COMPUTATIONAL HEAT AND FLUID FLOW (ME605)

Assignment 3: Unsteady Conduction in 1D and 2D

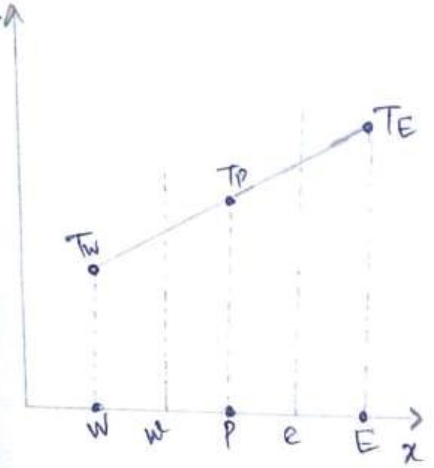
Anmoldeep Singh 180030002

Anmoldeep. Singh

180030002

Ques1 We have used a piecewise linear profile assumption for the temperature inside the control volume.

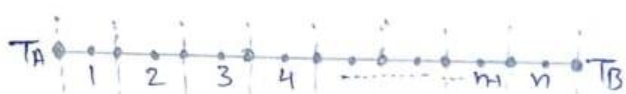
i) Grid Details



Using this profile, we estimate the temp. inside each cell. The entire system is discretized using Finite Control Volumes.

The entire domain is divided into (n-1) equal intervals with the intervals of length Δx each between 1 & n and of length $\frac{\Delta x}{2}$ between the boundaries & 1 and n respectively

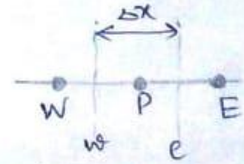
The control volume boundaries are located at the mid point of each grid length.



ii) The discretization is done using finite volume method with each element being of volume $(\Delta x \times 1 \times 1)$ units^3 for interior & the boundary elements being of volume $(\Delta x/2 \times 1 \times 1)$ units^3.

For interior points 2, 3, --, (n-2), (n-1)

Assuming: 1D, constant properties & source term = 0



Governing Eqn : $\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(k \frac{\partial T}{\partial x}\right)$ —①

Temp. at any node is : T_0 @ t=0 & T @ t+Δt. Integrating ① over space & time

$$\int_w^e \left(\rho c \int_t^{t+\Delta t} \frac{\partial T}{\partial t} dt\right) dV = \int_t^{t+\Delta t} \int_w^e \frac{\partial}{\partial x}\left(k \frac{\partial T}{\partial x}\right) dV\, dt$$

approximating $\int_t^{t+\Delta t} T_P dt$ as $[f T_P' + (1-f) T_P^\circ]\Delta t$, f = weighing factor from [0,1]

$$\Rightarrow \frac{\rho c \Delta x}{\Delta t}(T_P' - T_P^\circ) = \frac{k_e}{\delta x_e}[f T_E' + (1-f) T_E^\circ] - \frac{k_e}{\delta x_e}[f T_P' + (1-f) T_P^\circ] - \frac{k_w}{\delta x_w}[f T_P' + (1-f) T_P^\circ]$$
$$+ \frac{k_w}{\delta x_w}[f T_W' + (1-f) T_W^\circ]$$

$$\Rightarrow a_P T_P' = a_E\left(f T_E' + (1-f) T_E^\circ\right) + a_W\left(f T_W' + (1-f) T_W^\circ\right) + \left(a_P^\circ - (1-f) a_W - (1-f) a_E\right) T_P^\circ$$

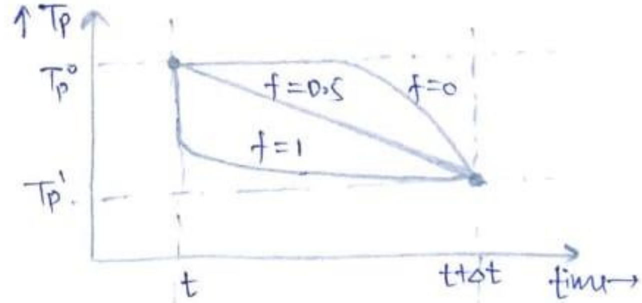
$$a_E = \frac{k_e}{\delta x_e} \; ; \; a_W = \frac{k_w}{\delta x_w} \; ; \; a_P^\circ = \rho c \frac{\Delta x}{\Delta t} \; ; \; a_P = f a_E + f a_W + a_P^\circ$$

let $\beta = \frac{\alpha \Delta t}{(\delta x)^2}$

$f \Rightarrow 0 \longrightarrow$ explicit scheme

$f = 1 \longrightarrow$ implicit scheme

$f = 0.5 \longrightarrow$ crank-nicolson scheme



for interior points : $i = 2$ to $(n-1)$

i) explicit scheme

$$\beta T_{i-1}^k + (1-2\beta) T_i^k + \beta T_{i+1}^k = T_i^{k+1}$$

k = current time step

k+1 = next time step.

ii) implicit scheme

$$-\beta T_{i-1}^{k+1} + (1+2\beta) T_i^{k+1} - \beta T_{i+1}^{k+1} = T_i^k$$

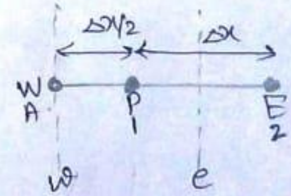
iii) crank nicolson scheme

$$-\beta T_{i-1}^{k+1} + (1+2\beta) T_i^{k+1} - \beta T_{i+1}^{k+1} = \beta T_{i-1}^k + (1-2\beta) T_i^k + \beta T_{i+1}^k$$

iii) Boundary condition implementation details

For expressions for boundaries, the governing equations are integrated over the boundaries for the left end + right end.

for left boundary



i) explicit scheme

$$(1-3\beta) T_1^k + \beta T_2^k = T_1^{k+1} - 2\beta T_A^k$$

ii) implicit scheme

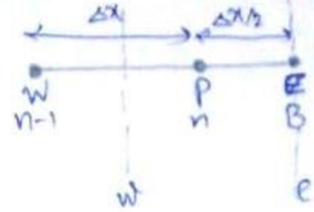
$$(1+3\beta) T_1^{k+1} - \beta T_2^{k+1} = T_1^k + 2\beta T_A^{k+1}$$

iii) crank nicolson scheme

$$(1+3\beta) T_1^{k+1} - \beta T_2^{k+1} = 2\beta T_A^k + 2\beta T_A^{k+1} + (1-3\beta) T_1^k + \beta T_2^k$$

$T_A = 100$ if $k = 0$ + $T_A = 300$ if $k > 0$

for right boundary



i) explicit scheme

$$\beta T_{n-1}^{k} + (1-3\beta) T_n^{k} = T_n^{k+1} - 2\beta T_B^{k}$$

ii) implicit scheme

$$-\beta T_{n-1}^{k+1} + (1+3\beta) T_n^{k+1} = T_n^{k} + 2\beta T_B^{k+1}$$

iii) Crank nicolson scheme

$$-\beta T_{n-1}^{k+1} + (1+3\beta) T_n^{k+1} = \beta T_{n-1}^{k} + T_n^{k} \; T_B^{k} \cdot 2\beta + 2\beta T_B^{k+1}$$

$$T_B = 100 \text{ if } k=0 \;\&\; T_B = 300 \text{ if } k>0$$

$\therefore$ The systems of equations can be represented as:

i) explicit scheme

$$\begin{bmatrix} 1-3\beta & \beta & 0 & \cdots & & 0 \\ \beta & 1-2\beta & \beta & \cdots & & \vdots \\ 0 & \beta & 1-2\beta & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & & 0 \\ & & & & 1-2\beta & \beta \\ 0 & \cdots & & 0 & \beta & 1+3\beta \end{bmatrix} \begin{bmatrix} T_1^k \\ T_2^k \\ \vdots \\ T_{n-1}^k \\ T_n^k \end{bmatrix} = \begin{bmatrix} T_1^{k+1} - 2\beta T_A^k \\ T_2^{k+1} \\ \vdots \\ T_{n-1}^{k+1} \\ T_n^{k+1} - 2\beta T_B^k \end{bmatrix}$$

ii) Implicit scheme

$$\begin{bmatrix} 1+3\beta & -\beta & 0 & \cdots & & 0 \\ -\beta & 1+2\beta & -\beta & \cdots & & \vdots \\ 0 & -\beta & \ddots & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & & 0 \\ & & & & 1+2\beta & -\beta \\ 0 & \cdots & & 0 & -\beta & 1+3\beta \end{bmatrix} \begin{bmatrix} T_1^{k+1} \\ T_2^{k+1} \\ \vdots \\ T_{n-1}^{k+1} \\ T_n^{k+1} \end{bmatrix} = \begin{bmatrix} T_1^{k} + 2\beta T_A^{k+1} \\ T_2^{k} \\ \vdots \\ T_{n-1}^{k} \\ T_n^{k} + 2\beta T_B^{k+1} \end{bmatrix}$$

iii) Crank Nicolson scheme

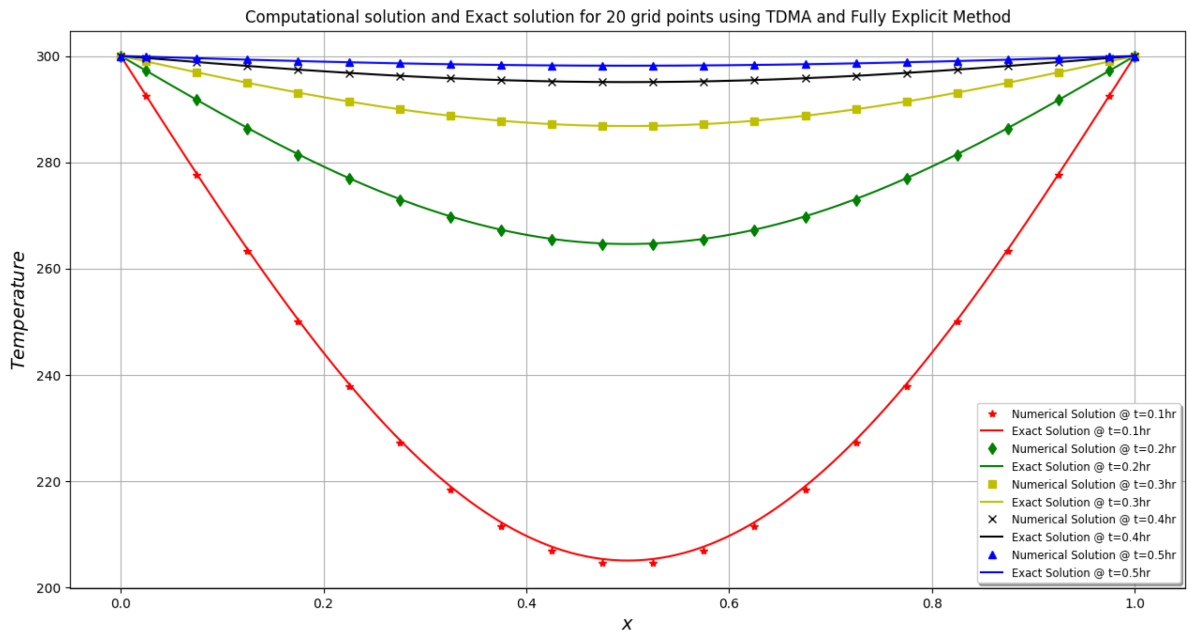
$$\begin{bmatrix} 1+3\beta & -\beta & 0 & \cdots & & 0 \\ -\beta & 1+2\beta & -\beta & \cdots & & \vdots \\ 0 & -\beta & \ddots & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & & 0 \\ & & & & 1+2\beta & -\beta \\ 0 & \cdots & & 0 & -\beta & 1+3\beta \end{bmatrix} \begin{bmatrix} T_1^{k+1} \\ T_2^{k+1} \\ \vdots \\ T_{n-1}^{k+1} \\ T_n^{k+1} \end{bmatrix} = \begin{bmatrix} T_1^{k}(1-3\beta) + \beta T_2^{k} + 2\beta(T_A^{k} + T_A^{k+1}) \\ T_1\beta + T_2^{k}(1-2\beta) + T_3^{k}\beta \\ \vdots \\ T_{n-2}^{k}\beta + T_{n-1}^{k}(1-2\beta) + T_n^{k}\beta \\ T_{n-1}^{k}\beta + T_n^{k}(1-3\beta) + 2\beta(T_B^{k} + T_B^{k+1}) \end{bmatrix}$$

Question1. Required output (plots/any other means) and Analysis of results

The temperature distribution at each 0.1 hr. interval from 0.0 to 0.5 hr.

a) Fully Explicit method

temporal step = 0.001 and spatial step = 0.05

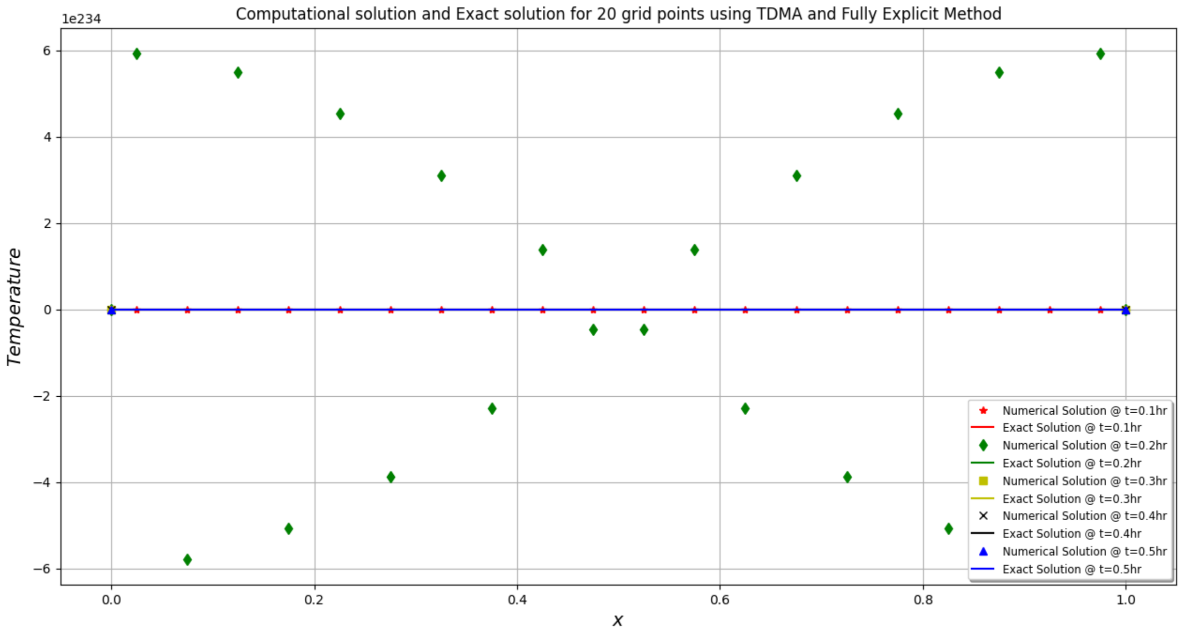


The stability criterion for fully explicit method is:

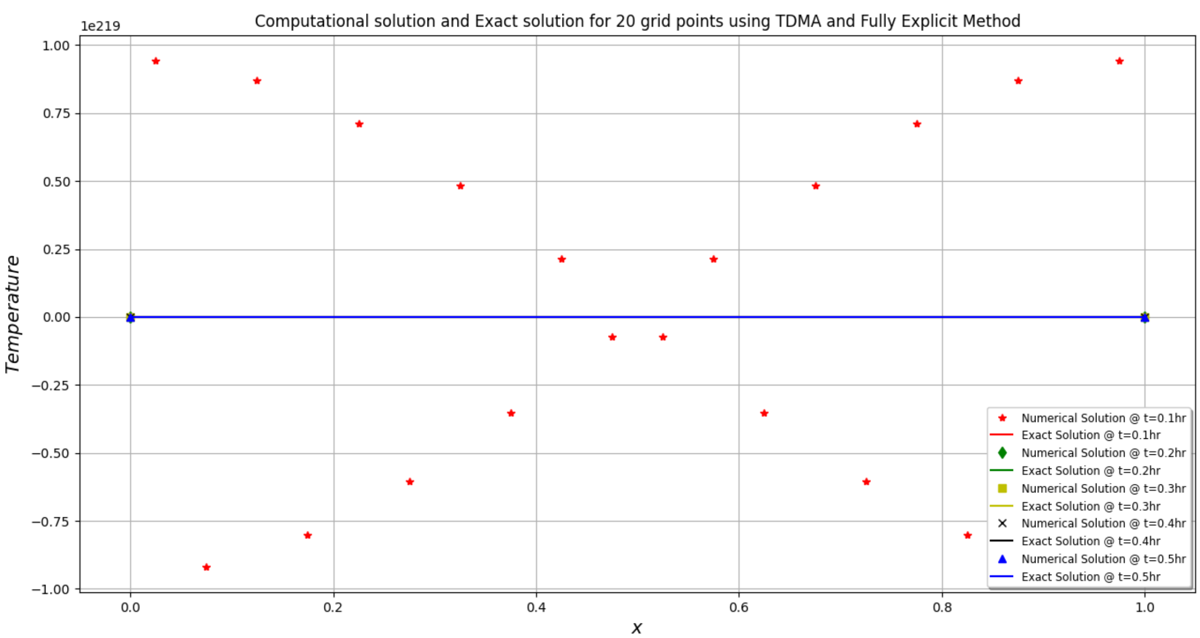
$$\frac{(\delta x)^2}{2\alpha\Delta t} > 1$$

Whenever the above relation between the temporal and spatial steps is violated, the solution diverges. In the above equation, plugging in the values of $\alpha = 1.0, \Delta t = 0.001$ and $\delta x = 0.05$ it gives us that $1.25 > 1$, which is true and thus the scheme converges.

Now suppose we purposely violate the condition by keeping $\alpha = 1.0$ and $\delta x = 0.05$ but putting $\Delta t = 0.01$ this gives us that $0.125 > 1$ which is not true and thus the scheme diverges and gives non-physical or non-realistic results as shown in the diagram below:



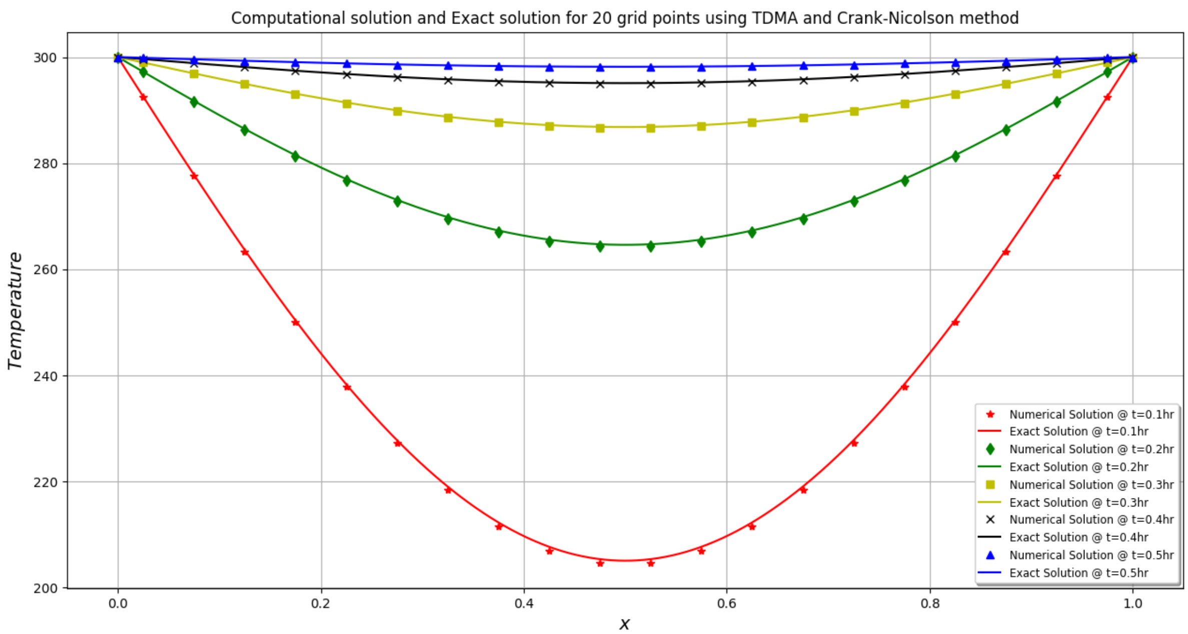
Again, keeping $\alpha = 1.0$ and $\delta x = 0.05$ but putting $\Delta t = 0.1$, this gives us that $0.0125 > 1$ which is again not true and thus the scheme diverges and gives non-physical or non-realistic results as shown in the diagram below:



Analysis of results: From the above plots and the equation for the stability criterion, we can deduce a trade-off between the temporal and spatial steps. The stability of the fully explicit scheme depends on the properties of the problem and the size of the temporal and spatial steps. Both the quantities are inversely proportional. If we increase the resolution (or consequently decrease the step size) of one, we have to decrease the resolution (or consequently increase the step size) of the other. It would take large number of time steps to compute the value of the solution at even a moderate time, e.g., t = 1second. Also, due to the machine limits, the propagation of round-off errors might then cause a significant reduction in the overall accuracy of the final solution values. Since not all choices of space and time steps lead to a convergent scheme, the explicit scheme is called conditionally stable.

b) Crank-Nicolson scheme

temporal step = 0.001 and spatial step = 0.05

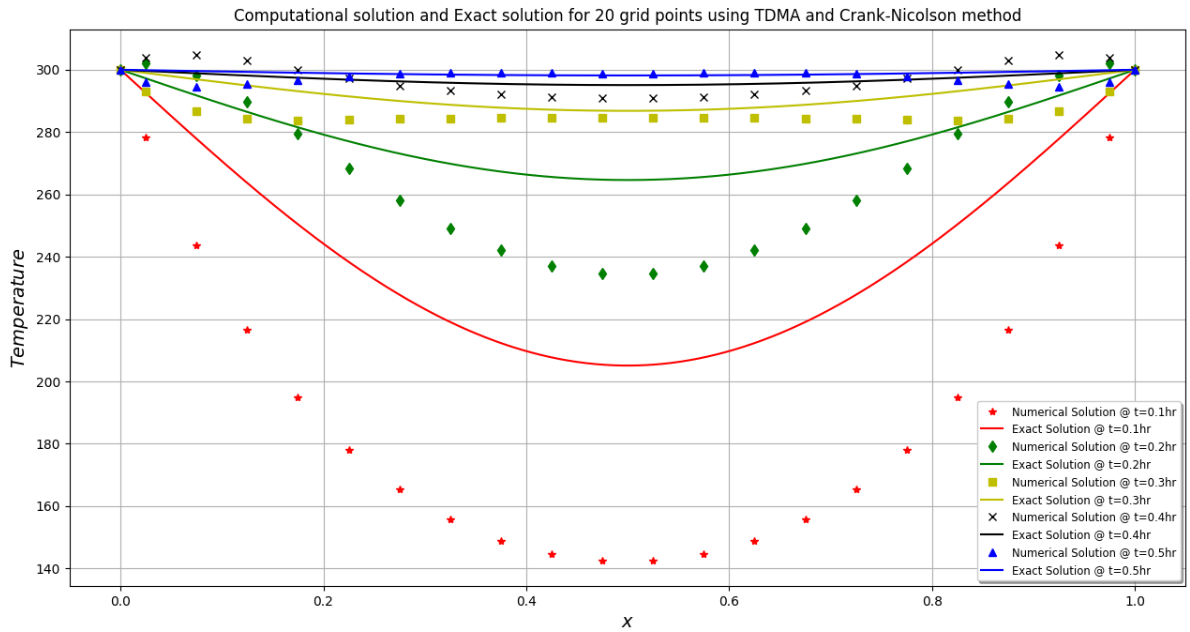


The stability criterion for fully explicit method is:

$$\frac{(\delta x)^2}{\alpha \Delta t} > 1$$

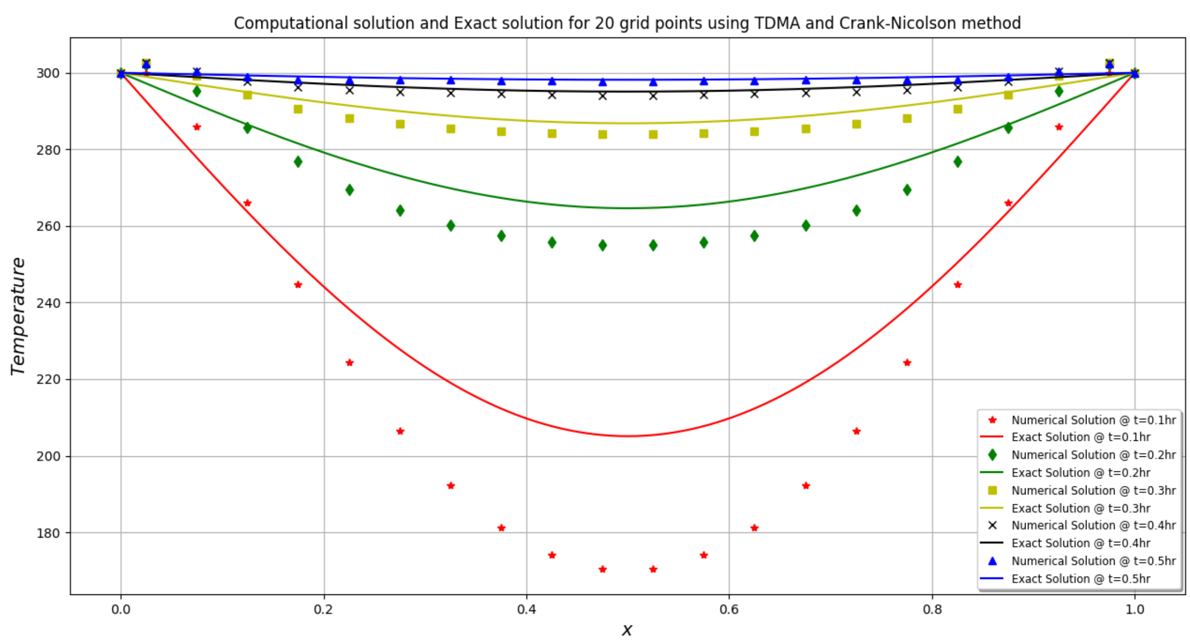
Whenever the above relation between the temporal and spatial steps is violated, the solution diverges. In the above equation, plugging in the values of $\alpha = 1.0, \Delta t = 0.001$ and $\delta x = 0.05$ it gives us that $2.5 > 1$, which is true and thus the scheme converges.

Now we keep $\alpha = 1.0$ and $\delta x = 0.05$ and put $\Delta t = 0.1$ this gives us that $0.025 > 1$ which is not true and thus the scheme diverges and gives non-physical or non-realistic results as shown below:



In the above plot we can observe that the temperature at few locations is exceeding 300ºC which is non-physical.

Again, keeping $\alpha = 1.0$ and $\delta x = 0.05$ but putting $\Delta t = 0.05$, this gives us that $0.025 > 1$ which is again not true and thus the scheme diverges and gives non-physical or non-realistic results as shown in the diagram below:



In the above plot we can observe that the temperature at few locations is exceeding 300ºC which is again, non-realistic.

Analysis of results: From the above plots and the equation for the stability criterion, we can deduce a trade-off between the temporal and spatial steps. The stability of the Crank-Nicolson scheme depends on the properties of the problem and the size of the temporal and spatial steps. Both the quantities are inversely proportional. If we increase the resolution (or consequently decrease the step size) of one, we have to decrease the resolution (or consequently increase the step size) of the other. From the above discussion we can deduce that Crank-Nicolson is also conditionally stable.

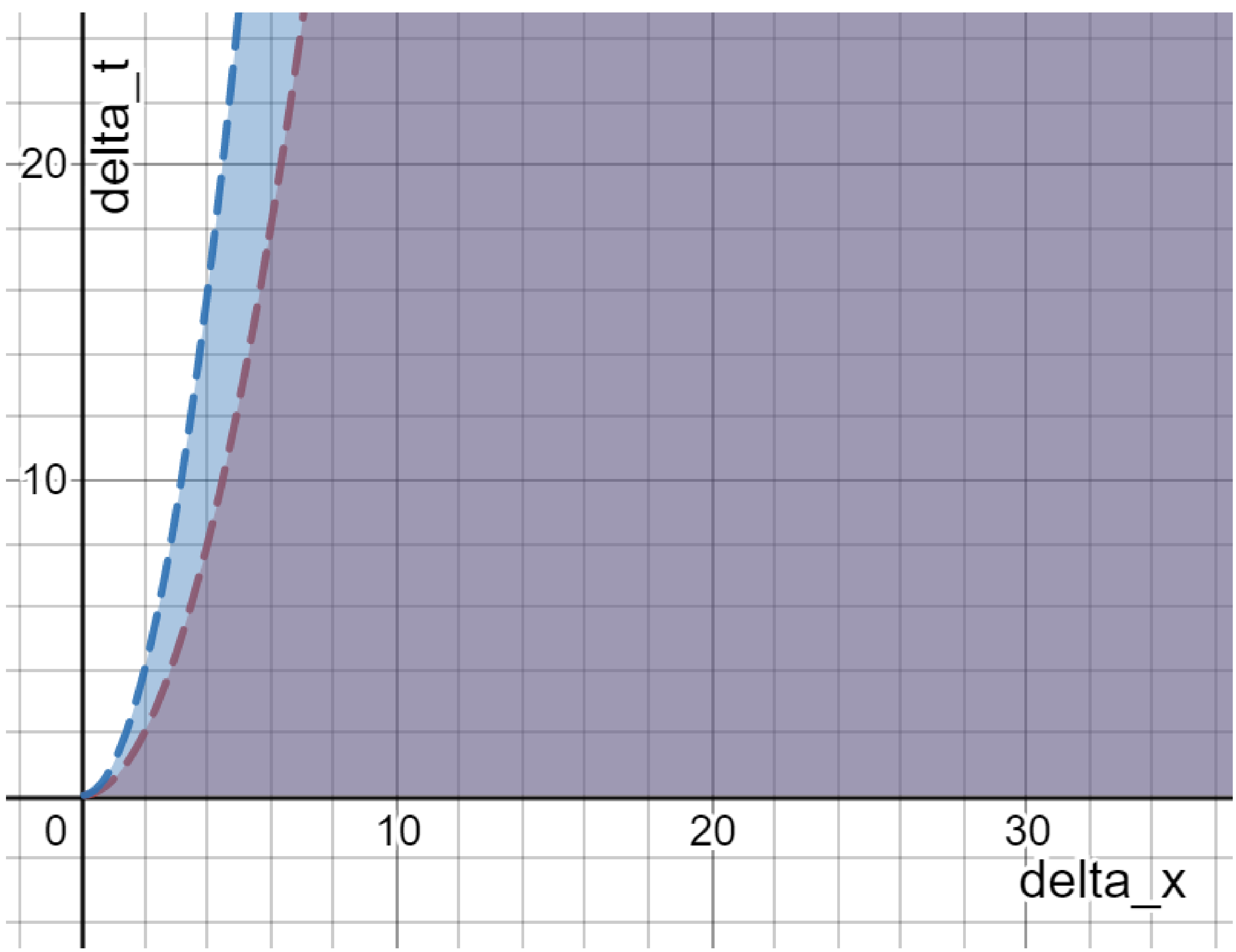
The one thing that is worth noting here is that the stability criterion of Crank-Nicolson is somewhat relaxed than that of Fully explicit scheme.

For Explicit scheme we have: $\frac{(\delta x)^2}{\alpha \Delta t} > 2$; represented by red region

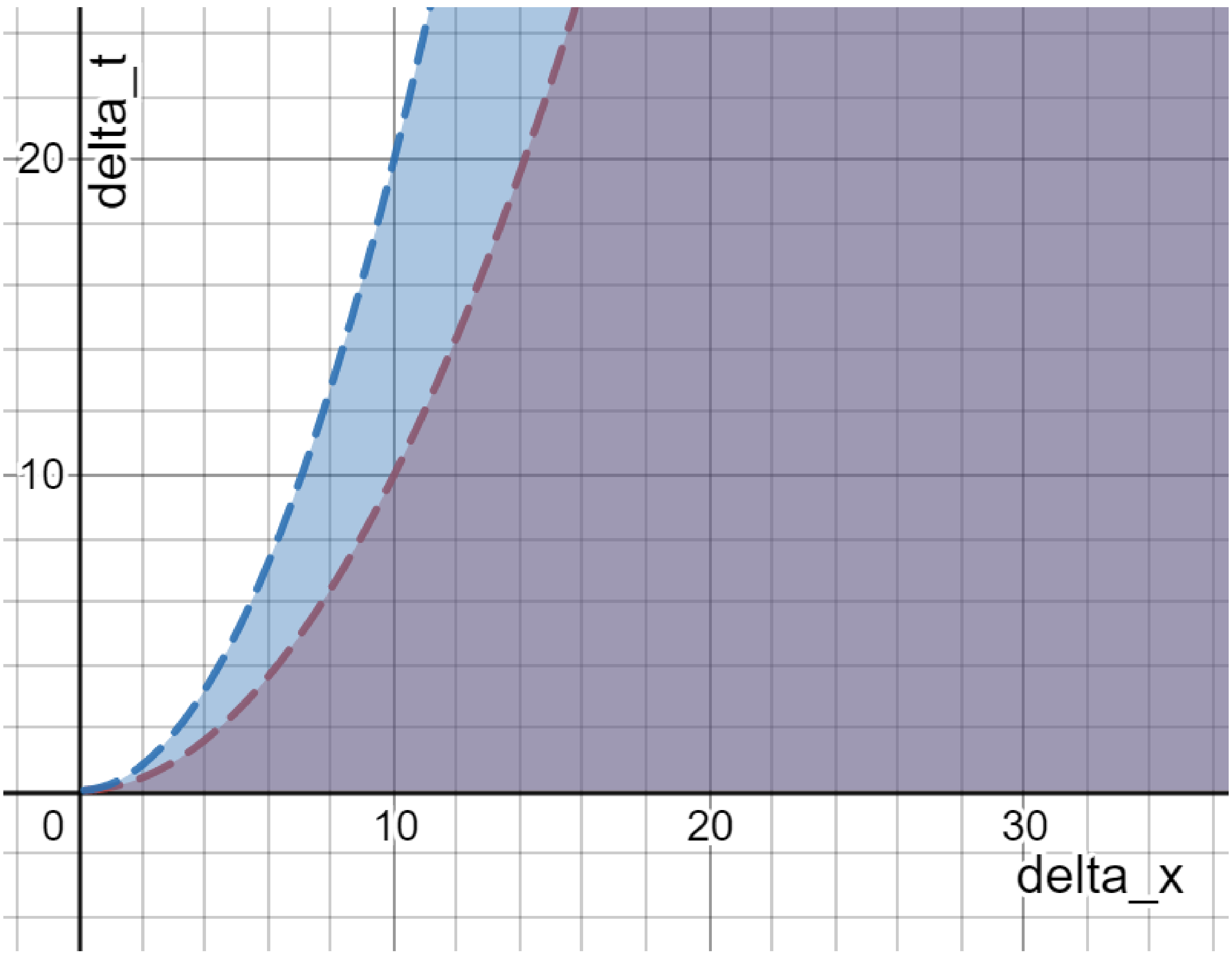
For Crank-Nicolson scheme we have: $\frac{(\delta x)^2}{\alpha \Delta t} > 1$; represented by blue region

If we plot δx on x-axis and Δt on y-axis while keeping α variable, we get the following graph:

i) $\alpha = 1.0$



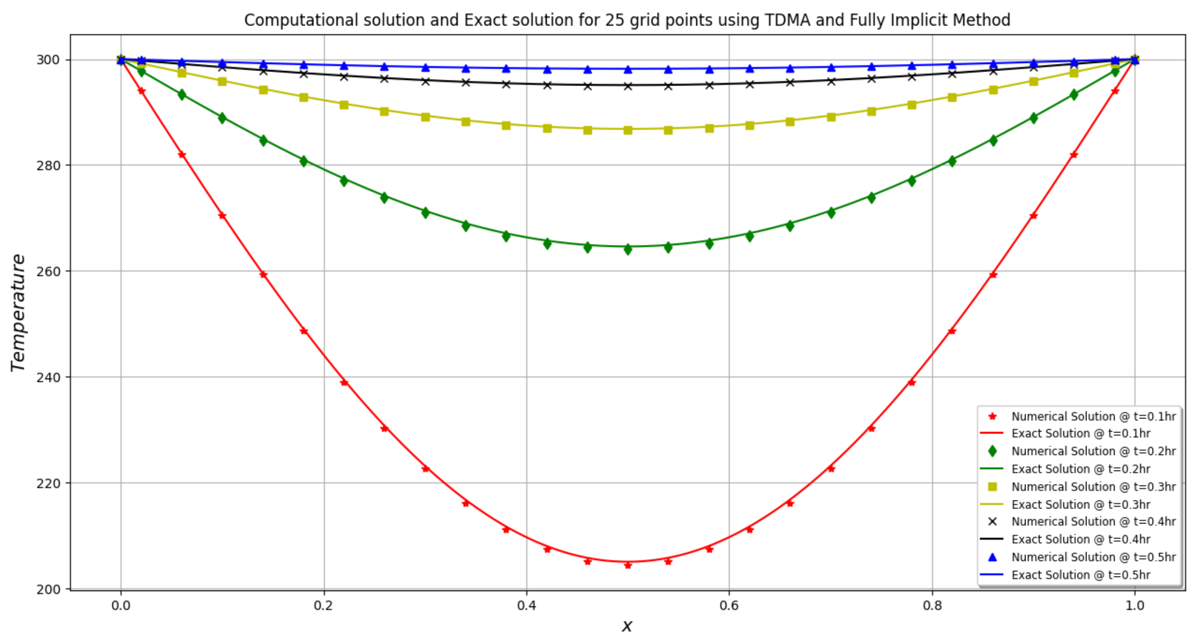
ii) $\alpha = 5.0$



From the above two plots, we can draw a conclusion that the Crank-Nicolson scheme provides a larger range of values that can be used for temporal and spatial steps than the range of values offered by Explicit scheme. This is due to the fact that the limiting value of the expression $\frac{(\delta x)^2}{\alpha \Delta t}$ for Explicit scheme double the limiting value of Crank-Nicolson scheme.

c) Fully Implicit scheme

temporal step = 0.001 and spatial step = 0.05



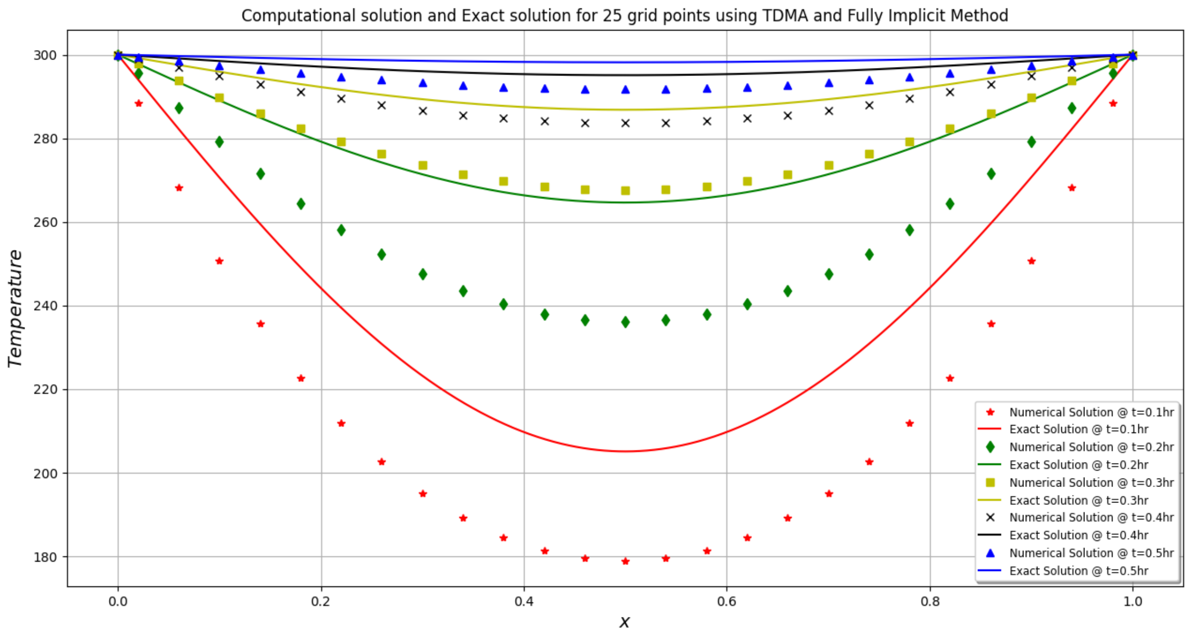
If we perform the Von Neumann stability analysis for fully implicit scheme, we will arrive at the result that the magnification factor, λ is defined as:

$$\boxed{\lambda = \frac{1}{1 + 4\beta sin^2\left({k\delta x}/{2}\right)}}$$

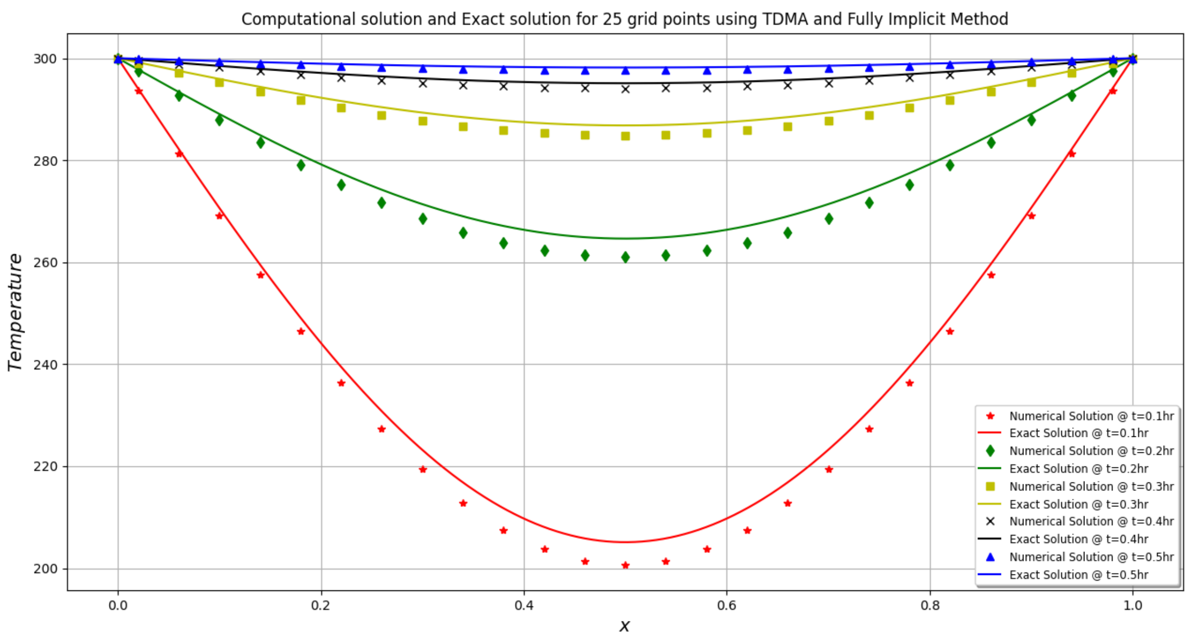
where $\beta = \frac{\alpha \Delta t}{(\delta x)^2}$, which is unconditionally stable, as all quantities are positive thus the magnification factor λ is always less than 1.

Now we plot the results when we have temporal step = 0.1 and temporal step = 0.01, at which the earlier two scheme produced non-physical results.

temporal step = 0.1



temporal step = 0.01



Analysis of results: From the above plots, we can deduce that the Fully Implicit Scheme is unconditionally stable as the plots show that, in addition to temporal steps of t=0.001 and t=0.0001, we see that the solution is not converging to the real solution, but the solutions are not non-physical, the temperature at each node is below the maximum permissible temperature. Thus, implicit scheme is better than the above two said schemes as there is no restriction on temporal and spatial steps.

$$T(x) = T_s + 2(T_i - T_s)\sum_{m=1}^{\infty} e^{-\left[\left(\frac{m\pi}{L}\right)^2 \alpha t\right]} \frac{1-(-1)^m}{m\pi} sin\left(\frac{m\pi x}{L}\right)$$

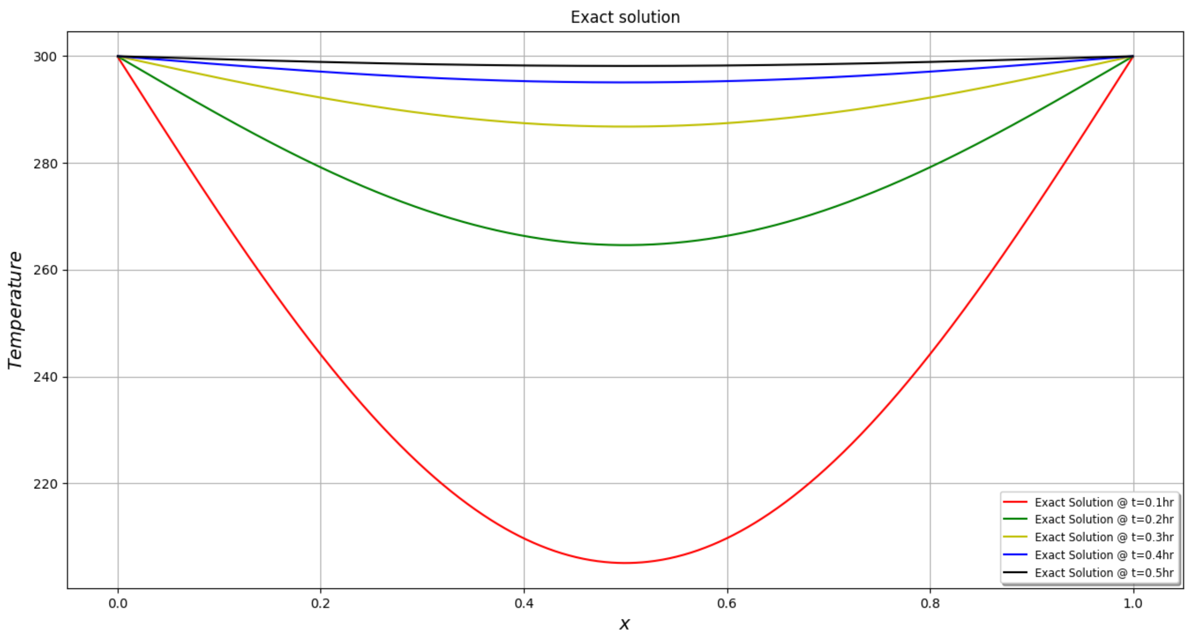
where $T_s = 300^0\text{C}$

$T_i = 100^0\text{C}$

$\alpha = 1.0\,\frac{m^2}{hr}$

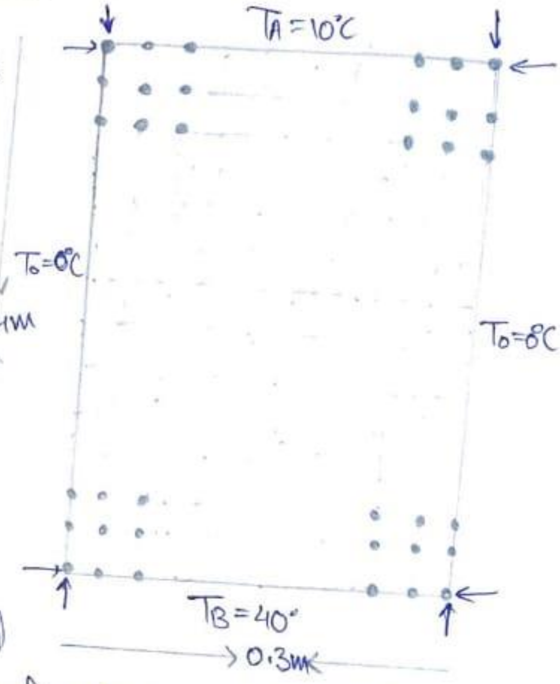
$L = 1\;m$

The plot for the exact solution is:



Ques2 We assume that the temperature at the node prevails and remains constant over the whole control volume, we get a total of 1200 nodes and thus a system of 1200 linear equations in 1200 variables.

i) We assume that the temperature at the outer most nodes, which are marked by arrows, the temp. in these rows + columns is maintained constant. The bottom most temperature is 40°C + the top most row has 10°C at all nodes. Similarly, the left + right most columns have 0°C. Now we employ our finite volume method + create a volume of $\Delta x \times \Delta y \times 1$ units^3 around each node.

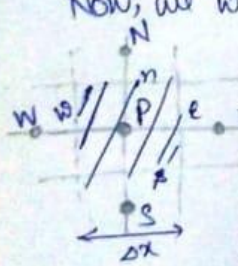


It is to be noted that the CVs for all the nodes will be having the same volume as the boundary nodes already have the applied constant temperature. Thus, we have a system of 39×29 simultaneous linear equations that we have to solve.

ii) The governing equation here is:

$$\rho c \frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right) + \text{Source(which is absent here)}$$

Now, we will discretize the above equation using finite volume method,



$$\int_s^n\int_w^e\int_t^{t+\Delta t} \rho c \frac{\partial T}{\partial t}\, dt\, dx\, dy \times 1 = \int_t^{t+\Delta t}\int_s^n\int_w^e \frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)(dx\, dy \times 1)\, dt + \int_t^{t+\Delta t}\int_w^e\int_s^n \frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)(dy\, dx \times 1)\, dt$$

assuming that $T = T_P^0$ @ $t = t_0$ + $T = T_P$ @ $t = t_0 + \Delta t$

$$\Rightarrow \rho c (T - T_P^0)\Delta x \Delta y = \int_t^{t+\Delta t}\left[\frac{k_e}{\delta x_e}(T_E - T_P) - \frac{k_w}{\delta x_w}(T_P - T_W)\right]\Delta y\, dt + \int_t^{t+\Delta t}\left[\frac{k_n}{\delta y_n}(T_N - T_P) - \frac{k_s}{\delta y_s}(T_P - T_S)\right]\Delta x\, dt$$

now approximating $\int_t^{t+\Delta t} T_P\, dt$ as $[f T_P' + (1-f) T_P^0]\Delta t$ + employing fully implicit approach by putting $f = 1$, we get the final discretized form

$$\Rightarrow T_P\left(\frac{4\alpha\Delta t}{(\delta x)^2}+1\right) = \frac{\alpha\Delta t}{(\delta x)^2}\left(T_E + T_W + T_N + T_S\right) + T_P^{\circ}$$

let $\frac{\alpha\Delta t}{(\delta x)^2} = \beta$

$\therefore$ The final equation is

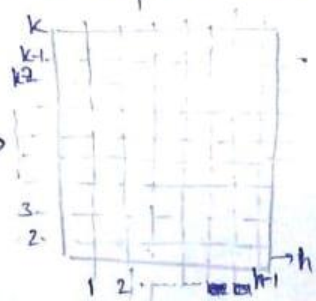
$$T_P(1+4\beta) = \beta T_W + \beta T_E + \beta T_S + \beta T_N + T_P^{\circ}$$

Here we have five unknowns & one known quantity.

iii) We can solve the above stated system of equations using two methods:

a) Line by Line TDMA method

Our system is a five-diagonal system, and what line by line TDMA does is it reduces the five-diagonal system to a three-diagonal system by treating two of the ~~known~~ unknown variables as known. It does so by employing TDMA in one direction only and the sweeping across the other direction. We employ the TDMA (aka Traverse Direction) along north-south lines & sweep across east-west lines. We treat the east & west temperatures as known.



for line 1: $u_W + u_E = 0$ for $n=0$

Node 1 : $-T_{21}^{n+1}\beta + T_{11}^{n+1}(1+4\beta) - T_B^{n+1}\beta = U_W^{n+1}\beta + T_{11}^{n} + U_E^{n+1}\beta$

Node 2 to k-1 : $-T_{i+1,1}^{n+1}\beta + T_{i,1}^{n+1}(1+4\beta) - T_{i-1,1}^{n+1}\beta = U_W^{n+1}\beta + T_{i,1}^{n} + U_E^{n+1}\beta$

Node k : $-T_A^{n+1}\beta + T_{k,1}^{n+1}(1+4\beta) - T_{k-1,1}^{n+1}\beta = U_W^{n+1}\beta + T_{k,1}^{n} + U_E^{n+1}\beta$

for line j: $u_W = T_i$ from line 1 & $u_E = 0$

Node 1: $-T_{i+1,j}^{n+1}\beta + T_{i,j}^{n+1}(1+4\beta) - T_B^{n+1}\beta = U_W^{n+1}\beta + T_{i,j}^{n} + U_E^{n+1}\beta$

Node 2 to k-1: $-T_{i+1,j}^{n+1}\beta + T_{i,j}^{n+1}(1+4\beta) - T_{i-1,j}^{n+1}\beta = U_W^{n+1}\beta + T_{ij}^{n} + U_E^{n+1}\beta$

Node k: $-T_A^{n+1}\beta + T_{i,j}^{n+1}(1+4\beta) - T_{i-1,j}^{n+1}\beta = U_W^{n+1}\beta + T_{i,j}^{n} + U_E^{n+1}\beta$

for line n: $U_W = T_i$ from line n-1 & $u_E = 0$

Node 1 : $-T_{i+1,h}^{n+1}\beta + T_{i,h}^{n+1}(1+4\beta) - T_B^{n+1}\beta = U_W^{n+1}\beta + T_{i,h}^{n} + U_E^{n+1}\beta$

Node 2 to k-1 : $-T_{i+1,h}^{n+1}\beta + T_{i,h}^{n+1}(1+4\beta) - T_{i-1,h}^{n+1}\beta = U_W^{n+1}\beta + T_{i,h}^{n} + U_E^{n+1}\beta$

Node k : $-T_A^{n+1}\beta + T_{k,h}^{n+1}(1+4\beta) - T_{k-1,h}^{n+1}\beta = U_W^{n+1}\beta + T_{k,h}^{n} + U_E^{n+1}\beta$

∴ In general, the equations are:

for first line at time step $(n+1)$ + for all nodes from 1 to k in north to south direction; $u_W = 0$ + u_E = ~~[scribble]~~ T for line 2 from ~~[scribble]~~ time step n.

$$-T_{i+1,1}^{n+1}\beta + T_{i,1}^{n+1}(1+4\beta) - T_{i-1,1}^{n+1}\beta = U_W^{n+1}\beta + T_{i,1}^{n} + U_E^{n+1}\beta$$

for intermediate lines, j at time step $(n+1)$ + for all the nodes from 1 to k in north - south direction; $U_W = T$ from last north-south iteration and $U_E = T$ for line $j+1$ from time step n

$$-T_{i+1,j}^{n+1}\beta + T_{i,j}^{n+1}(1+4\beta) - T_{i-1,j}^{n+1}\beta = U_W^{n+1}\beta + T_{i,j}^{n} + U_E^{n+1}\beta$$

for last line at time step $(n+1)$ + for all nodes from 1 to k in north to south direction; $U_W = T$ from last north-south iteration + $U_E = 0$.

$$-T_{i+1,h}^{n+1}\beta + T_{i,h}^{n+1}(1+4\beta) - T_{i-1,h}^{n+1}\beta = U_W^{n+1}\beta + T_{i,h}^{n} + U_E^{n+1}\beta$$

The boundaries conditions corresponds to equations when $j = 1$ + $j = h$. At these conditions, $U_W = 0$ + $U_E = 0$ respectively.

The above system of equations can be represented in the following manner, as a matrix equation $Ax = b$

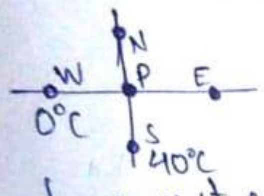
$$\begin{array}{ccc} A & x & b \\ \begin{bmatrix} 1+4\beta & -\beta & 0 & \cdots & & 0 \\ -\beta & 1+4\beta & -\beta & \cdots & & \\ 0 & -\beta & \ddots & \ddots & & \\ \vdots & & \ddots & \ddots & -\beta & 0 \\ 0 & \cdots & & -\beta & 1+4\beta & -\beta \\ 0 & 0 & \cdots & 0 & -\beta & 1+4\beta \end{bmatrix} & \begin{bmatrix} T_{11}^{n+1} \\ T_{21}^{n+1} \\ T_{31}^{n+1} \\ \vdots \\ T_{k-1,h}^{n+1} \\ T_{k,h}^{n+1} \end{bmatrix} = & \begin{bmatrix} U_W^{n+1}\beta + T_{11}^{n} + U_E^{n+1}\beta + T_B^{n+1}\beta \\ U_W^{n+1}\beta + T_{21}^{n} + U_E^{n+1}\beta \\ \vdots \\ \vdots \\ \vdots \\ U_W^{n+1}\beta + T_{k,h}^{n} + U_E^{n+1}\beta + T_A^{n+1}\beta \end{bmatrix} \end{array}$$

b) Point by Point Gauss-Seidel method

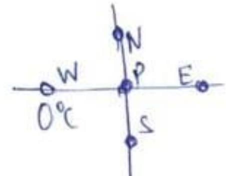
Since our system is a five diagonal system, we can use Gauss-Seidel at each point as well. The boundary conditions will be same as described earlier. Writing the equations in their general form

for row i + column 1: i=1 to k

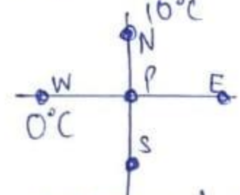
$$-T_{i-1,1}^{n+1}\beta - T_{i+1,1}^{n+1}\beta + T_{i,1}^{n+1}(1+4\beta) - T_{i,0}^{n+1}\beta - T_{i,2}^{n+1}\beta = T_{i,1}^{n}$$



Lower most point



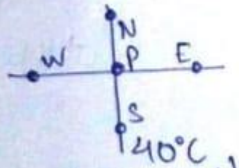
intermediate point



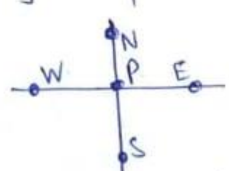
upper most point

for row i + column j: i=1 to k + j=2 to h-1

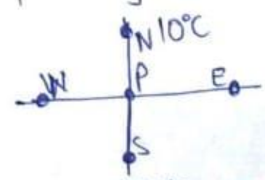
$$-T_{i-1,j}^{n+1}\beta - T_{i+1,j}^{n+1}\beta + T_{i,j}^{n+1}(1+4\beta) - T_{i,j-1}^{n+1}\beta - T_{i,j+1}^{n+1} = T_{i,j}^{n}$$



Lower most



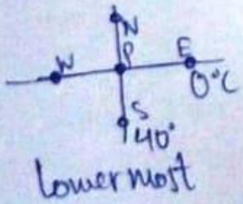
intermediate



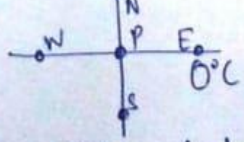
upper

for row i + column h: i=1 to k

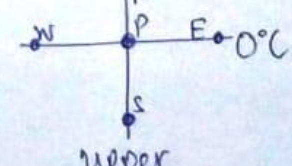
$$-T_{i-1,h}^{n+1}\beta - T_{i+1,h}^{n+1}\beta + T_{i,h}^{n+1}(1+4\beta) - T_{i,h-1}^{n+1}\beta - T_{i,h+1}^{n+1}\beta = T_{i,h}^{n}$$



Lower most



intermediate



upper

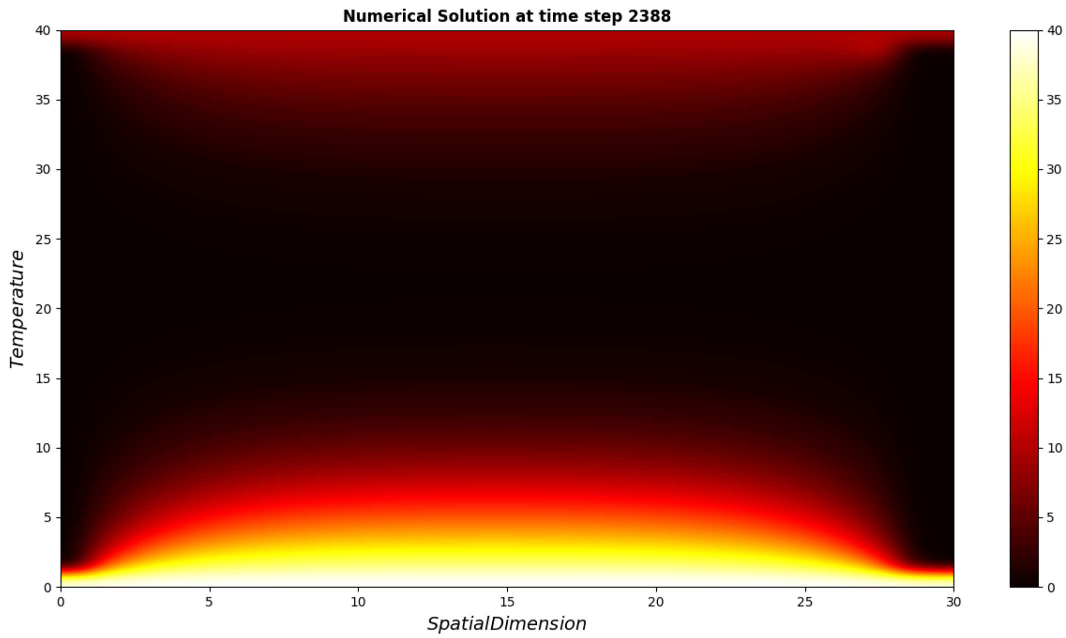
The above system of equations can be represented as a matrix equation $Ax = b$

$$
\overset{A}{\begin{bmatrix}
1+4\beta & -\beta & 0 & 0 & -\beta & 0 & \cdots & \cdots & 0 \\
-\beta & 1+4\beta & -\beta & 0 & 0 & \cdots & \cdots & \cdots & \vdots \\
0 & -\beta & \ddots & \ddots & -\beta & \cdots & \cdots & \cdots & \vdots \\
0 & 0 & \ddots & \ddots & 0 & -\beta & & & \vdots \\
-\beta & 0 & \ddots & \ddots & 0 & 0 & & & \vdots \\
0 & -\beta & \ddots & \ddots & -\beta & 0 & & & \vdots \\
\vdots & \vdots & \ddots & -\beta & 1+4\beta & 0 & \ddots & & \vdots \\
\vdots & \vdots & & \ddots & 0 & 1+4\beta & -\beta & \ddots & \vdots \\
\vdots & \vdots & & & 0 & -\beta & \ddots & \ddots & 0 \\
\vdots & \vdots & & & 0 & \ddots & \ddots & \ddots & -\beta \\
\vdots & \vdots & & & -\beta & \ddots & \ddots & \ddots & 0 \\
\vdots & \cdots & \cdots & \cdots & & \ddots & \ddots & -\beta & 0 \\
0 & \cdots & \cdots & \cdots & 0 & -\beta & 0 \quad 0 & -\beta & 1+4\beta
\end{bmatrix}}
\overset{x}{\begin{bmatrix}
T_{11}^{n+1} \\ T_{21}^{n+1} \\ T_{31}^{n+1} \\ \vdots \\ \vdots \\ T_{kh}^{n+1}
\end{bmatrix}}
=
\overset{b}{\begin{bmatrix}
40\beta + T_{11}^{n} \\ T_{21}^{n} \\ \vdots \\ 10\beta + T_{k,1}^{n} \\ \vdots \\ T_{kh}^{n} + 40\beta
\end{bmatrix}}
$$

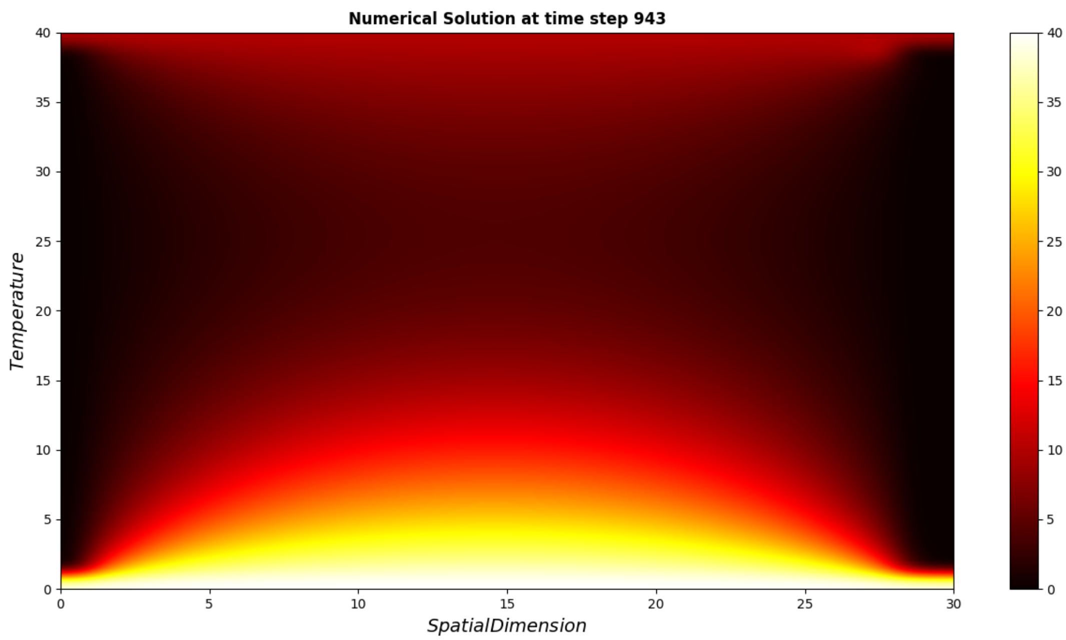
Question2. Required output (plots/any other means) and Analysis of results

a) Since the scheme is unconditionally stable, all the plots converge to the final solution. There are no convergence issues.

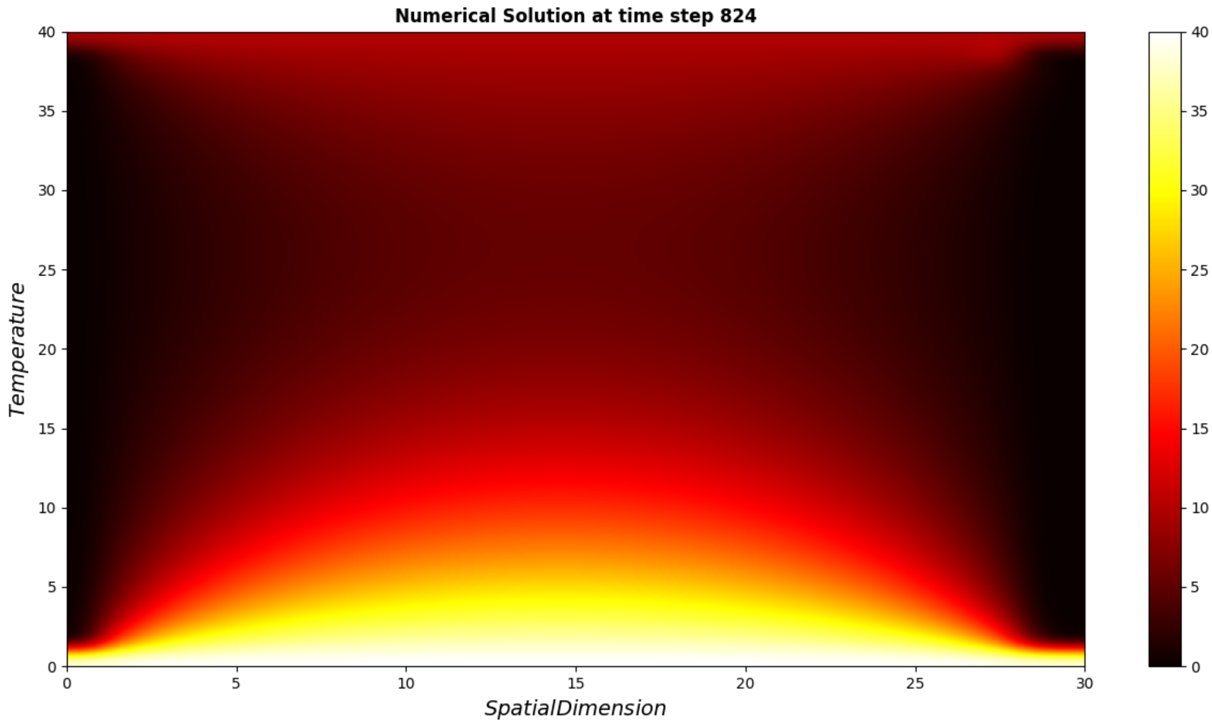
i) t = 0.008755hr



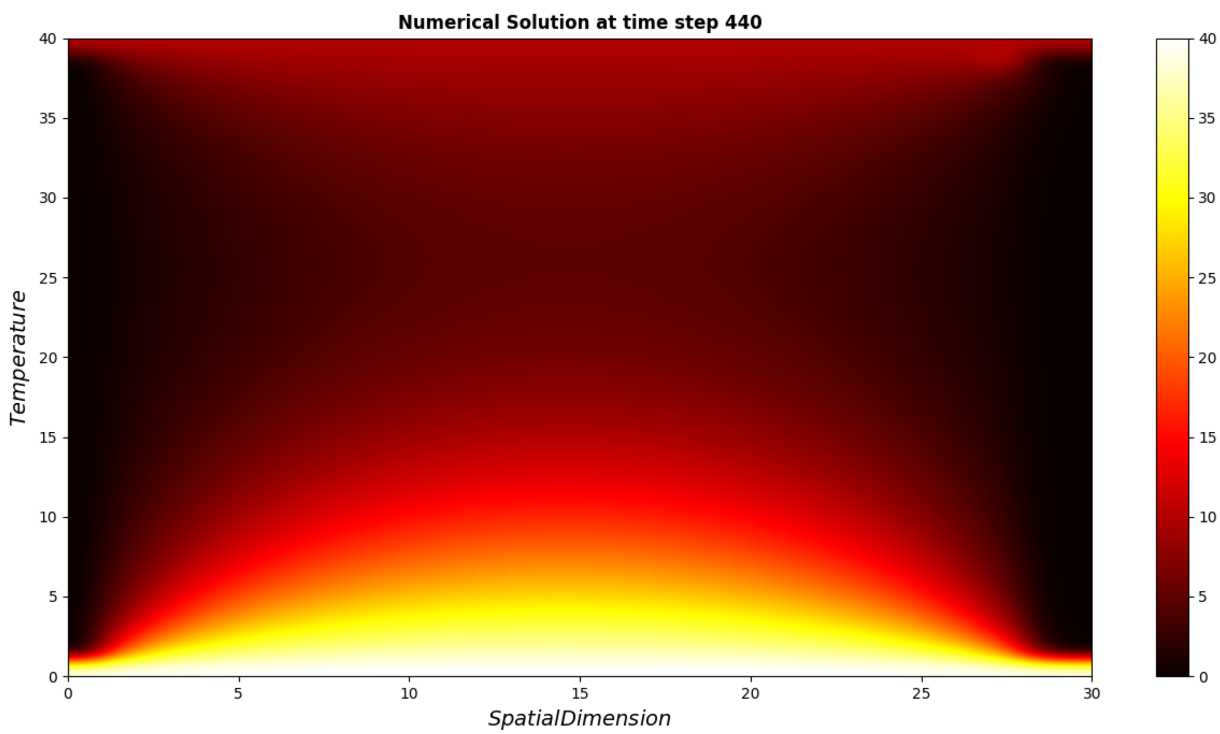
ii) t = 0.12397hr



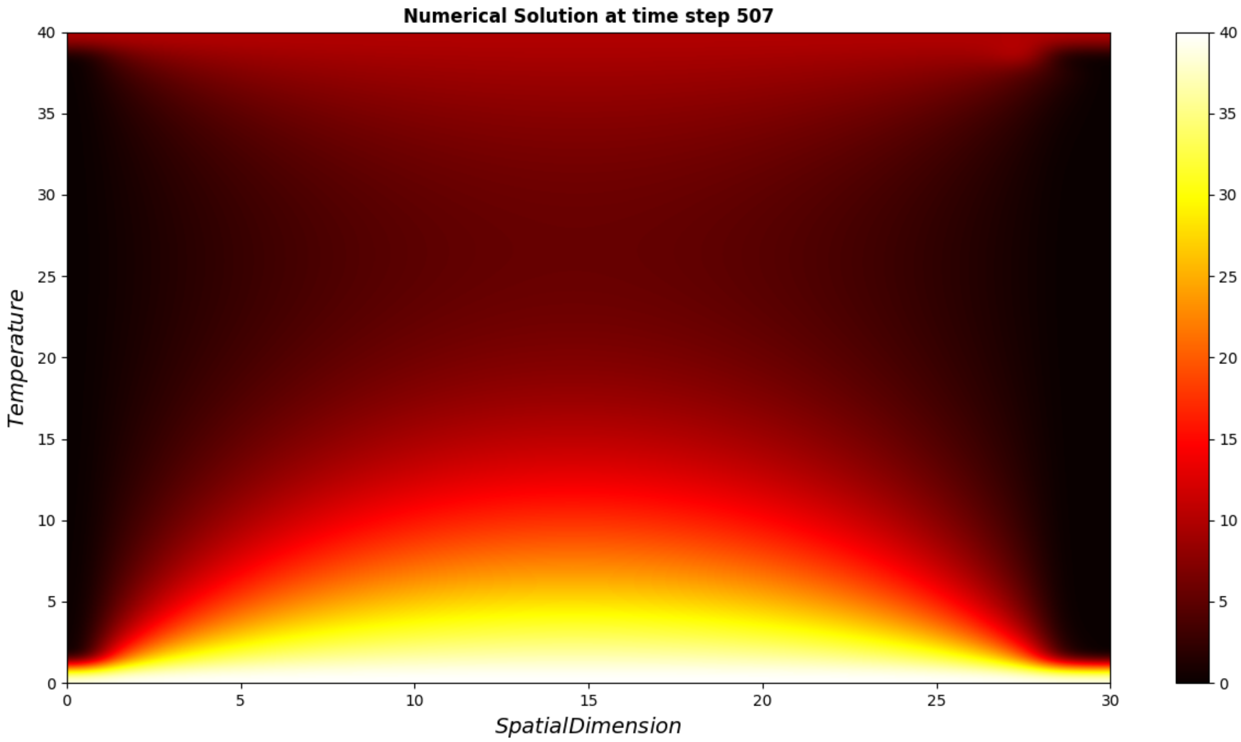
iii) t = 0.334455hr



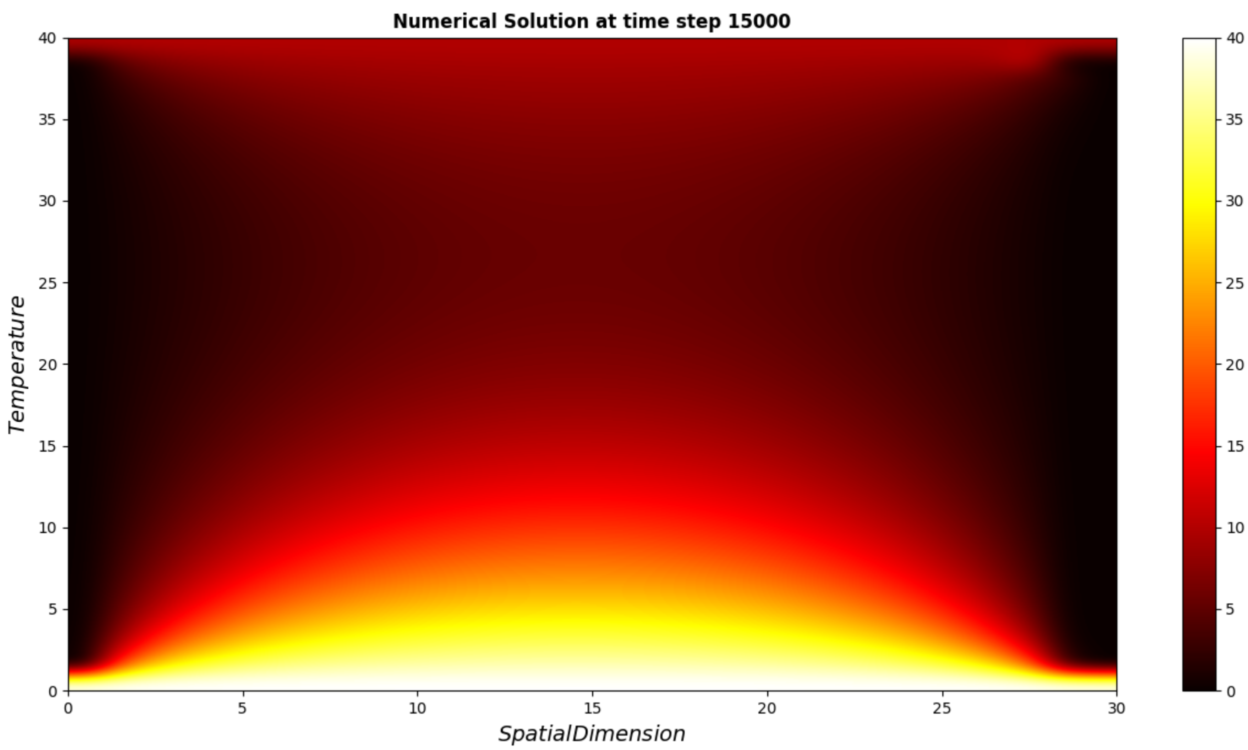
iv) t = 0.5hr



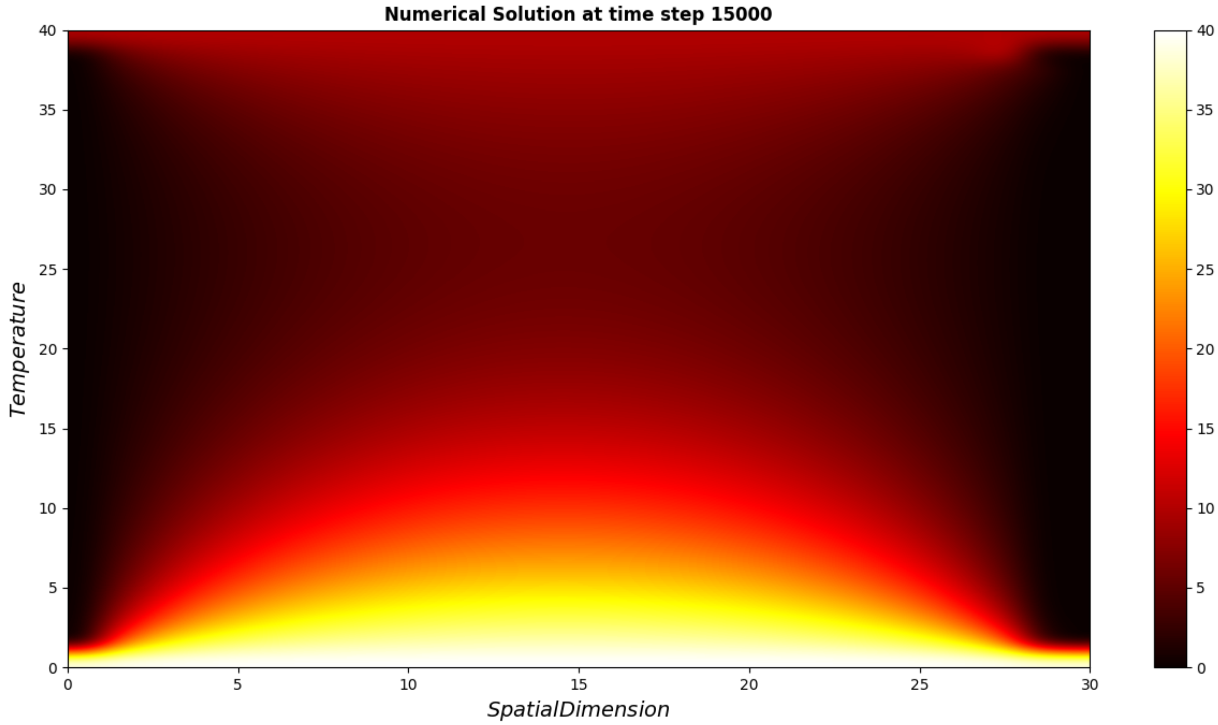
v) t = 1hr


Numerical Solution at time step 507
Temperature
SpatialDimension
0
5
10
15
20
25
30
35
40

vi) t = 5


Numerical Solution at time step 15000
Temperature
SpatialDimension
0
5
10
15
20
25
30
35
40

vii) t = 10

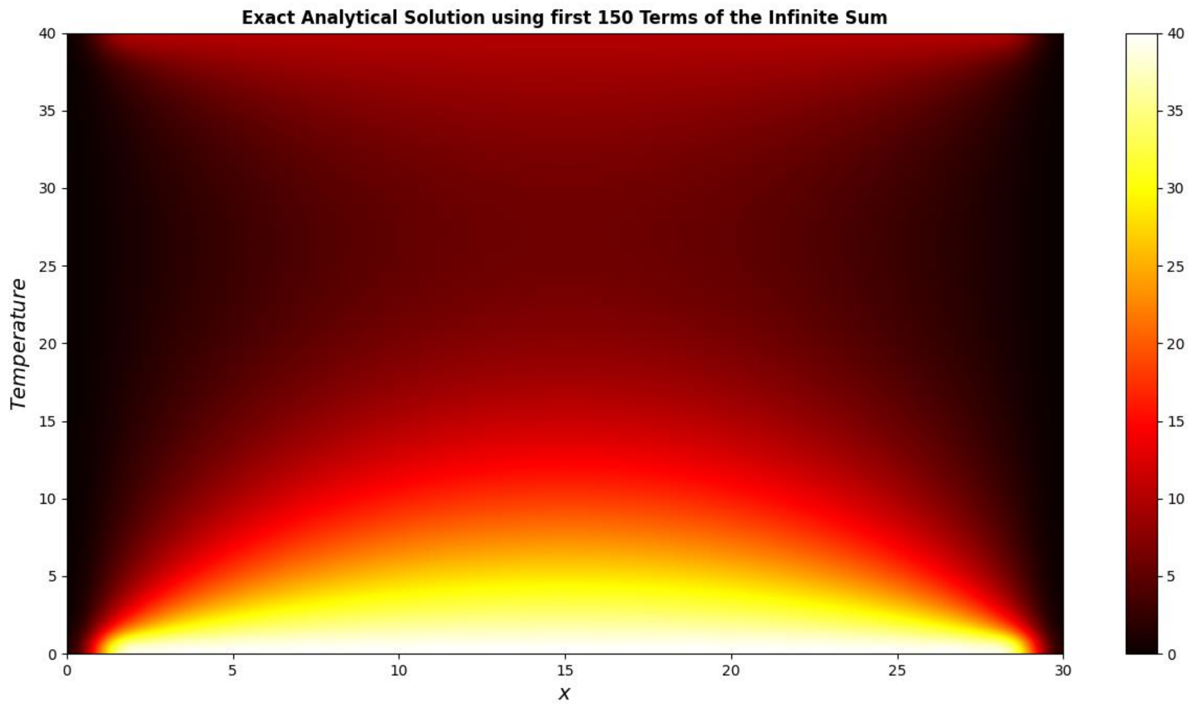


b) The system of linear algebraic equations takes the form of $\boldsymbol{A\overline{y}} = \ \boldsymbol{\overline{b}}$ in both cases, $\bar{\mathrm{b}}$ or RHS terms are not known. Therefore, $\boldsymbol{\overline{y}}$ *or* $\boldsymbol{\overline{T}}$ cannot be solved by inverting the 'matrix' formed on LHS. Therefore, there is no way one can march in $\boldsymbol{x}$ *or* $\boldsymbol{y}$ direction solving unknown variables, because marching in either direction (i *or* j) creates unknown variables in the other direction (j *or* i). We have to solve the 'entire space' at one time (without marching). We have two options to tackle this problem, we can either traverse in y-direction and sweep along x-direction or we can traverse in x-direction and sweep along y-direction. It has been observed that out of the two options, we should traverse in the direction where the expected solution is lesser stiff. Since in our case, the temperatures along the vertical walls are zero, the final solution values are less stiff as compared to the horizontal walls, therefore we should traverse in y-direction and sweep in x-direction as it provides better and faster convergence as compared to the other option.

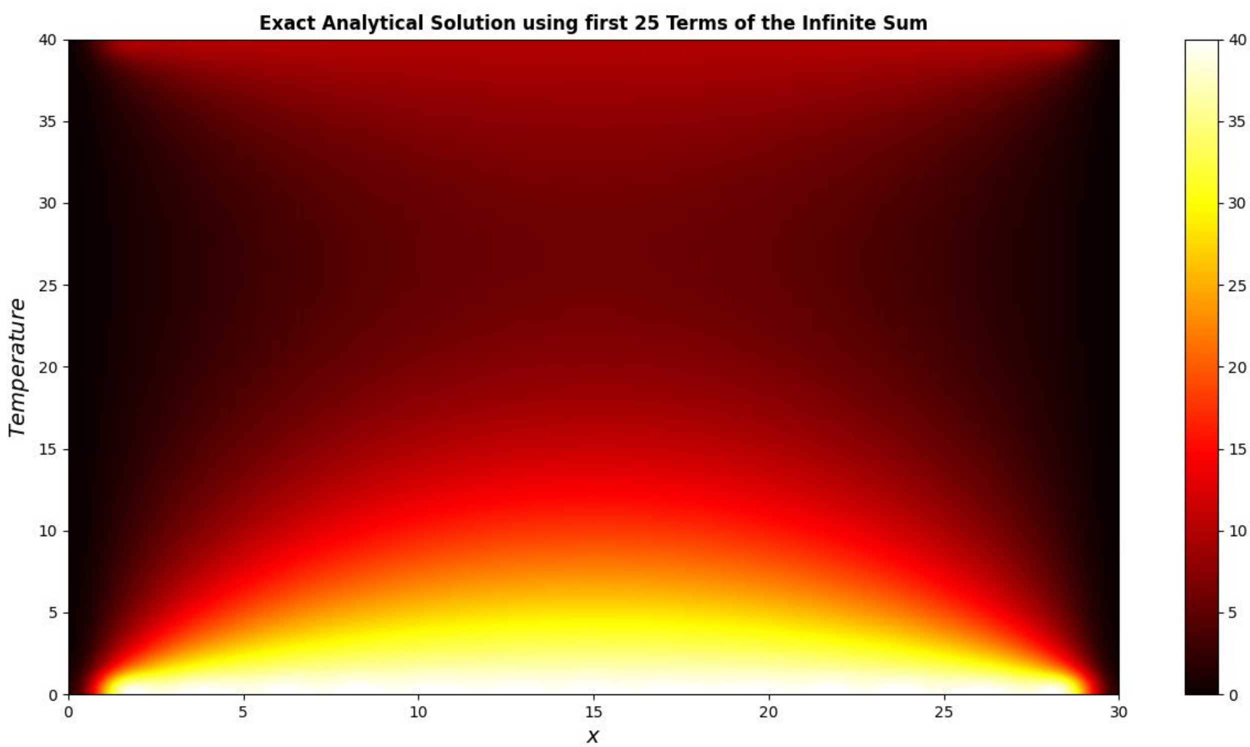
The final analytical solution of the problem is given as:

$$\boxed{T(x,y) = \frac{2}{\pi}\sum_{m=1}^{\infty}\frac{1-(-1)^m}{m\pi}\,\frac{\sin\left(\frac{m\pi x}{L}\right)}{\sinh\left(\frac{m\pi H}{L}\right)}\left\{T_{(y=0)}\sinh\left(\frac{m\pi(H-y)}{L}\right)+T_{(y=H)}\sinh\left(\frac{m\pi y}{L}\right)\right\}}$$

When the above equation is plotted, we get the following temperature distribution:



Here, we can get a very reasonably accurate plot by using the first 25 terms of the infinite series, but just for the sake of accuracy, we used the first 150 terms of the infinite series. The plot using 25 terms is not very different from the other, as can be seen in the following plot:



For the determination of the steady state, we can compare the values of the exact solution and the numerical solution at each grid point, and iterate until the difference between the two values is less than some desired quantity.

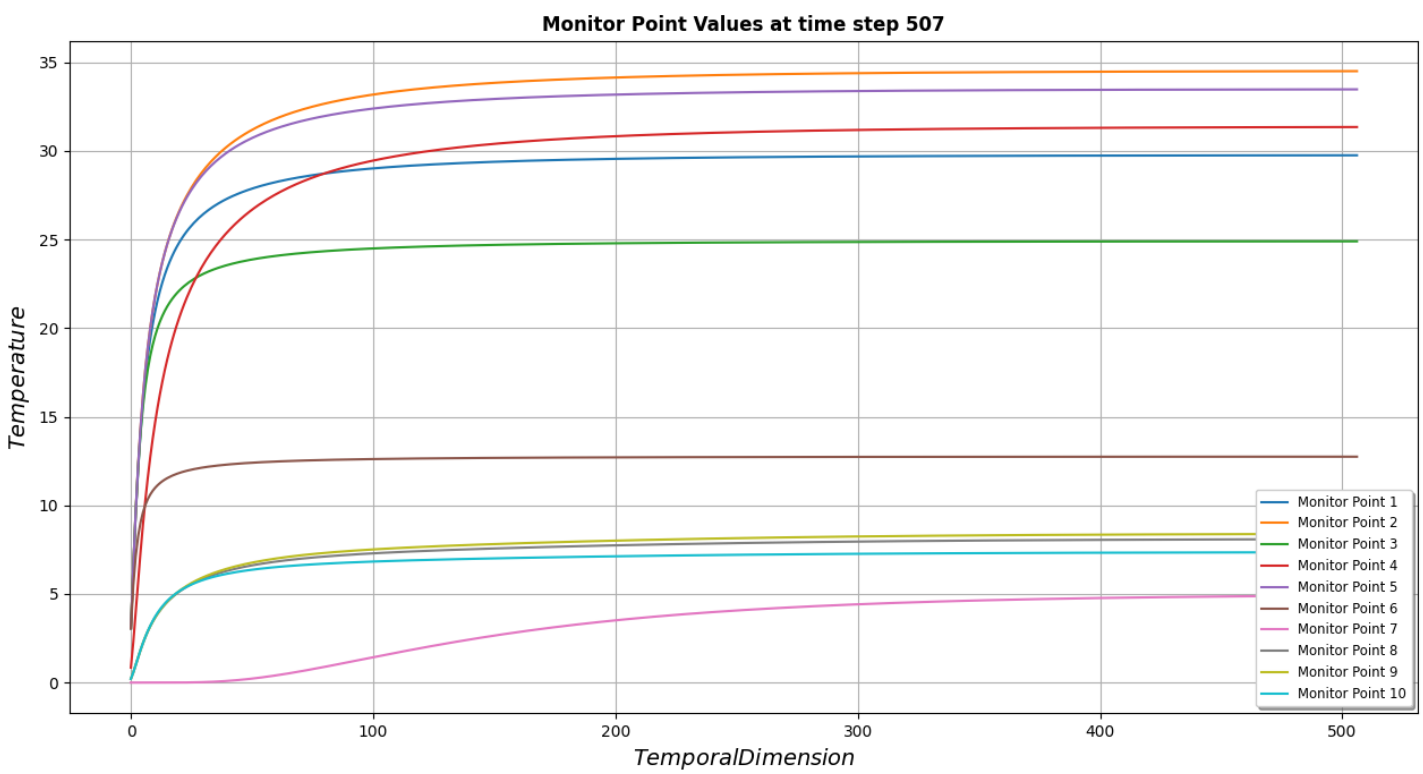
```
# Checking if the solution is converged by employing L(infinity) norm at each grid point
'''In case iteration count exceeds 15000, stopping as it provides a reasonably close answer'''
for p in range(0, len(num_sol_new)):
    if abs(num_sol_new[p] - num_check[p]) < error or iter_count == 15000:
        converged = True
print(iter_count)
```

Here the iteration continues until any of the two conditions is met:

i) The difference between the analytical and numerical values is less than the desired value which is stored in variable “error”.

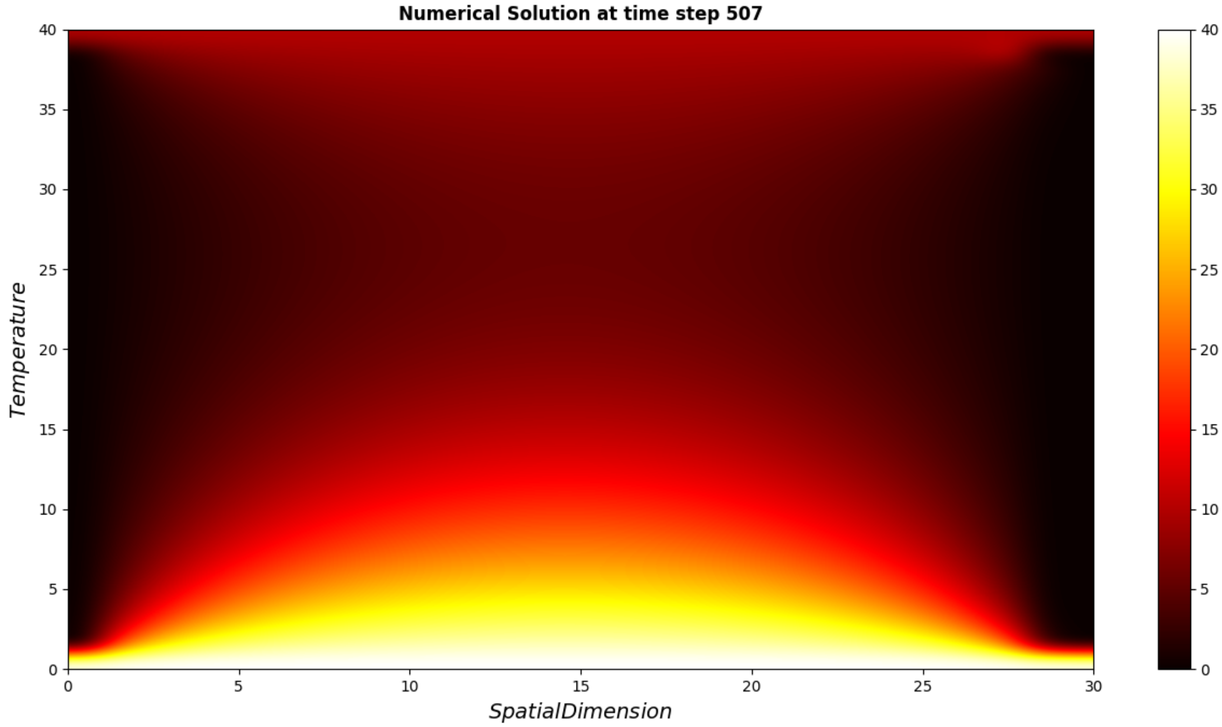
ii) The number of iterations has reached 15000, and by experimenting we have seen that the solution at such large number of iterations is reasonably close to the exact answer.

For visualizing the above criterion, we can see the following plot that shows the time history of the temperatures at a few monitor points in the domain:



Here we can observe that the values of temperature at monitor points increase non-linearly and attains as constant value.

And the corresponding final numerical solution looks like the following plot:

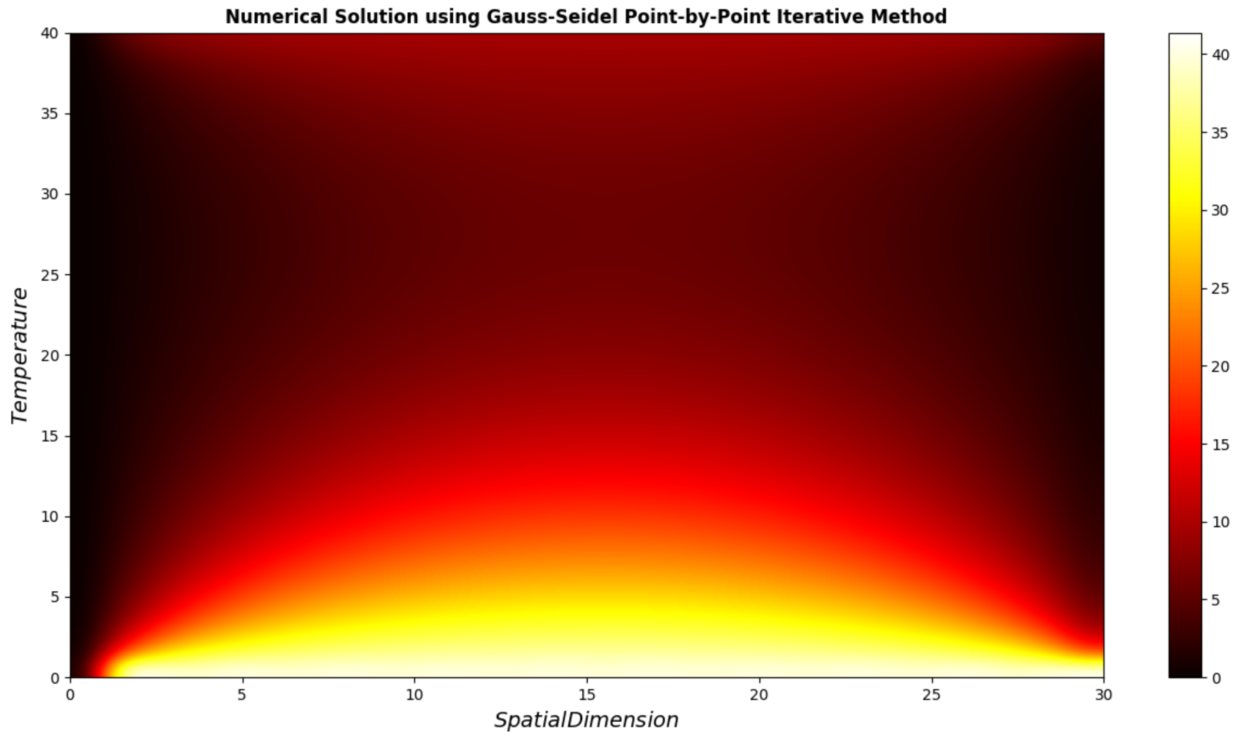


Analysis: We can see that this method is computationally expensive as we might have convergence issues due to the fact that the initial guess can be anything and depending on the values of the initial guess, the solution propagates and gradually reaches the steady state solution. Thus, the convergence heavily depends on the initial guess.

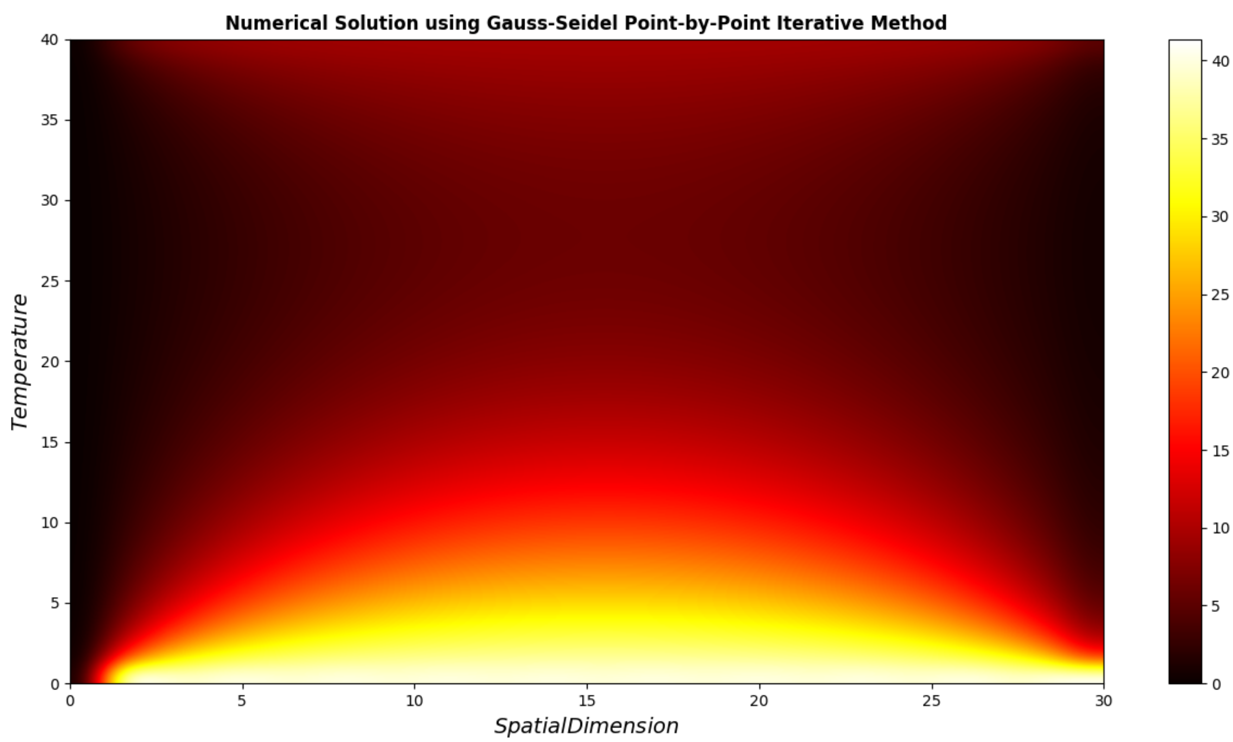
We can employ one technique similar to Guess-Seidel, where, when we are traversing along the north-south direction, instead of using the $\bar{\boldsymbol{b}}$ values from the previous time north-south iteration, we can update the $\bar{\boldsymbol{b}}$ values on the go while calculating the current temperature values. Employing this method may increase the convergence speed, as the solution reaches the final values faster as compared the standard line by line TDMA method.

c) Solving the problem using Point by Point Gauss-Seidel method:

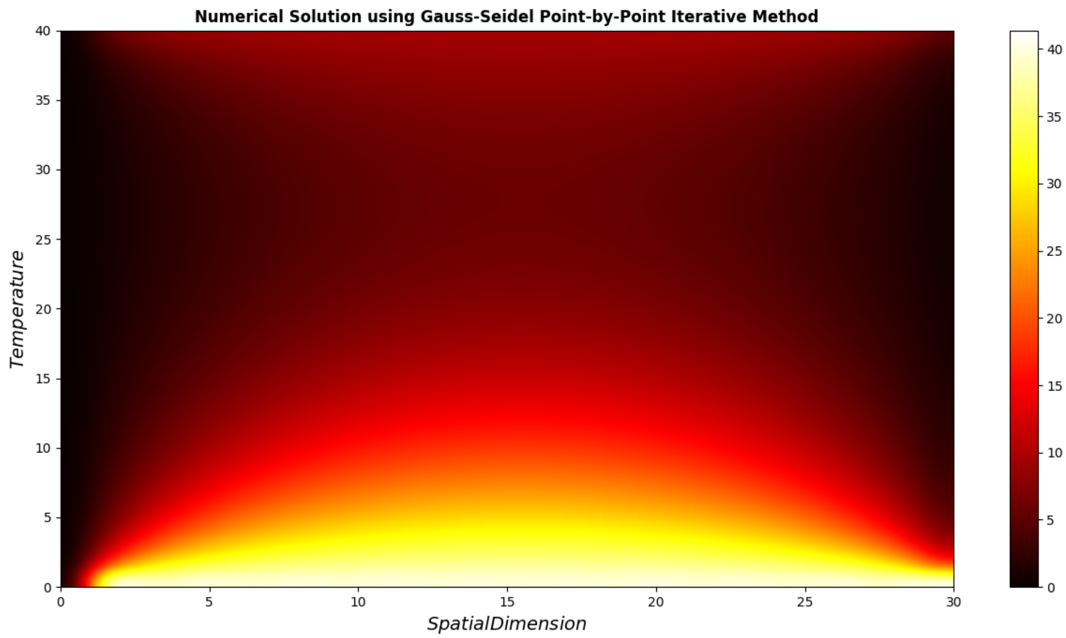
i) Temporal step = 1.0 and initial guess T_i = 10.0, Total number of iterations: 34



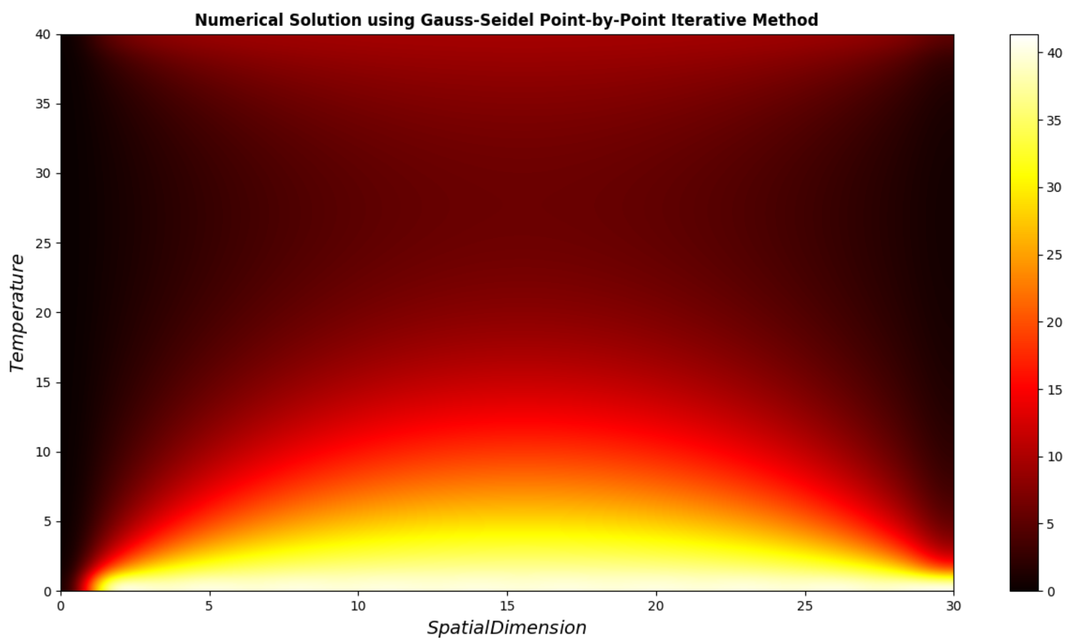
ii) Temporal step = 1.0 and initial guess T_i = 15.0, Total number of iterations: 36



iii) Temporal step = 1.0 and initial guess $T_i = 25.0$, Total number of iterations: 37



iv) Temporal step = 1.0 and initial guess $T_i = 0.0000125$, Total number of iterations: 29



Thus, we can observe that the number of iterations required for the converged solution increases as we increase the initial guess.

Bibliography

- *Finite Difference-based Numerical Methods in Chemical Engineering*. (2016). Nishith Verma. https://www.iitk.ac.in/che/faculty/nv/eLectures-Numerical%20Methods.pdf
- Dept. of Chemical Engineering, Indian Institute of Technology, Bombay. (2015). *Numerical Analysis Module 4 Solving Linear Algebraic Equations*. Sachin C. Patwardhan. https://nptel.ac.in/content/storage2/courses/103101111/downloads/Lecture-notes/Module_4_Solving_Ax=b.pdf
- Prasad, J. S. V. R. K., V. Patil, P., Department of Mathematics, M. J. College, & Department of Applied Science, SSBT's College of Engineering and Technology. (2015, August). *Algorithm for Solving Tri-diagonal Finite Volume Discretized Linear Systems*. https://www.pvamu.edu/aam/wp-content/uploads/sites/49/24_r758_patil_aam_r758_pp_100914_ready_amh.pdf
- *Gauss-Seidel Method - an overview | ScienceDirect Topics*. (2006). Gauss-Seidel Method. https://www.sciencedirect.com/topics/engineering/gauss-seidel-method
- https://numpy.org/doc/stable/reference/generated/numpy.linalg.solve.html#:~:text=solve%20(a%2C%20b)%5B,M%2C%20M)%20array_like
- https://pythonnumericalmethods.berkeley.edu/notebooks/chapter14.04-Solutions-to-Systems-of-Linear-Equations.html
- https://ocw.mit.edu/courses/mathematics/18-336-numerical-methods-for-partial-differential-equations-spring-2009/lecture-notes/MIT18_336S09_lec14.pdf
- https://en.wikipedia.org/wiki/Explicit_and_implicit_methods
- https://stackoverflow.com/questions/6081008/dump-a-numpy-array-into-a-csv-file
- https://stackoverflow.com/questions/41870968/dumping-numpy-array-into-an-excel-file