Lab 1: Wide Stencil Method
Solving Viscosity Solution for Monge–Ampère Equation

Ting Lin, 1700010644

May 11, 2020

Contents

1 Problem Setting and Finite Difference

1.1 Monge–Ampère Equation

We solve the following Monge–Ampère Equation in 2D.

$$\det(D^2 u) = f \text{ in } \Omega \tag{1}$$
$$u|_{\partial\Omega} = g \text{ on } \partial\Omega \tag{2}$$

with some convexity assumption ($f > 0$). Here and throughout this report we assume that $\Omega = [0,1]^2$. We can prove the existence and uniqueness of viscosity solution under certain condition. In this lab we focus on wide stencil method to solve the viscosity solution. Notice that

$$\det(D^2 u) = \min_{\{v_1,v_2\}\in\mathbb{V}} (v_1^T D^2 u v_1)(v_2^T D^2 u v_2) = \min_{\{v_1,v_2\}\in\mathbb{V}} (\frac{\partial^2 u}{\partial v_1^2})(\frac{\partial^2 u}{\partial v_2^2})$$

where $\mathbb{V}$ is the set of all othronormal bases. To enforce the convexity, we introduce

$$\text{MA} = \min_{\{v_1,v_2\}\in\mathbb{V}} (\frac{\partial^2 u}{\partial v_1^2})^+(\frac{\partial^2 u}{\partial v_2^2})^+$$

where $a^+ = \max(a, 0)$ is the positive part.

1.2 Wide Stencil Method

For WS method, we suppose there is a grid with size $h = 1/N$, and we consider the function value only on the interior grid point. Introduce the discrete MA function need to both approximate $\mathbb{V}$ and the second order directional derivative. For the former, we consider using the direction whose vector lies exactly on some stencil, for example $([1, 0], [0, 1])$. The typical and most simplest choices are 9-point basis

$$\mathbb{V}_9 = \{\{[1, 0], [0, 1]\}, \{[1, 1], [-1, 1]\}\};$$

17-point basis

$$\mathbb{V}_{17} = \{\{[1, 0], [0, 1]\}, \{[1, 1], [-1, 1]\}, \{[2, 1], [-1, 2]\}, \{[1, 2], [-2, 1]\}\};$$

and 33-point basis

$$\mathbb{V}_{33} = \mathbb{V}_{17} \ \cup \ \{\{[3, 1], [-1, 3]\}, \{[1, 3], [-3, 1]\}, \{[3, 2], [-2, 3]\}, \{[2, 3], [-3, 2]\}\};$$

For the former we use standard techniques in Poisson equation: Fix an interior point x_h and for some direction e, denote $\tilde{\rho}$ being the maximum positive real number such that $x_h + \tilde{r}hoe$ is in Ω. Then we set $\rho = \min(\tilde{\rho}, 1)$.

For $e^+, e^- = \pm e$, we use the following difference

$$\Delta_e u_h(x) = \frac{2}{(\rho^+ + \rho^-)|e|^2 h^2}\left[\frac{u_h(x_h + \rho^+ h e^+) - u_h(x_h)}{\rho^+} + \frac{u_h(x_h + \rho^- h e^-) - u_h(x_h)}{\rho^-}\right]$$

to approximate the second order derivatives, which only use the value of interior grid point and boundary condition. The implementation of directional difference and central difference is in **dir_diff.m** and **central_diff.m**.

Combing both, we obtain the discrete functional

$$MA_h[u_h](x_h) = \min_{\{v_1, v_2\} \in \mathbb{V}_h} (\Delta_{v_1} u_h(x_h))^+ (\Delta_{v_2} u_h(x_h))^+.$$

The WS method is consider the following problem

$$MA_h[u_h](x_h) = f(x_h)$$

in all interior grid point, and the following result guarantees the convergence. [TBC]

2 Implementation Detail

The equation $MA_h[u_h](x_h) = f(x_h)$ is a nonlinear equation and we will use damped newton method to solve it. Since a^+, $\min(a, b)$ is not smooth, which will hamper the behavior of Newton-type method, we use $a^{+,\delta} := max^\delta(a, 0)$ and $\min^\delta(a, b)$ to replace the original max and min. Here

$$\max^\delta(x, y) = \frac{1}{2}(x + y + \sqrt{(x-y)^2 + \delta}) \qquad \min^\delta(x, y) = \frac{1}{2}(x + y - \sqrt{(x-y)^2 + \delta})$$

2.1 Computations of mollified MA function and Jacobian

The mollified

$$MA_h^\delta[u_h](x_h) = \min{}^\delta{}_{\{v_1,v_2\}\in\mathbb{V}_h}(\Delta_{v_1}u_h(x_h))^{+,\delta}(\Delta_{v_2}u_h(x_h))^{+,\delta}.$$

is used in computation, and we choose 1e-6 in our implementation. Notice that here $\min_\delta$ is not associative, therefore the order of elements in $\mathbb{V}_h$ should be fixed. We list the algorithm below, and the implementation is in **MAFunction.m**.

Algorithm 1 $y_h = \mathrm{MA}_h^\delta[u_h](x_h)$

for $i = 1 : |\mathbb{V}_h|$ **do**
  $v_1, v_2 = (\mathbb{V}_h)_i$
  $A = \Delta_{v_1}u_h(x_h), B = \Delta_{v_2}u_h(x_h)$
  $MA[i] = \max^\delta(A, 0) * \max^\delta(B, 0)$
end for
$y_h = \min^\delta(MA[1], MA[2])$
for $i = 3 : |\mathbb{V}_h|$ **do**
  $y_h = \min^\delta(y_h, MA[i])$
end for
return y_h

For simplicity of Newton Direction, we will not construct the Jacobian directly since it is too ugly. Instead, we only gives the function handle computing the directional derivatives and use Krylov Subspace Method to find the Newton direction. In practice we use the Arnoldi procedure in GMRES to construct Jacobian. The Jacobian is implemented in **MAJacobi.m**, based on the affine property of Δ_e and the chain rule, we use Δ_e^0 denote the central difference of u_h under zero boundary condition.

Algorithm 2 $q_h = \mathrm{MAJ}_h^\delta[u_h](x_h; \phi_h)$

for $i = 1 : |\mathbb{V}_h|$ **do**
  $v_1, v_2 = (\mathbb{V}_h)_i$
  $A = \Delta_{v_1}u_h(x_h), B = \Delta_{v_2}u_h(x_h)$
  $GA = \Delta_{v_1}u_h(\phi_h), GB = \Delta_{v_2}u_h(\phi_h)$
  $MA[i] = \max^\delta(A, 0) * \max^\delta(B, 0)$
  $MAJ[i] = \max^{\delta'}(A, 0) * \max^\delta(B, 0) * GA \max^\delta(A, 0) * \max^{\delta'}(B, 0) * GB$
end for
$y_h = \min^\delta(MA[1], MA[2])$
$q_h = \min^{\delta'}(MA[1], MA[2]) * MAJ[1] + \min^{\delta'}(MA[2], MA[1]) * MAJ[2]$
for $i = 3 : |\mathbb{V}_h|$ **do**
  $q_h = \min^{\delta'}(y_h, MA[i]) * q_h + \min^{\delta'}(MA[i], y_h) * MAJ[i]$
  $y_h = \min^\delta(y_h, MA[i])$
end for
return y_h

2.2 Damped Newton Method

After getting the Newton direction d_h, we use damped newton method to solve the nonlinear equation. The algorithm is listed below and implemented in **WideStencil.m**.

Algorithm 3 Wide Stencil Method

Choose an initial guess $u_h = u_h^0$, $r_h = \|\,\mathrm{MA}_h^\delta(u_h) - f_h\|_F$
while $r_h < tol$ **do**
  Compute the Newton direction d_h.
  Renormalization $d_h = d_h/(1 + \|d_h\|_F)$
  $\lambda = 1, \sigma = 0.6$
  for $i = 1 : MAXIT$ **do**
    $v_h = u_h + \lambda d_h$
    $q_h = \|\,\mathrm{MA}_h^\delta(v_h) - f_h\|_F$
    if $q_h < r_h$ **then**
      break
    end if
    $\lambda = \lambda\sigma$
  end for
  if $i = MAXIT$ **then**
    $\lambda = 1, \sigma = 0.6, d_h = -d_h$
    for $i = 1 : MAXIT$ **do**
      $v_h = u_h + \lambda d_h$
      $q_h = \|\,\mathrm{MA}_h^\delta(v_h) - f_h\|_F$
      if $q_h < r_h$ **then**
        break
      end if
      $\lambda = \lambda\sigma$
    end for
  end if
  $u_h = v_h$
  $r_h = q_h$
end while

The tolerance is chosen as $\epsilon(1 + \|f_h\|_F)$, and ϵ is chosen as 1e-5 or 1e-6.

2.3 Avoid Singularity

We proposed two methods to avoid possible singularity raised in Newton method.

The first is first used a sufficient large δ (1 or 10 for example), and solve the MA equation with $\delta_i = \delta/10^i$ consequently. When $\delta_i = 1e - 6$ is out ultimate goal ,, we have a sufficiently good initial point and newton method will proceed without singularity.

The second is to find a proper initial value directly, by taking the boundary condition into consideration. We solve the following Poisson equation

$$\Delta u = 2\sqrt{f} \text{ in } \Omega \tag{3}$$
$$u = g \text{ on } \partial\Omega \tag{4}$$

and obtain the initial guess u_h^0.

In practice, both strategies works well and reduction much work. And the Poisson initialization is 2-3 times efficient than graded δ method. So we provide the solver in **solver.m** by poisson initialization method.

3 Numerical Results and Discussions

We first solving Question 1 without grading delta.

Table 1: Numerical Result of Question 1

Size	10		20		40		80	
Basis	Time	Error	Time	Error	Time	Error	Time	Error
9	0.32	2.81E-03	0.47	1.86E-03	1.15	1.65E-03	11.97	1.60E-03
17	0.63	1.34E-03	0.79	8.89E-04	1.86	5.68E-04	14.93	4.72E-04
33	1.25	4.63E-03	1.85	7.56E-04	4.81	3.61E-04	31.75	2.12E-04

We solve Question 2 from $\delta = 10$ and utilize graded delta strategy.

Table 2: Numerical Result of Question 2

Size	10		20		40		80	
Basis	Time	Error	Time	Error	Time	Error	Time	Error
9	1.11	3.50E-03	2.23	2.13E-03	6.46	1.99E-03	48.28	1.96E-03
17	1.99	2.50E-03	5.19	9.00E-04	12.77	5.88E-04	89.53	5.07E-04
33	3.98	2.70E-03	9.36	4.78E-04	25.61	6.85E-04	111.89	8.38E-04

We solve Question 3 from $\delta = 10$ and utilize graded delta strategy. Additional work is needed for size = 80, we start from $\delta = 100$ in that column.

Table 3: Numerical Result of Question 3

Size	10		20		40		80	
Basis	Time	Error	Time	Error	Time	Error	Time	Error
9	0.99	8.97E-03	2.76	3.19E-03	6.49	1.13E-03	104.59	8.47E-04
17	1.92	4.10E-03	4.92	3.19E-03	13.75	1.13E-03	129.56	4.00E-04
33	4.31	3.92E-03	8.67	1.28E-03	29.68	1.05E-03	224.39	4.00E-04