC,t3_Min_LASC,dt1_Min_LASC,dt2_Min_LASC,
    dt3_Min_LASC,d2t1_Min_LASC,d2t2_Min_LASC,d2t3_Min_LASC,d3t1_Min_LASC,
    d3t2_Min_LASC,d3t3_Min_LASC,n1_Min_LASC,n2_Min_LASC,n3_Min_LASC,dn1_Min
    _LASC,dn2_Min_LASC,dn3_Min_LASC,d2n1_Min_LASC,d2n2_Min_LASC,d2n3_Min
    _LASC,b1_Min_LASC,b2_Min_LASC,b3_Min_LASC,db1_Min_LASC,db2_Min_LASC,
    db3_Min_LASC,d2b1_Min_LASC,d2b2_Min_LASC,d2b3_Min_LASC,AlphaLAC,BetaLASC
    ,InitRCLAC,OmegaLASC,Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC)
    ;


pu[0]=Init_u;
pv[0]=Init_v;
u1_x1[0]=dummy_u1_x1;
u1_x2[0]=dummy_u1_x2;
u1_x3[0]=dummy_u1_x3;
u1_t1[0]=t1_Max_LAC/sqrt(pow(t1_Max_LAC,2)+pow(t2_Max_LAC,2)+pow(t3_Max_LAC,2))
    ;
u1_t2[0]=t2_Max_LAC/sqrt(pow(t1_Max_LAC,2)+pow(t2_Max_LAC,2)+pow(t3_Max
    _LAC,2));
u1_t3[0]=t3_Max_LAC/sqrt(pow(t1_Max_LAC,2)+pow(t2_Max_LAC,2)+pow(t3_Max
    _LAC,2));
u1_n1[0]=n1_Max_LAC/sqrt(pow(n1_Max_LAC,2)+pow(n2_Max_LAC,2)+pow(n3_Max
    _LAC,2));
u1_n2[0]=n2_Max_LAC/sqrt(pow(n1_Max_LAC,2)+pow(n2_Max_LAC,2)+pow(n3_Max
    _LAC,2));
u1_n3[0]=n3_Max_LAC/sqrt(pow(n1_Max_LAC,2)+pow(n2_Max_LAC,2)+pow(n3_Max
    _LAC,2));
```

```
u1_b1[0]=0;
u1_b2[0]=1.0;
u1_b3[0]=0;
u2_x1[0]=dummy_u2_x1;
u2_x2[0]=dummy_u2_x2;
u2_x3[0]=dummy_u2_x3;
u2_t1[0]=t1_Max_LASC/sqrt(pow(t1_Max_LASC,2)+pow(t2_Max_LASC,2)+pow(t3_Max_LASC
     ,2));
u2_t2[0]=t2_Max_LASC/sqrt(pow(t1_Max_LASC,2)+pow(t2_Max_LASC,2)+pow(
     t3_Max_LASC,2));
u2_t3[0]=t3_Max_LASC/sqrt(pow(t1_Max_LASC,2)+pow(t2_Max_LASC,2)+pow(
     t3_Max_LASC,2));
u2_n1[0]=n1_Max_LASC/sqrt(pow(n1_Max_LASC,2)+pow(n2_Max_LASC,2)+pow(
     n3_Max_LASC,2));
u2_n2[0]=n2_Max_LASC/sqrt(pow(n1_Max_LASC,2)+pow(n2_Max_LASC,2)+pow(
     n3_Max_LASC,2));
u2_n3[0]=n3_Max_LASC/sqrt(pow(n1_Max_LASC,2)+pow(n2_Max_LASC,2)+pow(
     n3_Max_LASC,2));
u2_b1[0]=b1_Max_LASC/sqrt(pow(b1_Max_LASC,2)+pow(b2_Max_LASC,2)+pow(
     b3_Max_LASC,2));
u2_b2[0]=b2_Max_LASC/sqrt(pow(b1_Max_LASC,2)+pow(b2_Max_LASC,2)+pow(
     b3_Max_LASC,2));
u2_b3[0]=b3_Max_LASC/sqrt(pow(b1_Max_LASC,2)+pow(b2_Max_LASC,2)+pow(
     b3_Max_LASC,2));
v1_x1[0]=dummy_v1_x1;
v1_x2[0]=dummy_v1_x2;
v1_x3[0]=dummy_v1_x3;
v1_t1[0]=t1_Min_LAC/sqrt(pow(t1_Min_LAC,2)+pow(t2_Min_LAC,2)+pow(t3_Min_LAC,2))
     ;
v1_t2[0]=t2_Min_LAC/sqrt(pow(t1_Min_LAC,2)+pow(t2_Min_LAC,2)+pow(t3_Min
     _LAC,2));
v1_t3[0]=t3_Min_LAC/sqrt(pow(t1_Min_LAC,2)+pow(t2_Min_LAC,2)+pow(t3_Min
     _LAC,2));
v1_n1[0]=n1_Min_LAC/sqrt(pow(n1_Min_LAC,2)+pow(n2_Min_LAC,2)+pow(n3_Min
     _LAC,2));
v1_n2[0]=n2_Min_LAC/sqrt(pow(n1_Min_LAC,2)+pow(n2_Min_LAC,2)+pow(n3_Min
     _LAC,2));
v1_n3[0]=n3_Min_LAC/sqrt(pow(n1_Min_LAC,2)+pow(n2_Min_LAC,2)+pow(n3_Min
     _LAC,2));
v1_b1[0]=1.0;
v1_b2[0]=0;
v1_b3[0]=0;
v2_x1[0]=dummy_v2_x1;
v2_x2[0]=dummy_v2_x2;
v2_x3[0]=dummy_v2_x3;
v2_t1[0]=t1_Min_LASC/sqrt(pow(t1_Min_LASC,2)+pow(t2_Min_LASC,2)+pow(t3_Min_LASC
     ,2));
```

```
v2_t2[0]=t2_Min_LASC/sqrt(pow(t1_Min_LASC,2)+pow(t2_Min_LASC,2)+pow(
     t3_Min_LASC,2));
v2_t3[0]=t3_Min_LASC/sqrt(pow(t1_Min_LASC,2)+pow(t2_Min_LASC,2)+pow(
     t3_Min_LASC,2));
v2_n1[0]=n1_Min_LASC/sqrt(pow(n1_Min_LASC,2)+pow(n2_Min_LASC,2)+pow(
     n3_Min_LASC,2));
v2_n2[0]=n2_Min_LASC/sqrt(pow(n1_Min_LASC,2)+pow(n2_Min_LASC,2)+pow(
     n3_Min_LASC,2));
v2_n3[0]=n3_Min_LASC/sqrt(pow(n1_Min_LASC,2)+pow(n2_Min_LASC,2)+pow(
     n3_Min_LASC,2));
v2_b1[0]=b1_Min_LASC/sqrt(pow(b1_Min_LASC,2)+pow(b2_Min_LASC,2)+pow(
     b3_Min_LASC,2));
v2_b2[0]=b2_Min_LASC/sqrt(pow(b1_Min_LASC,2)+pow(b2_Min_LASC,2)+pow(
     b3_Min_LASC,2));
v2_b3[0]=b3_Min_LASC/sqrt(pow(b1_Min_LASC,2)+pow(b2_Min_LASC,2)+pow(
     b3_Min_LASC,2));


for(int j=1;j<n;j++)
     {
     SSize_u=h;
     SSize_v=h;
     if((Init_u+SSize_u) > 1.0)
          {
          if(Init_u==1.0)
               {
               SSize_u=0.0;
               SSize_v=0.0;
               }
          else
               {
               ratio=(1.0-Init_u)/SSize_u;
               SSize_u*=ratio;
               SSize_v*=ratio;
               }
          }

     if((Init_v+SSize_v) > 1.0)
          {
          if(Init_v==1.0)
               {
               SSize_u=0.0;
               SSize_v=0.0;
               }
          else
               {
               ratio=(1.0-Init_v)/SSize_v;
```

```
        SSize_u*=ratio;
        SSize_v*=ratio;
        }
    }

FS_LAC(dummy_u1_x1,dummy_u1_x2,dummy_u1_x3,t1_Max_LAC,t2_Max_LAC,t3_Max
_LAC,dt1_Max_LAC,dt2_Max_LAC,dt3_Max_LAC,d2t1_Max_LAC,d2t2_Max_LAC,
d2t3_Max_LAC,d3t1_Max_LAC,d3t2_Max_LAC,d3t3_Max_LAC,n1_Max_LAC,n2_Max
_LAC,n3_Max_LAC,dn1_Max_LAC,dn2_Max_LAC,dn3_Max_LAC,d2n1_Max_LAC,
d2n2_Max_LAC,d2n3_Max_LAC,AlphaLAC,InitRCLAC,Lambda_Max_LAC,
ArcLength_Max_LAC,Init_u,SSize_u);

FS_LASC(dummy_u2_x1,dummy_u2_x2,dummy_u2_x3,t1_Max_LASC,t2_Max_LASC,
t3_Max_LASC,dt1_Max_LASC,dt2_Max_LASC,dt3_Max_LASC,d2t1_Max_LASC,
d2t2_Max_LASC,d2t3_Max_LASC,d3t1_Max_LASC,d3t2_Max_LASC,d3t3_Max_LASC,
n1_Max_LASC,n2_Max_LASC,n3_Max_LASC,dn1_Max_LASC,dn2_Max_LASC,dn3_Max
_LASC,d2n1_Max_LASC,d2n2_Max_LASC,d2n3_Max_LASC,b1_Max_LASC,b2_Max_LASC,
b3_Max_LASC,db1_Max_LASC,db2_Max_LASC,db3_Max_LASC,d2b1_Max_LASC,
d2b2_Max_LASC,d2b3_Max_LASC,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
Lambda_Max_LASC,InitRT_Max_LASC,ArcLength_Max_LASC,Init_u,SSize_u);


FS_LAC(dummy_v1_x1,dummy_v1_x2,dummy_v1_x3,t1_Min_LAC,t2_Min_LAC,t3_Min
_LAC,dt1_Min_LAC,dt2_Min_LAC,dt3_Min_LAC,d2t1_Min_LAC,d2t2_Min_LAC,
d2t3_Min_LAC,d3t1_Min_LAC,d3t2_Min_LAC,d3t3_Min_LAC,n1_Min_LAC,n2_Min
_LAC,n3_Min_LAC,dn1_Min_LAC,dn2_Min_LAC,dn3_Min_LAC,d2n1_Min_LAC,
d2n2_Min_LAC,d2n3_Min_LAC,AlphaLAC,InitRCLAC,Lambda_Min_LAC,
ArcLength_Min_LAC,Init_v,SSize_v);

FS_LASC(dummy_v2_x1,dummy_v2_x2,dummy_v2_x3,t1_Min_LASC,t2_Min_LASC,
t3_Min_LASC,dt1_Min_LASC,dt2_Min_LASC,dt3_Min_LASC,d2t1_Min_LASC,
d2t2_Min_LASC,d2t3_Min_LASC,d3t1_Min_LASC,d3t2_Min_LASC,d3t3_Min_LASC,
n1_Min_LASC,n2_Min_LASC,n3_Min_LASC,dn1_Min_LASC,dn2_Min_LASC,dn3_Min
_LASC,d2n1_Min_LASC,d2n2_Min_LASC,d2n3_Min_LASC,b1_Min_LASC,b2_Min_LASC,
b3_Min_LASC,db1_Min_LASC,db2_Min_LASC,db3_Min_LASC,d2b1_Min_LASC,
d2b2_Min_LASC,d2b3_Min_LASC,AlphaLAC,BetaLASC,InitRCLAC,OmegaLASC,
Lambda_Min_LASC,InitRT_Min_LASC,ArcLength_Min_LASC,Init_v,SSize_v);

pu[j]=Init_u;
pv[j]=Init_v;
u1_x1[j]=dummy_u1_x1;
u1_x2[j]=dummy_u1_x2;
u1_x3[j]=dummy_u1_x3;

u1_t1[j]=t1_Max_LAC/sqrt(pow(t1_Max_LAC,2)+pow(t2_Max_LAC,2)+pow(t3_Max
_LAC,2));
```

```
u1_t2[j]=t2_Max_LAC/sqrt(pow(t1_Max_LAC,2)+pow(t2_Max_LAC,2)+pow(t3_Max
_LAC,2));

u1_t3[j]=t3_Max_LAC/sqrt(pow(t1_Max_LAC,2)+pow(t2_Max_LAC,2)+pow(t3_Max
_LAC,2));

u1_n1[j]=n1_Max_LAC/sqrt(pow(n1_Max_LAC,2)+pow(n2_Max_LAC,2)+pow(n3_Max
_LAC,2));

u1_n2[j]=n2_Max_LAC/sqrt(pow(n1_Max_LAC,2)+pow(n2_Max_LAC,2)+pow(n3_Max
_LAC,2));

u1_n3[j]=n3_Max_LAC/sqrt(pow(n1_Max_LAC,2)+pow(n2_Max_LAC,2)+pow(n3_Max
_LAC,2));
u1_b1[j]=0;
u1_b2[j]=1.0;
u1_b3[j]=0;
u2_x1[j]=dummy_u2_x1;
u2_x2[j]=dummy_u2_x2;
u2_x3[j]=dummy_u2_x3;

u2_t1[j]=t1_Max_LASC/sqrt(pow(t1_Max_LASC,2)+pow(t2_Max_LASC,2)+pow(
t3_Max_LASC,2));

u2_t2[j]=t2_Max_LASC/sqrt(pow(t1_Max_LASC,2)+pow(t2_Max_LASC,2)+pow(
t3_Max_LASC,2));

u2_t3[j]=t3_Max_LASC/sqrt(pow(t1_Max_LASC,2)+pow(t2_Max_LASC,2)+pow(
t3_Max_LASC,2));

u2_n1[j]=n1_Max_LASC/sqrt(pow(n1_Max_LASC,2)+pow(n2_Max_LASC,2)+pow(
n3_Max_LASC,2));

u2_n2[j]=n2_Max_LASC/sqrt(pow(n1_Max_LASC,2)+pow(n2_Max_LASC,2)+pow(
n3_Max_LASC,2));

u2_n3[j]=n3_Max_LASC/sqrt(pow(n1_Max_LASC,2)+pow(n2_Max_LASC,2)+pow(
n3_Max_LASC,2));

u2_b1[j]=b1_Max_LASC/sqrt(pow(b1_Max_LASC,2)+pow(b2_Max_LASC,2)+pow(
b3_Max_LASC,2));

u2_b2[j]=b2_Max_LASC/sqrt(pow(b1_Max_LASC,2)+pow(b2_Max_LASC,2)+pow(
b3_Max_LASC,2));

u2_b3[j]=b3_Max_LASC/sqrt(pow(b1_Max_LASC,2)+pow(b2_Max_LASC,2)+pow(
b3_Max_LASC,2));
```

```
v1_x1[j]=dummy_v1_x1;
v1_x2[j]=dummy_v1_x2;
v1_x3[j]=dummy_v1_x3;

v1_t1[j]=t1_Min_LAC/sqrt(pow(t1_Min_LAC,2)+pow(t2_Min_LAC,2)+pow(t3_Min
_LAC,2));

v1_t2[j]=t2_Min_LAC/sqrt(pow(t1_Min_LAC,2)+pow(t2_Min_LAC,2)+pow(t3_Min
_LAC,2));

v1_t3[j]=t3_Min_LAC/sqrt(pow(t1_Min_LAC,2)+pow(t2_Min_LAC,2)+pow(t3_Min
_LAC,2));

v1_n1[j]=n1_Min_LAC/sqrt(pow(n1_Min_LAC,2)+pow(n2_Min_LAC,2)+pow(n3_Min
_LAC,2));

v1_n2[j]=n2_Min_LAC/sqrt(pow(n1_Min_LAC,2)+pow(n2_Min_LAC,2)+pow(n3_Min
_LAC,2));

v1_n3[j]=n3_Min_LAC/sqrt(pow(n1_Min_LAC,2)+pow(n2_Min_LAC,2)+pow(n3_Min
_LAC,2));
v1_b1[j]=1.0;
v1_b2[j]=0;
v1_b3[j]=0;
v2_x1[j]=dummy_v2_x1;
v2_x2[j]=dummy_v2_x2;
v2_x3[j]=dummy_v2_x3;

v2_t1[j]=t1_Min_LASC/sqrt(pow(t1_Min_LASC,2)+pow(t2_Min_LASC,2)+pow(
t3_Min_LASC,2));

v2_t2[j]=t2_Min_LASC/sqrt(pow(t1_Min_LASC,2)+pow(t2_Min_LASC,2)+pow(
t3_Min_LASC,2));

v2_t3[j]=t3_Min_LASC/sqrt(pow(t1_Min_LASC,2)+pow(t2_Min_LASC,2)+pow(
t3_Min_LASC,2));

v2_n1[j]=n1_Min_LASC/sqrt(pow(n1_Min_LASC,2)+pow(n2_Min_LASC,2)+pow(
n3_Min_LASC,2));

v2_n2[j]=n2_Min_LASC/sqrt(pow(n1_Min_LASC,2)+pow(n2_Min_LASC,2)+pow(
n3_Min_LASC,2));

v2_n3[j]=n3_Min_LASC/sqrt(pow(n1_Min_LASC,2)+pow(n2_Min_LASC,2)+pow(
n3_Min_LASC,2));

v2_b1[j]=b1_Min_LASC/sqrt(pow(b1_Min_LASC,2)+pow(b2_Min_LASC,2)+pow(
```

```
        b3_Min_LASC,2));

        v2_b2[j]=b2_Min_LASC/sqrt(pow(b1_Min_LASC,2)+pow(b2_Min_LASC,2)+pow(
        b3_Min_LASC,2));

        v2_b3[j]=b3_Min_LASC/sqrt(pow(b1_Min_LASC,2)+pow(b2_Min_LASC,2)+pow(
        b3_Min_LASC,2));

        }



}
__global__ void fun( float * xvalue, float * yvalue,
    float * zvalue, mint idx, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float pu[n], pv[n];
float u1_x1[n], u1_x2[n], u1_x3[n], u1_t1[n], u1_t2[n],
    u1_t3[n], u1_dt1[n], u1_dt2[n], u1_dt3[n], u1_d2t1[n],
    u1_d2t2[n], u1_d2t3[n], u1_d3t1[n], u1_d3t2[n], u1_d3t3[n];
float u2_x1[n], u2_x2[n], u2_x3[n], u2_t1[n], u2_t2[n], u2_t3[n],
    u2_dt1[n], u2_dt2[n], u2_dt3[n], u2_d2t1[n], u2_d2t2[n],
    u2_d2t3[n], u2_d3t1[n], u2_d3t2[n], u2_d3t3[n];
float v1_x1[n], v1_x2[n], v1_x3[n], v1_t1[n], v1_t2[n], v1_t3[n],
    v1_dt1[n], v1_dt2[n], v1_dt3[n], v1_d2t1[n], v1_d2t2[n],
    v1_d2t3[n], v1_d3t1[n], v1_d3t2[n], v1_d3t3[n];
float v2_x1[n], v2_x2[n], v2_x3[n], v2_t1[n], v2_t2[n], v2_t3[n],
    v2_dt1[n], v2_dt2[n], v2_dt3[n], v2_d2t1[n], v2_d2t2[n],
    v2_d2t3[n], v2_d3t1[n], v2_d3t2[n], v2_d3t3[n];

Integration( pu, pv, u1_x1, u1_x2, u1_x3, u1_t1, u1_t2, u1_t3, u1_dt1, u1_dt2,
    u1_dt3, u1_d2t1, u1_d2t2, u1_d2t3, u1_d3t1, u1_d3t2, u1_d3t3,
    u2_x1, u2_x2, u2_x3, u2_t1, u2_t2, u2_t3, u2_dt1, u2_dt2, u2_dt3,
    u2_d2t1, u2_d2t2, u2_d2t3, u2_d3t1, u2_d3t2, u2_d3t3, v1_x1, v1_x2,
    v1_x3, v1_t1, v1_t2, v1_t3, v1_dt1, v1_dt2, v1_dt3, v1_d2t1,
    v1_d2t2, v1_d2t3, v1_d3t1, v1_d3t2, v1_d3t3, v2_x1, v2_x2,
    v2_x3, v2_t1, v2_t2, v2_t3, v2_dt1, v2_dt2, v2_dt3, v2_d2t1,
    v2_d2t2, v2_d2t3, v2_d3t1, v2_d3t2, v2_d3t3, idx, optimum);

if (idx==0 && optimum==min)
{
if( index < ListSize)
{
    xvalue[index]= v1_x1[index];
    yvalue[index]= v1_x2[index];
    zvalue[index]= v1_x3[index];
```

```
}
}
else if (idx==0 && optimum==max)
{
if( index < ListSize)
{
    xvalue[index]= u1_x1[index];
    yvalue[index]= u1_x2[index];
    zvalue[index]= u1_x3[index];
}
}
else if (idx==1000 && optimum==min)
{
if( index < ListSize)
{
    xvalue[index]= v2_x1[index];
    yvalue[index]= v2_x2[index];
    zvalue[index]= v2_x3[index];
}
}
else if (idx==1000 && optimum==max)
{
if( index < ListSize)
{
    xvalue[index]= u2_x1[index];
    yvalue[index]= u2_x2[index];
    zvalue[index]= u2_x3[index];
}
}
else
{
if( index < ListSize)
{
    xvalue[index]= Coons_x1;
    yvalue[index]= Coons_x2;
    zvalue[index]= Coons_x3;
}
}
}

__global__ void fun2( float * pvalue, mint idx, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float pu[n], pv[n];
float u1_x1[n], u1_x2[n], u1_x3[n], u1_t1[n], u1_t2[n],
    u1_t3[n], u1_dt1[n], u1_dt2[n], u1_dt3[n], u1_d2t1[n],
    u1_d2t2[n], u1_d2t3[n], u1_d3t1[n], u1_d3t2[n], u1_d3t3[n];
```

```
float u2_x1[n], u2_x2[n], u2_x3[n], u2_t1[n], u2_t2[n], u2_t3[n],
    u2_dt1[n], u2_dt2[n], u2_dt3[n], u2_d2t1[n], u2_d2t2[n],
    u2_d2t3[n], u2_d3t1[n], u2_d3t2[n], u2_d3t3[n];
float v1_x1[n], v1_x2[n], v1_x3[n], v1_t1[n], v1_t2[n], v1_t3[n],
    v1_dt1[n], v1_dt2[n], v1_dt3[n], v1_d2t1[n], v1_d2t2[n],
    v1_d2t3[n], v1_d3t1[n], v1_d3t2[n], v1_d3t3[n];
float v2_x1[n], v2_x2[n], v2_x3[n], v2_t1[n], v2_t2[n], v2_t3[n],
    v2_dt1[n], v2_dt2[n], v2_dt3[n], v2_d2t1[n], v2_d2t2[n],
    v2_d2t3[n], v2_d3t1[n], v2_d3t2[n], v2_d3t3[n];

Integration( pu, pv, u1_x1, u1_x2, u1_x3, u1_t1, u1_t2, u1_t3, u1_dt1, u1_dt2,
    u1_dt3, u1_d2t1, u1_d2t2, u1_d2t3, u1_d3t1, u1_d3t2, u1_d3t3,
    u2_x1, u2_x2, u2_x3, u2_t1, u2_t2, u2_t3, u2_dt1, u2_dt2, u2_dt3,
    u2_d2t1, u2_d2t2, u2_d2t3, u2_d3t1, u2_d3t2, u2_d3t3, v1_x1, v1_x2,
    v1_x3, v1_t1, v1_t2, v1_t3, v1_dt1, v1_dt2, v1_dt3, v1_d2t1,
    v1_d2t2, v1_d2t3, v1_d3t1, v1_d3t2, v1_d3t3, v2_x1, v2_x2,
    v2_x3, v2_t1, v2_t2, v2_t3, v2_dt1, v2_dt2, v2_dt3, v2_d2t1,
    v2_d2t2, v2_d2t3, v2_d3t1, v2_d3t2, v2_d3t3, idx, optimum);

if( index < ListSize)
{
    pvalue[index]= Curvecurvature( index, pu, pv, u1_x1, u1_x2, u1_x3, u1_t1,
    u1_t2, u1_t3, u1_dt1, u1_dt2, u1_dt3, u1_d2t1, u1_d2t2, u1_d2t3,
    u1_d3t1, u1_d3t2, u1_d3t3, u2_x1, u2_x2, u2_x3, u2_t1, u2_t2, u2_t3,
    u2_dt1, u2_dt2, u2_dt3, u2_d2t1, u2_d2t2, u2_d2t3, u2_d3t1, u2_d3t2,
    u2_d3t3, v1_x1, v1_x2, v1_x3, v1_t1, v1_t2, v1_t3, v1_dt1, v1_dt2,
    v1_dt3, v1_d2t1, v1_d2t2, v1_d2t3, v1_d3t1, v1_d3t2, v1_d3t3,
    v2_x1, v2_x2, v2_x3, v2_t1, v2_t2, v2_t3, v2_dt1, v2_dt2, v2_dt3,
    v2_d2t1, v2_d2t2, v2_d2t3, v2_d3t1, v2_d3t2, v2_d3t3, optimum);
}

}



__global__ void fun4( float * uvalue, float * vvalue, float * xvalue, float *
    yvalue, float * zvalue, mint idx, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float pu[n], pv[n];
float u1_x1[n], u1_x2[n], u1_x3[n], u1_t1[n], u1_t2[n],
    u1_t3[n], u1_dt1[n], u1_dt2[n], u1_dt3[n], u1_d2t1[n],
    u1_d2t2[n], u1_d2t3[n], u1_d3t1[n], u1_d3t2[n], u1_d3t3[n];
float u2_x1[n], u2_x2[n], u2_x3[n], u2_t1[n], u2_t2[n], u2_t3[n],
    u2_dt1[n], u2_dt2[n], u2_dt3[n], u2_d2t1[n], u2_d2t2[n],
    u2_d2t3[n], u2_d3t1[n], u2_d3t2[n], u2_d3t3[n];
float v1_x1[n], v1_x2[n], v1_x3[n], v1_t1[n], v1_t2[n], v1_t3[n],
    v1_dt1[n], v1_dt2[n], v1_dt3[n], v1_d2t1[n], v1_d2t2[n],
```

```
    v1_d2t3[n], v1_d3t1[n], v1_d3t2[n], v1_d3t3[n];
float v2_x1[n], v2_x2[n], v2_x3[n], v2_t1[n], v2_t2[n], v2_t3[n],
    v2_dt1[n], v2_dt2[n], v2_dt3[n], v2_d2t1[n], v2_d2t2[n],
    v2_d2t3[n], v2_d3t1[n], v2_d3t2[n], v2_d3t3[n];

Integration( pu, pv, u1_x1, u1_x2, u1_x3, u1_t1, u1_t2, u1_t3, u1_dt1, u1_dt2,
    u1_dt3, u1_d2t1, u1_d2t2, u1_d2t3, u1_d3t1, u1_d3t2, u1_d3t3,
    u2_x1, u2_x2, u2_x3, u2_t1, u2_t2, u2_t3, u2_dt1, u2_dt2, u2_dt3,
    u2_d2t1, u2_d2t2, u2_d2t3, u2_d3t1, u2_d3t2, u2_d3t3, v1_x1, v1_x2,
    v1_x3, v1_t1, v1_t2, v1_t3, v1_dt1, v1_dt2, v1_dt3, v1_d2t1,
    v1_d2t2, v1_d2t3, v1_d3t1, v1_d3t2, v1_d3t3, v2_x1, v2_x2,
    v2_x3, v2_t1, v2_t2, v2_t3, v2_dt1, v2_dt2, v2_dt3, v2_d2t1,
    v2_d2t2, v2_d2t3, v2_d3t1, v2_d3t2, v2_d3t3, idx, optimum);

if( index < ListSize)
{
    uvalue[index]= pu[index];
    vvalue[index]= pv[index];
    xvalue[index]= Coons_x1;
    yvalue[index]= Coons_x2;
    zvalue[index]= Coons_x3;
}

}

__global__ void fun6( float * xvalue, float * yvalue,
    float * zvalue, mint idx, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float pu[n], pv[n];
float u1_x1[n], u1_x2[n], u1_x3[n], u1_t1[n], u1_t2[n],
    u1_t3[n], u1_dt1[n], u1_dt2[n], u1_dt3[n], u1_d2t1[n],
    u1_d2t2[n], u1_d2t3[n], u1_d3t1[n], u1_d3t2[n], u1_d3t3[n];
float u1_n1[n], u1_n2[n], u1_n3[n], u1_b1[n], u1_b2[n], u1_b3[n];
float u2_x1[n], u2_x2[n], u2_x3[n], u2_t1[n], u2_t2[n],
    u2_t3[n], u2_dt1[n], u2_dt2[n], u2_dt3[n], u2_d2t1[n],
    u2_d2t2[n], u2_d2t3[n], u2_d3t1[n], u2_d3t2[n], u2_d3t3[n];
float u2_n1[n], u2_n2[n], u2_n3[n], u2_b1[n], u2_b2[n], u2_b3[n];
float v1_x1[n], v1_x2[n], v1_x3[n], v1_t1[n], v1_t2[n],
    v1_t3[n], v1_dt1[n], v1_dt2[n], v1_dt3[n], v1_d2t1[n],
    v1_d2t2[n], v1_d2t3[n], v1_d3t1[n], v1_d3t2[n], v1_d3t3[n];
float v1_n1[n], v1_n2[n], v1_n3[n], v1_b1[n], v1_b2[n], v1_b3[n];
float v2_x1[n], v2_x2[n], v2_x3[n], v2_t1[n], v2_t2[n],
    v2_t3[n], v2_dt1[n], v2_dt2[n], v2_dt3[n], v2_d2t1[n],
    v2_d2t2[n], v2_d2t3[n], v2_d3t1[n], v2_d3t2[n], v2_d3t3[n];
float v2_n1[n], v2_n2[n], v2_n3[n], v2_b1[n], v2_b2[n], v2_b3[n];
float tangv[3];
```

```
if (idx==0 && optimum==min)
{
Integration2( pu, pv, u1_x1, u1_x2, u1_x3, u1_t1, u1_t2, u1_t3, u1_n1, u1_n2,
    u1_n3, u1_b1, u1_b2, u1_b3, u2_x1, u2_x2, u2_x3, u2_t1, u2_t2, u2_t3,
    u2_n1, u2_n2, u2_n3, u2_b1, u2_b2, u2_b3, v1_x1, v1_x2, v1_x3, v1_t1,
    v1_t2, v1_t3, v1_n1, v1_n2, v1_n3, v1_b1, v1_b2, v1_b3, v2_x1, v2_x2,
    v2_x3, v2_t1, v2_t2, v2_t3, v2_n1, v2_n2, v2_n3, v2_b1, v2_b2, v2_b3);

    if( index < ListSize)
    {
    xvalue[index]= v1_t1[index];
    yvalue[index]= v1_t2[index];
    zvalue[index]= v1_t3[index];
    }
}
else if (idx==0 && optimum==max)
{
Integration2( pu, pv, u1_x1, u1_x2, u1_x3, u1_t1, u1_t2, u1_t3, u1_n1, u1_n2,
    u1_n3, u1_b1, u1_b2, u1_b3, u2_x1, u2_x2, u2_x3, u2_t1, u2_t2, u2_t3,
    u2_n1, u2_n2, u2_n3, u2_b1, u2_b2, u2_b3, v1_x1, v1_x2, v1_x3, v1_t1,
    v1_t2, v1_t3, v1_n1, v1_n2, v1_n3, v1_b1, v1_b2, v1_b3, v2_x1, v2_x2,
    v2_x3, v2_t1, v2_t2, v2_t3, v2_n1, v2_n2, v2_n3, v2_b1, v2_b2, v2_b3);

    if( index < ListSize)
    {
    xvalue[index]= u1_t1[index];
    yvalue[index]= u1_t2[index];
    zvalue[index]= u1_t3[index];
    }
}
else if (idx==1000 && optimum==min)
{
Integration2( pu, pv, u1_x1, u1_x2, u1_x3, u1_t1, u1_t2, u1_t3, u1_n1, u1_n2,
    u1_n3, u1_b1, u1_b2, u1_b3, u2_x1, u2_x2, u2_x3, u2_t1, u2_t2, u2_t3,
    u2_n1, u2_n2, u2_n3, u2_b1, u2_b2, u2_b3, v1_x1, v1_x2, v1_x3, v1_t1,
    v1_t2, v1_t3, v1_n1, v1_n2, v1_n3, v1_b1, v1_b2, v1_b3, v2_x1, v2_x2,
    v2_x3, v2_t1, v2_t2, v2_t3, v2_n1, v2_n2, v2_n3, v2_b1, v2_b2, v2_b3);

    if( index < ListSize)
    {
    xvalue[index]= v2_t1[index];
    yvalue[index]= v2_t2[index];
    zvalue[index]= v2_t3[index];
    }
}
else if (idx==1000 && optimum==max)
```

```
{
Integration2( pu, pv, u1_x1, u1_x2, u1_x3, u1_t1, u1_t2, u1_t3, u1_n1, u1_n2,
    u1_n3, u1_b1, u1_b2, u1_b3, u2_x1, u2_x2, u2_x3, u2_t1, u2_t2, u2_t3,
    u2_n1, u2_n2, u2_n3, u2_b1, u2_b2, u2_b3, v1_x1, v1_x2, v1_x3, v1_t1,
    v1_t2, v1_t3, v1_n1, v1_n2, v1_n3, v1_b1, v1_b2, v1_b3, v2_x1, v2_x2,
    v2_x3, v2_t1, v2_t2, v2_t3, v2_n1, v2_n2, v2_n3, v2_b1, v2_b2, v2_b3);

    if( index < ListSize)
    {
    xvalue[index]= u2_t1[index];
    yvalue[index]= u2_t2[index];
    zvalue[index]= u2_t3[index];
    }
}
else
{
Integration( pu, pv, u1_x1, u1_x2, u1_x3, u1_t1, u1_t2, u1_t3, u1_dt1, u1_dt2,
    u1_dt3, u1_d2t1, u1_d2t2, u1_d2t3, u1_d3t1, u1_d3t2, u1_d3t3,
    u2_x1, u2_x2, u2_x3, u2_t1, u2_t2, u2_t3, u2_dt1, u2_dt2, u2_dt3,
    u2_d2t1, u2_d2t2, u2_d2t3, u2_d3t1, u2_d3t2, u2_d3t3, v1_x1, v1_x2,
    v1_x3, v1_t1, v1_t2, v1_t3, v1_dt1, v1_dt2, v1_dt3, v1_d2t1,
    v1_d2t2, v1_d2t3, v1_d3t1, v1_d3t2, v1_d3t3, v2_x1, v2_x2,
    v2_x3, v2_t1, v2_t2, v2_t3, v2_dt1, v2_dt2, v2_dt3, v2_d2t1,
    v2_d2t2, v2_d2t3, v2_d3t1, v2_d3t2, v2_d3t3, idx, optimum);

    if( index < ListSize)
    {

    DSurDs(index,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,u1_dt2,
    u1_dt3,u2_x1,u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,v1_x1,
    v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,v1_dt1,v1_dt2,v1_dt3,v2_x1,v2_x2,v2_x3,
    v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,optimum,tangv);
    xvalue[index]= tangv[0];
    yvalue[index]= tangv[1];
    zvalue[index]= tangv[2];
    }
}


}

__global__ void fun7( float * xvalue, float * yvalue,
    float * zvalue, mint idx, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float pu[n], pv[n];
float u1_x1[n], u1_x2[n], u1_x3[n], u1_t1[n], u1_t2[n],
```

```
    u1_t3[n], u1_dt1[n], u1_dt2[n], u1_dt3[n], u1_d2t1[n],
    u1_d2t2[n], u1_d2t3[n], u1_d3t1[n], u1_d3t2[n], u1_d3t3[n];
float u1_n1[n], u1_n2[n], u1_n3[n], u1_b1[n], u1_b2[n], u1_b3[n];
float u2_x1[n], u2_x2[n], u2_x3[n], u2_t1[n], u2_t2[n],
    u2_t3[n], u2_dt1[n], u2_dt2[n], u2_dt3[n], u2_d2t1[n],
    u2_d2t2[n], u2_d2t3[n], u2_d3t1[n], u2_d3t2[n], u2_d3t3[n];
float u2_n1[n], u2_n2[n], u2_n3[n], u2_b1[n], u2_b2[n], u2_b3[n];
float v1_x1[n], v1_x2[n], v1_x3[n], v1_t1[n], v1_t2[n],
    v1_t3[n], v1_dt1[n], v1_dt2[n], v1_dt3[n], v1_d2t1[n],
    v1_d2t2[n], v1_d2t3[n], v1_d3t1[n], v1_d3t2[n], v1_d3t3[n];
float v1_n1[n], v1_n2[n], v1_n3[n], v1_b1[n], v1_b2[n], v1_b3[n];
float v2_x1[n], v2_x2[n], v2_x3[n], v2_t1[n], v2_t2[n],
    v2_t3[n], v2_dt1[n], v2_dt2[n], v2_dt3[n], v2_d2t1[n],
    v2_d2t2[n], v2_d2t3[n], v2_d3t1[n], v2_d3t2[n], v2_d3t3[n];
float v2_n1[n], v2_n2[n], v2_n3[n], v2_b1[n], v2_b2[n], v2_b3[n];
float norv[3];

if (idx==0 && optimum==min)
{
Integration2( pu, pv, u1_x1, u1_x2, u1_x3, u1_t1, u1_t2, u1_t3, u1_n1, u1_n2,
    u1_n3, u1_b1, u1_b2, u1_b3, u2_x1, u2_x2, u2_x3, u2_t1, u2_t2, u2_t3,
    u2_n1, u2_n2, u2_n3, u2_b1, u2_b2, u2_b3, v1_x1, v1_x2, v1_x3, v1_t1,
    v1_t2, v1_t3, v1_n1, v1_n2, v1_n3, v1_b1, v1_b2, v1_b3, v2_x1, v2_x2,
    v2_x3, v2_t1, v2_t2, v2_t3, v2_n1, v2_n2, v2_n3, v2_b1, v2_b2, v2_b3);

    if( index < ListSize)
    {
    xvalue[index]= v1_n1[index];
    yvalue[index]= v1_n2[index];
    zvalue[index]= v1_n3[index];
    }
}
else if (idx==0 && optimum==max)
{
Integration2( pu, pv, u1_x1, u1_x2, u1_x3, u1_t1, u1_t2, u1_t3, u1_n1, u1_n2,
    u1_n3, u1_b1, u1_b2, u1_b3, u2_x1, u2_x2, u2_x3, u2_t1, u2_t2, u2_t3,
    u2_n1, u2_n2, u2_n3, u2_b1, u2_b2, u2_b3, v1_x1, v1_x2, v1_x3, v1_t1,
    v1_t2, v1_t3, v1_n1, v1_n2, v1_n3, v1_b1, v1_b2, v1_b3, v2_x1, v2_x2,
    v2_x3, v2_t1, v2_t2, v2_t3, v2_n1, v2_n2, v2_n3, v2_b1, v2_b2, v2_b3);

    if( index < ListSize)
    {
    xvalue[index]= u1_n1[index];
    yvalue[index]= u1_n2[index];
    zvalue[index]= u1_n3[index];
    }
}
```

```
else if (idx==1000 && optimum==min)
{
Integration2( pu, pv, u1_x1, u1_x2, u1_x3, u1_t1, u1_t2, u1_t3, u1_n1, u1_n2,
    u1_n3, u1_b1, u1_b2, u1_b3, u2_x1, u2_x2, u2_x3, u2_t1, u2_t2, u2_t3,
    u2_n1, u2_n2, u2_n3, u2_b1, u2_b2, u2_b3, v1_x1, v1_x2, v1_x3, v1_t1,
    v1_t2, v1_t3, v1_n1, v1_n2, v1_n3, v1_b1, v1_b2, v1_b3, v2_x1, v2_x2,
    v2_x3, v2_t1, v2_t2, v2_t3, v2_n1, v2_n2, v2_n3, v2_b1, v2_b2, v2_b3);

    if( index < ListSize)
    {
    xvalue[index]= v2_n1[index];
    yvalue[index]= v2_n2[index];
    zvalue[index]= v2_n3[index];
    }
}
else if (idx==1000 && optimum==max)
{
Integration2( pu, pv, u1_x1, u1_x2, u1_x3, u1_t1, u1_t2, u1_t3, u1_n1, u1_n2,
    u1_n3, u1_b1, u1_b2, u1_b3, u2_x1, u2_x2, u2_x3, u2_t1, u2_t2, u2_t3,
    u2_n1, u2_n2, u2_n3, u2_b1, u2_b2, u2_b3, v1_x1, v1_x2, v1_x3, v1_t1,
    v1_t2, v1_t3, v1_n1, v1_n2, v1_n3, v1_b1, v1_b2, v1_b3, v2_x1, v2_x2,
    v2_x3, v2_t1, v2_t2, v2_t3, v2_n1, v2_n2, v2_n3, v2_b1, v2_b2, v2_b3);

    if( index < ListSize)
    {
    xvalue[index]= u2_n1[index];
    yvalue[index]= u2_n2[index];
    zvalue[index]= u2_n3[index];
    }
}
else
{
Integration( pu, pv, u1_x1, u1_x2, u1_x3, u1_t1, u1_t2, u1_t3, u1_dt1, u1_dt2,
    u1_dt3, u1_d2t1, u1_d2t2, u1_d2t3, u1_d3t1, u1_d3t2, u1_d3t3,
    u2_x1, u2_x2, u2_x3, u2_t1, u2_t2, u2_t3, u2_dt1, u2_dt2, u2_dt3,
    u2_d2t1, u2_d2t2, u2_d2t3, u2_d3t1, u2_d3t2, u2_d3t3, v1_x1, v1_x2,
    v1_x3, v1_t1, v1_t2, v1_t3, v1_dt1, v1_dt2, v1_dt3, v1_d2t1,
    v1_d2t2, v1_d2t3, v1_d3t1, v1_d3t2, v1_d3t3, v2_x1, v2_x2,
    v2_x3, v2_t1, v2_t2, v2_t3, v2_dt1, v2_dt2, v2_dt3, v2_d2t1,
    v2_d2t2, v2_d2t3, v2_d3t1, v2_d3t2, v2_d3t3, idx, optimum);

    if( index < ListSize)
    {

    CurvenormaloS(index,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,
    u1_dt2,u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,
    u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,
```

```
    u2_d2t3,u2_d3t1,u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,
    v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,
    v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2
    ,v2_d2t3,v2_d3t1,v2_d3t2,v2_d3t3,optimum,norv);
    xvalue[index]= norv[0];
    yvalue[index]= norv[1];
    zvalue[index]= norv[2];
    }
}

}

__global__ void fun8( float * xvalue, float * yvalue,
    float * zvalue, mint idx, mint optimum, mint ListSize)
{
mint index = threadIdx.x + blockIdx.x * blockDim.x;
float pu[n], pv[n];
float u1_x1[n], u1_x2[n], u1_x3[n], u1_t1[n], u1_t2[n],
    u1_t3[n], u1_dt1[n], u1_dt2[n], u1_dt3[n], u1_d2t1[n],
    u1_d2t2[n], u1_d2t3[n], u1_d3t1[n], u1_d3t2[n], u1_d3t3[n];
float u1_n1[n], u1_n2[n], u1_n3[n], u1_b1[n], u1_b2[n], u1_b3[n];
float u2_x1[n], u2_x2[n], u2_x3[n], u2_t1[n], u2_t2[n],
    u2_t3[n], u2_dt1[n], u2_dt2[n], u2_dt3[n], u2_d2t1[n],
    u2_d2t2[n], u2_d2t3[n], u2_d3t1[n], u2_d3t2[n], u2_d3t3[n];
float u2_n1[n], u2_n2[n], u2_n3[n], u2_b1[n], u2_b2[n], u2_b3[n];
float v1_x1[n], v1_x2[n], v1_x3[n], v1_t1[n], v1_t2[n],
    v1_t3[n], v1_dt1[n], v1_dt2[n], v1_dt3[n], v1_d2t1[n],
    v1_d2t2[n], v1_d2t3[n], v1_d3t1[n], v1_d3t2[n], v1_d3t3[n];
float v1_n1[n], v1_n2[n], v1_n3[n], v1_b1[n], v1_b2[n], v1_b3[n];
float v2_x1[n], v2_x2[n], v2_x3[n], v2_t1[n], v2_t2[n],
    v2_t3[n], v2_dt1[n], v2_dt2[n], v2_dt3[n], v2_d2t1[n],
    v2_d2t2[n], v2_d2t3[n], v2_d3t1[n], v2_d3t2[n], v2_d3t3[n];
float v2_n1[n], v2_n2[n], v2_n3[n], v2_b1[n], v2_b2[n], v2_b3[n];
float binorv[3];

if (idx==0 && optimum==min)
{
Integration2( pu, pv, u1_x1, u1_x2, u1_x3, u1_t1, u1_t2, u1_t3, u1_n1, u1_n2,
    u1_n3, u1_b1, u1_b2, u1_b3, u2_x1, u2_x2, u2_x3, u2_t1, u2_t2, u2_t3,
    u2_n1, u2_n2, u2_n3, u2_b1, u2_b2, u2_b3, v1_x1, v1_x2, v1_x3, v1_t1,
    v1_t2, v1_t3, v1_n1, v1_n2, v1_n3, v1_b1, v1_b2, v1_b3, v2_x1, v2_x2,
    v2_x3, v2_t1, v2_t2, v2_t3, v2_n1, v2_n2, v2_n3, v2_b1, v2_b2, v2_b3);

    if( index < ListSize)
    {
    xvalue[index]= v1_b1[index];
    yvalue[index]= v1_b2[index];
```

```
    zvalue[index]= v1_b3[index];
    }
}
else if (idx==0 && optimum==max)
{
Integration2( pu, pv, u1_x1, u1_x2, u1_x3, u1_t1, u1_t2, u1_t3, u1_n1, u1_n2,
    u1_n3, u1_b1, u1_b2, u1_b3, u2_x1, u2_x2, u2_x3, u2_t1, u2_t2, u2_t3,
    u2_n1, u2_n2, u2_n3, u2_b1, u2_b2, u2_b3, v1_x1, v1_x2, v1_x3, v1_t1,
    v1_t2, v1_t3, v1_n1, v1_n2, v1_n3, v1_b1, v1_b2, v1_b3, v2_x1, v2_x2,
    v2_x3, v2_t1, v2_t2, v2_t3, v2_n1, v2_n2, v2_n3, v2_b1, v2_b2, v2_b3);

    if( index < ListSize)
    {
    xvalue[index]= u1_b1[index];
    yvalue[index]= u1_b2[index];
    zvalue[index]= u1_b3[index];
    }
}
else if (idx==1000 && optimum==min)
{
Integration2( pu, pv, u1_x1, u1_x2, u1_x3, u1_t1, u1_t2, u1_t3, u1_n1, u1_n2,
    u1_n3, u1_b1, u1_b2, u1_b3, u2_x1, u2_x2, u2_x3, u2_t1, u2_t2, u2_t3,
    u2_n1, u2_n2, u2_n3, u2_b1, u2_b2, u2_b3, v1_x1, v1_x2, v1_x3, v1_t1,
    v1_t2, v1_t3, v1_n1, v1_n2, v1_n3, v1_b1, v1_b2, v1_b3, v2_x1, v2_x2,
    v2_x3, v2_t1, v2_t2, v2_t3, v2_n1, v2_n2, v2_n3, v2_b1, v2_b2, v2_b3);

    if( index < ListSize)
    {
    xvalue[index]= v2_b1[index];
    yvalue[index]= v2_b2[index];
    zvalue[index]= v2_b3[index];
    }
}
else if (idx==1000 && optimum==max)
{
Integration2( pu, pv, u1_x1, u1_x2, u1_x3, u1_t1, u1_t2, u1_t3, u1_n1, u1_n2,
    u1_n3, u1_b1, u1_b2, u1_b3, u2_x1, u2_x2, u2_x3, u2_t1, u2_t2, u2_t3,
    u2_n1, u2_n2, u2_n3, u2_b1, u2_b2, u2_b3, v1_x1, v1_x2, v1_x3, v1_t1,
    v1_t2, v1_t3, v1_n1, v1_n2, v1_n3, v1_b1, v1_b2, v1_b3, v2_x1, v2_x2,
    v2_x3, v2_t1, v2_t2, v2_t3, v2_n1, v2_n2, v2_n3, v2_b1, v2_b2, v2_b3);

    if( index < ListSize)
    {
    xvalue[index]= u2_b1[index];
    yvalue[index]= u2_b2[index];
    zvalue[index]= u2_b3[index];
    }
```

```
}
else
{
Integration( pu, pv, u1_x1, u1_x2, u1_x3, u1_t1, u1_t2, u1_t3, u1_dt1, u1_dt2,
    u1_dt3, u1_d2t1, u1_d2t2, u1_d2t3, u1_d3t1, u1_d3t2, u1_d3t3,
    u2_x1, u2_x2, u2_x3, u2_t1, u2_t2, u2_t3, u2_dt1, u2_dt2, u2_dt3,
    u2_d2t1, u2_d2t2, u2_d2t3, u2_d3t1, u2_d3t2, u2_d3t3, v1_x1, v1_x2,
    v1_x3, v1_t1, v1_t2, v1_t3, v1_dt1, v1_dt2, v1_dt3, v1_d2t1,
    v1_d2t2, v1_d2t3, v1_d3t1, v1_d3t2, v1_d3t3, v2_x1, v2_x2,
    v2_x3, v2_t1, v2_t2, v2_t3, v2_dt1, v2_dt2, v2_dt3, v2_d2t1,
    v2_d2t2, v2_d2t3, v2_d3t1, v2_d3t2, v2_d3t3, idx, optimum);

    if( index < ListSize)
    {

    CurvebinormaloS(index,pu,pv,u1_x1,u1_x2,u1_x3,u1_t1,u1_t2,u1_t3,u1_dt1,
    u1_dt2,u1_dt3,u1_d2t1,u1_d2t2,u1_d2t3,u1_d3t1,u1_d3t2,u1_d3t3,u2_x1,
    u2_x2,u2_x3,u2_t1,u2_t2,u2_t3,u2_dt1,u2_dt2,u2_dt3,u2_d2t1,u2_d2t2,
    u2_d2t3,u2_d3t1,u2_d3t2,u2_d3t3,v1_x1,v1_x2,v1_x3,v1_t1,v1_t2,v1_t3,
    v1_dt1,v1_dt2,v1_dt3,v1_d2t1,v1_d2t2,v1_d2t3,v1_d3t1,v1_d3t2,v1_d3t3,
    v2_x1,v2_x2,v2_x3,v2_t1,v2_t2,v2_t3,v2_dt1,v2_dt2,v2_dt3,v2_d2t1,v2_d2t2
    ,v2_d2t3,v2_d3t1,v2_d3t2,v2_d3t3,optimum,binorv);
    xvalue[index]= binorv[0];
    yvalue[index]= binorv[1];
    zvalue[index]= binorv[2];
    }
}

}


";
```

```
Needs["CUDALink`"]
```

```
AbsoluteTiming[funcCS = CUDAFunctionLoad[kernelcodeCS,
   "fun", {{"Float", _, "Output"}, {"Float", _, "Output"},
    {"Float", _, "Output"}, _Integer, _Integer, _Integer}, 256]]
```

```
{385.268, CUDAFunction[<>, fun, {{Float, _, Output},
   {Float, _, Output}, {Float, _, Output}, Integer64, Integer64, Integer64}]}
```

```
AbsoluteTiming[funcCS2 = CUDAFunctionLoad[kernelcodeCS, "fun2",
    {{"Float", _, "Output"}, _Integer, _Integer, _Integer}, 256]];
AbsoluteTiming[funcCS4 = CUDAFunctionLoad[kernelcodeCS, "fun4",
    {{"Float", _, "Output"}, {"Float", _, "Output"},
     {"Float", _, "Output"}, {"Float", _, "Output"},
     {"Float", _, "Output"}, _Integer, _Integer, _Integer}, 256]];
AbsoluteTiming[funcCS6 = CUDAFunctionLoad[kernelcodeCS, "fun6",
    {{"Float", _, "Output"}, {"Float", _, "Output"},
     {"Float", _, "Output"}, _Integer, _Integer, _Integer}, 256]];
AbsoluteTiming[funcCS7 = CUDAFunctionLoad[kernelcodeCS, "fun7",
    {{"Float", _, "Output"}, {"Float", _, "Output"},
     {"Float", _, "Output"}, _Integer, _Integer, _Integer}, 256]];
AbsoluteTiming[funcCS8 = CUDAFunctionLoad[kernelcodeCS, "fun8",
    {{"Float", _, "Output"}, {"Float", _, "Output"},
     {"Float", _, "Output"}, _Integer, _Integer, _Integer}, 256]];

TangentLoCCS[idx_, optimum_, nn_] := Module[{buffer1, buffer2, buffer3, resCS},
   buffer1 = CUDAMemoryAllocate["Float", nn];
   buffer2 = CUDAMemoryAllocate["Float", nn];
   buffer3 = CUDAMemoryAllocate["Float", nn];
   resCS = funcCS6[buffer1, buffer2, buffer3, idx, optimum, nn];
   Transpose[
    {CUDAMemoryGet[buffer1], CUDAMemoryGet[buffer2], CUDAMemoryGet[buffer3]}]];
NormalLoCCS[idx_, optimum_, nn_] := Module[{buffer1, buffer2, buffer3, resCS},
   buffer1 = CUDAMemoryAllocate["Float", nn];
   buffer2 = CUDAMemoryAllocate["Float", nn];
   buffer3 = CUDAMemoryAllocate["Float", nn];
   resCS = funcCS7[buffer1, buffer2, buffer3, idx, optimum, nn];
   Transpose[
    {CUDAMemoryGet[buffer1], CUDAMemoryGet[buffer2], CUDAMemoryGet[buffer3]}]];
BinormalLoCCS[idx_, optimum_, nn_] := Module[{buffer1, buffer2, buffer3, resCS},
   buffer1 = CUDAMemoryAllocate["Float", nn];
   buffer2 = CUDAMemoryAllocate["Float", nn];
   buffer3 = CUDAMemoryAllocate["Float", nn];
   resCS = funcCS8[buffer1, buffer2, buffer3, idx, optimum, nn];
   Transpose[
    {CUDAMemoryGet[buffer1], CUDAMemoryGet[buffer2], CUDAMemoryGet[buffer3]}]];

CoordCS[idx_, optimum_, nn_] := Module[{buffer1, buffer2, buffer3, resCS},
   buffer1 = CUDAMemoryAllocate["Float", nn];
   buffer2 = CUDAMemoryAllocate["Float", nn];
   buffer3 = CUDAMemoryAllocate["Float", nn];
   resCS = funcCS[buffer1, buffer2, buffer3, idx, optimum, nn];
   Transpose[
    {CUDAMemoryGet[buffer1], CUDAMemoryGet[buffer2], CUDAMemoryGet[buffer3]}]];
```

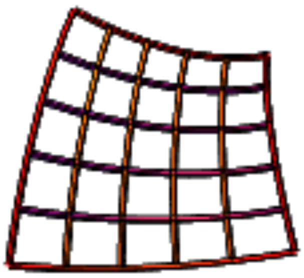
```
CurvatureCS[idx_, optimum_, nn_] := Module[{buffer1, resCS},
   buffer1 = CUDAMemoryAllocate["Float", nn];
   resCS = funcCS2[buffer1, idx, optimum, nn];
   Transpose[{Table[x, {x, 0, (nn - 1) / 1000, .001}], CUDAMemoryGet[buffer1]}]];

CoordCS4[idx_, optimum_, nn_] :=
 Module[{buffer1, buffer2, buffer3, buffer4, buffer5, resCS},
  buffer1 = CUDAMemoryAllocate["Float", nn];
  buffer2 = CUDAMemoryAllocate["Float", nn];
  buffer3 = CUDAMemoryAllocate["Float", nn];
  buffer4 = CUDAMemoryAllocate["Float", nn];
  buffer5 = CUDAMemoryAllocate["Float", nn];
  resCS = funcCS4[buffer1, buffer2, buffer3, buffer4, buffer5, idx, optimum, nn];
  Transpose[{CUDAMemoryGet[buffer1], CUDAMemoryGet[buffer2],
    CUDAMemoryGet[buffer3], CUDAMemoryGet[buffer4], CUDAMemoryGet[buffer5]}]]

CoordCS5[idx_, optimum_, nn_] := Module[{buffer1, resCS},
   buffer1 = CUDAMemoryAllocate["Float", nn];
   resCS = funcCS2[buffer1, idx, optimum, nn];
   CUDAMemoryGet[buffer1]];

plot[idx_, optimum_, nn_, radius_, color_] :=
 Module[{curve, tangent, normal, binormal,
   crosssection, crosssection0, pts, pts2, m, n, polys, LoC},
  curve = CoordCS[idx, optimum, nn];
  tangent = TangentLoCCS[idx, optimum, nn];
  normal = NormalLoCCS[idx, optimum, nn];
  binormal = BinormalLoCCS[idx, optimum, nn];
  crosssection0 = radius {{1, 1}, {-1, 1}, {-1, -1}, {1, -1}};
  crosssection =
   Evaluate[Table[crosssection0.{normal[[k]], binormal[[k]]}, {k, 1, nn, 1}]];
  pts = Flatten[Plus[Transpose[ConstantArray[curve, Length[crosssection0]]],
     crosssection], 1];
  pts2 = Transpose[Plus[Transpose[ConstantArray[curve, Length[crosssection0]]],
     crosssection]];
  m = Length[crosssection0];
  n = nn;
  polys =
   Partition[Flatten[BlockMap[Transpose[{Partition[#[[1]], 2, 1, #[[1, 1]]],
          Reverse /@ Partition[#[[2]], 2, 1, #[[2, 1]]]}] &,
      Transpose[Table[Range[i, m n, m], {i, 1, m}]], 2, 1]], 4];
  LoC = GraphicsComplex[pts, {FaceForm[color], Specularity[White, 40],
     EdgeForm[], Polygon[polys]}];
Show[Graphics3D[Line[pts2[[1]]]], Graphics3D[Line[pts2[[2]]]],
   Graphics3D[Line[pts2[[3]]]], Graphics3D[Line[pts2[[4]]]],
   Graphics3D[LoC, Lighting → "Neutral"]]
 ]
```

```
plot2[idx_, optimum_, nn_, radius_, {r11_, r12_}, color_] :=
 Module[{curve, tangent, normal, binormal, crosssection,
   crosssection0, pts, pts2, Psign, m, n, polys, LoC},
  curve = CoordCS[idx, optimum, nn];
  tangent = TangentLoCCS[idx, optimum, nn];
  normal = NormalLoCCS[idx, optimum, nn];
  binormal = BinormalLoCCS[idx, optimum, nn];
  crosssection0 = radius {{1, 1}, {-1, 1}, {-1, -1}, {1, -1}};
  crosssection =
   Evaluate[Table[crosssection0.{normal[[k]], binormal[[k]]}, {k, 1, nn, 1}]];
  pts = Flatten[Plus[Transpose[ConstantArray[curve, Length[crosssection0]]],
     crosssection], 1];
  pts2 = Transpose[Plus[Transpose[ConstantArray[curve, Length[crosssection0]]],
     crosssection]];
  Psign = Piecewise[{{Table[{-1, 1, 1}, nn], r11 == 1}}, Table[{1, -1, 1}, nn]];
  m = Length[crosssection0];
  n = nn;
  polys =
   Partition[Flatten[BlockMap[Transpose[{Partition[#[[1]], 2, 1, #[[1, 1]]],
         Reverse /@ Partition[#[[2]], 2, 1, #[[2, 1]]]}] &,
      Transpose[Table[Range[i, m n, m], {i, 1, m}]], 2, 1]], 4];
  LoC = GraphicsComplex[ReflectionTransform[{r11, r12, 0}][pts],
    {FaceForm[color], Specularity[White, 40], EdgeForm[], Polygon[polys]}];
Show[Graphics3D[Line[Psign * pts2[[1]]]], Graphics3D[Line[Psign * pts2[[2]]]],
   Graphics3D[Line[Psign * pts2[[3]]]], Graphics3D[Line[Psign * pts2[[4]]]],
   Graphics3D[LoC, Lighting → "Neutral"]]
 ]
```

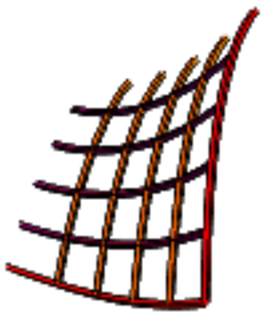
```
plot3[idx_, optimum_, nn_, radius_, color_] :=
 Module[{curve, tangent, normal, binormal, crosssection,
   crosssection0, pts, pts2, Psign, m, n, polys, LoC},
  curve = CoordCS[idx, optimum, nn];
  tangent = TangentLoCCS[idx, optimum, nn];
  normal = NormalLoCCS[idx, optimum, nn];
  binormal = BinormalLoCCS[idx, optimum, nn];
  crosssection0 = radius {{1, 1}, {-1, 1}, {-1, -1}, {1, -1}};
  crosssection =
   Evaluate[Table[crosssection0.{normal[[k]], binormal[[k]]}, {k, 1, nn, 1}]];
  pts = Flatten[Plus[Transpose[ConstantArray[curve, Length[crosssection0]]],
     crosssection], 1];
  pts2 = Transpose[Plus[Transpose[ConstantArray[curve, Length[crosssection0]]],
     crosssection]];
  Psign = Table[{-1, -1, 1}, nn];
  m = Length[crosssection0];
  n = nn;
  polys =
   Partition[Flatten[BlockMap[Transpose[{Partition[#[[1]], 2, 1, #[[1, 1]]],
         Reverse /@ Partition[#[[2]], 2, 1, #[[2, 1]]]}] &,
      Transpose[Table[Range[i, m n, m], {i, 1, m}]], 2, 1]], 4];
  LoC = GraphicsComplex[ReflectionTransform[{1, 0, 0}][
     ReflectionTransform[{0, 1, 0}][pts]],
    {FaceForm[color], Specularity[White, 40], EdgeForm[], Polygon[polys]}];
Show[Graphics3D[Line[Psign * pts2[[1]]]], Graphics3D[Line[Psign * pts2[[2]]]],
   Graphics3D[Line[Psign * pts2[[3]]]], Graphics3D[Line[Psign * pts2[[4]]]],
   Graphics3D[LoC, Lighting → "Neutral"]]
 ]
```

(c)

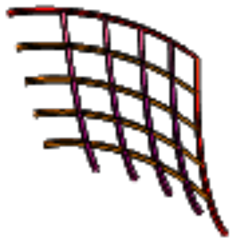
```
Show[plot[0, 0, 1001, .01, Red], plot[1000, 0, 1001, .01, Red],
 plot[200, 0, 923, .01, Orange], plot[400, 0, 975, .01, Orange],
 plot[600, 0, 1041, .01, Orange], plot[800, 0, 1112, .01, Orange],
 plot[0, 1, 1001, .01, Red], plot[1000, 1, 1001, .01, Red],
 plot[200, 1, 923, .01, Purple], plot[400, 1, 975, .01, Purple],
 plot[600, 1, 1041, .01, Purple], plot[800, 1, 1112, .01, Purple], Boxed → False]
```



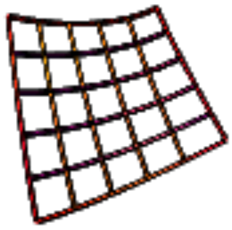
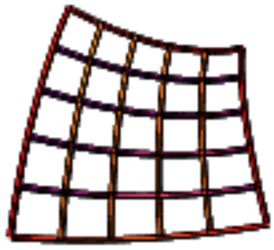
```
Show[plot2[1000, 0, 1001, .01, {1, 0}, Red],
 plot2[200, 0, 923, .01, {1, 0}, Orange], plot2[400, 0, 975, .01, {1, 0}, Orange],
 plot2[600, 0, 1041, .01, {1, 0}, Orange], plot2[800, 0, 1112, .01, {1, 0}, Orange],
 plot2[1000, 1, 1001, .01, {1, 0}, Red], plot2[200, 1, 923, .01, {1, 0}, Purple],
 plot2[400, 1, 975, .01, {1, 0}, Purple], plot2[600, 1, 1041, .01, {1, 0}, Purple],
 plot2[800, 1, 1112, .01, {1, 0}, Purple], Boxed → False]
```

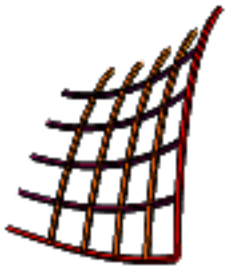
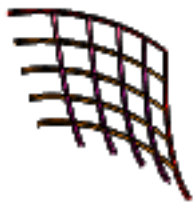


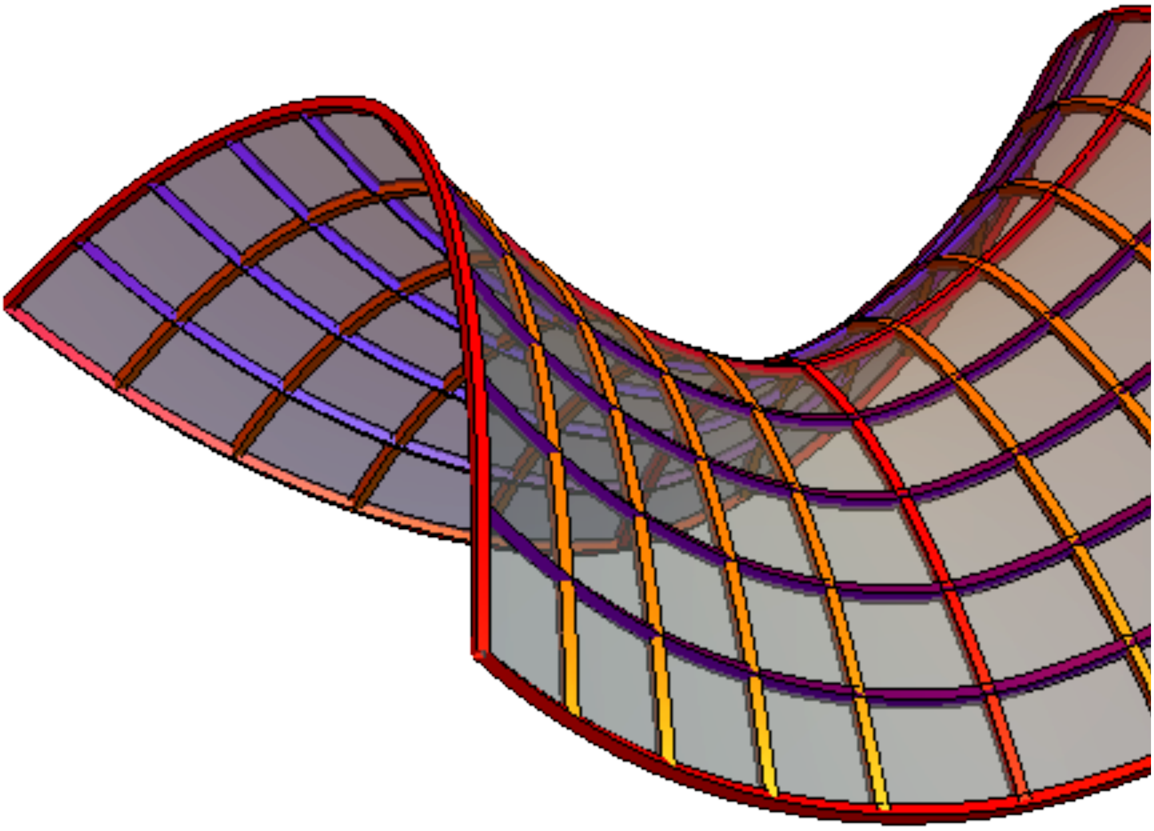
```
Show[plot2[1000, 0, 1001, .01, {0, 1}, Red],
 plot2[200, 0, 923, .01, {0, 1}, Orange], plot2[400, 0, 975, .01, {0, 1}, Orange],
 plot2[600, 0, 1041, .01, {0, 1}, Orange], plot2[800, 0, 1112, .01, {0, 1}, Orange],
 plot2[1000, 1, 1001, .01, {0, 1}, Red], plot2[200, 1, 923, .01, {0, 1}, Purple],
 plot2[400, 1, 975, .01, {0, 1}, Purple], plot2[600, 1, 1041, .01, {0, 1}, Purple],
 plot2[800, 1, 1112, .01, {0, 1}, Purple], Boxed → False]
```



```
ParametricPlot3D[
 {Coonspatch[u, v], ReflectionTransform[{1, 0, 0}][Coonspatch[u, v]],
  ReflectionTransform[{0, 1, 0}][Coonspatch[u, v]],
  ReflectionTransform[{0, 1, 0}][
   ReflectionTransform[{1, 0, 0}][Coonspatch[u, v]]]}, {u, 0, 1}, {v, 0, 1},
 PlotStyle → {{Opacity[0.6], Gray}, {Opacity[0.6], Gray}, {Opacity[0.6], Gray},
   {Opacity[0.6], Gray}}, Axes → False, Boxed → False, Mesh → None]
```

Show[ ,  ,

 ,  , ]



(e)

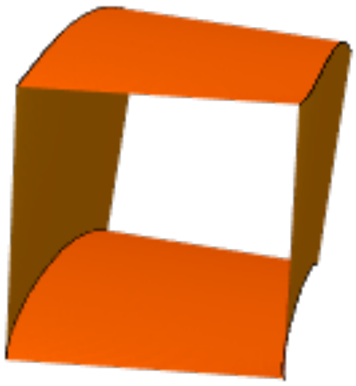
```
Show[plot3[600, 0, 1041, .01, Orange], Boxed → False](*J3*)
```

```
Show[plot3[800, 0, 1112, .01, Orange], Boxed → False]
```

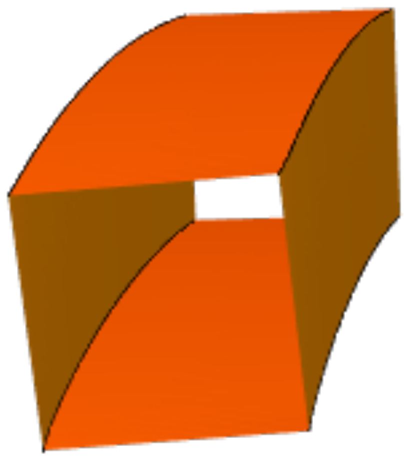
the rest

4 LoCs

```
Show[plot3[200, 0, 100, .01, Orange], Boxed → False]
```



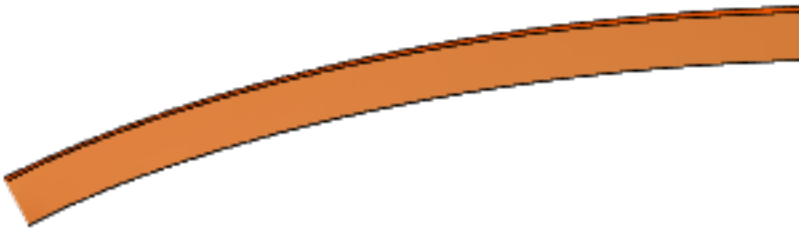
```
Show[plot3[400, 0, 100, .01, Orange], Boxed → False]
```



```
Show[plot3[600, 0, 100, .01, Orange], Boxed → False]
```

```
Show[plot3[800, 0, 100, .01, Orange], Boxed → False]
```

```
Show[plot[600, 0, 1040, .01, Orange], Boxed → False](*J3*)
```



```
Show[plot[800, 0, 1112, .01, Orange], Boxed → False](*J4*)
```