Fall 2022 - Final Project

Akhil Perimbeti

COMP 670

December 10, 2022

The Monge-Amperé Equation

The Monge-Amperé equation is a fully nonlinear elliptic degenerate Partial Differential Equation with applications in classical problems of prescribed Gaussian curvature, optimal mass transport, and several other applications. Due to its importance in applied science and the difficulties arising from the non-linearities, there is a rising interest in the application of accurate and efficient numerical methods to find the solution to this PDE. This project introduces some related numerical methods, and uses the monotonicity properties to approximate the MA operator, and then use the multigrid method with Gauss-Seidel operations on it. It attempts to implement a more robust and time-efficient scheme to solve the Monge-Amperé equation, and do convergence studies for different discretizations and full multi-grid schemes.

The general form of the 2D Monge-Amperé Equation is shown as follows:

$$det(D^2U(x,y)) = f(x,y), \quad in \ \ \Omega \subset \mathbb{R}^2$$
$$U(x,y) = g(x,y), \quad on \ \ \delta\Omega$$

where $det(D^2)$ is the MA operator, and stands for finding the Hessian Matrix of the function U:

$$det(D^2U(x,y)) = U_{xx}U_{yy} - U_{xy}^2$$

As seen above, the PDE comes with Dirichlet Boundary conditions (boundaries of the true solution). There must be a constraint that u must be convex, in order for the equation to be elliptic. The convexity constraint is necessary for uniqueness of the solutions and is essential for numerical stability.

There are substantial numerical difficulties in solving this kind of equation:

- Non-Linearity: The equation is fully non-linear, and therefore the weak solutions are either geometric or viscosity solutions (Benamou et al., 2010). This places restrictions on using FEM or general numerical methods.
- Singularity (Weak Solutions): For singular solutions, the appropriate notion of a weak solution must be used. The weak solutions can be singular especially if the source function f is not strictly positive. Oliker and Prussner (1989) presented a method that converges to the Aleksandrov solution and Oberman (2008) introduced a wide stencil finite difference method which converges to the viscosity solution. Both of these methods were restricted to two dimensions.
- Convexity Constraints: There must be convex solution$u$ in order for the Monge-Amperé equation to be elliptic. The convexity constraint is necessary for uniqueness of the solutions and is essential for numerical stability.
- Accuracy: Oberman (2008) implemented a convergent monotone scheme using a wide stencil, and the accuracy of the scheme depended on the width of the stencil, as we show below. Singular solutions using second-order finite difference methods produce results that are only first-order accurate. It is challenging to implement a fast solvers with higher accuracy, often trading one for the other.

Problem Introduction

For our project we consider the nonlinear Monge-Amperé equation in the following form as a smooth example:

$$det(D^2U(x,y)) = (1+x^2+y^2)\cdot e^{x^2+y^2}, \quad in\ \ \Omega \subset [0,1]\times[0,1]$$
$$U(x,y) = e^{x^2+y^2}, \quad on\ \ \delta\Omega$$

Figure 1 below shows the true analytical solution $U(x,y)$ on the Domain: [-1, 1] × [-1, 1] for the finest grid.

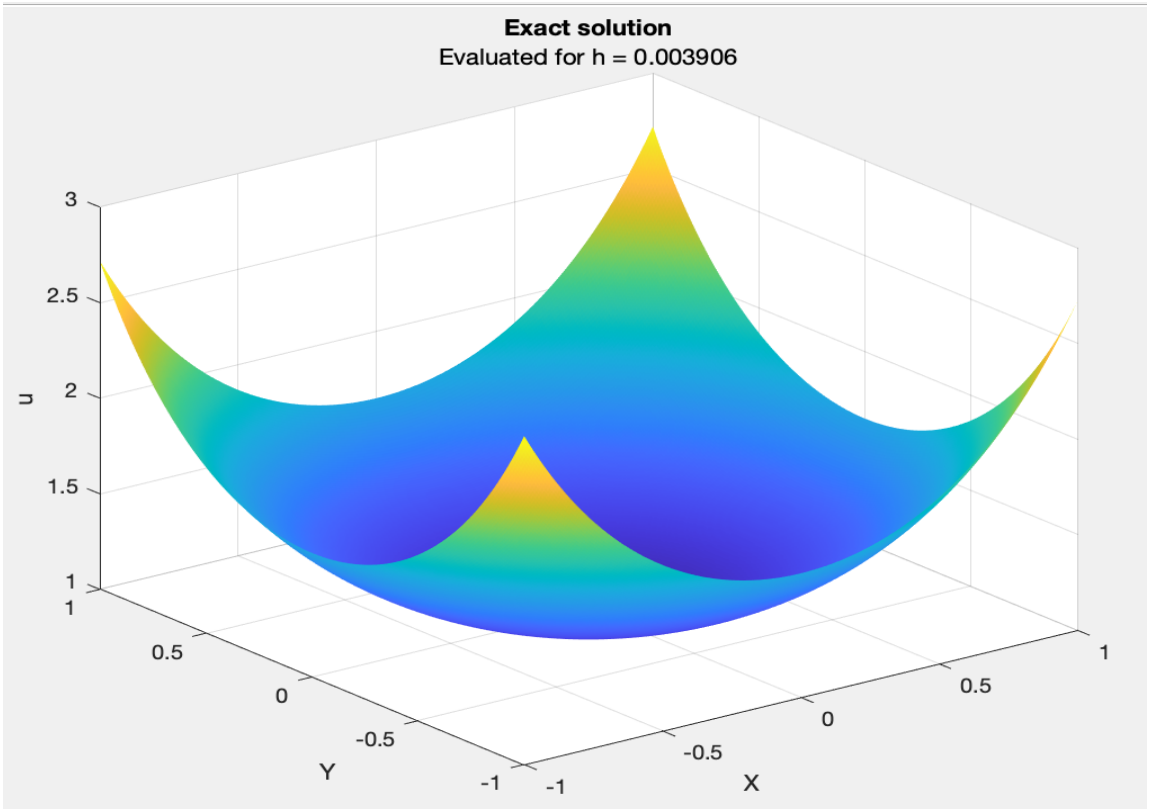


Figure 1: Exact solution on [-1, 1] × [-1, 1], N = 257

Multigrid Method with Wide-Stencil Scheme

We say an update scheme is monotone if for all the discrete solutions U and V, $\forall\ n \geq 1$:

$$\forall j,\ U_j^n \geq V_j^n \Longrightarrow \forall j,\ U_j^{n+1} \geq V_j^{n+1}$$

Monoticity is very important to the stability of numerical schemes. Most schemes are monotone if and only if the CFL condition is met (i.e forward euler scheme for the heat equation, upwind scheme for the advection equation, lax-friedrichs scheme for the advection equation).

The MA operator is monotone in the sense that

If we look at the simple numerical discretization of the determinant operator $u_{xx}u_{yy} - u_{xy}^2$ on a 9-point stencil:

$$det(D^2u(i,j)) = \frac{u_{i+1,j}+u_{i-1,j}-2u_{i,j}}{h^2} \times \frac{u_{i,j+1}+u_{i,j-1}-2u_{i,j}}{h^2} - \frac{(u_{i+1,j+1}+u_{i-1,j-1}-u_{i+1,j-1}+u_{i-1,j+1})}{h^4}$$

We can denote the root finding algorithm by **P**, i.e,

$$U_{i,j}^{n+1} = P(U_{i+1,j+1}^n, U_{i+1,j}^n, U_{i+1,j-1}^n, U_{i,j+1}^n, U_{i,j-1}^n, U_{i-1,j+1}^n, U_{i-1,j}^n, U_{i-1,j-1}^n, f_{ij})$$

However if P increases as $U_{i+1,j+1}^n, U_{i-1,j-1}^n$ increases, then P will decrease as $U_{i-1,j+1}^n, U^{i+1,j-1}$ increases, and vice versa. It is impossible for P to be monotonically increasing with respect to the four arguments $U_{i+1,j+1}^n, U_{i-1,j-1}^n, U_{i-1,j+1}^n$, and $U_{i+1,j-1}^n$ at the same time. There is no way to guarantee that $(MA_h u)_{ij} \geq (MA_h v)_{ij}$, and so this simple 9-point discretization will never be monotone, which contradicts the monoticity of the determinant operator.

Using the convexified MA operator $(MA)^+$ proposed by Oberman et. al (2011), we can adopt a new discretization on the (2L+1)(2L-1)(L$\in \mathbb{N}$) stencil (Benamou et. al, 2016):

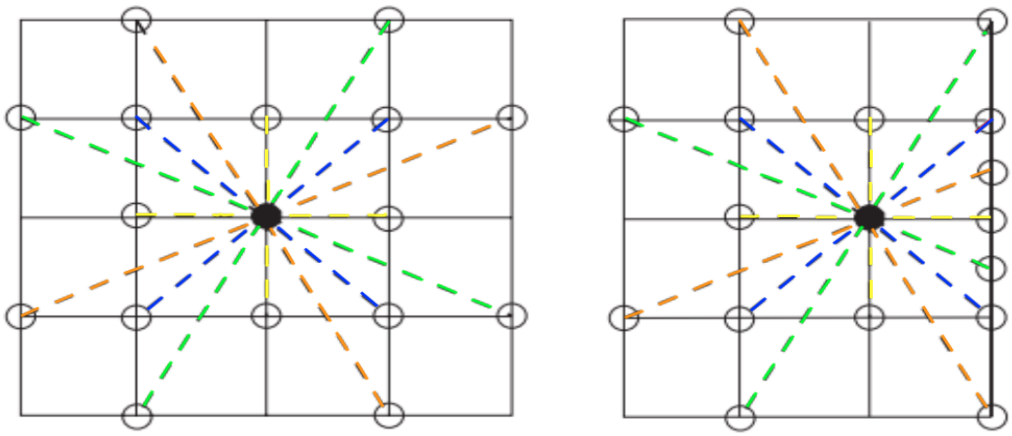
$$det(D^2 u_i j) \approx max\{0, (MA_h^+ u)_i j\},$$

where,

$$(MA_h u)_i j = \min_{\substack{0<m\leq L, 0\leq l\leq L \\ gcd(m,l)=1}} \frac{u_{i+m,j+l} - 2u_{ij} + u_{i-m,j-l}}{m^2h^2 + l^2h^2} \times \frac{u_{i+l,j-m} - 2u_{i,j}, + u_{i-l,j+m}}{l^2h^2 + m^2h^2}$$

Since u is convex, we know that the $det(D^2 u) \geq 0$, so we can take the maximum value of 0 and the minimization result. This discretization is called a Wide-Stencil scheme denoted as D_V^{WS}. Figure 2 below displays a 5x5 stencil (using L=2), where the four different colors represent the four orthogonal bases of different orientations. If we were to chose a 3x3 stencil (L=1), then only the yellow and blue bases would be available.

Yellow - 0°, Orange - 27°, Blue - 45°, Green - 63°



(a) In the interior. (b) Near the boundary.

Figure 2: D_V^{WS} on a 5x5 stencil (Oberman and Froese, 2011)

The main advantage of this new scheme is that $u_{i+m,j+l} - 2u_{ij} + u_{i-m,j-l}$ is the standard central second-order difference, which has a better monotonicity than $u_{i+1,j+1} + u_{i-1,j-1} + u_{i-1,j+1} + u_{i+1,j-1}$. After finding the minimizing pair m^* and l^*, the following equation is solved with respect to U_{ij}^{n+1} (choosing the smaller root for the sake of convexity):

$$\frac{U_{i+m^*,j+l^*}^n - 2U_{ij}^{n+1} + U_{i-m^*,j-l^*}^n}{(m^*h)^2 + (l^*h)^2} \times \frac{U_{i+l^*,j+m^*}^n - 2U_{ij}^{n+1} + U_{i-l^*,j-m^*}^n}{(l^*h)^2 + (m^*h)^2} = f_{ij}$$

We can denote the root finding algorithm by **P**, i.e,

$$U_{i,j}^{n+1} = P(U_{i+m^*,j+l^*}^n, U_{i-m*,j-l^*}^n, U_{i+l^*,j-m^*}^n, U_{i-l^*,j+m^*}^n, f_{ij}, m^*, l^*)$$

It can be shown that P increases as $U^n_{i+m^*,j+l^*}, U^n_{i-m*,j-l^*}, U^n_{i+l^*,j-m^*} and U^n_{i-l^*,j+m^*}$ increases, allowing P to be monotonically increasing with respect to those four arguments. This new scheme is monotone, as it has inherited the monotonicity of the continuum MA operator, and it can be guaranteed that $(MA^+_h u)_{ij} \geq (MA^+_h v)_{ij}$. (Benamou et al, 2016). However, a drawback of this new discretization is that, in order to increase accuracy, a very wide stencil must be used since angular resolution must be considered, and this often results in error dominating in the cases where the solutions are singular. Therefore our code and results below implement a 3x3 stencil, with only the yellow 0° and blue 45° orthogonal bases available.

Iterative algorithm using D^{WS}_V on Multigrid

After discretization, the Monge-Amperé equation is reduced to a system of non-linear equations. The Full-Approximation Scheme (FAS) is used to solve these equations efficiently on multigrid. The algorithm is structured as follows:

NOTE: This is the FAS for a depth of only 2 grid layers (h and $2h$).

Solve the equation: $MA_h(u^h) = f^h \longrightarrow$ **Full Approximation Scheme (FAS)** (Briggs et al, 2000)

Aim: To solve discrete non-linear equations on a fine grid: $MA_h(u^h) = f^h$, with grid size h and threshold ε
Ensure: The numerical solution v^h

---

1. Initialze v^h, a current intital approximation of the solution on a fine grid.
2. Compute the residual r^h on the fine grid: $r^h = f^h - MA_h(v_h)$
3. **WHILE** (residual $r^h >$ threshold ε), **then**
4. Restrict residual r^h to the course grid: $r^{2h} = I_{h\to 2h} r^h$
5. Restrict the approximation solution v^h to the course grid: $v^{2h} = I_{h\to 2h} v^h$
6. Solve the restricted problem on the course grid: $MA_{2h}(u^{2h}) = MA_{2h} v^{2h} + r^{2h}$
7. Calculate the Error on the course grid: $e^2 h = u^{2h} - v^{2h}$
8. Interpolate error e^{2h} to the fine grid: $e^h = I_{2h\to h} e^{2h}$
9. Use the error e^h to update the current approximation v^h on the fine grid: $v^h \leftarrow v^h + e^h$
10. Compute the residual r^h on the fine grid: $r^h = f^h - MA_h(v_h)$.
11. **END** (if exit condition $r^h > \varepsilon$ is met)
12. **RETURN:** Fully updated numerical solution v^h

NOTE: $I_{h\to 2h}$ denotes the restriction operator and $I_{2h\to h}$ denotes the interpolation operator.

It is important to note that the non-linear equations solved on the course grid is equivalent to:

$$MA_{2h}(u^{2h}) = I_{h\to 2h} f^h + MA_{2h}(I_{h\to 2h} v^h) - I_{h\to 2h} MA_h(v^h)$$

instead of: $MA_{2h}(u^{2h}) = I_{h\to 2h} f^h$, because MA operator and $I_{h\to 2h}$ operator do not commute in general. This means that the equations on the course grid depend on the current solution on the fine grid. Briggs, Mccormick and Henson (2000) introduce the τ (tau) correction term $MA_{2h}(I_{h\to 2h} v^h) - I_{h\to 2h} MA_h(v^h)$ above, that alters the equations on the course grid and reduces the resolution error arising from the relatively large step size of $2h$.

As stated previously, the algorithm shown above is the FAS implemented with two-layers. The FAS method can also be applied by recursively solving the restricted problem on a coarser grid, thereby reducing the size of the problem, i.e. MA_h, MA_{2h}, MA_{4h}, MA_{8h}, MA_{16h}, ... ,$MA_{2^{(L-1)}h}$ for an L-layer approach, which can be implemented with a V-cycle scheme. The V-cycle scheme is essentially the same as the two-grid FAS algorithm described above, with the difference being that the V-cycle is the two-grid FAS algorithm implemented recursively for solving with more than just two layers. This project implements the two-layer version of multi-grid algorithm to solve this equation. The number of iterations represents one V-cycle in the multigrid which can be from 1-4 levels in depth.

Results

The true solution was plotted above, in Figure 1, for a grid size of N=129 ($h = 0.003906$). Figure 3 below displays the numerical solution v calculated using the D_V^{WS} scheme using multigrid, for a grid size of N=129 ($h = 0.003906$). We can see that at the finest grid, the numerical solution is almost identical to the exact solution, indicating that scheme performs very well and accurately in approximating a numerical solution.

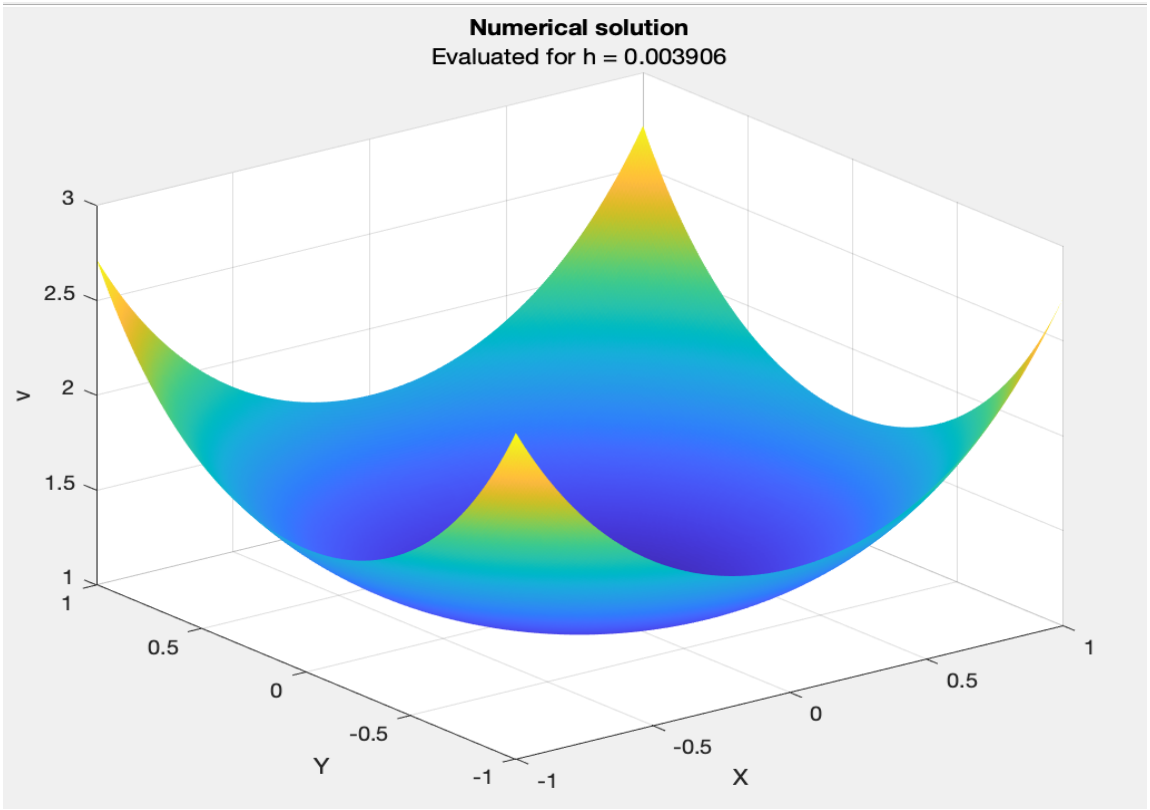


Figure 3: Numerical solution on [-1, 1] × [-1, 1], N = 257

Tables 1-4 displays the summary of the efficiency results of the scheme for each different grid size at every depth level. The number of iterations represents one V-cycle in the 4-level multigrid, which decreases with every level of recursion. The residual tolerance was set to $h^2/10$, allowing us to control the error, which decreases as mesh size increases. Refer to Figure 8 for visualizing error behavior over different meshes and efficiency at different depths.

N	h	total iterations
33	0.0313	4
65	0.0156	16
129	0.0078	68
257	0.0039	298

Table 1: **depth = 1**

N	h	total iterations
33	0.0313	2
65	0.0156	3
129	0.0078	14
257	0.0039	61

Table 2: **depth = 2**

N	h	total iterations
33	0.0313	2
65	0.0156	3
129	0.0078	3
257	0.0039	15

Table 3: **depth = 3**

N	h	total iterations
33	0.0313	2
65	0.0156	3
129	0.0078	3
257	0.0039	8

Table 4: **depth = 4**

Figure 4 contains 4 subplots below displaying the absolute error ($|u(x,y) - v(x,y)|$) at different iteration steps. Figure 5 contains 4 subplots below displaying the residual error ($f(x,y) - detD^2(v(x,y))$) at different iteration steps. These figures show the error and residual behavior on the domain over a single iteration cycle. Both behaviors were computed using a mesh of N = 257 on the **first-level (depth = 1)** of the multigrid (i.e. first recursion).

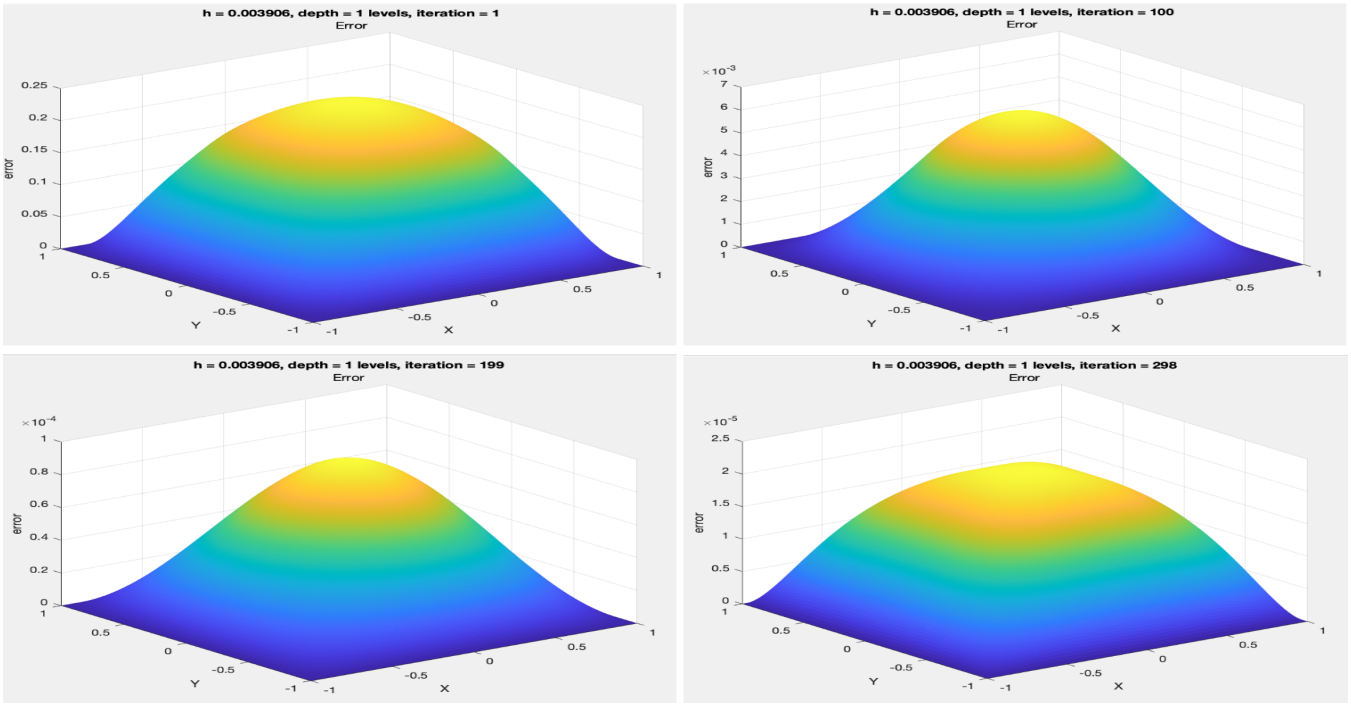


Figure 4: Absolute Error Distributions on mesh N = 257, depth = 1
at iteration steps: 1, 100, 199, 298 respectively

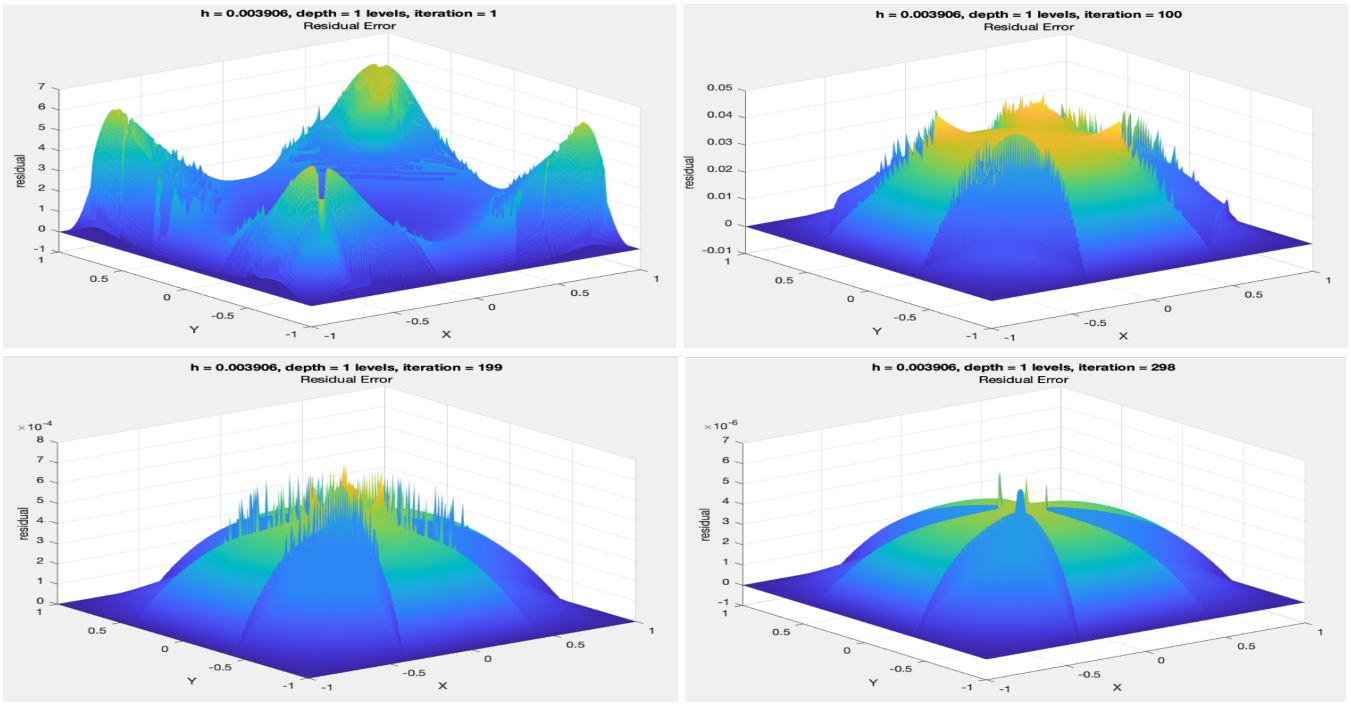


Figure 5: Residual Error Distributions on mesh N = 257, depth = 1
at iteration steps: 1, 100, 199, 298 respectively

Figure 6 contains 4 subplots below displaying the absolute error ($|u(x,y) - v(x,y)|$) at different iteration steps. Figure 7 contains 4 subplots below displaying the residual error ($f(x,y) - detD^2(v(x,y))$) at different iteration steps. These figures show the error and residual behavior on the domain over a single iteration cycle. Both behaviors were computed using a mesh of N = 257 on the **second-level (depth = 2)** of the multigrid (i.e. second recursion).

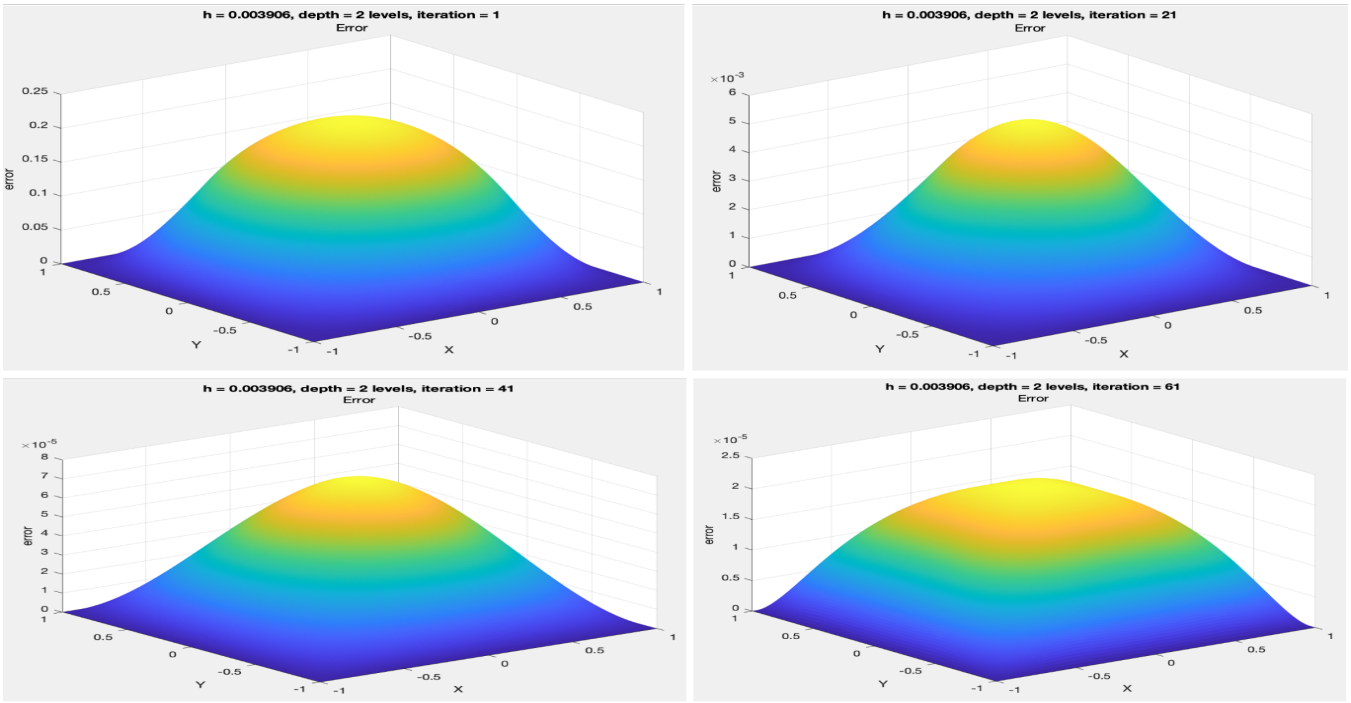


Figure 6: Absolute Error Distributions on mesh N = 257, depth = 2
at iteration steps: 1, 21, 41, 61 respectively

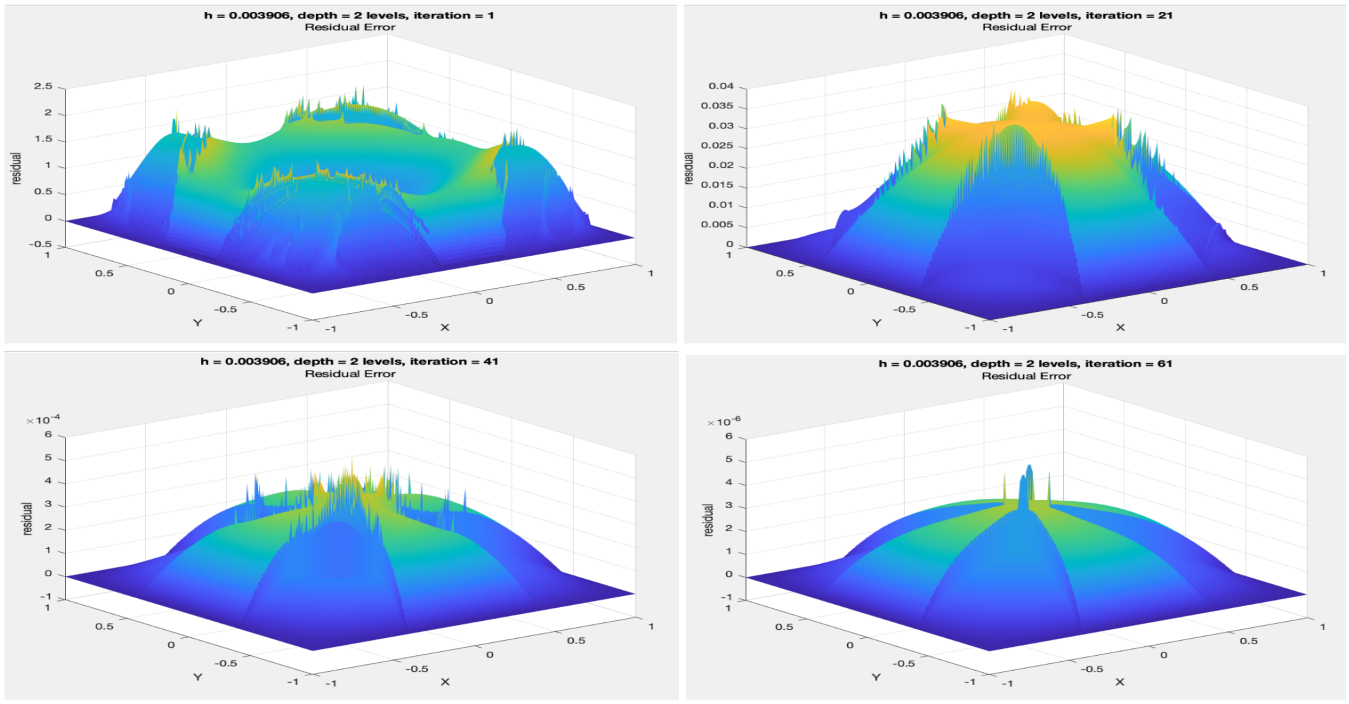


Figure 7: Residual Error Distributions on mesh N = 257, depth = 2
at iteration steps: 1, 21, 41, 61 respectively

Looking at Figure 4 and Figure 6, it is clear that the errors keep positive and gradually decay to zero as the number of iterations increase (denoted on the z-axis as the scale shifts from 10^{-2} to 10^{-6}). It is important to note that a few Gauss-Seidel iterations were used damp the high-frequency components of the error, and were represented on a coarser mesh once the smoothing had occurred. Looking at Figure 5 and Figure 7, it is clear to see that the residuals oscillate intensely with iterations but show a overall decreasing and smoothing trend as the number of iterations increase. Only the error and residual behaviors for the first two levels are displayed above, but it is safe to say that the results for depths 3 and 4 essentially follow a similar pattern, but with fewer iterations.

The magnitude of error at different mesh sizes h is explored in Figure 8 (2) below, which uses the results displayed in Tables 1, 2, 3, and 4 above. Figure 8 (2) plots the total iterations for each h at every depth level, displaying the efficiency of using multiple levels of recursion in multigrid. We can see that the more levels we use on a finer grid, the fewer iterations it takes for the numerical scheme to reach the set error threshold. The D_V^{WS} method on multigrid seems to be efficient when scaling to finer grids, and can be built further upon for better solvers of the MA equation.

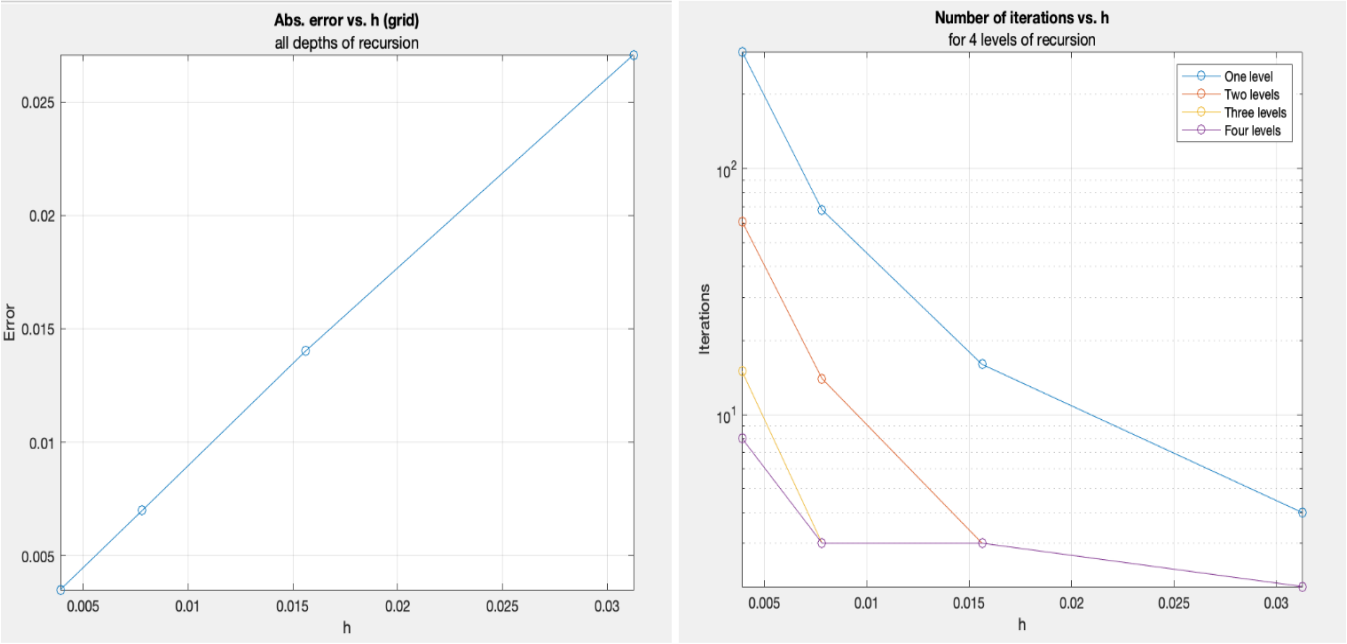


Figure 8: (1) Absolute Error behavior for different mesh sizes and (2) Iteration counts for different mesh sizes N =[33, 65, 129, 257], h = [0.0313, 0.0156, 0.0078, 0.0039]

Figure 8 (1) shows how the error behaves, which seems pretty linear (1^{st} order accurate) as we have set it to be. We can prove this by calculating the Order of Accuracy (m) of the numerical scheme, using calculated error results for $h = 0.0313$ and $\frac{h}{2} = 0.0156$, and plugging them into the slope formula: (rise)/(run) on a log-log scale $\rightarrow$

$$m = \frac{log(E_h) - log(E_{h/2})}{log(h) - log(h/2} = \frac{log(E_h/E_{h/2})}{log(h/h/2)} = \frac{log(E_h/E_{h/2})}{log(2)} \longrightarrow \frac{log(0.0279/0.0139)}{log(2)} \implies m \approx \mathbf{1.00518}$$

.

Conclusions

This project builds on the work of other researchers and previous projects. We use a numerical method that combines a nonlinear Gauss-Seidel iterative method with a centered difference discretization on a variety of different coordinate systems, i.e wide-stencil with orthogonal bases. This method was shown to stable because the underlying scheme preserves monotonicity. In order to solve these non-linear systems efficiently, a V-cycle FAS on multigrid method is used. A residual tolerance is set which exploits error correction within the recursive algorithm; This scheme is used to leverage the low cost of computation on the coarse grids to build up the finer grids. This project shows computational results that demonstrate the time-efficiency and robustness of the numerical algorithm for solving the Monge-Amperé equation. The code for this project is attached seperately (Monge_Ampere.zip).

References

Adam M. Oberman. Wide stencil finite difference schemes for the elliptic Monge-Ampère equation and functions of the eigenvalues of the Hessian. *Discrete and Continuous Dynamical Systems. Series B*, Volume 1. 2008.

Brittany D. Froese ans Adam M. Oberman. Convergent finite difference solvers for viscosity solutions of the elliptic monge–ampere equation in dimensions two and higher. *SIAM Journal on Numerical Analysis, 49(4):1692–1714*, 2011.

Cristian E. Gutiérrez and Haim Brezis. *The Monge Ampere equation.* Volume 44, Springer. 2001

Grégoire Loeper and Francesca Rapetti. Numerical solution of the monge–ampere equation by a newton's algorithm. *Comptes Rendus Mathematique, 340(4):319–324*, 2005.

Ivan Mitevski, Matthew Illingworth, David Youssif, and Brittany Froese . *Numerical methods for solving Monge-Ampere equation.* Preliminary Report. 2017

Jean-David Benamou, Brittany D. Froese and Adam M. Oberman. Two numerical methods for the elliptic Monge-Ampère equation. *ESAIM: Mathematical Modelling and Numerical Analysis, Volume 44, no. 4, pp. 737-758.* 2010

Jean-David Benamou, Francis Collino, and Jean-Marie Mirebeau. Monotone and consistent discretization of the monge-ampere operator. *Mathematics of computation, 85(302):2743–2775*, 2016.

William S. Briggs and Steve F. Mcormick *A Multigrid Tutorial,* 2^{nd} *edition*, Volume 72, Ch.6, pp. 95-109. 2000