Numerical Solutions of Differential Equations : HW1

THOMAS Garrett & WEICKER David

8^{th} 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

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

We compute the eigenvectors as follows

$$\begin{pmatrix} 0 & 1 & 0 \\ a & b & 0 \\ c & d & e \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \lambda_1 \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

which gives the system of equations

$$\begin{aligned}
y &= (\frac{u_2}{u_1} - \sqrt{p'(u_1)})x \\
ax + by &= (\frac{u_2}{u_1} - \sqrt{p'(u_1)})y \\
cx + dy + ez &= (\frac{u_2}{u_1} - \sqrt{p'(u_1)})z
\end{aligned}$$

We pick $x = 1$, and after solving we get

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{u_2}{u_1} - \sqrt{p'(u_1)} \\ \frac{u2\sqrt{p'(u_1)} - (u_3 + p(u_2)))}{u_1} \end{pmatrix}$$

We compute the second eigenvector similarly and get

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{u_2}{u_1} + \sqrt{p'(u_1)} \\ \frac{u2\sqrt{p'(u_1)} + (u_3 + p(u_2)))}{u_1} \end{pmatrix}$$

The last eigenvalue is computed from the system of equations

$$\begin{aligned}
y &= ex \\
ax + by &= ey \\
cx + dy + ez &= ez
\end{aligned}$$

from which we get

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

1.1 Linearisation

We have

$$u_t + \frac{\partial f}{\partial u} u_x = 0 \tag{5}$$

where $\frac{\partial f}{\partial u}$ is dependent on u. This equation immediately becomes linear if we fix $\frac{\partial f}{\partial u}$ at a specific point. We call it u_0, and we denote $\frac{\partial f}{\partial u}(u_0)$ as $\frac{\partial f}{\partial u}$ evaluated at this point. Thus, we have a linear equation

$$u_t + \frac{\partial f}{\partial u}(u_0) u_x = 0$$

which behaves like the nonlinear equation in a neighbourhood around u_0.

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 $p'(u_1) \geq 0$. We can also see that the eigenvectors are independent as long as either $p'(u_1) \neq 0$ or $u_3 + p(u_2) \neq 0$

1.3 Transport Equation

We now derive conditions on a and b for the two-dimensional transport equation

$$u_t + a(x,y)u_x + b(x,y)u_y = 0$$

to be a conservation law. we look at the basic form of a conservation law in two dimensions

$$u_t + \nabla f(u) = 0 \tag{6}$$

and we find conditions which enable us to write our transport equation in this form. We assume

$$f = \begin{pmatrix} a(x,y) & b(x,y) \end{pmatrix}$$

so plugging into (5) we get

$$u_t + \nabla(\begin{pmatrix} a(x,y) & b(x,y) \end{pmatrix} u) = u_t + (a(x,y))_x u + a(x,y)u_x + (b(x,y))_y u + b(x,y)u_y = 0$$

We now notice that if

$$(a(x,y))_x = -(b(x,y))_y$$

we get

$$u_t + \nabla(\begin{pmatrix} a(x,y) & b(x,y) \end{pmatrix} u) = u_t + a(x,y)u_x + b(x,y)u_y = 0$$

which is our two dimensional transportation equation. Thus,

$$(a(x,y))_x = -(b(x,y))_y$$

is our conservation law condition.

2 Heat Equation

2.1 Flux vector

From the course text book, 'Finite Volume Methods for Hyperbolic Problems', we know that

$$\text{flux} = -\beta q_x$$

when the heat equation is in the form

$$q_t = \beta q_{xx} + S$$

Comparing with our equation

$$q_t = \nabla \cdot (\nabla q) + S$$

we can see that $\beta = 1$. Therefore the flux vector is $-\nabla q$.

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).

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

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 $I(c)$ the $c \times c$ identity matrix and $A(c)$ the following $c \times c$:

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

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

$$B = \frac{1}{\Delta x^2} I(n) \otimes A(m)$$

Where $\otimes$ is the kronecker product. We can check that the size of B is indeed $mn \times mn$ because $A(m)$ is of size $m \times m$.

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

$$C = \frac{1}{\Delta y^2} \begin{pmatrix} -I(m) & I(m) & 0 & \cdots & 0 \\ I(m) & -2I(m) & I(m) & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & I(m) & -2I(m) & I(m) \\ 0 & \cdots & 0 & I(m) & -I(m) \end{pmatrix} = \frac{1}{\Delta y^2} A(n) \otimes I(m)$$

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 :

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

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

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 :

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

$Qvect^{(0)}$ 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

It is convenient to use the matrix form because we do not have to think about changing lines to take the boundary conditions into account and we thus avoid kronecker products.

If we use this to state the problem, we have :

$$\frac{dQ}{dt} = QT_x + T_yQ + S$$

Using implicit euler and the same notation as before, we get :

$$\frac{Q^{(T+1)} - Q^{(T)}}{\Delta t} = Q^{(T+1)}T_x + T_yQ^{(T+1)} + S$$

$$Q^{(T+1)} - \Delta t Q^{(T+1)}T_x - \Delta t T_y Q^{(T+1)} = Q^{(T)} + \Delta t S$$

This recursion equation is unfortunately not solvable because $Q^{(T+1)}$ is multiplied on the left once and on the right another time. This is why we have to write Q as a vector to solve the problem with implicit euler.

However, the matrix form allows us to prove that the finite volume method is exactly conservative. Indeed, since the walls are insulated, if the source term is 0, we should keep the same amount of heat in the domain.

$$\frac{d}{dt}\int_0^1 \int_0^1 Q dx dy = 0$$

Let us see if that is indeed the case for our method. Because Q_{ij} is defined as the average on cell (i, j), we have that :

$$\int_0^1 \int_0^1 Q dx dy = \Delta x \Delta y \sum_i \sum_j Q_{ij}$$

So we only have to prove that, in the absence of a source term, the sum of all entries does not vary in time. Since summing a matrix is multiply it by vectors filled with zeroes :

$$\frac{d}{dt}(\begin{pmatrix} 1 & \cdots & 1 \end{pmatrix} Q \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}) = \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix} \frac{dQ}{dt} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

Using our scheme (and remembering that the source term is 0...), we get :

$$\begin{pmatrix} 1 & \cdots & 1 \end{pmatrix} \frac{dQ}{dt} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix} QT_x \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} + \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix} T_y Q \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

Because of the structure of T_x and T_y, we also have :

$$T_x \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & \cdots & 1 \end{pmatrix} T_y = \begin{pmatrix} 0 & \cdots & 0 \end{pmatrix}$$

So the right hand side is equal to zero.
We can then conclude that the finite volume method is exactly conservative.

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

5 Refinements

5.1 Variable coefficients

In this part, we will change the equation a little and add variable coefficients so that the equation becomes :

$$q_t = a(y)q_{xx} + b(x)q_{yy} + S$$

Using the same principles as in section 3, we get the following :

$$\frac{1}{\Delta x \Delta y} \iint_{(i,j)} q_t dxdy = \frac{1}{\Delta x \Delta y} \iint_{(i,j)} \nabla \cdot \begin{pmatrix} a(y)q_x \\ b(x)q_y \end{pmatrix} dxdy + \frac{1}{\Delta x \Delta y} \iint_{(i,j)} S dxdy$$

$$\frac{dQ_{ij}}{dt} = \frac{1}{\Delta x \Delta y} (\int_E a(y)q_x dy + \int_N b(x)q_y dx - \int_W a(y)q_x dy - \int_S b(x)q_y dx) + S_{ij}$$

Using finite differences to evaluate q_x and q_y, we get :

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

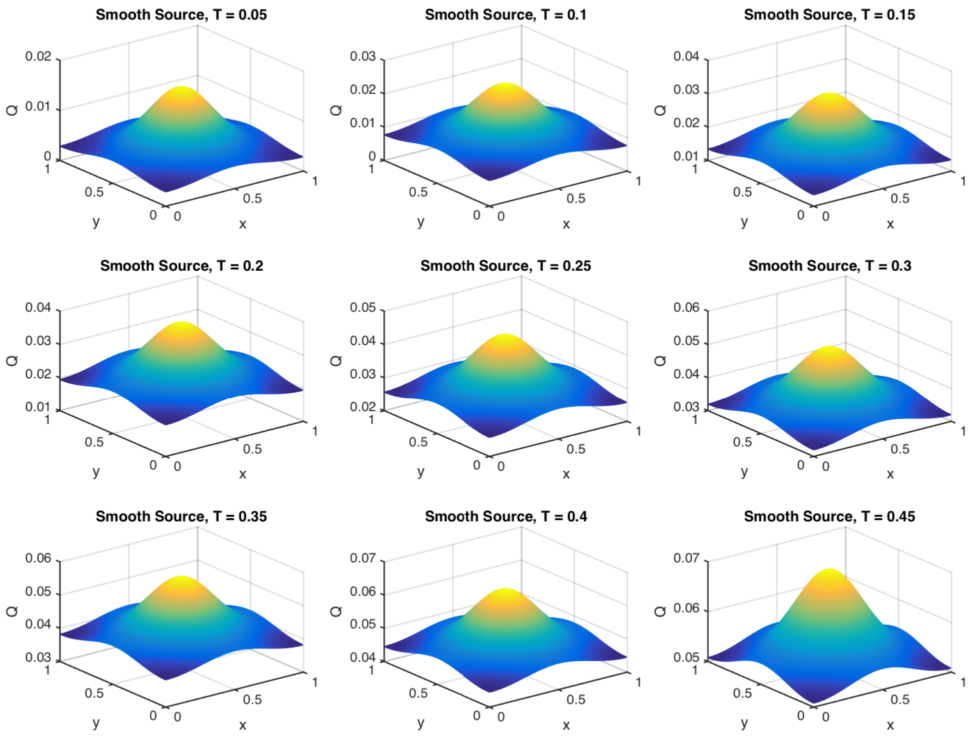

Figure 1: Digraph.

Where a, b, c, d, e, f are the same coefficients as those defined in section 3.1. E and F take into account the variable coefficient and are defined by :

$$E = \frac{1}{\Delta y}\int_E a(y)dy = \frac{1}{\Delta y}\int_W a(y)dy$$

$$F = \frac{1}{\Delta x}\int_N b(x)dx = \frac{1}{\Delta x}\int_S b(x)dx$$

So, if we use the same notations as in section 3.1, the only thing we have to do is premultiply B and C with the corresponding matrix and after that, the implementation is unchanged. We can note that if $a(y) = b(x) = 1$ then E and F are identity matrices and we have the same rule as previously.

In this problem, we have always used the smooth source term. We also have used the trapezoidal rule to evaluate the integral. Let us define $A1$ the $n \times n$ matrix as :

$$A1 = diag(0.5(a(0) + a(\Delta y)), 0.5(a(\Delta y) + a(2\Delta y)), ..., 0.5(a((n-1)\Delta y) + a(1)))$$

Which is a diagonal matrix containing the different evaluations of the integral of a. With this matrix, we can build B as :

$$B1 = (A1 \otimes I(m)) * B$$

Identically, we can define $A2$ as :

$$A2 = diag(0.5(b(0) + b(\Delta x)), 0.5(b(\Delta x) + b(2\Delta x)), ..., 0.5(b((m-1)\Delta x) + b(1)))$$

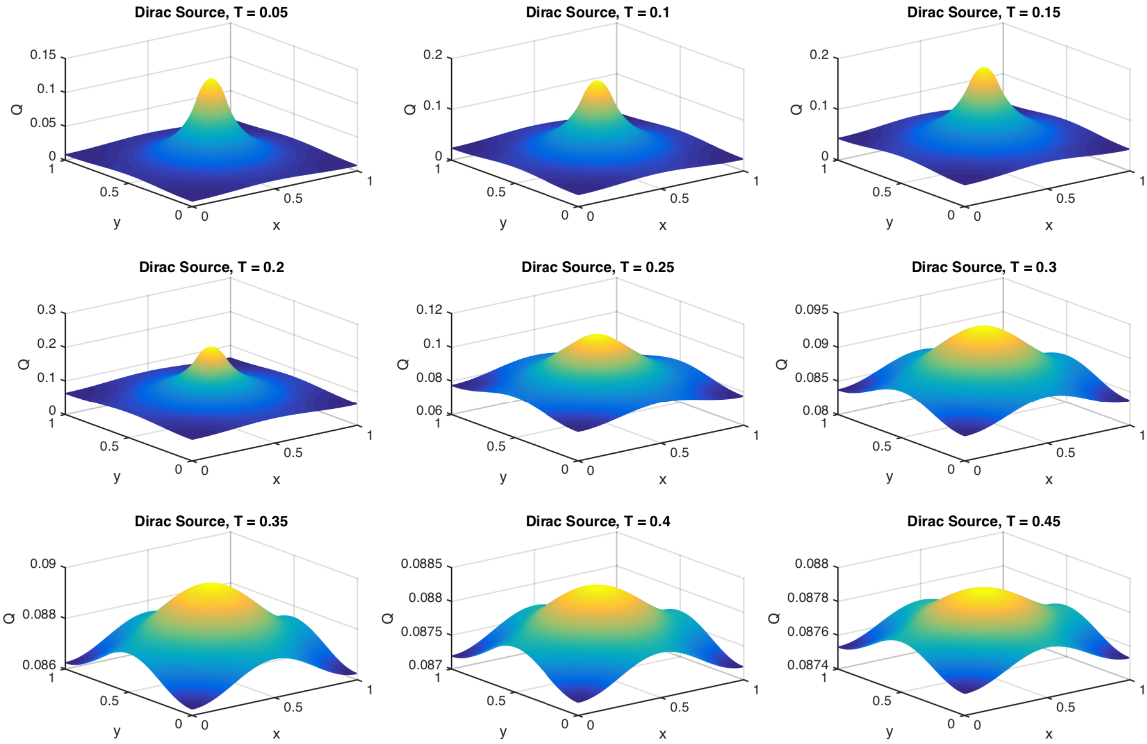

Figure 2: Digraph.

And then :

$$C1 = (I(n) \otimes A2) * C$$

And finally :

$$M = B1 + C1$$

Once we have build M, the methodology is exactly the same as before.
The implementation of the method is done in $variableCoeff.m$.

5.2 Boundary conditions

We are now going to try and change the boundary conditions to see what happens. We have a Neuman condition on the left and right boundaries (just as before except it is no longer homogeneous). And we now have a Dirichlet condition on the upper and lower boundaries. Because in the previous grid, we had no point exactly on the boundary, we are going to shift the grid of $\frac{\Delta y}{2}$ in the y-direction. That way, we will have center of cells at $y = 0$ and $y = 1$. If we still keep $n\Delta y = 1$, that means that we will now have $m(n-1)$ unknowns since the value on the upper and lower boundaries are known.

We use the same methodology as before. We rewrite Q as $Qvect$ and we will show that we will get a system of ODE's of the following type :

$$\frac{dQvect}{dt} = A_x Qvect + A_y Qvect + b_x + b_y + S$$

We already know the definition of S, it is unchanged. A_x and b_x represents the derivative in the x-direction (b_x is not zero because of the non homogeneous boundary conditions). Respectively, A_y and b_y take the y-derivative into account.

The x-derivative operator does not change but an additional term appears. Indeed, for a cell that is on the right boundary, we no longer no flux on the East edge but instead :

$$\int_E q_x dy = \int_E (-1) dy = -\Delta y$$

In the same way, for a cell on the left boundary, we have the following West flux :

$$-\int_W q_x dy = -\int_W (-1) dy = \Delta y$$

So, if we define T_x as the following $m \times m$ matrix :

$$T_x = \frac{1}{\Delta x^2} \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 1 & -2 & 1 \\ 0 & \cdots & 0 & 1 & -1 \end{pmatrix}$$

We can use kronecker products to express the x-derivative :

$$A_x = I(n-1) \otimes T_x$$

Where $I(n-1)$ is the $(n-1) \times (n-1)$ identity matrix. We have to take the non homogeneous conditions into account and so :

$$b_x = ones(n-1, 1) \otimes \begin{pmatrix} -\frac{1}{\Delta x} \\ 0 \\ \vdots \\ 0 \\ \frac{1}{\Delta x} \end{pmatrix}$$

The work for the y-derivative is quite similar. But because we are only solving for the interior cells, we have "ghost cells" on the upper and lower boundary. This means that the $(n-1) \times (n-1)$ matrix T_y changes a bit :

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

And A_y becomes :

$$A_y = T_y \otimes I(m)$$

To take the Dirichtlet conditions into account we have to use b_y defined by :

$$b_y = \begin{pmatrix} q(x_1, 0, t) & \cdots & q(x_m, 0, t) & 0 & \cdots & 0 & q(x_1, 1, t) & \cdots & q(x_m, 1, t) \end{pmatrix}^T$$

Where x_i means the value of x in the center of cell $(i, 1)$ or $(i, n-1)$.

Finally, we can define $M = A_x + A_y$ and $S_{new} = S + b_x + b_y$, we get a system of the form :

$$\frac{dQvect}{dt} = MQvect + S_{new}$$

This system is of the same form as in section 3 so we can use the same method to implement implicit Euler.