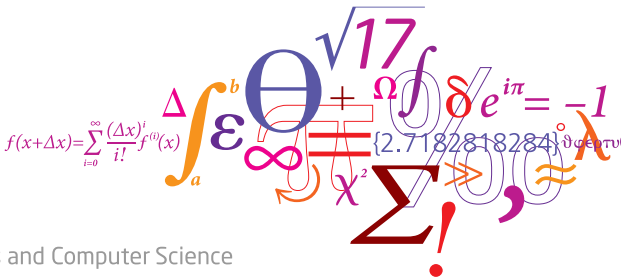
Introduction to Spectral Methods

02689 Advanced Numerical Methods for Differential Equations

Allan Peter Engsig-Karup
Scientific Computing Section
DTU Compute
Technical University of Denmark



DTU Compute
Department of Applied Mathematics and Computer Science

Overview

01 Introduction and motivation for the use of spectral methods
Spectral Approximation of Periodic Functions

02 Assignment #1 work

03 Spectral Approximation of Non-Periodic Functions

04 Assignment #1 work

---

05 Boundary Value Problems in 1D/2D

06 Time-integration methods

07 Non-smooth and nonlinear problems

08 Assignment #2 work

---

09 Introduction to Continuous / Discontinuous Galerkin Methods

10 More on Continuous / Discontinuous Galerkin Methods

11 Introduction to Mesh generation

12 Introduction to Scientific Machine Learning

13 Assignment #3 work

History

Important developments in numerical methods and techniques

Methods	**Theoretical developments**	**Industry**
Finite Difference Methods	1950s	1960s
Boundary Element Methods	1960s	1970s
Finite Volume Methods	1960s	1970s
Finite Element Methods	1970s	1980s
Meshless Methods	1970s	2000s-
Spectral Methods	1970s	1990s
Spectral Element Methods	1980s	1990s-
Hybrid Methods	1980s	2000s-
Scientific Machine Learning	1990s	2020s-

Milestones

- 1967 The Fast Fourier Transform was described by Cooley and Turkey.
- 1972 The Pseudospectral Method was first proposed by Kreiss and Oliger.

Motivation for using numerical methods

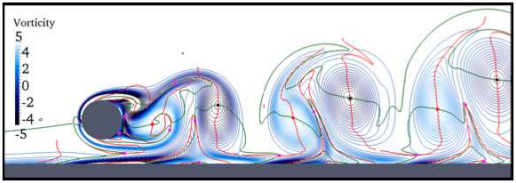


Figure: Analysis of vortex structures behind a cylinder using a Spectral/hp Element solver (M.Sc. Rasmus E. Christensen, DTU Compute)

Why solve differential equations using numerical methods?

- Find approximate solutions to differential equations.
- Difficult to solve by other means, e.g., due to problems with complicated boundaries.
- Verify correctness of an analytical solution.
- Get insight into the behaviour and sensitivity of solutions.
- Validate models and/or suggest theoretical improvements.
- Perform cost-efficient numerical experiments to replace or complement physical experiments.
- Develop data-driven numerical methods for fast predictions and complex analysis.

Coarse goal and Learning objectives (brief)

The course goal is to give a solid grounding in theory and practice in the use of numerical methods for efficient and robust solution of differential equations in science and engineering. A student who has met the objectives of the course will be able to:

- Apply basic principles for numerical approximation/discretization of spectral methods.
- Apply Fourier and Polynomial Transforms.
- Analyze convergence and stability properties of spectral methods.
- Setup and solve boundary value problems for partial differential equations.
- Implement methods for time integration of semi-discrete equation systems.
- Implement and apply algorithms for periodic and non-periodic problems.
- Construct spectral approximations for partial derivatives.
- Implement spectral methods in Python/Matlab/Julia.

Practical concerns with numerical methods for PDEs

Low-order numerical methods based on finite differences, finite volume and finite element methods are (still!) today by far the most widely used methods in industry where essential requirements are

- Sufficiently accurate predictions (Accuracy)
- Acceptable computational costs (Efficiency)
- Acceptable human costs (Productivity)
- Fast turn around (Cost-efficiency)

High-order methods are often critiqued for being

- Costly
- Less robust
- Complicated to understand
- Complicated to code

Important application areas

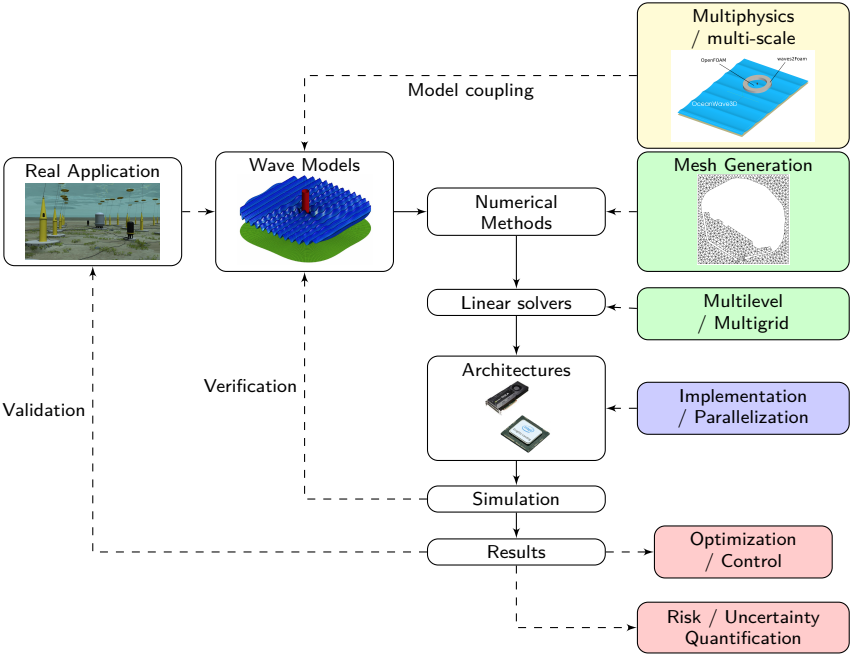
Advanced numerical methods (fx. high-order accurate spectral methods) are today increasingly used and in widespread use in applications for

- Wave propagation problems
  (fluid flow, acoustics, elastic, seismic, and electromagnetic)
- Plasma physics modelling
- Weather forecasting
- Mathematical analysis (uncertainty quantification, machine learning, model order reduction, etc.)
- and much more....

Problems in these areas are characterized by a need to resolve varying length and time scales of important physical processes.

Many applications can benefit from the improved accuracy and required computational resources (e.g. less computational time and reduced memory requirements) that are offered by advanced and high-order numerical methods to achieve a certain level of accuracy.

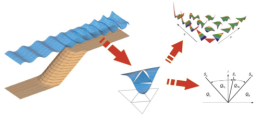
Development of simulation tools



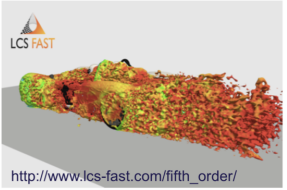
Scientific computing have major impact across sciences and engineering

SPECTRAL/*hp* ELEMENT METHODS

- *p*-type finite elements due to Babuska et al (1981)
- Spectral element method (SEM) due to Patera (1984)
- Spectral/*hp* Element Methods for CFD (Karniadakis and Sherwin, 1999)
- Nodal high-order DG methods (Hesthaven and Warburton, 2008)
- Today – frameworks (nektar++, nek5000, fenics, deal.II, etc)
- Often used for DNS (uDNS/ILES) simulations of bluff bodies
- Next generation NWP models are high-order DG
- Industrial engineering applications emerging now

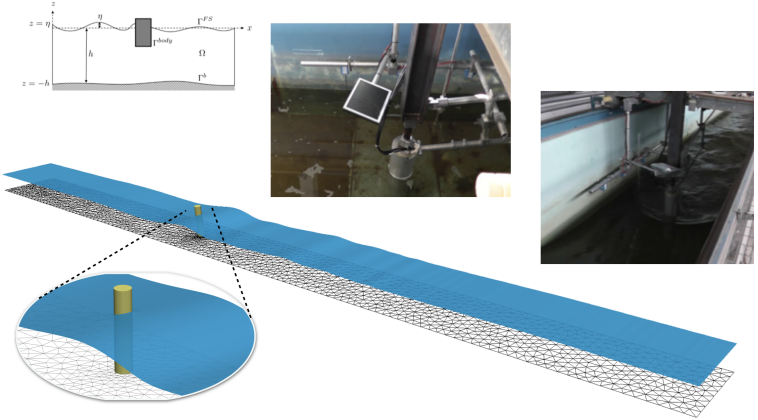


NEKTAR++
SPECTRAL/HP ELEMENT FRAMEWORK



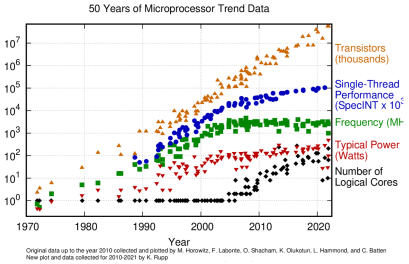
- Increased modelling fidelity using high-order numerical methods.
- High-order methods can be significantly more cost-efficient than low-order methods
- High-order methods maps to modern many-core hardware for improved performance

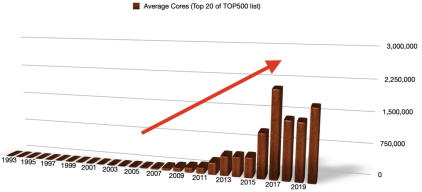
Scientific computing have major impact across sciences and engineering



- Numerical Wave Tank based on state-of-the-art Spectral Element Method.
- Numerical simulation of focusing wave group impacting a fixed truncated cylinder.
- Increasing adoption of digital twins where real world processes are linked with cost-efficient virtual models.

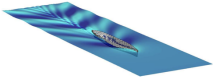
Scientific computing have major impact across sciences and engineering





- In multicore era : Performance scales with number of cores (= software problem)
- Trend in increasing number of cores clear from Top 500 super computers.

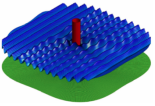
Scientific computing have major impact across sciences and engineering

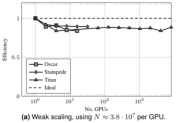


OceanWave3D (Open Source) Software

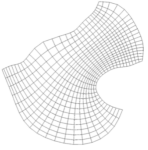
DTU Compute GPULAB C/C++ Library & OceanWave3D-GPU

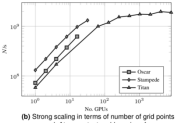
- ✦ FNPF solver developed since 2009 at DTU Compute.
- ✦ Massively parallel multigrid solver.
- ✦ CUDA/OpenCL + MPI implementation.
- ✦ Curvilinear Multi-block implementation.
- ✦ Benchmarked on desktops and super clusters **>1.000.000.000 DOFs** in linear solver.
- ✦ Fast large-scale simulations in marine regions with variable bathymetry using FNPF.
- ✦ Massively scalable simulations on heterogeneous clusters using **DTU Compute GPULAB Library**



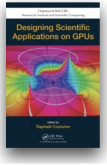


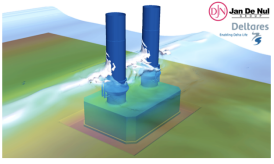
(a) Weak scaling, using $N \approx 3.8 \cdot 10^7$ per GPU.

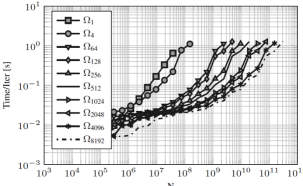




(b) Strong scaling in terms of number of grid points per second. At a constant problem size of $N \approx 3.8 \cdot 10^7$.

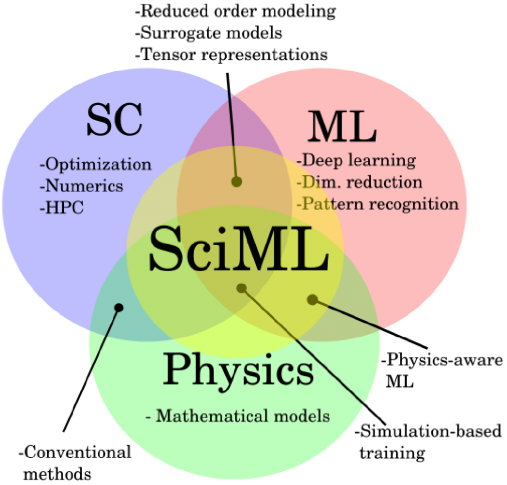






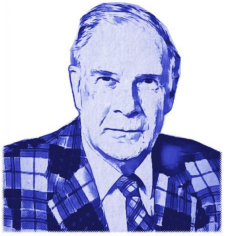
Website: http://www2.compute.dtu.dk/~apek/OceanWave3D/

Scientific Machine Learning on the rise



- Areas of statistics, data-driven machine learning method and mathematical physics growing research area.
- Many opportunities for improving the scientific computing toolbox.

Theoretical insights and modern scientific computing give opportunities...



"The purpose of computing is insight, not numbers."

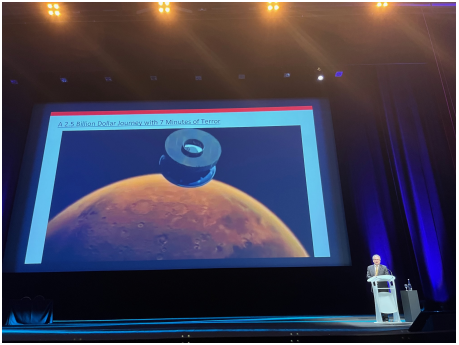
- Richard W. Hamming (1915–1998),
Numerical Methods for Scientists and Engineers (1962).

For example, find out how to

- enable much faster or more cost-efficient calculations.
- produce new tools for entire engineering areas.
- perform analysis with high accuracy and efficiency.
- make predictions that support decision making subject to uncertainty.
- ...

A 2.5 Billion Dollar Problem with 7 minutes of terror...

A plenary talk by Prof. Farhat Charbel, Stanford, at the 22nd Computational Fluids Conference 2023



Modeling, Simulation and Validation of Supersonic Parachute Inflation Dynamics during Mars Landing

Daniel Z. Huang, Philip Avery, Charbel Farhat, Jason Rabinovitch, Armen Derkevorkian and Lee D. Peterson

AIAA 2020-0313
Session: Experimental-Computational High-Speed FSI I

Published Online: 5 Jan 2020 • https://doi.org/10.2514/6.2020-0313

Abstract:

A high fidelity multi-physics Eulerian computational framework is presented for the simu- lation of supersonic parachute inflation during Mars landing. Unlike previous investigations in this area, the framework takes into account an initial folding pattern of the parachute, the flow compressibility effect on the fabric material porosity, and the interactions between super- sonic fluid flows and the suspension lines. Several adaptive mesh refinement (AMR)-enabled, large edge simulation (LES)-based, simulations of a full-size disk-gap-band (DGB) parachute inflating in the low-density, low-pressure, carbon dioxide (CO2) Martian atmosphere are re- ported. The comparison of the drag histories and the first peak forces between the simulation results and experimental data collected during the NASA Curiosity Rover's Mars atmospheric entry shows reasonable agreements. Furthermore, a rudimentary material failure analysis is performed to provide an estimate of the safety factor for the parachute decelerator system. The proposed framework demonstrates the potential of using Computational Fluid Dynam- ics (CFD) and Fluid-Structure Interaction (FSI)-based simulation tools for future supersonic parachute design.

How to derisk expensive supersonic parachute inflation during Mars landing when it is difficult to do experiments on Earth. Simulations can be the answer.

Reference: Modeling, Simulation and Validation of Supersonic Parachute Inflation Dynamics during Mars Landing (2020)

Sustainability in Scientific Computing

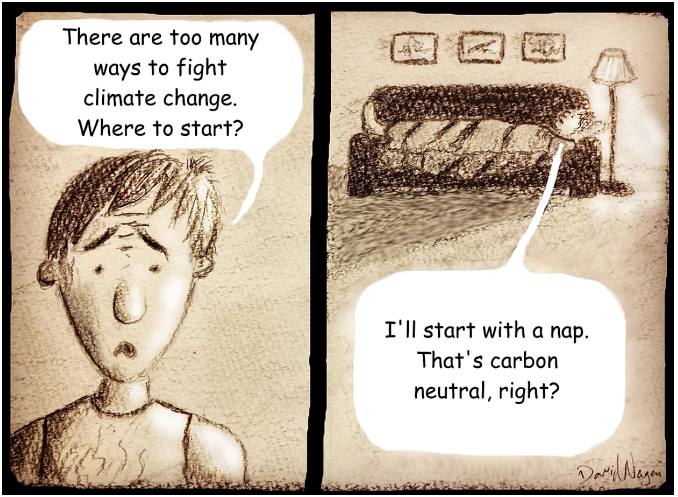
In this course we may even seek illuminating and understanding better means for meeting sustainable development goals through Scientific Computing approaches and reporting.

Scientific Computing have increasing impact on sustainable development goals

- Computing in data centers impact CO2e consumption.
- Efficient algorithms and computing approaches for reduced/optimized computational footprints.
- Smart algorithms for intelligent ressource optimization, etc.

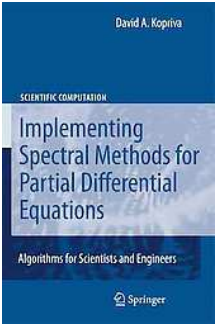
Also, for the last assignment projects, this course gives the student an opportunity to prepare a project that may participate in DTU's Study Conference on sustainability, climate technology, and the environment (GRØN DYST). More information http://www.groendyst.dtu.dk/english.

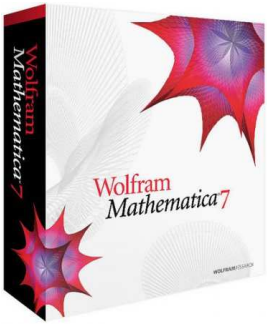
How to solve the clima problem?

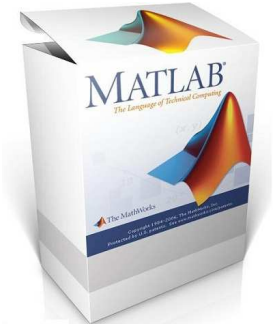


Material

Make good use of...

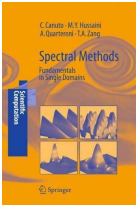


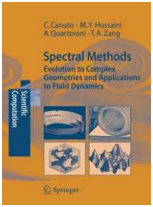


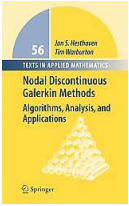


- Book: David A. Kopriva (2009) Implementing Spectral Methods for Partial Differential Equations - Algorithms for Scientists and Engineers, Springer.
- Symbolic manipulation: Mathematica/Maple/Sage.
- Programming environment: Python/Matlab/Julia.
- Slides

Recommended Supplementary Material







Theory of Spectral Methods

- Book: C. Canuto, M. Y. Hussaini, A. Quarteroni and T. A. Zang (2006) Spectral Methods: Fundamentals in Single Domains, Springer.
- Book: C. Canuto, M. Y. Hussaini, A. Quarteroni and T. A. Zang (2007) Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics, Springer.
- Book: Jan S. Hesthaven, Sigal Gottlieb and David Gottlieb (2007) Spectral Methods for Time-Dependent Problems, Cambridge.

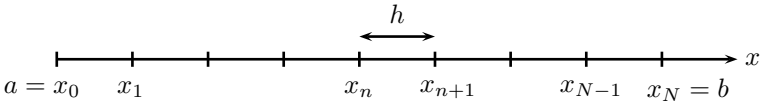
Theory and algorithms for Discontinuous Galerkin Finite Element Methods

- Book: J. S. Hesthaven and T. Warburton (2007) Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications, Springer.

The Finite Difference Method

- from local to global approximations

From local to global approximations



We can introduce a uniform grid $a \leq x \leq b$ consisting of the nodes

$$x_j = jh, \quad h = \frac{(b-a)}{N}, \quad j = 0, 1, ..., N$$

On this grid, we can represent a function $U_n = u(x_n)$ and seek the approximations $U_n' \approx u'(x_n)$, $n = 0, 1, ..., N$.

Finite Difference Operators

It can be useful to define some linear operators for deriving finite difference approximations of derivatives of a smooth function $u(x)$. The following operators seeks to approximate $u'(\bar{x})$ from known function values of $u(x)$

$$
\begin{aligned}
D_+u(\bar{x}) &\equiv \frac{u(\bar{x}+h)-u(\bar{x})}{h} \quad \text{(Forward)} \\
D_-u(\bar{x}) &\equiv \frac{u(\bar{x})-u(\bar{x}-h)}{h} \quad \text{(Backward)} \\
D_0u(\bar{x}) &\equiv \frac{u(\bar{x}+h)-u(\bar{x}-h)}{2h} = \frac{1}{2}(D_+u(\bar{x})+D_-u(\bar{x}))
\end{aligned}
$$

Usage of D_- and D_+ leads to one-sided approximations and D_0 to a centered approximation. Remark, in the asymptotic limit

$$
\lim_{h\to 0} D_\pm u(\bar{x}) = \lim_{h\to 0} D_0 u(\bar{x}) = u'(\bar{x})
$$

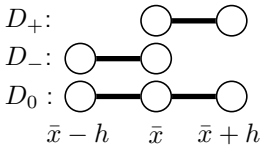
Finite Difference Operators

For finite difference approximations of the form

$$\frac{d^m u(\bar{x})}{dx^m} \approx \sum_{n=-\alpha}^{\beta} a_n u(\bar{x} + hn)$$

we have $r = \alpha + \beta + 1$ coefficients (weights) a_m to be used on a uniform mesh with constant grid spacing h.

We can illustrate the approximations based on $D_\pm$ and D_0 using stencil's based on a grid with spacing h as



Each stencil indicates which neighbouring function values that will be used in the weighted approximation.

Order of accuracy

$$\frac{d^m u(\bar{x})}{dx^m} = \sum_{n=-\alpha}^{\beta} a_n u(\bar{x} + hn) + \mathcal{O}(h^p)$$

The concept of *order of accuracy* is related to the asymptotic behavior when $h \to 0$ of the local truncation errors of the finite difference approximation.

- h is a measure of the grid spacing.
- An approximation to a function is p'th order accurate if the leading order term in the error function is $\mathcal{O}(h^p)$ when $h \to 0$.

$$e(x) = \frac{d^m u(x)}{dx^m} - \sum_{n=-\alpha}^{\beta} a_n u(x + hn) = \mathcal{O}(h^p)$$

A numerical approximation is high-order accurate if $p > 2$.

Order of accuracy

Analysis can be based on

- The truncation error is defined in terms of the exact solution $u(x)$ and can be determined by replacing $u(\bar{x} + hn)$ in a stencil with Taylor expansions about $\bar{x}$.

$$u(x_i + mh) = \sum_{n=0}^{\infty} \frac{(mh)^n}{n!} \frac{\partial^n u}{\partial x^n}\Big|_{x=x_i}$$

- For h sufficiently small, the error will be dominated by the leading order term in the error.

Better accuracy can be achieved through refinement in two ways

- Increase stencil to enable higher order approximations (p-convergence)
- Increase number of nodes (h-convergence)

Note: Optimal refinement path may be a combination.

A general approach

To compute finite difference stencils for the approximation of $u^{(k)}(\bar{x})$ evaluated at $\bar{x}$, based on an arbitrary stencil of n points, x_i, $i = 1, 2, ..., n$.

Assuming that $u(x)$ is sufficiently smooth, write the Taylor series expansion of $u(x)$ about $\bar{x}$ for each point in the grid x_i as

$$u(x_i) = u(\bar{x}) + (x_i - \bar{x})u'(\bar{x}) + \cdots + \frac{1}{k!}(x_i - \bar{x})^k u^{(k)}(\bar{x}) + \cdots$$

We want an approximation to the k'th derivative where the order p is as high as possible, i.e. require

$$c_1 u(x_1) + c_2 u(x_2) + \cdots + c_n u(x_n) = u^{(k)}(\bar{x}) + \mathcal{O}(h^p)$$

Note: h is some average grid spacing or for uniform grids the constant grid spacing.

A general approach

By the Method of undetermined coefficients we need to solve for the unknown coefficients by solving the coupled order conditions

$$\frac{1}{(i-1)!}\sum_{j=1}^{n} c_j (x_j - \bar{x})^{(i-1)} = \begin{cases} 1 & \text{, if } i-1=k \\ 0 & \text{, otherwise} \end{cases}$$

for $i = 1, 2, ..., N$. This system of equations can be written compactly as

$$A\boldsymbol{c} = \boldsymbol{e}_{i-1}, \quad A_{ij} = \frac{1}{(i-1)!}(x_j - \bar{x})^{(i-1)}$$

where $A \in \mathbb{R}^{N\times N}$ is a Vandermonde matrix and $\boldsymbol{e}_i$ is the i'th unit vector. See `fdcoeffV.m`.

A numerical stable and more efficient algorithm for calculating the coefficients is `fdcoeffF.m`.
(Provided at website for textbook by Leveque (2007))

Finite Difference Stencils

	Index (m) for collocation nodes							
Order of accuracy	-4	-3	-2	-1	0	1	2	3
2				$-\frac{1}{2}$	0	$\frac{1}{2}$		
4			$\frac{1}{12}$	$-\frac{3}{2}$	0	$\frac{2}{3}$	$-\frac{1}{12}$	
6		$-\frac{1}{60}$	$\frac{3}{20}$	$-\frac{3}{4}$	0	$\frac{3}{4}$	$-\frac{3}{20}$	$\frac{1}{60}$
1					-1	1		
2					$-\frac{2}{3}$	2	$-\frac{1}{2}$	
3					$-\frac{11}{6}$	3	$-\frac{3}{2}$	$\frac{1}{3}$
3				$\frac{1}{3}$	$-\frac{1}{2}$	1	$-\frac{1}{6}$	
4				$-\frac{1}{4}$	$-\frac{5}{6}$	$\frac{3}{2}$	$-\frac{1}{2}$	$\frac{1}{12}$
4	$\frac{1}{4}$	$-\frac{4}{3}$	3	-4	$\frac{25}{12}$			

Table: First derivative approximations at x=0. All coefficients needs to be scaled with $\frac{1}{h}$.

Finite Difference Stencils

	Index (m) for collocation nodes							
Order of accuracy	-4	-3	-2	-1	0	1	2	3
2				1	-2	1		
4			-$\frac{1}{12}$	$\frac{4}{3}$	-$\frac{5}{2}$	$\frac{4}{3}$	-$\frac{1}{12}$	
6		$\frac{1}{90}$	-$\frac{3}{20}$	$\frac{3}{2}$	-$\frac{49}{18}$	$\frac{3}{2}$	-$\frac{3}{20}$	$\frac{1}{90}$
1					1	-2	1	
2				1	-2	1	0	
2					2	-5	4	-1
3				$\frac{11}{12}$	-$\frac{5}{3}$	$\frac{1}{2}$	$\frac{1}{3}$	-$\frac{1}{12}$
4	$\frac{11}{12}$	$-\frac{14}{3}$	$\frac{19}{2}$	$-\frac{26}{3}$	$\frac{35}{12}$			

Table: Second derivative approximations at x=0. All coefficients needs to be scaled with $\frac{1}{h^2}$.

From local to global approximations

In the following we will make a convenient change in notation and assume

$$x_j = \bar{x}, \quad x_{j+n} = \bar{x} + hn$$

Consider simple case

$$\left.\frac{du}{dx}\right|_{x_j} \approx \frac{u_{j+1} - u_{j-1}}{2h}$$

in which case

$$a_1 = \frac{1}{2h}, \quad a_{-1} = -\frac{1}{2h}$$

This works well for smooth problems when $h << 1$.

From local to global approximations

On a periodic grid ($x_0 = x_N$) with grid spacing $h = \frac{b-a}{N}$ we can compute the first derivative approximations using a matrix vector product of the form

$$\begin{bmatrix} u'(x_0) \\ u'(x_1) \\ \vdots \\ u'(x_{N-1}) \end{bmatrix} \approx \frac{1}{2h} \begin{bmatrix} 0 & -1 & & & 1 \\ 1 & 0 & -1 & & \\ & \ddots & \ddots & \ddots & \\ -1 & & & 1 & 0 \end{bmatrix} \begin{bmatrix} u(x_0) \\ u(x_1) \\ \vdots \\ u(x_{N-1}) \end{bmatrix}$$

or in compact form

$$u'(\mathbf{x}) \approx D_0 u(\mathbf{x})$$

with D_0 a sparse $N \times N$ differentiation matrix.

If $h \ll \mathcal{O}(1)$ it can be costly! What can be done??

From local to global approximations

We have options

- Increase order of approximations
  (larger stencils = more work per grid point, faster convergence)
- Increase h
  (less points=less work, less accuracy)

Will there be any advantages here?

Let's generalize the approach for approximating a first order derivative using finite differences.

From local to global approximations

The general $2p$'th order accurate central scheme for approximating a first derivative can be expressed in the form

$$\frac{du}{dx}\Big|_{x_j} \approx \sum_{n=1}^{m} \alpha_n^m D_n$$

where

$$\alpha_n^m = -2(-1)^n \frac{(m!)^2}{(m-n)!(m+n)!}, \quad D_n = \frac{u_{j+n} - u_{j-n}}{2nh}$$

What happens when $m \to \infty$? (infinite order FD scheme)

- Stencil increases
- Accuracy increases
- How about work effort?

From local to global approximations

Let's express stencil as

$$\frac{du}{dx}\Big|_{x_j} = \sum_{n=1}^{m} \alpha_n^m \frac{u_{j+n} - u_{j-n}}{2nh}$$

and assume N even and defining a grid $x_j = jh$, $j = 0, 1..., N$ with spacing $h = \frac{2\pi}{N+1}$. Then

$$\frac{du}{dx}\Big|_{x_j} = \frac{N+1}{4\pi}\left(\sum_{n=1}^{m} \frac{\alpha_n^m}{n} u_{j+n} + \sum_{n=-m}^{-1} \frac{\alpha_n^m}{n} u_{j+n}\right)$$

The stencils are central and anti-symmetric, i.e. $\alpha_n^m = \alpha_{-n}^m$. Introduce

$$\beta_n^m = \begin{cases} \alpha_n^m, & n \neq 0 \\ 0, & n = 0 \end{cases}$$

to enable the expression to be written in a simpler form

$$\frac{du}{dx}\Big|_{x_j} = \frac{N+1}{4\pi} \sum_{n=-m}^{m} \frac{\beta_n^m}{n} u_{j+n}$$

From local to global approximations

Consider the limit $m \to \infty$. We find $\beta_n^\infty = (-2)(-1)^n$ such that

$$\frac{du}{dx}\Big|_{x_j} = \frac{N+1}{4\pi} \sum_{n=-\infty}^{\infty} \frac{-2(-1)^n}{n} u_{j+n}$$

If we assume the periodicity property for the function

$$u_j = u_{j+p(N+1)}, \quad p = ... -1, 0, 1, ...$$

the expression can by a change in index be expressed in the form

$$\frac{du}{dx}\Big|_{x_j} = \frac{N+1}{4\pi} \sum_{n=j}^{j+N} \sum_{p=-\infty}^{\infty} \frac{-2(-1)^{n+p(N+1)}}{n+p(N+1)} u_{j+n}$$

From local to global approximations

We recognize a closed-form expression for the infinite sum

$$\sum_{k=-\infty}^{\infty} \frac{(-1)^k}{k+x} = \frac{\pi}{\sin(\pi x)}$$

Thus, we find by change of index (recall definition of β_n^m)

$$\sum_{p=-\infty}^{\infty} \frac{(-1)^p}{p+n/(N+1)} = \begin{cases} 0, & n=0 \\ \frac{\pi}{\sin\left(\pi \frac{n}{N+1}\right)}, & n \neq 0 \end{cases}$$

Thus,

$$\left.\frac{du}{dx}\right|_{x_j} = \sum_{n=j}^{j+N} -\frac{1}{2}(-1)^n \left[\sin\left(\frac{\pi n}{N+1}\right)\right]^{-1} u_{j+n}$$

From local to global approximations

Finally, by changing index to $i = j + n$ we find

$$\frac{du}{dx} = Du, \quad D_{ij} = \begin{cases} -\frac{1}{2}(-1)^{i-j}\left[\sin\left(\frac{\pi(i-j)}{N+1}\right)\right]^{-1} & , i \neq j \\ 0 & , i = j \end{cases}$$

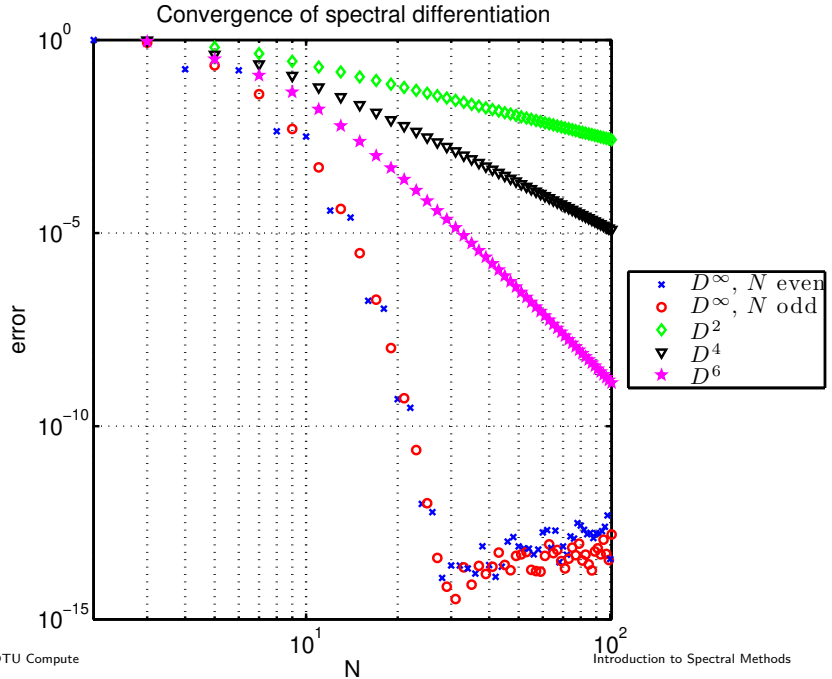
with D a dense discrete pseudospectral differentiation matrix for the first derivative of an interpolant taken on the grid.

Now

- The method is a global method that uses all available info.
- The differentiation matrix went from sparse to dense.
  (work per node increased!)

Is it useful?

From local to global approximations



From local to global approximations

We have seen that we need different numbers of points to achieve same tolerance level of, say, 1e-5

- 2nd order finite difference > 1000 grid points
- 4th order finite difference ~ 300 grid points
- Infinite order finite difference ~ 15 grid points

So, the more information is used the more accurate our approximations... however, amount of work (flops) required in each case is different.

From local to global approximations

Now, consider the simple wave equation

$$\frac{\partial u}{\partial t} = -c\frac{\partial u}{\partial x}, \quad 0 \leq x \leq 2\pi$$

with exact solution $u(x,t) = f(x - ct)$. For example, choose $f(y) = e^{-\sin(y)}$.

This equation can be solved using a Method of Lines (MoL) discretization approach.

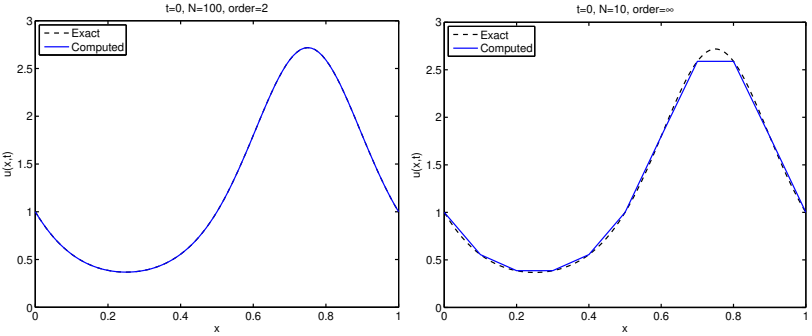
The spatial discretization is done using symmetric finite difference approximations of the form

$$\frac{\partial u}{\partial x}\Big|_{x_j} \approx \sum_{n=-\alpha}^{\alpha} c_{j+n} u_{j+n}$$

The temporal discretization can be done using a classical 4th order Runge-Kutta Method with sufficiently small time step to ensure stability.

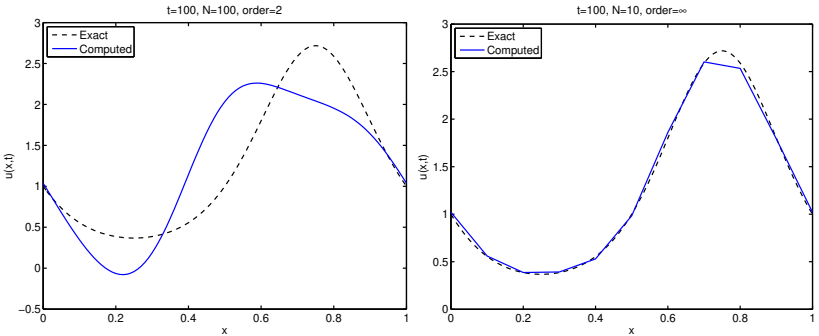
From local to global approximations

We could discretize and solve this set of equation to observe the behavior of the solutions.

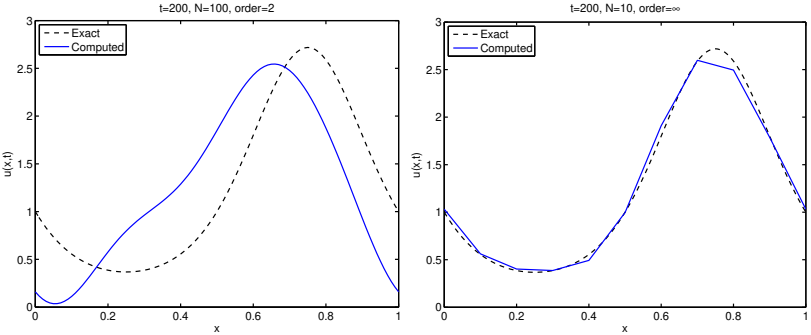


We use same initial condition for a second order method with $N = 100$ nodes and for a pseudospectral collocation method with $N = 10$ nodes.

From local to global approximations



From local to global approximations



From local to global approximations

These observations suggest that there might be an advantage to using higher order numerical methods

- Better accuracy for pseudospectral method
- Less points for pseudospectral method
- Better work effort? (not measured)

Phase error analysis

To solve the wave equation

$$\frac{\partial u}{\partial t} = -c\frac{\partial u}{\partial x}, \quad 0 \leq x \leq 2\pi$$

We seek to find a grid function $v(x)$ corresponding to a harmonic which satisfies the semi-discrete system

$$\frac{dv_i}{dt} = -c\sum_{j=1}^{N} D_{ij}v_j, \quad v_j(0) = e^{ikx_j}, \quad i = 1, 2, ..., N$$

If $v(x,t)$ satisfies any of these equations the solution will be

$$v(x,t) = e^{ik(x-c_m(k)t)}$$

where $c_m(k) \approx c$ is the numerical wave speed. The dependence of c_m on the wave number k is known as the dispersion relation.

Phase error analysis

The phase error $e_m(k)$ is defined as the leading term in a Taylor expansion of the the relative error between the exact solution $u(x,t)$ and the approximate solution $v(x,t)$

$$\left|\frac{u(x,t)-v(x,t)}{u(x,t)}\right| = |1-e^{ik(c-c_m(k))t}| \cong |k(c-c_m(k))t| = e_m(k)$$

With no differences in amplitude for the solution, the main error contribution will be the phase error of the schemes.

Let's analyze the phase errors for different schemes to try

- identifying the most efficient scheme satisfying a phase accuracy requirement over a specified period of time.

To determine the phase error vi need to determine the numerical phase speed $c_m(k)$.

Phase error analysis

We recognize that if the exact solution to the semi-discrete equation is given as

$$v(x,t) = e^{ik(x-c_m(k)t)}$$

then the rate of change of the solution is

$$\frac{d}{dt}v(x,t) = -ikc_m(k)e^{ik(x-c_m(k)t)} = -ikc_m(k)v(x,t)$$

We must also have

$$\frac{d}{dt}v(x,t) = -c\sum_{n=1}^{N} D_{ij}v_j = Sv(x,t), \quad S = -ic\sum_{j=1}^{m}\alpha_n^m\frac{\sin(nkh)}{nh}$$

Thus, the numerical phase speed is defined as

$$c_m(k) = -\frac{S}{ik} = c\sum_{n=1}^{m}\alpha_n^m\frac{\sin(nkh)}{nkh}$$

Phase error analysis

For a second order scheme

$$\frac{\partial v(x,t)}{\partial t} = -c\frac{v(x+h,t) - v(x-h,t)}{2h}, \quad v(x,0) = e^{ikx}$$

we find the numerical phase speed

$$c_1(k) = c\left(\frac{\sin(kh)}{kh}\right)$$

For $h << 1$ we can taylor expand the sine function such that

$$c_1(k) = c\left(1 - \frac{(kh)^2}{6} + \mathcal{O}((kh)^4)\right)$$

which shows that the scheme is second order accurate as expected.

Phase error analysis

The corresponding estimates of phase errors are

$$e_1(k,t) = kct\left|1 - \frac{\sin(kh)}{kh}\right|$$

$$e_2(k,t) = kct\left|1 - \frac{8\sin(kh) - \sin(2kh)}{6kh}\right|$$

$$e_3(k,t) = kct\left|1 - \frac{45\sin(kh) - 9\sin(2kh) + \sin(3kh)}{30kh}\right|$$

$$e_m(k,t) = kct\left|1 - \sum_{n=1}^{N} \alpha_n^m \frac{\sin(nkh)}{nkh}\right|$$

Clearly, phase errors depend on

- Mesh size h
- Length of time integration t

For a specific numerical scheme (i.e. choice of stencil) we are left with tuning h to achieve a desired accuracy level.

Phase error analysis

We can introduce a measure of how many points per wave length (p) to be used

$$p = \frac{l}{h} = \frac{2\pi}{kh} \quad \leftrightarrow \quad kh = \frac{2\pi}{p} = 2\pi p^{-1}$$

The length of the integration interval in time is also important.

We can measure this in terms of the number of wave periods elapsed

$$v = \frac{t}{T} = \frac{ct}{l} = \frac{kct}{2\pi} \quad \leftrightarrow \quad kct = 2\pi v$$

Phase error analysis

Rewriting phase errors in terms of p and v we obtain

$$e_1(p,v) = 2\pi v \left| 1 - \frac{\sin(2\pi p^{-1})}{2\pi p^{-1}} \right|$$

$$e_2(p,v) = 2\pi v \left| 1 - \frac{8\sin(2\pi p^{-1}) - \sin(4\pi p^{-1})}{12\pi p^{-1}} \right|$$

$$e_m(p,v) = 2\pi v \left| 1 - \sum_{n=1}^{m} \alpha_n^m \frac{\sin(n2\pi p^{-1})}{n2\pi p^{-1}} \right|$$

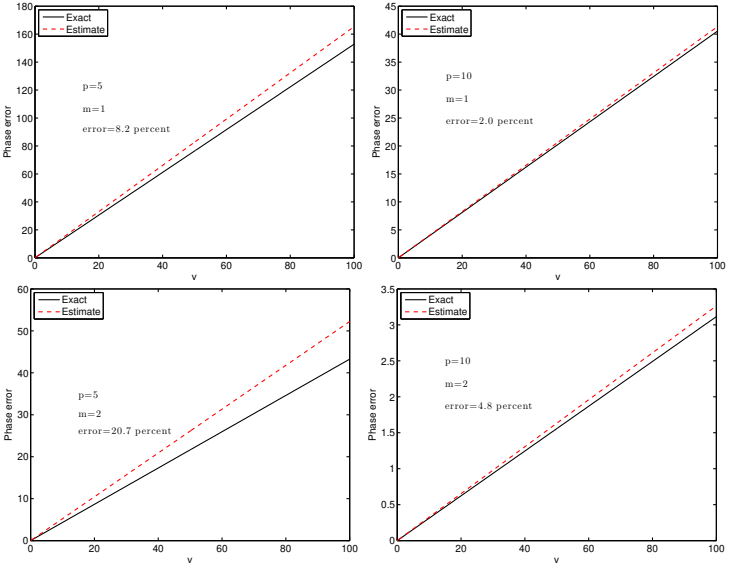
By expanding the sine function in a Maclaurin series it is possible to obtain the following asymptotic estimates (p sufficiently large)

$$e_1(p,v) \cong \frac{\pi v}{3}\left(\frac{2\pi}{p}\right)^2, \quad e_2(p,v) \cong \frac{\pi v}{15}\left(\frac{2\pi}{p}\right)^4$$

$$e_m(p,v) \cong \gamma^m \pi v \left(\frac{2\pi}{p}\right)^{2m}, \quad \gamma^m = \frac{2(-1)^{m+1}}{(2m+1)!}\left(\sum_{n=1}^{m} \alpha_n^m n^{2m}\right)$$

Phase error analysis

Testing accuracy of assumptions in phase error analysis



Phase error analysis

How many points per wave length is needed to guarantee a certain accuracy level ϵ?

Rewrite expressions for phase errors to obtain lower bounds on points per wave length

$$p_1(\epsilon, v) \geq 2\pi \sqrt{\frac{v\pi}{3\epsilon}}$$

$$p_2(\epsilon, v) \geq 2\pi \sqrt[4]{\frac{v\pi}{15\epsilon}}$$

$$p_m(\epsilon, v) \geq 2\pi \sqrt[2m]{\gamma^m \frac{v\pi}{\epsilon}}$$

How about estimating required work effort to achieve a certain accuracy level?

Phase error analysis

We can estimate the required work for time integration as

$$
\begin{aligned}
W_m(p_m, v) &\sim (\text{Work per time step}) \cdot (\text{Number of time steps}) \\
&= (\text{flops per point} \cdot \text{points per wave length}) \cdot (N_{steps}) \\
&= (2m \cdot p_m) \cdot \left(\frac{t}{\Delta t}\right) \\
&= \frac{2mvp_m^2}{CFL_m}
\end{aligned}
$$

where we have assumed a conditional CFL stability condition for the explicit time integration scheme of the form

$$
\Delta t \leq CFL_m \frac{h}{c_m} \quad \rightarrow \quad CFL_m = c_m \frac{\Delta t}{h}
$$

where CFL_m is selected according to conditional stability limit of chosen time integration method.

Phase error analysis

from which we obtain

$$\frac{t}{\Delta t} = \frac{vTc}{CFL_m h} = \frac{vl}{CFL_m h} = \frac{vp_m}{CFL_m}$$

For a fourth-order Runge-Kutta method, we can find

$$CFL_1 = 2.8, \quad CFL_2 = 2.1, \quad CFL_3 = 1.75$$

which leads to the following work estimates

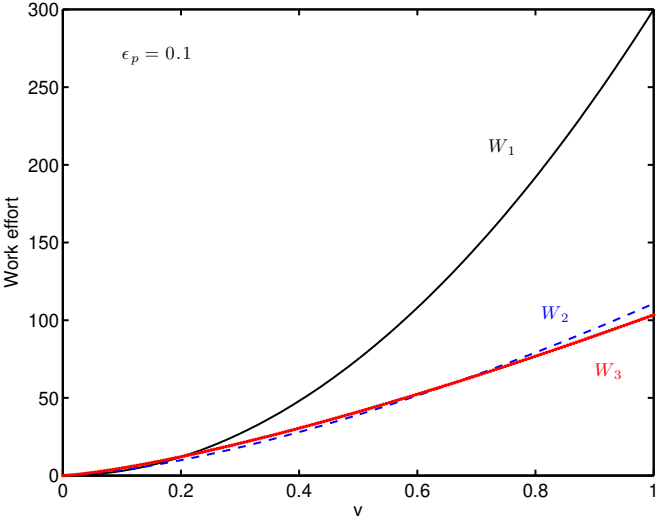
$$W_1 \sim 30v\frac{v}{\epsilon_p}$$

$$W_2 \sim 35v\sqrt{\frac{v}{\epsilon_p}}$$

$$W_m \sim \left(\frac{8m\sqrt[m]{\gamma^m}\pi^{(2+1/m)}}{CFL_m}\right) v\sqrt[m]{\frac{v}{\epsilon_p}}$$

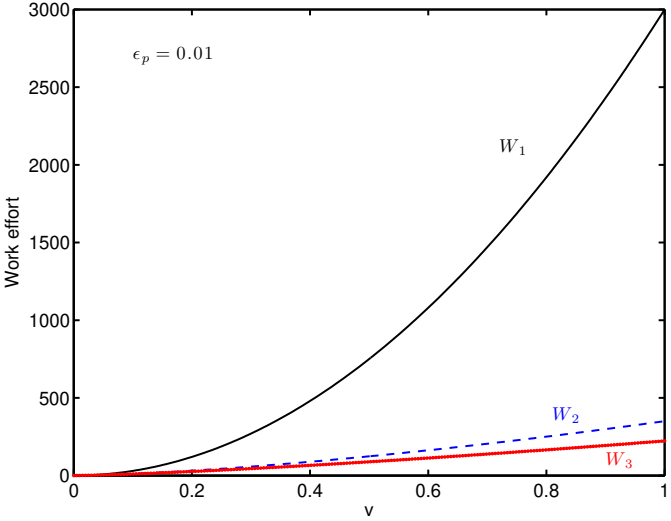
Phase error analysis

Work effort comparison for different finite difference schemes



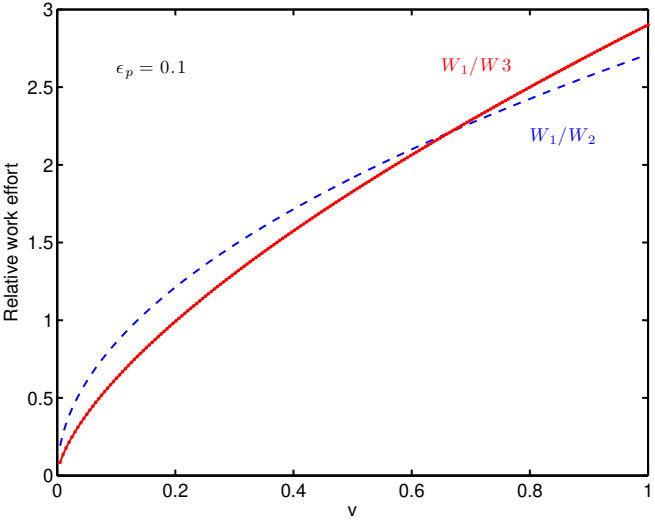
Phase error analysis

Relative work effort comparison for different finite difference schemes



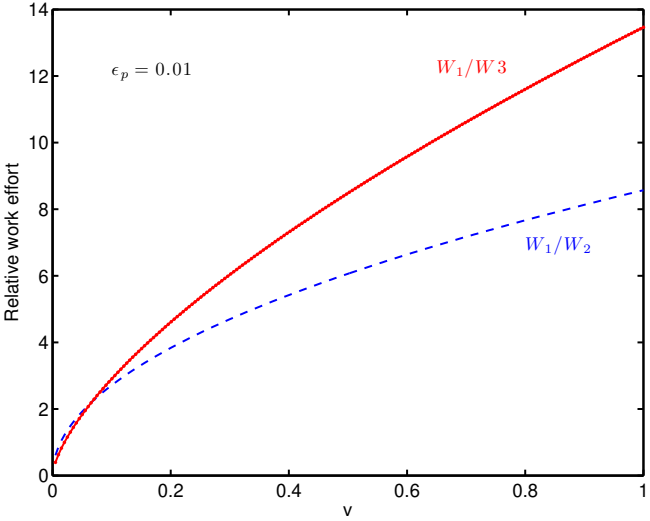
Phase error analysis

Work effort comparison for different finite difference schemes



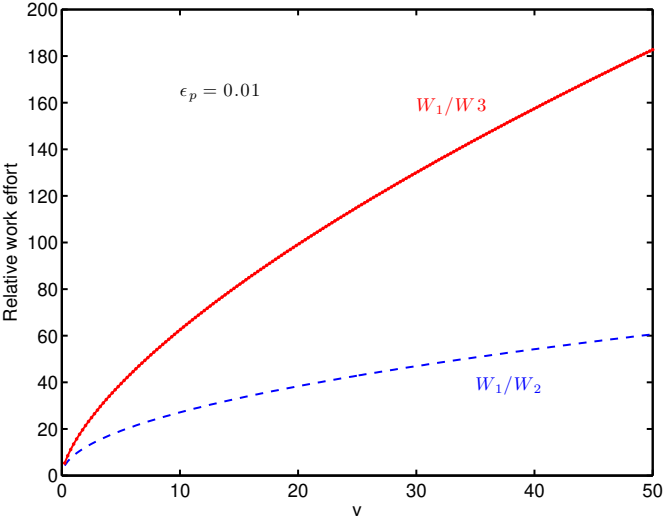
Phase error analysis

Relative work effort comparison for different finite difference schemes



Phase error analysis

Relative work effort comparison for different finite difference schemes



Phase error analysis in practice

$\alpha \setminus N_x$	10	20	40	80	160	320
1	1.3e+00	1.1e+00	1.2e+00	3.2e-01	8.1e-02	**2.0e-02**
2	1.4e+00	1.0e-01	**6.4e-03**	4.0e-04	2.5e-05	1.6e-06
3	1.2e-01	**2.1e-03**	3.4e-05	5.4e-07	8.5e-09	4.9e-10
4	1.0e-02	4.9e-05	3.4e-07	1.1e-08	2.9e-10	3.6e-10
5	9.4e-04	3.6e-06	1.6e-07	1.0e-08	2.8e-10	3.6e-10

Table: Max-norm amplitude error at $t = t_{final}/T = 50$. $C_r = 0.1$. RK4.

$\alpha \setminus N_x$	10	20	40	80	160	320
1	1.00	2.03	4.34	9.73	23.22	**60.50**
2	1.01	2.06	**4.49**	10.08	25.25	67.42
3	1.02	**2.16**	4.72	11.37	29.10	83.41
4	1.03	2.19	4.81	11.58	30.35	88.37
5	1.02	2.25	4.97	12.15	32.94	98.83

Table: Normalized CPU time to reach $t = t_{final}/T = 50$. $C_r = 0.1$. RK4.

Wave-based Spectral Element Method for Room acoustics

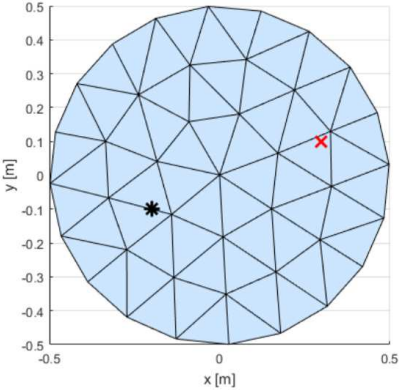


FIG. 8: Experiment setup. 2D mesh with with 60 elements, 521 DOF's (supporting P=4 basis functions). Source location is shown with a red cross $((s_x, s_y) = (0.3, 0.1))$ and the receiver location is shown with a black star $((r_x, r_y) = (-0.2, -0.1))$.

Source: Finnur, Mejling, Engsig-Karup, Jeong and Strømann-Andersen (2018), The Journal of the Acoustical Society of America Room acoustic simulations using the spectral element method.

Wave-based Spectral Element Method for Room acoustics

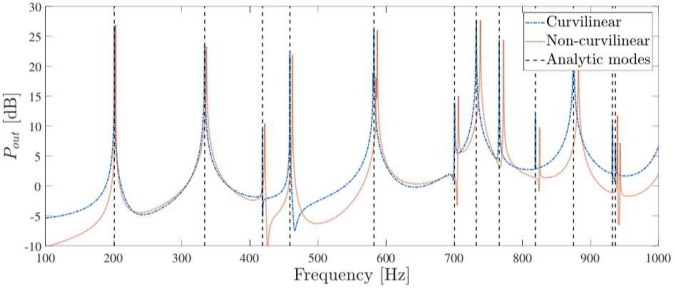


FIG. 8. (Color online) Simulated frequency response of a 2D circle, with and without curvilinear boundary elements. Analytic modes are dashed vertical lines, calculated using Green's function.[72] The source is at $(s_x, s_y) = (0.3, 0.1)$ m and the receiver is at $(r_x, r_y) = (-0.2, -0.1)$.

Source: Finnur, Mejling, Engsig-Karup, Jeong and Strømann-Andersen (2018), The Journal of the Acoustical Society of America Room acoustic simulations using the spectral element method.

Wave-based Spectral Element Method for Room acoustics

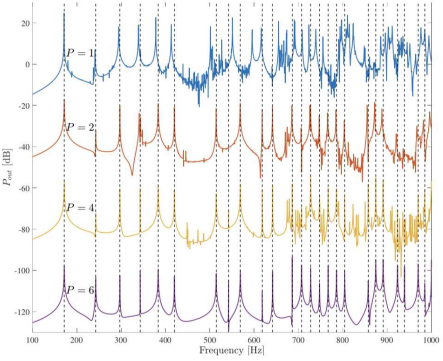


FIG. 10. (Color online) Simulated frequency responses for the different orders tested, while using fixed amount of DOF's on the mesh (15,625 nodes). Analytic modes are shown as dashed vertical lines.[72] The source location is $(s_x, s_y, s_z) = (0.25, 0.75, 0.60)$ and the receiver location is $(r_x, r_y, r_z) = (0.85, 0.30, 0.80)$.

Source: Finnur, Mejling, Engsig-Karup, Jeong and Strømann-Andersen (2018), The Journal of the Acoustical Society of America Room acoustic simulations using the spectral element method.

Ultimate goal for high-order methods

The ultimate goal for practical use of high-order methods is to

- Algorithmic viewpoint: Minimize errors for same number of nodes or degrees of freedom to achieve a certain accuracy level.
- Practical viewpoint: Minimize cost to achieve a solution of acceptable accuracy.

Effectively, this implies improving utilisation of computational resources in the pursue of improved turn around times for computations.

For practitioners a key concern is stated as

- Finding sufficiently accurate solutions to a given engineering problem in a reasonable amount of time.

High-order methods can be one ingredient that can support this goal in computations.