ПРИЛОЖЕНИЕ. Экспорт результатов вычисления в инженерный графопостроитель tecplot 360 для последующего анализа.

Передаваемая в tecplot 360 учебная область из четырёх треугольников представлена на нижеследующем рисунке 1.

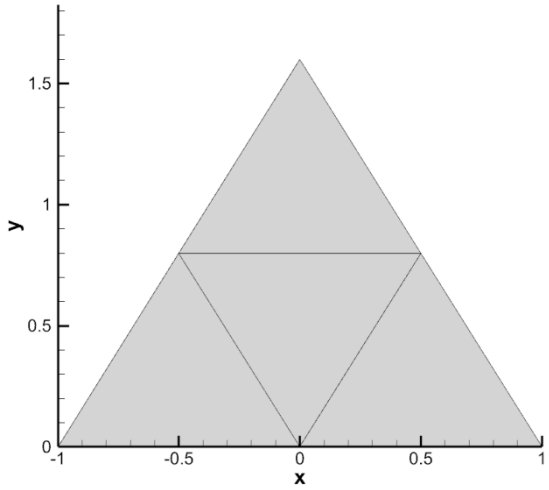


Рисунок 1 – Переданная в tecplot 360 сетка из четырёх треугольников.

1. Описание данных для передачи графопостроителю tecplot 360.

```
1.  int main()
2.  {
3.
4.      float* x = new float[6];
5.      float* y = new float[6];
6.
7.      int nodi = 6;
8.      int maglie = 4;
9.      x[0] = -1.0f; y[0] = 0.0f;
10.     x[1] = 0.0f; y[1] = 0.0f;
11.     x[2] = 1.0f; y[2] = 0.0f;
12.     x[3] = -0.5f; y[3] = 0.8f;
13.     x[4] = 0.5f; y[4] = 0.8f;
14.     x[5] = 0.0f; y[5] = 1.6f;
15.     int numvar = 2;
16.
17.
18.     int** n123 = new int* [3];
19.     for (int i = 0; i < 3; ++i) {
20.         n123[i] = new int[maglie];
21.     }
22.     n123[0][0] = 1;  n123[1][0] = 2;  n123[2][0] = 4;
23.     n123[0][1] = 2;  n123[1][1] = 3;  n123[2][1] = 5;
24.     n123[0][2] = 2;  n123[1][2] = 5;  n123[2][2] = 4;
25.     n123[0][3] = 4;  n123[1][3] = 5;  n123[2][3] = 6;
26.
27.     write_tecplot360_PLT0(nodi,  x,  y,  maglie,  numvar, n123);
28.     // либо человеческий формат (текстовый).
29.     write_tecplot360_DAT(nodi, x, y, maglie, numvar, n123);
30.
31.     delete[] x;
32.     delete[] y;
33.
34.     for (int i = 0; i < 3; ++i) {
35.         delete[] n123[i];
36.     }
```

```
37.
38.
39.     delete[] n123;
40. }
```

1. Человеческий (текстовый) формат записи данных для графопостроителя (.dat)

```
// создание файла для записи и передачи в программу tecplot 360 в текстовом виде.
void write_tecplot360_DAT(int nodi, float*& x, float*& y, int maglie, int numvar, int**&
n123) {
    FILE* fp;
    errno_t err;


    if ((err = fopen_s(&fp, "fedorenko31.dat", "w")) != 0) {
        printf("Create File Error\n");
    }
    else {

        // запись имён переменных
        fprintf(fp, "TITLE = \" Easy mesh\" \n");
        fprintf(fp, "VARIABLES = \"x, m\", \"y, m\" \n");
        fprintf(fp, "ZONE T=\"Rampant\", N=%d, E=%d, ET=TRIANGLE, F=FEBLOCK \n\n", nodi,
maglie);

        for (int i = 0; i < nodi; ++i) {

            fprintf(fp, "%e ", x[i]);
            if ((i != 0) && (i % 10 == 0)) fprintf(fp, "\n");
        }
        fprintf(fp, "\n");

        for (int i = 0; i < nodi; ++i) {

            fprintf(fp, "%e ", y[i]);
            if ((i != 0) && (i % 10 == 0)) fprintf(fp, "\n");
        }
        fprintf(fp, "\n");

        for (int i = 0; i < maglie; ++i) {
            fprintf(fp, "%d %d %d \n", n123[0][i], n123[1][i], n123[2][i]);
        }

        fclose(fp);
    }
   }
```

Текстовый файл .dat для tecplot 360 имеет следующий вид:

```
TITLE = " Easy mesh"
VARIABLES = "x, m", "y, m"
ZONE T="Rampant", N=6, E=4, ET=TRIANGLE, F=FEBLOCK

-1.000000e+00 0.000000e+00 1.000000e+00 -5.000000e-01 5.000000e-01
0.000000e+00
0.000000e+00 0.000000e+00 0.000000e+00 8.000000e-01 8.000000e-01
1.600000e+00
1 2 4
2 3 5
2 5 4
4 5 6
```

Это учебный пример, но если текстовый файл для tecplot большого размера более 10млн неизвестных, то открытие такого файла занимает целый рабочий день. Поэтому используется машинный двоичный формат файла.

2. **Машинный двоичный формат файла**.

```
void write_tecplot360_PLT0(int nodi, float* &x, float* &y, int maglie, int numvar, int** &n123)
{
    int  buttaINT,
        FileType, ZoneType, StrandID, VarLoc, * nodeCELLcenter,
       jmax;
    //
    float  dt, BUTTA;
    double t;//, DBLEmin, DBLEmax;
    float  ** CSprint;

    char header[10];
    char aster;
    const char NULCHAR = '\0';

    CSprint = new float* [std::max(maglie, nodi) + 1];
    for (int i = 0; i < std::max(maglie, nodi) + 1; ++i) {
       CSprint[i] = new float[numvar+1];
    }
    nodeCELLcenter = new int[numvar+1];

    //allocate(CSprint(max(maglie, nodi), numvar), nodeCELLcenter(numvar))
      //
      //write BINARY FILE FOR TECPLOT
      //
    FILE* fp;

#ifdef MINGW_COMPILLER
```

```
        int err = 0;
        fp = fopen64("AliceFlow_v0_66.PLT", "wb");
    #else
        errno_t err;
        err = fopen_s(&fp, "AliceFlow_v0_66.PLT", "wb");
    #endif

        if ((err) != 0) {
            printf("Create binary File AliceFlow_v0_66.PLT Error in function export_thermal_conductivity in tecplot_binary_writer.cpp\n");
            system("pause");

        }

        //int din = 0;
        //All character strings are null terminated(i.e.terminated by a zero value)
        //header = '#!TDV112'
        //header
        header[0] = '#';
        header[1] = '!';
        header[2] = 'T';
        header[3] = 'D';
        header[4] = 'V';
        header[5] = '1';
        header[6] = '1';
        header[7] = '1';
        header[8] = '\0';
        header[9] = '\n';

        for (int i = 0; i < 8; ++i) {
```

```
        aster = header[i];
        fwrite(&aster, sizeof(char), 1, fp);
      }
      buttaINT = 1;
      fwrite(&buttaINT, sizeof(int), 1, fp);

      //iii.Title and variable names.
      FileType = 0; //Title and variable names. 0 = FULL, 1 = GRID, 2 = SOLUTION
     // write(36, rec = iwrite) FileType; iwrite = iwrite + 1
      fwrite(&FileType, sizeof(int), 1, fp);
      aster = 'T';
      buttaINT = (int)aster;
      fwrite(&buttaINT, sizeof(int), 1, fp);
      buttaINT = (int)NULCHAR;
      fwrite(&buttaINT, sizeof(int), 1, fp);

      fwrite(&numvar, sizeof(int), 1, fp);

      //DO I = 1, NUMVAR
      //Variable names(INT32 * N).N = L[1] + L[2] + ....L[NumVar]where:L[i] = length of the ith variable name + 1(for the  terminating 0 value).
      aster = 'x';
      buttaINT = (int)aster;
      fwrite(&buttaINT, sizeof(int), 1, fp);
      buttaINT = (int)NULCHAR;
      fwrite(&buttaINT, sizeof(int), 1, fp);
      aster = 'y';
      buttaINT = (int)aster;
      fwrite(&buttaINT, sizeof(int), 1, fp);
      buttaINT = (int)NULCHAR;
```

```
fwrite(&buttaINT, sizeof(int), 1, fp);


BUTTA = 299.0; //Zone marker.
fwrite(&BUTTA, sizeof(float), 1, fp);
//Zone name(INT32 * N).N = (length of zone name) + 1(for the terminating 0 value).
aster = 'Z';
buttaINT = (int)aster;
fwrite(&buttaINT, sizeof(int), 1, fp);
aster = 'O';
buttaINT = (int)aster;
fwrite(&buttaINT, sizeof(int), 1, fp);
aster = 'N';
buttaINT = (int)aster;
fwrite(&buttaINT, sizeof(int), 1, fp);
aster = 'E';
buttaINT = (int)aster;
fwrite(&buttaINT, sizeof(int), 1, fp);
aster = '1';
buttaINT = (int)aster;
fwrite(&buttaINT, sizeof(int), 1, fp);
buttaINT = (int)NULCHAR;
fwrite(&buttaINT, sizeof(int), 1, fp);


buttaINT = -1;
fwrite(&buttaINT, sizeof(int), 1, fp);  //0 //ParentZone: 0 = indicates that  this zone is not associated with a parent zone. > 0 = A value greater  than zero is considered this zone's parent.


StrandID = 0;
fwrite(&StrandID, sizeof(int), 1, fp); //IN REALTA IL MANUALE DICE  StrandID = 0 = > static zone  StrandID : -2 = pending strand ID for  assignmentby Tecplot; -1 = static strand ID; 0 <= N < 32700 valid  strand ID
```

```
t = 0.0;
dt = 0.0;
t = t + static_cast<double>(dt);
fwrite(&t, sizeof(double), 1, fp);
buttaINT = -1;
fwrite(&buttaINT, sizeof(int), 1, fp);//Not used.

ZoneType = 2;
fwrite(&ZoneType, sizeof(int), 1, fp); //ZoneType 0 = ORDERED, 1 = FELINESEG, 2 = FETRIANGLE, 3 = FEQUADRILATERAL, 4 = FETETRAHEDRON, 5 = FEBRICK, 6 = FEPOLYGON, 7 = FEPOLYHEDRON

   // DataPacking 0 = Block, 1 = Point
buttaINT = 1;
fwrite(&buttaINT, sizeof(int), 1, fp);

VarLoc = 1;
fwrite(&VarLoc, sizeof(int), 1, fp);  //Specify Var Location.  0 = Don't specify, all data is locatedat the nodes. 1 = Specify

if (VarLoc == 1) {
   nodeCELLcenter[0] = 0;
   nodeCELLcenter[1] = 0;
   for (int i = 0; i < numvar; ++i) {
      buttaINT = nodeCELLcenter[i];
      fwrite(&buttaINT, sizeof(int), 1, fp); //0 = Node, 1 = Cell Centered(See note 5.)
   }
}

buttaINT = 0;
```

```
        fwrite(&buttaINT, sizeof(int), 1, fp); //Are raw local 1 - to - 1 face neighbors supplied ? (0 = FALSE 1 = TRUE).

        buttaINT = 0;

        fwrite(&buttaINT, sizeof(int), 1, fp); //Number of miscellaneous user - defined  face neighbor connections(value >= 0).This value is in addition to  the face neighbors


        fwrite(&nodi, sizeof(int), 1, fp);

        fwrite(&maglie, sizeof(int), 1, fp);


        //ICellDim, JCellDim, KCellDim(for future use; set to zero)

        buttaINT = 0;

        fwrite(&buttaINT, sizeof(int), 1, fp);

        buttaINT = 0;

        fwrite(&buttaINT, sizeof(int), 1, fp);

        buttaINT = 0;

        fwrite(&buttaINT, sizeof(int), 1, fp);


        //1 = Auxiliary name / value pair to follow;  0 = No more Auxiliary name / value pairs.

        buttaINT = 0;

        fwrite(&buttaINT, sizeof(int), 1, fp);



        //v.Geometries

        //write(36, rec = iwrite) 399.0e0 !Geometry marker.Value = 399.0

        //iwrite = iwrite + 1

        //vi.Text

        //write(36, rec = iwrite) 499.0e0 !Text marker.Value = 499.0

        //iwrite = iwrite + 1

        //vii.CustomLabel

        //write(36, rec = iwrite) 599.0e0 !CustomLabel Marker; F = 599
```

```
//iwrite = iwrite + 1

//viii.UserRec

//write(36, rec = iwrite) 699.0e0 !UserRec Marker; F = 699

//iwrite = iwrite + 1

//ix.Dataset Auxiliary data.

//write(36, rec = iwrite) 799.0e0 !DataSetAux Marker; F = 799.0

//iwrite = iwrite + 1

//x.Variable Auxiliary data.

//write(36, rec = iwrite) 999.0e0 !VarAux Marker; F = 899.0

//iwrite = iwrite + 1

BUTTA = 357.0e0; //EOHMARKER, value = 357.0.

fwrite(&BUTTA, sizeof(float), 1, fp);


//II.DATA SECTION

BUTTA = 299.0e0;

fwrite(&BUTTA, sizeof(float), 1, fp); //Zone marker Value = 299.0


        //i.For both ordered and fe zones :

for (int i_1 = 1; i_1 <= numvar; ++i_1) {

   buttaINT = 1; //Variable data format(INT32 * N), N = Total number of vars 1 = Float, 2 = Double, 3 = LongInt, 4 = ShortInt, 5 = Byte, 6 = Bit

   fwrite(&buttaINT, sizeof(int), 1, fp);

}


buttaINT = 0;

fwrite(&buttaINT, sizeof(int), 1, fp); //Has passive variables : 0 = no, 1 = yes.

buttaINT = 0;

fwrite(&buttaINT, sizeof(int), 1, fp); //Has variable sharing 0 = no, 1 = yes.
```

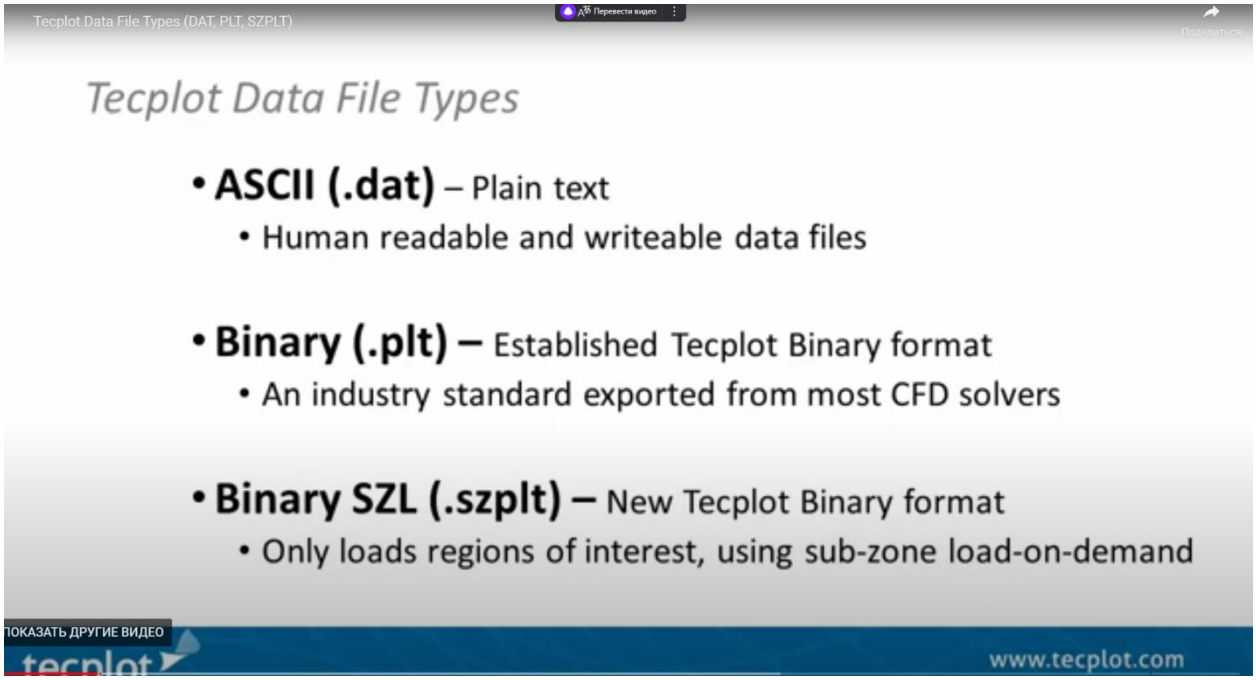
```
        buttaINT = -1; //Zero based zone number to share  connectivity list with(-1 = no sharing).FEPOLYGON and FEPOLYHEDRON  zones use this zone number to share face map data.
        fwrite(&buttaINT, sizeof(int), 1, fp);

        //Compressed list of min / max pairs for each non - shared and non - passive variables
        for (int i_1 = 1; i_1 <= nodi; ++i_1) {
            CSprint[i_1][1] = x[i_1-1];
            CSprint[i_1][2] = y[i_1-1];
        }

        for (int i_1 = 1; i_1 <= numvar; ++i_1) {
            if (nodeCELLcenter[i_1 - 1] == 0) {
                jmax = nodi;
            }
            else if (nodeCELLcenter[i_1 - 1] == 1) {
                jmax = maglie;
            }

            double DBLEmin = 1.0e36;

            for (int i_2 = 1; i_2 <= jmax; ++i_2) {
                if (CSprint[i_2][i_1] < DBLEmin) {
                    DBLEmin = CSprint[i_2][i_1];
                }
            }

            // DBLEmin = DBLE(MINVAL(CSprint(1:jmax, i)))

            double DBLEmax = -1.0e36;
```

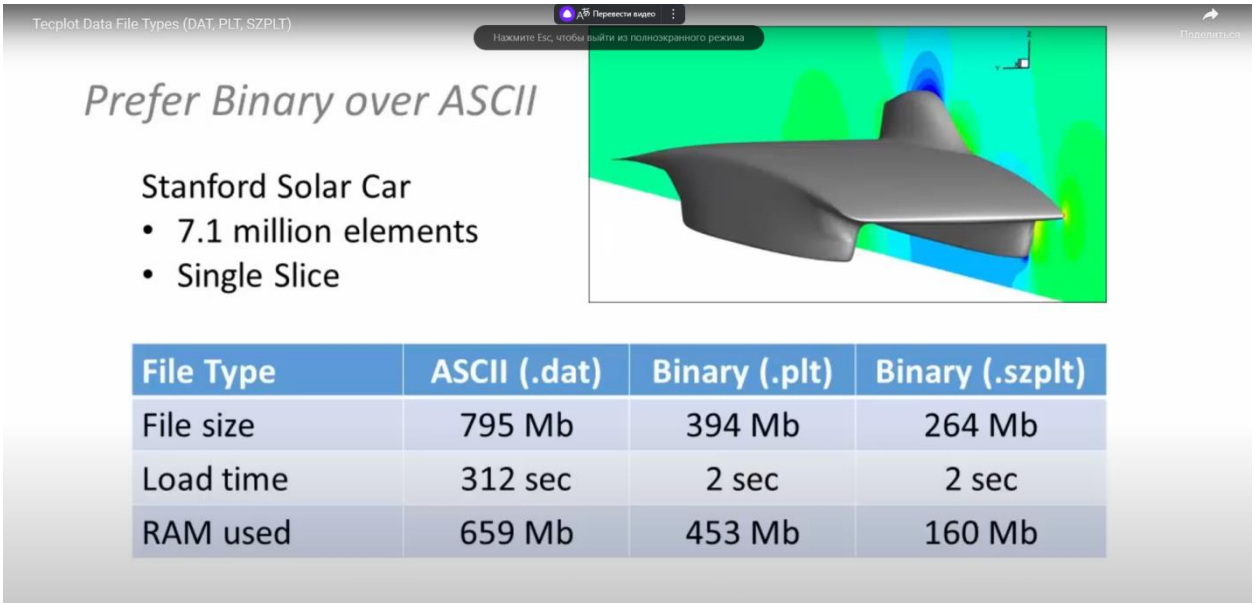
```
    //DBLEmax = DBLE(MAXVAL(CSprint(1:jmax, i)))
    for (int i_2 = 1; i_2 <= jmax; ++i_2) {
        if (CSprint[i_2][i_1] > DBLEmax) {
            DBLEmax = CSprint[i_2][i_1];
        }
    }
    fwrite(&DBLEmin, sizeof(double), 1, fp);
    fwrite(&DBLEmax, sizeof(double), 1, fp);

    std::cout << DBLEmin << " " << DBLEmax << "\n";
}
{
    jmax = nodi;

    for (int i_2 = 1; i_2 <= jmax; ++i_2) {
        BUTTA = CSprint[i_2][1];
        fwrite(&BUTTA, sizeof(float), 1, fp);
        BUTTA = CSprint[i_2][2];
        fwrite(&BUTTA, sizeof(float), 1, fp);
    }
}
//ii.specific to ordered zones
//iii.specific to fe zones
for (int i = 0; i < maglie; ++i) {
    for (int k = 0; k < 3; ++k)
    {
        buttaINT = n123[k][i]-1;
        fwrite(&buttaINT, sizeof(int), 1, fp);
    }
}
```

```
    for (int i = 0; i < std::max(maglie, nodi) + 1; ++i) {
        delete[] CSprint[i];
    }
    delete[] CSprint;
    delete[] nodeCELLcenter;
    fclose(fp);
}
```

Конечно приводимый программный код записи в бинарный файл тяжеловат для восприятия человеком. Зато модели записанные в бинарном виде для техплот (tecplot 360) и имеющие более 10млн узлов открываются теперь мгновенно.

3. Быстрое открытие больших моделей в бинарном виде подтверждается в документации самого tecplot 360:





Отличие во времени открытия файла 156 раз.

4. Преобразование из одного формата в другой средствами tecplot 360

```
std::cout << "start convert to .szplt\n";
                                    //system("tec360 ALICEFLOW0_07_temp.dat -o
ALICEFLOW0_66_temp.szplt");
                                    system("tec360 AliceFlow_v0_66.PLT -o
ALICEFLOW0_66_temp.szplt");
```

Только текстовый dat при преобразовании будет считываться также долго, как и при открытии. А преобразование бинарного PLT в более сжатый и еще более быстрый szplt заслуживает внимания (и использования).