Derivation of a Numerical Method for solving Hyperbolic PDEs

I. FORMULATION.

Practically speaking, hyperbolic partial differential equations have a second partial derivatives in both space and time.

$$\frac{\partial^2 T}{\partial t^2} = K\left(x, t, T, \nabla T, \nabla^2 T\right) \tag{I.1}$$

If we look at this, we should be reminded of how we solved a second-order ODE of the form:

$$\frac{d^2 T}{dt^2} = K\left(x, t, T\right) \tag{I.2}$$

The way that we handled this was to reduce the second-order ODE into two first-order ODEs. Since we already had a procedure for solving a system of first-order ODEs, once we accomplished the reduction, we could solve the problem. As a reminder, we wrote equation (I.2) as

$$\frac{dT^{(1)}}{dt} = K^{(1)}\left(x, t, T^{(1)}\right) = T^{(2)} \tag{I.3a}$$

$$\frac{dT^{(2)}}{dt} = K^{(2)}\left(x, t, T^{(1)}\right) = K\left(x, t, T^{(1)}\right) \tag{I.3b}$$

In this formulation our new function $T^{(1)}$ is our original function T and our new function $T^{(2)}$ is the first derivative with respect to time of original function T.

We repeat the same procedure for the hyperbolic PDE in equation (I.1) to obtain

$$\frac{\partial T^{(1)}}{\partial t} = K^{(1)}\left(x, t, T^{(1)}, \nabla T^{(1)}, \nabla^2 T^{(1)}\right) = T^{(2)} \tag{I.4a}$$

$$\frac{\partial T^{(2)}}{\partial t} = K^{(2)}\left(x, t, T^{(1)}, \nabla T^{(1)}, \nabla^2 T^{(1)}\right) = K\left(x, t, T^{(1)}, \nabla T^{(1)}, \nabla^2 T^{(1)}\right) \tag{I.4b}$$

Again, our new function $T^{(1)}$ is our original function T and our new function $T^{(2)}$ is the first partial derivative with respect to time of original function T.

These two first order PDEs are parabolic. We already know how to solve a system of parabolic PDEs. Thus, solving a single hyperbolic PDE is equivalent to solving a system of 2 parabolic PDEs. We already know how do this.

II. Linear versus Non-Linear hyperbolic PDEs

The algorithm that we outlined previously for solving a system of parabolic PDEs was for the most general non-linear form, (which would work for linear forms). However, we could have derived an algorithm specific to a system of linear equations, had we so desired, following the same outline as that which we used to develop the linear algorithm for a single linear parabolic PDE. Since, hyperbolic PDEs are just a subset of systems of parabolic PDEs, the same argument applies. Linear hyperbolic PDEs could be solved using the Crank-Nicolson method applied to systems of parabolic PDEs. Non-linear hyperbolic PDEs can be solved using the Runge-Kutta method in conjunction with finite-difference formulae for the spatial derivatives.

III. Systems of hyperbolic PDEs

We have said above that a single hyperbolic PDE can be reduced to a system of 2 parabolic PDEs. Therefore a system of n_u hyperbolic PDEs can be reduced to a system of $2n_u$ parabolic PDEs. In our algorithm for the solution of a system of PDEs, it doesn't matter whether we have 2 or $2n_u$ equations. We already know how to solve this problem.

IV. The Gory Details

There are some nuances that arise in the solution of a hyperbolic equation as a system of 2 parabolic PDEs. We investigate these nuances by considering the following PDE:

$$\frac{\partial^2 U}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 U}{\partial t^2}$$

Consider a spatially one-dimensional problem where x ranges from 0 to L. The hyperbolic problem generally requires two initial conditions (one for each order of the time derivative). Sample initial conditions are given below.

$$U(x, t = 0) = a\left(x^2 - Lx\right)$$

$$\frac{dU}{dt}(x, t = 0) = 0.0$$

The hyperbolic PDE generally provides two boundary conditions. In the case of a string with each end fixed, these boundary conditions have the form:

$$U(x = 0, t) = 0.0$$

$$U(x = L, t) = 0.0$$

Now, let's transform this hyperbolic PDE to a system of 2 parabolic PDEs.

Let $y^{(1)} = U$ and $y^{(2)} = \frac{\partial U}{\partial t}$. We resort to using subscripts in parentheses because subscripts will later denote position and superscripts time. So our two parabolic PDEs are

$$\frac{\partial y^{(1)}}{\partial t} = y^{(2)} \qquad\qquad \frac{\partial y^{(2)}}{\partial t} = c^2 \frac{\partial^2 y^{(1)}}{\partial x^2}$$

with the initial conditions

$$y^{(1)}(x, t = 0) = a\left(x^2 - Lx\right) \qquad\qquad y^{(2)}(x, t = 0) = 0.0$$

and the boundary conditions:

$$y^{(1)}(x = 0, t) = 0.0 \qquad\qquad y^{(1)}(x = L, t) = 0.0$$

We should immediately see the problem. The generic form of the hyperbolic PDE does not give us the boundary conditions for $y^{(2)} = \frac{\partial U}{\partial t}$. This is a problem since, in order to solve the equations as a system of 2 parabolic PDEs, we require boundary conditions on each variable. What to do? What to do?

Well, the solution is simple. Since $y^{(2)} \equiv \frac{\partial y^{(1)}}{\partial t}$, this ought to hold even at the boundary conditions. Therefore, missing boundary conditions on $y^{(2)}$ are the time derivatives of the boundary conditions on $y^{(1)}$. In the case of Dirichlet boundary conditions, the boundary conditions on $y^{(1)}$ will be of the form

$$y^{(1)}(x = 0, t) = f_1(t) \qquad\qquad y^{(1)}(x = L, t) = f_2(t)$$

so the boundary conditions on $y^{(2)}$ will be of the form

$$y^{(2)}(x = 0, t) = \frac{\partial f_1(t)}{\partial t} \qquad\qquad y^{(2)}(x = L, t) = \frac{\partial f_2(t)}{\partial t} .$$

In the case of Neumann boundary conditions, the boundary conditions on $y^{(1)}$ will be of a more general form. Two examples are given below.

$$\left.\frac{\partial y^{(1)}}{\partial x}\right|_{x=0,t} = f_1(t) \qquad\qquad \left.\frac{\partial y^{(1)}}{\partial x}\right|_{x=L,t} = f_2(t) y^{(1)}$$

Since $y^{(2)} \equiv \frac{\partial y^{(1)}}{\partial t}$ the boundary conditions on $y^{(2)}$ will be of the form

$$\frac{\partial}{\partial t}\left(\left.\frac{\partial y^{(1)}}{\partial x}\right|_{x=0,t}\right) = \frac{\partial f_1(t)}{\partial t} \qquad \frac{\partial}{\partial t}\left(\left.\frac{\partial y^{(1)}}{\partial x}\right|_{x=L,t}\right) = \frac{\partial\left(f_2(t)y^{(1)}\right)}{\partial t}$$

which can be manipulated in the following manner.

$$\left.\frac{\partial}{\partial x}\left(\frac{\partial y^{(1)}}{\partial t}\right)\right|_{x=0,t} = \frac{\partial f_1(t)}{\partial t} \qquad \left.\frac{\partial}{\partial x}\left(\frac{\partial y^{(1)}}{\partial t}\right)\right|_{x=L,t} = f_2(t)\frac{\partial y^{(1)}}{\partial t} + y^{(1)}\frac{\partial f_2(t)}{\partial t}$$

$$\left.\frac{\partial y^{(2)}}{\partial x}\right|_{x=0,t} = \frac{\partial f_1(t)}{\partial t} \qquad \left.\frac{\partial y^{(2)}}{\partial x}\right|_{x=L,t} = f_2(t)y^{(2)} + y^{(1)}\frac{\partial f_2(t)}{\partial t}$$

So that we now have commensurate Neumann boundary conditions on $y^{(2)}$.

In the simple example, we were working out above, the boundary conditions on $y^{(2)}$ are:

$$y^{(2)}(x=0,t) = 0.0 \qquad y^{(2)}(x=L,t) = 0.0$$

An Example:

PDE:

$$\frac{\partial^2 U}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 U}{\partial t^2}$$

Initial conditions:

$$U(x,t=0) = a \cdot \sin\left(\frac{2\pi x}{L}\right)$$

$$\frac{dU}{dt}(x,t=0) = 0.0$$

Boundary conditions:

$$U(x=0,t) = 0.0$$

$$U(x=L,t) = 0.0$$

parameters: c = 2, a = 0.01, L = 2

Solve from t = 0 to t = 5. Show position and velocity at a few times. We used 40 spatial intervals and 1000 temporal intervals.

Using the code to solve systems of parabolic PDEs, we must change the relevant functions. Below are the initial condition and boundary condition functions, located at the bottom of the file syspde_para.m

```
%
%  BOUNDARY CONDITION FUNCTION
%
function y_out = syspde_boundary_conditions(k_eq,k_bc,x,t,y1,i_x, dx);
if (k_bc == 1)
    if (k_eq == 1)
    y_out = 0;
    elseif (k_eq == 2)
        y_out = 0;
   end
else
    if (k_eq == 1)
    y_out = 0;
    elseif (k_eq == 2)
        y_out = 0;
   end
end

%
%  INITIAL CONDITION FUNCTION
%
function y_out = syspde_initial_conditions(k_eq,x);
global xo xf
if (k_eq == 1)
   a = 0.01;
   L = xf - xo;
   y_out = a*sin(2*pi*x/L);
elseif (k_eq == 2)
    y_out = 0;
end
```

Below is the PDE input, located in the file syspde_para_input.m

```
function dydt = syspde_para_input(n_eq,i_x_v,y1,t,dxi,mx)
% evaluate  spatial derivatives
for k_eq = 1:1:1
   for i_x = i_x_v(k_eq,1):1:i_x_v(k_eq,2)
    %dydx_matrix(k_eq, i_x) = 0.5*( y1(k_eq,i_x+1) - y1(k_eq,i_x-1) )*dxi;
      d2ydx2_matrix(k_eq,i_x) = ( y1(k_eq,i_x+1) - 2.0*y1(k_eq,i_x) + y1(k_eq,i_x-1) )*dxi^2;
   end
end
%
c = 2;
c2 = c^2;
%  evaluate temporal derivative for all equations at interior nodes
dydt = zeros(n_eq,mx);
% for k_eq = 1 (position)
k_eq = 1;
   for i_x = i_x_v(k_eq,1):1:i_x_v(k_eq,2)
      dydt(k_eq,i_x) = y1(2,i_x);
   end
% for k_eq = 2 (velocity)
k_eq = 2;
   for i_x = i_x_v(k_eq,1):1:i_x_v(k_eq,2)
      dydt(k_eq,i_x) = c2*d2ydx2_matrix(1,i_x);
   end
```

Here are some plots of the position (black) and velocity (red) at different times.