# Parabolic PDEs

# Parabolic PDEs in Science and Engineering

- Several parabolic PDEs are important in science and engineering  By far the most important is the diffusion equation

$$\frac{\partial \varphi}{\partial t} - \vec{\nabla} \cdot \left( D \nabla \varphi \right) = 0$$

- Where ϕ is the density and D is the diffusion coefficient.  In general, D can be a function of x and y.

# Heat Equation

- If D is constant, we obtain the classical heat equation

$$\frac{\partial \varphi}{\partial t} - D\nabla^2 \varphi = 0$$

- We would usually write

$$D = k / c\rho$$

- Where c is the specific heat capacity, k is the diffusion coefficient, and ρ is the density.

# Explicit Methods FTCS

- The most straightforward attack is to write a finite-difference approximation. Choose a centered difference for the spatial derivative and a forward difference for the time derivative.
  - Why use a forward time-derivative and not a centered difference?
  - It requires only one time level of information (and we have only one initial condition)

# FTCS for Diffusion

- Plugging into our difference equations gives

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} + O(\Delta x^2)$$

$$\frac{\partial u}{\partial t} = \frac{u_i^{n+1} - u_i^n}{\Delta t} + O(\Delta t)$$

- Or, after some algebra,

$$u_i^{n+1} = \frac{k\Delta t}{c\rho(\Delta x)^2}(u_{i+1}^n + u_{i-1}^n) + \left(1 - \frac{2k\Delta t}{c\rho(\Delta x)^2}\right)u_i^n$$

# FTCS for Diffusion (continued)

- The forward-time centered-space method is an Euler method in time (first order) and a centered-derivative method (second order) in space.
- It is *conditionally stable*. It can be shown that this method is stable as long as (for one spatial dimension):

$$\frac{k\Delta t}{c\rho(\Delta x)^2} \le \frac{1}{2}$$

# Example

- A simple one-dimensional diffusion equation was coded up with a "hat" function for the initial conditions. The first run was for

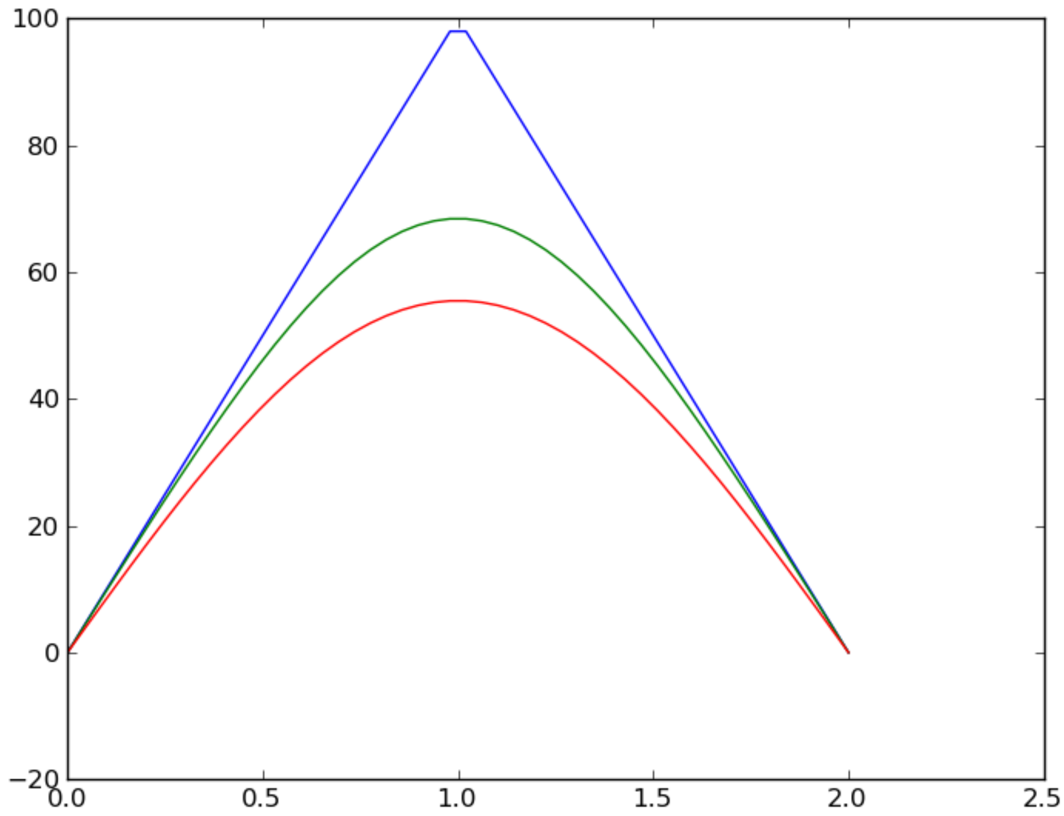
$$\Delta t = 0.75 * \frac{c\rho\Delta x^2}{2k}$$

The second was for

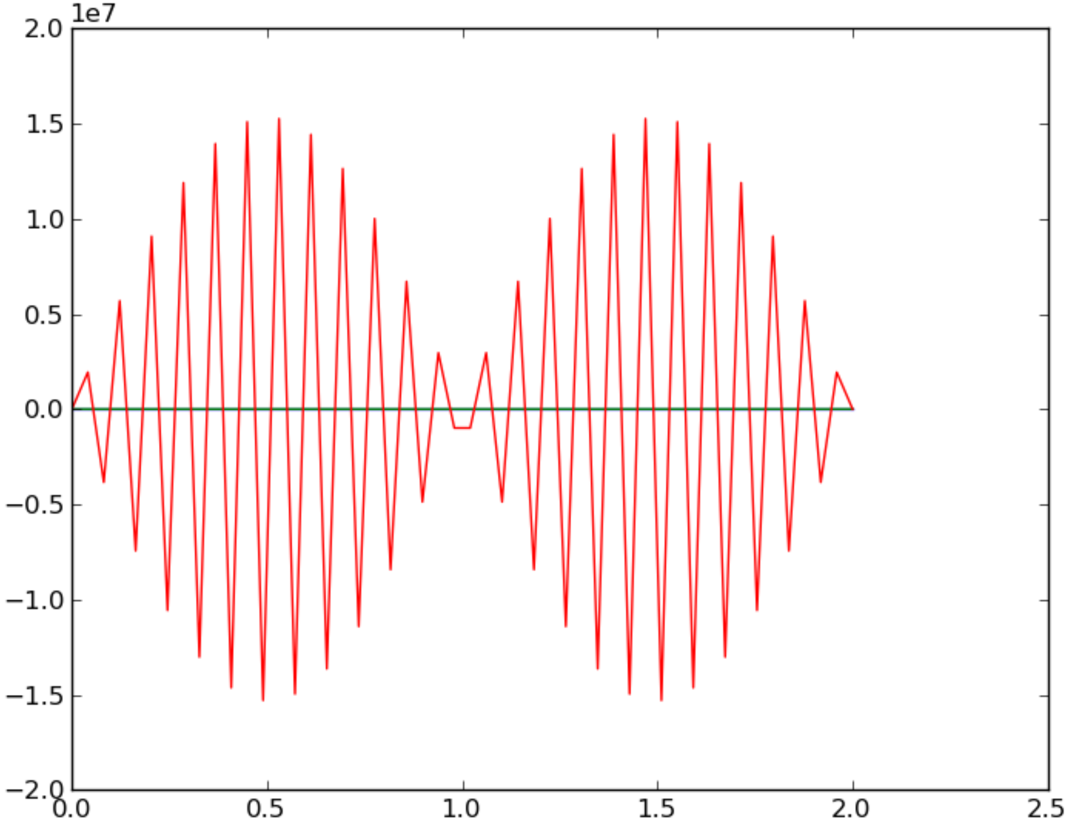
$$\Delta t = 1.05 * \frac{c\rho\Delta x^2}{2k}$$

Both were run for 250 timesteps with Δx=0.04 (this is quite large compared to "realistic" values).

# FTCS Stable



# FTCS Unstable



# Crank-Nicolson Method

- Suppose that we take the “forward difference” to be a central difference centered at time

  $t^{n+1/2}$

  We must average the spatial difference in time:

$$\frac{1}{2}\left(\frac{u_{i+1}^{n+1}-2u_i^{n+1}+u_{i-1}^{n+1}}{\Delta x^2}+\frac{u_{i+1}^{n}-2u_i^{n}+u_{i-1}^{n}}{\Delta x^2}\right)=\frac{c\rho}{k}\left(\frac{u_i^{n+1}-u_i^n}{\Delta t}\right)$$

- Then let us define

  $r=k\Delta t/c\rho(\Delta x)^2$

- After rearranging this we obtain

$$-ru_{i-1}^{n+1}+(2+2r)u_i^{n+1}-ru_{i+1}^{n+1}=ru_{i-1}^{n}+(2-2r)u_i^{n}+ru_{i+1}^{n}$$

# Crank-Nicolson (Continued)

- The Crank-Nicolson method is *implicit*. We must solve a matrix equation at each timestep.
- C-N is *unconditionally stable*. Thus the timestep is not limited as it is for the explicit FTCS method. This can be a significant advantage. However, the explicit method is much faster per timestep, so if small timesteps are required for other reasons then C-N can be very slow.

# Higher Dimensions

- The explicit method works for any number of dimensions but is limited by the stability condition, which in two dimensions becomes
$$\frac{k\Delta t}{c\rho\left[\left(\Delta x\right)^2+\left(\Delta y\right)^2\right]}\le\frac{1}{8}$$
- The Crank-Nicolson method works for one dimension—how can we extend it to higher dimensions?
- Note: in PDEs the number of dimensions always refers to the spatial dimensions – time doesn’t count.

# Alternating-Direction Explicit
## (Saul'yev, Barakat and Clark,MacCormick)

- Another approach is the ADE method, which utilizes the fact that we can integrate convert a semi-implicit method to an explicit computation if we merely update in the correct directions.
- This gives two equations (1-D example)

$$(1+r)u_i^{n+1} = u_i^n + r\left(u_{i-1}^n - u_i^n + u_{i+1}^n\right)\ldots Saul'yev\ A$$

$$(1+r)u_i^{n+1} = u_i^n + r\left(u_{i+1}^{n+1} - u_i^n - u_{i-1}^n\right)\ldots Saul'yev\ B$$

# ADE (Continued)

- This is equivalent to introducing two variables P and Q such that

$$(1+r)P_i^{n+1} = u_i^n + r\left(P_{i-1}^{n+1} - u_i^n + u_{i+1}^n\right)$$

$$(1+r)Q_i^{n+1} = u_i^n + r\left(Q_{i+1}^{n+1} - u_i^n + u_{i-1}^n\right)$$

We then average to get our final updated values

$$u_i^{n+1} = \frac{1}{2}\left(P_i^{n+1} + Q_i^{n+1}\right)$$

This method is stable.

# Alternating-Direction Implicit (ADI)

- The ADI method is derived similarly to the C-N method; we take central differences in space and average the differences in time. This results in a set of MxN simultaneous equations (M in x direction, N in y direction). A clever trick is to split it into two directions, so that we do the implicit solution first in the x direction and then in the y direction.

# ADI (continued)

- Specifically, we take a first step

$$u_{i,j}^{n+1/2} - u_{i,j}^{n} = r\left[u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}\right] + r\left[u_{i,j+1}^{n+1/2} - 2u_{i,j}^{n+1/2} + u_{i,j-1}^{n+1/2}\right]$$

Followed by

$$u_{i,j}^{n+1} - u_{i,j}^{n+1/2} = r\left[u_{i+1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i-1,j}^{n+1}\right] + r\left[u_{i,j+1}^{n+1/2} - 2u_{i,j}^{n+1/2} + u_{i,j-1}^{n+1/2}\right]$$

# ADE versus ADI

- ADE is evaluated like an explicit method so it is very fast, but it does require a small timestep for accuracy.
- ADI has a smaller truncation error at the same step size but requires solving tridiagonal matrices at each timestep, so it is slower.
- If you need to use a small timestep for other reasons, ADE is a good method, otherwise use ADI.

# Method of Lines

- The method of lines discretizes only the spatial dimensions, thus reducing the PDE to an initial-value ODE.  Often we will use a centered difference.
- If we use the Euler method for the IVP solver then we get back the FTCS method.  We know this is only conditionally stable, so we might use better methods for the time integration and achieve a better result.