Numerical Solutions of Differential Equations : HW1

Thomas Garrett & Weicker David

8th February 2016

Introduction

This is the introduction bitches!

1 Conservation laws

We are given the following Euler equations in $u = (\rho, \rho v, E)$,

$$\rho_t + (\rho v)_x = 0 \quad \text{(Conservation of mass)} \tag{1}$$
$$(\rho v)_t + (\rho v^2 + p)_x = 0 \quad \text{(Conservation of momentum)} \tag{2}$$
$$E_t + (v(E + p))_x = 0 \quad \text{(Conservation of energy)} \tag{3}$$

and the variable change $u = (\rho, \rho v, E) = (u_1, u_2, u_3)$. We then have

$$f(u) = (f_1, f_2, f_3)^T = (u_2, u_2^2/u_1 + p(u_1), u_2(u_3 + p(u_1))/u_1)$$

Using the u coordinates, we can write equations (1), (2), (3) as

$$u_t + f(u)_x = 0$$

From the chain rule, we can rewrite our equation as

$$u_t + \frac{\partial f}{\partial u} u_x = 0 \tag{4}$$

We now wish to find the eigenvalues and eigenvectors of the matrix

$$\frac{\partial f}{\partial u} = \begin{pmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \frac{\partial f_1}{\partial u_3} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \frac{\partial f_2}{\partial u_3} \\ \frac{\partial f_3}{\partial u_1} & \frac{\partial f_3}{\partial u_2} & \frac{\partial f_3}{\partial u_3} \end{pmatrix}$$

Using simple partial differentiation, we calculate

$$\frac{\partial f}{\partial u} = \begin{pmatrix} 0 & 1 & 0 \\ a & b & 0 \\ c & d & e \end{pmatrix}$$

with $a = -(\frac{u_2}{u_1})^2 + p'(u_1)$,
$b = 2(\frac{u_2}{u_1})$,
$c = \frac{u_2 p'(u_1)}{u_1} - \frac{u_2(u_3+p(u_1))}{u_1^2}$,
$d = \frac{u_3+p(u_1)}{u_1}$,
and $e = \frac{u_2}{u_1}$

To get the eigenvalues, we look at the roots of the characteristic polynomial of this matrix

$$\lambda^3 - e\lambda^2 - b\lambda^2 + be\lambda - a\lambda + ae = 0$$

which gives us the three eigenvalues

$$\begin{aligned}\lambda_1 &= \frac{1}{2}(b - \sqrt{4a + b^2}) \\ \lambda_2 &= \frac{1}{2}(b + \sqrt{4a + b^2}) \\ \lambda_3 &= e\end{aligned}$$

Plugging in the actual values of a, b, and e, we get

$$\begin{aligned}\lambda_1 &= \frac{u_2}{u_1} - \sqrt{p'(u_1)} \\ \lambda_2 &= \frac{u_2}{u_1} + \sqrt{p'(u_1)} \\ \lambda_3 &= \frac{u_2}{u_1}\end{aligned}$$

After computing, we get the corresponding eigenvectors

$$\begin{pmatrix} 1 \\ \frac{u_2}{u_1} - \sqrt{p'(u_1)} \\ \frac{u2\sqrt{p'(u_1)-(u_3+p(u_2)))}}{u_1} \end{pmatrix}, \quad \begin{pmatrix} 1 \\ \frac{u_2}{u_1} + \sqrt{p'(u_1)} \\ \frac{u2\sqrt{p'(u_1)+(u_3+p(u_2)))}}{u_1} \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

1.1 Linearisation

We use formal linearisation, so we first define

$$u' = \begin{pmatrix} u_1' \\ u_2' \\ u_3' \end{pmatrix} = \begin{pmatrix} U_1 + \delta v_1 \\ U_2 + \delta v_2 \\ U_3 + \delta v_3 \end{pmatrix}$$

where $U_{1,2,3}$ is a solution of the (1) and δ is assumed to be small. Inserting u' into (1), we get

$$\begin{aligned}u_t' + \frac{\partial f}{\partial u'} u_x' = 0 \rightarrow \\ (u_1')_t + (u_2')_x = 0 \\ (u_2')_t + (-(\frac{u_2'}{u_1'})^2 + p'(u_1'))(u_1)_x + 2(\frac{u_2'}{u_1'})(u_2')_x = 0 \\ (u_3')_t + (\frac{u_2' p'(u_1)}{u_1} - \frac{u_2'(u_3' + p(u_1'))}{u_1'^2})(u_1')_x + (\frac{u_3' + p(u_1')}{u_1'})(u_2')_x + \frac{u_2'}{u_1'}(u_3') = 0\end{aligned}$$

We linearise each equation separately

$$\begin{aligned}(u_2')_t + (-(\frac{u_2'}{u_1'})^2 + p'(u_1'))(u_1)_x + 2(\frac{u_2'}{u_1'})(u_2')_x = 0 \rightarrow \\ -u_2'^2(u_1)_x + u_1'^2 p'(u_1')(u_1)_x + 2(u_2' u_1')(u_2')_x = 0 \rightarrow \\ 2p'(u_1')\delta v_1 U_1 + U_2^2(\delta v_1)_x + \\ +\delta v_2(U_1(U_2)_x + U_1 U_2(U_2)_x + 2U_2(U_1)_x) + U_2 U_1(\delta v_2)_x + U_2^2(U_1)_x + U_2 U_1(U_2)_x + p'(u_1')U_1^2 O(\delta^2) = 0\end{aligned}$$

because δ is assumed to be small, we disregard the terms of $O(\delta^2)$. Dividing by δ, we get our linear equation with variable coefficients

$$a_2 v_1 + b_2 (v_1)_x + c_2 v$$

1.2 Hyperbolicity

We know that an equation of this form is hyperbolic if $\frac{\partial f}{\partial u}$ has real eigenvalues and linearly independent eigenvectors. It can be seen easily that the eigenvalues are real as long as $p'(u_1) \geq 0$. WHAT ABOUT EIGEN VECTORS? can p' = 0??

2 Heat Equation

2.1 Flux vector

2.2 Q(t)

We are now interested in computing :

$$Q(t) = \int_0^1 \int_0^1 q(x, y, t) dx dy$$

Using the fundamental theorem of calculus, we have :

$$Q(t) = Q(0) + \int_0^t Q'(\tau) d\tau$$

The initial condition gives :

$$Q(0) = \int_0^1 \int_0^1 q(x, y, 0) dx dy = \int_0^1 \int_0^1 0 \, dx dy = 0$$

We also know that (using the definition of Q) :

$$Q'(t) = \frac{d}{dt}(\int_0^1 \int_0^1 q(x, y, t) dx dy) = \int_0^1 \int_0^1 q_t(x, y, t) dx dy$$

We are now going to use the PDE to substitute q_t. This yields :

$$Q'(t) = \int_0^1 \int_0^1 q_t(x, y, t) dx dy = \int_0^1 \int_0^1 \nabla \cdot (\nabla q(x, y, t)) dx dy + \int_0^1 \int_0^1 S(x, y, t) dx dy$$

With divergence theorem, and if S means the boundary of the domain and $\mathbf{n}$ the outward unit normal to the boundary, we have :

$$Q'(t) = \oint_S \nabla q(x, y, t) \cdot \mathbf{n} \, ds + \int_0^1 \int_0^1 S(x, y, t) dx dy$$

Because the given boundary condition given is :

$$\nabla q(x, y, t) \cdot \mathbf{n} = 0$$

We finally have :

$$Q'(t) = \int_0^1 \int_0^1 S(x, y, t) dx dy$$

This yields an expression for Q as a function of t.

$$Q(t) = \int_0^t \int_0^1 \int_0^1 S(x, y, \tau) dx dy d\tau$$

Because S is a given function, this expression can always be computed as a function of t.

3 Discretization and implementation

In this section, we will implement a finite volume method to solve the problem given.

3.1 Finite Volume method

First, we need to define the method used. Let us start with the PDE and take the average over cell (i, j).

$$\frac{1}{\Delta x \Delta y} \iint_{(i,j)} q_t dxdy = \frac{1}{\Delta x \Delta y}\left(\iint_{(i,j)} \nabla \cdot (\nabla q) dxdy + \iint_{(i,j)} S\, dxdy\right)$$

Using the definition of Q_{ij}, divergence theorem and defining $S_{ij}(t) = \frac{1}{\Delta x \Delta y} \iint_{(i,j)} S\, dxdy$, we get :

$$\frac{dQ_{ij}}{dt} = \frac{1}{\Delta x \Delta y} \oint_{(i,j)} \nabla q \cdot \mathbf{n}\, ds + S_{ij}(t)$$

Because each cell is a rectangle, the first term in the right-hand side can be expressed as the sum of the integral evaluated at each side (East, North, West and South) :

$$\oint_{(i,j)} \nabla q \cdot \mathbf{n}\, ds = \int_E q_x dy + \int_N q_y dx - \int_W q_x dy - \int_S q_y dx$$

The next step is to discretize those integral. We will use finite differences. Let us assume that we are not on the boundary.

$$\begin{aligned}
\int_E q_x dy &\approx \Delta y \frac{Q_{i+1,j} - Q_{i,j}}{\Delta x} \\
\int_W q_x dy &\approx -\Delta y \frac{Q_{i-1,j} - Q_{i,j}}{\Delta x} \\
\int_N q_y dx &\approx \Delta x \frac{Q_{i,j+1} - Q_{i,j}}{\Delta y} \\
\int_S q_y dx &\approx -\Delta x \frac{Q_{i,j-1} - Q_{i,j}}{\Delta y}
\end{aligned}$$

If we are on the boundary, we use the boundary condition to have :

$$\begin{aligned}
i = 1 &\implies \int_W q_y dx = 0 \\
i = m &\implies \int_E q_y dx = 0 \\
j = 1 &\implies \int_S q_x dy = 0 \\
j = n &\implies \int_N q_x dy = 0
\end{aligned}$$

Using the finite differences above, in general, we have the following discrete equation :

$$\frac{dQ_{ij}}{dt} = \frac{aQ_{i+1,j} + bQ_{i,j} + cQ_{i-1,j}}{\Delta x^2} + \frac{dQ_{i,j+1} + eQ_{i,j} + fQ_{i,j-1}}{\Delta y^2} + S_{ij}(t)$$

Where a, b, c, d, e, f are given in the two tables below :

	a	b	c
i = 1	1	-1	0
i = 2:m-1	1	-2	1
i = m	0	-1	1

	d	e	f
j = 1	1	-1	0
j = 2:n-1	1	-2	1
j = n	0	-1	1

In order to solve this numerically, we also need to rewrite the unknown matrix Q as a vector. We do this by defining $Qvect$ as :

$$Qvect_k = Q_{i,j} \iff k = (j-1)m + i$$

So we now get a system of linear ODE's to solve (where M is a $mn \times mn$ matrix) :

$$\frac{dQvect}{dt} = MQvect + S$$

The only remaining thing to do is to build M. In order to do this let us define I the $m \times m$ identity matrix and A the following $m \times m$:

$$A = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 1 & -2 & 1 \\ 0 & \cdots & 0 & 1 & -1 \end{pmatrix}$$

We can now build B, representing the x-derivative :

$$B = \frac{1}{\Delta x^2} diag(A, n)$$

Where $diag(A, n)$ means a block diagonal matrix composed of n times the matrix A. We can check that the size of B is indeed $mn \times mn$ because A is of size $m \times m$.

We can also build C, representing the y-derivative :

$$C = \frac{1}{\Delta y^2} \begin{pmatrix} -I & I & 0 & \cdots & 0 \\ I & -2I & I & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & I & -I & I \\ 0 & \cdots & 0 & I & -I \end{pmatrix}$$

Finally :

$$M = B + C$$

3.2 Implicit Euler

Now that the problem is spatially discretized, we can start solving the system of ODE's. Let us first introduce the time step Δt and some notation :

$$Qvect^{(T)} \approx Qvect(T\Delta t)$$

$$S_k^{(T)} \approx \frac{1}{\Delta x \Delta y} \iint_{(i,j)} S(x, y, T\Delta t) dxdy$$

Because, for the second heat source, S depends on t. We also have to numerically compute the mean of the heat source on each cell. We will use the classic central point approximation.

$$S_k^{(T)} = S(x_i, y_j, T\Delta t)$$

We can now use implicit euler for the time derivative. We also have to remember that M does not depends on t and thus :

$$\frac{Qvect^{(T+1)} - Qvect^{(T)}}{\Delta t} = MQvect^{(T+1)} + S^{(T+1)}$$

If Id is the $mn \times mn$ identity matrix, we have :

$$(Id - \Delta tM)Qvect^{(T+1)} = Qvect^{(T)} + S^{(T+1)}$$

$Qvect^{(0)}$ is filled with zeroes because of the initial condition and the equation above gives the recursion to compute every $Qvect$. We can then reshape $Qvect$ to obtain the matrix Q.

Here, implicit euler is used because the problem is stiff (M has eigenvalues of very different magnitudes). Thus, an explicit method would require us to take a very little time step in order to have a stable solution. The use of an implicit method allows us to take a much bigger time step and thus to be more efficient.

3.3 Problem in matrix form

3.4 Approximation of the dirac

Remember that the discretized problem says :

$$S_k^{(T)} = S(x_i, y_j, T\Delta t)$$

But, when S is the dirac function, we cannot do that because it is impossible to evaluate the dirac function. So, we will approximate it by the given function.

3.5 Implementation

The Matlab code for the implementation of the finite volume method for this problem can be found at the end of the report.

4 Numerical results

blabla bitches

5 Refinements

blabla