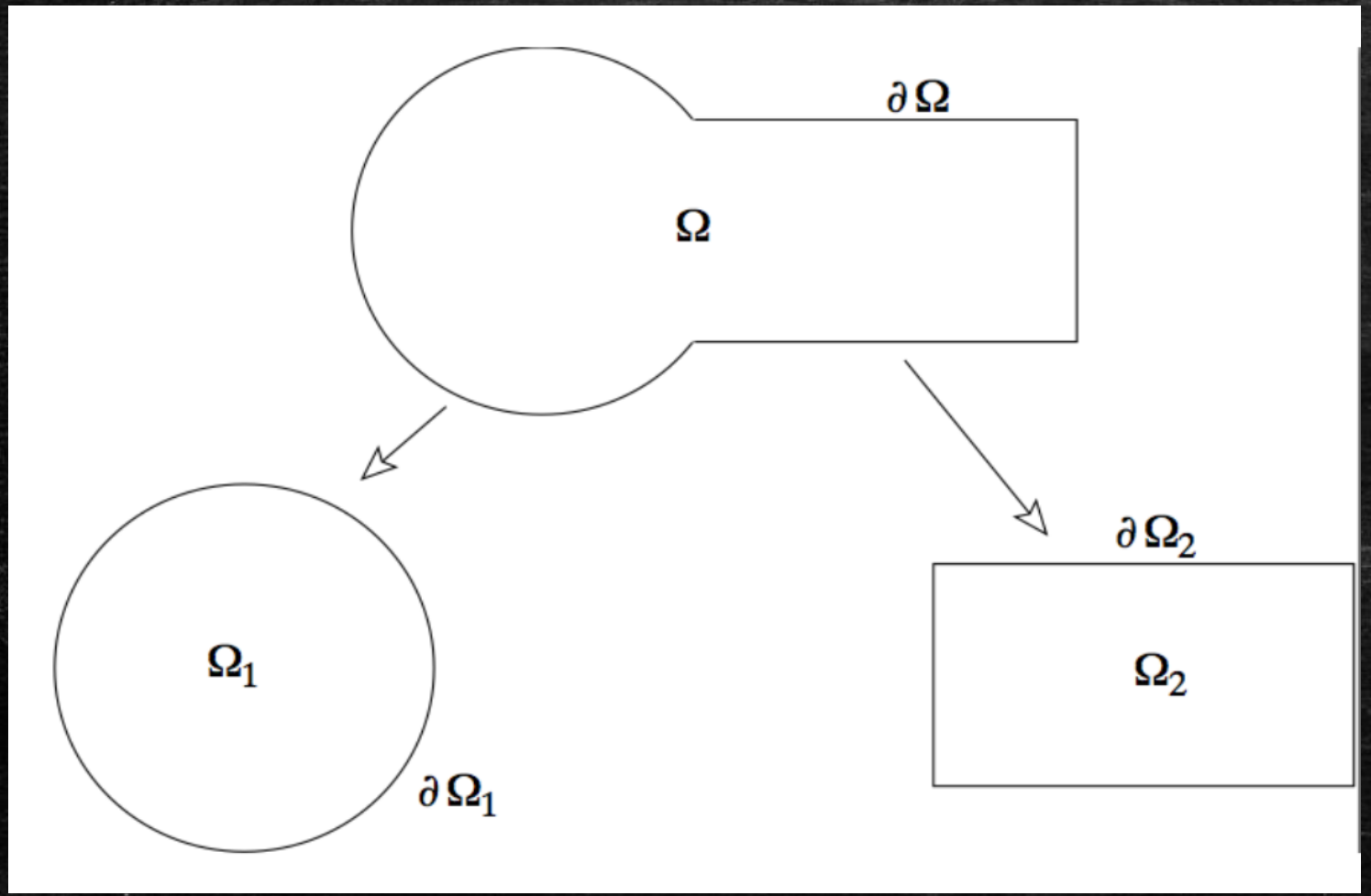
Conducción de calor en 1D empleando el Algoritmo alternante de Schwarz

Alumno: Rogelio Manuel Carrillo González
Profesor: Dr. Luis Miguel de la Cruz Salas

¿En qué consiste el algoritmo?



Dominio original

Ω

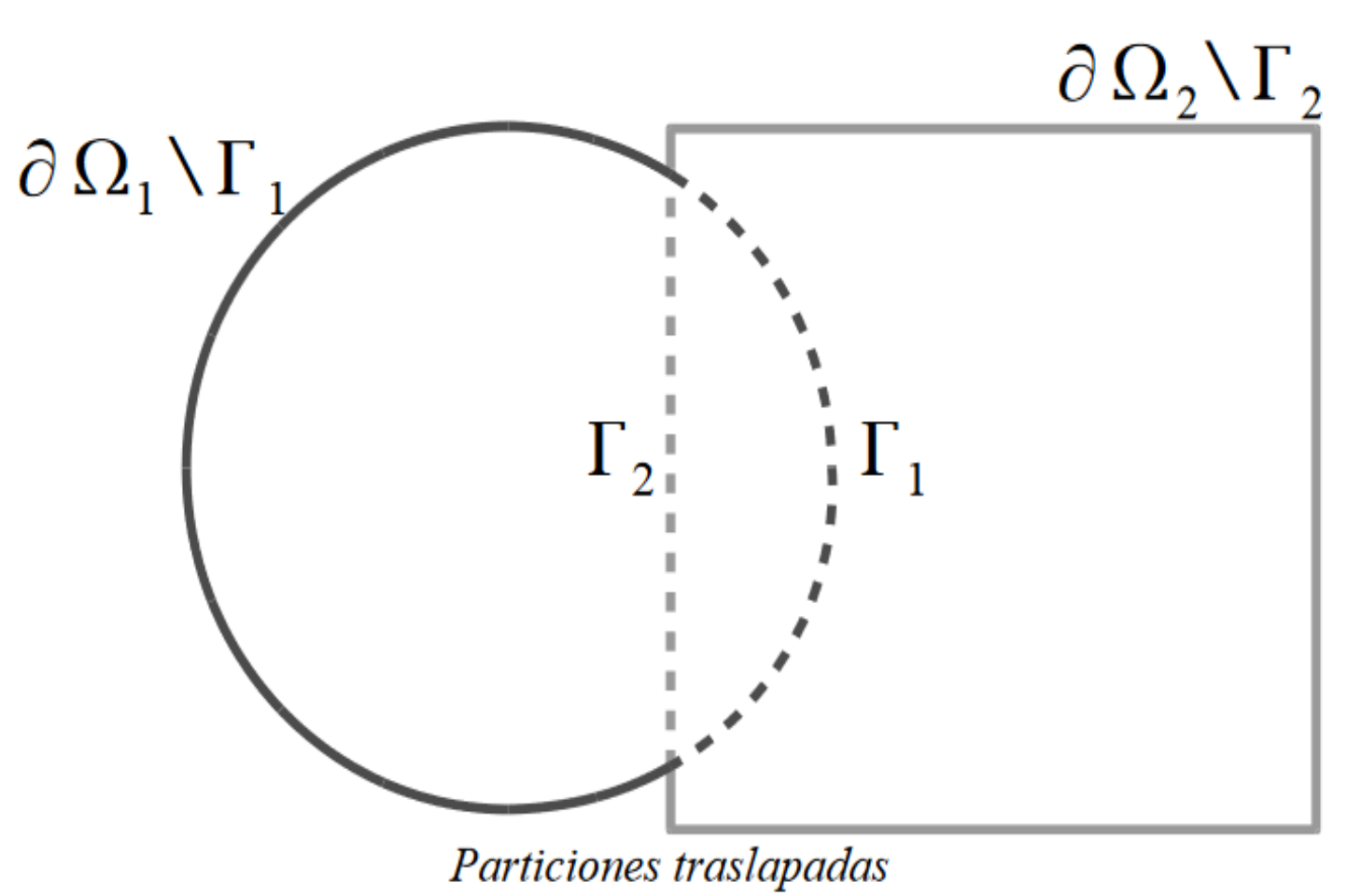
Particiones

Ω_1 y Ω_2

Fronteras

$\delta\Omega_1$ y $\delta\Omega_2$

¿En qué consiste el algoritmo?



Particiones traslapadas

Particiones se traslapan

$\Omega = \Omega_1 \cup \Omega_2$

Fronteras artificiales

$\Gamma_1 \; y \; \Gamma_2$

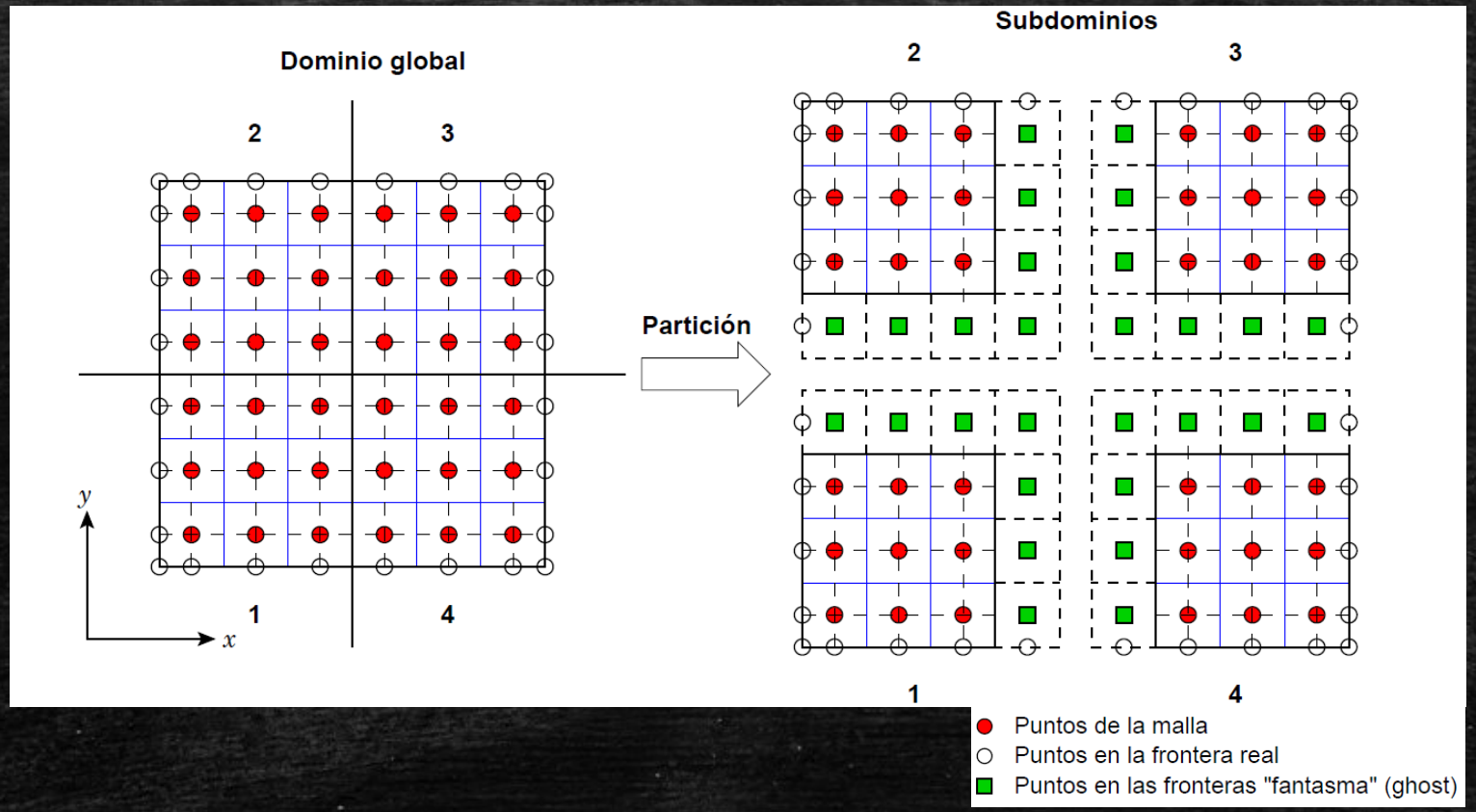
Fronteras reales

$\delta\Omega_1 \setminus \Gamma_1 \; y \; \delta\Omega_2 \setminus \Gamma_2$

¿En qué consiste el algoritmo?

- Se suponen conocidas las fronteras internas $\Gamma_1 \; y \; \Gamma_2$
- Se resuelve de manera simultanea en cada subdominio
  - Con la condición g en la frontera real $\delta\Omega_1 \backslash \Gamma_1 \; y \; \delta\Omega_2 \backslash \Gamma_2$
- Se modifican los valores de las fronteras Γ_1 y Γ_2 con iteración anterior
- Se resuelve de manera simultanea en cada subdominio
  - Con la condición g en la frontera real $\delta\Omega_1 \backslash \Gamma_1 \; y \; \delta\Omega_2 \backslash \Gamma_2$
  - Valores de la iteración anterior en las fronteras Γ_1 y Γ_2

Partición del dominio (MPI)



MPI.COMM_WORLD.Create_cart (dimm, periods, reorder)

Comunicación de los subdominios (1D)

- Get_coords( )
- Shift ( )

```
grid_rows = int(1)
grid_cols = comm.size
```

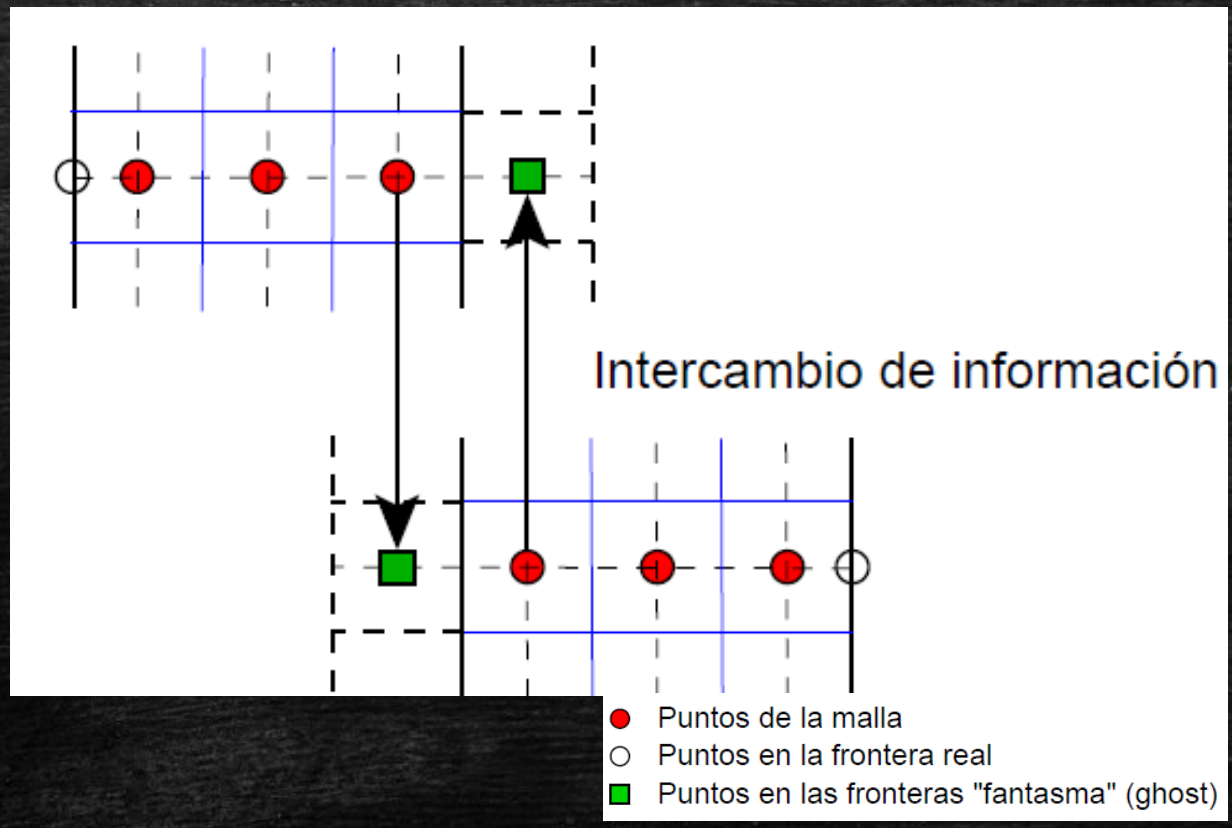
```
Building a 1 x 1 grid topology
Process = 0                row = 0                column = 0
          neighboor_processes[LEFT] = -1
          neighboor_processes[RIGHT] = -1
          neighboor = [-1, -1]
```

```
Building a 1 x 2 grid topology
Process = 0                row = 0                column = 0
          neighboor_processes[LEFT] = -1
          neighboor_processes[RIGHT] = 1
          neighboor = [-1, 1]
Process = 1                row = 0                column = 1
          neighboor_processes[LEFT] = 0
          neighboor_processes[RIGHT] = -1
          neighboor = [0, -1]
```

```
Building a 1 x 3 grid topology
Process = 0                row = 0                column = 0
          neighboor_processes[LEFT] = -1
          neighboor_processes[RIGHT] = 1
          neighboor = [-1, 1]
Process = 1                row = 0                column = 1
          neighboor_processes[LEFT] = 0
          neighboor_processes[RIGHT] = 2
          neighboor = [0, 2]
Process = 2                row = 0                column = 2
          neighboor_processes[LEFT] = 1
          neighboor_processes[RIGHT] = -1
          neighboor = [1, -1]
```

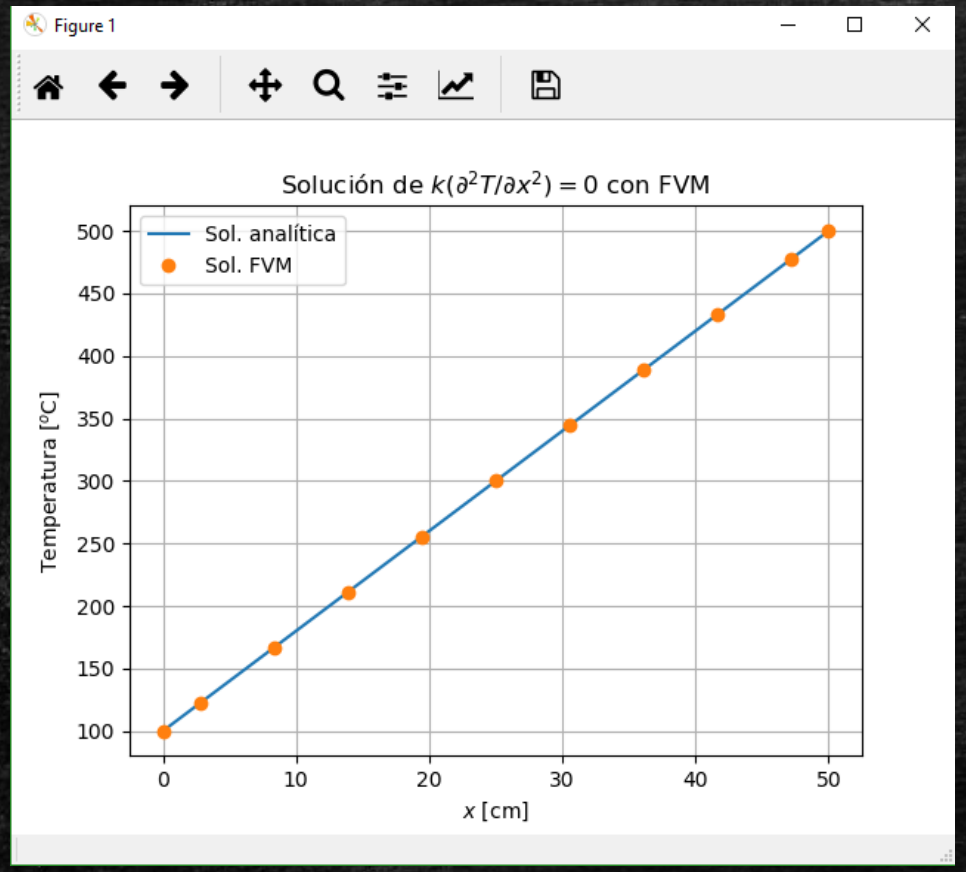
mpiexec -n numproc python programa.py

Comunicación de los subdominios (1D)



comm.sendrecv(data, dataType,dest,tag,origen,tag)

Resultados grid 1x1




```
Building a 1 x 1 grid topology
Solucionaremos la T =  [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.] en el rank = 0
ahora t vale =  [100.   0.   0.   0.   0.   0.   0.   0.   0.   0. 500.]
Solución para T= [100.         122.22222222 166.66666667 211.11111111 255.55555556
 300.         344.44444444 388.88888889 433.33333333 477.77777778
 500.        ]
.------------------------------------------------------------------.
```

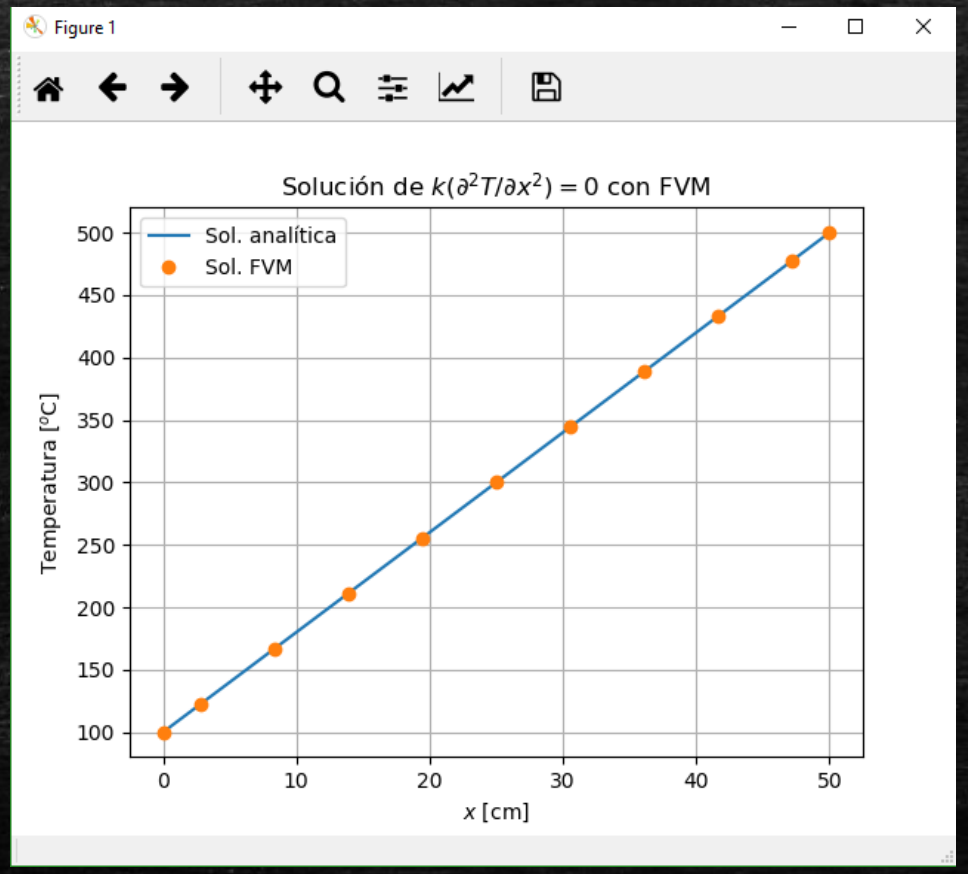
Resultados grid 1x2

```
.
Building a 1 x 2 grid topology
Solucionaremos la T =  [0. 0. 0. 0. 0. 0. 0.] en el rank = 0
ahora t vale =  [100.   0.   0.   0.   0.   0.   0.]
Solución para T= [100.  90.  70.  50.  30.  10.   0.]
.-------------------------------------------------------------.
Solución para T = [100.  95.  85.  75.  65.  55.  50.]
.-------------------------------------------------------------.
Solución para T = [100.   95.9  87.7  79.5  71.3  63.1  59. ]
.-------------------------------------------------------------.
Solución para T = [100.    99.95  99.85  99.75  99.65  99.55  99.5 ]
.-------------------------------------------------------------.
Solución para T = [100.    100.679 102.037 103.395 104.753 106.111 106.79 ]
.-------------------------------------------------------------.
Solución para T = [100.     103.9595 111.8785 119.7975 127.7165 135.6355 139.595 ]
.-------------------------------------------------------------.
Solución para T = [100.      104.54999 113.64997 122.74995 131.84993 140.94991 145.4999 ]
.-------------------------------------------------------------.
Solución para T = [100.       107.207195 121.621585 136.035975 150.450365 164.864755
 172.07195 ]
.-------------------------------------------------------------.
Solución para T = [100.        107.6854919 123.0564757 138.4274595 153.7984433 169.1694271
 176.854919 ]
.-------------------------------------------------------------.
Solución para T = [100.         109.83782795 129.51348385 149.18913975 168.86479565
 188.54045155 198.3782795 ]
.-------------------------------------------------------------.
Solución para T = [100.         110.22524844 130.67574532 151.12624219 171.57673907
 192.02723595 202.25248439]
.-------------------------------------------------------------.
Solución para T = [100.         111.96864064 135.90592192 159.8432032  183.78048448
 207.71776576 219.68640639]
.-------------------------------------------------------------.
Solución para T = [100.         112.28245124 136.84735371 161.41225618 185.97715865
```

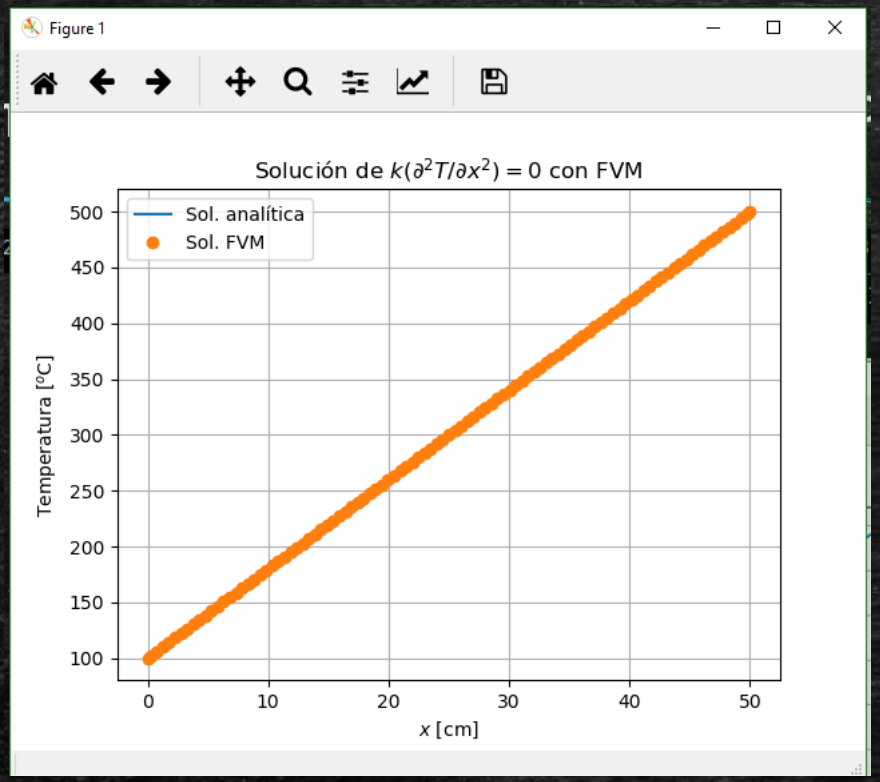
```
Solucionaremos la T =  [0. 0. 0. 0. 0. 0. 0.] en el rank = 1
ahora TT vale =  [  0.   0.   0.   0.   0.   0. 500.]
Solución para TT= [  0.  50. 150. 250. 350. 450. 500.]
.-------------------------------------------------------------.
Solución para TT = [ 10.  59. 157. 255. 353. 451. 500.]
.-------------------------------------------------------------.
Solución para TT = [ 55.   99.5 188.5 277.5 366.5 455.5 500. ]
.-------------------------------------------------------------.
Solución para TT = [ 63.1  106.79 194.17 281.55 368.93 456.31 500.  ]
.-------------------------------------------------------------.
Solución para TT = [ 99.55  139.595 219.685 299.775 379.865 459.955 500.   ]
.-------------------------------------------------------------.
Solución para TT = [106.111  145.4999 224.2777 303.0555 381.8333 460.6111 500.    ]
.-------------------------------------------------------------.
Solución para TT = [135.6355  172.07195 244.94485 317.81775 390.69065 463.56355 500.     ]
.-------------------------------------------------------------.
Solución para TT = [140.94991  176.854919 248.664937 320.474955 392.284973 464.094991
 500.      ]
.-------------------------------------------------------------.
Solución para TT = [164.864755  198.3782795 265.4053285 332.4323775 399.4594265 466.4864755
 500.       ]
.-------------------------------------------------------------.
Solución para TT = [169.1694271  202.25248439 268.41859897 334.58471355 400.75082813
 466.91694271 500.        ]
.-------------------------------------------------------------.
Solución para TT = [188.54045155 219.68640639 281.97831608 344.27022577 406.56213546
 468.85404515 500.        ]
.-------------------------------------------------------------.
Solución para TT = [192.02723595 222.82451236 284.41906517 346.01361798 407.60817079
 469.2027236  500.        ]
.-------------------------------------------------------------.
Solución para TT = [207.71776576 236.94598918 295.40243603 353.85888288 412.31532973
```

Resultados grid 1x2

```
Solucion =  [100.0, 121.05263157894733, 163.15789473684197, 205.26315789473665, 247.36842105263133,
310.52631578947336, 352.63157894736815, 394.73684210526295, 436.8421052631578, 478.9473684210526, 500.0]
```



Comparación Secuencial vs Paralelo

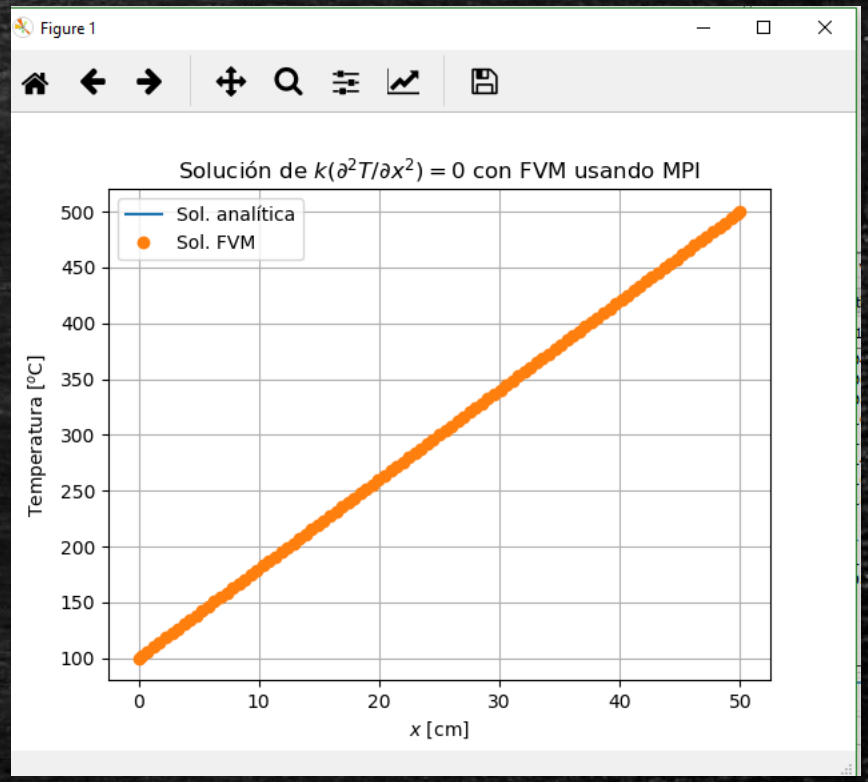


N = 100

Time = 0.006984

500

0.022998



N = 100

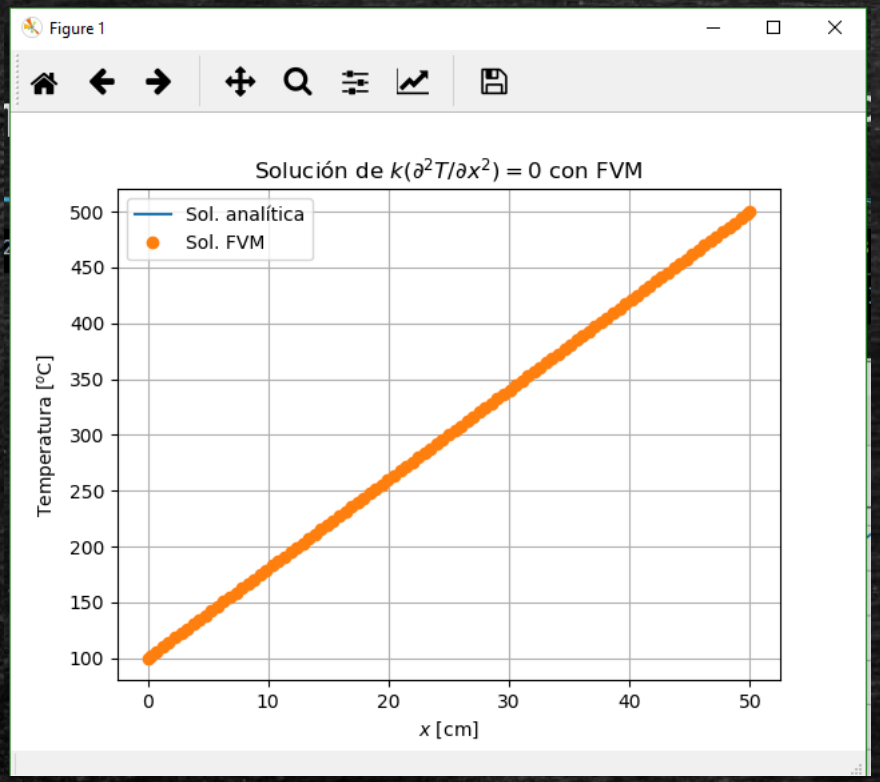
Time = 0.002298

500

0.016984

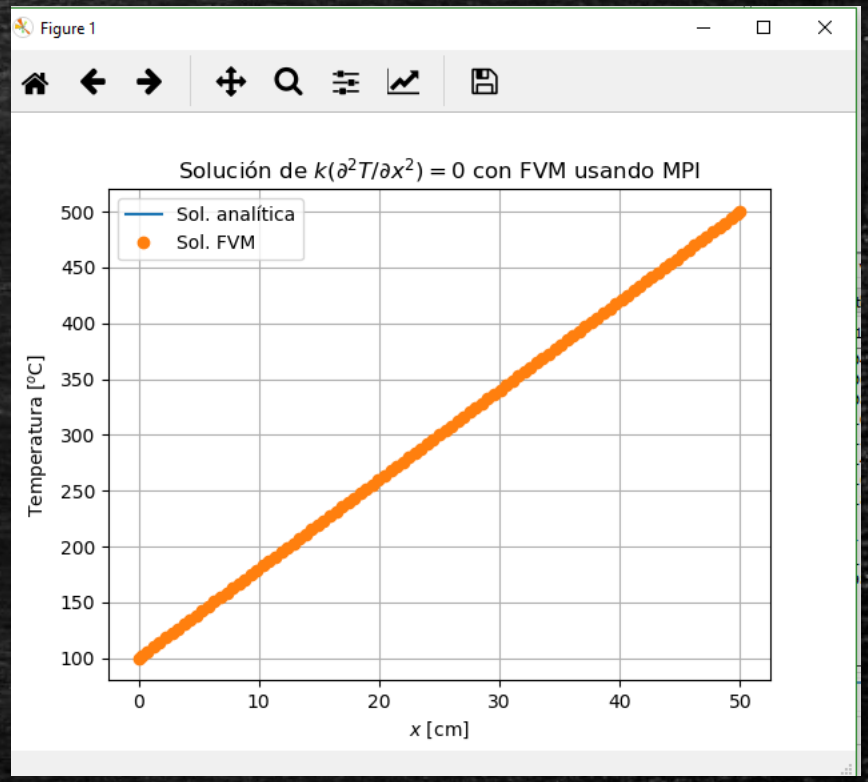
1x2 grid

Comparación Secuencial vs Paralelo



N = 1000 10,000

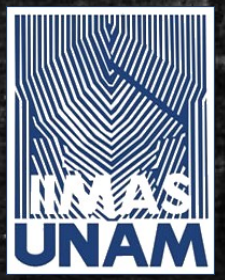
Time = 0.03298 8.43076



N = 1000 10,000

Time = 0.024974 6.808732


UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
POR MI RAZA HABLARA EL ESPIRITU


IIMAS
UNAM

Gracias