OLD DOMINION UNIVERSITY

Partial Differential Equations

A. Godunov

1. Hyperbolic PDE
2. Wave equation
3. Convection equation: explicit methods
4. Convection equation: implicit methods
5. Wave equation: special considerations

1

Part 1:

Hyperbolic PDEs

2

Hyperbolic PDEs

The wave equation

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

where c is the wave propagation speed. Waves travel in both directions at the velocities $+c$ and $-c$.

The convection equation

$$\frac{\partial f}{\partial t} + v\,\frac{\partial f}{\partial x} = 0$$

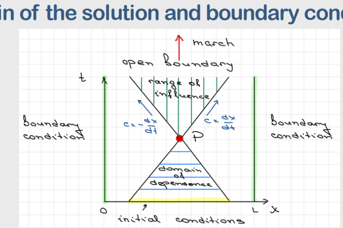
where v is is the convection velocity. The convection equation models a wave travelling in one direction, the direction of the velocity v.

Hyperbolic PDEs are are initial-boundary-value problems in open domains. However, hyperbolic PDEs have a finite physical information propagation speed.

3

3

The domain of the solution and boundary conditions

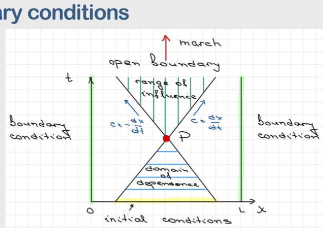


The hyperbolic PDEs have a finite physical information propagation speed. As a result, the solution at a given point P at time level n depends on the solution at points in the area defined by the propagation speed c.

4

4

The boundary conditions



The solution must satisfy an initial conditions at $t = 0, f(x,0) = F(x)$ for the convection equation and additional condition $f_t(x,0) = G(x)$ for the wave equation. The time coordinate has an open final value.

For the convection equation – one boundary condition must be specified, for the wave equation (second order in the spatial coordinate, two boundary conditions are required.

5

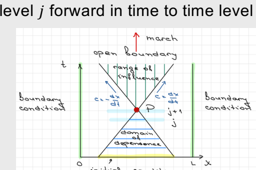
5

Finite-difference method

The finite difference method is a numerical procedure which solves a partial differential equation (PDE) by

1. discretizing the continuous physical domain
2. approximating the individual exact partial derivatives in the PDE by algebraic finite difference approximations (FDAs),
3. substituting the FDAs into the PDE
4. and solving the resulting algebraic finite difference equations

The objective of the numerical solution of a hyperbolic PDE is to march the solution at time level j forward in time to time level $j + 1$



6

6

Part : 2

Wave equation

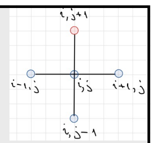
7

Part : 2A

CTCS: centered time centered space

8

Finite-differences



The wave equation

$$f_{tt} = c^2 f_{xx}$$

Similar to the elliptic equation we replace second derivatives (now for time and space) on

$$f_{tt} = \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{\Delta t^2}, \qquad f_{xx} = \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{\Delta x^2}$$

$$\frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{c^2 \Delta t^2} = \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{\Delta x^2}$$

using $a = c\Delta t/\Delta x$

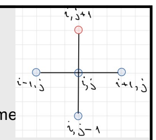
$$f_{i,j+1} = 2(1-a^2)f_{i,j} + a^2 f_{i-1,j} + a^2 f_{i+1,j} - f_{i,j-1}$$

The explicit algorithm propagates the wave from two earlier times j and $j-1$, and from three nearby positions, $i-1$, i, and $i+1$, to a later time $j+1$ and a single space position i

9

9

Initializing the steps



Initializing the solution requires displacements from two earlier times, but the initial conditions are for only one time

However, we can use the three-point centered-difference formula to approximate the first time derivative of the solution (just for the first step)

$$f_t(i,j) = \frac{f_{i,j+1} - f_{i,j-1}}{2\Delta t}, \qquad f_{i,j-1} = f_{i,j+1} - 2\Delta t f_t(i,j)$$

then the initialization of the algorithm is

$$f_{i,j+1} = 2(1-a^2)f_{i,j} + a^2 f_{i-1,j} + a^2 f_{i+1,j} - f_{i,j+1} + 2\Delta t f_t(i,j)$$

or the first step is calculated as

$$f_{i,2} = f_{i,1} + \frac{a^2}{2}\left(f_{i+1,1} - 2f_{i,1} + f_{i-1,1}\right) + \Delta t f_t(i,1)$$

10

10

Stability and accuracy

For stability the method

$$f_{i,j+1} = 2(1-a^2)f_{i,j} + a^2 f_{i-1,j} + a^2 f_{i+1,j} - f_{i,j-1}$$

the $c \le \Delta x/\Delta t$ or $c\Delta t \le \Delta x$ (Courant condition).

The condition means that the solution gets better with smaller time steps but gets worse for smaller space step

Since c is the wave speed, this means that the distance $c\Delta t$ traveled by the solution in one time step should not exceed the space step Δx.

Attention: if $c\Delta t = \Delta x$, then the method provides higher accuracy (the leading term in the error vanishes!)

The accuracy of the algorithm is $O(\Delta t^2) + O(\Delta x^2)$ if $c\Delta t < \Delta x$ (and even higher for $c\Delta t = \Delta x$.)

11

11

The final touch …

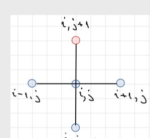
The explicit algorithm has various names: time-stepping, CTCS (centered-time centered-space method)

$$f_{i,j+1} = 2(1-a^2)f_{i,j} + a^2 f_{i-1,j} + a^2 f_{i+1,j} - f_{i,j-1}$$

It is applied with initial and boundary conditions for $0 \le x \le L$ and $t \ge 0$

$$f(x,0) = g(x), \qquad f_t(x,0) = h(x)$$

$$f(0,t) = \mu(t), \qquad f(L,t) = \eta(t)$$



12

12

Example: MatLab code

```
%{
  Solving the wave equation with Dirichlet BCs
  Method: Centered-time Centered-space difference
  INPUT:
  f(i,j)    initial and boundary condition
  ft(i)     initial condition for the derivative
  dx, dy    grid increments
  nx        number of grid points in x direction
  nt        number of grid points in y direction
  c         speed
  OUTPUT
  f2(x,y)   the solution
%}
function[f2] = pdeW1(f,ft, dx,dt,nx,nt,c)
% preparation
f2 = zeros(nx,nt);
a = c*dt/dx;
a2 = a*a;
fprintf(' CFL number a = %6.4f \n',a)
if a > 1.0
  fprintf(' ATTENTION: the CFL is too large for the method \n ')
end
```

13

13

Example: MatLab code

```
% Preparing the first iteration (with the derivatives)
for i=2:nx-1
  f(i,2) = f(i,1) + 0.5*a2*(f(i+1,1)-2*f(i,1)+f(i-1,1))+dt*ft(i);
end
% Marching forward
for j=2:nt
  for i=2:nx-1
    f(i,j+1) = 2.0*(1-a2)*f(i,j) +a2*(f(i+1,j)+f(i-1,j)) - f(i,j-1);
  end
end
% Prepare OUTPUT
for j=1:nt
  for i=1:nx
    f2(i,j)= f(i,j);
  end
end
end
```

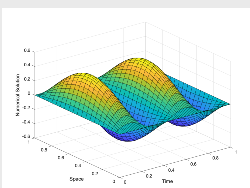
14

14

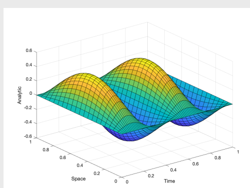
Example: Using explicit CTCS method

$$f_{tt} = 16 f_{xx}$$

Numerical solutions



Analytic solution



for the same initial conditions (the boundary conditions set to zero)

$$f(x,0) = 0, \qquad f_t(x,0) = 2\pi \sin \pi x$$

analytic solution: $f(x,t) = \frac{1}{2}\sin(\pi x)\sin(4\pi t)$

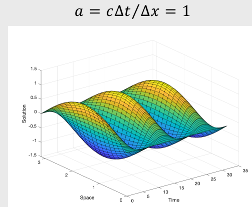
15

15

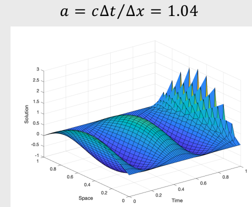
Example: Instability when CFL number >1

$$f_{tt} = c^2 f_{xx}$$

$a = c\Delta t/\Delta x = 1$



$a = c\Delta t/\Delta x = 1.04$



Example of instability

16

16

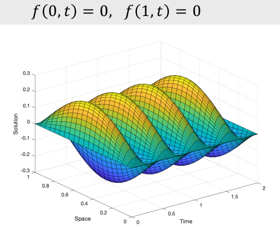
Example: solutions with non-zero boundary conditions

$$f_{tt} = c^2 f_{xx}$$

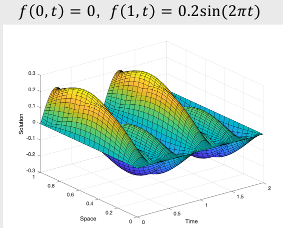
initial conditions: $f(x,0) = 0, \quad f_t(x,0) = \pi \sin \pi x$

Boundary conditions:

$f(0,t) = 0, \; f(1,t) = 0$



$f(0,t) = 0, \; f(1,t) = 0.2\sin(2\pi t)$



17

17

Part 3:

Convection equation: explicit methods

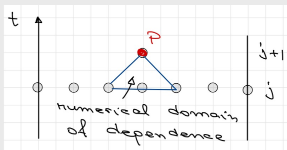
$$f_t + v f_x = 0$$

18

Explicit methods

The objective of the numerical solution of a parabolic PDE is to march the solution at time level j forward in time to time level j+1.

Finite difference methods when the solution at point *P* at time level *j*+1 depends only on the solution at neighboring points at time level *j* are called explicit methods. Explicit methods are computationally faster than implicit methods because there is no system of finite difference equations to solve.

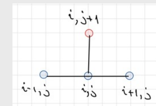


19

19

The forward-time centered-space (FTCS) method

The convection PDE $f_t + v f_x = 0$



$$\frac{f_{i,j+1} - f_{i,j}}{\Delta t} + v\frac{f_{i+1,j} - f_{i-1,j}}{2\Delta x} = 0$$

Solving for $f_{i,j+1}$

$$f_{i,j+1} = f_{i,j} - \frac{c}{2}(f_{i+1,j} - 2f_{i-1,j})$$

where $c = v\Delta t/\Delta x$ is called the convection number

Attention: the FTCS approximation of the convection equation is unconditionally unstable. Consequently, it is unsuitable for solving the convection equation, or any other hyperbolic PDE.

20

20

The Lax method

The idea – approximating $f_{i,j}$ in FTCS $f_{i,j+1} = f_{i,j} - \frac{c}{2}(f_{i+1,j} - 2f_{i-1,j})$

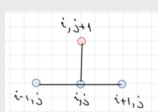
as $f_{i,j} = (f_{i+1,j} + f_{i-1,j})/2$

Then

$$f_{i,j+1} = \frac{1}{2}(f_{i+1,j} + f_{i-1,j}) - \frac{c}{2}(f_{i+1,j} - 2f_{i-1,j})$$

The Lax approximation of the convection equation is conditionally stable

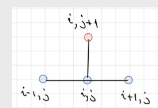
Attention: The Lax equation is not a consistent approximation of the convection equation. Don't use it



21

21

The Lax-Wendroff one-step method



The method is $O(\Delta t^2) + O(\Delta x^2)$ and very popular

For the convection equation

$$f_{i,j+1} = f_{i,j} - \frac{c}{2}(f_{i+1,j} - f_{i-1,j}) + \frac{c^2}{2}(f_{i+1,j} - 2f_{i,j} + f_{i-1,j})$$

The Lax-Wendroff approximation of the convection equation is conditionally stable for

$$c = \frac{v\Delta t}{\Delta x} \leq 1$$

The Lax-Wendroff one-step approximation of the convection equation is consistent and conditionally stable.

However, the method is quite complicated for nonlinear PDEs, systems of PDEs, and two- and three-dimensional physical spaces.

22

22

Other methods

- The extension of the Lax-Wendroff method by Richmyer. The Richtmyer method is much simpler than the Lax-Wendroff one-step method for nonlinear PDEs and systems of PDEs. The method is $O(\Delta t^2) + O(\Delta x^2)$
- The MacCormack (predictor-corrector) method: The method can be used to solve linear partial differential equations, nonlinear PDEs, and systems of PDEs with equal ease. It is identical to Lax-Wendroff method for linear PDEs. The method is $O(\Delta t^2) + O(\Delta x^2)$
- Upwind methods (the first- and second-order methods)
  The first-order method is nor very accurate
  The second-order method is $O(\Delta t^2) + O(\Delta x^2)$ and conditionally stable for $c < 2$.

23

23

Part 4:

Convection equation: implicit methods

24

Basics

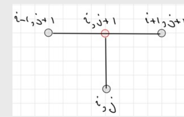
- Explicit methods share one undesirable feature: they are only conditionally stable. Consequently, the allowable time step is usually quite small, and the amount of computational effort required to obtain the solution of some problems is immense.
- Implicit finite difference methods are unconditionally stable.
- There is no limit on the allowable time step required to achieve a stable solution.
- There is, of course, some practical limit on the time step required to maintain the truncation errors within reasonable limits, but this is not a stability consideration; it is an accuracy consideration.
- One disadvantage: the solution at a point at the solution time level $j+1$ depends on the solution at neighboring points at level $j+1$, which are also unknown

25

25

Backward-time centered-space (BTCS) method

The convection PDE $f_t + v f_x = 0$



$$\frac{f_{i,j+1} - f_{i,j}}{\Delta t} = v\,\frac{f_{i+1,j+1} - f_{i-1,j+1}}{2\Delta x}$$

$$-\frac{c}{2} f_{i-1,j+1} + f_{i,j+1} + \frac{c}{2} f_{i+1,j+1} = f_{i,j}$$

$c = v\Delta t/\Delta x$ is the convection number

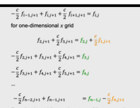
The BTCS method applied to the convection equation is consistent and unconditionally stable. The method is $O(\Delta t) + O(\Delta x^2)$.

The BTCS approximation of the convection equation yields poor results, except for very small values of the convection number, for which explicit methods are generally more efficient.

26

26

Application


$-\frac{c}{2} f_{i-1,j+1} + f_{i,j+1} + \frac{c}{2} f_{i+1,j+1} = f_{i,j}$
for one-dimensional x grid
$f_{2,j+1} + \frac{c}{2} f_{3,j+1} = f_{2,j} + \frac{c}{2} f_{1,j+1}$
$-\frac{c}{2} f_{2,j+1} + f_{3,j+1} + \frac{c}{2} f_{4,j+1} = f_{3,j}$
$-\frac{c}{2} f_{3,j+1} + f_{4,j+1} + \frac{c}{2} f_{5,j+1} = f_{4,j}$
...
$-\frac{c}{2} f_{n-2,j+1} + f_{n-1,j+1} = f_{n-1,j} - \frac{c}{2} f_{n,j+1}$
Equation is a tridiagonal system of linear equations.

$$-\frac{c}{2} f_{i-1,j+1} + f_{i,j+1} + \frac{c}{2} f_{i+1,j+1} = f_{i,j}$$

for one-dimensional x grid

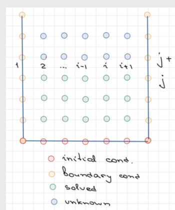
$$f_{2,j+1} + \frac{c}{2} f_{3,j+1} = f_{2,j} + \frac{c}{2} f_{1,j+1}$$

$$-\frac{c}{2} f_{2,j+1} + f_{3,j+1} + \frac{c}{2} f_{4,j+1} = f_{3,j}$$

$$-\frac{c}{2} f_{3,j+1} + f_{4,j+1} + \frac{c}{2} f_{5,j+1} = f_{4,j}$$

...

$$-\frac{c}{2} f_{n-2,j+1} + f_{n-1,j+1} \qquad = f_{n-1,j} - \frac{c}{2} f_{n,j+1}$$



Equation is a tridiagonal system of linear equations.

27

27

Part : 5

Wave equation: more methods

28

Replacing the wave equation into a system

The wave equation

$$f_{tt} = c^2 f_{xx}$$

is equivalent to the following set two coupled first-order convection equations:

$$f_t + c g_x = 0$$

$$g_t + c f_x = 0$$

Equations above suggest that the wave equations can be solved by the same methods that are employed to solve the convection equation: the FTCS method, the Lax method, upwind methods, the leap-frog method, the Lax-Wendroff one-step method, the Lax-Wendroff two-step method, the BTCS method, the Crank-Nicolson method, the hopscotch method.

However, it's more efficient to solve the wave equation using central second order differences straight with $f_{tt} = c^2 f_{xx}$.

29

29

The Lax-Wendroff one-step method for wave equation

For the system

$$f_t + c g_x = 0$$

$$g_t + c f_x = 0$$

$$f_{i,j+1} = f_{i,n} - \frac{a}{2}\left(g_{i+1,j} - g_{i-1,j}\right) + \frac{a^2}{2}\left(f_{i+1,j} - 2f_{i,k} + f_{i-1,j}\right)$$

$$g_{i,j+1} = g_{i,n} - \frac{a}{2}\left(f_{i+1,j} - f_{i-1,j}\right) + \frac{a^2}{2}\left(g_{i+1,j} - 2g_{i,k} + g_{i-1,j}\right)$$

with $a = c\Delta t/\Delta x$

for stability $a \le 1$.

30

30

d'Alembert solution

The wave equation

$$f_{tt} = c^2 f_{xx}$$

with given initial conditions

$$f(x,0) = \varphi(x), \qquad f_t(x,0) = \mu(x).$$

for infinitely long string has analytic solution (d'Alembert solution)

$$f(x,t) = \frac{\varphi(x+ct) + \varphi(x-ct)}{2} + \frac{1}{2c}\int_{x-ct}^{x+ct} \mu(p)dp\,.$$

In this case we only need to evaluate the integral above.

The solution represents a superposition of two traveling waves, a right-traveling wave and a left-traveling wave.

31

31

Part : 6

PDE solvers

hyperbolic, parabolic, elliptical

32

COMSOL

Simulate real-world designs, devices, and processes with multiphysics software from COMSOL
https://www.comsol.com

Modeling with Partial Differential Equations in COMSOL Multiphysics

- Poisson's and Laplace Equations
- Diffusion-Type Equations
- Convection–Diffusion Equations
- Coordinate Transformations
- Helmholtz Equation

and many more …

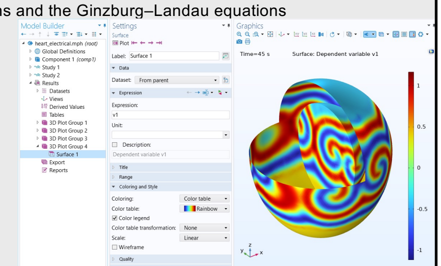
33

33

COMSOL

Example: Diffusion-Type Equations

Modeling the electrical activity in cardiac tissue using the General Form PDE mathematics interface. In this model, two different time-dependent nonlinear partial differential equation systems are used: the FitzHugh–Nagumo equations and the Ginzburg–Landau equations



34

MatLab:

Partial Differential Equation Toolbox:
Solve partial differential equations using finite element analysis

- Structural Mechanics
- Heat Transfer
- Electromagnetics
- General PDEs
- Geometry and Meshing
- Visualization and Postprocessing

35

35

And …

The are many software packages and libraries for solving PDE with application to engineering, physical sciences, medicine, manufacturing and many more …

36

36