The Spectral-Element Method

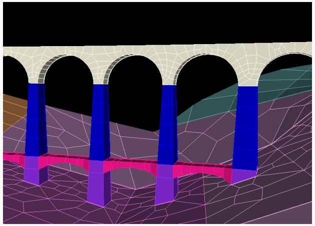
Heiner Igel

Department of Earth and Environmental Sciences
Ludwig-Maximilians-University Munich

Introduction

Motivtion

- **High accuracy** for wave propagation problems
- **Flexibility** with Earth model **geometries**
- **Accurate** implementation of **boundary conditions**
- **Efficient parallelization** possible



History

- Originated in **fluid dynamics** (Patera, 1984, Maday and Patera, 1989)
- First applications to **seismic wave propagation** by Priolo, Carcione and Seriani, 1994)
- Initial concepts around **Chebyshev** polynomials (Faccioli, Maggio, Quarteroni and Tagliani, 1996)
- Use of Lagrange polynomials (**diagonal mass matrix**) by Komatitsch and Vilotte (1998)
- Application to spherical geometry using the **cubed sphere** concept (Chaljub and Vilotte, 2004)
- Spectral elements in **spherical coordinates** by Fichtner et al. (2009)
- Community code "**specfem**" widely used for simulaton and inversion (e.g., Peter et al. 2011).

History

- Originated in **fluid dynamics** (Patera, 1984, Maday and Patera, 1989)
- First applications to **seismic wave propagation** by Priolo, Carcione and Seriani, 1994)
- Initial concepts around **Chebyshev** polynomials (Faccioli, Maggio, Quarteroni and Tagliani, 1996)
- Use of Lagrange polynomials (**diagonal mass matrix**) by Komatitsch and Vilotte (1998)
- Application to spherical geometry using the **cubed sphere** concept (Chaljub and Vilotte, 2004)
- Spectral elements in **spherical coordinates** by Fichtner et al. (2009)
- Community code "**specfem**" widely used for simulaton and inversion (e.g., Peter et al. 2011).

History

- Originated in **fluid dynamics** (Patera, 1984, Maday and Patera, 1989)
- First applications to **seismic wave propagation** by Priolo, Carcione and Seriani, 1994)
- Initial concepts around **Chebyshev** polynomials (Faccioli, Maggio, Quarteroni and Tagliani, 1996)
- Use of Lagrange polynomials (**diagonal mass matrix**) by Komatitsch and Vilotte (1998)
- Application to spherical geometry using the **cubed sphere** concept (Chaljub and Vilotte, 2004)
- Spectral elements in **spherical coordinates** by Fichtner et al. (2009)
- Community code "**specfem**" widely used for simulaton and inversion (e.g., Peter et al. 2011).

History

- Originated in **fluid dynamics** (Patera, 1984, Maday and Patera, 1989)
- First applications to **seismic wave propagation** by Priolo, Carcione and Seriani, 1994)
- Initial concepts around **Chebyshev** polynomials (Faccioli, Maggio, Quarteroni and Tagliani, 1996)
- Use of Lagrange polynomials (**diagonal mass matrix**) by Komatitsch and Vilotte (1998)
- Application to spherical geometry using the **cubed sphere** concept (Chaljub and Vilotte, 2004)
- Spectral elements in **spherical coordinates** by Fichtner et al. (2009)
- Community code "**specfem**" widely used for simulaton and inversion (e.g., Peter et al. 2011).

History

- Originated in **fluid dynamics** (Patera, 1984, Maday and Patera, 1989)
- First applications to **seismic wave propagation** by Priolo, Carcione and Seriani, 1994)
- Initial concepts around **Chebyshev** polynomials (Faccioli, Maggio, Quarteroni and Tagliani, 1996)
- Use of Lagrange polynomials (**diagonal mass matrix**) by Komatitsch and Vilotte (1998)
- Application to spherical geometry using the **cubed sphere** concept (Chaljub and Vilotte, 2004)
- Spectral elements in **spherical coordinates** by Fichtner et al. (2009)
- Community code "**specfem**" widely used for simulaton and inversion (e.g., Peter et al. 2011).

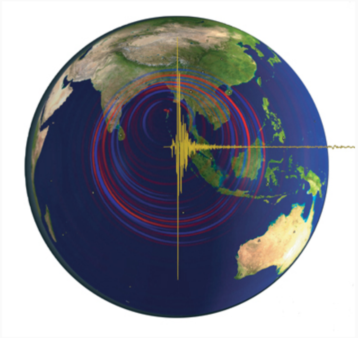
History

- Originated in **fluid dynamics** (Patera, 1984, Maday and Patera, 1989)
- First applications to **seismic wave propagation** by Priolo, Carcione and Seriani, 1994)
- Initial concepts around **Chebyshev** polynomials (Faccioli, Maggio, Quarteroni and Tagliani, 1996)
- Use of Lagrange polynomials (**diagonal mass matrix**) by Komatitsch and Vilotte (1998)
- Application to spherical geometry using the **cubed sphere** concept (Chaljub and Vilotte, 2004)
- Spectral elements in **spherical coordinates** by Fichtner et al. (2009)
- Community code "**specfem**" widely used for simulaton and inversion (e.g., Peter et al. 2011).

History

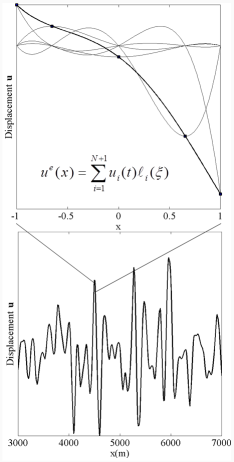
- Originated in **fluid dynamics** (Patera, 1984, Maday and Patera, 1989)
- First applications to **seismic wave propagation** by Priolo, Carcione and Seriani, 1994)
- Initial concepts around **Chebyshev** polynomials (Faccioli, Maggio, Quarteroni and Tagliani, 1996)
- Use of Lagrange polynomials (**diagonal mass matrix**) by Komatitsch and Vilotte (1998)
- Application to spherical geometry using the **cubed sphere** concept (Chaljub and Vilotte, 2004)
- Spectral elements in **spherical coordinates** by Fichtner et al. (2009)
- Community code "**specfem**" widely used for simulaton and inversion (e.g., Peter et al. 2011).

Specfem 3D



Logo of the wellknown community code with a snapshot of global wave propagation for a simulation of the devastating M9.1 earthquake near Sumatra in December, 2004.
The open-source code is hosted by the CIG project
(www.geodynamics.org).

Spectral Elements in a Nutshell



- Snapshot of the displacement field *u* during a simulation in a strongly **heterogeneous** medium.
- Zoom into the displacement field inside an element discretised with order N=5 **collocation points**
- Exact interpolate using **Lagrange polynomials**.
- **Gauss-Lobatto-Legendre Numerical integration** scheme.

1-D elastic wave equation

$$\boxed{\rho(x)\ddot{u}(x,t) = \partial_x\left[\mu(x)\partial_x u(x,t)\right] + f(x,t)}$$

u displacement

f external force

ρ mass density

μ shear modulus

Boundary condition

No traction perpendicular to the Earth's free surface

$$\sigma_{ij}\, n_j = 0$$

normal vector n_j, σ_{ij} is the symmetric stress tensor

$$\mu \partial_x u(x,t)|_{x=0,L} = 0$$

where our spatial boundaries are at $x = 0, L$ and the stress-free condition applies at both ends

Spectral Element: Essentials

- **Weak formulation** of the wave equation
- Transformation to the **elemental level** (Jacobian)
- Approximation of unknown function u using **Lagrange polynomials**
- Evaluation of the 1st derivatives of the Lagrange polynomials
- Numerical integration scheme based on **GLL quadrature**
- Calculation of **system matrices** at elemental level
- **Assembly** of global system of equations
- **Extrapolation** in time using a simple finite-difference scheme

Galerkin Principle

- The underlying principle of the finite-element method
- Developed in context with structural engineering (Boris Galerkin, 1871-1945)
- Also developed by Walther Ritz (1909) - variational principle
- Conversion of a continuous operator problem (such as a differential equation) to a discrete problem
- **Constraints on a finite set of basis functions**

Weak Formulations

Multiplication of pde with test function $w(x)$ on both sides.
G is here the complete computational domain defined with $x \in G = [0, L]$.

$$\int\limits_G w \, \rho \, \ddot{u} \, \mathrm{d}x - \int\limits_G w \, \partial_x \left(\mu \, \partial_x u\right) \mathrm{d}x = \int\limits_G w \, f \, \mathrm{d}x$$

Integration by parts

$$\int\limits_G w\,\rho\,\ddot{u}\,\mathrm{d}x + \int_G \mu\,\partial_x w\,\partial_x\,u\,\mathrm{d}x = \int\limits_G w\,f\,\mathrm{d}x$$

where we made use of the boundary condition

$$\partial_x u(x,t)|_{x=0} = \partial_x u(x,t)|_{x=L} = 0$$

The approximate displacement field

$$u(x,t) \approx \overline{u}(x,t) = \sum_{i=1}^{n} u_i(t)\, \psi_i(x).$$

- Discretization of space introduced with this step
- Specific basis function not yet specified
- In principle basis functions defined on entire domain (-> PS method)
- Locality of basis functions lead to finite-element type method

Global system after discretization

Use approximation of u in weak form and (!) use the same basis function as test function

$$\int_G \psi_i \rho \ddot{\overline{u}} \mathrm{d}x + \int_G \mu \partial_x \psi_i \partial_x \overline{u} \mathrm{d}x = \int_G \psi_i f \mathrm{d}x$$

with the requirement that the medium is at rest a t=0.

Including the function approximation for $u(x, t)$

... leads to an equation for the unknown coefficients $u_i(t)$

$$\sum_{i=1}^{n} \left[\ddot{u}_i(t) \int_{G_e} \rho(x)\, \psi_j(x)\, \psi_i(x)\, \mathrm{d}x \right] + \sum_{i=1}^{n} \left[u_i(t) \int_{G_e} \mu(x)\, \partial_x \psi_j(x)\, \partial_x \psi_i(x)\, \mathrm{d}x \right] = \int_G \psi_i\, f(x, t)\, \mathrm{d}x$$

for all basis functions ψ_j with $j = 1, ..., n$. This is the classical algebro-differential equation for **finite-element** type problems.

Matrix notation

This system of equations with the coefficients of the basis functions (meaning?!) as unknowns can be written in matrix notation

$$\mathbf{M} \cdot \ddot{\mathbf{u}}(t) \ + \mathbf{K} \cdot \mathbf{u}(t) = \mathbf{f}(t)$$

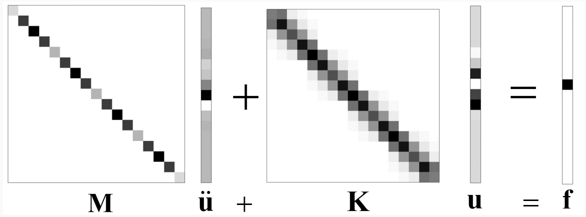
M Mass matrix

K Stiffness matrix

terminology originates from structural engineering problems

Matrix system graphically

The figure gives an actual graphical representation of the matrices for our 1D problem.



The unknown vector of coefficients **u** is found by a simple finite-difference procedure. The solution requires the inversion of mass matrix **M** which is trivial as it is diagonal. That is the key feature of the spectral element method.

Mass and stiffness matrix

Definition of the - at this point - global mass matrix

$$M_{ji} = \int_G \rho(x)\, \psi_j(x)\, \psi_i(x)\, \mathrm{d}x$$

and the stiffness matrix

$$K_{ji} = \int_G \mu(x)\, \partial_x \psi_j(x)\, \partial_x \psi_i(x)\, \mathrm{d}x$$

and the vector containing the volumetric forces $f(x,t)$

$$f_j(t) = \int_G \psi_i\, f(x,t)\, \mathrm{d}x \quad .$$

Mapping

A simple centred finite-difference approximation of the 2nd derivative and the following mapping

$$\begin{aligned} \mathbf{u}^{new} &\rightarrow \mathbf{u}(t + \mathrm{d}t) \\ \mathbf{u} &\rightarrow \mathbf{u}(t) \\ \mathbf{u}^{old} &\rightarrow \mathbf{u}(t - \mathrm{d}t) \end{aligned}$$

leads us to the solution for the coefficient vector $\mathbf{u}(t + \mathrm{d}t)$ for the next time step as already well known from the other solution schemes in previous schemes.

Solution scheme

$$\mathbf{u}^{new} = \mathrm{d}t^2 \left[\mathbf{M}^{-1} \left(\mathbf{f} - \mathbf{K} \;\; \mathbf{u} \right) \right] + 2\mathbf{u} - \mathbf{u}^{old}$$

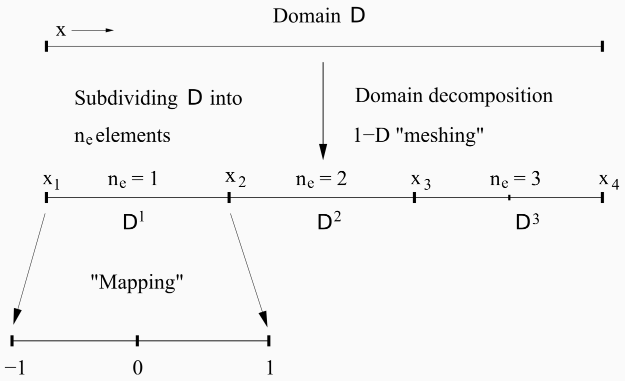
General solution scheme for finite-element method (wave propagation)

Independent of choice of basis functions

Mass matrix needs to be inverted!

To Do: good choice of basis function, integration scheme for calculation of M and K

Element level



In order to facilitate the calculation of the space-dependent integrals we transform each element onto the standard interval [-1,1], illustrated here for $n_e = 3$ elements. The elements share the boundary points.

System at element level

$$
\begin{aligned}
&\sum_{i=1}^{n}\left[\ddot{u}_i(t)\sum_{e=1}^{n_e}\int_{G_e}\rho(x)\psi_j(x)\psi_i(x)\mathrm{d}x\right] \\
&+\sum_{i=1}^{n}\left[u_i(t)\sum_{e=1}^{n_e}\int_{G_e}\mu(x)\partial_x\psi_j(x)\partial_x\psi_i(x)\mathrm{d}x\right] \\
&=\sum_{e=1}^{n_e}\int_{G_e}\psi_j(x)f(x,t)\mathrm{d}x \quad .
\end{aligned}
$$

Because of the sum over n_e there is a global dependence of the coefficients. How can we avoid this?

Local basis functions

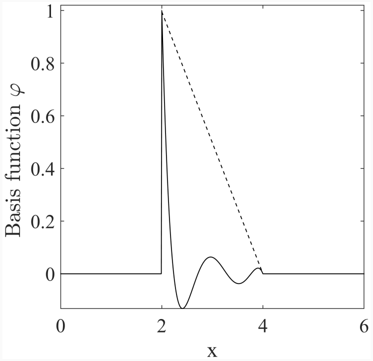


Illustration of local basis functions. By defining basis functions only inside elements the integrals can be evaluated in a local coordinate system. The graph assumes three elements of length two. A finite-element type linear basis function (dashed line) is shown along side a spectral-element type Lagrange polynomial basis function of degree N=5 (solid line).

u inside element

$$\overline{u}(x,t)|_{x\in G_e} = \sum_{i=1}^{n} u_i^e(t)\psi_i^e(x)$$

Now we can proceed with all calculations locally in G_e (inside one element)

Here is the difference to pseudospectral methods

Sum is now over all basis functions inside one element (n turns out to be the order of the polynomial scheme)

Local system of equations

Note that we still have **physical coordinates** here!

$$
\begin{aligned}
&\sum_{i=1}^{n} \ddot{u}_i^e(t) \int\limits_{G_e} \rho(x)\psi_j^e(x)\psi_i^e(x)\mathrm{d}x \\
+&\sum_{i=1}^{n} u_i^e(t) \int\limits_{G_e} \mu(x)\partial_x\psi_j^e(x)\partial_x\psi_i^e(x)\mathrm{d}x \\
=&\int\limits_{G_e} \psi_j^e(x)f(x,t)\mathrm{d}x \quad .
\end{aligned}
$$

Matrix notation for local system

$$\mathbf{M}^e \cdot \ddot{\mathbf{u}}^e(t) + \mathbf{K}^e \cdot \mathbf{u}^e(t) \;=\; \mathbf{f}^e(t), \qquad e \;=\; 1, \ldots, n_e$$

Here $\mathbf{u}^e$, $\mathbf{K}^e$, $\mathbf{M}^e$, and $\mathbf{f}^e$ are the coefficients of the unknown displacement inside the element, stiffness and mass matrices with information on the density and stiffnesses, and the forces, respectively.

Coordinate transformation

$$F_e : [-1,1] \to G_e, \qquad x = F_e(\xi),$$
$$\xi = \xi(x) = F_e^{-1}(\xi), \qquad e = 1,\dots,n_e$$

from our global system $x \in G$ to our local coordinates that we denote $\xi \in F_e$

$$x(\xi) = F_e(\xi) = \Delta e\frac{(\xi+1)}{2} + x_e$$

where n_e is the number of elements, and $\xi \in [-1,1]$. Thus the physical coordinate x can be related to the local coordinate ξ via

$$\xi(x) = F_e^{-1}(\xi) = \frac{2(x-x_e)}{\Delta e} - 1$$

Integration of arbitrary function

$$\int_{G_e} f(x)\mathrm{d}x = \int_{-1}^{1} f^e(\xi)\frac{\mathrm{d}x}{\mathrm{d}\xi}\mathrm{d}\xi$$

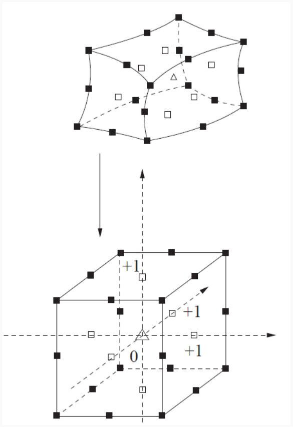
the integrand has to be multiplied by the Jabobian J before integration.

$$J = \frac{\mathrm{d}x}{\mathrm{d}\xi} = \frac{\Delta e}{2} \; .$$

we will also need

$$J^{-1} = \frac{\mathrm{d}\xi}{\mathrm{d}x} = \frac{2}{\Delta e} \, .$$

Skewed 3D elements



In 3D elements might be skewed and have curved boundaries. The calculation of the Jacobian is then carried out analytically by means of shape functions.

Assembly of system of equations (local coordinates)

$$
\begin{aligned}
&\sum_{i=1}^{n} \ddot{u}_i^e(t) \int\limits_{-1}^{1} \rho\,[x(\xi)]\,\psi_j^e\,[x(\xi)]\,\psi_i^e\,[x(\xi)]\,\frac{\mathrm{d}x}{\mathrm{d}\xi}\mathrm{d}\xi \\
+&\sum_{i=1}^{n} u_i^e(t) \int\limits_{-1}^{1} \mu\,[x(\xi)]\,\frac{\mathrm{d}\psi_j^e\,[x(\xi)]}{\mathrm{d}\xi}\,\frac{\mathrm{d}\psi_i^e\,[x(\xi)]}{\mathrm{d}\xi}\left(\frac{\mathrm{d}\xi}{\mathrm{d}x}\right)^2\frac{\mathrm{d}x}{\mathrm{d}\xi}\mathrm{d}\xi \\
=&\int\limits_{-1}^{1} \psi_j^e\,[x(\xi)]\,f\,[(x(\xi)),t]\,\frac{\mathrm{d}x}{\mathrm{d}\xi}\mathrm{d}\xi \quad .
\end{aligned}
$$

System of *n* equations for each index *j* corresponding to one particular basis function. We need to find basis functions that allow **efficient and accurate calculation of the required integrals**.

Lagrange polynomials

Remember we seek to approximate $u(x,t)$ by a sum over space-dependent basis functions ψ_i weighted by time-dependent coefficients $u_i(t)$.

$$u(x,t) \;\approx\; \overline{u}(x,t) \;=\; \sum_{i=1}^{n} u_i(t)\,\psi_i(x)$$

Our final choice: **Lagrange polynomials**:

$$\psi_i \;\to\; \ell_i^{(N)} \;:=\; \prod_{k=1,\;k\neq i}^{N+1} \frac{\xi-\xi_k}{\xi_i-\xi_k}, \qquad i = 1, 2, \ldots, N+1$$

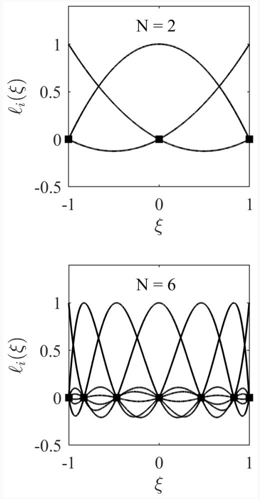
where x_i are fixed points in the interval $[-1,1]$.

Orthogonality of Lagrange polynomials

$$\ell_{\mathbf{i}}^{(\mathbf{N})}(\xi_{\mathbf{j}}) = \delta_{\mathbf{ij}}$$

They fulfill the condition that they *exactly* **interpolate the function at N+1 collocation points.** Compare with discrete Fourier series on regular grids or Chebyshev polynomials on appropriate grid points (-> pseudospectral method).

Lagrange polynomials graphically



Top: Family of $N + 1$ Lagrange polynomials for $N = 2$ defined in the interval $\xi \in [-1, 1]$. Note their maximum value in the whole interval does not exceed unity.

Bottom: Same for $N = 6$. The domain is divided into N intervals of uneven length. When using Lagrange polynomials for function interpolation the values are exactly recovered at the collocation points (squares).

Gauss-Lobatto-Legendre points

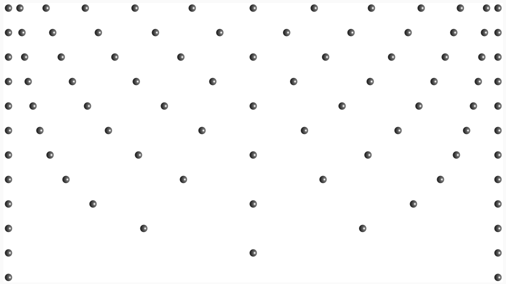


Illustration of the spatial distribution of **Gauss-Lobatto-Legendre points** in the interval [-1,1] from bottom to top for polynomial order 1 to 12. Note the increasing difference of largest to smallest interval between collocation points! Consequences?

Lagrange polynomials: some properties

Mathematically the collocation property is expressed as

$$\ell_i^{(N)}(\xi_i) = 1 \text{ and } \dot{\ell}_i^{(N)}(\xi_i) \ = \ 0$$

where the dot denotes a spatial derivative. The fact that

$$|\ell_i^{(N)}(\xi)| \leq 1, \qquad \xi \in [-1, 1]$$

minimizes the interpolation error in between the collocation points due to numerical inaccuracies.

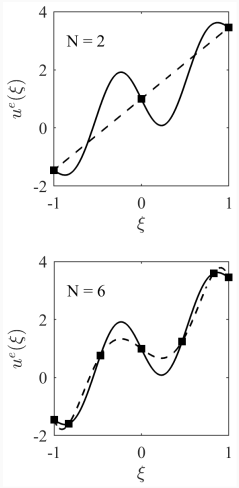
Function approximation

This is the final mathematical description of the unknown field $u(x, t)$ for the spectral-element method based on **Lagrange polynomials.**

$$u^e(\xi) \; = \; \sum_{i=1}^{N+1} u^e(\xi_i)\ell_i(\xi)$$

Other options at this point are the **Chebyhev polynomials.** They have equally good approximation properties (but ...)

Interpolation with Lagrange Polynomials



The function to be approximated is given by the solid lines. The approximation is given by the dashed line exactly interpolating the function at the GLL points (squares). **Top:** Order $N = 2$ with three grid points. **Bottom:** Order $N = 6$ with seven grid points.

Spectral-element system with basis functions (local coordinates)

Including the Legendre polynomials in our local (element-based) system leads to

$$\sum_{i=1}^{N+1} \ddot{u}_i^e(t) \int_{-1}^{1} \rho(\xi)\ell_j(\xi)\ell_i(\xi)\frac{\mathrm{d}x}{\mathrm{d}\xi}\mathrm{d}\xi$$

$$+\sum_{i,k=1}^{N+1} u_i^e(t) \int_{-1}^{1} \mu(\xi)\partial_\xi\ell_j(\xi)\partial_\xi\ell_i(\xi)\left(\frac{\mathrm{d}\xi}{\mathrm{d}x}\right)^2\frac{\mathrm{d}x}{\mathrm{d}\xi}\mathrm{d}\xi$$

$$=\int_{-1}^{1} \ell_j(\xi)f(\xi,t)\frac{\mathrm{d}x}{\mathrm{d}\xi}\mathrm{d}\xi$$

Because we want that ρ, μ, and f vary inside one element there is no way out carrying out the **integration numerically.**

Integration scheme for an arbitrary function $f(x)$

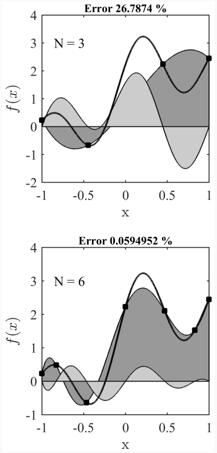
$$\int_{-1}^{1} f(x)\mathrm{d}x \approx \int_{-1}^{1} P_N(x)\mathrm{d}x = \sum_{i=1}^{N+1} w_i f(x_i)$$

defined in the interval $x \in [-1, 1]$ with

$$P_N(x) = \sum_{i=1}^{N+1} f(x_i)\ell_i^{(N)}(x)$$

$$w_i = \int_{-1}^{1} \ell_i^{(N)}(x)\mathrm{d}x \quad .$$

Numerical integration scheme



- Exact function (thick solid line)
- Approximation by Lagrange polynomials (thin solid line)
- Difference between true and approximate function (light gray)

Collocation points and integration weights

N	ξ_i	ω_i
2:	0	4/3
	± 1	1/3
3:	$\pm\sqrt{1/5}$	5/6
	± 1	1/6
4:	0	32/45
	$\pm\sqrt{3/7}$	49/90
	± 1	1/10

Collocation points and integration weights of the Gauss-Lobatto-Legendre quadrature for order $N = 2, \ldots, 4$.

After integration

With the numerical integration scheme we obtain

$$
\begin{aligned}
&\sum_{i,k=1}^{N+1} \ddot{u}_i^e(t) w_k \rho(\xi) \ell_j(\xi) \ell_i(\xi) \frac{\mathrm{d}x}{\mathrm{d}\xi}\bigg|_{\xi=\xi_k} \\
&+ \sum_{i,k=1}^{N+1} w_k u_i^e(t) \mu(\xi) \partial_\xi \ell_j(\xi) \partial_\xi \ell_i(\xi) \left(\frac{\mathrm{d}\xi}{\mathrm{d}x}\right)^2 \frac{\mathrm{d}x}{\mathrm{d}\xi}\bigg|_{\xi=\xi_k} \\
&\approx \sum_{i,k=1}^{N+1} w_k \ell_j(\xi) f(\xi, t) \frac{\mathrm{d}x}{\mathrm{d}\xi}\bigg|_{\xi=\xi_k}
\end{aligned}
$$

What is still missing is a formulation for the **derivative of the Lagrange polynomials** at the collocation points. But: Major progress! We have replaced the integrals by sums!

cont'd...

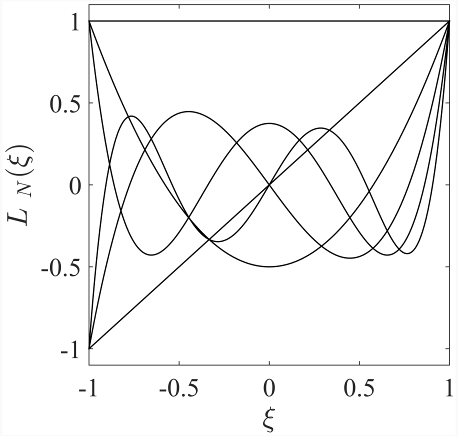
Solution equation for our spectral-element system at the element level

$$\sum_{i=1}^{N+1} M^e_{ji}\ddot{u}^e_i(t) + \sum_{i=1}^{N+1} K^e_{ji}u^e_i(t) \;=\; f^e_j(t), \qquad e \;=\; 1,\ldots,n_e$$

with

$$\begin{aligned}
M^e_{ji} &= w_j \rho(\xi)\frac{\mathrm{d}x}{\mathrm{d}\xi}\delta_{ij}\bigg|_{\xi=\xi_j} \\
K^e_{ji} &= \sum_{k=1}^{N+1} w_k \mu(\xi)\partial_\xi \ell_j(\xi)\partial_\xi \ell_i(\xi)\left(\frac{\mathrm{d}\xi}{\mathrm{d}x}\right)^2 \frac{\mathrm{d}x}{\mathrm{d}\xi}\bigg|_{\xi=\xi_k} \\
f^e_j &= w_j f(\xi,t)\frac{\mathrm{d}x}{\mathrm{d}\xi}\bigg|_{\xi=\xi_j}
\end{aligned}$$

Illustration of Legendre Polynomials



$$L_N(\xi) = \frac{1}{2^N N!}\frac{d^N}{d\xi^N}\left(\xi^2 - 1\right)^N$$

The Legendre polynomials are used to calculate the first derivatives of the Legendre polynomials. They can also be used to calculate the integration weights of the GLL quadrature.

Derivatives of Lagrange polynomials

Legendre polynomials can be calculated through a recursive formula:

$$\begin{aligned} L_0(\xi) &= 1 \\ L_1(\xi) &= \xi \\ L_{n\geq 2}(\xi) &= \frac{1}{n}[(2n-1)\,\xi\, L_{n-1}(\xi) - (n-1)L_{n-2}(\xi)] \end{aligned}$$

Derivatives of Lagrange polynomials

Then the derivatives

$$\partial_\xi \ell_k(\xi_i) = \sum_{j=0}^{N} d_{ij} \ell_k(\xi_j) \, , \quad k = 0, \ldots, N$$

Can be calculated with

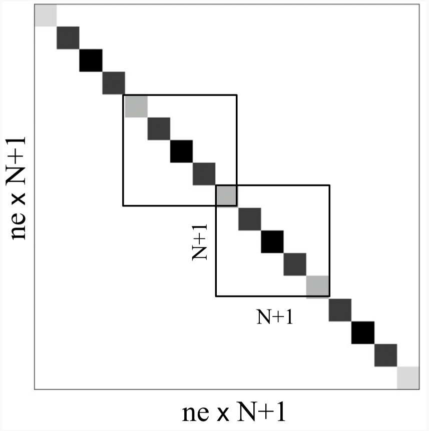
$$d_{ij} \quad = \quad \begin{cases} -\frac{1}{4} N(N+1) & \text{if } i = j = 0 \\ \\ \frac{L_N(\xi_i)}{L_N(\xi_j)} \frac{1}{\xi_i - \xi_j} & \text{if } 0 \leq i \leq N, \ 0 \leq j \leq N, \ i \neq j \\ \\ 0 & \text{if } 1 \leq i = j \leq N-1 \\ \\ \frac{1}{4} N(N+1) & \text{if } i = j = N \end{cases}$$

Global Assembly and Solution

Global Assembly for the diagonal of the mass matrix

$$\mathbf{M}_g = \begin{pmatrix} M_{1,1}^{(1)} \\ M_{2,2}^{(1)} \\ M_{3,3}^{(1)} \\ \\ \\ \\ \\ \end{pmatrix} + \begin{pmatrix} \\ \\ M_{1,1}^{(2)} \\ M_{2,2}^{(2)} \\ M_{3,3}^{(2)} \\ \\ \\ \end{pmatrix} + \begin{pmatrix} \\ \\ \\ \\ M_{1,1}^{(3)} \\ M_{2,2}^{(3)} \\ M_{3,3}^{(3)} \end{pmatrix} = \begin{pmatrix} M_{1,1}^{(1)} \\ M_{2,2}^{(1)} \\ M_{3,3}^{(1)} + M_{1,1}^{(2)} \\ M_{2,2}^{(2)} \\ M_{3,3}^{(2)} + M_{1,1}^{(3)} \\ M_{2,2}^{(3)} \\ M_{3,3}^{(3)} \end{pmatrix}$$

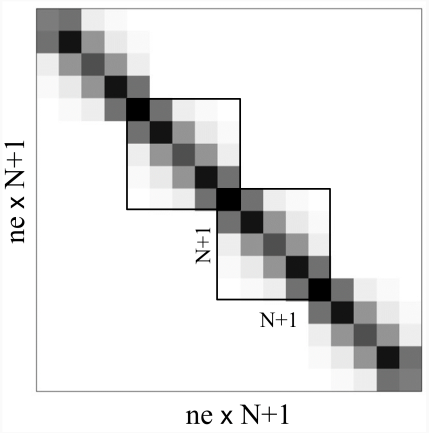
Mass Matrix



Global Assembly for the diagonal of the stiffness matrix

$$\mathbf{K}_g = \begin{pmatrix} K_{1,1}^{(1)} & K_{1,2}^{(1)} & K_{1,3}^{(1)} & & & & \\ K_{2,1}^{(1)} & K_{2,2}^{(1)} & K_{2,3}^{(1)} & & & \mathbf{0} & \\ K_{3,1}^{(1)} & K_{3,2}^{(1)} & K_{3,3}^{(1)} + K_{1,1}^{(2)} & K_{1,2}^{(2)} & K_{1,3}^{(2)} & & \\ & & K_{2,1}^{(2)} & K_{2,2}^{(2)} & K_{2,3}^{(2)} & & \\ & & K_{3,1}^{(2)} & K_{3,2}^{(2)} & K_{3,3}^{(2)} + K_{1,1}^{(3)} & K_{1,2}^{(3)} & K_{1,3}^{(3)} \\ & \mathbf{0} & & & K_{2,1}^{(3)} & K_{2,2}^{(3)} & K_{2,3}^{(3)} \\ & & & & K_{3,1}^{(2)} & K_{3,2}^{(2)} & K_{3,3}^{(2)} \end{pmatrix}$$

Stiffness Matrix

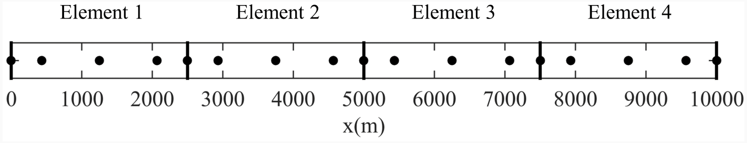


Vector with information on the source

$$\mathbf{f}_g = \begin{pmatrix} f_1^{(1)} \\ f_2^{(1)} \\ f_3^{(1)} + f_1^{(2)} \\ f_2^{(2)} \\ f_3^{(2)} + f_1^{(3)} \\ f_2^{(3)} \\ f_3^{(3)} \end{pmatrix}$$

Exercise:

The figure shows elements and collocations points for a 1D spectral element problem (4 elements, $N = 4$) . Develop equation(s) for the 1D (2D, 3D) number of degrees of freedom n_g as a function of number of elements n_e and order N of the Lagrange polynomials. Give examples with realistic numbers.



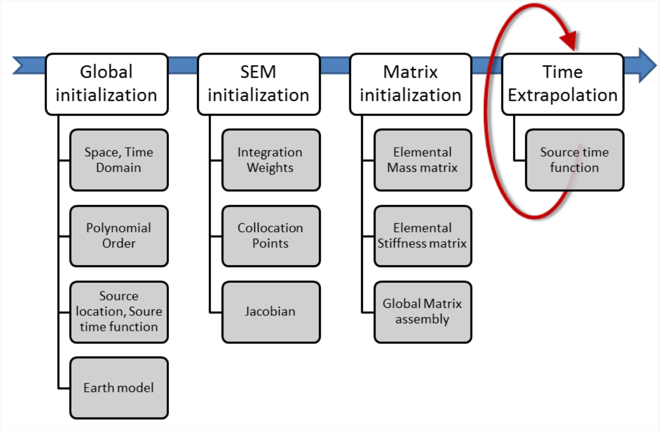
Extrapolation for time-dependent coefficients $\mathbf{u}_g$

This is our final algorithm as it is implemented using Matlab or Python

$$\mathbf{u}_g(t+\mathrm{d}t) \; = \mathrm{d}t^2 \left[\mathbf{M}_g{}^{-1} \left(\mathbf{f}_g(t) - \mathbf{K}_g \; \mathbf{u}_g(t)\right) \right] \\ + 2\mathbf{u}_g\mathbf{(t)} \; - \mathbf{u}_g(t-\mathrm{d}t)$$

Looks fairly simple, no?

Spectral elements: work flow



A substantial part consists of preparing the interpolation and integration procedures required to initialize the global mass- and stiffness matrices. The final time-extrapolation is extremely compact and does not require the inversion of a global matrix as is the case in classic finite-element methods.

Python Code: Mass Matrix

```python
# Global Mass matrix
# ----------------------------------------------------------
k = -1
m = -1
ng = (ne-1)*N + N + 1
M = np.zeros(2*ng)
for i in range(1, ne+1):
    # ------------------------------------
    # Elemental Mass matrix
    # ------------------------------------
    for l in range(0, N+1):
        m += 1
        Me[l] = rho[m] * w[l] * J     #stored as a vector sir
    m -= 1
    # ------------------------------------
    for j in range(0, N+1):
        k = k + 1
        if i>1:
            if j==0:
                k = k - 1
        M[k] = M[k] + Me[j]

# Inverse matrix of M
# ----------------------------------------------------------
Minv = np.identity(ng)
for i in range(0,ng):
    Minv[i,i] = 1./M[i]
```

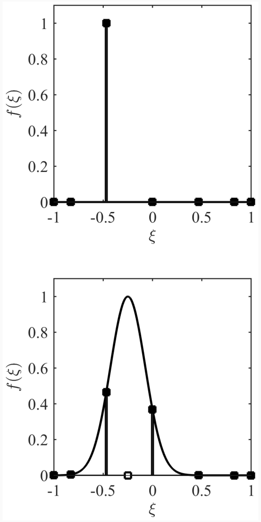
Pyton Code: Stiffness Matrix

```python
# Global Stiffness Matrix
# ----------------------------------------------------------------
K = np.zeros([ng, ng])
xe = 0

for e in range(1, ne + 1):
    i0 = (e - 1)*N + 1
    j0 = i0
    # ------------------------------------
    # Elemental Stiffness Matrix
    # ------------------------------------
    for i in range(-1, N):
        for j in range(-1, N):
            sum = 0
            for k in range(-1, N):
                sum = sum + mu[k+1+xe] * w[k+1] * Ji**2 * J * l1d[i+1,k+1] * l1d[j+1,k+1]
            Ke[i+1, j+1] = sum
    xe += N

    for i in range(-1,N):
        for j in range(-1, N):
            K[i0+i, j0+j] += Ke[i+1, j+1]
```

Pyton Code:Time Extrapolation

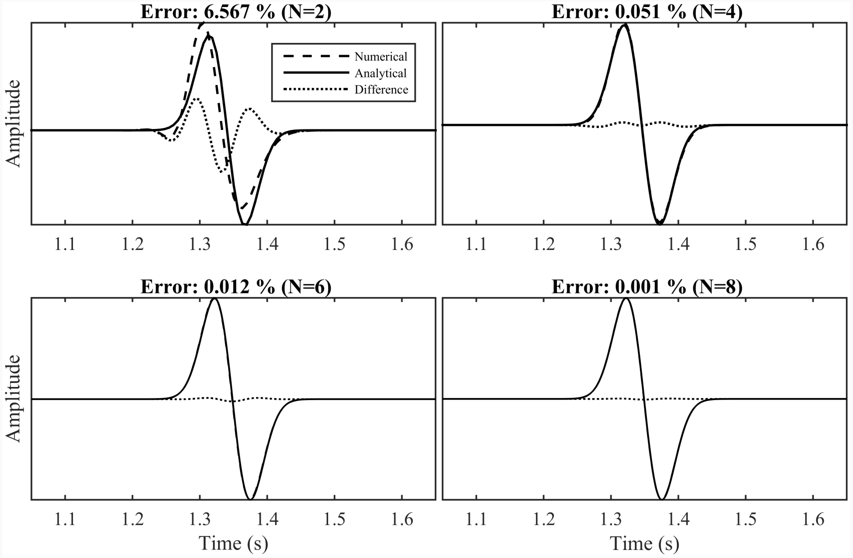
```python
# ---------------------------------------------------------
# Time extrapolation
# ---------------------------------------------------------
x_t = []
for it in range(nt):
    # Source initialization
    f = np.zeros(ng)
    if it < len(src):
        f[isrc-1] = src[it-1]

    # Time extrapolation
    unew = dt**2 * Minv @ (f - K @ u) + 2 * u - uold
    uold, u = u, unew

    # Solution in space-time
    x_t.append(u)
```

Point source initialization

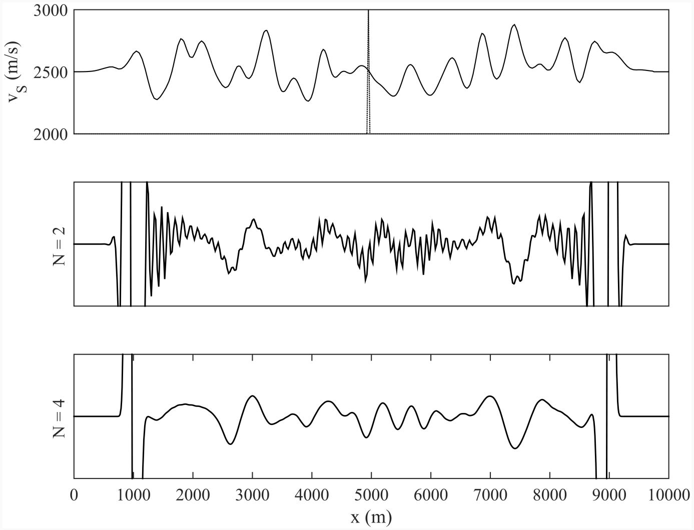


Top: A point-source polynomial representation (solid line) of a δ-function (red bar).
Bottom: A finite-source polynomial representation (solid line) as a superposition of point sources injected at some collocation points. For comparison with analytical solutions it is important to note that the spatial source function actually simulated is the polynomial representation of the (sum over) δ-function(s).

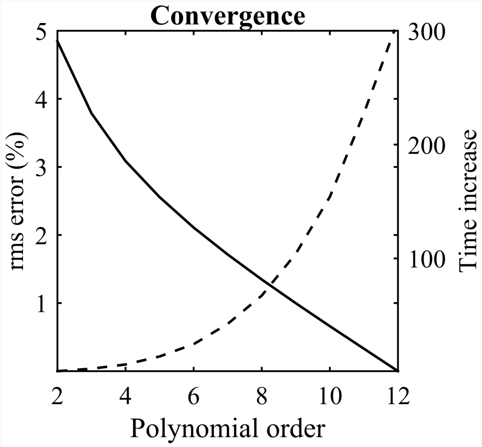
sem1d: simulation examples



sem1d: Convergence test



sem1d: Convergence test



Summary

- The spectral element method combines the **flexibility of finite-element methods** concerning computational meshes with the **spectral convergence** of **Lagrange basis functions** used inside the elements.
- The enormous success of the spectral element method is based upon the **diagonal structure of the mass matrix** that needs to be inverted to extrapolate the system in time. Due to this diagonality no matrix inversion techniques need to be employed allowing **straight forward parallelisation** of the algorithm. The diagonal mass matrix is possible through the coincidence of the collocation points of both interpolation and integration schemes (Gauss-Lobatto-Legendre).

Summary

- The **errors** of the spectral-element scheme accumulate from the (usually low-order finite-difference) time extrapolation scheme and the **numerical integration** using Gauss-Lobatto-Legendre quadrature.
- The spectral-element method is particularly useful for simulation problems where the **free-surface** plays an important role, and/or in which surface waves need to be accurately modelled. Technically the reason is that the **free-surface boundary is implicitly solved**.
- A well engineered community code (**SPECFEM3D**, www.geodynamics.org) is available for Cartesian and spherical geometries including global Earth (or planetary scale) calculations.
- Recently a commercial spectral-element code has been developed (**salvus**, mondaic.com) by the ETH group (Andreas Fichtner)