The Spectral Element Method in 1D

Introduction of the Basic Concepts and Comparison to Optimal FD Operators

Seismology and Seismics Seminar 17.06.2003
Ludwigs-Maximilians-Universität München
by
Bernhard Schuberth
Outline

- Motivation
- Mathematical Concepts and Implementation
  - Weak Formulation
  - Mapping Function → irregular grids
  - Interpolation and integration
  - Diagonal mass matrix???
  - What about the stiffness matrix?
  - Assembly of the global linear system
  - Boundary conditions – examples
- Summary of SEM
Comparison to Optimal FD Operators
  - Setup
  - Seismograms
  - How to compare the performance?
  - Results of for a homogeneous and a two layered model

Conclusions

Future Work
Motivation – why SEM?

- High accuracy
- Parallel implementation is fairly easy ↔ diagonal mass matrix
- Advantages of meshing like in FEM
  - Better representation of topography and interfaces
  - Possibility of deformed elements

Pictures taken from Komatitsch and Vilotte (1998) and Komatitsch and Tromp (2002a-b)
Motivation – Why still looking at 1D?

- Simple analytical solution
  - Quantitative comparisons

- Educational aspect
  - all concepts can be explained considering a 1D case
  - Extensions to higher dimension are then rather straightforward
  - formulas are looking simpler 😊
From the „Weak Formulation“ to a Global Linear System

Aim of SEM (FEM) formulation: invertible linear system of equations

Starting with 1D wave equation:
\[ \rho \frac{\partial^2 u}{\partial t^2} - \frac{\partial}{\partial x} \left( \mu \frac{\partial u}{\partial x} \right) = f(x) \]

\[ \int_{\Omega} \rho \ v \ \ddot{u} \ dx \ - \int_{\Gamma} v \ \mu \ \nabla u \ dx + \int_{\Omega} \nabla v \ \mu \ \nabla u \ dx = \int_{\Omega} v \ f \ dx \]

Free surface boundary conditions:
\[ \mu \ \frac{\partial u}{\partial x} = \sigma = 0 \]

Weak Formulation:
\[ \int_{\Omega} \rho \ v \ \ddot{u} \ dx + \int_{\Omega} \nabla v \ \mu \ \nabla u \ dx = \int_{\Omega} v \ f \ dx \]

Linear System of equations – matrix formulation
\[ \mathbf{M} \ddot{\mathbf{U}} + \mathbf{K} \mathbf{U} = \mathbf{F} \]
5 Steps to get the Global Matrix Equation

1. Domain decomposition → Mesh of elements
   → Transformation between physical and local element coordinates
   = Mapping

2. Interpolation of functions on the elements → Lagrange polynomials
   → Gauss-Lobatto-Legendre (GLL) points

3. Integration over the element → GLL integration quadrature
   GLL points and weights

4. The elemental matrices:
   - mass matrix → diagonal using Lagrange polynomials and GLL quadrature
can be used as in FEM but it is easier to calculate forces (see later)
   - stiffness matrix

5. Assembly → Connectivity Matrix
   → Global linear system
1. Domain Decomposition – Mapping Function

Domain $\Omega$

Subdividing $\Omega$ into elements → 1D "meshing"

$\Omega_1$ → $\Omega_2$ → $\Omega_3$

Coordinate transformation

Mapping Function

$$x(\xi) = \sum_{a=1}^{n_a} N_a(\xi) x_a$$
Mapping Function – Coordinate Transformation

! Now 2D! \[ x(\xi, \eta) = \sum_{a=1}^{n_a} N_a(\xi, \eta) \ x_a \]

product of degree 1 Lagrange polynomials

\[ N_1(\xi, \eta) = \ell_0^1(\xi) \ \ell_0^1(\eta), \quad \ell_0^1(\xi) = \frac{1 - \xi}{2}, \]

\[ N_2(\xi, \eta) = \ell_1^1(\xi) \ \ell_0^1(\eta) \quad \ell_1^1(\xi) = \frac{1 + \xi}{2} \]

Examples

product of degree 2 Lagrange polynomials

\[ N_1(\xi, \eta) = \ell_0^2(\xi) \ \ell_0^2(\eta), \quad \ell_0^2(\xi) = \frac{\xi(\xi - 1)}{2}, \]

\[ N_2(\xi, \eta) = \ell_1^2(\xi) \ \ell_0^2(\eta) \quad \ell_1^2(\xi) = 1 - \xi^2 \]

Notice! In 3D it is a triple product!
Shape Function – 2D Examples
Later, when calculating derivatives and integrals, we will have to correct for the coordinate transformation. **HOW is it done?**

### 1D

Jacobi Matrix and its determinant called Jacobian

\[
J = \frac{\partial x}{\partial \xi}, \quad \mathcal{J} = \left| \frac{\partial x}{\partial \xi} \right|
\]

\[
\frac{\partial x(\xi)}{\partial \xi} = \sum_{a=1}^{n_a} \frac{\partial N_a(\xi)}{\partial \xi} x_a
\]

The Jacobian describes the volume change of the element

### 3D

\[
J = \begin{pmatrix}
\frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\
\frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\
\frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta}
\end{pmatrix}
\]

\[
\mathcal{J} = \left| \|J\| \right| = \left| \frac{\partial (x, y, z)}{\partial (\xi, \eta, \zeta)} \right|
\]

\[
dx \ dy \ dz = \mathcal{J} \ d\xi \ d\eta \ d\zeta
\]
2. Interpolation on the Elements

Interpolation is done using Lagrange polynomials defined on the Gauss-Lobatto-Legendre points.

\[ u_e(\xi) \approx \sum_{i=0}^{N} u_e(\xi_i) \ell_i(\xi) \]

\[ \nabla u_e(\xi) \approx \sum_{i=0}^{N} u_e(\xi_i)\ell'_i(\xi) \]

Interpolating functions:

\[ \ell_i = \prod_{\substack{j=0 \atop j \neq i}}^{N} \frac{\xi - \xi_j}{\xi_i - \xi_j} \]

\[ \ell_i(\xi_j) = \delta_{ij} \]

Polynomial degree N for interpolation is usually higher than that for the mapping.

GLL points: The N+1 roots of the Legendre polynomial \( P_N \) of degree N.
Lagrange Polynomials - Examples

All 6 Lagrange polynomials of degree 5

Lagrange polynomial of degree 8
3. Integration Over the Element

Gauss-Lobatto-Legendre quadrature for spatial integration

!BIG advantage!
(compared to quadratures using Chebychev polynomials)

same collocation points for interpolation and integration → diagonal mass matrix
(this we will see on the next slides – remember \( \ell_i(\xi_j) = \delta_{ij} \))

\[
\int_{-1}^{1} f(\xi) \, d\xi = \sum_{i=0}^{N} \omega_i \, f(\xi_i)
\]

GLL weights of integration

\[
\omega_i = \frac{2}{N(N+1)[P_N(\xi_i)]^2} \quad (\xi_i \neq \pm 1)
\]

\[
\omega_i = \frac{2}{N(N+1)} \quad (\xi_i = \pm 1)
\]
4. The Elemental Matrices – Mass Matrix

Now - the most important issue of SEM – How does it become diagonal? (Sorry, nasty formulas inevitable ;-) 

Starting with 1. term of the weak formulation:

\[ \int_{\Omega_e} \rho(x) \ v(x) \ \ddot{u}(x) \ dx = \int_{\Lambda} \rho(\xi) \ v(\xi) \ \ddot{u}(\xi) \ \mathcal{J} \ d\xi \]

coordinate transformation

\[ = \int_{-1}^{1} \rho(\xi) \left[ \sum_{i=0}^{N} v_i \ \ell_i(\xi) \right] \left[ \sum_{j=0}^{N} \ddot{u}_j \ \ell_j(\xi) \right] \ J \ d\xi \]

interpolating v and u

integration quadrature

\[ = \sum_{k=0}^{N} \left\{ \rho(\xi_k) \ \omega_k \left[ \sum_{i=0}^{N} v_i \ \ell_i(\xi_k) \right] \left[ \sum_{j=0}^{N} \ddot{u}_j \ \ell_j(\xi_k) \right] \ J(\xi_k) \right\} \]
Mass Marix cont´d

\[ \sum_{k=0}^{N} \left\{ \rho_k \omega_k \left[ \sum_{i=0}^{N} v_i \ell_i(\xi_k) \right] \left[ \sum_{j=0}^{N} \ddot{u}_j \ell_j(\xi_k) \right] \mathcal{J}_k \right\} = \]

Thanks to Kronecker delta the formula is getting simpler!

\[ = \sum_{k=0}^{N} \left\{ \rho_k \omega_k \left[ \sum_{i=0}^{N} \delta_{ik} \right] \left[ \sum_{j=0}^{N} \ddot{u}_j \delta_{jk} \right] \mathcal{J}_k \right\} \]

Rearranging we get which can be expressed as

\[ = \sum_{j=0}^{N} \left\{ \ddot{u}_j \left[ \sum_{i=0}^{N} \sum_{k=0}^{N} \rho_k \omega_k \delta_{ik} \delta_{jk} \mathcal{J}_k \right] \right\} \quad \triangleq \quad \ddot{u}_j m e_{ij} \]

Finally the world is simple again!

\[ m e_{ij} = \rho_i \omega_i \mathcal{J}_i \delta_{ij} \]
Having to factorize an even more complicated equation we obtain the stiffness matrix

Note! The „Kronecker delta relation“ does not hold for the derivatives of the Lagrange polynomials → the stiffness matrix is not diagonal!

\[ k_{e_{ij}} = \sum_{k=0}^{N} \mu_k \omega_k \ell'_i(\xi_k) \ell'_j(\xi_k) J_k \]

All elements of the elemental stiffness matrix are therefore nonzero
5. The Assembly Process – Connectivity Matrix

How do the elemental matrices contribute to the global system?

Important information we need:
- How are the elements connected?
- Which elements share nodes
- To which elements contributes a certain node?

→ “Connectivity Matrix”

\[
C_{ij} = \begin{pmatrix}
1 & 5 & [9 = (3 - 1)N + 1] & \ldots & [ne - 1)N + 1] \\
2 & 6 & = (j - 1)N + 1] & \ldots & [ne - 1)N + 2] \\
3 & 7 & \ldots & \ldots & [ne - 1)N + N + 1] \\
4 & 8 & \ldots & \ldots & \ldots \\
5 & [9 = 2N + 1] & \ldots & \ldots & \ldots \\
\end{pmatrix}
\]
Assembling the Global Matrices

How do we use the information contained in the Connectivity Matrix?

\[ M(C_{j,i}) = M(C_{j,i}) + me_{j}^{(i)} \]

\[ K(C_{k,i}, C_{j,i}) = K(C_{k,i}, C_{j,i}) + ke_{k,j}^{(i)} \]

\[ M = \begin{pmatrix} * & * & \circ \\ * & * & \circ \\ \circ & \circ & \circ \end{pmatrix} = \begin{pmatrix} * + \circ \\ \circ + \circ \\ \circ + \circ \end{pmatrix} \]
Two Ways to get the Global Matrix Equation

1. Explicitly calculating the global stiffness matrix once for the whole simulation

2. Calculating the forces at all nodes for every timestep and then summing the forces at each node (= assembling the global force vector $\mathbf{F}$)
   Advantage: much easier to implement in 2- and 3D
   Drawback: CPU time increases

Calculation of Forces

strain at node $i$:

$$\frac{\partial u_i}{\partial x} = \frac{\partial u_i}{\partial \xi} \frac{\partial \xi}{\partial x}$$

$$= \sum_j u_j \ell_j'(\xi_i) \cdot J_i^{-1}$$

Hooke’s Law $\rightarrow$ stress:

$$\sigma_i = \mu_i \frac{\partial u_i}{\partial x}$$

$$f_{int}^{(e)} = \int_{\Omega_e} \nabla v \sigma$$

$$= \int_{-1}^{1} \nabla v \sigma J \frac{\partial \xi}{\partial x}$$

$$= \sum_k [\sum_j \ell_j'(\xi_k)] J_k^{-1} \sigma_k J_k$$

Note: Here we need to correct for the coordinate transformation (same for stiffness matrix)

$\rightarrow$ 2 x Jacobi Matrix of inverse transformation $\rightarrow$ 1 x Jacobian Determinant
Implementation of different boundary conditions in the SEM formulation is very easy

1. **Free surface** boundary conditions are implicitly included in the weak formulation → nothing to be done → time integration (explicit Newmark scheme)

\[ U_{t_{i+1}} = \Delta t^2 \cdot M^{-1} F + 2U_{t_i} - U_{t_{i-1}} \]

2. **Rigid** boundaries are easily applied by not inverting the linear system for boundary nodes

\[ U_{t_{i+1}}(2 : ng - 1) = \Delta t^2 \cdot M^{-1}(2 : ng - 1) F(2 : ng - 1) + 2U_{t_i}(2 : ng - 1) - U_{t_{i-1}}(2 : ng - 1) \]

This corresponds to setting \( U(1) \) and \( U(ng) \) equal to zero for all times.
Boundary Conditions cont’d

3. Periodic boundary conditions: sum forces and masses at the edges

\[
\rightarrow F_{\text{periodic}}^{(1)} = F_{\text{fs}}^{(1)} + F_{\text{fs}}^{(ng)} \\
\rightarrow M_{\text{periodic}}^{(1)} = M_{\text{fs}}^{(1)} + M_{\text{fs}}^{(ng)}
\]

same for \( F_{\text{periodic}}^{(ng)} \) and \( M_{\text{periodic}}^{(ng)} \)

4. Absorbing boundary conditions:

stress conditions at the edges:
\[
\sigma = \rho \, v_S \, \dot{u}
\]

\[
\rightarrow F_{\text{absorbing}}^{(1)} = F_{\text{fs}}^{(1)} + \rho(1) \, v_S(1) \, \dot{u}(1)
\]

\[
\rightarrow F_{\text{absorbing}}^{(1)} = F_{\text{fs}}^{(ng)} + \rho(ng) \, v_S(ng) \, \dot{u}(ng)
\]
Summary of SEM Concepts

- Weak formulation
- Therefore few problems with boundary conditions
- Use forces instead of the stiffness matrix
- Additional information is needed – Connectivity
- Assembly is time consuming
- Lagrange polynomials in connection with Gauss-Lobatto-Legendre quadrature
- Most important: the diagonal mass matrix
Comparison of SEM and Optimal FD Operators

Why comparing to Optimal Operators (Opt. Op.)?

- Both methods are said to be more accurate than conventional FE and FD methods
- Both were developed in the last 10 years and are still not commonly used (especially Opt. Op. are not fully established in computational Seismology)
### Comparision

#### The Setup

<table>
<thead>
<tr>
<th>Item</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model size</strong></td>
<td>1200 grid points</td>
</tr>
<tr>
<td><strong>Source</strong></td>
<td>Delta peak in space and time</td>
</tr>
<tr>
<td><strong>Receiver Array</strong></td>
<td>every 5\textsuperscript{th} gridpoint $\rightarrow$ 240 receivers</td>
</tr>
<tr>
<td><strong>Effective Courant Number</strong></td>
<td>0.5 (grid spacing varies in SEM)</td>
</tr>
<tr>
<td><strong>Filtering</strong></td>
<td>5 to 40 points per wavelength (ppw)</td>
</tr>
<tr>
<td></td>
<td>frequencies chosen correspondingly</td>
</tr>
<tr>
<td><strong>Propagation length</strong></td>
<td>1 to 40 propagated wavelengths (npw)</td>
</tr>
<tr>
<td><strong>Analytical solution</strong></td>
<td>Heaviside shaped</td>
</tr>
<tr>
<td>(displacement)</td>
<td>first arrival in seismogram at time $t = \frac{x}{v}$</td>
</tr>
</tbody>
</table>
Seismograms
Homogeneous and Two Layered Medium
How to check for performance?

synthetic seismograms + analytical solution → relative solution error (rse)

The used setup allows for a huge database of rse covering a wide range of ppw and npw

\[ \delta E = \frac{\int (u_{\text{analyt}} - u_{\text{sim}})^2 dt}{\int u_{\text{analyt}}^2 dt} \]

What determines the performance of a numerical method?

The effort (or CPU time) it takes to achieve a certain accuracy for a given problem – i.e. the points per wavelength one has to use to reach an acceptable error after propagating the signal a wanted number of wavelengths

\[ \text{Cost} = \text{ppw (rse) \cdot npw \cdot \frac{cpt}{nx}} \]

and the memory usage.
CPU Cost

Benchmarking the CPU time

Several runs with different model sizes (1000 – 20000 nodes) → CPU time per grid node

![Graph showing CPU time per model size for different methods: SEM 12, SEM 8, SEM 5, Opt. Op. 5pt, Taylor 5pt. The x-axis represents model size, and the y-axis represents CPU time per time step.](image)
Results – Homogeneous Case
Homogeneous Model– RSE Difference and Relative CPU time

Difference $\Delta \text{rse} \%$:

$\text{RSE(SEM)} - \text{RSE(Opt. Op.)}$

Relative CPU time:

$rse \%$ vs $\text{npw}$ and $\text{ppw}$
Two Layered Model– RSE Difference and Relative CPU time

difference RSE(SEM) – RSE(Opt. Op.)
Conclusions

- Memory → no big role in 1D, but may be in 3D
- Similar behaviour of errors
  - SEM slightly better for less than 20 npw and for 7 ppw and less
- CPU cost is much higher for SEM
  - Opt. Op. win for the used setup (without boundaries) in 1D
- Simulations including surface waves in 3D models may be better with SEM
- Perhaps better approximation of interfaces
Future Work

1D
- Comparison of several subroutines for the calculation of the GLL points and weights
- Comparison of *stiffness matrix* and *calculation of forces* implementations

3D
- Installation of a 3D code written by Komatitsch and Tromp (2003) on the Hitachi
- Comparison of 3D SEM-simulations in the Cologne Basin model with the results of the FD simulations performed by Michael Ewald
Thank You For Your Attention!

Bernhard Schuberth
Boundary Conditions – Examples
Free Surface
Boundary Conditions – Examples
Rigid Boundaries
Boundary Conditions – Examples
Periodic Boundaries
Boundary Conditions – Examples
Absorbing Boundaries
Results – Two Layered Medium

Spectral Element Method of Order 8

Optimal Modified 3-point FD Operator