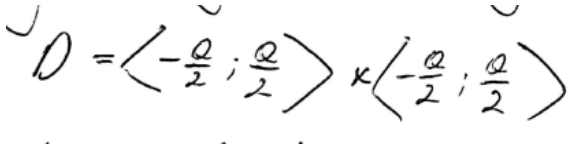
Kacper Kafara, Algorytmy równoległe, lab. 3

Wstęp

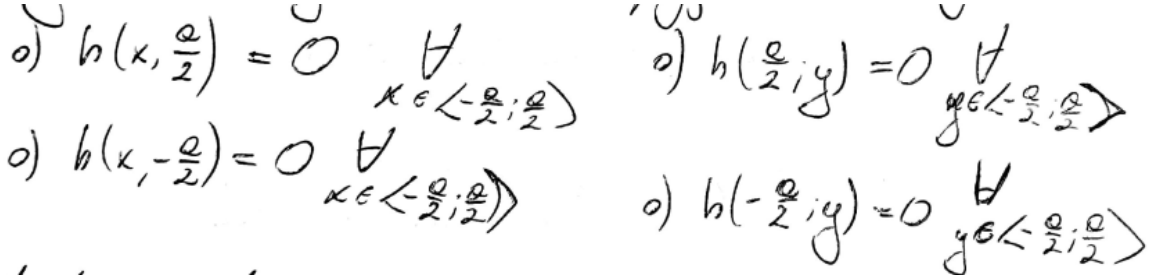
W ramach zadania należało zaimplementować równoległy algorytm rozwiązujący problem wybrany na lab. 1. W moim przypadku było to równanie opisujące odkształcenie membrany. Prawą stronę równania (iloraz ciśnienia i temperatury) określamy zbiorczym parametrem theta:

$$\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} = \theta$$

Rozwiązanie przedstawione w ramach lab. 1 zakłada rozwiązywanie tego równania metodą Jacobiego w domenie D:



Przy warunkach brzegowych:



Implementacja

Implementacja różni się nieznacznie od koncepcji przedstawionej w lab. 1: w każdej iteracji algorytmu wymieniane są tylko dwa komunikaty, ale za to wymieniane jest więcej danych niż to potrzebne. Różni się także dekompozycja - łatwiej implementacyjnie było podzielić problem (membranę) wzdłóż osi X, unikamy wtedy konieczności obsługi “trójkątów” (opisane w lab. 1, gdzie rozwiązujemy równanie tylko w pierwszej ćwiartce pod y = x). Na listingu poniżej przedstawiam zaimplementowany algorytm.

```python
#!/usr/bin/env python

import sys
import numpy as np
from mpi4py import MPI

class Args:
    def __init__(self):
        assert len(sys.argv) == 5
        # Points per process
        self.ppc: int = sys.argv[1]
        # Membrane side length
        self.a: float = sys.argv[2]
        self.theta: float = sys.argv[3]
        # Iteration count of the Jacobi method
        self.iters: int = sys.argv[4]

def compute(comm, rank, size, delta, ppc, theta, iters):
    """
    :param delta: resolution of the grid
    """

    side = rank * ppc
    # Rectangle this process is responsible for
    H = np.zeros((ppc, side))

    x_min = 0 if rank > 0 else 1
    x_max = ppc if rank < size - 1 else ppc - 1

    for i in range(iters):
        H_i = np.zeros((ppc, side), dtype=np.float64)
        if i > 0:
            # We receive values from last iteration from our neighs
            recv_buff = np.empty(side, dtype=np.float64)
            if rank > 0:
                comm.Recv(recv_buff, rank - 1, i - 1)
                H_i[0] += recv_buff
            if rank < size - 1:
                comm.Recv(recv_buff, rank + 1, i - 1)
                H_i[-1] += recv_buff

        # We apply the formula
        H_i[x_min: x_max, 1: side - 1] -= delta ** 2 * theta
        H_i[x_min:x_max, 1: side - 1] += H[x_min:x_max, 0: side - 2]
        H_i[x_min:x_max, 1: side - 1] += H[x_min:x_max, 2:side]
        H_i[1:x_max, :] += H[0: x_max - 1, :]
        H_i[x_min: ppc - 1, :] += H[x_min + 1: ppc, :]

        H_i /= 4
        H = H_i

        if rank > 0:
            comm.Isend(H[0].copy(), rank - 1, i)

        if rank < size - 1:
            comm.Isend(H[-1].copy(), rank + 1, i)

    return H

def main():
    args = Args()
    comm = MPI.COMM_WORLD
    rank = comm.Get_rank()
    size = comm.Get_size()

    delta = args.a / (args.ppc * size - 1)
    stride = compute(comm, rank, size, delta, args.ppc, args.theta, args.iters)
    recv_buff = comm.gather(stride, root=0)
    if rank == 0:
        recv_buff = np.concatenate(recv_buff, axis=0)
        np.savetxt(sys.stdout.buffer, recv_buff)

if __name__ == "__main__":
    main()
```

Listing 1: Implementacja algorytmu rozwiązującego postawiony problem

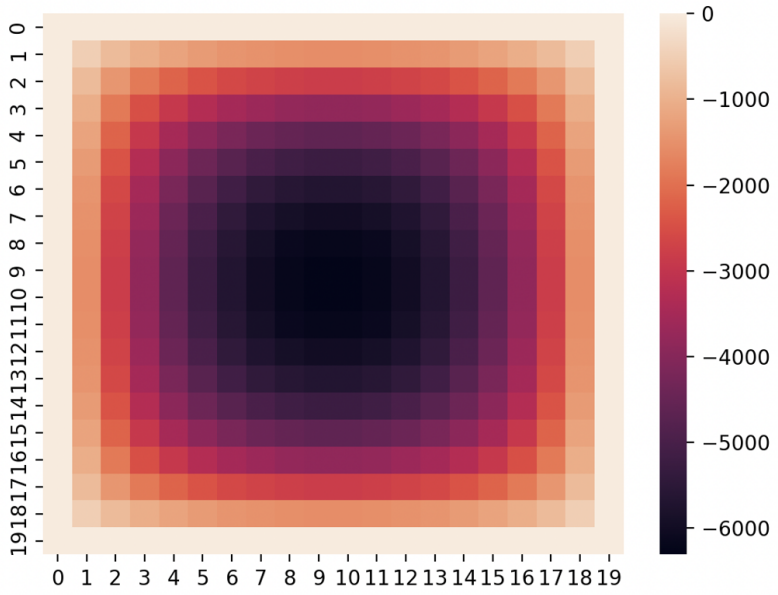
Wyniki

Obliczenia zostały wykonane na komputerze Ares. Program został uruchomiony z parametrami:

- –nodes=1, –ntasks-per-node=4,
- 20 punktów na proces,
- 20 długość boku membrany,
- 300 - wartość parametru theta,
- 100 iteracji w metodzie iteracyjnej.

Punkty na proces oznaczają rozdzielczość siatki dla pojedynczego procesu (cała długość membrany dzielona jest na n procesów, a taki prostokąt z kolei jest dzielony na siatkę na podstawie tego parametru).

Wyniki liczbowe wrzucam na osobnej stronie na samym końcu, z uwagi na ich objętość. Poniżej zamieszczam heatmapę obrazującą wyniki.



Rysunek 1: Odkształcenie membrany (jednostki pominięte jako, że nie potrzebujemy interpretować tych wyników)

Widzimy, że z pewnością mogliśmy wykonać obliczenia dla większej liczby “punktów na proces” — poprawiłoby to rozdzielczość. Otrzymane wyniki są jednak zgodne z oczekiwaniami.

Bibliografia

1. Metody iteracyjne, metoda Jacobiego https://en.wikipedia.org/wiki/Jacobi_method
2. Opracowanie dla lab. 1
3. Wykład MOWNiT
4. Designing and Building Parallel Programs, Ian Foster, 2.3.1 Local communication, https://www.mcs.anl.gov/~itf/dbpp/text/node17.html#SECTION02330000000000000000

Dodatek - wyniki liczbowe

```
[[    0.            0.            0.            0.
      0.            0.            0.            0.
      0.            0.            0.            0.
      0.            0.            0.            0.
      0.            0.            0.            0.       ]
 [    0.         -494.79543376  -825.23959808 -1058.78824179
  -1227.89856767 -1350.75387441 -1438.66700938 -1499.06941345
  -1536.85488925 -1555.02927726 -1555.02927726 -1536.85488925
  -1499.06941345 -1438.66700938 -1350.75387441 -1227.89856767
  -1058.78824179  -825.23959808  -494.79543376     0.        ]
 [    0.          -825.23959808 -1422.27548015 -1860.3147503
  -2183.45278714 -2420.56952559 -2591.22828606 -2708.89005108
  -2782.64710639 -2818.16284704 -2818.16284704 -2782.64710639
  -2708.89005108 -2591.22828606 -2420.56952559 -2183.45278714
  -1860.3147503  -1422.27548015  -825.23959808     0.        ]
 [    0.         -1058.78824179 -1860.3147503  -2465.45159842
  -2919.85270584 -3257.02828103 -3501.43699026 -3670.71743723
  -3777.13372859 -3828.45533177 -3828.45533177 -3777.13372859
  -3670.71743723 -3501.43699026 -3257.02828103 -2919.85270584
  -2465.45159842 -1860.3147503  -1058.78824179     0.        ]
 [    0.         -1227.89856767 -2183.45278714 -2919.85270584
  -3480.96390787 -3901.59405553 -4208.65415298 -4422.34664109
  -4557.09599253 -4622.19419064 -4622.19419064 -4557.09599253
  -4422.34664109 -4208.65415298 -3901.59405553 -3480.96390787
  -2919.85270584 -2183.45278714 -1227.89856767     0.        ]
 [    0.         -1350.75387441 -2420.56952559 -3257.02828103
  -3901.59405553 -4388.89474398 -4746.83894641 -4997.03946719
  -5155.2711124  -5231.84115215 -5231.84115215 -5155.2711124
  -4997.03946719 -4746.83894641 -4388.89474398 -3901.59405553
  -3257.02828103 -2420.56952559 -1350.75387441     0.        ]
 [    0.         -1438.66700938 -2591.22828606 -3501.43699026
  -4208.65415298 -4746.83894641 -5144.14848549 -5422.88765912
  -5599.60878836 -5685.25040811 -5685.25040811 -5599.60878836
  -5422.88765912 -5144.14848549 -4746.83894641 -4208.65415298
  -3501.43699026 -2591.22828606 -1438.66700938     0.        ]
 [    0.         -1499.06941345 -2708.89005108 -3670.71743723
  -4422.34664109 -4997.03946719 -5422.88765912 -5722.48642055
  -5912.80187548 -6005.13661755 -6005.13661755 -5912.80187548
  -5722.48642055 -5422.88765912 -4997.03946719 -4422.34664109
  -3670.71743723 -2708.89005108 -1499.06941345     0.        ]
 [    0.         -1536.85488925 -2782.64710639 -3777.13372859
  -4557.09599253 -5155.2711124  -5599.60878836 -5912.80187548
  -6112.01451716 -6208.74124125 -6208.74124125 -6112.01451716
  -5912.80187548 -5599.60878836 -5155.2711124  -4557.09599253
  -3777.13372859 -2782.64710639 -1536.85488925     0.        ]
 [    0.         -1555.02927726 -2818.16284704 -3828.45533177
  -4622.19419064 -5231.84115215 -5685.25040811 -6005.13661755
  -6208.74124125 -6307.63935114 -6307.63935114 -6208.74124125
  -6005.13661755 -5685.25040811 -5231.84115215 -4622.19419064
  -3828.45533177 -2818.16284704 -1555.02927726     0.        ]
 [    0.         -1555.02927726 -2818.16284704 -3828.45533177
  -4622.19419064 -5231.84115215 -5685.25040811 -6005.13661755
  -6208.74124125 -6307.63935114 -6307.63935114 -6208.74124125
  -6005.13661755 -5685.25040811 -5231.84115215 -4622.19419064
  -3828.45533177 -2818.16284704 -1555.02927726     0.        ]
 [    0.         -1536.85488925 -2782.64710639 -3777.13372859
  -4557.09599253 -5155.2711124  -5599.60878836 -5912.80187548
  -6112.01451716 -6208.74124125 -6208.74124125 -6112.01451716
```

```
 -5912.80187548 -5599.60878836 -5155.2711124  -4557.09599253
 -3777.13372859 -2782.64710639 -1536.85488925     0.        ]
[    0.         -1499.06941345 -2708.89005108 -3670.71743723
 -4422.34664109 -4997.03946719 -5422.88765912 -5722.48642055
 -5912.80187548 -6005.13661755 -6005.13661755 -5912.80187548
 -5722.48642055 -5422.88765912 -4997.03946719 -4422.34664109
 -3670.71743723 -2708.89005108 -1499.06941345     0.        ]
[    0.         -1438.66700938 -2591.22828606 -3501.43699026
 -4208.65415298 -4746.83894641 -5144.14848549 -5422.88765912
 -5599.60878836 -5685.25040811 -5685.25040811 -5599.60878836
 -5422.88765912 -5144.14848549 -4746.83894641 -4208.65415298
 -3501.43699026 -2591.22828606 -1438.66700938     0.        ]
[    0.         -1350.75387441 -2420.56952559 -3257.02828103
 -3901.59405553 -4388.89474398 -4746.83894641 -4997.03946719
 -5155.2711124  -5231.84115215 -5231.84115215 -5155.2711124
 -4997.03946719 -4746.83894641 -4388.89474398 -3901.59405553
 -3257.02828103 -2420.56952559 -1350.75387441     0.        ]
[    0.         -1227.89856767 -2183.45278714 -2919.85270584
 -3480.96390787 -3901.59405553 -4208.65415298 -4422.34664109
 -4557.09599253 -4622.19419064 -4622.19419064 -4557.09599253
 -4422.34664109 -4208.65415298 -3901.59405553 -3480.96390787
 -2919.85270584 -2183.45278714 -1227.89856767     0.        ]
[    0.         -1058.78824179 -1860.3147503  -2465.45159842
 -2919.85270584 -3257.02828103 -3501.43699026 -3670.71743723
 -3777.13372859 -3828.45533177 -3828.45533177 -3777.13372859
 -3670.71743723 -3501.43699026 -3257.02828103 -2919.85270584
 -2465.45159842 -1860.3147503  -1058.78824179     0.        ]
[    0.          -825.23959808 -1422.27548015 -1860.3147503
 -2183.45278714 -2420.56952559 -2591.22828606 -2708.89005108
 -2782.64710639 -2818.16284704 -2818.16284704 -2782.64710639
 -2708.89005108 -2591.22828606 -2420.56952559 -2183.45278714
 -1860.3147503  -1422.27548015  -825.23959808     0.        ]
[    0.          -494.79543376  -825.23959808 -1058.78824179
 -1227.89856767 -1350.75387441 -1438.66700938 -1499.06941345
 -1536.85488925 -1555.02927726 -1555.02927726 -1536.85488925
 -1499.06941345 -1438.66700938 -1350.75387441 -1227.89856767
 -1058.78824179  -825.23959808  -494.79543376     0.        ]
[    0.             0.             0.             0.
     0.             0.             0.             0.
     0.             0.             0.             0.
     0.             0.             0.             0.
     0.             0.             0.             0.
     0.             0.             0.             0.        ]]
```