

## Монте-Карло

```
#include<mpi.h>
#include<math.h>
#include<stdlib.h>
#include<iostream>
double f(double x, double y)
{
    return x/(y*y);
}
using namespace std;
int main(int argc, char**argv)
{
    setlocale(LC_ALL, "rus");
    double starttime=MPI_Wtime();
    int rank, size, r, i, j;
    double ax=0, bx=1, ay=2, by=5;

    MPI_Init(&argc,&argv);
    MPI_Comm_rank(MPI_COMM_WORLD,&rank);
    MPI_Comm_size(MPI_COMM_WORLD,&size);

    srand(MPI_Wtime()+rank*size*1000);
    double integral;
    double randX, randY;
    double temp;
    double znakx, znaky;
    double n=10000000;
    double x=bx-ax, y=by-ay;
    temp = 0;
    for(i=0; i<n/size; i++)
    {
        double zx = (rand()%1);
        if (zx=0)
            znakx = -1;
        else
            znakx = 1;

        double zy = (rand()%1);
        if (zy=0)
            znaky = -1;
        else
            znaky = 1;

        randX=ax+(double)(rand())/RAND_MAX*x;
        randY=ay+(double)(rand())/RAND_MAX*y;
        //randX=ax+((rand()%100)/(10000*1.0))*x;
        //randY=ay+((rand()%100)/(10000*1.0))*y;

        if((randX>ax)&&(randX<bx)&&(randY>ay)&&(randY<by))
        //if ((fabs(randX)<randY)&&(randY<1))
        {
            //cout << "!!!" << endl;
            temp+=f(randX,randY);
        }
    }
    MPI_Reduce(&temp,&integral,1,MPI_DOUBLE,MPI_SUM,0,MPI_COMM_WORLD);

    if (rank==0)
        integral=integral*x*y/n;

    if (rank==0)
    {
        cout << "Time = " << MPI_Wtime()-starttime << endl;
        cout << "Integral = " << integral << endl;
    }
    MPI_Finalize();
    return 0;
}
```

Time = 0.419025

Integral = 0.150015

16

Time = 0.411726

Integral = 0.150002

Сдваивание

```
#include<stdio.h>
#include<mpi.h>
#include<math.h>
#define q 4
main(int argc, char** argv)
{int rank, i, size,np;
double time_start, time_finish;
MPI_Status status;
MPI_Init(&argc,&argv);
MPI_Comm comm;
comm=MPI_COMM_WORLD;
MPI_Comm_size(comm,&size);
MPI_Comm_rank(comm,&rank);
int n=pow(2,q);
double S=rank;
double S1;
MPI_Barrier(comm);
time_start=MPI_Wtime();
int k=2;
int kk=1;
for(i=0;i<q;i++) {
if ((rank+1)%k==kk)
{MPI_Send(&S,1,MPI_DOUBLE,rank+kk,0,comm);
}S1=0;
if ((rank+1)%k==0) {
MPI_Recv(&S1,1,MPI_DOUBLE,rank-kk,MPI_ANY_TAG,comm,&status);
S+=S1;}
//{printf("Summa=%f rank=%i i=%i\n",S,rank,i);}
kk=k;
k*=2; }
time_finish=MPI_Wtime()-time_start;
MPI_Barrier(comm);
if(rank==n-1)
{ printf("Summa=%f\n",S);
printf("np:%i time=%f\n",rank,time_finish );}
MPI_Finalize();
return 0;}
```

q=2

Summa=6.000000

np:3 time=0.000027

q=3

Summa=28.000000

np:7 time=0.000028

q=4

Summa=120.000000

np:15 time=0.000047

Трапеции

```
#include<math.h>
#include<stdio.h>
#include<mpi.h>

double f(double x)
{ return exp(-x*x+0.38)/(2+sin(1/(1.5+x*x)));
}

main(int argc, char **argv)
{
    int const n=1000,esp=1e-5;
    int rank, size, tag=3,rc,i;
    double sum,sum1,sum2,eps1,h0,h,t1,t2,a,b,x;
    double const a0=0.4, b0=1;
    MPI_Status status;
    MPI_Comm comm;
    comm=MPI_COMM_WORLD;
    rc=MPI_Init(&argc,&argv);
    rc=MPI_Comm_size(comm,&size);
    rc=MPI_Comm_rank(comm,&rank);
    t1=MPI_wtime();
    h0=(b0-a0)/size;
    a=a0+rank*h0;
    b=a+h0;
    h=(b-a)/(n/size);
    sum=0;
    for(i=1;i<n/size;i++)
    {x=a+i*h;
    sum=sum+f(x);}
    sum1=h*sum+h*(f(a)+f(b))/2;
    rc=MPI_Reduce(&sum1, &sum2, 1, MPI_DOUBLE, MPI_SUM, 0, comm);
    if (rank==0)
    { //sum1=h*sum2+h*(f(a0)-f(b0))/2;
    t2=MPI_wtime();
    printf("rank = %d, size = %d, Integral = %f \n", rank,size,sum2);
    printf("Time = %f \n", t2-t1);}
    rc=MPI_Finalize();
}
2
```

rank = 0, size = 2, Integral = 0.215480  
Time = 0.000073

4  
rank = 0, size = 4, Integral = 0.215480  
Time = 0.000083

8  
rank = 0, size = 8, Integral = 0.215480  
Time = 0.000055

16  
rank = 0, size = 16, Integral = 0.215480  
Time = 0.000764