Arjun Ray

Facility for Rare Isotope Beams, Michigan State University, East Lansing, MI 48824 USA

2025

MICHIGAN STATE UNIVERSITY

U.S. DEPARTMENT OF ENERGY | Office of Science

This material is based upon work supported by the U.S. Department of Energy Office of Science under Cooperative Agreement DE-SC0000661, the State of Michigan and Michigan State University. Michigan State University operates FRIB as a DOE Office of Science National User Facility in support of the mission of the Office of Nuclear Physics.

BOTTLENECK POINTS AND POSSIBLE OPTIMIZATIONS METHODS

SOME BOTTLENECK POINTS:

Function digam in e_Tceis

```
if(z == c1)  {  psi = -c1*gama;  return;}


if(z.real() < 0)  psi = diser(c1-z) - PI*cos(PI*z)/sin(PI*z);
else              psi = diser(z);
}
```

Function chypser in e_Tceis

```
848                 while(1)
849                     {
850     3.42 min        cfac=((caa*cbb)/ccc)*cfac;
851     1.42 min        cfac=cfac*x/double(n);
852      31.16 s        cf=ctemp+cfac;
853      48.81 s        if(cf == ctemp) break;
854
855                     ctemp=cf;
856   920.00 ms         n=n+1;
857      34.51 s        caa=caa+c1;
858      31.05 s        cbb=cbb+c1;
859      28.84 s        ccc=ccc+c1;
860                     }
```

BOTTLENECK POINTS AFTER PRESSING THE "CONTINUE" BUTTON

```
int f_numPP(int NCO)
{
int numPP=NUM3[NCO+1];
return numPP;
}
```

In e_F4

```
void CalcOlegSum(double *V)
{
static int counter=0;

      for(int i=0; i<8; i++) sumM[i]=0;

      for(int I=1;   I<=63;   I++) sumM[0] += V[I];
      for(int I=64;  I<=66;   I++) sumM[1] += V[I];
      for(int I=67;  I<=73;   I++) sumM[2] += V[I];
      for(int I=74;  I<=84;   I++) sumM[3] += V[I];
      for(int I=85;  I<=293;  I++) sumM[4] += V[I];
      for(int I=294; I<=326;  I++) sumM[5] += V[I];
      for(int I=327; I<=1283; I++) sumM[6] += V[I];

      for(int i=0; i<7; i++)
                        if(i!=4)
                              sumM[7] += (sumM[i]-1.);

counter++;

}
```

In e_ETACHA

POSSIBLE SOLUTIONS:

-First step will be to optimize numerical calculations by reducing redundant calculations(reducing iterations and more efficient methods of calculation, using temporary variables)

-Parallelize the computation of loops in heavy functions.

-Approximations for Large Iterations