# **Computational Seismology**

Lecture 5: Spectral-element Method

March 24, 2021

University of Toronto

1. History of SEM

2. SEM in a nutshell

3. Elemental level

**History of SEM** 

### FD vs. FEM vs. SEM

- **FD**: difficulty with accurate implementation of free-surface b.c. and complex geometry.
- Classic low-order FEM: free surface b.c. FOR FREE; unstructured tetrahedral and hexahedral mesh for complex model geometry. → a large linear system, challenging on parallel computers.
- SEM: one of the most widely used numerical methods for seismic wave propagation. It is an FEM with high-order Lagrange polynomials as interpolating functions within an element (quadrilateral elements in 2D and hexahedral elements in 3D) for the field (stress, strain and displacement) combined with interpolation scheme based on the Gauss-Lobatto-Legendre (GLL) collocation points → diagonal mass matrix; meshing can be challenging

#### Meshing challenge in SEM



A spectral-element mesh to model soil-structure interactions with a hexahedral mesh. Note the variations in element size and the deformation of the hexahedral elements with curved boundaries (Stuppazzini et al 2005)

# **History of SEM**

- SEM is inspired by pseudospectral methods applied at elemental level (spectral/exponential convergence properties of the basis functions)
- Fluid dynamics: Patera (1984) and Maday and Patera (1989)
- Elastic wave propagation: Priolo et al (1994), Seriani and Priolo (1994), and Faccioli et al. (1996): first with Chebyshev polynomials  $\rightarrow$  need to invert large linear system
- Breakthrough: Komatitisch and Vilotte (1998): combination of Lagrange polynomials as interpolants and an integration scheme based on Gauss quadrature defined on the GLL points → diagonal mass matrix, easy to implement in parallel.

### History of SEM: development

 Wave propagation in 3D heterogeneous spherical Earth: first by Chaljub (2000) and Chaljub et al. (2003) using the cubed-sphere concept (Ronchi et al 1996). Later on community code SPEFEM3D\_GLOBE (Komatitsch and Tromp, 2002a,b) which also allows regional wave propagation using one to three chunks.



### History of SEM: regional code

- 3D spherical sections SEM in a spherical coordinates: SES3D (Fichtner and Igel, 2008, Gokhberg and Fichtner, 2016)
- SEM for complex local cartesian models: SPECFEM3D\_CARTESIAN (Komatitsch et al 2004, Peter et al 2011)
- Hexhedral meshing: challenging task to accommodate free-surface topography, and the curved, discontinuous internal boundaries. → Cubit/Trelis, GMSH, etc

# **SEM in a nutshell**

### SEM in 1D

For 1D elastic wave equation over domain [0, *L*]

 $\rho \partial_t^2 u = \partial_x (\mu \partial_x u) + f$ 

with traction-free boundary condition on both ends

$$\sigma_{ij}n_j\big|_{surface} = \mu \partial_x u(x,t)\big|_{x=0,L} = 0$$

(top) wave field interpolation in an element by a set of basis functions with order N = 4 collocated points. (bottom) snapshot of a 1D displacement wavefield simulated with SEM for a medium with a random distribution of elastic parameters.



7

- Collocated points on the element are NOT evenly distributed: their choice in combination with the Gauss-Lobatto-Legendre (GLL) quadrature rule gives us the diagonal mass matrix
- **Procedures**: weak-form; transformation to elemental level; interpolation of field by Lagrange polynomials; calculate 1st order derivative of Lagrange polynomial (related to Legendre polynomial); numerical integration on GLL quadrature and system matrix at elemental level; assemble into global system of equations.

#### SEM step 1: weak form

Multiple both sides by an independent test function v(x) (both v(x) and  $\dot{v}(x)$  are square-integrable), and integral over the domain D = [0, L].

$$\int_{D} v \rho \partial_t^2 u \, dx - \int_{D} v \partial_x (\mu \partial_x u) \, dx = \int_{D} v f \, dx \tag{1}$$

Through integration by parts and the use of free boundary (implicitly filled),

$$\partial_x u(x,t)\big|_{x=0} = \partial_x u(x,t)\big|_{x=L} = 0$$
 (2)

we have

$$\int_{D} \mathbf{v} \rho \partial_{t}^{2} u \, dx + \int_{D} \mu \partial_{x} u \partial_{x} v \, dx = \int_{D} \mathbf{v} f \, dx \tag{3}$$

### Step 2: discretization into elements and basis functions

we approximate the exact solution for u(x, t) by a finite superposition of *n* (global) basis functions  $\phi_i(x)$ ,  $i = 1, \dots, N_p$  (note  $N_p$  is usually much less than n)

$$u(x,t) \approx \overline{u}(x,t) = \sum_{i=1}^{N_p} u_i(t)\phi_i(x)$$
(4)

Applying Gelerkin principle (same basis functions as test functions)

$$\int_{D} \phi_{j} \rho \partial_{t}^{2} u \, dx + \int_{D} \mu \partial_{x} u \partial_{x} \phi_{j} \, dx = \int_{D} \phi_{j} f \, dx \tag{5}$$

with i.c. that the medium is at rest at t = 0.

$$\sum_{i=1}^{N_p} \partial_t^2 u_i(t) \int_D \rho \phi_i(x) \phi_j(x) \, dx + \sum_{i=1}^{N_p} u_i(t) \int_D \mu \partial_x \phi_i(x) \partial_x \phi_j(x) \, dx = \int_D \phi_j f \, dx$$
(6)

### Step 3: Assembly

The above  $N_p$  equations can be written into matrix-vector form

$$\mathbf{M}\partial_t^2 \mathbf{u}(t) + \mathbf{K} \mathbf{u}(t) = (\mathbf{t}) \tag{7}$$

where the mass matrix

$$M_{ji} = \int_D \rho \phi_i(x) \phi_j(x) \, dx \tag{8}$$

and the stiffness matrix

$$K_{ji} = \int_{D} \mu \partial_{x} \phi_{i}(x) \partial_{x} \phi_{j}(x) \, dx \tag{9}$$

The solved coefficients gives *u* field at a set of global points.



11

With 
$$u^{new} = u(t + dt)$$
,  $u = u(t)$  and  $u^{old} = u(t - dt)$   
 $u^{new} = 2u - u^{old} + dt^2 [\mathbf{M}^{-1}(-\mathbf{Ku})]$  (10)

# **Elemental level**

# Meshing

Divide the entire domain D into subdomains  $D_e$  which can be of different sizes and allow discontinuity of material interfaces to be honored, leading to possible discontinuity in stress and strain.



### Weak form at elemental level

After the discretization of *u* based on basis function and applying the Galerkin principle, we can conduct the integrations at elemental level:

$$\sum_{i=1}^{N_p} \partial_t^2 u_i(t) \sum_{e=1}^{N_e} \int_{D_e} \rho \phi_i(x) \phi_j(x) dx$$
$$+ \sum_{i=1}^{N_p} u_i(t) \sum_{e=1}^{N_e} \int_{D_e} \mu \partial_x \phi_i(x) \partial_x \phi_j(x) dx = \sum_{e=1}^{N_e} \int_{D_e} \phi_j(x) f dx \qquad (11)$$

Note that the basis functions  $\phi_i(x)$  although are global, for the purpose of elemental integration (with transformation to standard domain), they only need to be defined locally over the subdomain  $D_e$  (or defined piecewise over elements) as  $\phi_i^e(x)$ .

$$\overline{u}(x,t)\Big|_{x\in D_{\mathsf{e}}} = \sum_{i=1}^{N_{\mathsf{p}}} u_i^{\mathsf{e}}(t)\phi_i^{\mathsf{e}}(x) \tag{12}$$

where  $N_p$  is the number of basis functions needed to sum up the field within an element.

#### Integrals at elemental level

Also carry out all the integrations in the elemental level as

$$\sum_{e=1}^{N_e} \sum_{i=1}^{N_p} \partial_t^2 u_i^e(t) \int_{D_e} \rho \phi_i^e(x) \phi_j^e(x) dx$$
$$+ \sum_{e=1}^{N_e} \sum_{i=1}^{N_p} u_i^e(t) \int_{D_e} \mu \partial_x \phi_i^e(x) \partial_x \phi_j^e(x) dx = \sum_{e=1}^{N_e} \int_{D_e} \phi_j^e(x) f dx$$
(13)

and can be assembled into

$$\sum_{e=1}^{N_e} \left[ \mathbf{M}^e \partial_t^2 \mathbf{u}^e(t) + \mathbf{K}^e \mathbf{u}^e(t) \right] = \sum_{e=1}^{N_e} \mathbf{f}^e(t)$$
(14)

For SEM, the standard 1D domain is defined as [-1, 1], and the mapping between this standard domain and  $D_e = [x_e, x_{e+1}]$  is

$$x = F_{e}(\xi), \quad \xi = \xi(x) = F_{e}^{-1}(x), \quad , e = 1, \cdots, N_{e}$$
 (15)

For example in a linear mapping,

$$x(\xi) = F_{e}(\xi) = h_{e} \frac{\xi + 1}{2} + x_{e}, \quad \xi(x) = 2 \frac{x_{e}}{h_{e}} - 1$$
 (16)

where  $h_e$  is the element size given by  $x_{e+1} - x_e$ . And integrals become

$$\int_{D_e} f(x) \, dx = \int_{-1}^{1} f(\xi) \frac{dx}{d\xi} \, d\xi \tag{17}$$

with the jacobian and inverse jacobian defined as

$$J = \frac{dx}{d\xi} = \frac{h_e}{2}, \quad J^{-1} = \frac{d\xi}{dx} = \frac{2}{h_e}$$
 (18)

Therefore at elemental level, for each test function chosen as  $\phi_j(x)$ , the three integrals become

$$\sum_{i=1}^{N_p} \partial_t^2 u_i^{\mathsf{e}}(t) \int_{D_e} \rho(\mathbf{x}(\xi)) \phi_i^{\mathsf{e}}(\mathbf{x}(\xi)) \phi_j^{\mathsf{e}}(\mathbf{x}(\xi)) \frac{d\mathbf{x}}{d\xi} d\xi$$
$$+ \sum_{i=1}^{N_p} u_i^{\mathsf{e}}(t) \int_{D_e} \mu(\mathbf{x}(\xi)) \partial_{\xi} \phi_i^{\mathsf{e}}(\mathbf{x}(\xi)) \partial_{\xi} \phi_j^{\mathsf{e}}(\mathbf{x}(\xi)) \left(\frac{d\xi}{d\mathbf{x}}\right)^2 \frac{d\mathbf{x}}{d\xi} d\xi$$
$$\sim \int_{D_e} \phi_j^{\mathsf{e}}(\mathbf{x}(\xi)) f(\mathbf{x}(\xi)) \frac{d\mathbf{x}}{d\xi} d\xi \qquad (19)$$

What we need to do next: 1) find the proper basis functions  $\phi_j(x(\xi)), j = 1, \dots, N_p$  for interpolation; 2) find an integration scheme; 3) assemble the system matrices and vectors.

### Interpolation with Lagrange polynomials

Choose the Lagrange polynomials as the interpolation functions within the standard domain [-1, 1]

$$\phi(\mathbf{x}(\xi)) \to l_i^N(\xi) := \prod_{j \neq i}^{N+1} \frac{\xi - \xi_j}{\xi_i - \xi_j}, \quad i, j = 1, 2, \cdots, N, N+1$$
 (20)

Or explicitly

$$I_{i}^{N}(\xi) = \frac{\xi - \xi_{1}}{\xi_{i} - \xi_{1}} \frac{\xi - \xi_{2}}{\xi_{i} - \xi_{2}} \cdots \frac{\xi - \xi_{i-1}}{\xi_{i} - \xi_{i-1}} \frac{\xi - \xi_{i+1}}{\xi_{i} - \xi_{i+1}} \frac{\xi - \xi_{N}}{\xi_{i} - \xi_{N}} \frac{\xi - \xi_{N+1}}{\xi_{i} - \xi_{N+1}}$$
(21)

which implies that

$$I_i^N(\xi_j) = \delta_{ij} \tag{22}$$

Another questions: how do we choose the  $\xi_i$ ,  $i = 1, \dots, N+1$  points?  $\rightarrow$  Gauss-Lobatto-Legendre (GLL) points.

### Why GLL points?

#### With these choice of GLL points, it allows

$$I_{i}^{(N)}(\xi_{i}) = 1, \quad \dot{I}_{i}^{(N)}(\xi_{i}) = 0, \rightarrow |I_{i}^{(N)}(\xi_{i})| \leq 1$$
 (23)



19

# GLL points of order N

GLL points are the roots of the first derivative of the Legendre polynomials LN of degree N



- The densification of points towards the boundaries avoids overshooting of the interpolated function near the boundaries
- an integration scheme exists (GLL quadrature) that uses precisely this point set, leading to a diagonal mass matrix.

### Interpolation of wavefield

Interpolation of wavefield at elemental level by N + 1 polynomials of order N:  $u^{e}(\xi) = \sum_{i=1}^{N+1} u^{e}(\xi_{i}) I_{i}(\xi)$ 



Accuracy of interpolation increases with increasing order N, but the highest possible order is not necessarily the best.  $\rightarrow$  what is the right balance?

# System Matrices under GLL

With the Lagrange polynomials defined over the GLL points

$$\sum_{i=1}^{N+1} \partial_t^2 u_i^{\mathsf{e}}(t) \int_{D_e} \rho(x(\xi)) l_i(x(\xi)) l_j(x(\xi)) \frac{dx}{d\xi} d\xi + \sum_{i=1}^{N+1} u_i^{\mathsf{e}}(t) \int_{D_e} \mu(x(\xi)) \partial_{\xi} l_i(x(\xi)) \partial_{\xi} l_j(x(\xi)) \left(\frac{d\xi}{dx}\right)^2 \frac{dx}{d\xi} d\xi \sim \int_{D_e} l_j(x(\xi)) f(x(\xi)) \frac{dx}{d\xi} d\xi$$
(24)

where  $j = 1, \dots, N + 1$ . The only unknown are  $u_i$ 's and  $\partial_t u_i$ 's. Density and shear moduli need to be known at each collocation point (so no need to assume they are constant over the element as in FEM)

These integrals have to be evaluated numerically  $\rightarrow$  quadrature rules for numerical integration

Numerical integration (synonym for numerical quadrature) is a vast field of its own (even intervals: midpoint, trapezoid, Simpson's; varying intervals: Gaussian quadrature)

The idea is to replace the integration function f(x) by a polynomial approximation (e.g., degree 2N + 1 that can be integrated analytically (e.g., with only N + 1 collocation points). In the Gaussian quadarture, the N + 1 points do not include the boundary points. Instead the Gauss-Lobatto-Legendre (GLL) quadrature can include the boundary points (i.e, over all GLL points).

$$\int_{-1}^{1} f(x) \, dx \approx \int_{-1}^{1} P_N(x) \, dx = \sum_{i=1}^{N+1} w_i f(x_i) \tag{25}$$

where  $P_N(x) = \sum_{i=1}^{N+1} f(x_i) I_i^{(N)}(x)$  and weights  $w_i = \int_{-1}^{1} I_i^{(N)}(x) dx$  23

Collocation points  $\xi_i$  and weights  $w_i$  for different order *N*.

Table 7.1 Collocation points and integration weights of the GLL quadrature for order $N=2,\ldots,4.$		
Ν	ξi	$\omega_{i}$
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

#### Numerical integration example

For a sinusoidal function

$$f(\xi) = \sum_{i=1}^{5} \sin\left(\frac{\pi}{a_i}\xi + a_i\right) \qquad (26)$$

with a = [0.5, 1, -3, -2, -5, 4] which can be integrated analytically. (Right):

true function (thick line) vs approx Lagrange polynomial (thin line) and integration (dark grey)  $\rightarrow$  increased accuracy with higher order



# Applying GLL quadrature to System Matrices

$$\sum_{i=1}^{N+1} \partial_t^2 u_i^{\mathsf{e}}(t) \int_{D_e} \rho(x(\xi)) l_i(x(\xi)) l_j(x(\xi)) \frac{dx}{d\xi} d\xi$$

$$+ \sum_{i=1}^{N+1} u_i^{\mathsf{e}}(t) \int_{D_e} \mu(x(\xi)) \partial_{\xi} l_i(x(\xi)) \partial_{\xi} l_j(x(\xi)) \left(\frac{d\xi}{dx}\right)^2 \frac{dx}{d\xi} d\xi$$

$$\sim \int_{D_e} l_j(x(\xi)) f(x(\xi)) \frac{dx}{d\xi} d\xi \Rightarrow \qquad (27)$$

$$\sum_{i,k=1}^{N+1} \partial_t^2 u_i^{\mathsf{e}}(t) w_k \rho(\xi_k) l_i(\xi_k) l_j(\xi_k) \frac{dx}{d\xi} \Big|_{\xi=\xi_k}$$

$$+ \sum_{i,k=1}^{N+1} u_i^{\mathsf{e}}(t) w_k \mu(\xi_k) \partial_{\xi} l_i(\xi_k) \partial_{\xi} l_j(\xi_k) \left(\frac{d\xi}{dx}\right)^2 \frac{dx}{d\xi} \Big|_{\xi=\xi_k}$$

$$\sim \int_{D_e} l_j(x(\xi)) f(x(\xi)) \frac{dx}{d\xi} d\xi \qquad (28)$$

26

### **GLL Quadrature to system matrices**

Using 
$$I_i^{(N)}(\xi_j) = \delta_{ij}$$
, it is simplified to  
$$\sum_{i=1}^{N+1} M_{ji}^e \partial_t^2 u_i^e(t) + \sum_{i=1}^{N+1} K_{ji}^e u_i^e(t) \sim f_j^e(t), \quad e = 1, \cdots, n_e$$

where

$$M_{ji}^{e} = w_{j}\rho(\xi_{j}) \frac{dx}{d\xi} \Big|_{\xi = \xi_{j}} \frac{\delta_{ij}}{\delta_{ij}}$$

$$K_{ji}^{e} = \sum_{k=1}^{N+1} w_{k}\mu(\xi_{k})\partial_{\xi}I_{i}(\xi_{k})\partial_{\xi}I_{j}(\xi_{k}) \left(\frac{d\xi}{dx}\right)^{2} \frac{dx}{d\xi} \Big|_{\xi = \xi_{k}}$$

$$f_{j}^{e} = w_{j}f(\xi, t)\frac{dx}{d\xi_{j}}\Big|_{\xi = \xi_{j}}$$
(29)

These are the core computation, especialy  $K_{ji}^e$ . Note we need also the derivatives of the Lagrange polynomials at the collocation points.

### **Derivative of Lagrange polynomial**

For details, see Funaro (1993):

$$\partial_{\xi}I_k(\xi_i) = \sum_{j=1}^N d_{ij}I_k(\xi_j), \quad k = 0, \cdots, N$$
(30)

where

$$d_{ij} = \begin{cases} -\frac{1}{4}N(N+1) & i = j = 0, N \\ 0 & 1 \le i = j \le N-1 \\ \frac{L_N(\xi_i)}{L_N(\xi_j)} \frac{1}{\xi_i - \xi_j} & i \ne j \end{cases}$$
(31)

In the SEM code,  $\partial_{\xi} l_k(\xi_i)$  at collocation point are given as an array, and can be used to compute field derivatives as

$$\partial_{\xi} u^{\mathsf{e}}(\xi) = \sum_{i=0}^{N+1} u^{\mathsf{e}}(\xi_i) \partial_{\xi} l_i(\xi)$$
(32)

### **Global Assembly**

Recall the weak form of the equation

$$\sum_{e} \sum_{i=1}^{N+1} M_{ji}^{e} \partial_{t}^{2} u_{i}^{e}(t) + \sum_{e} \sum_{i=1}^{N+1} K_{ji}^{e} u_{i}^{e}(t) = \sum_{e} f_{j}^{e}(t), \quad e = 1, \cdots, n_{e}$$
$$M_{ji}^{e} = w_{j} \rho(\xi_{j}) \left| \frac{dx}{d\xi} \right|_{\xi = \xi_{j}} \delta_{ij} \quad diagonal$$
$$K_{ji}^{e} = \sum_{k=1}^{N+1} w_{k} \mu(\xi_{k}) \partial_{\xi} I_{i}(\xi_{k}) \partial_{\xi} I_{j}(\xi_{k}) \left( \frac{d\xi}{dx} \right)^{2} \frac{dx}{d\xi} \Big|_{\xi = \xi_{k}}$$
$$f_{j}^{e} = w_{j} f(\xi, t) \frac{dx}{d\xi_{j}} \Big|_{\xi = \xi_{j}}$$
(33)

Assuming continuity of displacement fields at the elemental boundaries, we can assemble it into global matrices



29

#### **Global mass matrices**

In 1D , the global number of collocation points  $N_g = n_e \times N_e + 1$  (NGLOB), and the global mass matrix assembled from the local mass matrix is diagonal (i.e., as a vector).

Example below (left)  $N_e = 3$  and N = 2, (right)  $N_e = 4$ , N = 4.



# **Global Stiffness matrix**

#### Global stiffness matrix (banded in 1D) $K_g =$



with each element being the size of  $(N + 1) \times (N + 1)$ .

### Global stiffness matrix and source vector

For general irregular grids (hexahedral or tetrahedral) in 2D and 3D (hence irregular connectivity),  $K_g$  becomes



#### **Source Vector**

Source vector assembly: Single point source vs. finite source. Collocation point vs. non-collocation point; In the vicinity of a point source, the solution may be erroneous as the near-field terms are not properly represented, but accurate when two elements away.



Assembly them together we obtain  $N_g$  number of system of equations (a linear system of  $N_g \times N_g$ )

$$\mathbf{M}_{g}\ddot{\mathbf{u}}_{g} + \mathbf{K}_{g}\mathbf{u}_{g} = \mathbf{f}_{g} \tag{34}$$

and displacement field can be updated by extrapolation

$$u_g(t+dt) = dt^2 [M_g^{-1}(f_g(t) - K_g u_g(t))] + 2u_g(t) - u_g(t-dt)$$
(35)

### SEM implementation: workflow



#### SEM: force vector and elementary matrix

```
# Initialization
       fe = zeros((ne, N+1))
       s = zeros((ne, N+1))
       # Point source
       s[int(ne/2), 1] = 1
       # Force vector
       for k in range (ne) :
           for i in range(N+1):
               fe[k,i] = s[k,i] * w[i] * J
# Elemental Mass matrix
# stored as a vector since it's diagonal
Me = zeros(N+1)
for i in range(0,N+1):
   Me[i] = rho * w[i] * J
# [...]
# Elemental Stiffness Matrix
Ke = zeros([N+1,N+1])
for i in range(0,N+1):
    for j in range(0, N+1):
        for k in range(0, N+1):
            Ke[i,j] = Ke[i,j] + mu*w[k]*Ji*l1d[i,k]
                       *l1d[j.k]
```

# [...]

#### SEM: Global stiffness matrix and time extrapolation

```
# Global Stiffness Matrix
    K = zeros([nq, nq])
     # Values except at element boundaries
     for k in range(1,ne+1):
         i0 = (k-1) * N+1
         j0 = i0
         for i in range(-1,N):
             for j in range(-1,N):
                 K[i0+i,j0+j] = Ke[i+1,j+1]
    # Values at element boundaries
     for k in range(2,ne+1):
         i0 = (k-1) * N
         i0 = i0
         K[i0, j0] = Ke[0, 0] + Ke[N, N]
# [...]
# Time extrapolation
for it in range(nt):
# Extrapolation
   unew = dt**2 * Minv @ (f - K @ u) + 2 * u - uold
   uold, u = u, unew
# [...]
```

#### SEM example: homogeneous medium

Wave simulation on a domain [0, 10] km, with  $N_e = 250$ ,  $V_s = 2.5$  km/s,  $\rho = 2000 kg/m^3$ , and N = 2 - 8 (in practical simuations, N = 4),  $\epsilon = 0.8$ ,  $T_{dom} = 0.15$  s. STF is the first-order derivative of Gaussian function:  $s(t) = -2a(t - t_0)e^{-(t - t_0)^2/a^2}$ , with  $a = 4/T_{dom}$ .



#### SEM example: heterogeneous model

Parameters similarly as the homogeneous example, except  $T_{dom} = 1.2$  s, N = 2, 4.

- · How to benchmark results?
- · Convergence with order (grid spacing/dt vs num. cost)?



## SEM in 2D and 3D (References)

- 2D SEM: Komatitsch and Vilotte (1998)
- 3D SEM + Newmark scheme: Komatitsch et al. (1999), and Komatitsch and Tromp (1999)
- 3D SEM for fluid-solid media + anisotropy: Komatitsch et al. (2000a,b).
- 3D SEM for global wave simulations: Chaljub et al. (2003), Komatitsch and Tromp (2002a,b)
- Regular grid 3D SEM: Fichtner and Igel (2008), Fichtner (2009), and Fichtner (2010).
- Axisem (2D) for global wave propagation: Nissen-Meyer et al. (2007) and Nissen-Meyer et al. (2014), Instaseis by van Driel et al. (2015b)
- SEM to triangular/tetrehedral meshes (solve the global linear system): Mercerat et al. (2006), May et al. (2016)

#### Next step

Further methods should be able to

- based on tetrahedral (or arbitrarily shaped) elements that are easier to adapt to complex geometries
- can better handle discontinuities in the solution field or the geophysical parameters

resulting in finite-volume and the discontinuous Galerkin methods.

