SparseMatrixSolutor v3.2 Numerical Report

Anqi Lv*
PHYS 2200011316

2025.10.22

I Numerical Method

i Process

In this part I will explain how this program works.

The program is used for solving ParabolaCN and PoissonEQ. PoissonEQ has been introduced in the previous version. Now ParabolaCN is updated. It is used for solving eqations like:

$$\begin{cases} u_t + \alpha(u_{xx} + u_{yy}) = f(x, y, t) & (1) \\ u(x, y, 0) = u_0(x, y) \ \ ((x, y) \in \bar{\Omega}) & (2) \\ u(x, y, t) = g(x, y, t) \ \ ((x, y) \in \partial\Omega) & (3) \end{cases}$$

The program has PoissonEQ class and ParabolaCN class. ParabolaCN class contains PoissonEQ solver. So you should delare a PoissonEQ first, which contains net info and boundary info. Then you can declare ParabolaCN with PoissonEQ, alpha time net info.

The program initialize itself by using makeMatrix and makeConst method in PoissonEQ class, in order to initialize the LHS and RHS of differential form respectively.

Then to make sure the differential form is consistant with Parabola Equation, we have to operate the equation. I choose CN scheme to solve the equation:

$$\frac{U_j^{m+1}}{\tau} + \frac{\alpha}{2} * \frac{U_{j+1}^{m+1} - 2U_j^{m+1} + U_{j-1}^{m+1}}{h^2} = \frac{U_j^m}{\tau} - \frac{\alpha}{2} * \frac{U_{j+1}^m - 2U_j^m + U_{j-1}^m}{h^2} + f_j^{m+1/2}$$

*email:2200011316@stu.pku.edu.cn tel:18315391730

So I use Laplace Matrix by $\frac{\alpha\tau}{2}$ and add its diagonal with 1 to construct LHS. Multiply it to $u_{previous}$ and add $u_{previous}$ itself and source term to construct RHS.

The equation will then be solved by Gauss-Sider solver.

ii Grid

The grid method is the same as PoissonEQ. See the previous Numerical test for version 2.3 for details.

iii Boundary Condition

The ParabolaCN use the same method as in PoissonEQ, I use the number of couple term to confirm if it is a boundary term. If true, then the boundary condition will be kept.

iv Explicit Scheme

In v3.2 I add ParabolaExplicit class. It runs the same as CN scheme with a PoissonEQ to control the Boundary Condition. The only difference is that Explicit scheme use another differential scheme:

$$U_j^{m+1} = U_j^m - \tau\alpha * \frac{U_{j+1}^m - 2U_j^m + U_{j-1}^m}{h^2} + \tau f_j^m$$

It is worth mention that Explicit Scheme is much more faster than CN Scheme. But it's only stable when $\frac{\tau}{h^2} \leq \frac{1}{2}$. If this condition is broken, the solution will goes infinity(even when inf not happens, the solution is also incorrect).

II Numerical Test on CN Scheme

i Target Equation

For target equation:

$$\begin{cases} u_t - (u_{xx} + u_{yy}) = 100\delta(x-1, y-2.5)sin(50t) & (4) \\ u(x,y,0) = e^{-5*((x-1)^2+(y-2)^2)} \ \ ((x,y) \in \bar{\Omega}) & (5) \end{cases}$$

Boundary condition goes like:

$PolygonPoint(0,0), PolygonPoint(2,0), POLY_DIRICHLET, return1.0$

$PolygonPoint(2,0), PolygonPoint(1,1), POLY_DIRICHLET, return1.5$

$PolygonPoint(1,1), PolygonPoint(2,3), POLY_DIRICHLET, return0.0$
$PolygonPoint(2,3), PolygonPoint(1,4), POLY_DIRICHLET, return0.0$
$PolygonPoint(1,4), PolygonPoint(0,3), POLY_DIRICHLET, return0.8$
$PolygonPoint(0,3), PolygonPoint(0,0), POLY_DIRICHLET, return0.5$
Use grid hx=0.02, hy=0.02, dt=0.001, run 1s.
The result is shown below:

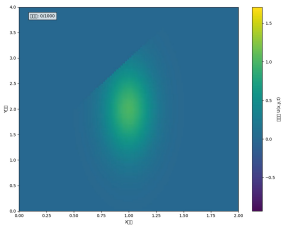

Figure 1: 0.00s

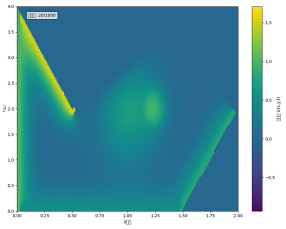

Figure 2: 0.02s

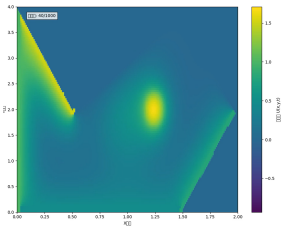

Figure 3: 0.04s

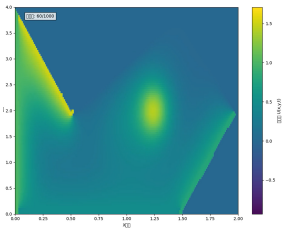

Figure 4: 0.06s

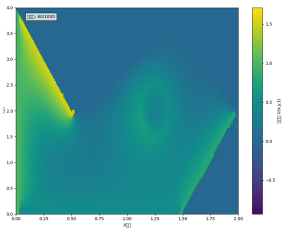

Figure 5: 0.08s

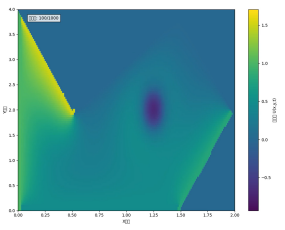

Figure 6: 0.10s

Obviously, this Parabola Equation is about heat conduction which means the boundary temperature is fixed, a heat source shakes in sine function at (1,2.5) and the initial temperature distribution is a Gauss distribution centered at (1,2).

The equation will goes as the tempreture goes stable and will have a shake region centered at (1,2.5). Based on this knowledge, the numerical solution is belivable.

ii Test Equation

We construct a Parabola Equation with a certain colsed form solution:

$$\begin{cases} u_t - (u_{xx} + u_{yy}) = e^{x+y}(cost - 2sint) & (6) \\ u(x,y,0) = 0 \ \ ((x,y) \in \bar{\Omega}) & (7) \\ u(x,y,t) = e^{x+y}sint \ \ ((x,y) \in \partial\Omega) & (8) \end{cases}$$

Its solution is:

$$u(x,y,t) = e^{x+y}sint$$

I compare the numerical solution and closed form solution by calculating L^2 Norm of their difference. L^n Norm is defined below:

$$||U||_{L^n} = (\frac{\sum_{i,j}^{Nx,Ny} |U_{i,j}|^p}{NxNy})^{\frac{1}{p}}$$

When I choose hx = 0.02, hy=0.02, dt = 0.0001. The Error is 0.0033, showing extremly low error.

Keep h=0.08, change dt, we get how error changes with dt:

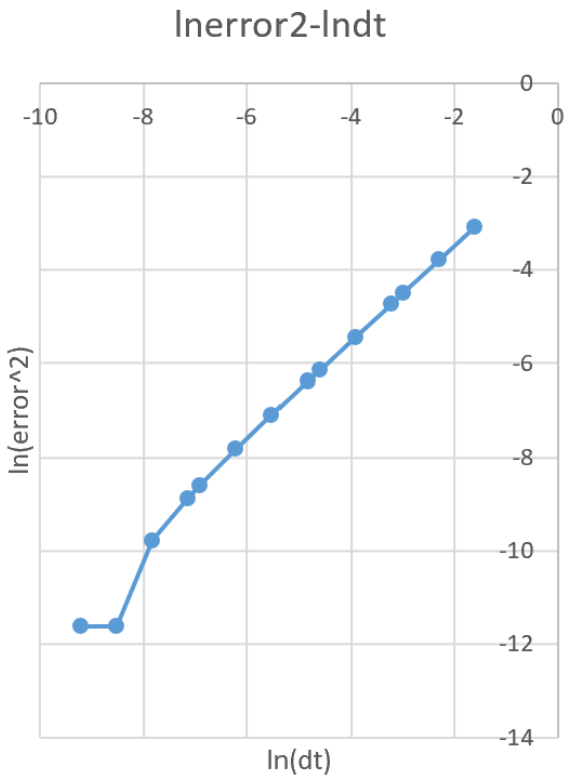

Figure 7: ln($error^2$) vs. ln(dt)

Because Gauss-Sider's tolerance is 1e-4, low error in low dt comes from Gauss-Sider residual. High part of it is about 1-power related to dt.

Keep dt=0.01, change h, we get how error changes with h:

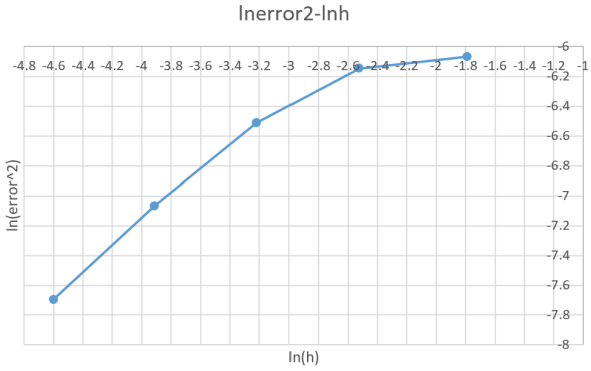

Figure 8: $\ln(error^2)$ vs. ln(h)

Low part of it is about 1-power related to h.

III Numerical Test on Explicit Scheme

i Target Equation

For the same target equation:

$$\begin{cases} u_t - (u_{xx} + u_{yy}) = 100\delta(x-1, y-2.5)sin(50t) & (9) \\ u(x,y,0) = e^{-5*((x-1)^2+(y-2)^2)} \quad ((x,y) \in \bar{\Omega}) & (10) \end{cases}$$

Boundary condition goes like:

$PolygonPoint(0,0), PolygonPoint(2,0), POLY_DIRICHLET, return1.0$
$PolygonPoint(2,0), PolygonPoint(1,1), POLY_DIRICHLET, return1.5$
$PolygonPoint(1,1), PolygonPoint(2,3), POLY_DIRICHLET, return0.0$
$PolygonPoint(2,3), PolygonPoint(1,4), POLY_DIRICHLET, return0.0$
$PolygonPoint(1,4), PolygonPoint(0,3), POLY_DIRICHLET, return0.8$
$PolygonPoint(0,3), PolygonPoint(0,0), POLY_DIRICHLET, return0.5$

Because the Explicit Scheme is not stable when space interval goes definitely small, I have to sacrifice some space accuracy. Use grid hx=0.067, hy=0.067, dt=0.001s, run 1s.

The result is shown below:

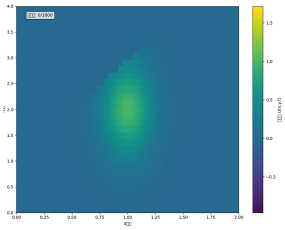

Figure 9: 0.00s

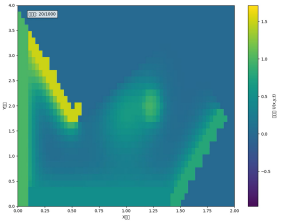

Figure 10: 0.02s

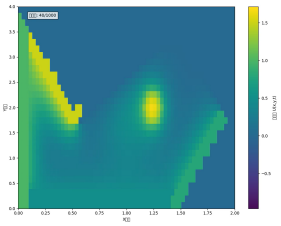

Figure 11: 0.04s

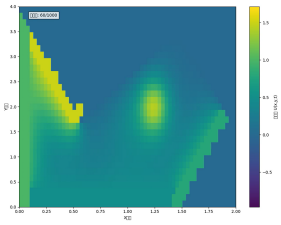

Figure 12: 0.06s

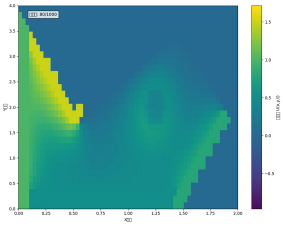

Figure 13: 0.08s

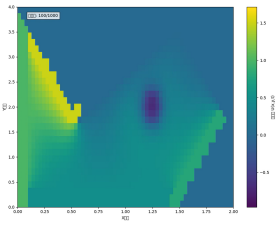

Figure 14: 0.10s

Comparing to the result of CN scheme, this result is belivable.

ii Test Equation

For the same test equation:

$$\begin{cases} u_t - (u_{xx} + u_{yy}) = e^{x+y}(cost - 2sint) & (11) \\ u(x, y, 0) = 0 \ \ ((x, y) \in \bar{\Omega}) & (12) \\ u(x, y, t) = e^{x+y} sint \ \ ((x, y) \in \partial\Omega) & (13) \end{cases}$$

Its solution is:

$$u(x, y, t) = e^{x+y} sint$$

I compare the numerical solution and closed form solution by calculating L^2 Norm of their difference. L^n Norm is defined below:

$$||U||_{L^n} = (\frac{\sum_{i,j}^{Nx,Ny} |U_{i,j}|^p}{NxNy})^{\frac{1}{p}}$$

When I choose hx = 0.02, hy=0.02, dt = 0.0001. The Error is 0.0033, showing extremly low error.

Keep h=0.08, change dt, we get how error changes with dt:

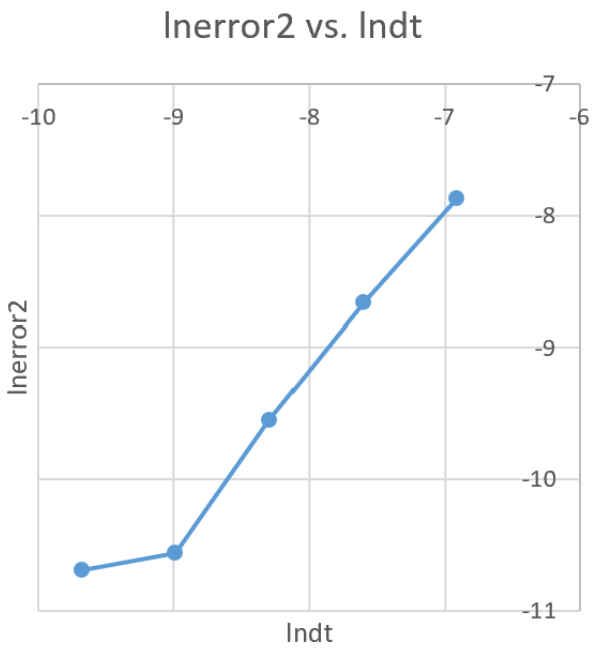

Figure 15: ln($error^2$) vs. ln(dt)

Because Explicit Scheme don't need Gauss-Sider Iteration to converge, its residual is 0. Its error seems not power related to dt.