SE289

Scientific Computation with PDEs

Atanu Mohanty

SERC, I.I.Sc., Bangalore

Topics

- Simple finite difference methods for parabolic and hyperbolic PDEs
- Analysis of stability and accuracy
- Programs and examples for simple problems
- Demonstration of the difficulties presented by Hyperbolic PDEs

Objectives

- All equations presented here have known analytical solutions.

- We apply commonly used finite difference schemes to get numerical solutions which are then compared with the known exact solutions.

- One objective is to see how good or bad these numerical methods are.

- Another objective is to understand the stability analysis of these methods.

Examples of PDEs

- Laplace Equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

- Heat or Diffusion Equation:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

- Wave Equation:

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

- Advection Equation:

$$\frac{\partial u}{\partial t} = -c \frac{\partial u}{\partial x}$$

Discretization Notation

We shall discuss the parabolic heat equation and the hyperbolic advection equation in one space dimension (x) only. In what follows we shall replace various partial derivatives by differences taken on rectangular grid in the x-t plane.

- Time step will be written as Δt.

- The index in the t direction will be denoted n and will be written as a superscript.

- Space step will be written as Δx.

- The index in the x direction will be denoted m and will be written as a subscript.

So $u(a + m\Delta x, n\Delta t)$ will be written as u_m^n.

Example Heat Conduction Problem

A metal rod of length L, whose ends are kept at a fixed low temperature, is heated at its centre for a long time. The heating is stopped at $t = 0$. How does the temperature evolve for $t > 0$?

Solve

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

for $t > 0$, subject to conditions,

$$u(x, 0) = 1 - |1 - 2x/L|,$$

for $0 \leq x \leq L$, and

$$u(0, t) = u(L, t) = 0,$$

for $t > 0$.

Fourier Series Solution:

$$u(x, t) =$$

$$\frac{8}{\pi^2} \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)^2} e^{-\frac{(2k+1)^2 \pi^2 \alpha t}{L^2}} \sin \frac{(2k+1)\pi x}{L}$$

FTCS Scheme for the Heat Equation

- Heat Equation:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

- Forward in Time, Centered in Space

- $\frac{\partial u}{\partial t} \approx \frac{u_m^{n+1} - u_m^n}{\Delta t}$

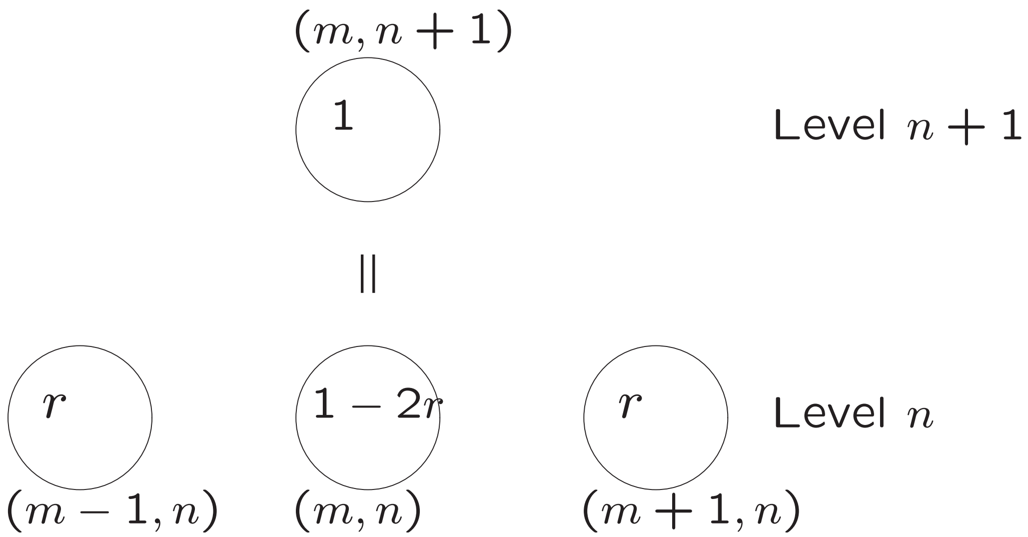
- $\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{m+1}^n - 2u_m^n + u_{m-1}^n}{(\Delta x)^2}$

- Scheme:

$$u_m^{n+1} = (1 - 2r)u_m^n + r(u_{m-1}^n + u_{m+1}^n)$$

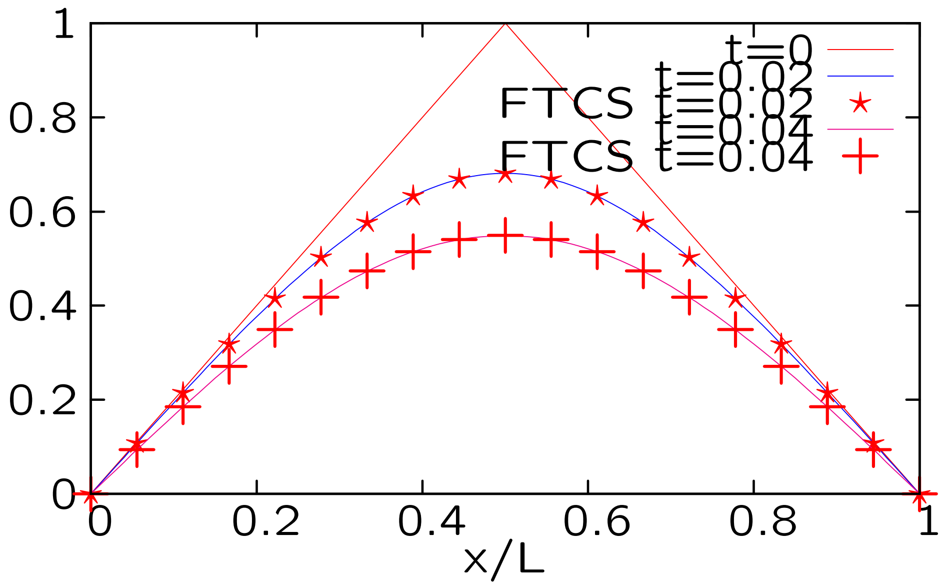
- where, $r = \dfrac{\alpha \Delta t}{(\Delta x)^2}$

FTCS Computational Molecule



- $r = \dfrac{\alpha \Delta t}{(\Delta x)^2}$

Heat Equation FTCS Scheme Results

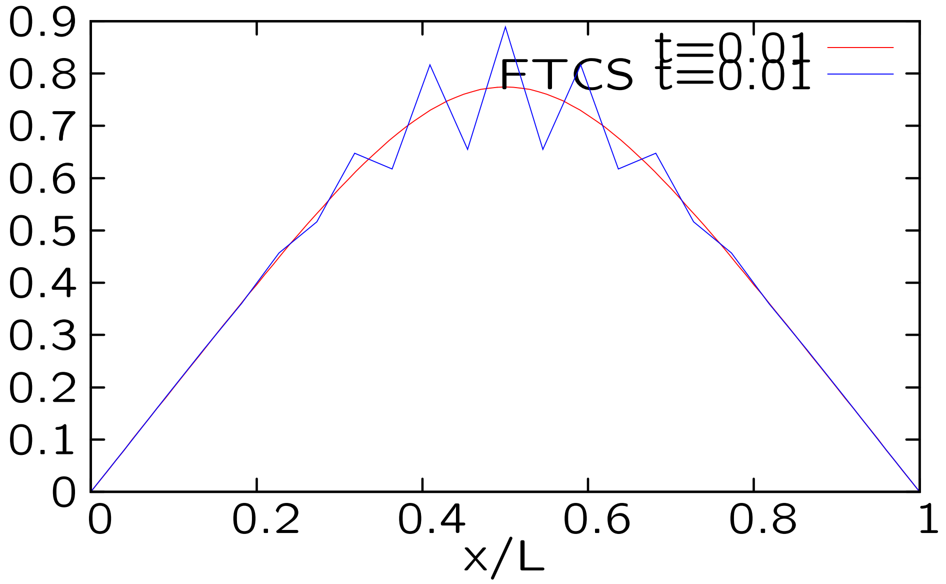


- $\Delta t = \dfrac{1}{800}\dfrac{L^2}{\alpha}, \Delta x = \dfrac{L}{18}$

- $r = \dfrac{\alpha \Delta t}{(\Delta x)^2} = 0.405$

- Note: The time values are normalized with respect to L^2/α.

FTCS with smaller Δx



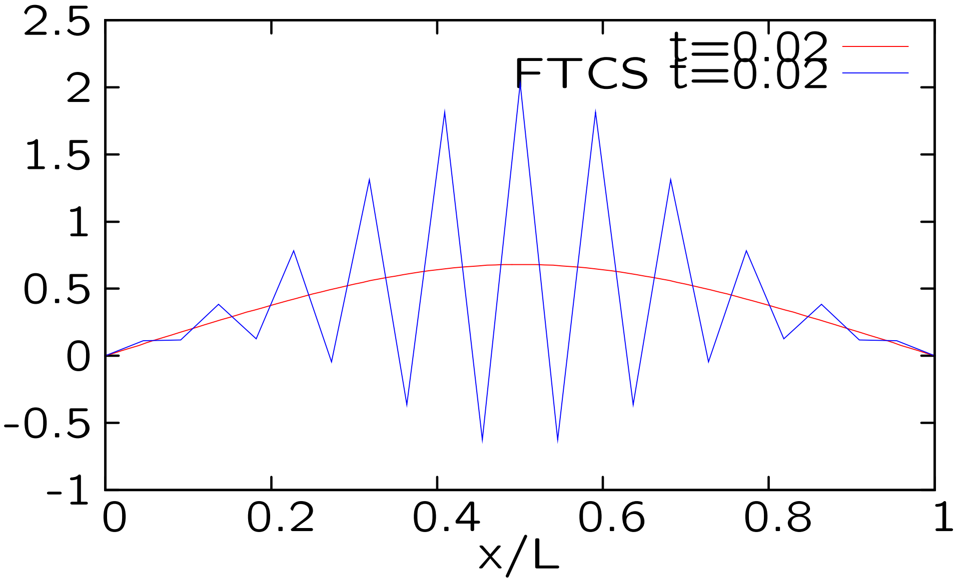
- $\Delta t = \dfrac{1}{800}\dfrac{L^2}{\alpha}, \Delta x = \dfrac{L}{22}$

- $r = \dfrac{\alpha \Delta t}{(\Delta x)^2} = 0.605$

- Should the result not be more accurate?

- What went wrong!

FTCS with smaller Δx



- $\Delta t = \dfrac{1}{800}\dfrac{L^2}{\alpha}, \Delta x = \dfrac{L}{22}$

- $r = \dfrac{\alpha \Delta t}{(\Delta x)^2} = 0.605$

- Unstable!

- Stability analysis needed.

Stability Analysis (von Neumann)

The exponentials are the eigenfunctions of all linear difference operators. Assuming $u(x,t) = U(t)e^{ikx}$ we get,

$$U_m^{n+1} = AU_m^n,$$

where,

$$A = 1 - 2r + 2r\cos(k\Delta x))$$

$$= 1 - 4r\sin^2(k\Delta x/2).$$

We know that spatial variations are smoothed by the diffusion process, so for stability we need $|A|$ to be less than 1. Assuming that in the worst case the sine square term can become unity, for $|A|$ to be less than 1 we must have $4r < 2$, or

$$r < \frac{1}{2}.$$

This type of stability analysis is called *von Neumann stability analysis*. It looks at what happens to waves for various wavelengths.

Limitations of von Neumann Analysis

Strictly speaking von Neumann stability analysis is valid only for unbounded domains, since it does not take boundary conditions into account. For a bounded domain like a rod, a matrix eigenvalue analysis will give exact results. But it is the short wavelength modes which cause more trouble, so von Neumann analysis is usually good enough.

Consequences of $r < 1/2$

Since $r = \dfrac{\alpha \Delta t}{(\Delta x)^2}$ needs to remain below a critical value for stability, doubling spatial resolution(halving Δx) requires a simultaneous reduction in time step by a factor of four. This is a very unfortunate situation.

Stability is poor for explicit schemes such as the FTCS scheme. One can overcome these instabilities by using *implicit* schemes. This we know from ODE courses where it is shown that the implicit trapezoidal Euler method is more stable and accurate than the explicit Euler forward method. The PDE version of the trapezoidal method is known as the Crank-Nicolson method. It is more complex than FTCS, but it is stable for all step sizes.

The Crank-Nicolson Method

In this method $\frac{\partial^2 u}{\partial x^2}$ is taken as the average of

$$\frac{u_{m+1}^{n} - 2u_{m}^{n} + u_{m-1}^{n}}{(\Delta x)^2}$$

and

$$\frac{u_{m+1}^{n+1} - 2u_{m}^{n+1} + u_{m-1}^{n+1}}{(\Delta x)^2}.$$

Then we get

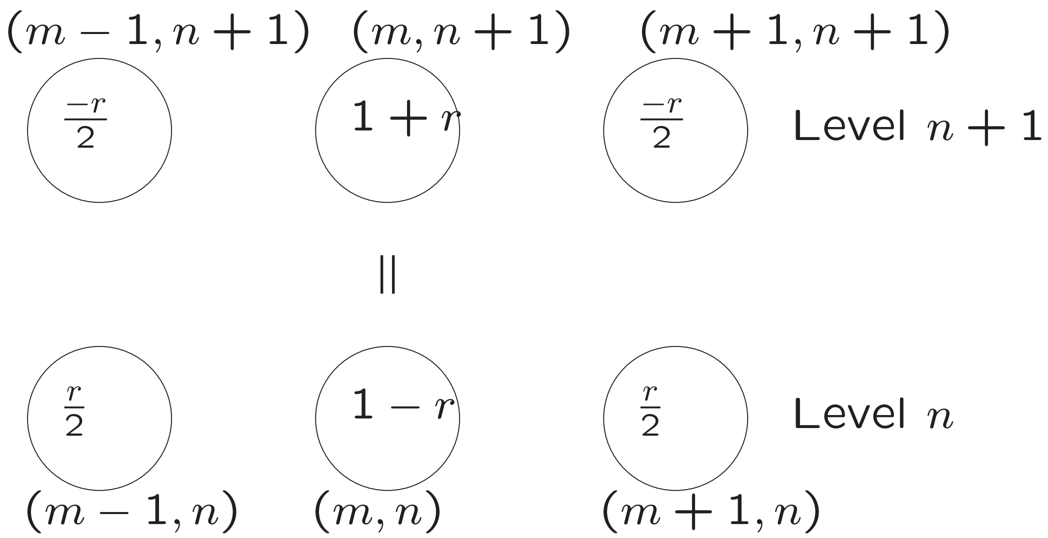
$$(1+r)u_m^{n+1} - \frac{r}{2}(u_{m-1}^{n+1} + u_{m+1}^{n+1})$$

$$= (1-r)u_m^{n} + \frac{r}{2}(u_{m-1}^{n} + u_{m+1}^{n}).$$

where $r = \frac{\alpha \Delta t}{(\Delta x)^2}$.

This is an implicit scheme. To find the u values at level $n+1$, we must solve a tridiagonal matrix equation at each step.

Crank-Nicolson Computational Molecule



Crank-Nicolson Stability

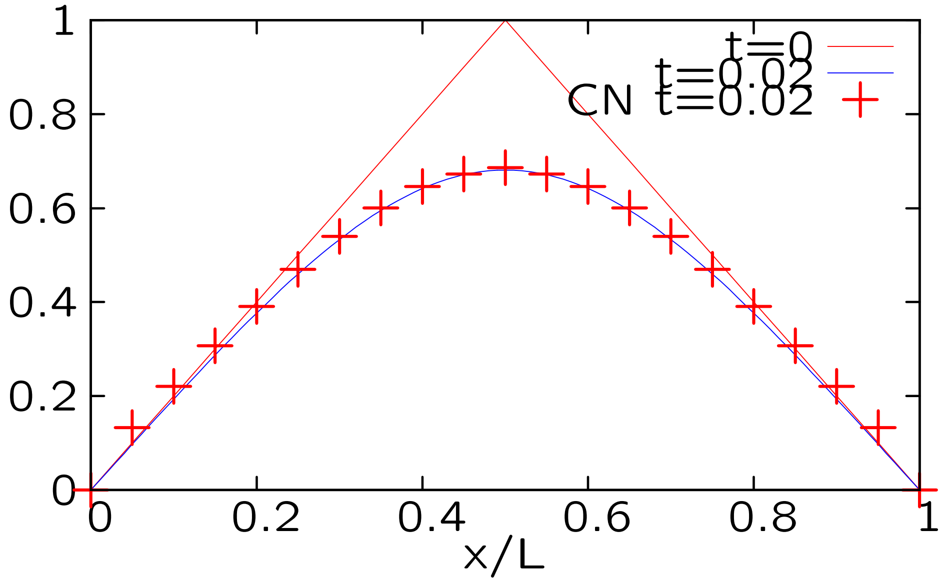
Using von Neumann stability analysis we get

$$A = \frac{1 - r + r\cos(k\Delta x)}{1 + r - r\cos(k\Delta x)}$$

$$= \frac{1 - 2r\sin^2(k\Delta x/2)}{1 + 2r\sin^2(k\Delta x/2)}$$

So $|A| < 1$ for all values of k. So the Crank-Nicolson method is unconditionally stable. So the price of solving a tridiagonal system at each step is worth paying since this method allows large step sizes. This is the most popular numerical method for the diffusion equation.

Crank-Nicolson Results



- $\Delta t = \frac{1}{200}\frac{L^2}{\alpha}, \Delta x = \frac{L}{20}$

- $r = \frac{\alpha \Delta t}{(\Delta x)^2} = 2$

- Works even with $r > 1$.

The 1D Advection Equation

- Simplest hyperbolic PDE

- $\dfrac{\partial u}{\partial t} = -c\dfrac{\partial u}{\partial x}$

- Describes a scalar field $u(x,t)$ carried by a flow at constant speed c

- Solution: $u(x,t) = F(x - ct)$

Initial data

- $u(x, 0)$ specified for all x, OR,

- $u(x, 0)$ specified for $a \leq x \leq b$, and $u(a,t)$ specified for $t > 0$

The FTCS Differencing Scheme

- $u_t \approx \frac{u_m^{n+1} - u_m^n}{\Delta t}$

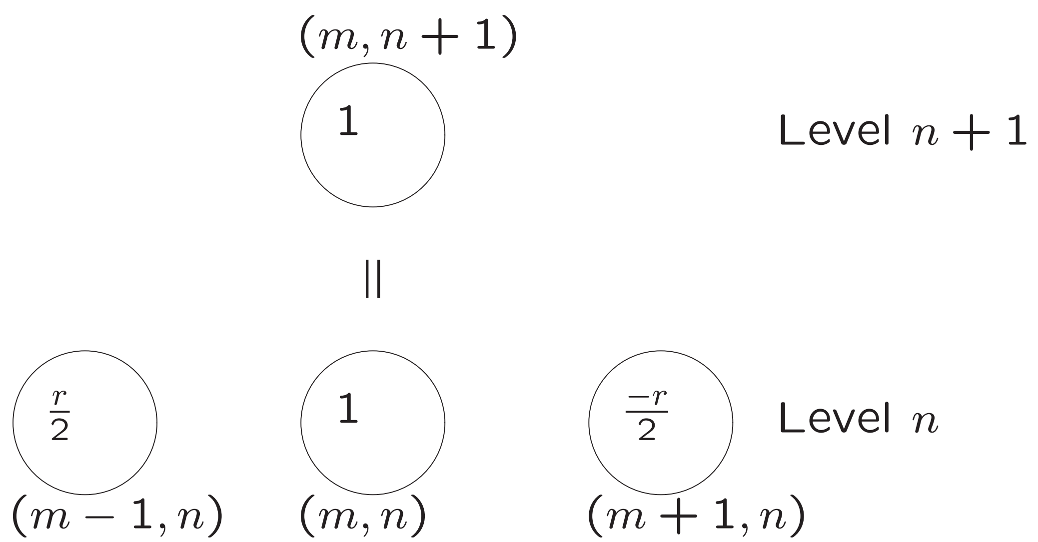
- $u_x \approx \frac{u_{m+1}^n - u_{m-1}^n}{2\Delta x}$

- $$u_m^{n+1} = u_m^n - \frac{r}{2}(u_{m+1}^n - u_{m-1}^n)$$

- Explicit scheme, easy to use

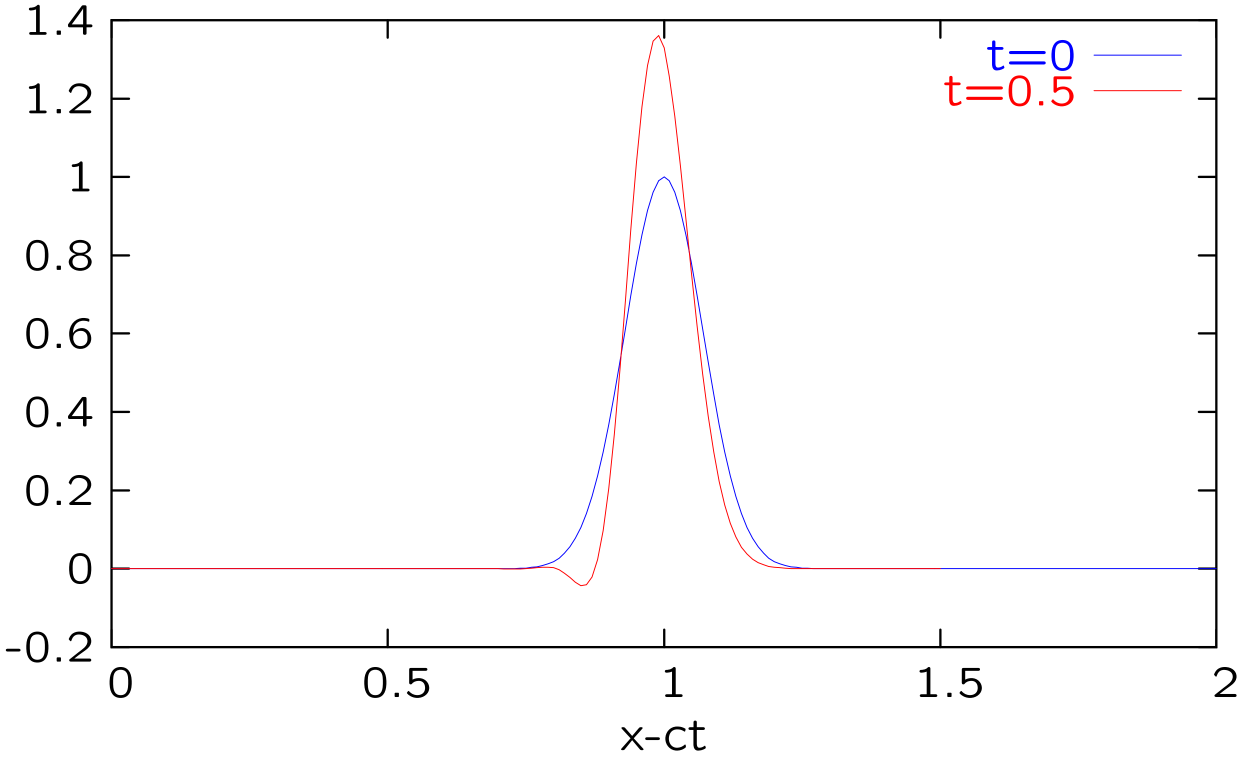
- Is it stable?

FTCS Computational Molecule



- $r = \frac{c\Delta t}{\Delta x}$ (Courant Number)

FTCS Scheme Results



- Pulse grows in amplitude

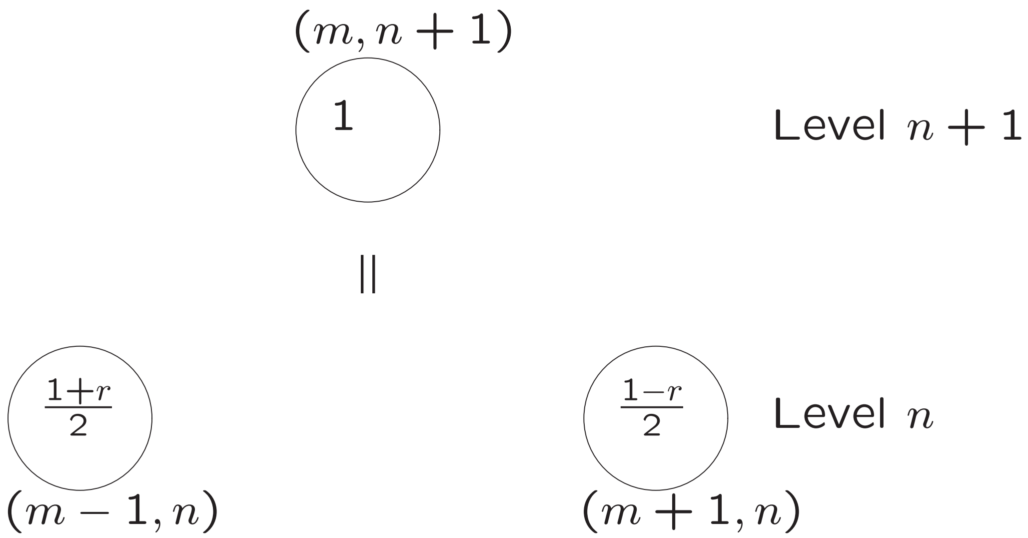
- $A = 1 - ir\sin(k\Delta x)$

- $|A| > 1$. Always unstable!

The Lax Scheme

To fix the instability of the FTCS scheme, Lax replaced u_m^n by the average of its left and right neighbours, $(u_{m-1}^n + u_{m+1}^n)/2$, to get

$$u_m^{n+1} = \frac{1+r}{2} u_{m-1}^n + \frac{1-r}{2} u_{m+1}^n$$



Stability Analysis of the Lax Scheme

Writing $u(x,t) = U(t)e^{ikx}$ we get, $U_m^{n+1} = AU_m^n$, where

$$A = \cos(k\Delta x) - ir\sin(k\Delta x).$$

The Lax scheme is thus unconditionally stable($|A| < 1$ for all k), provided $|r| < 1$.

Since, $r = \dfrac{c\Delta t}{\Delta x}$, this means

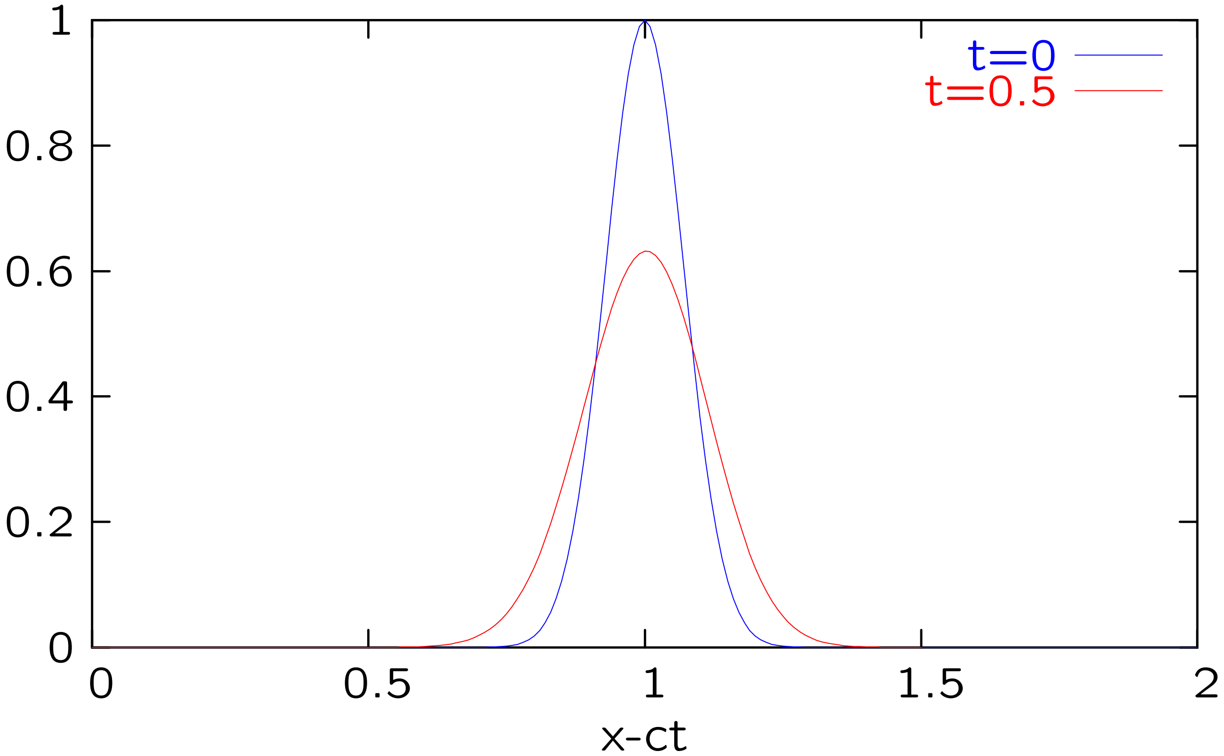
$$\Delta t < \frac{\Delta x}{c}$$

This is the well-known Courant-Friedrichs-Lewy(or CFL) stability criterion. *All explicit* stable differencing schemes are subject to the CFL constraint. The physical meaning is that having Δt larger than the CFL limit artificially slows down the speed at which the numerical information propagates and thus prevents convergence.

The Lax Scheme/MATLAB Code

```
function unew = stepLax( u, r )
  N = length( u ) - 1;
  unew = zeros( N + 1, 1 );
  cm = 0.5 * ( 1 + r );
  cp = 0.5 * ( 1 - r );
  for ii = 2:N
    unew(ii) = cm * u(ii-1) + cp * u(ii+1);
  end

  unew(1) = 0.0;
  unew(N+1) = 0.0;
```

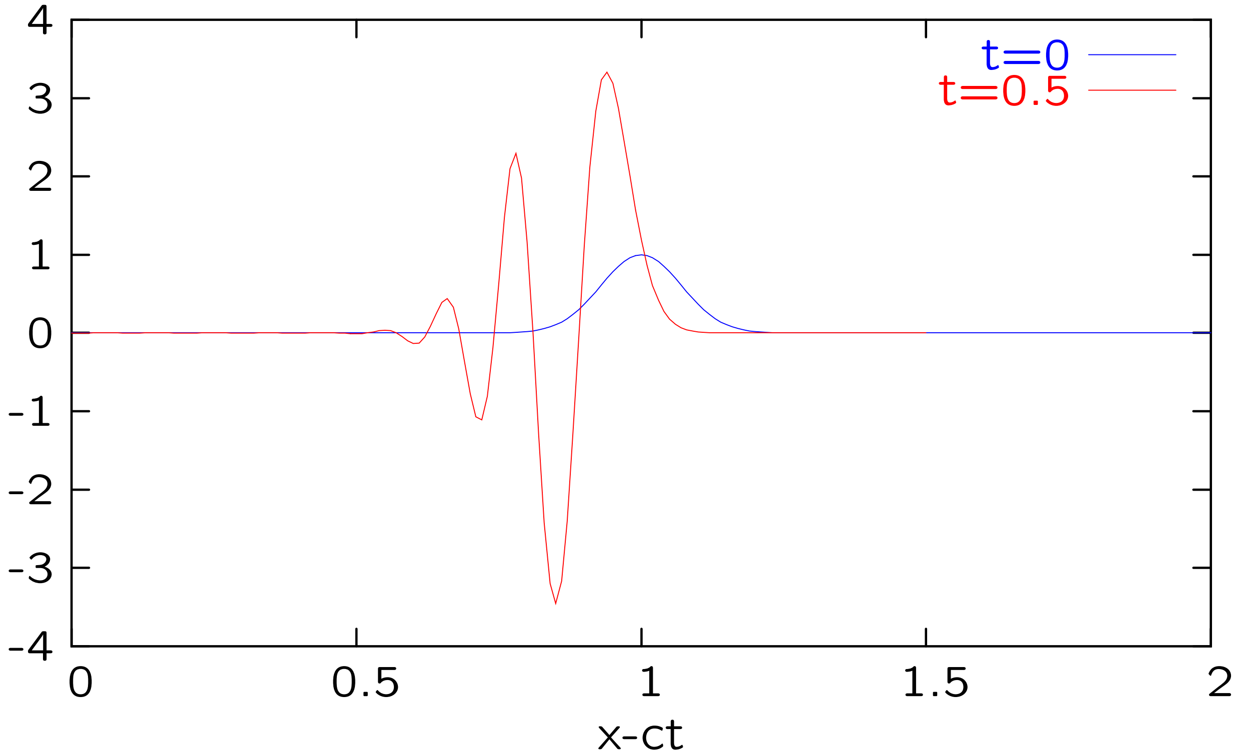
Lax Scheme Results



- $r = 0.5$

- Shows Dispersion

Lax Scheme/CFL Violated



- $r = 2$

Crank-Nicolson for Advection

- Scheme

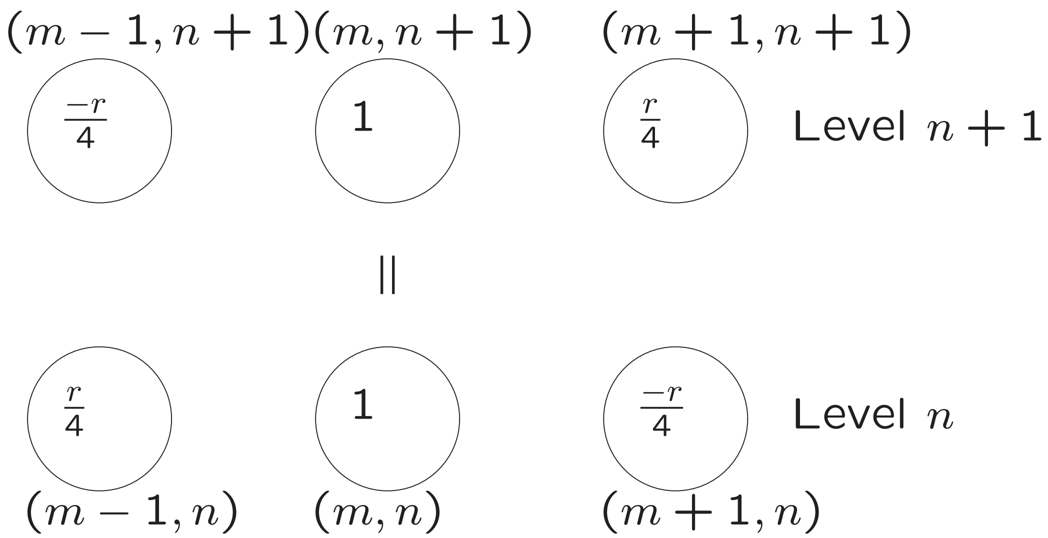
$$u_m^{n+1} + \frac{r}{4}(u_{m+1}^{n+1} - u_{m-1}^{n+1})$$

$$= u_m^n - \frac{r}{4}(u_{m+1}^n - u_{m-1}^n).$$

- Stable for all r.

$$A = \frac{1 - i(r/2)\sin(k\Delta x)}{1 + i(r/2)\sin(k\Delta x)}$$

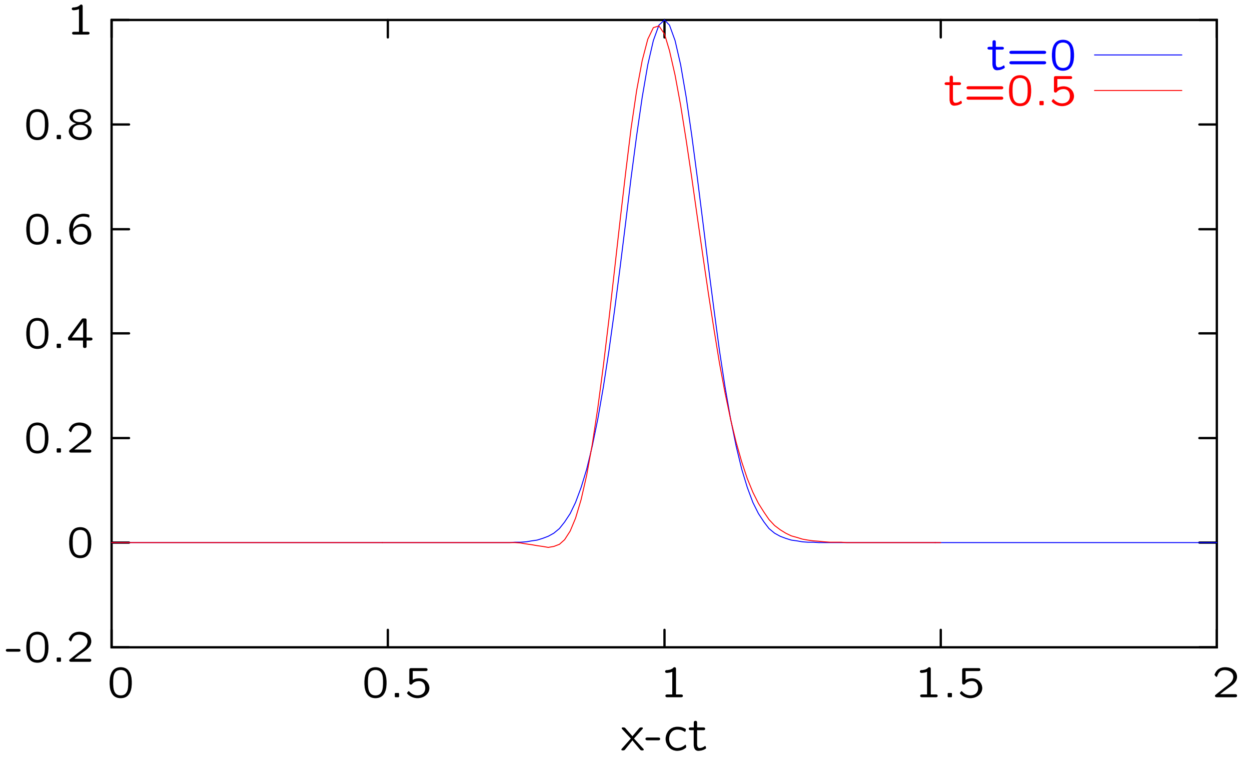
- Implicit, not subject to CFL constraint



Crank-Nicolson MATLAB Code

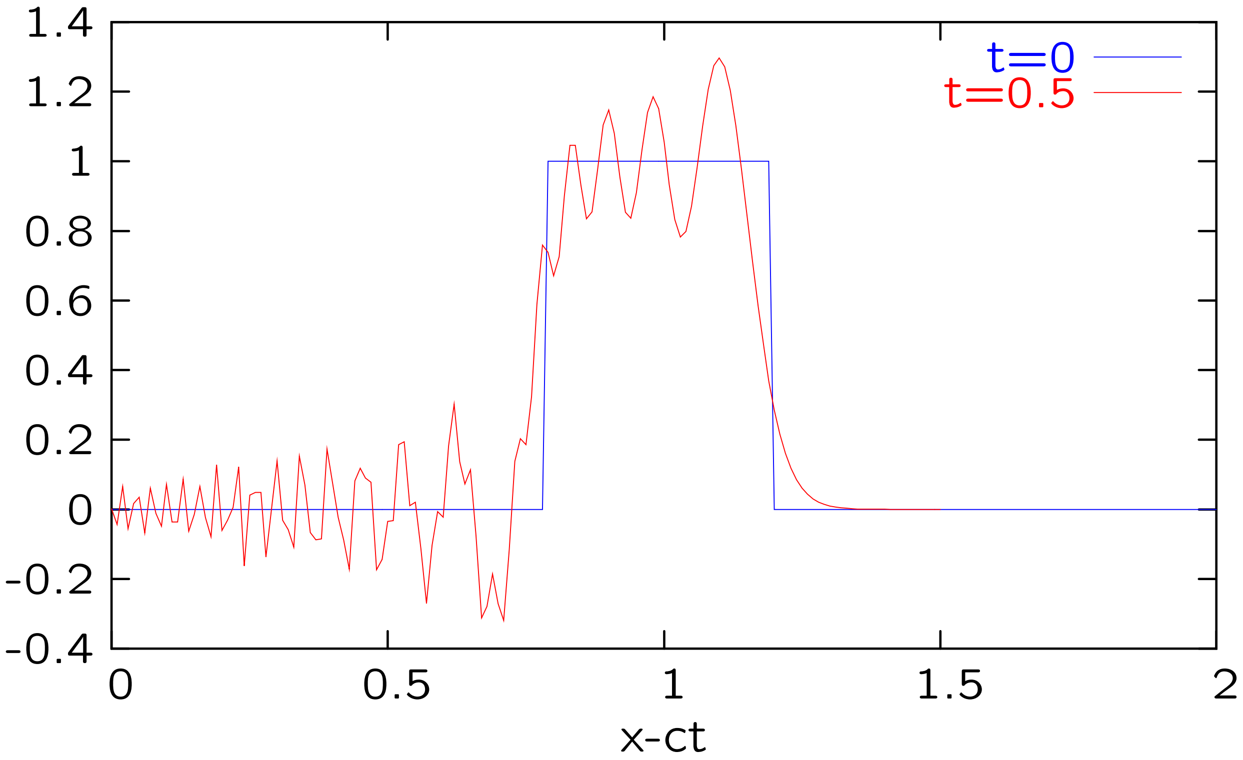
```
function unew = stepCN( u, r )
  N = length( u ) - 1;

  w = zeros( N + 1, 1 );
  A = eye( N+1 );
  for ii = 2:(N+1)
    A(ii-1,ii) = 0.25 * r;
  end
  for ii = 1:N
    A(ii+1,ii) = -0.25 * r;
  end

  for ii = 2:N
    w(ii) = u(ii) - 0.25*r*(u(ii+1) - u(ii-1));
  end

  unew = A \ w;

  unew(1) = 0.0;
  unew(N+1) = 0.0;
```

Crank-Nicolson/Gaussian Pulse



- No Dispersion, Stable
- $r = 2$ (No CFL Constraint)
- Each step requires the solution of a tridiagonal system

Crank-Nicolson/Square Pulse



- Spurious Oscillations

- Not good for waveforms with sharp edges

- All central difference schemes have this problem

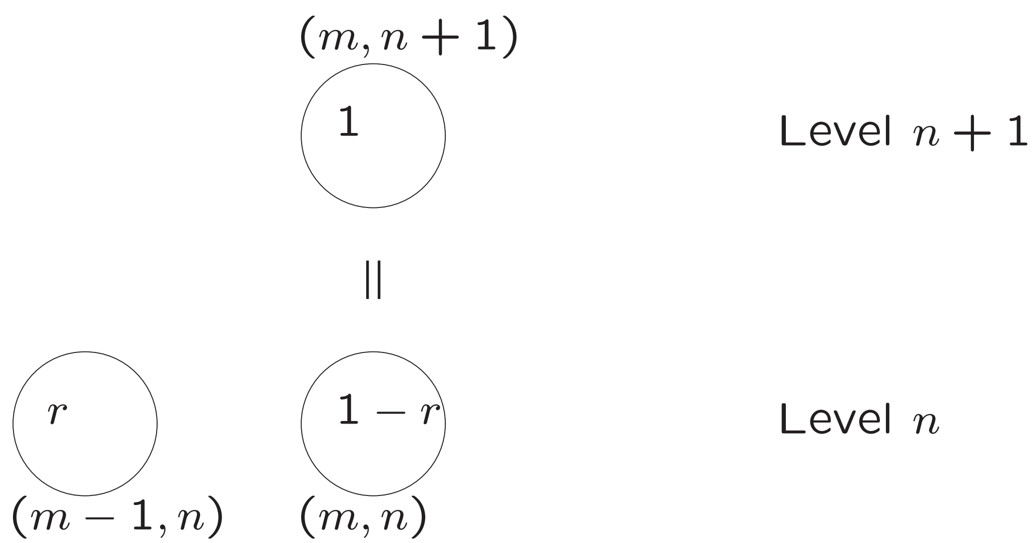
Upwind Differencing

The only way to suppress spurious oscillations at sharp edges is to use the *upwind* differencing scheme.

$$\frac{u_m^{n+1} - u_m^n}{\Delta t} = -c\frac{u_m^n - u_{m-1}^n}{\Delta x}$$

This gives the explicit scheme

$$u_m^{n+1} = u_m^n - r(u_m^n - u_{m-1}^n)$$



$r = \frac{c\Delta t}{\Delta x}$

Upwind Differencing Properties

Stability:

$$A = 1 - r(1 - \cos(k\Delta x)) - ir\sin(k\Delta x)$$

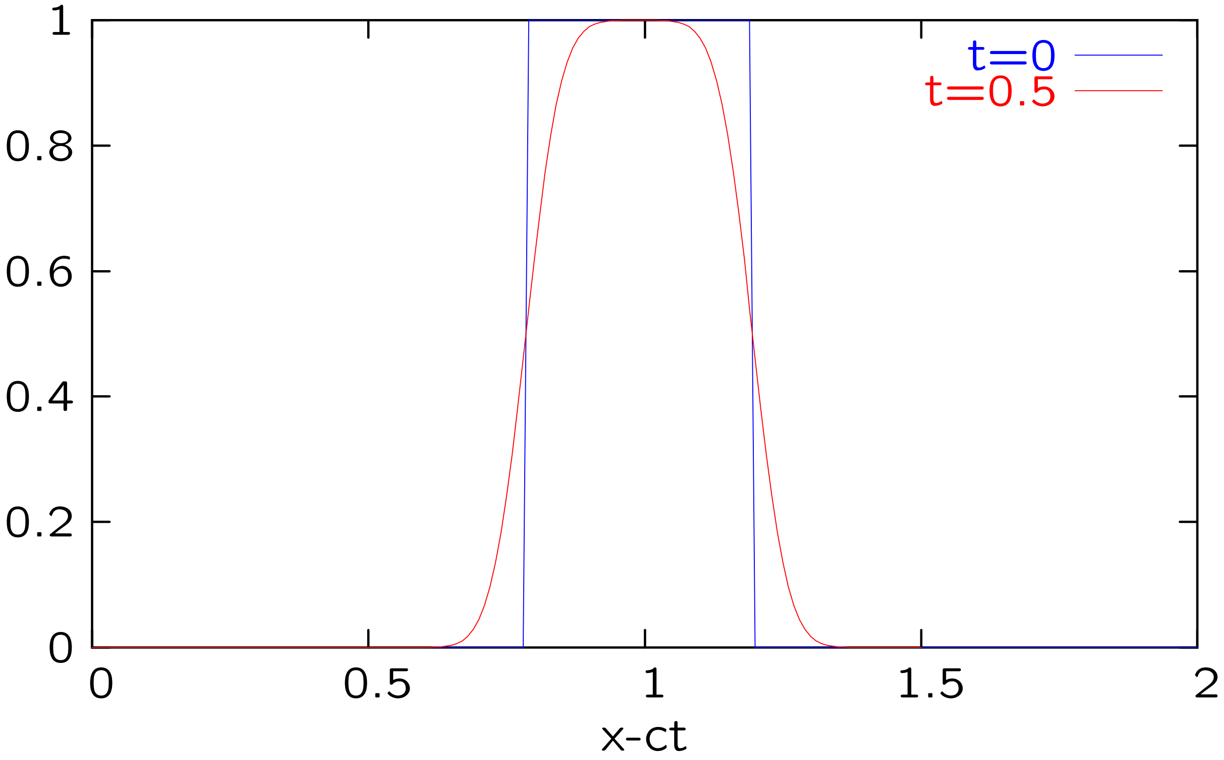
So

$$|A|^2 = 1 - 2r(1-r)(1 - \cos(k\Delta x))$$

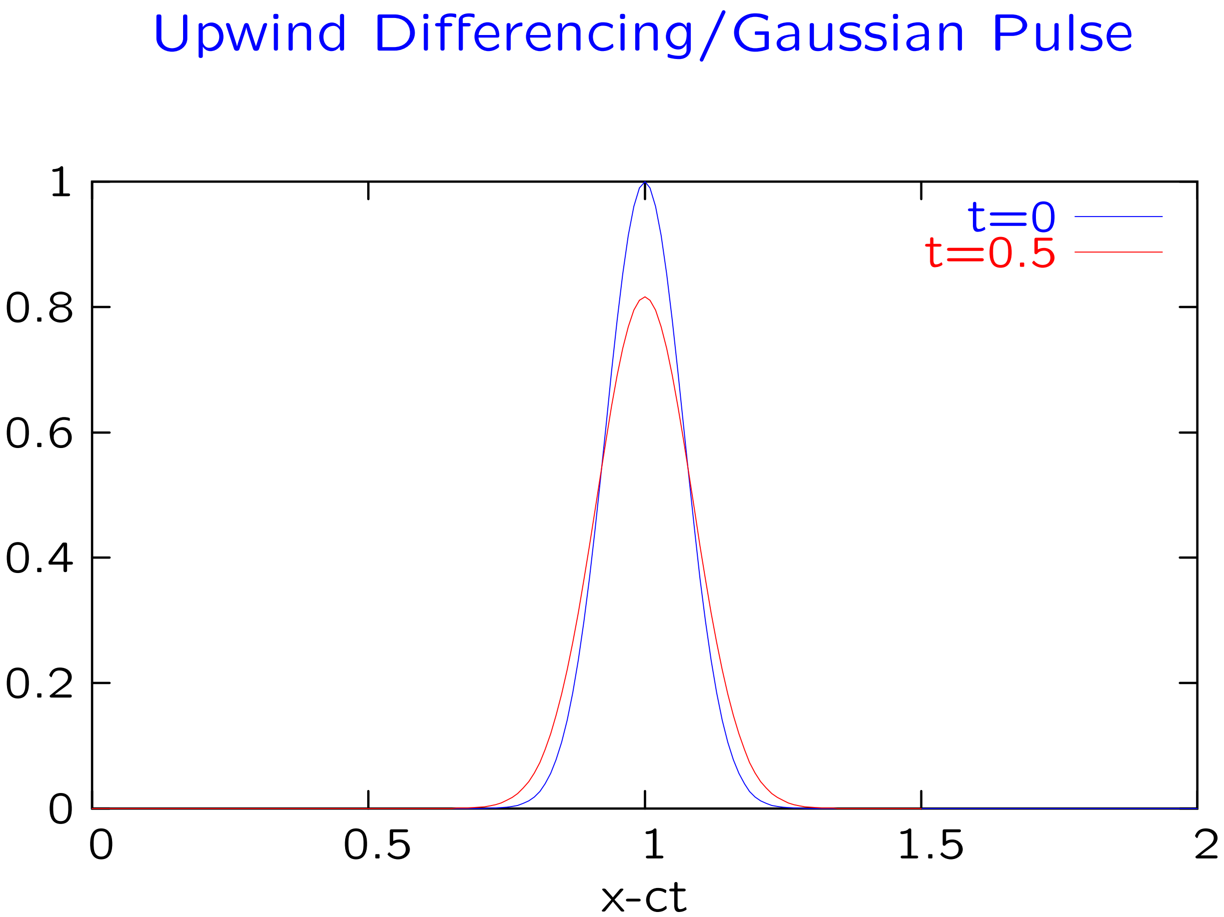
$|A| < 1$, for all k provided $r < 1$. So the CFL condition needs to be satisfied.

This method is only first-order accurate in space.

Upwind Differencing/Square Pulse



- No oscillations
- Stable if CFL condition is satisfied
- Still dispersive



- Not as dispersive as Lax

Lax-Wendroff Scheme

$$u(x,t+\Delta t) = u(x,t) + (\Delta t)\frac{\partial u}{\partial t}$$

$$+\frac{(\Delta t)^2}{2}\frac{\partial^2 u}{\partial t^2} + \cdots$$

But by the advection equation,

$$\frac{\partial u}{\partial t} = -c\frac{\partial u}{\partial x}$$

and

$$\frac{\partial^2 u}{\partial t^2} = c^2\frac{\partial^2 u}{\partial x^2}.$$

So

$$u(x,t+\Delta t) = u(x,t) - (c\Delta t)\frac{\partial u}{\partial x}$$

$$+\frac{(c\Delta t)^2}{2}\frac{\partial^2 u}{\partial x^2} + \cdots$$

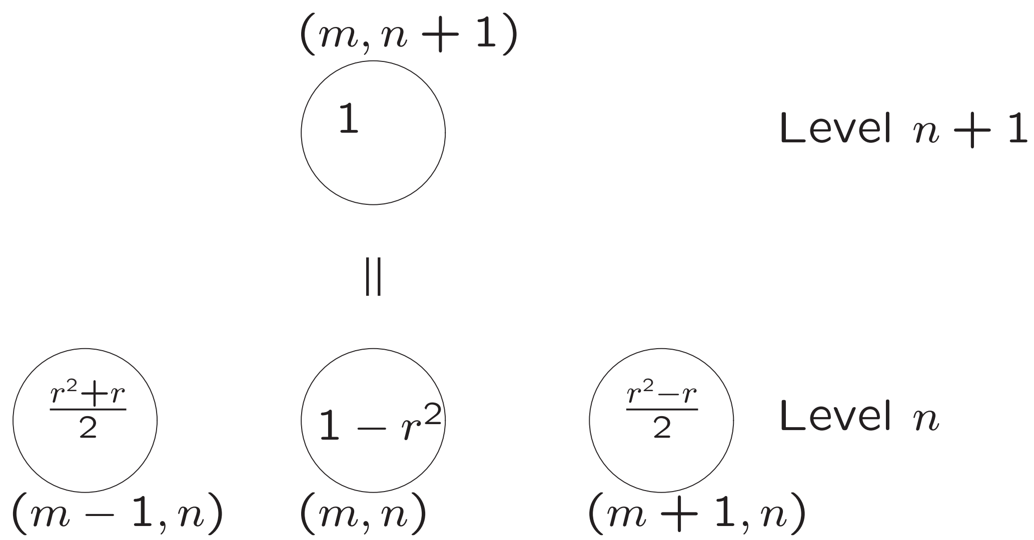
The advantage of having spatial derivatives instead of time derivatives is that at one level we have many points to compute the space derivatives.

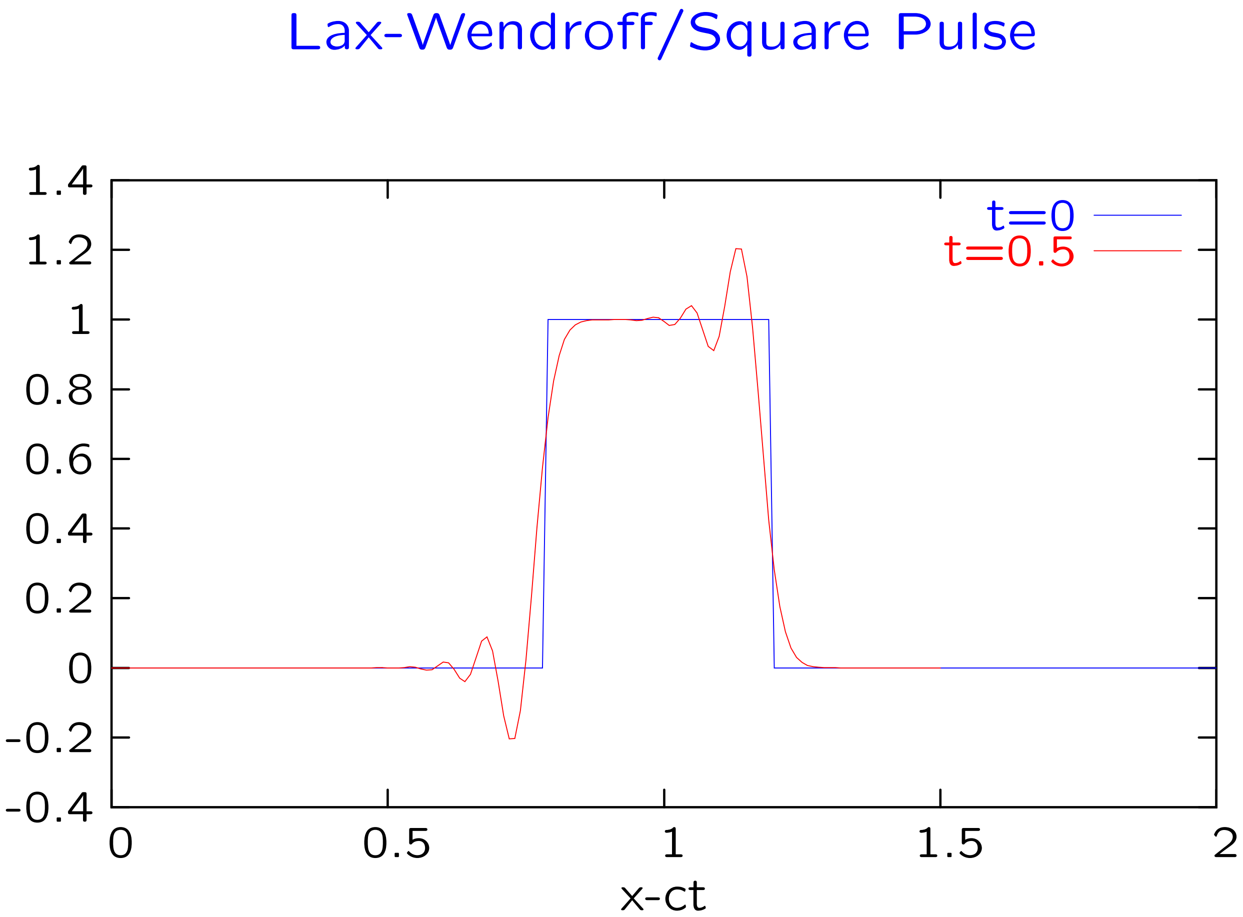
Lax-Wendroff Scheme

Discretizing the above equation gives us the more stable and more accurate scheme

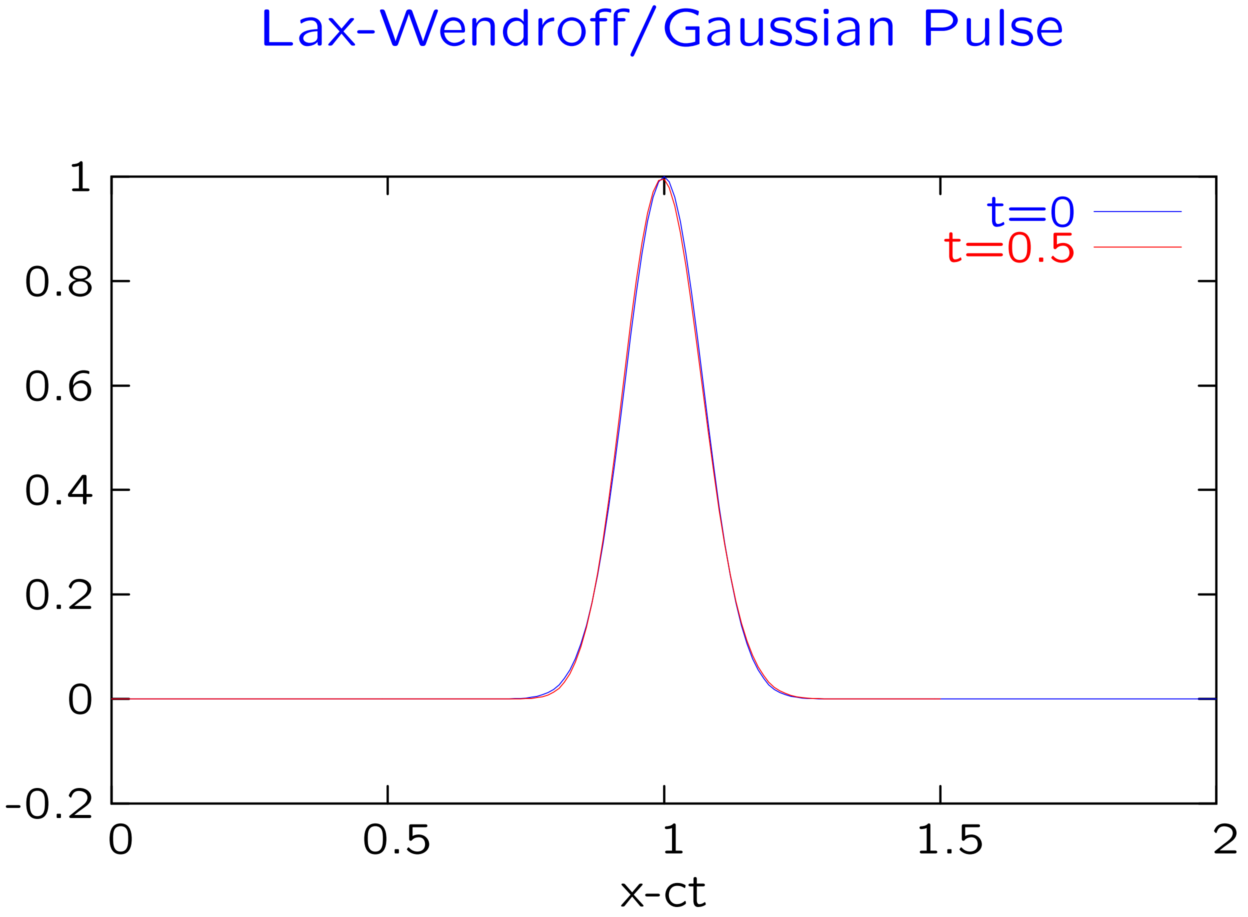
$$u_m^{n+1} = (1 - r^2)u_m^n + \frac{r^2 + r}{2}u_{m-1}^n + \frac{r^2 - r}{2}u_{m+1}^n$$

known as the Lax-Wendroff scheme. It is very popular.





- Less oscillations than Crank-Nicolson



- Not very dispersive

Which Method?

- No one knows a differencing scheme which is both non-dispersive and can handle sharp wave-fronts.

- There has been a lot of research in this area.

- There is a need to understand hyperbolic PDEs better.

Conclusion

- All numerical methods need to be analyzed for stability and accuracy.

- What seems like a minor change can have a profound effect on the stability and the accuracy of a numerical method.