Kernel Optimizations in SORD earthquake dynamic rupture code

Geoffrey P. Ely
Leadership Computing Facility
Argonne National Laboratory

MiraCon
March 7, 2013

Acknowledgements

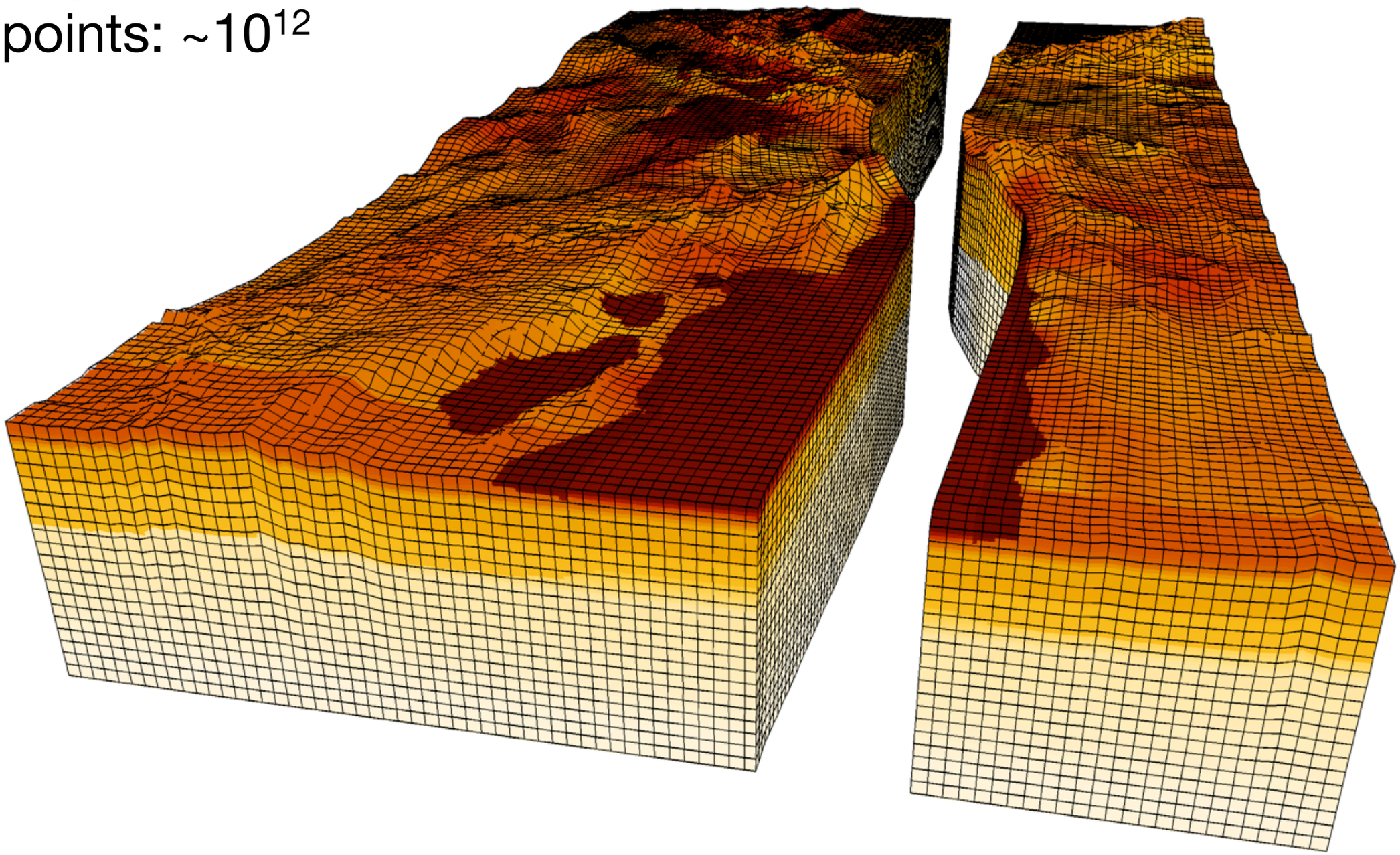
Tareq Malas
King Abdullah Univ. of Science & Technology

Vitali Morozov
Argonne Leadership Computing Facility

Karen Magerlein
IBM Watson Research Center

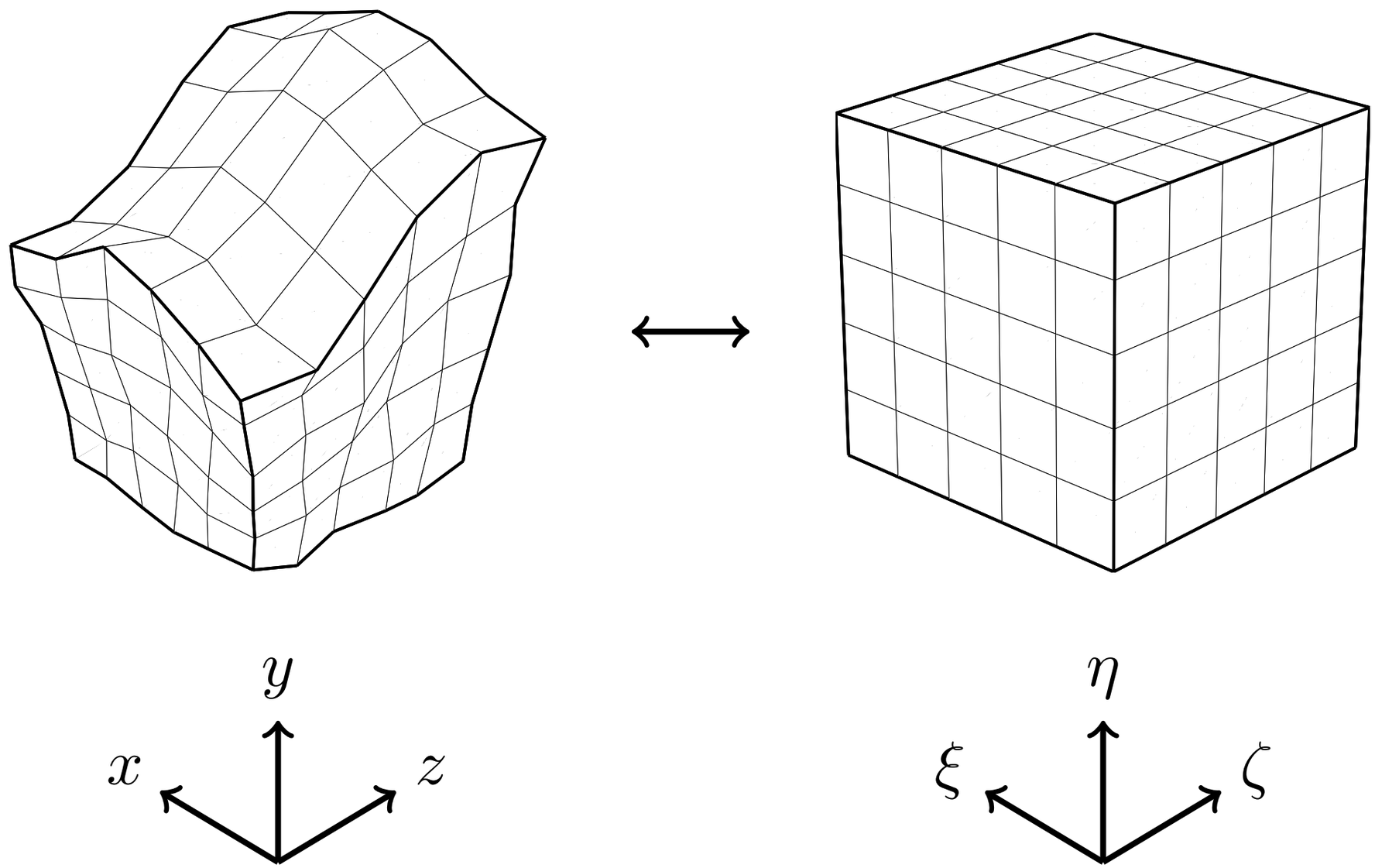
Simulation Scale

Outer length scale: ~500km
Inner length scale: ~50m
Mesh points: ~10^{12}



Spatial Derivatives

$$f(\boldsymbol{\xi}) = \sum_{\alpha,\beta,\gamma=0}^{n} N_{\alpha\beta\gamma}(\boldsymbol{\xi}) f_{\alpha\beta\gamma} \qquad \boldsymbol{\nabla} f = \frac{\partial f}{\partial \boldsymbol{\xi}} \cdot \mathbf{J}^{-1}$$



Equations of Motion

Viscoelastic Solid (Kelvin-Voigt Model)

$$\ddot{\mathbf{u}} = \frac{1}{\rho}\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}$$

$$\mathbf{g} = \boldsymbol{\nabla}(\mathbf{u} + \gamma\dot{\mathbf{u}})$$

$$\boldsymbol{\sigma} = \lambda\,\mathrm{trace}(\mathbf{g}) + \mu(\mathbf{g} + \mathbf{g}^{\mathrm{T}})$$

ρ	density
λ, μ	elastic moduli
γ	viscosity
$\mathbf{u}$	displacement
$\boldsymbol{\sigma}$	stress tensor

Free surface boundary condition

$$\boldsymbol{\tau} = \boldsymbol{\sigma}\cdot\hat{\mathbf{n}} = 0$$

```
! Kernel with 12 Streams
do l = 1, 400
do k = 1, 400
do j = 1, 400
    df(j,k,l) &
    = b(j,k,l,1) * f(j,  k,  l) &
    + b(j,k,l,2) * f(j+1,k,  l) &
    + b(j,k,l,3) * f(j,  k+1,l) &
    + b(j,k,l,4) * f(j+1,k+1,l) &
    + b(j,k,l,5) * f(j,  k,  l+1) &
    + b(j,k,l,6) * f(j+1,k,  l+1) &
    + b(j,k,l,7) * f(j,  k+1,l+1) &
    + b(j,k,l,8) * f(j+1,k+1,l+1)
end do
end do
end do
```

```
! Kernel with 5 streams
do l = 1, 400
do k = 1, 400
do j = 1, 400
    df(j,k,l) &
    = b(1,j,k,l) * f(j,  k,  l) &
    + b(2,j,k,l) * f(j+1,k,  l) &
    + b(3,j,k,l) * f(j,  k+1,l) &
    + b(4,j,k,l) * f(j+1,k+1,l) &
    + b(5,j,k,l) * f(j,  k,  l+1) &
    + b(6,j,k,l) * f(j+1,k,  l+1) &
    + b(7,j,k,l) * f(j,  k+1,l+1) &
    + b(8,j,k,l) * f(j+1,k+1,l+1)
end do
end do
end do
```

```
! Kernel with 6 streams
do l = 1, 400
do k = 1, 400
do j = 1, 400
    df(j,k,l) &
    = b(j,k,l,1) * f(j,  k,  l) &
    + b(j,k,l,2) * f(j+1,k,  l) &
    + b(j,k,l,3) * f(j,  k+1,l) &
    + b(j,k,l,4) * f(j+1,k+1,l)
end do
do j = 1, 400
    df(j,k,l) + df(j,k,l) &
    + b(j,k,l,5) * f(j,  k,  l+1) &
    + b(j,k,l,6) * f(j+1,k,  l+1) &
    + b(j,k,l,7) * f(j,  k+1,l+1) &
    + b(j,k,l,8) * f(j+1,k+1,l+1)
end do
end do
end do
```

How Did We Do?

- Prefetch and data locality optimizations: 1.6X speed-up
- Cache-tiling optimizations: 1.6X speed-up
- QPX Vectorization: Work in progress!

The End

Golcuk Mosque, Turkey, AP Photo by Enric Marti, 1999