Finite-Volume Second Order Flux Solution to Steady-State Temperature Distribution of Uniformly Heated Disc

William E. Zhang

December 2024

Motivation and Background

Throughout the discussion of a first-order approximation of temperature gradients for the integral, the approximation was limited to the first nearest-neighbors of the approximation. At maximum, there are up to 3 such cells derived for a triangle whose adjacent cells share faces, say in terms of k-hopes at $k = 1$ triangle units away. However, the original derivation neglected the neighboring cells of these adjacent j-th cells, leading to over-diffusion in problems with low resolution. The end result is an underprediction for temperature at the center of each mesh tested. Typically, diffusivity issues like this are caused when the flux from adjacent j-th cells to further points (i.e., $k = 2, 3, 4...$ triangles hops away) are not considered, leading to an underprediction of flux.

However, by considering the first-order derivative with further neighboring adjacent cells (e.g., $k = 1, 2, 3...$ levels of neighbors), over-diffusion may be corrected with respect to the original cell. This assignment is to use a second-order finite difference approximation for a uniform triangular mesh to improve this accuracy. The goal of this project is to provide public access to a 2nd order finite volume solver alongside exploring the increased numerical accuracy of using such a solver over the 1st order.

Analysis Brief Overview

The following steps were applied to transform the PDE into a FV (Finite-Volume) with Gauss-Divergence Theorem. T_i represents the temperature of the element at the circumcenter, and T_j is the temperature of neighboring elements at the circumcenter, defined by the average of their respective triangular nodes. d_{ij} represents the circumcenter distance from $i \to j$, and l_{ij} represents the elemental face length between circumcenter $i \to j$ in this representation. An illustration of these variables with element $i \to j$ is shown in Figure 2.

$$\lambda \nabla^2 T + Q = 0 \tag{1}$$

$$\int_V \left(\lambda \nabla^2 T\right) dA + \int_V Q dA \tag{2}$$

Now apply Gauss-Divergence: (3)

$$\lambda \int_S \nabla T \hat{n} dS + Q_i A_i = 0 \tag{4}$$

The gradient is: (5)

$$\nabla T = \frac{T_i - T_j}{d_{ij}} \tag{6}$$

Now combining the gradient into the integral term: (7)

$$\lambda \int_S \nabla T \hat{n} dS = \lambda \sum_{j \epsilon i} \frac{T_i - T_j}{d_{ij}} l_{ij} \tag{8}$$

$$\lambda \sum_{j \epsilon i} \frac{T_i - T_j}{d_{ij}} l_{ij} + Q_i A_i = 0 \tag{9}$$

$$\lambda \sum_{j \epsilon i} \frac{T_i}{d_{ij}} l_{ij} = \lambda \sum_{j \epsilon i} \frac{T_j}{d_{ij}} l_{ij} - Q_i A_i \tag{10}$$

$$T_i = \left(\sum_{j \epsilon i} \frac{T_j}{d_{ij}} l_{ij} - \frac{Q_i A_i}{\lambda} \right) / \sum_{j \epsilon i} \frac{l_{ij}}{d_{ij}} \tag{11}$$

This is the formulation used for a first-order approximation of flux for the temperature. However, considering a uniform grid and 2nd order finite forward differencing of the following is proposed for the flux scheme.

The following steps were applied to transform the PDE into a FV (Finite-Volume) with Gauss-Divergence Theorem. T_i represents the temperature of the element at the circumcenter, and T_j is the temperature of neighboring elements at the circumcenter, defined by the average of their respective triangular nodes. d_{ij} represents the circumcenter distance from $i \rightarrow j$, and l_{ij} represents the elemental face length between circumcenter $i \rightarrow j$ in this representation. An illustration of these variables with element $i \rightarrow j$ is shown in Figure 2.

$$T_i = \left(\sum_{j \epsilon \mathcal{N}_k(i)} \frac{T_j}{d_{ij}} l_{ij} - \frac{Q_i A_i}{\lambda} \right) / \sum_{j \epsilon \mathcal{N}_k(i)} \frac{l_{ij}}{d_{ij}} \tag{12}$$

$\mathcal{N}_k(i) =$ Set of neighbors that are k hops away from triangle i

Level 1: Weighting $= 1$ (direct neighbors).
Level 2: Weighting $= -\frac{4}{3}$ (for neighbors of level 1).
Level 3: Weighting $= \frac{1}{3}$ (for neighbors of level 2).

For each outward neighbor, a reduced weighting is applied to each level. To correct for this additional layering of cells, the following weightings are considered. A flux weighting of 1 is applied for each j cell at $k = 1$. For each $k = 2$ neighboring cell (up to 6 additional cells), a weighting of

$-\frac{4}{3}$ is applied to account for flux loss assignments. For the third level, a weighting of $\frac{1}{3}$ is imposed on $k = 3$ neighbors. The weighting is following a 2nd order accurate forward-difference scheme, relative to the j cell neighbor.

The analytical solution is as follows, which will be compared and bench-marked in the numerical results.

$$T_i = \frac{Q}{4\lambda}\left(R - r_i^2\right)$$

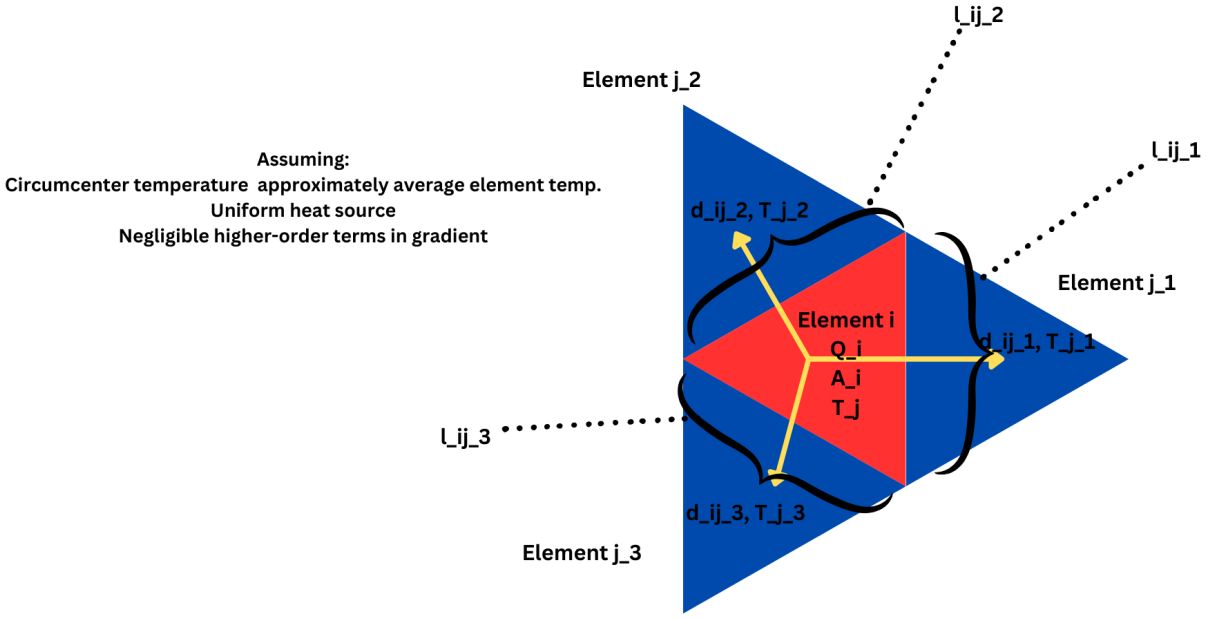


Figure 1: Illustration of FV elements proposed

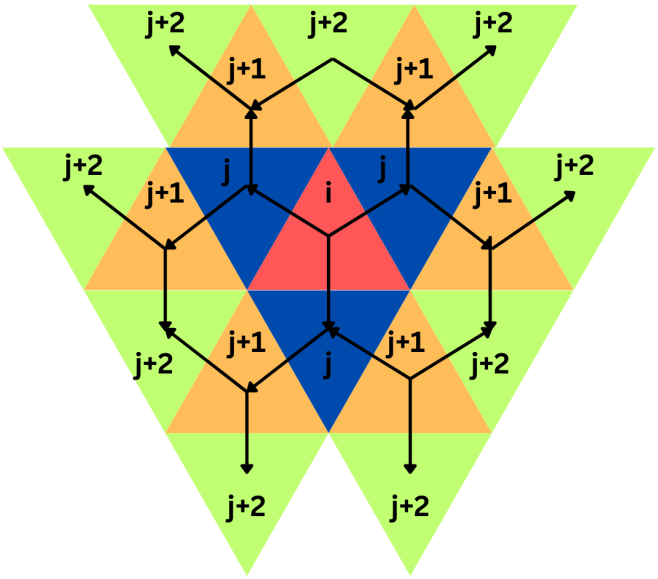


Figure 2: Illustration of FV elements array neighbors, 2nd-order upwind flux correction

Assumptions:

The element temperature can be assumed linear gradients between each circum-center, such that the circumcenter is a linear approximation of average element temperature. Additionally, it is assumed the circumcenter temperature represents the whole element's average temperature such that it is the unweighted average of all nodal temperatures of such element. The source term Q does not vary within the control volume, meaning the heat generation is uniform within each triangular element. Assumes negligible higher order terms for ∇T discretization approximation during the derivation.

Formulation Conditions

The source terms Q is uniformly applied to each element as Q_i, which contributes to the overall heat source of a finite element as $Q_i A_i$.

The triangles on the boundary must satisfy the $T = 0$ wall conditions via the Dirichlet boundary condition. The Dirichlet boundary conditions are implemented as a 0-flux temperature boundary, meaning for every iteration in max_{iters}, there are ghost cell elements outside that are equal and opposite to the adjacent boundary cells to enforce $T = 0$ boundary conditions for all elements adjacent to the boundary. If there are not sufficient triangle neighbors formed within j-set neighbors (i.e. $k \neq 3$ at areas like the edge) then a first order approximation is used for that cell.

The following code piece is implemented per index iteration, to demonstrate such application of the Dirichlet and source term contributions. Note that the Dirichlet condition is applied for every boundary element and is updated after every iteration and balances ghost cells with boundary-condition elements. Note that the heat term Q_i applied is constant throughout, and it is therefore not changing with each iteration.

In this code segment, the temperature of ghost elements at the boundary are identical and negative to the boundary elements. These elements are then summed with adjacent weights to apply Dirichlet conditions within a first-order approximation.

Stopping criteria

The iteration will stop when a tolerance of residual (norm of the difference between two consecutive iterations) is smaller than a tolerance = 10^{-5}. The tolerance is defined as the limit a

residual must be; the residual is defined as the norm between the current and prior iteration's solution. In the case of Gauss-Seidel, for example, it is as follows in the code.

Gauss-Seidel w/ SOR ($\omega = 1.3$)

Akin to Gauss-Seidel, a relaxation factor ω is introduced to relax the solution

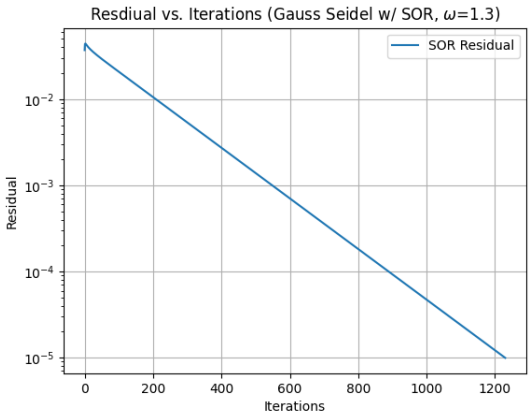
$$T_i^{(k+1)} = (1-\omega)T_i^{(k)} + \frac{\omega}{a_{ii}}\left(b_i - \sum_{j<i} a_{ij}T_j^{(k+1)} - \sum_{j>i} a_{ij}T_j^{(k)}\right), \quad i = 1, 2, \ldots, n. \tag{13}$$

Where b_i refers to the source terms of $Q_i A_i/\lambda$ each element of i, and the product sum of $a_{ij}T_j$ refers to the flux terms $\sum_{j\epsilon i} \frac{T_j}{d_{ij}} l_{ij}$.

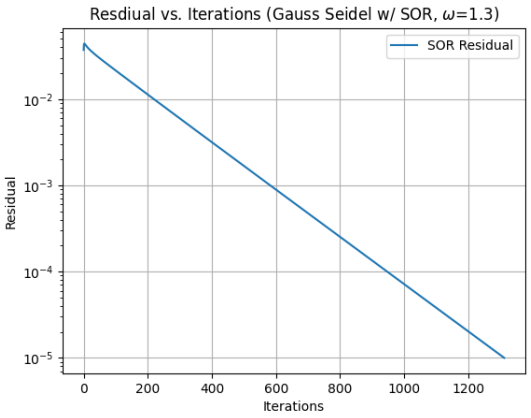
Numerical Results

Iteration Convergence Rate

The following graphics for $N = 2000$ are generated for the Gauss-SOR with $\omega = 1.3$ is provided for the residual-per-iteration.



Gauss-Seidel w/ SOR, $\omega = 1.3$ (1st Order Flux)



Gauss-Seidel w/ SOR, $\omega = 1.3$ (2nd Order Flux)

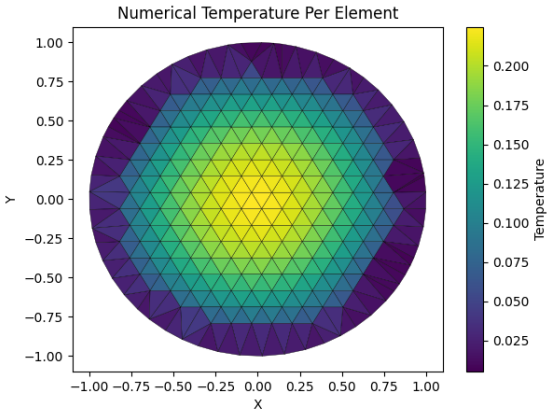
Figure 3: Comparison of iterations for Gauss-Seidel w/ SOR, $\omega = 1.3$, using 1st and 2nd order schemes.

From both graphics, the residual decreases linearly for each iteration, both having a slope of approximately -2, indicating their 2nd order convergence regardless of flux scheme. Note that although the number of iterations are similar, the time for each computation is significantly more complex for the 2nd order flux scheme due to accounting for 2 extra levels of neighbors per cell away from the edge.

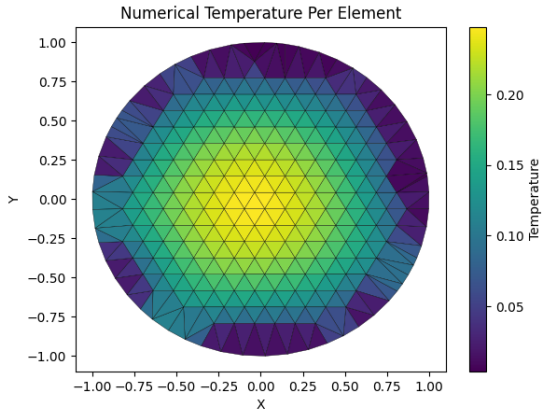
Convergence

Assuming the temperature T_i is uniform across the element, the following maps were made based on their circumcenter temperature. As shown by the graphs, the solution become more resolute with more triangle elements included.

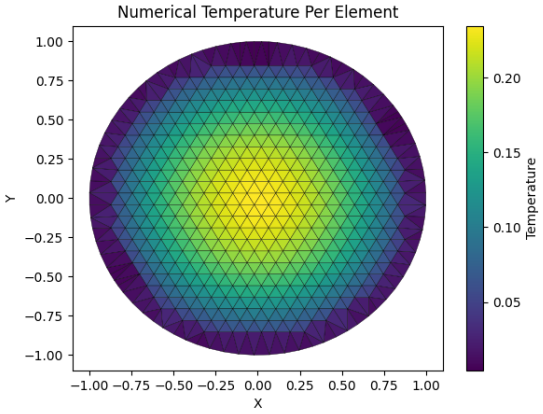
However, as seen in the results 2nd order flux in Figure 5, the figure is not convergent with respects to the boundary conditions. Due to neighboring $k = 2, 3$ cells present near the cell, the ghost flux correction (which is 1st order) is not convergent and subsequently is not calculated correctly. However, these cells are less-important in high-resolutions where diffusion may be present in 1st order schemes. It does, however, bias the center solution towards the left-side due to falsely higher temperatures present on the left.



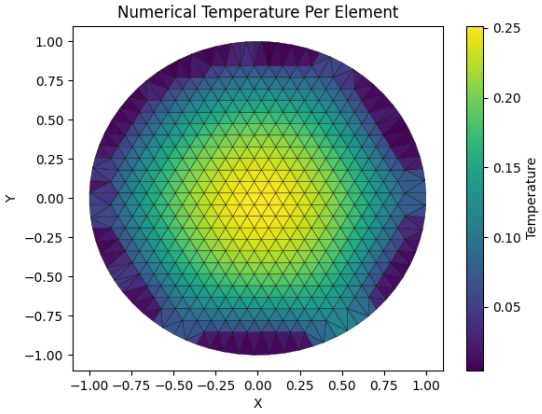
500 N – 1st Order



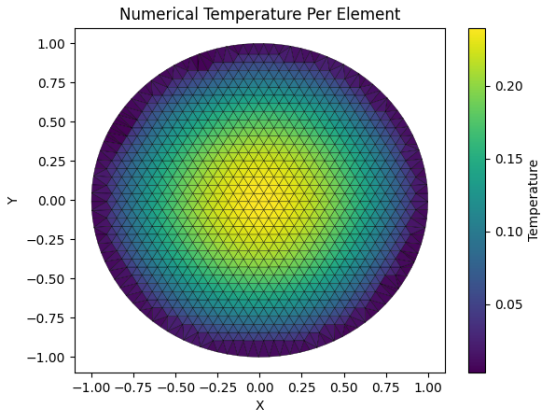
500 N – 2nd Order



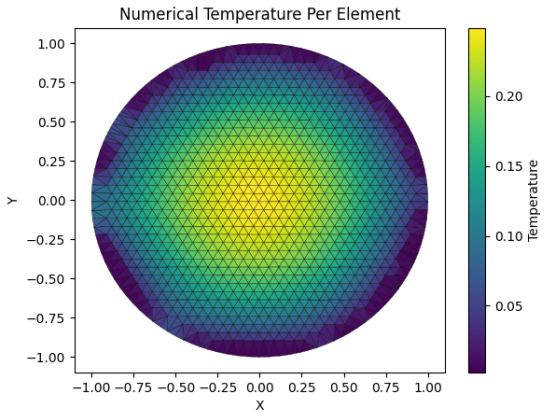
1000 N – 1st Order



1000 N – 2nd Order



2000 N – 1st Order



2000 N – 2nd Order

Figure 5: Comparison of numerical solutions for 500, 1000, 2000 nodes using 1st and 2nd order flux schemes.

Analytical Centerline Comparison

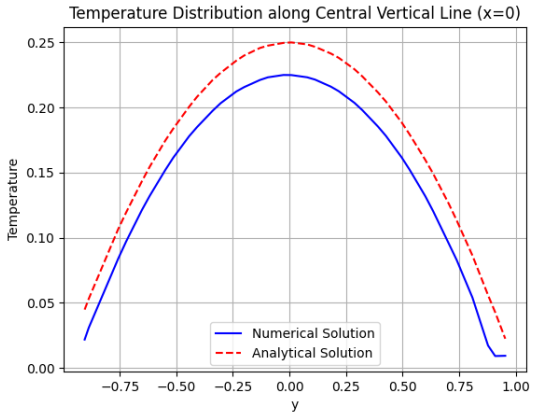
Assumptions

The following graphs were generated at the center-line. The main assumptions when generating this file is uniform temperature solution per element, such that plotting T_i at the center-line is the same as the circumcenter temperature of each element that is in contact with the center-line. The analytical solution was similarly generated uniformly at the circumcenter coordinates of the elements that intersected the center-line.

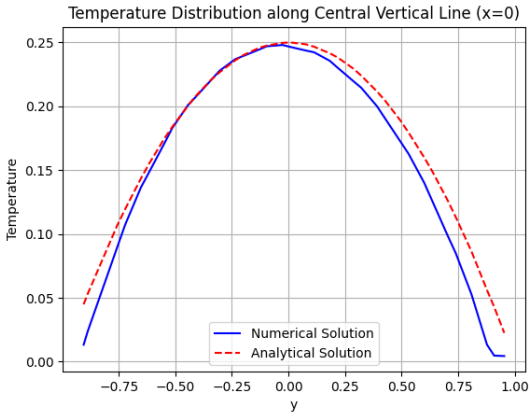
Graphing

The following graphs were generated with a $tol = 10^{-4}$ and Gauss-Seidel w/ SOR at $\omega = 1.3$.

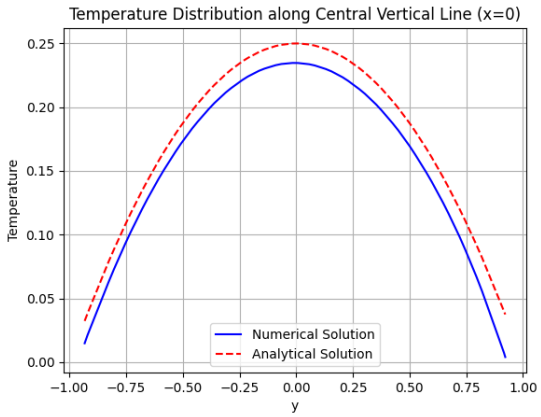
From the convergence of the solution across multiple $N = 500, 1000, 2000$, the numerical solution approaches to the desired analytical solution as the resolution increases. Thus, the solution is qualitatively convergent for both the first and second order flux.



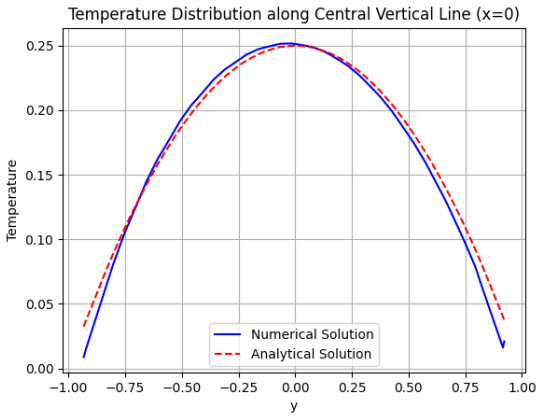
500 nodes – 1st Order



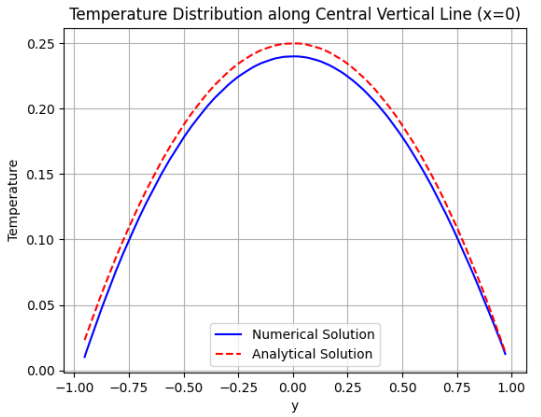
500 nodes – 2nd Order



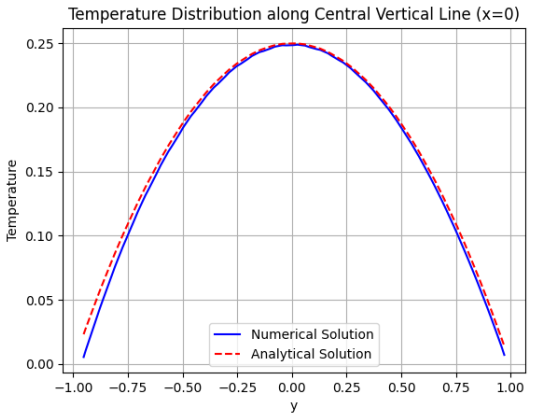
1000 nodes – 1st Order



1000 nodes – 2nd Order



2000 nodes – 1st Order



2000 nodes – 2nd Order

Figure 6: Comparison of centerline solutions for different resolutions using 1st and 2nd order schemes.

From the attached graphics in Figures 6, the 2nd order flux correction is quickly convergent with respect to the centerline, reducing the diffuseness from 1st order. However, this scheme ideation is not convergent in the BC treatment, allowing the edge conditions in each centerline of 2nd order to be off by some value. In low-resolution conditions such as Figures 6(a), 6(b), the 2nd order flux solution is biased from the boundary-elements present. Such boundary elements will pull the numerical solution away from the center with overly-strong flux. This over-correction may also lead to numerical overprediction in Figure 6(f).

Convergence vs. N

The metric chosen is a L2 norm between the analytical and numerical temperature data. This norm was chosen, as it compares the solution between the entire global domain for differences in T solutions against the entire domain. The L2 norm also emphasizes larger errors due to square scaling. Thus, this scaling considers all unknown numerical temperatures to the analytical solution within the full 2D grid, with a standard metric that emphasizes the larger discrepancies between results.

The following graph of L2 norm error was generated using a tolerance of $tol = 10^{-5}$. Duly note the numerical N used in this plot is the numerical diameter elements present for the uniform normalization.

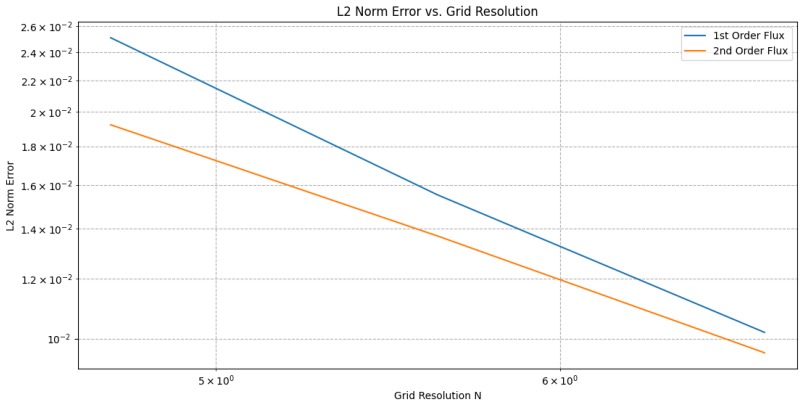


Figure 7: L2 Norm error convergence for given N spatial resolution

From the graph, the error will decrease in a exponential pattern with respect to the presence of nodes. Although initially the solution converges steeply with increased mesh resolution, the convergence slows significantly with the presence of more nodes. Thus, the relation between the error and grid convergence is not linear, but rather it is exponentially dependent on grid sizing. This could be due to uneven and skewed triangles, alongside errant triangles near the boundary compounding prior error to incorrectly bias the solution.

However, in the beginning the convergence is 2nd order due to the convective flux scheme being 2nd order accurate. The graph slope in the beginning is near -2, which is verified in the code as follows:

```
initial_slope = (np.log(err2nd[2] / err2nd[1])) / (np.log(N[2] / N[1]))
```

Which outputs the following solution for the initial slope:

1st Order Slope = -2.4258012644575504

2nd Order Slope = -2.059885783681103

Which is consistent with the scheme being 2nd order accurate. Duly note this slope is not maintained as N increases, which is caused by mesh-dominant convergence issues. Note the 1st-order scheme is high, but will decrease to the 2nd order flux scheme's rate of -2 later.

Conclusion

The second-order flux scheme demonstrates convergence at higher grid resolutions and consistently yields lower residuals compared to the first-order scheme. However, the slope of the spatial convergence rate remains similar to that of the first-order scheme, indicating that the second-order scheme does not show any substantial improvement in accuracy. Despite this, it consistently results in lower residual errors across varying spatial resolutions.

Therefore, the second-order flux scheme is most effective for low- to medium-resolution grids where central results, away from the boundary conditions, are of primary interest. For simulations involving the edges of geometries with varying mesh sizes, this scheme may under-predict the solution and produce inconsistent results.

Code

The author's code used within this paper is public-access and is provided in the below link.
https://github.com/limebat/2nd-Order-Flux–Upwind–for-Finite-Volume-Method-CFD/tree/main