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

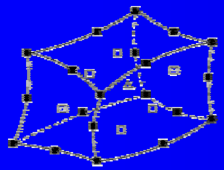
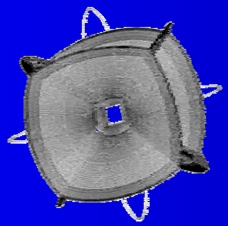
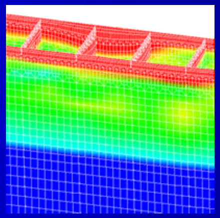
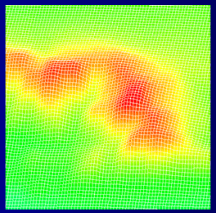
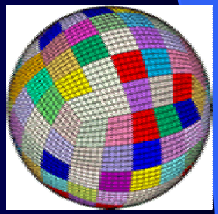
- Motivation
- Mathematical Concepts and Implementation
  - Weak Formulation
  - Mapping Function $\rightarrow$ irregular grids
  - Interpolation and integration
  - Diagonal mass matrix???
  - What about the stiffness matrix?
  - Assembly of the global linear system
  - Boundary conditions – examples
- Summary of SEM

Outline cont´d

- 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 $\leftrightarrow$ 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: $$\rho \frac{\partial^2 u}{\partial t^2} - \frac{\partial}{\partial x} (\mu \frac{\partial u}{\partial x}) = f(x)$$

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

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

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

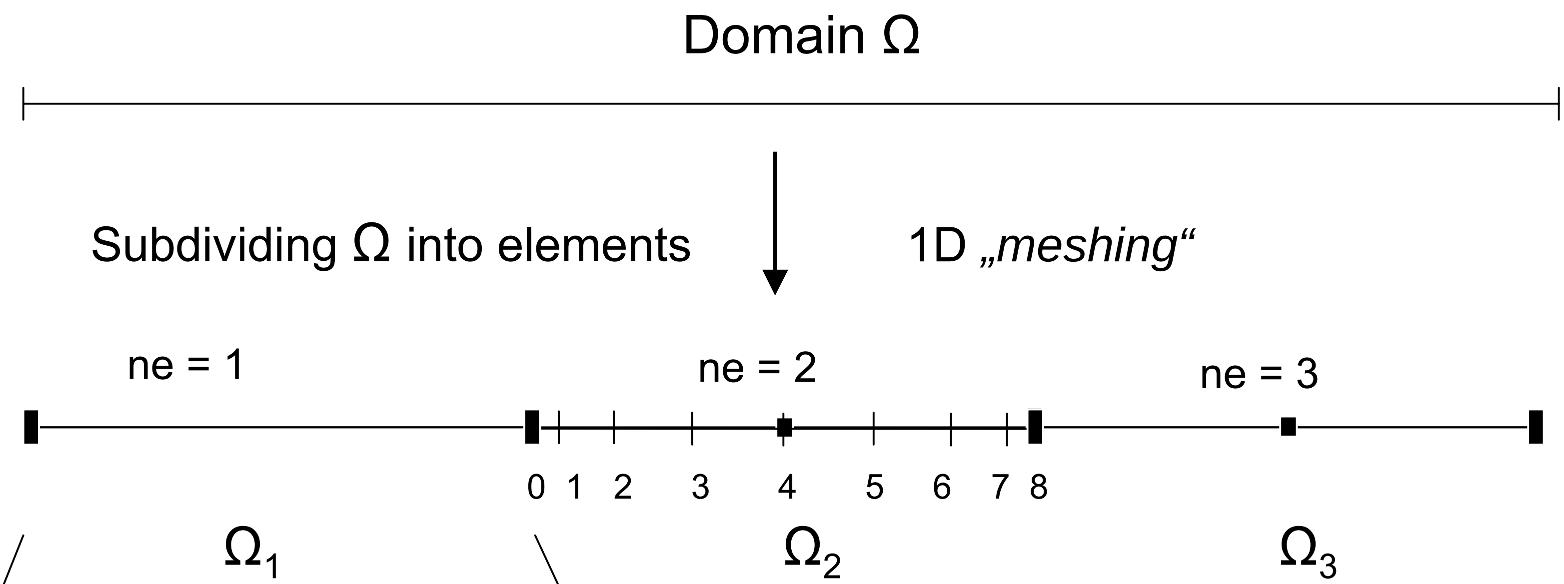
Linear System of equations – matrix formulation

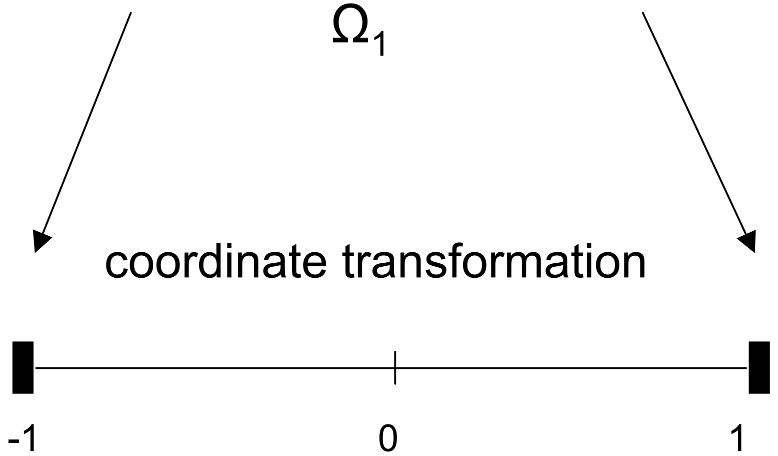
$$\boxed{\mathbf{M\ddot{U}} \; + \; \mathbf{KU} \; = \; \mathbf{F}}$$

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
	- **stiffness matrix**	→	can be used as in FEM but it is easier to calculate forces (see later)
5.	**Assembly**	→	Connectivity Matrix
		→	**Global linear system**

1. Domain Decomposition – Mapping Function



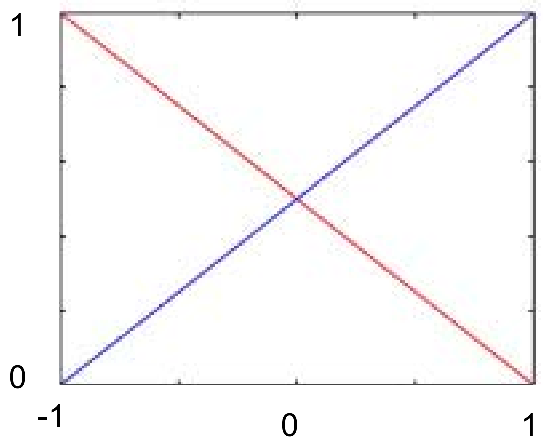


Mapping Function

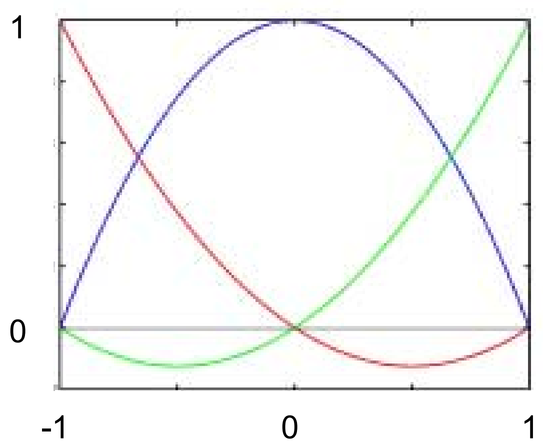
$$x(\xi) = \sum_{a=1}^{n_a} N_a(\xi)\, x_a$$

Mapping Function – Coordinate Transformation

non deformed elements



deformed elements



! Now 2D!

$$x(\xi,\eta) \;=\; \sum_{a=1}^{n_a} N_a(\xi,\eta)\; x_a$$

product of degree **1** Lagrange polynomials

$$N_1(\xi,\eta) \;=\; \ell_0^1(\xi)\;\ell_0^1(\eta), \qquad \ell_0^1(\xi) \;=\; \frac{1-\xi}{2},$$

$$N_2(\xi,\eta) \;=\; \ell_1^1(\xi)\;\ell_0^1(\eta) \qquad \ell_1^1(\xi) \;=\; \frac{1+\xi}{2}$$

Examples

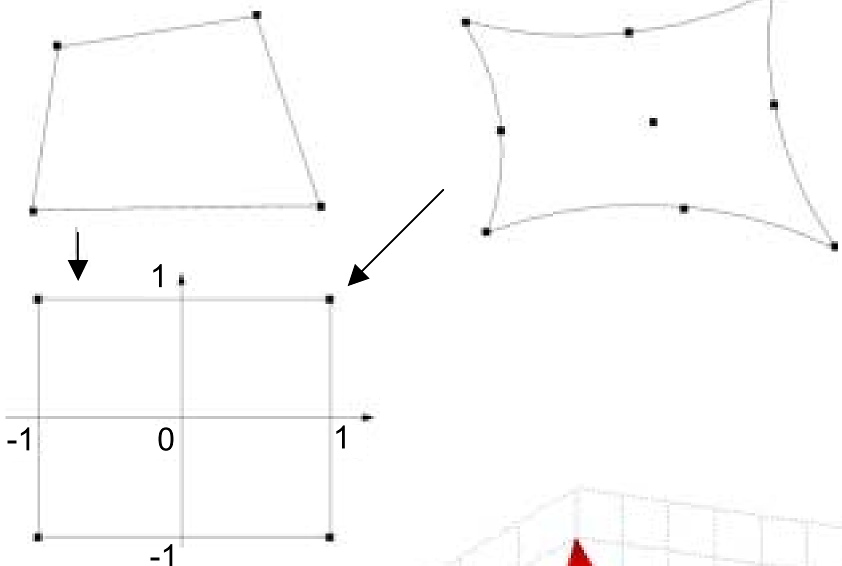
product of degree **2** Lagrange polynomials

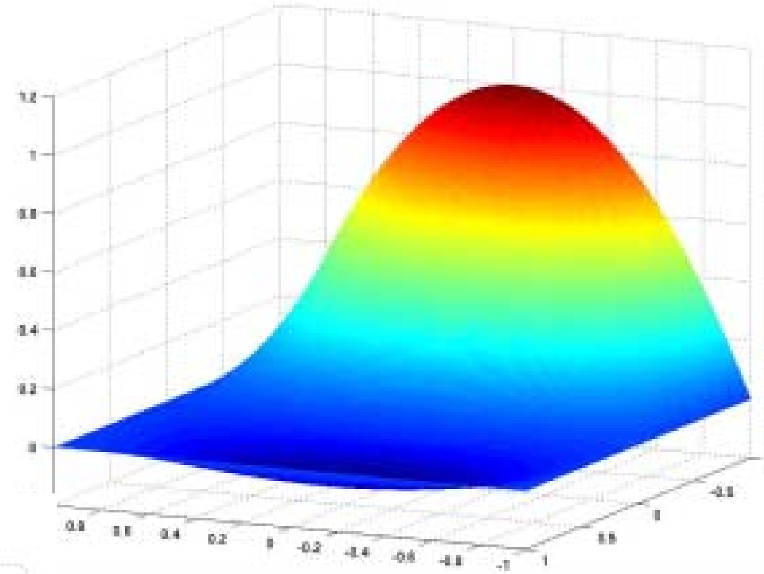
$$N_1(\xi,\eta) \;=\; \ell_0^2(\xi)\;\ell_0^2(\eta), \qquad \ell_0^2(\xi) \;=\; \frac{\xi(\xi-1)}{2},$$

$$N_2(\xi,\eta) \;=\; \ell_1^2(\xi)\;\ell_0^2(\eta) \qquad \ell_1^2(\xi) \;=\; 1-\xi^2$$

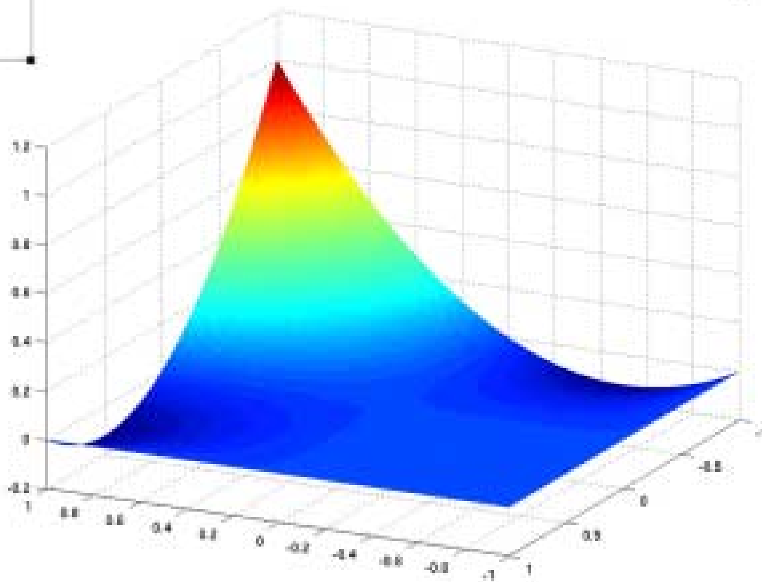
Notice! In 3D it is a triple product!

Shape Function – 2D Examples





$\bullet_0^2 \bullet_1^2$



$\bullet_0^2 \bullet_2^2$

Coordinate Transformation cont´d
Jacobi Matrix and Jacobian Determinant

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

$$\mathbf{J} = \frac{\partial x}{\partial \xi}, \qquad \mathcal{J} = \left|\frac{\partial x}{\partial \xi}\right|$$

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

The Jacobian describes the volume change of the element

3D

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

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

$$dx\ dy\ dz = \mathcal{J}\ d\xi\ d\eta\ d\zeta$$

2. Interpolation on the Elements

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

interpolation

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

$$\nabla u_e(\xi) \approx \sum_{i=0}^{N} u_e(\xi_i)\ell_i'(\xi)$$

interpolating functions

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

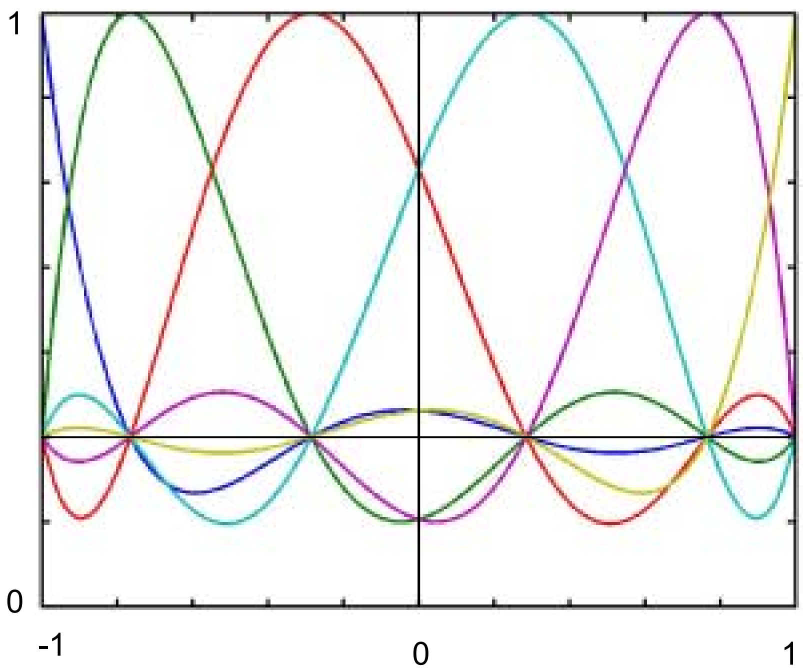
$$\ell_i(\xi_j) = \delta_{ij}$$

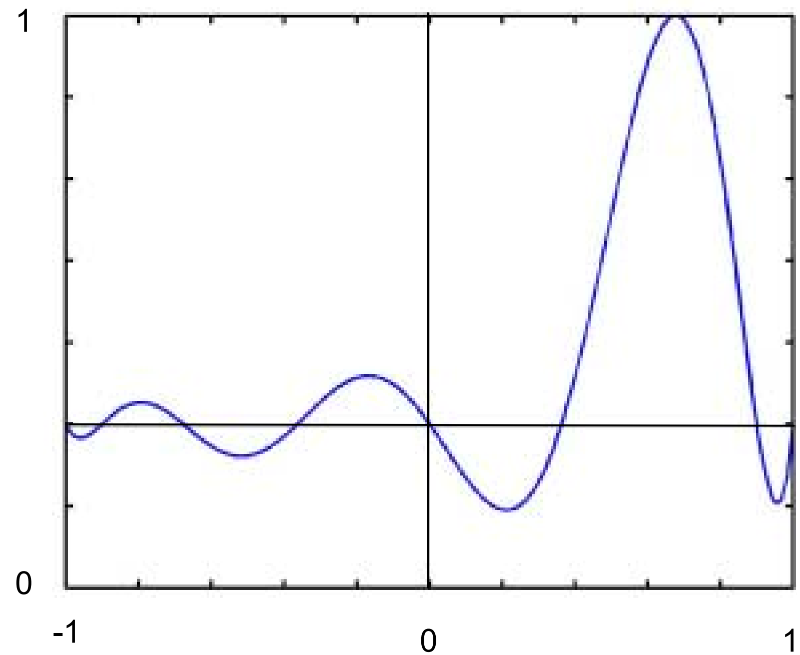
Polynomial degree N for interpolation is usually higher than that for the mapping

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 ●$_7$
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

$\rightarrow$ diagonal mass matrix

(this we will see on the next slides – remember $\ell_i(\xi_j) = \delta_{ij}$)

$$\int_{\Lambda} f(\xi)\ dx = \sum_{i=0}^{N} \omega_i\ f(\xi_i)$$

GLL weights of integration

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

$$\omega_i = \frac{2}{N(N+1)} \qquad (\xi_i = \pm 1)$$

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: $\int_\Omega \rho \; v \; \ddot{u} \; dx \; + \; \int_\Omega \nabla v \; \mu \; \nabla u \; dx = \int_\Omega v \; f \; dx$

coordinate transformation

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

interpolating v and u

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

integration quadrature

$$= \sum_{k=0}^{N} \{\rho(\xi_k) \; \omega_k [\sum_{i=0}^{N} v_i \; \ell_i(\xi_k)] \; [\sum_{j=0}^{N} \ddot{u}_j \; \ell_j(\xi_k)] \; \mathcal{J}(\xi_k)\}$$

weights

Mass Marix cont´d

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

Thanks to Kronecker delta the formula is getting simpler!

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

Rearranging we get

which can be expressed as

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

Finally the world is simple again!

$$\boxed{m e_{ij} = \rho_i \;\omega_i \;\mathcal{J}_i \;\delta_{ij}}$$

4. The Elemental Matrices – Stiffness Matrix

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
$\rightarrow$ the stiffness matrix is not diagonal!

$$ke_{ij} = \sum_{k=0}^{N} \mu_k \; \omega_k \; \ell_i^{'}(\xi_k)\ell_j^{'}(\xi_k) \; \mathcal{J}_k$$

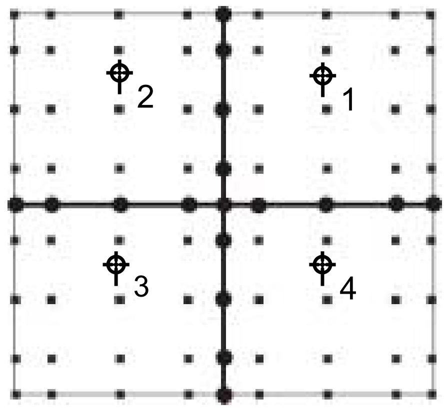
All elements of the elemental stiffness matrix are therefore nonzero

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?



$\rightarrow$ „Connectivity Matrix“

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

Assembling the Global Matrices

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

i indicates element number
j and k indicate the N+1 nodes

$$\mathbf{M}(\mathbf{C_{j,i}}) = \mathbf{M}(\mathbf{C_{j,i}}) + me_j^{(i)}$$

$$\mathbf{K}(\mathbf{C_{k,i}}, \mathbf{C_{j,i}}) = \mathbf{K}(\mathbf{C_{k,i}}, \mathbf{C_{j,i}}) + ke_{k,j}^{(i)}$$

$$\mathbf{M} = \begin{pmatrix} \star \\ \star \\ \star \\ \\ \\ \\ \\ \end{pmatrix} + \begin{pmatrix} \\ \\ \circ \\ \circ \\ \circ \\ \\ \\ \end{pmatrix} + \begin{pmatrix} \\ \\ \\ \\ \diamond \\ \diamond \\ \diamond \end{pmatrix} = \begin{pmatrix} \star \\ \star \\ \star + \circ \\ \circ \\ \circ + \diamond \\ \diamond \\ \diamond \end{pmatrix}$$

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

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 **F**)

   Advantage: <u>much</u> easier to implement in 2- and 3D

   Drawback: CPU time increases

<u>Calculation of Forces</u>

strain at node i: $$\frac{\partial u_i}{\partial x} = \frac{\partial u_i}{\partial \xi}\,\frac{\partial \xi_i}{\partial x}$$

1. x

$$= \sum_j u_j\, \ell_j'(\xi_i) \cdot \mathbf{J}_i^{-1}$$

Hooke´s Law → stress: $$\sigma_i = \mu_i \frac{\partial u_i}{\partial x}$$

$$\mathbf{f}_{int}^{(e)} = \int_{\Omega_e} \nabla v\ \sigma$$

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

2. x

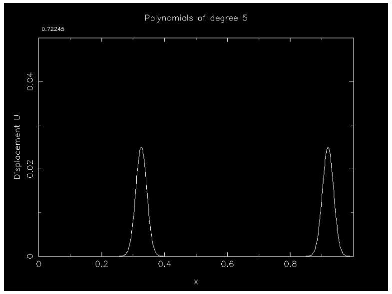
$$= \sum_k \left[\sum_j \ell_j'(\xi_k)\right] \mathbf{J}_k^{-1}\ \sigma_k\ \mathcal{J}_k$$

Note: Here we need to correct for the coordinate transformation (same for stiffness matrix)
→ 2 x Jacobi Matrix of inverse transformation → 1 x Jacobian Determinant

Boundary Conditions

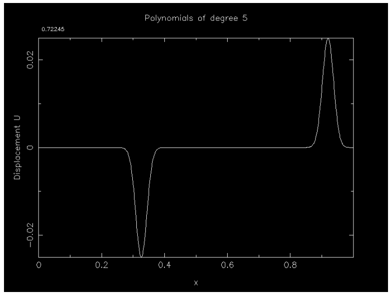
Implementation of different boundary conditions in the SEM formulation is very easy

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



$$\mathbf{U}_{t_{i+1}} = \Delta t^2 \cdot \mathbf{M}^{-1}\,\mathbf{F} + 2\mathbf{U}_{t_i} - \mathbf{U}_{t_{i-1}}$$

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

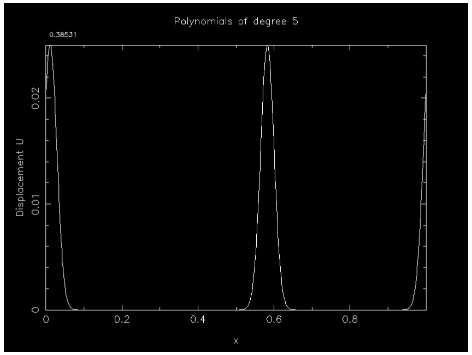


$$\mathbf{U}_{t_{i+1}}(2:ng-1) = \Delta t^2 \cdot \mathbf{M}^{-1}(2:\mathrm{ng}-1)\ \mathbf{F}(2:\mathrm{ng}-1) + $$
$$+ 2\mathbf{U}_{t_i}(2:ng-1) - \mathbf{U}_{t_{i-1}}(2:ng-1)$$

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

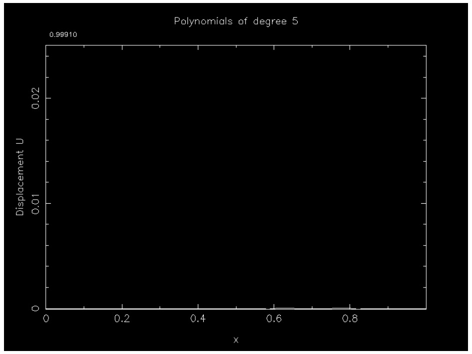


$\rightarrow$ $F(1)_{periodic} = F(1)_{fs} + F(ng)_{fs}$ same for $F(ng)_{periodic}$

$\rightarrow$ $M(1)_{periodic} = M(1)_{fs} + M(ng)_{fs}$ and $M(ng)_{periodic}$

4. Absorbing boundary conditions:

stress conditions at the edges: $\boxed{\sigma_\Gamma = \rho\, v_S\, \dot{u}}$



$\rightarrow$ $F(1)_{absorbing} = F(1)_{fs} + \rho(1)\, v_S(1)\, \dot{u}(1)$

$\rightarrow$ $F(1)_{absorbing} = F(1)_{fs} + \rho(ng)\, v_S(ng)\, \dot{u}(ng)$

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**

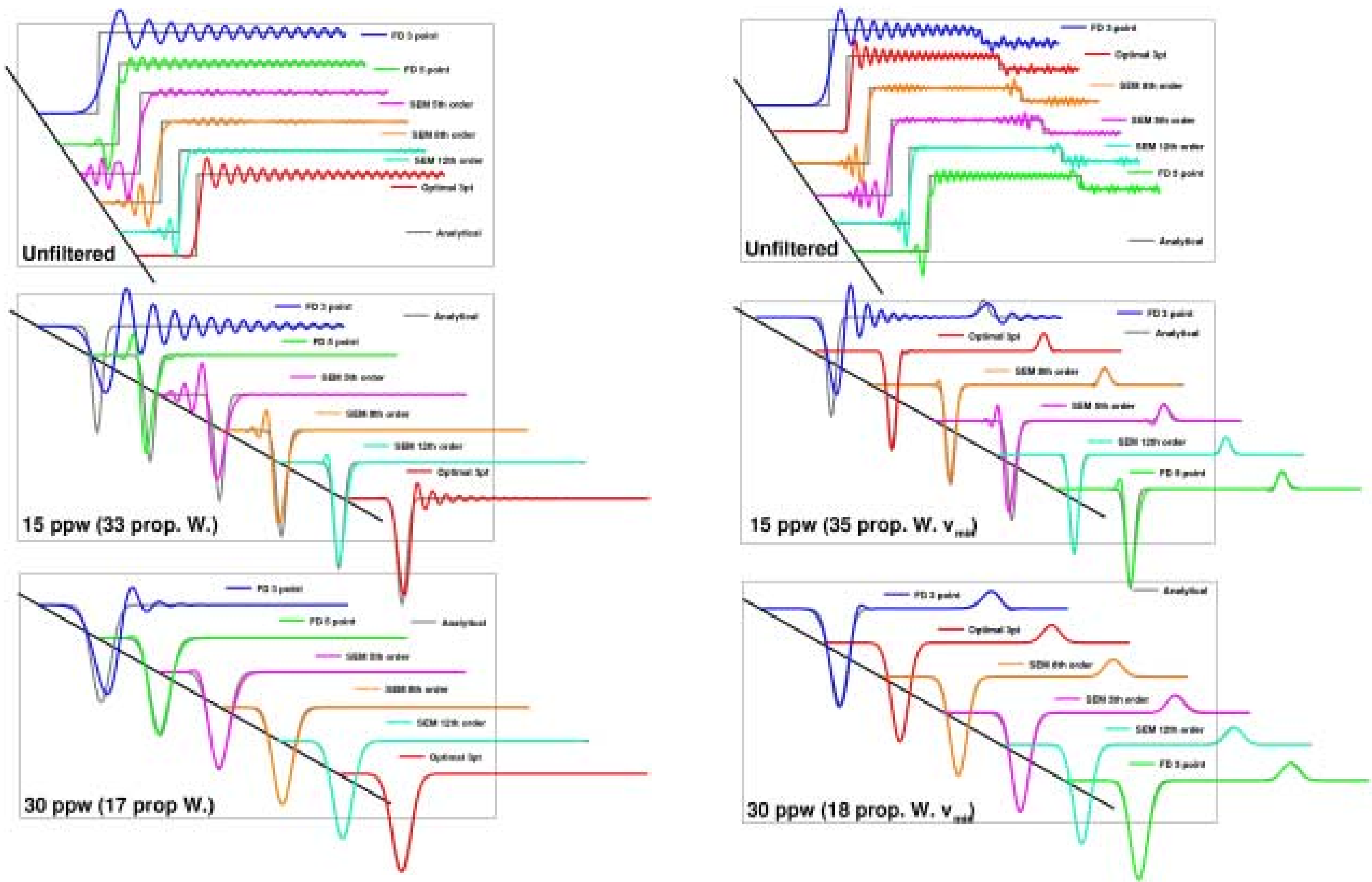
Comparison of SEM and Optimal FD Operators

- 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

- Model size — 1200 grid points
- Source — Delta peak in space and time
- Receiver Array — every 5th gridpoint $\rightarrow$ 240 receivers
- Effective Courant Number — 0.5 (grid spacing varies in SEM)
  0.82 used in stability criterion
- Filtering — 5 to 40 points per wavelength (ppw)
  frequencies chosen correspondingly
- Propagation length — 1 to 40 propageted wavelengths (npw)
- Analytical solution (displacement) — Heaviside shaped
  first arrival in seismogram at time $t = \frac{x}{v}$

Seismograms
Homogeneous and Two Layered Medium



How to check for performance?

synthetic seismograms + analytical solution $\rightarrow$ relative solution error (rse)

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

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

What determines the perfomance 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

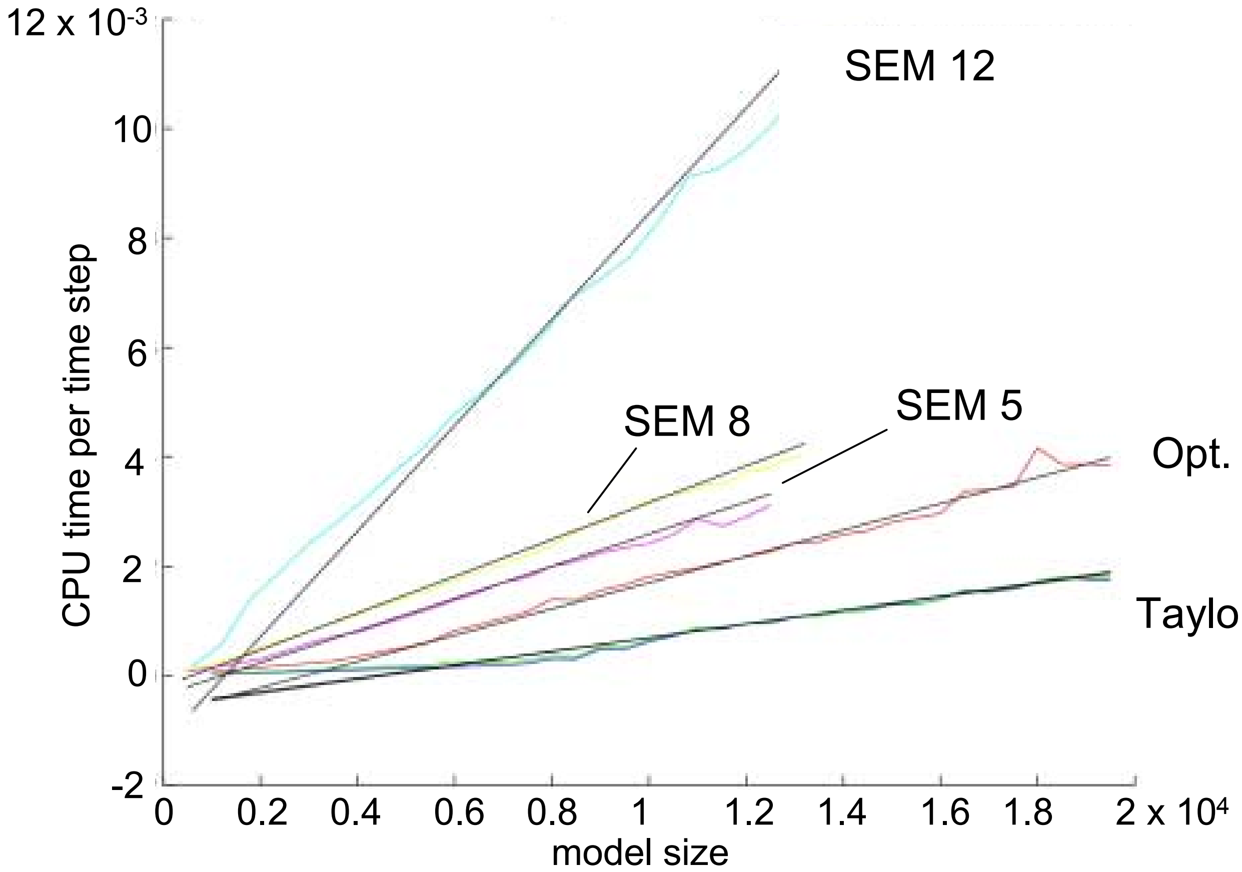
$$Cost = ppw\,(rse) \cdot npw \cdot \frac{cpt}{nx}$$

and the memory usage.

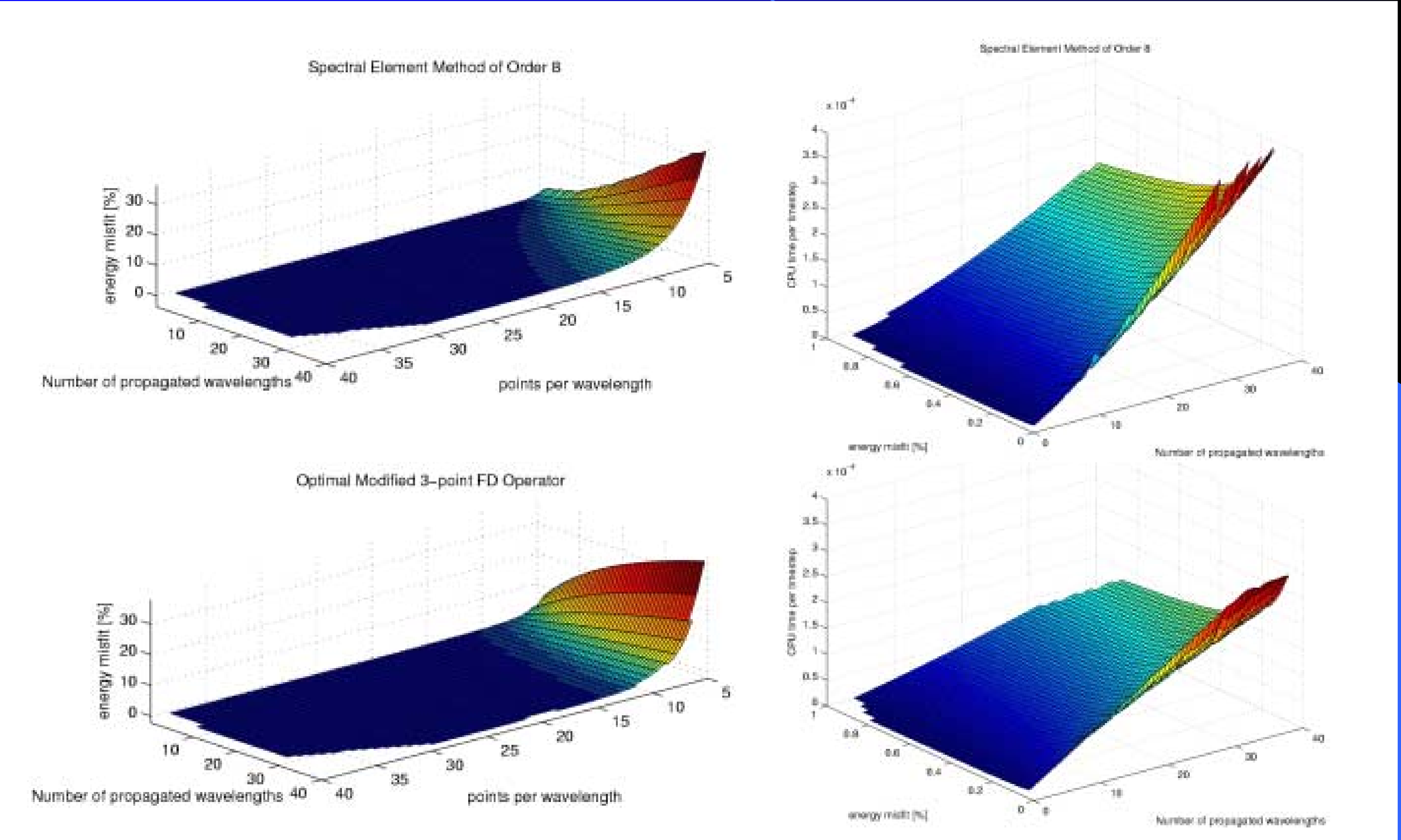
CPU Cost

Benchmarking the CPU time

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

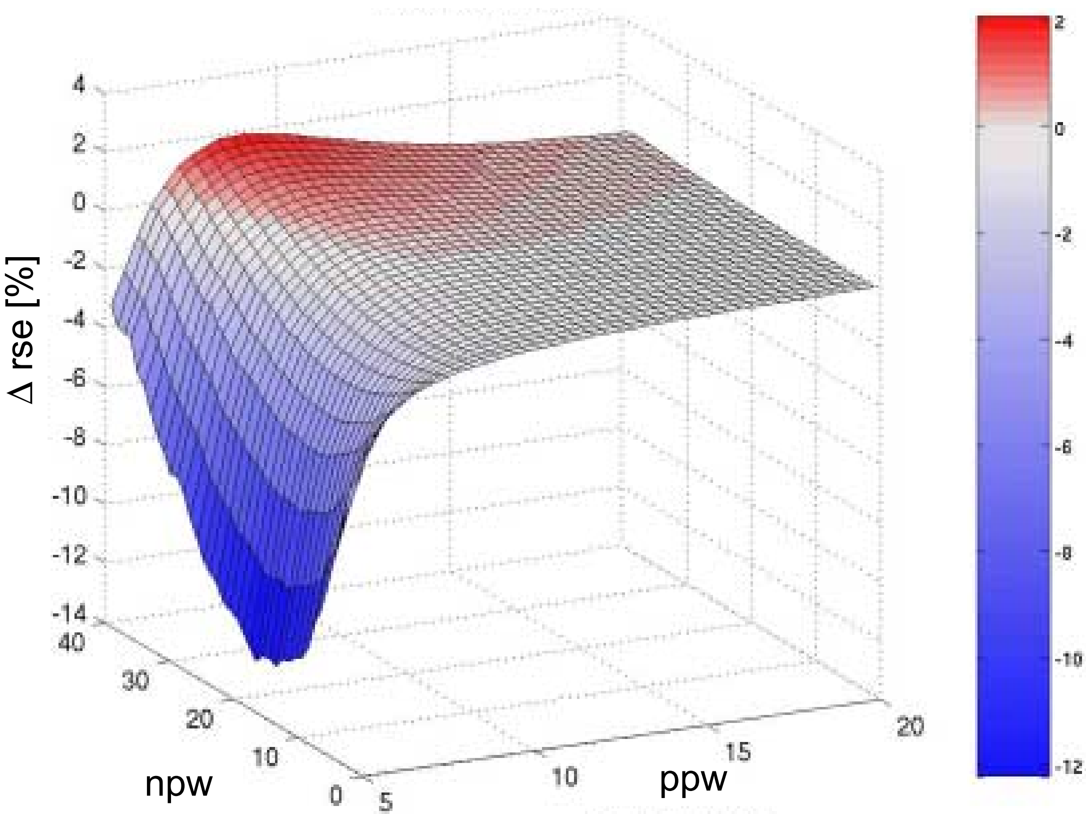


Results – Homogeneous Case

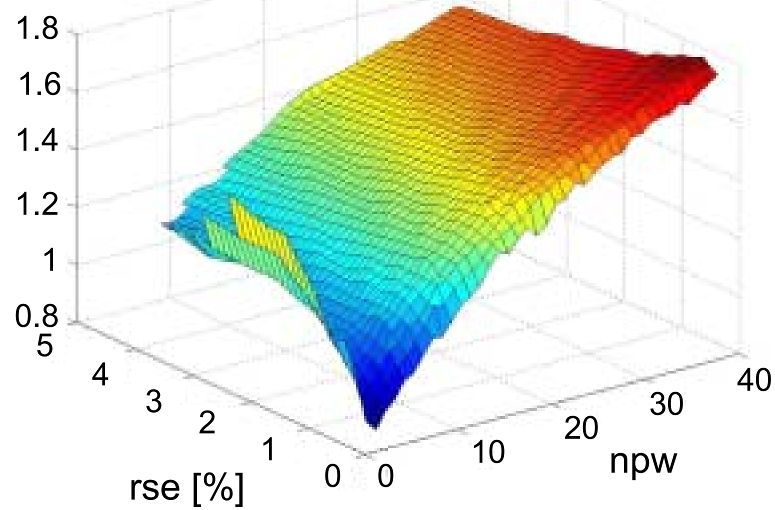


Homogeneous Model– RSE Difference and Relative CPU time

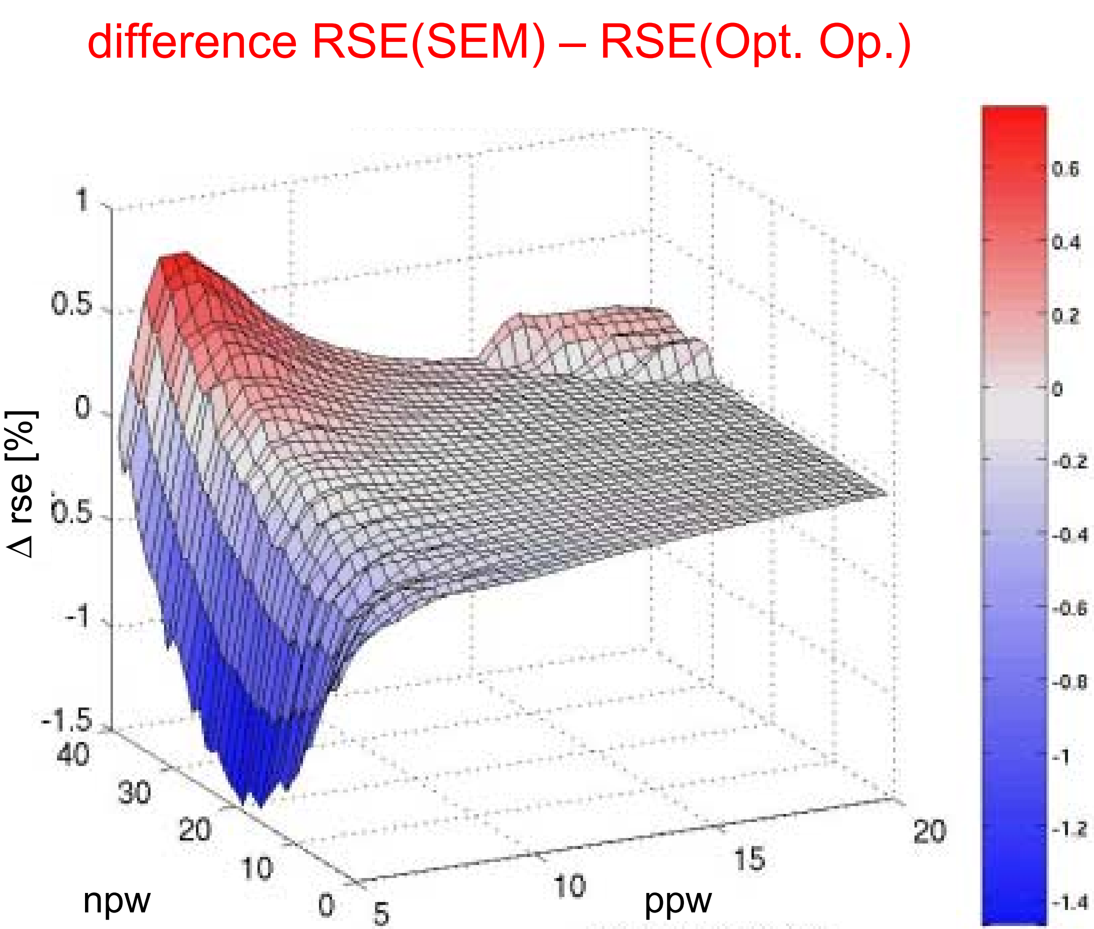
difference RSE(SEM) – RSE(Opt. Op.)

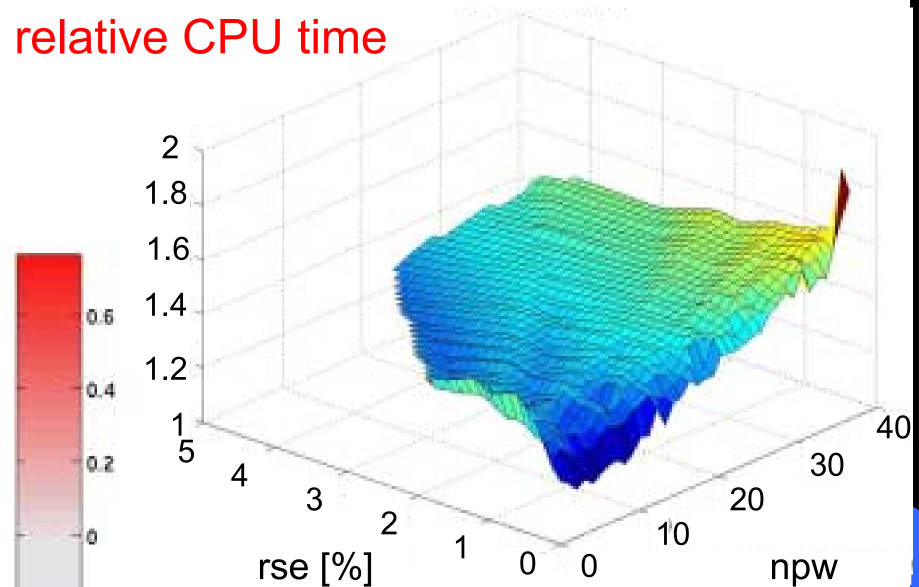


relative CPU time



Two Layered Model– RSE Difference and Relative CPU time





Conclusions

- Memory $\rightarrow$ 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

  $\rightarrow$ Opt. Op. win for the used setup (without boundaries) in 1D

  $\rightarrow$ SEM in heterogeneous model better than in homogeneous compared to Opt. Op.
- 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

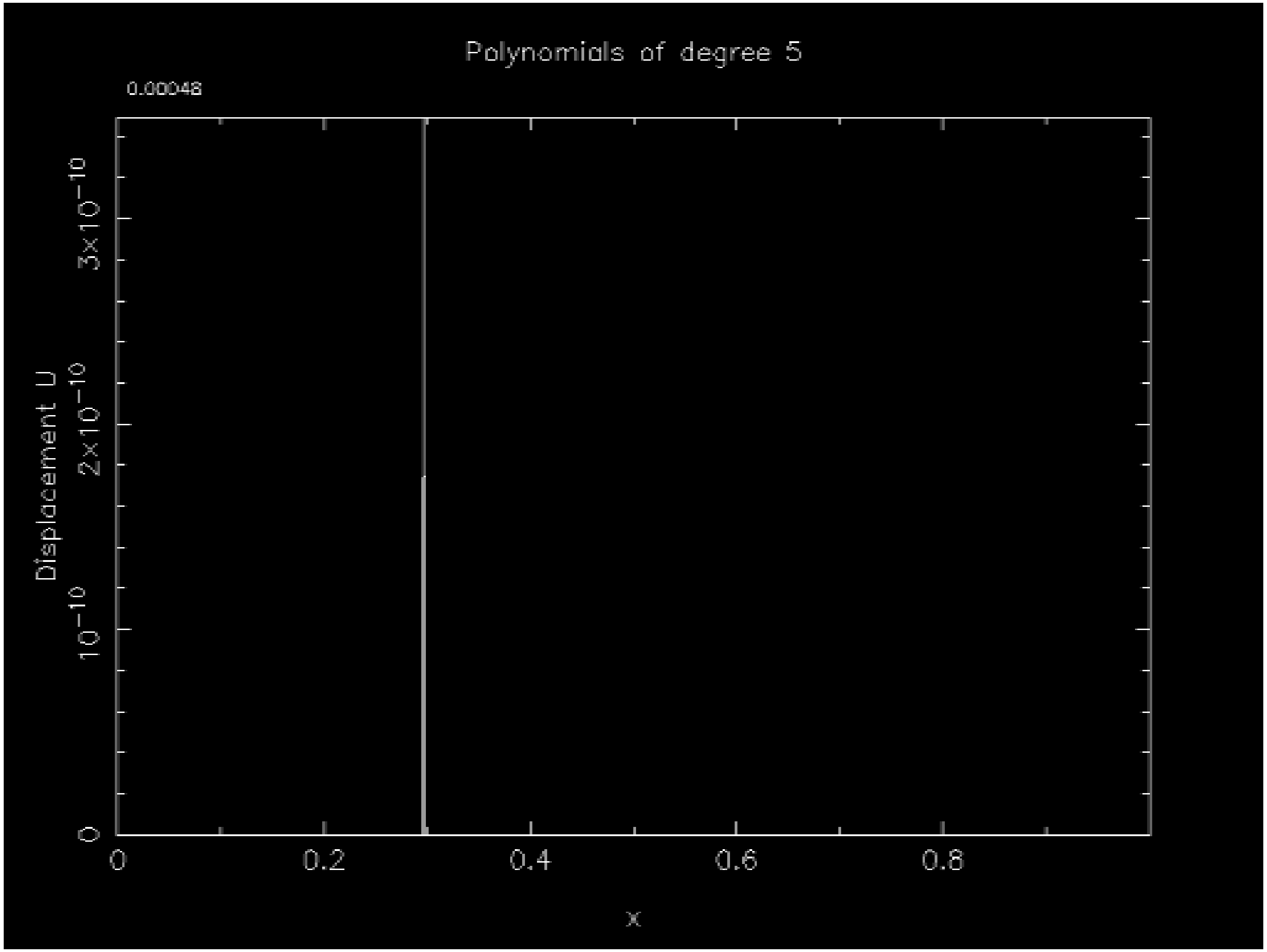
3D

- 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!

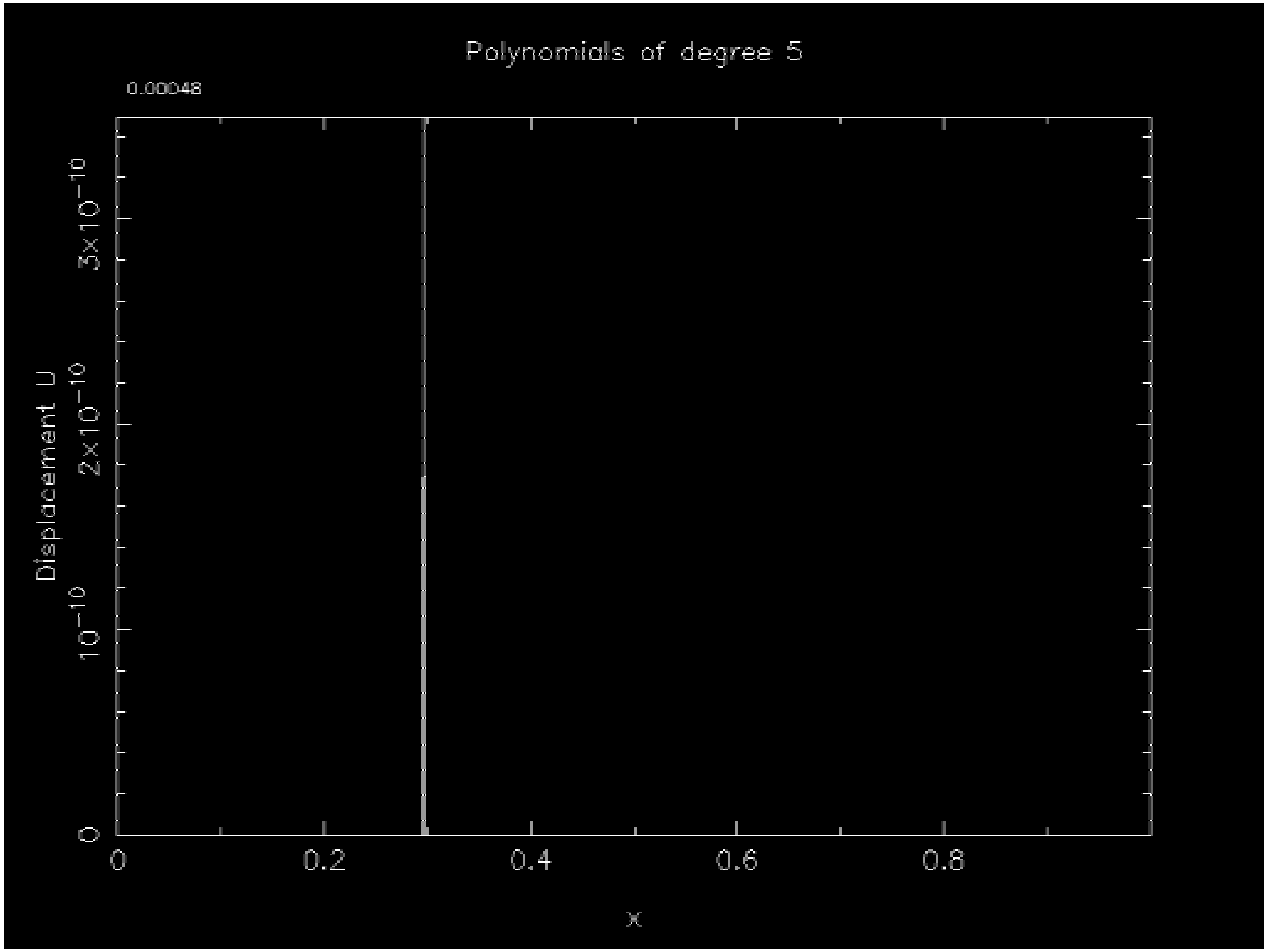
Bernhard Schuberth

Boundary Conditions – Examples Free Surface



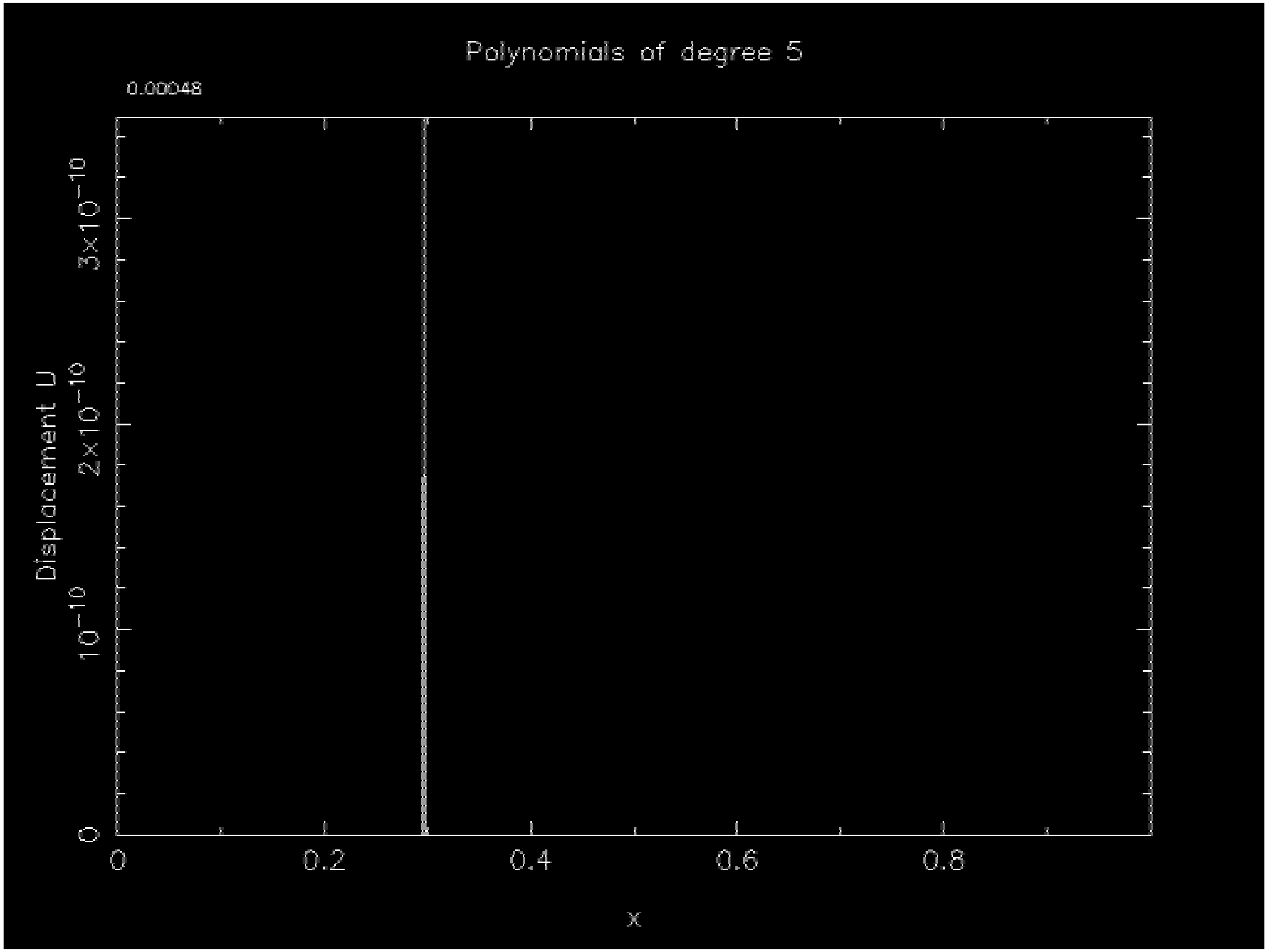
back

Boundary Conditions – Examples
Rigid Boundaries



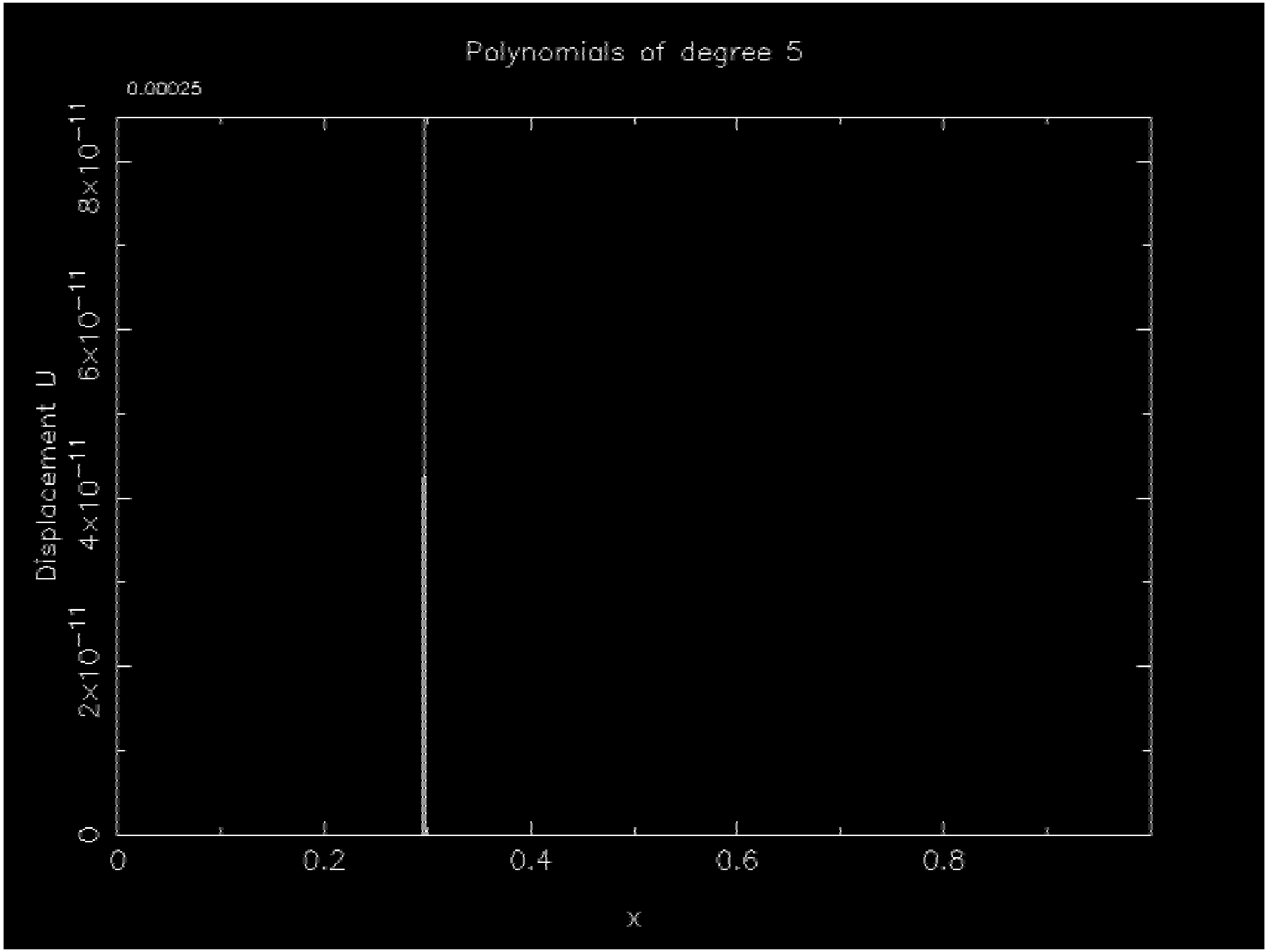
back

Boundary Conditions – Examples
Periodic Boundaries



back

Boundary Conditions – Examples Absorbing Boundaries



back

Results – Two Layered Medium