AWP-ODC: Numerics

Gautam Wilkins

I. Notation

1) ∂_t is a derivative in time
2) $\partial_x, \partial_y, \partial_z$ are spatial derivatives
3) ∇ always refers to derivatives in space
4) Δt is the discrete time step size and h is the spatial discretization size (it is the same in all three dimensions)
5) ρ is density, λ and μ are the Lamé coefficients (all three vary in space)
6) v is the velocity vector and σ is the 3×3 symmetric stress tensor
7) I is the 3×3 identity matrix

The individual elements of v and σ are denoted as:

$$v = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} \quad \sigma = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \end{pmatrix}$$

II. Governing Equations

$$\begin{aligned} \partial_t v &= \frac{1}{\rho} \nabla \cdot \sigma \\ \partial_t \sigma &= \lambda(\nabla \cdot v) I + \mu(\nabla v + \nabla v^T) \end{aligned}$$

Expanded, these equations become:

$$\partial_t v = \frac{1}{\rho} \begin{pmatrix} \partial_x \sigma_{xx} + \partial_y \sigma_{xy} + \partial_z \sigma_{xz} \\ \partial_x \sigma_{xy} + \partial_y \sigma_{yy} + \partial_z \sigma_{yz} \\ \partial_x \sigma_{xz} + \partial_y \sigma_{yz} + \partial_z \sigma_{zz} \end{pmatrix}$$

$$\partial_t \sigma = \lambda(\partial_x v_x + \partial_y v_y + \partial_z v_z) I + \\ \mu \begin{pmatrix} 2\partial_x v_x & \partial_y v_x + \partial_x v_y & \partial_z v_x + \partial_x v_z \\ \partial_x v_y + \partial_y v_x & 2\partial_y v_y & \partial_z v_y + \partial_y v_z \\ \partial_x v_z + \partial_z v_x & \partial_y v_z + \partial_z v_y & 2\partial_z v_z \end{pmatrix}$$

So there are 3 scalar velocity equations and 6 scalar stress equations.

III. Finite Differences

A. Time

AWP-ODC uses staggered grids in both time and space. Time derivatives are approximated by:

$$\partial_t v(t) \approx \frac{v(t + \frac{\Delta t}{2}) - v(t - \frac{\Delta t}{2})}{\Delta t} \tag{1}$$

$$\partial_t \sigma \left(t + \frac{\Delta t}{2} \right) \approx \frac{\sigma(t + \Delta t) - \sigma(t)}{\Delta t} \tag{2}$$

Note that the staggering of the time grid means that we first update the velocity components then use the *updated* velocity components to update the stress components.

B. Space

Let Φ denote either a velocity or stress component. Then the approximation to the partial derivative with respect to x at grid node (i, j, k) is:

$$\partial_x \Phi_{i,j,k} \approx \frac{c_1 \left(\Phi_{i+\frac{1}{2},\ j,k} - \Phi_{i-\frac{1}{2},\ j,k} \right) + c_2 \left(\Phi_{i+\frac{3}{2},\ j,k} - \Phi_{i-\frac{3}{2},\ j,k} \right)}{h} \tag{3}$$

where $c_1 = 9/8$ and $c_2 = -1/24$. The finite differences for derivatives in other spatial dimensions are analogous.

IV. Velocity Stencils

A. Velocity x: 13 point

To update v_x at grid node (i, j, k), applying finite difference stencils (1) and (3) to the governing equation gives:

$$v_x[i, j, k] = v_x[i, j, k] + \frac{\Delta t}{h\rho} D$$

where D is:

$$\begin{gathered} c_1(\sigma_{xx}[i, j, k] - \sigma_{xx}[i-1, j, k]) + \\ c_2(\sigma_{xx}[i+1, j, k] - \sigma_{xx}[i-2, j, k]) \\ +c_1(\sigma_{xy}[i, j, k] - \sigma_{xy}[i, j-1, k]) + \\ c_2(\sigma_{xy}[i, j+1, k] - \sigma_{xy}[i, j-2, k]) \\ +c_1(\sigma_{xz}[i, j, k] - \sigma_{xz}[i, j, k-1]) + \\ c_2(\sigma_{xz}[i, j, k+1] - \sigma_{xz}[i, j, k-2]) \end{gathered}$$

Since ρ varies in space, it is set to the following average:

$$\frac{\rho[i, j, k] + \rho[i, j-1, k] + \rho[i, j, k-1] + \rho[i, j-1, k-1]}{4}$$

B. Velocity y: 13 point

To update v_y at grid node (i, j, k), applying finite difference stencils (1) and (3) to the governing equation gives:

$$v_y[i, j, k] = v_y[i, j, k] + \frac{\Delta t}{h\rho} D$$

where D is:

$$\begin{gathered} c_1(\sigma_{xy}[i+1, j, k] - \sigma_{xy}[i, j, k]) + \\ c_2(\sigma_{xy}[i+2, j, k] - \sigma_{xy}[i-1, j, k]) \\ +c_1(\sigma_{yy}[i, j+1, k] - \sigma_{yy}[i, j, k]) + \\ c_2(\sigma_{yy}[i, j+2, k] - \sigma_{yy}[i, j-1, k]) \\ +c_1(\sigma_{yz}[i, j, k] - \sigma_{yz}[i, j, k-1]) + \\ c_2(\sigma_{yz}[i, j, k+1] - \sigma_{yz}[i, j, k-2]) \end{gathered}$$

Since ρ varies in space, it is set to the following average:

$$\frac{\rho[i, j, k] + \rho[i+1, j, k] + \rho[i, j, k-1] + \rho[i+1, j, k-1]}{4}$$

C. Velocity z: 13 point

To update v_z at grid node (i, j, k), applying finite difference stencils (1) and (3) to the governing equation gives:

$$v_z[i,j,k] = v_z[i,j,k] + \frac{\Delta t}{h\rho} D$$

where D is:

$$\begin{gathered} c_1(\sigma_{xz}[i+1,j,k] - \sigma_{xz}[i,j,k]) + \\ c_2(\sigma_{xz}[i+2,j,k] - \sigma_{xz}[i-1,j,k]) \\ +c_1(\sigma_{yz}[i,j,k] - \sigma_{yz}[i,j-1,k]) + \\ c_2(\sigma_{yz}[i,j+1,k] - \sigma_{yz}[i,j-2,k]) \\ +c_1(\sigma_{zz}[i,j,k+1] - \sigma_{zz}[i,j,k]) + \\ c_2(\sigma_{zz}[i,j,k+2] - \sigma_{zz}[i,j,k-1]) \end{gathered}$$

Since ρ varies in space, it is set to the following average:

$$\frac{\rho[i,j,k] + \rho[i+1,j,k] + \rho[i,j-1,k] + \rho[i+1,j-1,k]}{4}$$

V. Stress Stencils

A. Stress xx: 13 point

To update σ_{xx} at grid node (i, j, k), applying finite difference stencils (2) and (3) to the governing equation gives:

$$\sigma_{xx}[i,j,k] = \sigma_{xx}[i,j,k] + \frac{\Delta t}{h}\left[(2\mu+\lambda)D_x + \lambda D_y + \lambda D_z\right]$$

where

$$\begin{aligned} D_x &= c_1(v_x[i+1,j,k] - v_x[i,j,k]) + \\ &\quad c_2(v_x[i+2,j,k] - v_x[i-1,j,k]) \\ D_y &= c_1(v_y[i,j,k] - v_y[i,j-1,k]) + \\ &\quad c_2(v_y[i,j+1,k] - v_y[i,j-2,k]) \\ D_z &= c_1(v_z[i,j,k] - v_z[i,j,k-1]) + \\ &\quad c_2(v_z[i,j,k+1] - v_z[i,j,k-2]) \end{aligned}$$

Since λ and μ vary in space, they are set to the following averages:

$$\begin{aligned} \lambda = 8(&\lambda[i,j,k] + \lambda[i+1,j,k] + \\ &\lambda[i,j-1,k] + \lambda[i+1,j-1,k] + ... \\ &\lambda[i,j,k-1] + \lambda[i+1,j,k-1] + \\ &\lambda[i,j-1,k-1] + \lambda[i+1,j-1,k-1])^{-1} \end{aligned}$$

$$\begin{aligned} \mu = 8(&\mu[i,j,k] + \mu[i+1,j,k] + \\ &\mu[i,j-1,k] + \mu[i+1,j-1,k] + \\ &\mu[i,j,k-1] + \mu[i+1,j,k-1] + \\ &\mu[i,j-1,k-1] + \mu[i+1,j-1,k-1])^{-1} \end{aligned}$$

B. Stress yy: 13 point

To update σ_{yy} at grid node (i, j, k), applying finite difference stencils (2) and (3) to the governing equation gives:

$$\sigma_{yy}[i,j,k] = \sigma_{yy}[i,j,k] + \frac{\Delta t}{h}\left[\lambda D_x + (2\mu+\lambda)D_y + \lambda D_z\right]$$

where

$$\begin{aligned} D_x &= c_1(v_x[i+1,j,k] - v_x[i,j,k]) + \\ &\quad c_2(v_x[i+2,j,k] - v_x[i-1,j,k]) \\ D_y &= c_1(v_y[i,j,k] - v_y[i,j-1,k]) + \\ &\quad c_2(v_y[i,j+1,k] - v_y[i,j-2,k]) \\ D_z &= c_1(v_z[i,j,k] - v_z[i,j,k-1]) + \\ &\quad c_2(v_z[i,j,k+1] - v_z[i,j,k-2]) \end{aligned}$$

Since λ and μ vary in space, they are set to the following averages:

$$\begin{aligned} \lambda = 8(&\lambda[i,j,k] + \lambda[i+1,j,k] + \\ &\lambda[i,j-1,k] + \lambda[i+1,j-1,k] + ... \\ &\lambda[i,j,k-1] + \lambda[i+1,j,k-1] + \\ &\lambda[i,j-1,k-1] + \lambda[i+1,j-1,k-1])^{-1} \end{aligned}$$

$$\begin{aligned} \mu = 8(&\mu[i,j,k] + \mu[i+1,j,k] + \\ &\mu[i,j-1,k] + \mu[i+1,j-1,k] + \\ &\mu[i,j,k-1] + \mu[i+1,j,k-1] + \\ &\mu[i,j-1,k-1] + \mu[i+1,j-1,k-1])^{-1} \end{aligned}$$

C. Stress zz: 13 point

To update σ_{zz} at grid node (i, j, k), applying finite difference stencils (2) and (3) to the governing equation gives:

$$\sigma_{zz}[i,j,k] = \sigma_{zz}[i,j,k] + \frac{\Delta t}{h}\left[\lambda D_x + \lambda D_y + (2\mu+\lambda)D_z\right]$$

where

$$\begin{aligned} D_x &= c_1(v_x[i+1,j,k] - v_x[i,j,k]) + \\ &\quad c_2(v_x[i+2,j,k] - v_x[i-1,j,k]) \\ D_y &= c_1(v_y[i,j,k] - v_y[i,j-1,k]) + \\ &\quad c_2(v_y[i,j+1,k] - v_y[i,j-2,k]) \\ D_z &= c_1(v_z[i,j,k] - v_z[i,j,k-1]) + \\ &\quad c_2(v_z[i,j,k+1] - v_z[i,j,k-2]) \end{aligned}$$

Since λ and μ vary in space, they are set to the following averages:

$$\begin{aligned} \lambda = 8(&\lambda[i,j,k] + \lambda[i+1,j,k] + \\ &\lambda[i,j-1,k] + \lambda[i+1,j-1,k] + ... \\ &\lambda[i,j,k-1] + \lambda[i+1,j,k-1] + \\ &\lambda[i,j-1,k-1] + \lambda[i+1,j-1,k-1])^{-1} \end{aligned}$$

$$\begin{aligned} \mu = 8(&\mu[i,j,k] + \mu[i+1,j,k] + \\ &\mu[i,j-1,k] + \mu[i+1,j-1,k] + \\ &\mu[i,j,k-1] + \mu[i+1,j,k-1] + \\ &\mu[i,j-1,k-1] + \mu[i+1,j-1,k-1])^{-1} \end{aligned}$$

D. Stress xy: 9 Point

To update σ_{xy} at grid node (i, j, k), applying finite difference stencils (2) and (3) to the governing equation gives:

$$\sigma_{xy}[i,j,k] = \sigma_{xy}[i,j,k] + \frac{\mu \Delta t}{h}[D_x y + D_y x]$$

where

$$\begin{aligned} D_x y = c_1(v_x[i,j+1,k] - v_x[i,j,k]) + \\ c_2(v_x[i,j+2,k] - v_x[i,j-1,k]) \\ D_y x = c_1(v_y[i,j,k] - v_y[i-1,j,k]) + \\ c_2(v_y[i+1,j,k] - v_y[i-2,j,k]) \end{aligned}$$

Since μ varies in space, it is set to:

$$2(\mu[i,j,k] + \mu[i,j,k-1])^{-1}$$

E. Stress xz: 9 Point

To update σ_{xz} at grid node (i, j, k), applying finite difference stencils (2) and (3) to the governing equation gives:

$$\sigma_{xz}[i,j,k] = \sigma_{xz}[i,j,k] + \frac{\mu \Delta t}{h}[D_x z + D_z x]$$

where

$$\begin{aligned} D_x z = c_1(v_x[i,j,k+1] - v_x[i,j,k]) + \\ c_2(v_x[i,j,k+2] - v_x[i,j,k-1]) \\ D_z x = c_1(v_z[i,j,k] - v_z[i-1,j,k]) + \\ c_2(v_z[i+1,j,k] - v_z[i-2,j,k]) \end{aligned}$$

Since μ varies in space, it is set to:

$$2(\mu[i,j,k] + \mu[i,j-1,k])^{-1}$$

F. Stress yz: 9 Point

To update σ_{yz} at grid node (i, j, k), applying finite difference stencils (2) and (3) to the governing equation gives:

$$\sigma_{yz}[i,j,k] = \sigma_{yz}[i,j,k] + \frac{\mu \Delta t}{h}[D_y z + D_z y]$$

where

$$\begin{aligned} D_y z = c_1(v_y[i,j,k+1] - v_y[i,j,k]) + \\ c_2(v_y[i,j,k+2] - v_y[i,j,k-1]) \\ D_z y = c_1(v_z[i,j+1,k] - v_z[i,j,k]) + \\ c_2(v_z[i,j+2,k] - v_z[i,j-1,k]) \end{aligned}$$

Since μ varies in space, it is set to:

$$2(\mu[i,j,k] + \mu[i+1,j,k])^{-1}$$

VI. Absorbing Boundary Conditions

Truncating the 3D domain leads to undesired reflections at the boundary. To deal with this AWP implements a set of sponge layers along the boundaries in all 3 dimensions, with the exception of the free surface boundary at the top. The default is for the 20 nodes nearest the boundaries in all dimensions to be sponge layers.

Three coefficient vectors are stored: crj_x, crj_y, and crj_z. If we are using 20 sponge layers then the values of:

$$\begin{matrix} crj_x[0:19] & crj_x[end-19:end] \\ crj_y[0:19] & crj_y[end-19:end] \\ crj_z[0:19] & crj_z[end-19:end] \end{matrix}$$

are all less than 1.0.

$$crj_x[20:end{-}20] \quad crj_y[20:end{-}20] \quad \text{and } crj_z[20:end{-}20]$$

would all be 1.0 since they are not sponge layer nodes. At the end of the sponge layer boundary, homogeneous Dirichlet boundaries are used (i.e. all stress and velocity components are set to 0 for the finite differences).

A. Velocity: Sponge Layers

To incorporate the sponge layer coefficients for the velocity computation for v_x at node (i, j, k), we first update a velocity term v_x using the finite difference stencil from before. We then perform the scaling:

$$v_x[i,j,k] = v_x[i,j,k] * crj_x[i] * crj_y[j] * crj_z[k]$$

Note that if node (i, j, k) is in the interior then all of the crj coefficients will be 1.0 and will not change the value.

B. Stress: Sponge Layers

To incorporate the sponge layer coefficients for the stress computation for σ_{xx} at node (i, j, k), we first update the stress term σ_{xx} using the finite difference stencil from before. We then perform the scaling:

$$\sigma_{xx}[i,j,k] = \sigma_{xx}[i,j,k] * crj_x[i] * crj_y[j] * crj_z[k]$$

Note that if node (i, j, k) is in the interior then all of the crj coefficients will be 1.0 and will not change the value.

VII. Free Surface Boundary

The nodes with highest z value in the computational grid correspond to the free surface, and we implement the traction-free boundary condition at the free surface. This means that at the free surface

$$\sigma_{xz} = \sigma_{yz} = \sigma_{zz} = 0.$$

A. Free Surface Velocity

While we do not need to apply a finite difference stencil to compute stress at the free surface, we still need to apply a fourth order finite difference stencil to compute stress at the nodes directly below the free surface, which means that we need velocity values at one point above the free surface. We use the difference formulas:

$$
\begin{aligned}
v_x[i,j,end+1] &= v_x[i,j,end] - \\
&\quad (v_z[i,j,end] - v_z[i-1,j,end]) \\
v_y[i,j,end+1] &= v_y[i,j,end] - \\
&\quad (v_z[i,j+1,end] - v_z[i,j,end]) \\
v_z[i,j,end+1] &= v_z[i,j,end] - \\
&\quad \frac{\mu[i,j,end]^{-1}}{2\mu[i,j,end]^{-1} + \lambda[i,j,end]^{-1}} * \\
&\quad ((D_x - v_x[i,j,end+1]) + \\
&\quad (v_x[i+1,j,end] - v_x[i,j,end]) + \\
&\quad (v_y[i,j,z+1] - D_y) + \\
&\quad (v_y[i,j,end] - v_y[i,j-1,end]))
\end{aligned}
$$

where the node (i,j,end) is a point on the free surface with arbitrary x and y position, and D_x and D_y are given by the formulas:

$$
\begin{aligned}
D_x &= v_x[i+1,j,end] - (v_z[i+1,j,end] - v_z[i,j,end]) \\
D_y &= v_y[i,j-1,end] - (v_z[i,j,end] - v_z[i,j-1,end])
\end{aligned}
$$

B. Free Surface Stress

We need to apply fourth order finite difference stencils to compute velocity components up to and including the points at the free surface, which means that we need stress components at two points above the free surface. Since the stress at the free surface must be zero, we use the following formulas:

$$
\begin{aligned}
\sigma_{zz}[i,j,end+1] &= -\sigma_{zz}[i,j,end] \\
\sigma_{zz}[i,j,end+2] &= -\sigma_{zz}[i,j,end-1] \\
\sigma_{xz}[i,j,end] &= 0 \\
\sigma_{xz}[i,j,end+1] &= -\sigma_{xz}[i,j,end-1] \\
\sigma_{xz}[i,j,end+2] &= -\sigma_{xz}[i,j,end-2] \\
\sigma_{yz}[i,j,end] &= 0 \\
\sigma_{yz}[i,j,end+1] &= -\sigma_{yz}[i,j,end-1] \\
\sigma_{yz}[i,j,end+2] &= -\sigma_{yz}[i,j,end-2]
\end{aligned}
$$

VIII. Anelastic Attenuation

Seismic waves experience anelastic attenuation as they travel through the earth, and this attenuation can be quantified using quality factors for S waves and P waves, referred to as Q_S and Q_P, respectively. The AWP-ODC code has two stress kernel options. The first just uses the standard equations from Section V that do not incorporate anelastic attenuation. The second kernel uses a coarse-grained implementation of memory variables in order to account for the attenuation.

A. Anelastic Stress Equations

Each of the six stress components also stores a memory variable. The memory variables are referred to as ξ_{xx}, ξ_{yy}, ξ_{zz}, ξ_{xy}, ξ_{xz}, ξ_{yz}. The stress equations then become:

$$\partial_t \sigma = \lambda(\nabla \cdot v)I + \mu(\nabla v + \nabla v^T - \partial_t) - \xi \tag{4}$$

where ξ is the matrix of memory variables:

$$\begin{pmatrix} \xi_{xx} & \xi_{xy} & \xi_{xz} \\ \xi_{yx} & \xi_{yy} & \xi_{yz} \\ \xi_{zx} & \xi_{zy} & \xi_{zz} \end{pmatrix}$$

The memory variables themselves are governed by the following equation:

$$\tau \partial_t \xi_{ij} + \xi_{ij} = w\left[2\mu A_S \epsilon_{ij} + ((2\mu + \lambda)A_p - 2\mu A_S))\epsilon_{kk}\delta_{ij}\right] \tag{5}$$

The equation uses the Einstein summation convention, where repeated indices denote summation over all values. The value of τ is an integer between 1 and 8 and is different for each node (i,j,k). The value for w is a weight coefficient that is different for each node. The values A_S and A_P are coefficients that are computed from the quality factors Q_S and Q_P. δ_{ij} is the Kronecker delta function, and ϵ is the strain, which is approximated from the velocity using the equation:

$$\epsilon_{ij} = \frac{1}{2h}\left[\partial_i v_j + \partial_j v_i\right],$$

where the spatial derivatives on the right are computed using the finite difference stencil from Equation 3. Equation 5 is solved using an exponential integrator, which advances the solution in time using the equation:

$$\xi_{ij}^{t+1} = \tau_1 \xi_{ij}^t + \tau_2 F_{ij} \tag{6}$$

where F_{ij} is the right hand side of Equation 5, $\tau_1 = \exp[-\Delta t/\tau]$, $\tau_2 = \frac{1}{2}(1 - \exp[-\Delta t/\tau])$, ξ_{ij}^t is the value of the memory variable at time t, and ξ_{ij}^{t+1} is the value of the memory variable at time $t+1$.

Lastly, the updated stress equation is solved by first performing the same finite difference stencils in Section V, and then performing the update:

$$\sigma_{ij} = \sigma_{ij} + (\xi_{ij}^t + \xi_{ij}^{t+1})\Delta t,$$

where ξ_{ij}^t is the value of the memory variable before the update in Equation 6, and ξ_{ij}^{t+1} is the value of the memory variable after performing the update in Equation 6.