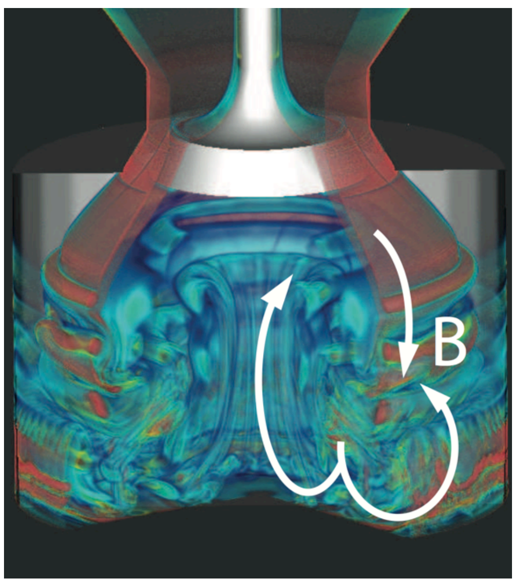
CS598: High-Order Methods for PDEs

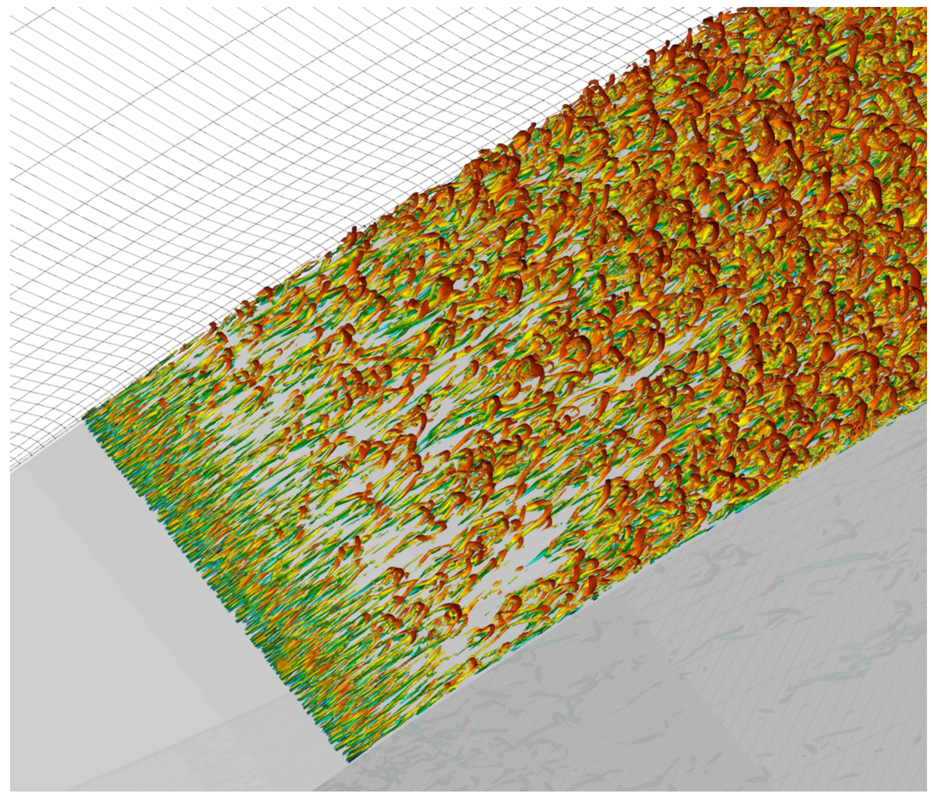
1131 Siebel Center, TR 9:30—10:45

Paul Fischer

fischerp@illinois.edu

http://fischerp.cs.illinois.edu/cs598/





Course Objectives

- Goal:
  - Equip you with the ability to analyze high-order methods for complex PDEs and to implement these efficiently.
  - By the end of the course you will have written a multidomain spectral code for complex nonlinear PDEs in space and time.
- Grading:
  - Grading will be based on weekly homework plus a final project.
  - The homework will involve a mix of coding and analysis.

Topics: 1/4

- 1D model problems using finite differences.
  - Formulation, accuracy, spectra.
  - Equivalent differential equation.

- A tour of timesteppers: accuracy, efficiency, stability.

- 1D problems using spectral Galerkin, spectral element, FEM:
  - Formulation, accuracy, spectra.
  - Comparison with finite differences.
  - Neumann and Robin boundary conditions.

Topics: 2/4

- Extension to two space dimensions.
  - Tensor-product forms
  - Fast operator evaluation / fast solvers.
  - Poisson equation with constant and variable coefficients.
  - General geometries: isoparametric mappings.
  - Boundary condition choices.

- Extension to three space dimensions.

- Comparison of different bases: FEM/spectral/Fourier.

Topics: 3/4

- Multi-domain spectral methods: 1D, 2D; matrix assembly.
- Iterative solvers:
  - Krylov-subspace projection: CG / Lanczos / GMRES.
  - Eigenvalue estimates, projection for unsteady problems.
  - Exponential integrators.

- Preconditioning: overlapping Schwarz, multigrid.

- Stabilization in higher space dimensions:
  - Dealiasing / filtering / bubble functions / SUPG.

Topics: 4/4

- Other systems:
  - Anisotropic diffusion
  - Stokes & Navier-Stokes
  - Maxwell's equations

- Characteristics-based timesteppers.

- Spectral-element/discontinuous-Galerkin (SEDG) methods.

Performance Considerations

- Performance matters.

- A high-order code, per gridpoint, must be as efficient as its low-order counterpart if it is to be adopted by engineers and scientists.

- What really matters is the performance for 3D problems.

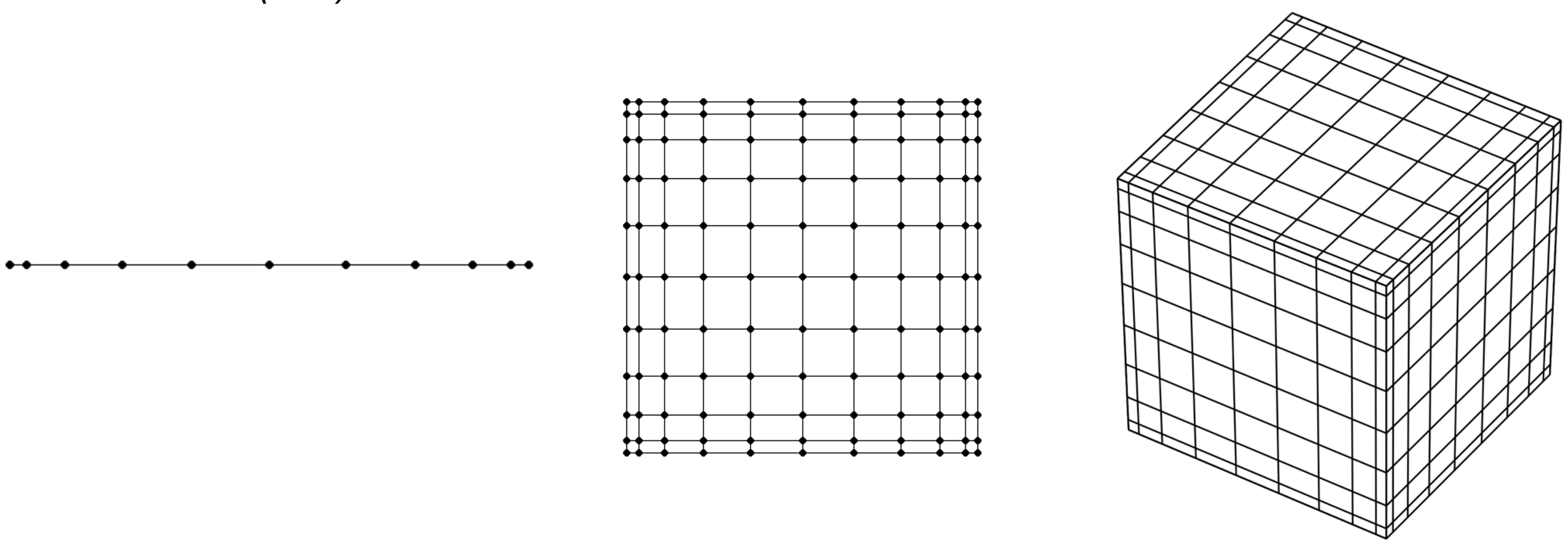
- 1D problems matter if they are a reasonable surrogate for 3D –

  - They are reasonable surrogates for analysis (and we will look at them in detail).

  - In general, they are not good surrogates for performance.

Performance Considerations

- Consider computing du/dx in 1D, 2D, and 3D
  - Operator access is $O(N^2)$ for high-order and $O(N)$ for low-order method.
  - Total data access is $O(N^d)$ in d space dimensions.
  - Memory Access Overhead for High-Order:
    - *1D: N-fold increase*
    - *2D: O(1)*
    - *3D: O(1/N)*



Debugging Strategies

- A macroscopic approach to debugging is generally more fruitful and more rewarding than a microscopic approach.

- Consequently, we spend a lot of time (in class and in research) answering questions for which we know the answer.
  - Thus, knowing the physics and/or mathematics is essential.
  - How would you know, otherwise, that your code has produced a meaningful result?

- With that intro…. Let’s turn to some examples.

Examples

- We’ll look at several examples this week… but first a quick peek at one important one…

Linear Advection Example

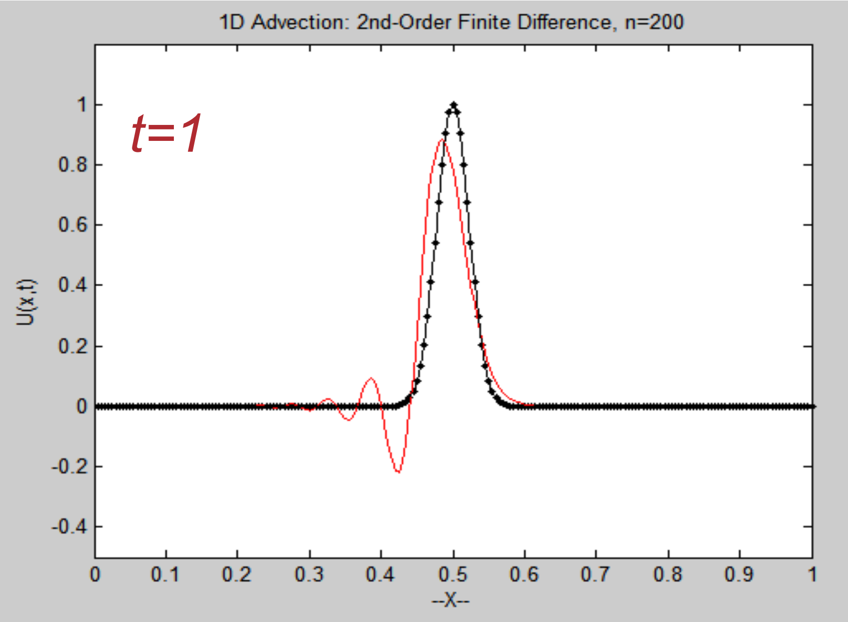
- Here, we consider linear advection with periodic BCs on [0,1]:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0, \qquad u(0,t) = u(1,t) \qquad u(x,0) = u_0.$$

- With speed c = 1, the travelling wave solution should return to the initial condition after each unit time.

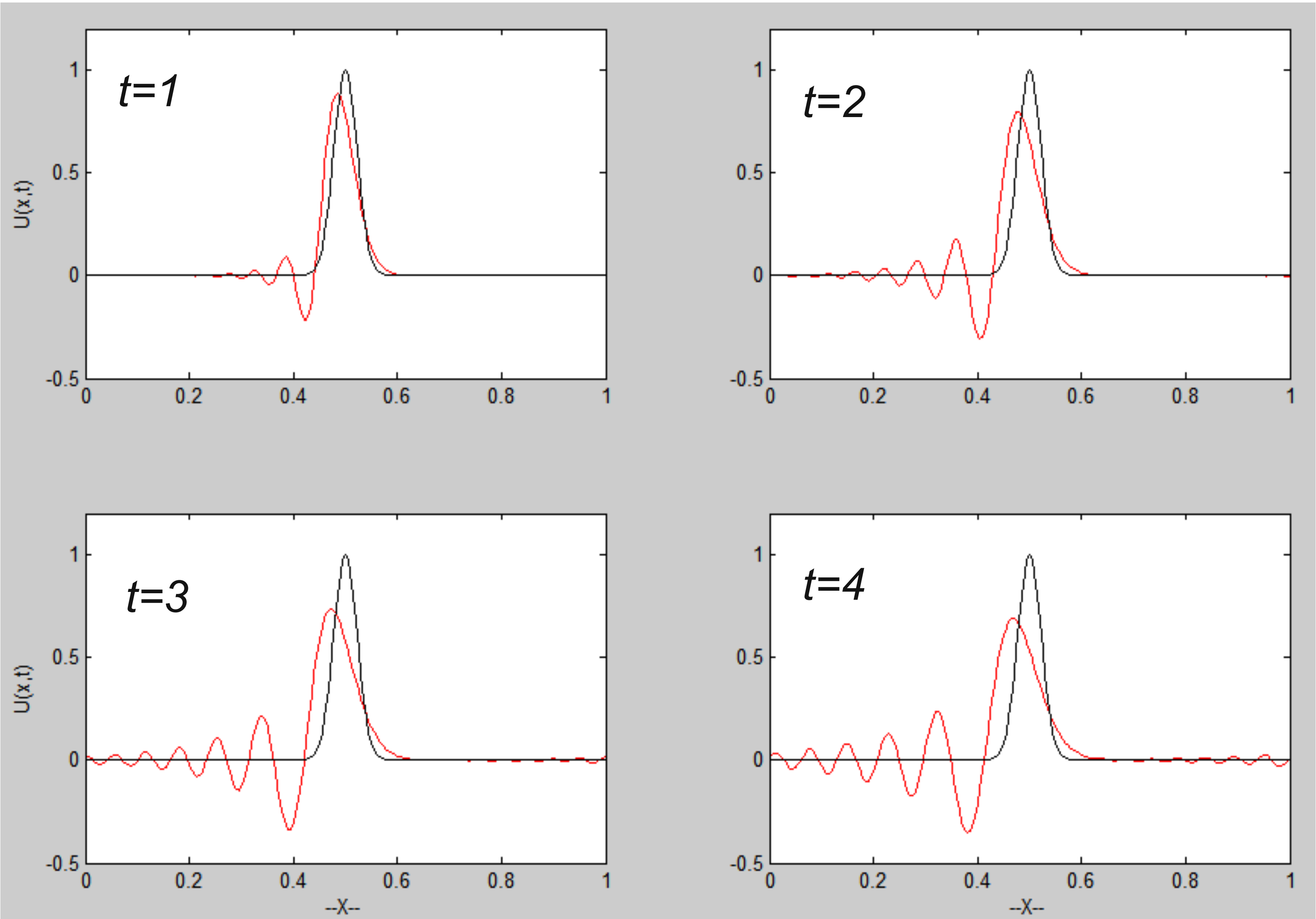
- This result is not realized numerically, especially for low-order discretizations.

- Although the initial condition (black) is well-resolved with n=200 points, the 2^{nd}-order solution exhibits trailing waves (red) even after one revolution.



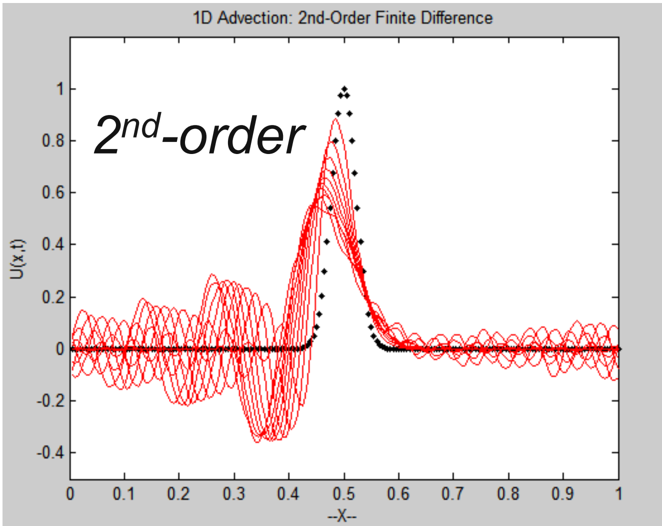
Numerical Dispersion, 2nd-order Spatial Discretization

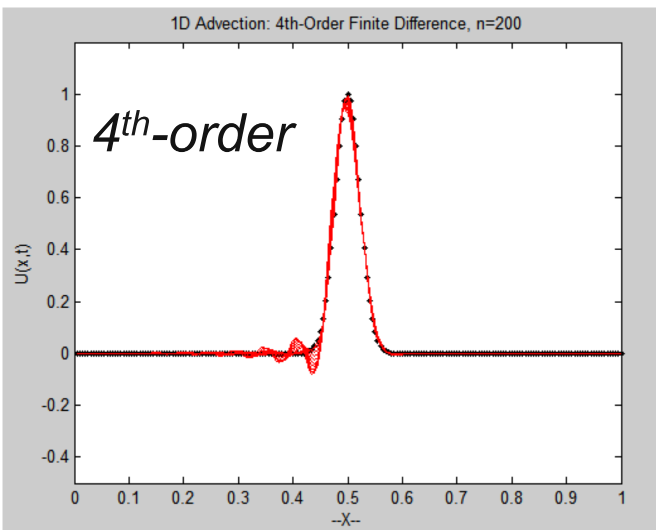
- At later times, the dispersion just becomes worse…



Cumulative Dispersion at t=10 for Varying Order & Resolution

Finite Difference, n=200





SEM, n=90