

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ ПО ТЕМЕ

**«Решение дифференциальных уравнений
в частных производных»**

Выполнила:

Студентка 3 курса 2 группы

Курец Любовь Олеговна

№ варианта	$f_1(y)$	$f_2(y)$	$f_3(x)$	$f_4(x)$	$F(x,y)$
2	$\sin y$	y	x^3	x^2	xy

Параллельный алгоритм для 1D композиции

Чтобы распараллелить последовательный алгоритм, нужно параллелизовать цикл. Для этого необходимо распределить данные (в нашем случае массивы u , u_{new} , f) по процессам. Одна из простейших декомпозиций состоит в следующем: физическая область разделяется на слои, каждый из них обрабатывается отдельным процессом. Эта декомпозиция может быть описана следующим образом:

```

integer i, j, n, s, e
double precision u(0:n+1, s:e), unew(0:n+1, s:e)
do 10 j = s, e
  do 10 i = 1, n
    unew(i, j) = 0.25*(u(i-1, j)+ u(i, j+1)+ u(i, j-1) + u(i+1, j)) - h * h * f(i, j)
  10 continue

```

Здесь s, e указывают значения номеров строк слоя, за которые ответственен данный процесс.

При выполнении итераций каждый процесс использует не только элементы, принадлежащие его слою, но и элементы из смежных процессов. В этом случае организовать вычисления можно следующим образом. Расширим слой сетки для каждого процесса так, чтобы он содержал необходимые данные (рис. 8.1). Элементы среды, которые используются, чтобы сохранять данные из других процессов, называются “теньвыми”.

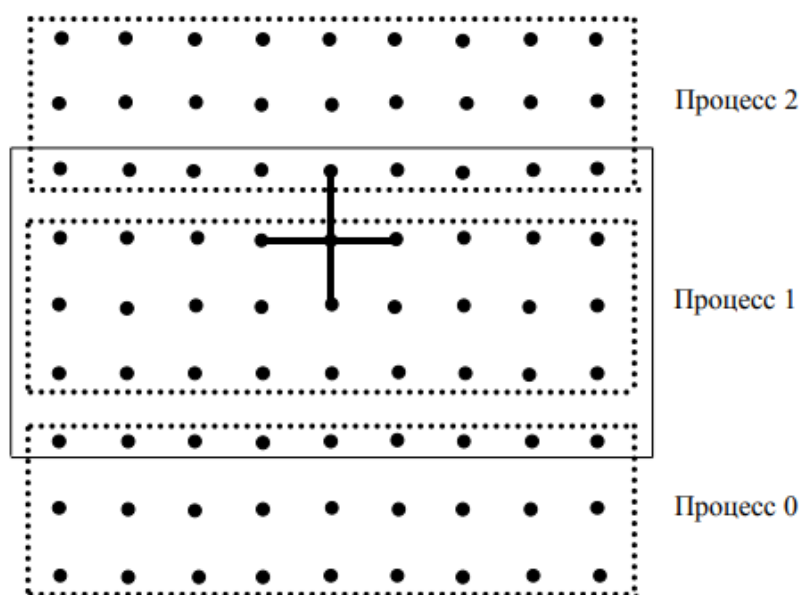


Рис. 8.1. Область с теньвыми точками для процесса 1

Определим сетку, содержащую точки (x_i, y_i) , задаваемые как:

$$x_i = \frac{i}{n+1}, i = 0, \dots, n+1, \quad y_j = \frac{j}{n+1}, j = 0, \dots, n+1.$$

Результат

Точность – 1.0e-5;

Исходная матрица для одного процесса(n =10; m=10;):

```
0.000 0.091 0.181 0.269 0.356 0.439 0.519 0.594 0.665 0.730 0.789 0.841
0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.008
0.006 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.033
0.020 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.074
0.048 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.132
0.094 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.207
0.162 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.298
0.258 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.405
0.385 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.529
0.548 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.669
0.751 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.826
0.000 0.091 0.182 0.273 0.364 0.455 0.545 0.636 0.727 0.818 0.909 1.000
```

Для четырех процессов:

Process: 2

Matrix:

```
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.094 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.207
0.162 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.298
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
```

Process: 1

Matrix:

```
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.020 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.074
0.048 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.132
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
```

Process: 0

Matrix:

0.000	0.091	0.181	0.269	0.356	0.439	0.519	0.594	0.665	0.730	0.789	0.841
0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.008
0.006	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.033
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Number of processes: 4

Size: n= 10; m= 10

Time spent 0.005 sec

Process: 3

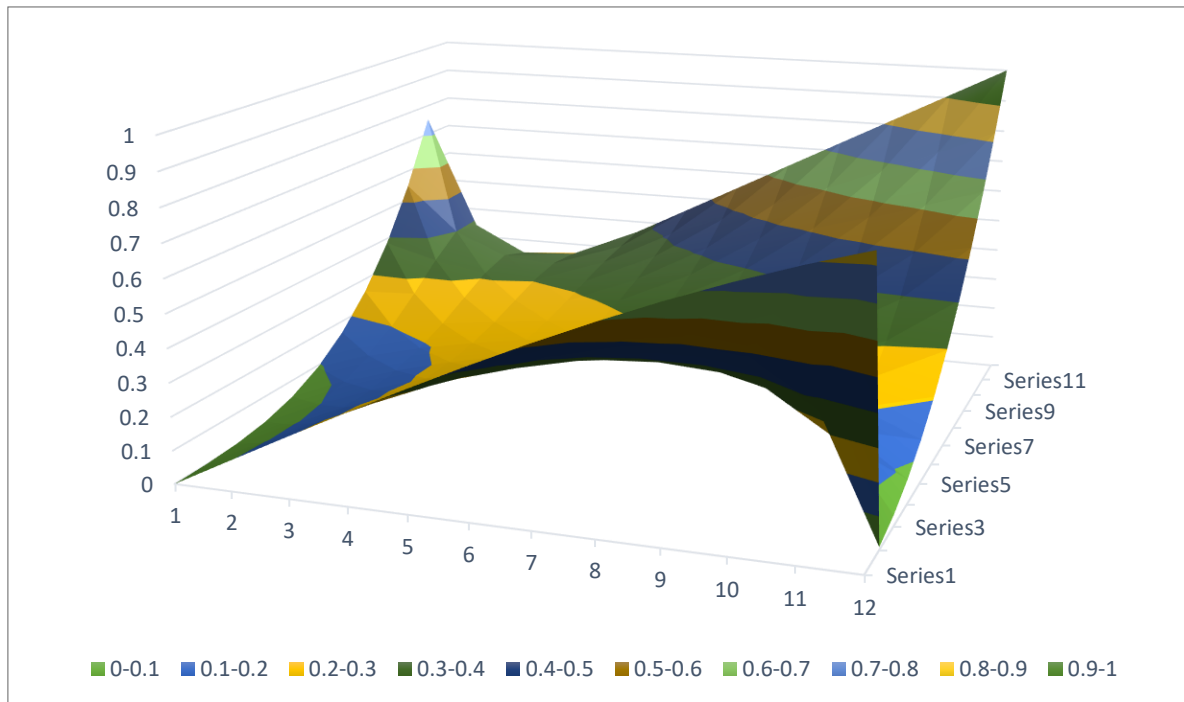
Matrix:

0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.258	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.405
0.385	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.529
0.548	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.669
0.751	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.826
0.000	0.091	0.182	0.273	0.364	0.455	0.545	0.636	0.727	0.818	0.909	1.000

Результат работы после итераций Якоби

0	0.0907839	0.180818	0.269359	0.355675	0.439054	0.518807	0.594275	0.664835	0.729904	0.788945	0.841471
0.000751315	0.0863822	0.167603	0.244311	0.316007	0.381576	0.438892	0.483909	0.508244	0.492121	0.38163	0.00826446
0.00601052	0.0866663	0.159454	0.225102	0.283569	0.333727	0.372926	0.396146	0.394304	0.351169	0.239926	0.0330579
0.0202855	0.0953702	0.159549	0.214729	0.261643	0.29959	0.326236	0.33729	0.326045	0.283252	0.199317	0.0743802
0.0480841	0.115808	0.170295	0.215103	0.251988	0.28088	0.300084	0.306496	0.295911	0.263869	0.207915	0.132231
0.0939144	0.150586	0.192923	0.226706	0.254727	0.277355	0.293315	0.300377	0.296004	0.278251	0.24718	0.206612
0.162284	0.201073	0.226859	0.248195	0.268357	0.287363	0.303679	0.315289	0.320432	0.318266	0.309612	0.297521
0.257701	0.266213	0.268543	0.275803	0.289732	0.308296	0.328618	0.348177	0.36531	0.379541	0.391886	0.404959
0.384673	0.339459	0.309139	0.302506	0.314153	0.337065	0.365828	0.39691	0.428414	0.459933	0.492566	0.528926
0.547708	0.400003	0.330434	0.317507	0.336077	0.37094	0.413862	0.460547	0.509008	0.558896	0.611385	0.669421
0.751315	0.384877	0.300015	0.308402	0.351562	0.409071	0.472905	0.539636	0.607863	0.677399	0.749249	0.826446
0	0.0909091	0.181818	0.272727	0.363636	0.454545	0.545455	0.636364	0.727273	0.818182	0.909091	1

Поверхность



Время

Процессор: Intel(R) Core i5 8th Gen, 4 ядра

Number of processes: 1	Number of processes: 4
Size: n= 100; m= 100	Size: n= 100; m= 100
Time spent 1.2251 sec	Time spent 0.543859 sec

Number of processes: 1	Number of processes: 4
Size: n= 200; m= 200	Size: n= 200; m= 200
Time spent 7.86407 sec	Time spent 4.94953 sec

Параллельный алгоритм для 2D композиции

Небольшая модификация программы преобразует ее из одномерной в двумерную. Прежде всего следует определить декомпозицию двумерной среды, используя функцию **MPI_CART_CREATE**.

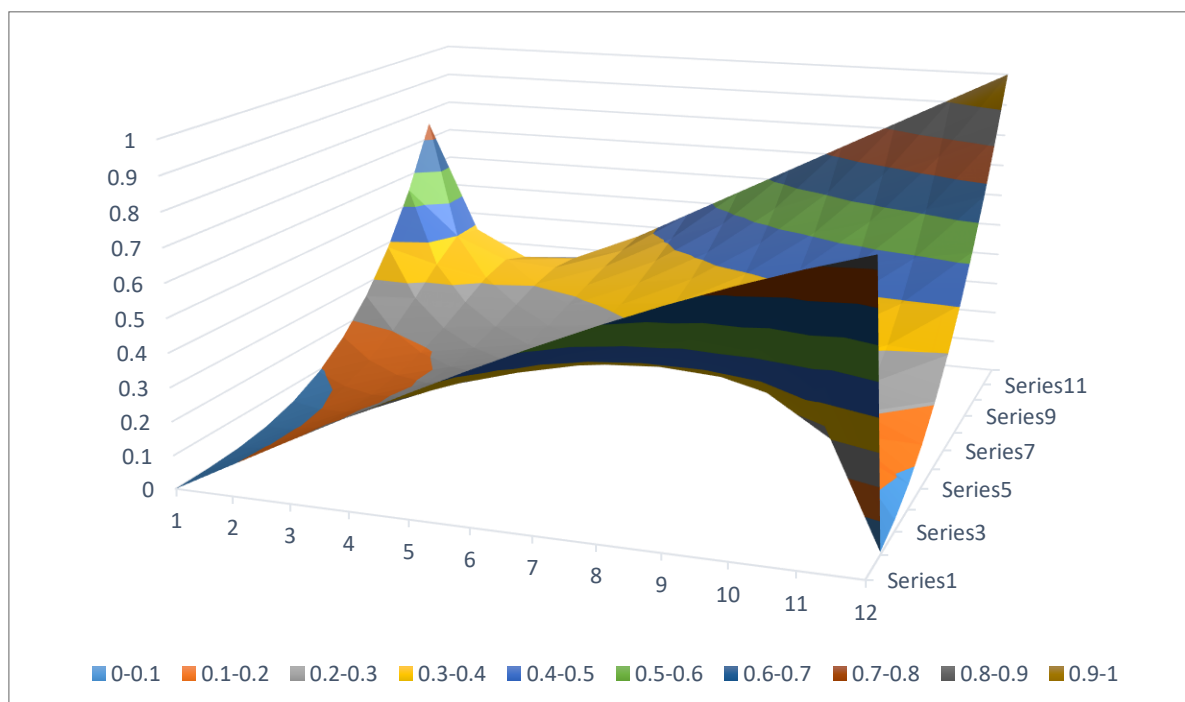
```
isperiodic(1) = .false.
isperiodic(2) = .false.
reorder      = .true.
call MPI_CART_CREATE(MPI_COMM_WORLD, 2, dims, isperiodic,
                    reorder, comm2d, ierr )
```

Для получения номеров левых **nbrleft** и правых **nbrright** процессов-соседей, так же как и верхних и нижних в предыдущем параграфе, воспользуемся функцией **MPI_CART_SHIFT**.

```
call MPI_CART_SHIFT(comm2d, 0, 1, nbrleft, nbrright, ierr )
call MPI_CART_SHIFT(comm2d, 1, 1, nbrbottom, nbrtop, ierr )
```

Матрица после преобразований:

```
0.0907839 0.180818 0.269359 0.355675 0.439054 0.518807 0.594275 0.664835 0.729904 0.788945 0.841471
0.000751315 0.0863822 0.167603 0.244311 0.316007 0.381576 0.438892 0.483909 0.508244 0.492121 0.38163 0.00826446
0.00601052 0.0866663 0.159454 0.225102 0.283569 0.333727 0.372926 0.396146 0.394304 0.351169 0.239926 0.0330579
0.0202855 0.0953702 0.159549 0.214729 0.261643 0.29959 0.326236 0.33729 0.326045 0.283252 0.199317 0.0743802
0.0480841 0.115808 0.170295 0.215103 0.251988 0.28088 0.300084 0.306496 0.295911 0.263869 0.207915 0.132231
0.0939144 0.150586 0.192923 0.226706 0.254727 0.277355 0.293315 0.300377 0.296004 0.278251 0.24718 0.206612
0.162284 0.201073 0.226859 0.248195 0.268357 0.287363 0.303679 0.315289 0.320432 0.318266 0.309612 0.297521
0.257701 0.266213 0.268543 0.275803 0.289732 0.308296 0.328618 0.348177 0.36531 0.379541 0.391886 0.404959
0.384673 0.339459 0.309139 0.302506 0.314153 0.337065 0.365828 0.39691 0.428414 0.459933 0.492566 0.528926
0.547708 0.400003 0.330434 0.317507 0.336077 0.37094 0.413862 0.460547 0.509008 0.558896 0.611385 0.669421
0.751315 0.384877 0.300015 0.308402 0.351562 0.409071 0.472905 0.539636 0.607863 0.677399 0.749249 0.826446
0 0.0909091 0.181818 0.272727 0.363636 0.454545 0.545455 0.636364 0.727273 0.818182 0.909091 1
```



Время

Процессор: Intel(R) Core i5 8th Gen, 4 ядра

Number of processes: 1 Size: n= 100; m= 100 Time spent 0.99832 sec	Number of processes: 4 Size: n= 100; m= 100 Time spent 0.681852 sec
Number of processes: 1 Size: n= 200; m= 200 Time spent 7.64539 sec	Number of processes: 4 Size: n= 200; m= 200 Time spent 5.86235 sec