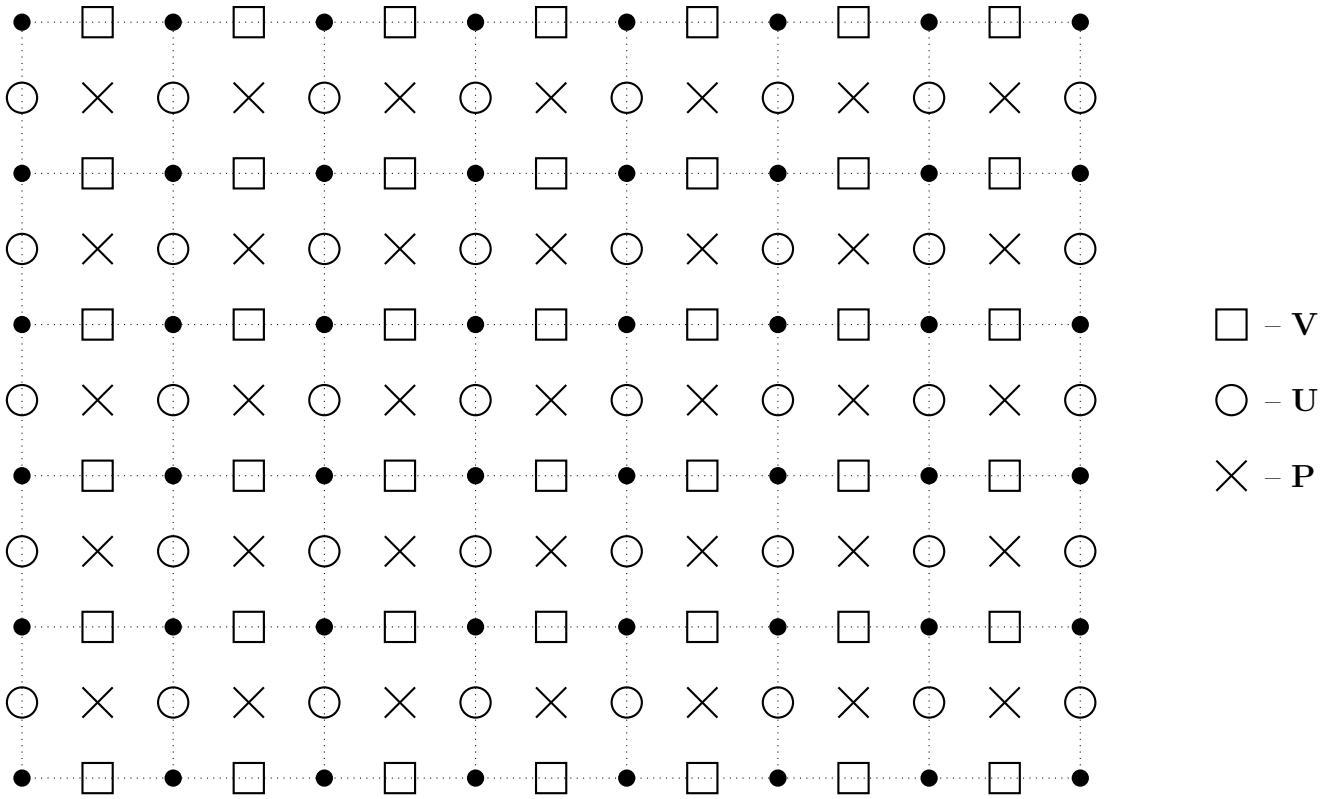

$$U[0...N_x][0...N_y-1] \quad U[i][j] \sim u(x_i,\, u_j), \text{ где } x_i = i \cdot h_x, \quad i = 0...N_x$$
$$y_j = \left(j + \frac{1}{2}\right) \cdot h_y, \quad j = 0...N_y - 1$$
$$V[0...N_x-1][0...N_y] \quad V[i][j] \sim v(x_i,\, u_j), \text{ где } x_i = \left(i + \frac{1}{2}\right) \cdot h_x, \quad i = 0...N_x - 1$$
$$y_j = j \cdot h_y, \quad j = 0...N_y$$
$$P[0...N_x-1][0...N_y-1] \quad P[i][j] \sim p(x_i,\, u_j), \text{ где } x_i = \left(i + \frac{1}{2}\right) \cdot h_x, \quad i = 0...N_x - 1$$
$$y_j = \left(j + \frac{1}{2}\right) \cdot h_y, \quad j = 0...N_y - 1$$

Listing 1: Differential Operators

```
1  void udxdx (double *u_out, double *u_in, int N_x, int N_y, double h_x, double h_y,
       double L_x, double L_y) {
2  ...
3  for (i = 0; i < N_y; ++i) {
4    u_out[i*(N_x+1)+0] = 0;
5    u_out[i*(N_x+1)+N_x] = 0;
6  }
7
8  for (i = 0; i < N_y; ++i)
9    for (j = 1; j < N_x; ++j)
10     u_out[i*(N_x+1)+j] = (u_in[i*(N_x+1)+j+1]-2u_in[i*(N_x+1)+j]+u_in[i*(N_x+1)+j-
         1])/(h_x*h_x);
11 ...
12   return;
13 }
14
15 void udydy (double *u_out, double *u_in, int N_x, int N_y, double h_x, double h_y,
       double L_x, double L_y) {
16 ...
17 for (j = 0; j < N_x+1; ++j) {
18   u_out[j] = (u_in[(N_x+1)+j]-u_in[j])/(h_y*h_y);
19   u_out[(N_y-1)*(N_x+1)+j] = (u_in[(N_y-1)*(N_x+1)+j]-u_in[(N_y-2)*(N_x+1)+j])/(h_y*
       h_y);
20 }
21
22 for (i = 1; i < N_y-1; ++i)
23   for (j = 0; j < N_x+1; ++j)
24     u_out[i*(N_x+1)+j] = (u_in[(i+1)*(N_x+1)+j]-2u_in[i*(N_x+1)+j]+u_in[(i-1)*(N_x
         +1)+j])/(h_y*h_y);
25 ...
26   return;
27 }
28
29
30
```

```
31 void udxdy (double *u_out, double *u_in, int N_x, int N_y, double h_x, double h_y,
       double L_x, double L_y) {
32 ...
33 for (i = 0; i < N_y; ++i) {
34   u_out[i*(N_x+1)+0] = 0;
35   u_out[i*(N_x+1)+N_x] = 0;
36 }
37
38 for (j = 0; j < N_x+1; ++j) {
39   u_out[j] = 0;
40   u_out[(N_y-1)*(N_x+1)+j] = 0;
41 }
42
43 for (i = 1; i < N_y-1; ++i)
44   for (j = 1; j < N_x; ++j)
45     u_out[i*(N_x+1)+j] = (u_in[(i+1)*(N_x+1)+j+1]-u_in[(i+1)*(N_x+1)+j-1]-u_in[(i-1)
         *(N_x+1)+j+1]+u_in[(i-1)*(N_x+1)+j-1])/(h_x*h_y);
46 ...
47   return;
48 }
49
50 void vdxdx (double *v_out, double *v_in, int N_x, int N_y, double h_x, double h_y,
       double L_x, double L_y) {
51 ...
52 for (i = 0; i < N_y+1; ++i) {
53   v_out[i*N_x + 0] = (v_in[i*N_x + 0]-v_in[i*N_x + 1])/(h_x*h_x);
54   v_out[i*N_x+N_x-1] = (v_in[*N_x+N_x 2]-v_in[i*N_x+N_x-1])/(h_x*h_x);
55 }
56
57 for (i = 0; i < N_y+1; ++i)
58   for (j = 1; j < N_x-1; ++j)
59     v_out[i*(N_x)+j] = (v_in[i*(N_x)+j+1]-2v_in[i*(N_x)+j]+v_in[i*(N_x)+j-1])/(h_x*
         h_x);
60 ...
61   return;
```

```
62 }
63
64 void vdydy (double *v_out, double *v_in, int N_x, int N_y, double h_x, double h_y,
       double L_x, double L_y) {
65 ...
66 for (j = 0; j < N_x; ++j) {
67   v_out[j] = 0;
68   v_out[N_y*N_x+j] = 0;
69 }
70
71 for (i = 1; i < N_y; ++i)
72   for (j = 0; j < N_x; ++j)
73     v_out[i*N_x+j] = (v_in[(i+1)*N_x+j]-2v_in[i*N_x+j]+v_in[(i-1)*N_x+j])/(h_y*h_y)
         ;
74 ...
75   return;
76 }
77
78
79
80 void vdxdy (double *v_out, double *v_in, int N_x, int N_y, double h_x, double h_y,
       double L_x, double L_y) {
81 ...
82 for (i = 0; i < N_y+1; ++i) {
83   v_out[i*N_x+0] = 0;
84   v_out[i*N_x+N_x-1] = 0;
85 }
86
87 for (j = 0; j < N_x; ++j) {
88   v_out[j] = 0;
89   v_out[N_y*N_x+j] = 0;
90 }
91
92 for (i = 1; i < N_y; ++i)
93   for (j = 1; j < N_x-1; ++j)
```

```
94      v_out[i*N_x+j] = (v_in[(i+1)*N_x+j+1]-v_in[(i+1)*N_x+j-1]-v_in[(i-1)*N_x+j+1]+
            v_in[(i-1)*N_x+j-1])/(h_x*h_y);
95  ...
96    return;
97  }
98
99  void udx (double *p_out, double *u_in, int N_x, int N_y, double h_x, double h_y,
        double L_x, double L_y) {
100 ...
101 for (i = 0; i < N_y; ++i)
102   for (j = 0; j < N_x; ++j)
103     p_out[i*N_x+j] = (u_in[i*(N_x+1)+j+1]-u_in[i*(N_x+1)+j])/h_x;
104 ...
105   return;
106 }
107
108 void vdy (double *p_out, double *v_in, int N_x, int N_y, double h_x, double h_y,
        double L_x, double L_y){
109 ...
110 for (i = 0; i < N_y; ++i)
111   for (j = 0; j < N_x; ++j)
112     p_out[i*N_x+j] = (v_in[(i+1)*N_x+j]-v_in[i*N_x+j])/h_y;
113 ...
114   return;
115 }
116
117 void pdx (double *u_out, double *p_in, int N_x, int N_y, double h_x, double h_y,
        double L_x, double L_y){
118 ...
119 for (i = 0; i < N_y; ++i) {
120   u_out[i*(N_x+1)+0] = 0;
121   u_out[i*(N_x+1)+N_x] = 0;
122 }
123 for (i = 0; i < N_y; ++i)
124   for (j = 1; j < N_x; ++j)
```

```
125     u_out[i*(N_x+1)+j] = (p_in[i*N_x+j]-p_in[i*N_x+j-1])/h_x;
126 ...
127   return;
128 }
129
130 void pdy (double *v_out, double *p_in, int N_x, int N_y, double h_x, double h_y,
      double L_x, double L_y){
131 ...
132 for (j = 0; j < N_x; ++j) {
133   v_out[j] = 0;
134   v_out[N_y*N_x+j] = 0;
135 }
136 for (i = 1; i < N_y; ++i)
137   for (j = 0; j < N_x; ++j)
138     v_out[i*N_x+j] = (p_in[i*N_x+j]-p_in[(i-1)*N_x+j])/h_y;
139 ...
140   return;
141 }
```