

```

1 #define N 1024
2 double field[N*N];
3 double tmpfield[N*N];
4 #define pos(i,j) (j)*N+(i)
5
6 void relaxation(double *oldValue, double *newValue) {
7     #pragma acc loop
8     for (int x=1; x<N-1; x++) {
9         #pragma acc loop
10        for (int y=1; y<N-1; y++) {
11            double sum = oldValue[pos(x,y)];
12            sum += oldValue[pos(x-1,y)];
13            sum += oldValue[pos(x+1,y)];
14            sum += oldValue[pos(x,y+1)];
15            sum += oldValue[pos(x,y-1)];
16            sum += oldValue[pos(x+1,y+1)];
17            sum += oldValue[pos(x-1,y+1)];
18            sum += oldValue[pos(x+1,y-1)];
19            sum += oldValue[pos(x-1,y-1)];
20            sum = sum / 9;
21            newValue[pos(x,y)] = sum;
22        }
23    }
24 }
25
26 void transform() {
27     ...
28     #pragma acc data copy(field) create(tmpfield)
29     {
30         for (int i=0; i<128; i++) {
31             #pragma acc parallel
32             relaxation(field, tmpfield);
33             #pragma acc parallel
34             relaxation(tmpfield, field);
35             if (check) {
36                 #pragma acc update host(field)
37                 display(field);
38             }
39         }
40     }
41     ...
42 }

```