2D HEAT EQUATION SOLVER USING MPI

Saurabh Wanivadekar

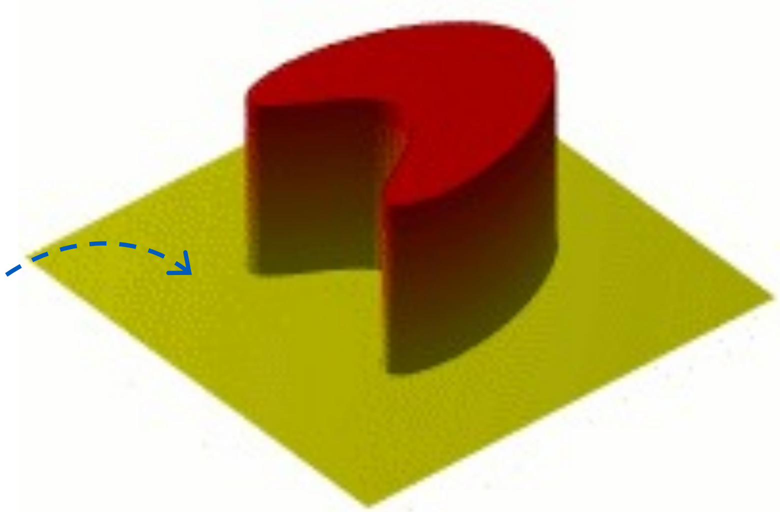
University at Buffalo The State University of New York

Partial Differential Equations

The heat equation is a PDE, an equation that relates the partial derivatives of the involved terms.

The 2D Heat Equation can be stated as:

$$\frac{\partial u}{\partial t} = \alpha \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$



Diffusion of heat in a flat plane of material. Redder is hotter.

Why are PDEs huge?

- Partial Differential Equations are great analytical models of the real world.
- One example is modelling the flow of wind in aerodynamic studies of Formula 1 cars.
- Another example is minimal surfaces.

$$(1+u_x^2)u_{yy} - 2u_x u_y u_{xy} + \left(1+u_y^2\right)u_{xx} = 0$$

Finding the Formula

- Solving the PDEs give you the underlying function that determines the exact relationship between the variables.
- There are multiple *solvers* of varying complexity and detail from Finite Difference Methods, Finite Element Methods, to Finite Volume Methods.
- To solve the 2D heat equation, we will use three methods: Jacobi, Gauss-Seidel and SOR methods and calculate the time it takes to reach L2 convergence.

```
* ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
* T_SRC0 @ X - (W/2,H)
*        |
* *******X*******
* *.............*
* *.............*
* *.............*
* *.............*
* *.............* ~ 0.0 @ all bdy by "X" (W/2,H)
* *.............*
* *.............*
* *.............*
* *.............*
* *.............*
* ***************
* 2D domain - WIDTH x HEIGHT
* "X" = T_SRC0
* "*" = 0.0
* "." = internal node suceptible to heating
* ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Formula #1: Jacobi Method

- $v_{m,l}^{n+1} = \frac{1}{4}\left(v_{m+1,l}^{n} + v_{m-1,l}^{n} + v_{m,l+1}^{n} + v_{m,l-1}^{n}\right) - \frac{h^2}{4} f_{ml}$
- The iterative method calculates the new value for each point by taking the average of its neighbors.
- f_{ml} is zero since there is no internal source of heat that is being simulated.

Formula #2: Gauss-Seidel

- $v_{m,l}^{n+1} = \frac{1}{4}\left(v_{m+1,l}^{n} + v_{m-1,l}^{n+1} + v_{m,l+1}^{n} + v_{m,l-1}^{n+1}\right) - \frac{h^2}{4} f_{ml}$
- This method is the same as Jacobi, with the exception that the neighbors are divided into reds and blacks where, reds have $m + l$ odd and blacks have $m + l$ even.
- The reds are calculated first, and the blacks are calculated using the values of the reds.

Formula #3: SOR

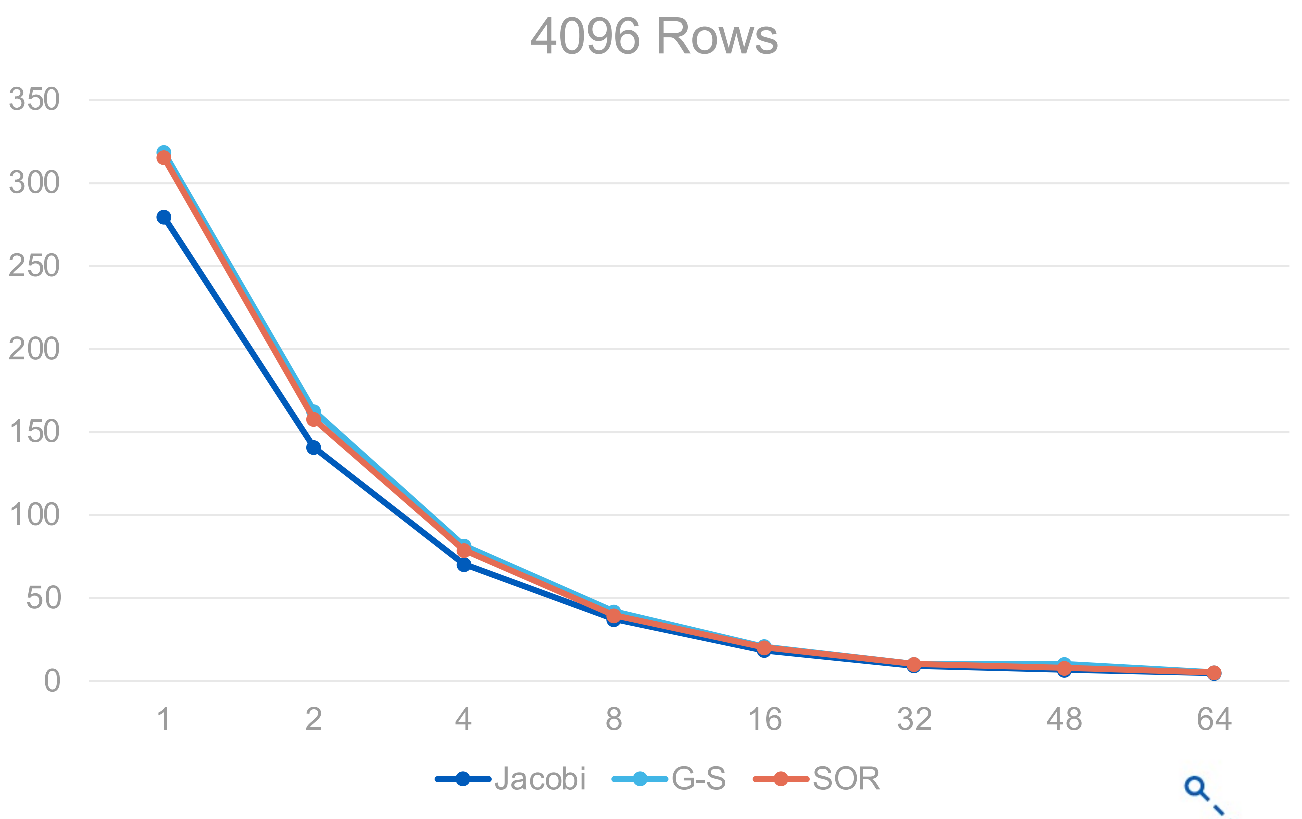
- $v_{m,l}^{n+1} = (1-w)v_{m,l}^{n} + \frac{w}{4}\left(v_{m+1,l}^{n} + v_{m-1,l}^{n+1} + v_{m,l+1}^{n} + v_{m,l-1}^{n+1}\right) - \frac{wh^2}{4} f_{ml}$
- The SOR method also uses the concept of reds and blacks.
- The value of w is kept at 1.5

Results

- The matrix is divided row wise to each node.
- The results are the measure of time taken to reach convergence, i.e., $v^{n+1} \cong v^{n}$
- The first two results are for a fixed size of matrix run on increasing number of nodes to study speedup.
- The next two results are for a fixed number of rows per node to determine scalability.

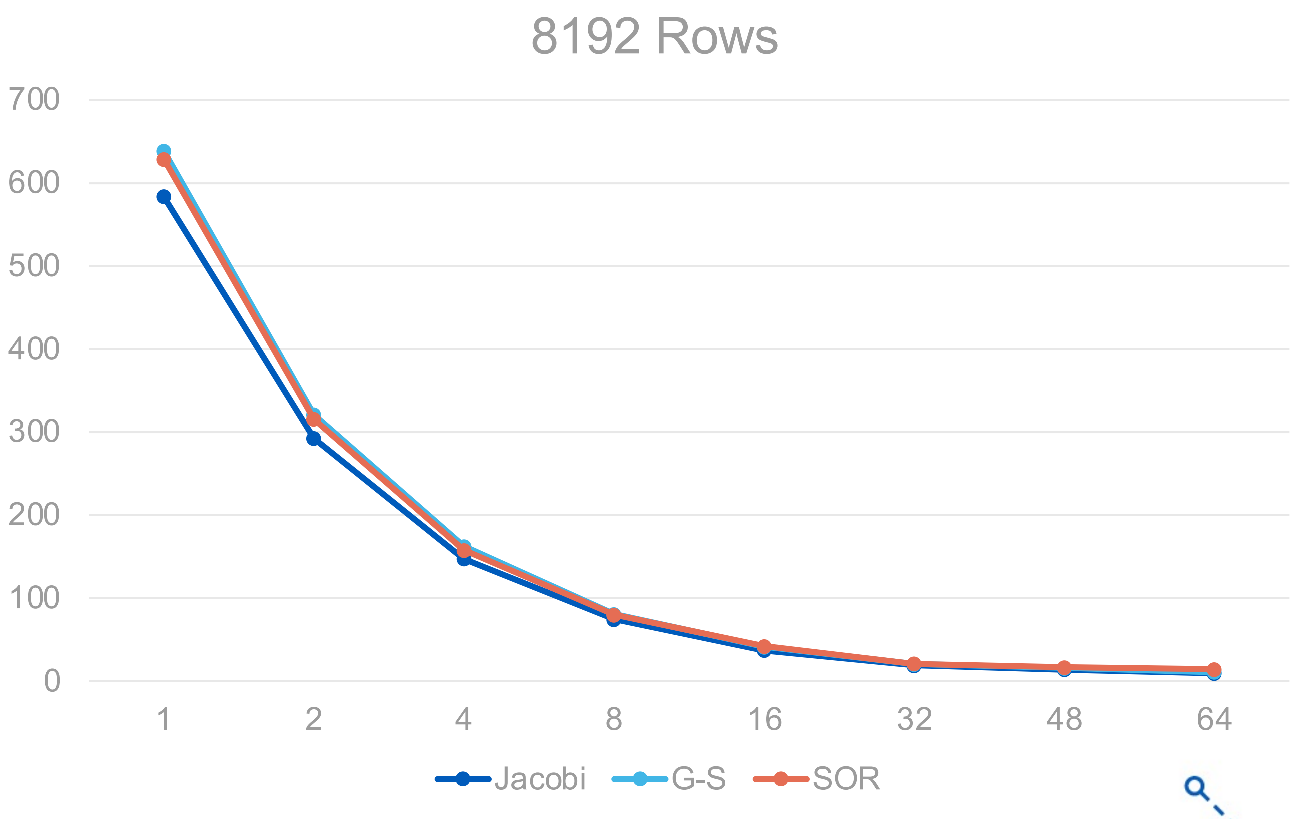
Height=4096

Nodes	Jacobi	G-S	SOR
1	279.668	318.614	315.375
2	140.816	162.556	157.668
4	70.443	81.614	78.940
8	37.184	41.901	39.607
16	18.549	20.811	20.294
32	9.374	10.317	10.270
48	6.902	10.266	7.936
64	4.900	5.354	5.242



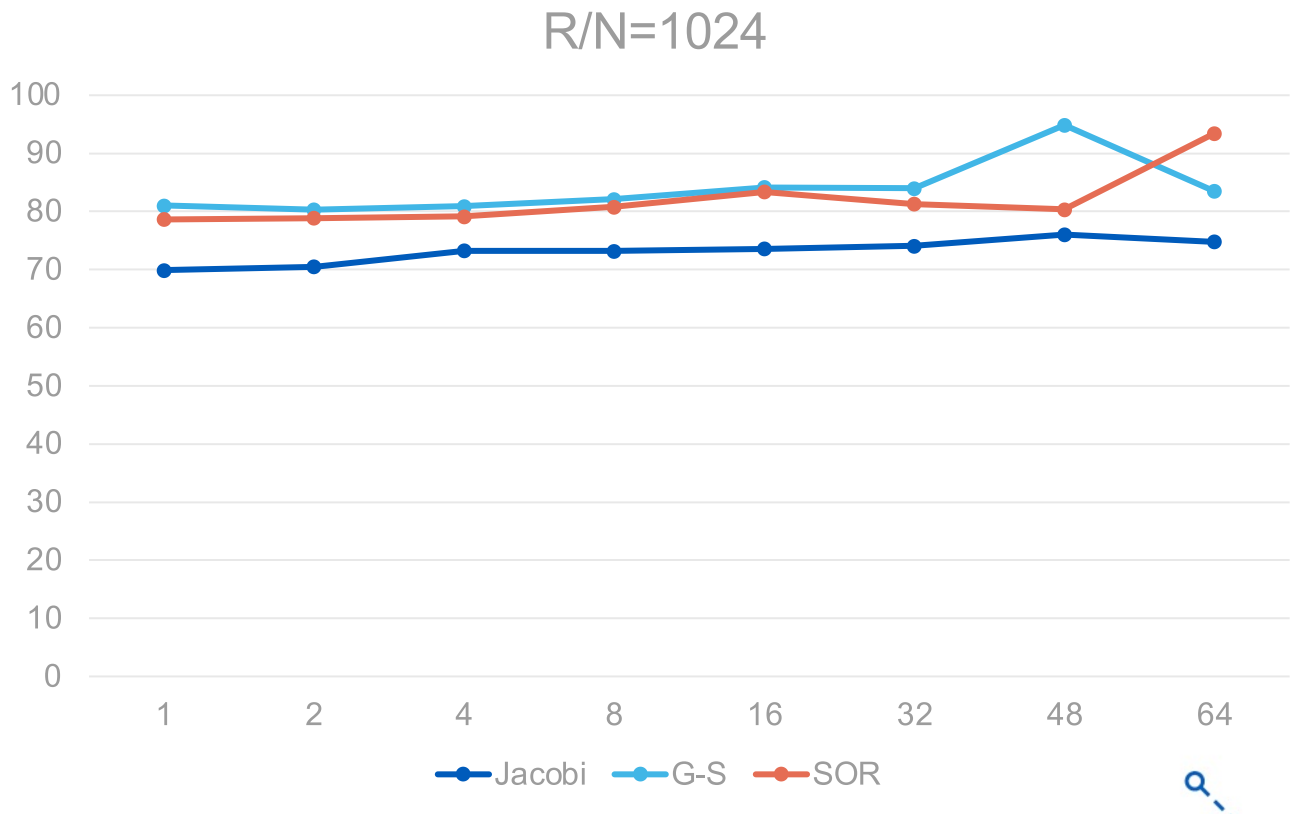
Height=8192

Nodes	Jacobi	G-S	SOR
1	583.576	638.850	628.930
2	292.181	320.893	315.263
4	147.555	162.261	157.940
8	74.685	80.854	79.753
16	37.230	41.547	42.325
32	18.917	20.428	20.914
48	13.901	15.803	16.729
64	9.593	10.842	14.529



Rows/Node=1024

Nodes	Jacobi	G-S	SOR
1	69.917	81.017	78.628
2	70.472	80.311	78.880
4	73.259	80.907	79.134
8	73.192	82.090	80.766
16	73.576	84.142	83.402
32	74.093	84.001	81.260
48	76.027	94.865	80.375
64	74.787	83.500	93.455



Rows/Node=2048

Nodes	Jacobi	G-S	SOR
1	139.738	166.376	157.176
2	147.147	160.004	157.742
4	146.148	164.240	158.072
8	146.533	161.113	159.714
16	147.303	189.807	194.131
32	150.163	168.207	166.197
48	147.483	169.836	161.968
64	150.216	169.770	246.193