В. В. ДАВЫДОВСКАЯ[1], Е. В. ДАНЧЕНКО[2]
[1]УО МГПУ им. И. П. Шамякина (г. Мозырь, Беларусь)
[2]УО Полоцкий государственный университет (г. Полоцк, Беларусь)

ПОВЫШЕНИЕ СКОРОСТИ ВЫЧИСЛЕНИЙ ПРИ ЧИСЛЕННОМ РЕШЕНИИ ЗАДАЧ МАТЕМАТИЧЕСКОЙ ФИЗИКИ В СРЕДЕ MATLAB

Решение различных инженерных и физических задач часто содержит численные алгоритмы решения дифференциальных уравнений в частных производных, а также систем уравнений. При решении достаточно сложных задач требуются определенные вычислительные затраты, что отражается на увеличении времени выполнения программы.

В работе описаны основные принципы написания программ в MATLAB, которые позволяют снижать время счета и экономить память.

В основном выполнение программы занимает значительное время в том случае, когда листинг программы содержит циклические конструкции, так как каждая строка цикла интерпретируется столько раз, сколько выполняется цикл. Поэтому для увеличения скорости выполнения программы при ее разработке необходимо свести исполнение циклов к минимуму [1].

При работе с матрицами и векторами в MATLAB существует возможность заменить циклы поэлементными операциями.

Рассмотрим численное решение уравнения Пуассона.

$$\frac{\partial^2 u}{\partial x} + \frac{\partial^2 u}{\partial y} = G(x,y),$$

где $-1 \le x \le 1, -1 \le y \le 1$, $\quad G(x,y) = x^2 - y^2$.

Решение осуществлялось с использованием метода Якоби в математическом пакете MathCAD и среде MATLAB.

Были написаны две программы с использованием циклических структур.

В математическом пакете MathCAD время вычислений составило 9,5 секунд (Листинг 1) [2].

Листинг 1

$start := time(0)$

$U(x,y) := x^2 - y^2$

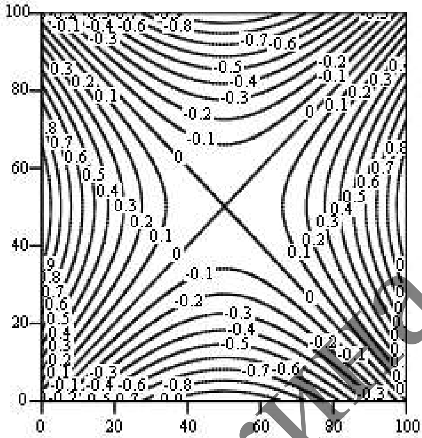
$nx := 100 \quad ny := 100 \quad L1 := 2 \quad L2 := 2$

$i := 0..nx \quad j := 0..ny \quad hx := \frac{L1}{nx} \quad hy := \frac{L2}{ny} \quad h$

$x_i := -1 + hx \cdot i \quad y_j := -1 + hy \cdot j$

$G_{i,j} := U(x_i, y_j)$

$$h(u,\varepsilon,n) := \left| \begin{array}{l} b \leftarrow u \\ \text{for } k \in 0..n \\ \quad \left| \begin{array}{l} \text{for } i \in 1..nx-1 \\ \quad \text{for } j \in 1..ny-1 \\ \quad\quad u_{i,j} \leftarrow \frac{1}{4} \cdot \left(u_{i+1,j} + u_{i-1,j} + u_{i,j-1} + u_{i,j+1}\right) \\ a \leftarrow norm1(b-u) \\ b \leftarrow u \\ iter \leftarrow k \\ \text{break if } a < \varepsilon \end{array} \right. \\ (iter \;\; u) \end{array} \right.$$



$u_{0,j} := G_{0,j} \quad u_{nx,j} := G_{nx,j} \quad u_{i,ny} := G_{i,ny} \quad u_{i,0} := G_{i,0}$

$resh := h(u, 0.001, 1000)$

$u := resh_{0,1} \quad k := resh_{0,0} \quad k = 1 \times 10^3$

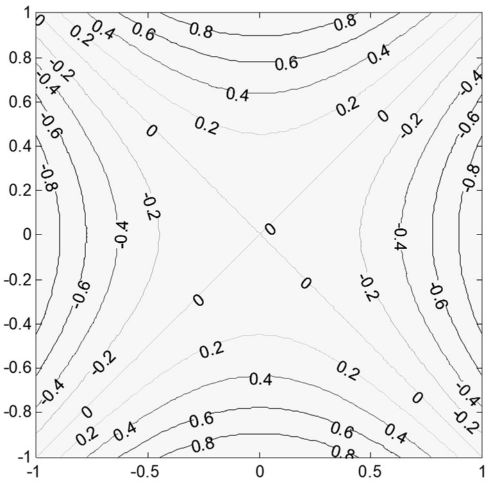
$finish := время(0)$

$calc\_time := finish - start = 9.552$

При реализации данного алгоритма в среде MATLAB скорость вычислений составила 3,5 секунды, что более чем в два раза превосходит скорость вычислений в пакете MathCAD (Листинг 2) [1, 3].

Листинг 2

```
tic;
nx=100;
ny=100;
n=1000;
L1=2;
L2=2;
eps=10^-3;
hx=L1/(nx-1);
hy=L2/(ny-1);
for i=1:nx
    x(i)=-1+hx*(i-1);
end;
for i=1:ny
    y(i)=-1+hy*(i-1);
end;
for i=1:nx
    for j=1:ny
G(i,j)=(x(i))^2-(y(j))^2;
    end;
end;
B=G;
for i=1:nx
    U(i,ny)=G(i,ny);
    U(i,1)=G(i,1);
end;
for j=1:ny
    U(1,j)=G(1,j);
    U(nx,j)=G(nx,j);
```



```
        end;
        for k=0:n
        for i=2:nx-1
            for j=2:ny-1
                U(i,j)=(1/4)*(U(i+1,j)+U(i-1,j)+U(i,j-1)+U(i,j+1));
                eps1(i,j)=B(i,j)-U(i,j);
                B(i,j)=U(i,j);
            end;
        end;
        iter=k;
                if norm(eps1)<=eps break ;
        end;
        end;
        figure;
        surf(x,y,U);
        figure;
        [x,y] = meshgrid(-1:hx:1);
        [C,h] = contour(x,y,U);
        clabel(C,h);
        axis square;
        clc;
        t1=toc;
```

Далее устраним из программы все циклы, используя поэлементные операции с векторами и матрицами, а также в качестве аналога уравнения Пуассона составим конечно-разностную матрицу, которая может быть определена как сумма тензорных произведений двух одномерных матриц D × I + I×D, где I – единичная, а D – соответствует аппроксимации второй производной (пример взят из справочной системы MATLAB).

Время вычисления программы, после проведенной оптимизации составило 0,3 секунды. Таким образом, можно говорить, что скорость вычисления программы увеличилась более чем в 10 раз (Листинг 3).

Листинг 3

```
tic;
nx=100;
ny=100;
n=nx;
L1=2;
L2=2;
eps=10^-3;
hx=L1/(nx-1);
hy=L2/(ny-1);
x=-L1/2:hx:L1/2;
y=-L2/2:hx:L2/2;
[X,Y]=meshgrid(x,y);
G=X.^2-Y.^2;
B=G;
U=zeros(nx,ny);
U(:,ny)=G(:,ny);
U(:,1)=G(:,1);
U(1,:)=G(1,:);
U(nx,:)=G(nx,:);
I = speye(n,n);
E = sparse(2:n,1:n-1,1,n,n);
D = E+E'-2*I;
A = kron(D,I)+kron(I,D);
U1=reshape(G',1,n*n)';
U2=A*U1;
U2=A\U2;
U3=reshape(U2,n,n);
figure;
surf(X,Y,U3);
figure;
[C,h] = contour(X,Y,U3);
clabel(C,h);
axis square;
clc;
t2=toc;
t1/t2
```

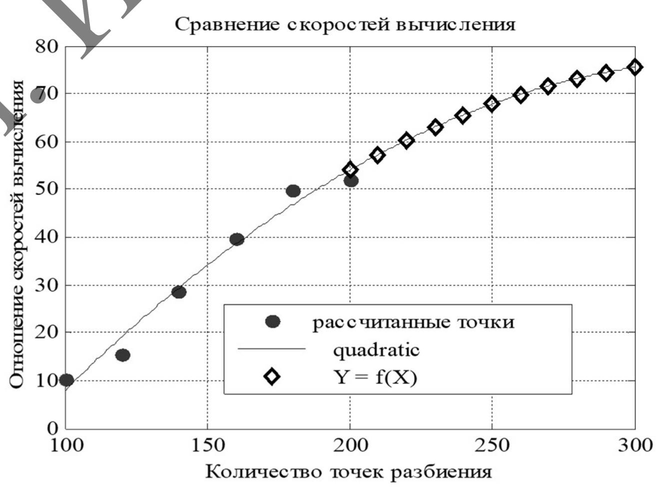


Рисунок 1. – Динамика изменения скорости вычислений при увеличении количества точек разбиения по осям *x* и *y*.

Исследуем динамику повышения скорости вычислений при увеличении количества разбиений расчетной области. Из рисунка 1 видно, что «выигрыш» в скорости повышается, рассчитанные точки в интервале количества разбиений от 100 до 200. Далее с помощью встроенной функции Basic Fitting проведем аппроксимацию полученных точек полиномом второй степени и экстраполируем исходные данные в интервале количества разбиений от 200 до 300. Из рисунка 1 видно, что «выигрыш» в скорости увеличивается при дальнейшем увеличении количества точек разбиения.

Следует отметить, что на разных компьютерах данные могут отличаться от полученных в данной работе, так как скорость вычисления зависит от характеристик компьютера и версии MATLAB. Однако увеличение скорости вычислений также будет наблюдаться.

Таким образом, в работе показана одна из возможностей оптимизировать программный код в MATLAB для увеличения скорости его выполнения, что является актуальной проблемой при написании программ связанных с различными численными расчетами.

ЛИТЕРАТУРА

1. Ануфриев, И. Е. Matlab 7. Наиболее полное руководство / И. Е. Ануфриев, Л. Б. Смирнов, Е. Н. Смирнова. – СПб. : БХВ-Петербург, 2005. – 1104 с.

2. Поршнев, С. В. Компьютерное моделирование физических процессов в пакете MATLAB. / С. В. Поршнев. – М. : Горячая Линия – Телеком, 2003. – 592 с.

3. Поршнев, С. В. Компьютерное моделирование физических процессов с использованием пакета MathCAD / С. В. Поршнев. – М. : Горячая Линия – Телеком, 2004. – 319 с.