EASI - A library for the easy setup of large scale earthquake simulations and other applications

Carsten Uphoff, Michael Bader

Technical University of Munich

What is EASI?

An extensible header-only library to evaluate

$$f : \mathbb{R}^m \to \mathbb{R}^n$$

via

- YAML configuration files,
- function composition and filters,
- a run-time JIT compiler [1],
- and a pArallel Server for Adaptive GeoInformation [2].

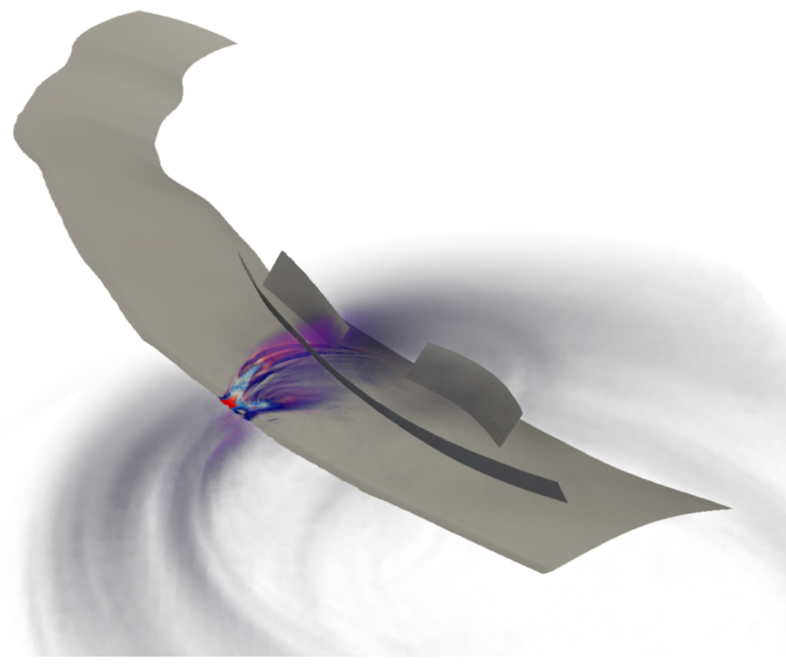
TPV26 scenario: User code

```fortran
! based on SCEC TPV26/27 test right-lateral strike-slip fault, z negative in depth
! 26 with elastic and 27 with viscoplastic material properties
b11 = 0.926793
b33 = 1.073206
b13 = -0.169029

VertexSide(1,:) =  (/ 1, 3, 2 /)   ! Local tet. vertices of tet. side I   !
VertexSide(2,:) =  (/ 1, 2, 4 /)   ! Local tet. vertices of tet. side II  !
VertexSide(3,:) =  (/ 1, 4, 3 /)   ! Local tet. vertices of tet. side III !
VertexSide(4,:) =  (/ 2, 3, 4 /)   ! Local tet. vertices of tet. side IV  !

! Loop over every mesh element
DO i = 1, MESH%Fault%nSide
    ! element ID
    iElem = MESH%Fault%Face(i,1,1)
    iSide = MESH%Fault%Face(i,2,1)
    ! constant background stress tensor and state variale
    EQN%IniBulk_xx(i,:)  =  EQN%Bulk_xx_0
    EQN%IniBulk_yy(i,:)  =  EQN%Bulk_yy_0
    EQN%IniBulk_zz(i,:)  =  EQN%Bulk_zz_0
    EQN%IniShearXY(i,:)  =  EQN%ShearXY_0
    EQN%IniShearYZ(i,:)  =  EQN%ShearYZ_0
    EQN%IniShearXZ(i,:)  =  EQN%ShearXZ_0
    EQN%IniStateVar(i,:) =  EQN%RS_sv0
    IF (EQN%GPwise.EQ.1) THEN
    ! Gauss node coordinate definition and stress assignment
    ! get vertices of complete tet
    IF (MESH%Fault%Face(i,1,1) == 0) THEN
        ! iElem is in the neighbor domain
        ! The neighbor element belongs to a different MPI domain
        iNeighbor           = MESH%Fault%Face(i,1,2)
        iLocalNeighborSide  = MESH%Fault%Face(i,2,2)
        iObject  = MESH%ELEM%BoundaryToObject(iLocalNeighborSide,iNeighbor)
        MPIIndex = MESH%ELEM%MPINumber(iLocalNeighborSide,iNeighbor)
        !
        xV(1:4) = BND%ObjMPI(iObject)%NeighborCoords(1,1:4,MPIIndex)
        yV(1:4) = BND%ObjMPI(iObject)%NeighborCoords(2,1:4,MPIIndex)
        zV(1:4) = BND%ObjMPI(iObject)%NeighborCoords(3,1:4,MPIIndex)
    ELSE
        xV(1:4) = MESH%VRTX%xyNode(1,MESH%ELEM%Vertex(1:4,iElem))
        yV(1:4) = MESH%VRTX%xyNode(2,MESH%ELEM%Vertex(1:4,iElem))
        zV(1:4) = MESH%VRTX%xyNode(3,MESH%ELEM%Vertex(1:4,iElem))
    END IF
    DO iBndGP = 1,DISC%Galerkin%nBndGP
            ! Transformation of boundary GP's into XYZ coordinate system
            chi  = MESH%ELEM%BndGP_Tri(1,iBndGP)
            tau  = MESH%ELEM%BndGP_Tri(2,iBndGP)
            CALL TrafoChiTau2XiEtaZeta(xi,eta,zeta,chi,tau,iSide,0)
            CALL TetraTrafoXiEtaZeta2XYZ(xGP,yGP,zGP,xi,eta,zeta,xV,yV,zV)
            !depth, negative in depth
            !average = zGP   ! Averaging not needed here
            Pf = 9800.0D0* abs(zGP) !fluid pressure, hydrostatic with water table at the surface
            IF (zGP.GE. -15000.0D0) THEN !depth less than 15000m
             omega = 1.0D0
            ELSEIF ((zGP.LT. -15000.0D0) .AND. (zGP .GE. -20000.0D0) ) THEN !depth between 15000 a
             omega = (20000.0D0-abs(zGP))/5000.0D0
            ELSE ! depth more than 20000m
             omega = 0.0D0
            END IF
           EQN%IniBulk_zz(i,iBndGP)  = -2670D0*9.8D0 * abs(zGP)
           EQN%IniBulk_xx(i,iBndGP)  = omega*(b11*(EQN%IniBulk_zz(i,iBndGP)+Pf)-Pf) &
                                        +(1-omega)*EQN%IniBulk_zz(i,iBndGP)
           EQN%IniBulk_yy(i,iBndGP)  = omega*(b33*(EQN%IniBulk_zz(i,iBndGP)+Pf)-Pf) &
                                        +(1-omega)*EQN%IniBulk_zz(i,iBndGP)
           EQN%IniShearXY(i,iBndGP)  = omega*(b13*(EQN%IniBulk_zz(i,iBndGP)+Pf))
           EQN%IniShearXZ(i,iBndGP)  = 0.0D0
           EQN%IniShearYZ(i,iBndGP)  = 0.0D0
           !add fluid pressure
           EQN%IniBulk_xx(i,iBndGP)  = EQN%IniBulk_xx(i,iBndGP)+Pf
           EQN%IniBulk_yy(i,iBndGP)  = EQN%IniBulk_yy(i,iBndGP)+Pf
           EQN%IniBulk_zz(i,iBndGP)  = EQN%IniBulk_zz(i,iBndGP)+Pf
           !depth dependent frictional cohesion, negative in seissol, in benchmark positive
          IF (zGP.GE.-5000.0D0) THEN
            DISC%DynRup%cohesion(iBndGP,i) = -0.4D6 - 0.00072D6*(5000D0-abs(zGP))
          ELSE
            DISC%DynRup%cohesion(iBndGP,i) = -0.4D6
          END IF
    END DO ! iBndGP
    ! element wise stress assignment
    ELSE
    ! (...)
  END IF !node or elementwise
END DO !    MESH%Fault%nSide
```

TPV26 scenario: EASI file

```yaml
!FunctionMap
map:
  z:    return z;
  Pf:   return 9800.0 * abs(z);
  s_zz: return -2670.0 * 9.8 * abs(z);
  omega: |
    if (z >= -15000.0) {
      return 1.0;
    }
    if (z <= -15000.0 && z >= -20000.0) {
      return (20000.0-abs(z))/5000.0;
    }
    return 0.0;
components: !FunctionMap
  map:
    s_xx: return Pf + omega*( 0.926793*(s_zz+Pf)-Pf) + (1-omega)*s_zz;
    s_yy: return Pf + omega*( 1.073206*(s_zz+Pf)-Pf) + (1-omega)*s_zz;
    s_zz: return Pf + s_zz;
    s_xy: return omega*(-0.169029*(s_zz+Pf));
    s_yz: return 0.0;
    s_xz: return 0.0;
```

Earthquake simulation



Simulation snapshot of the 2004 Sumatra earthquake [3].

How do you replace 5000 lines of "ugly" Fortran user code?

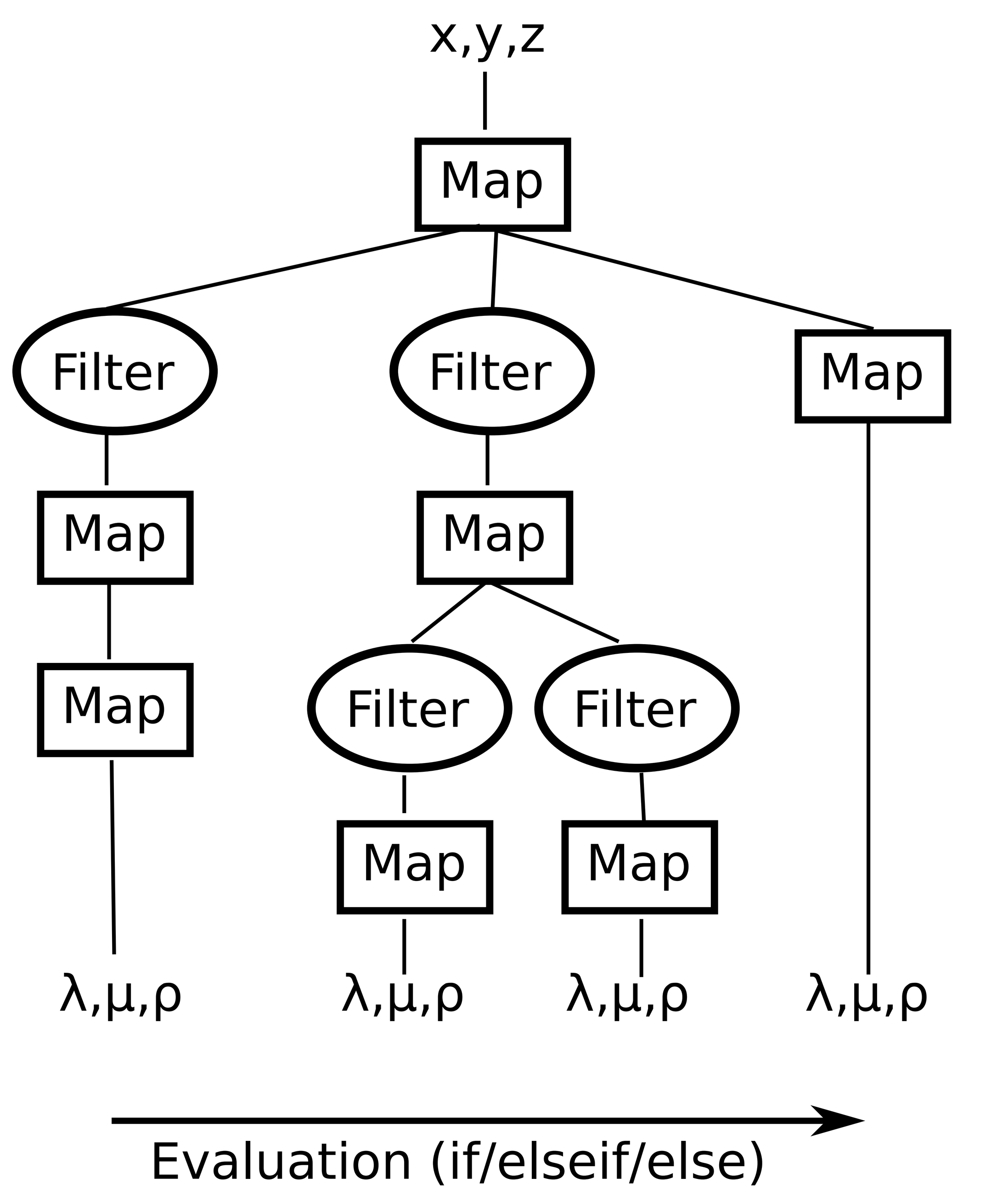
github.com/SeisSol/easi
uphoff@in.tum.de

[1] Manuel Fasching. "JIT compilation to realize flexible data access in simulation software". Master's thesis. Institut für Informatik, Technische Universität München, Mar. 2017.

[2] Sebastian Rettenberger et al. "ASAGI - A Parallel Server for Adaptive Geoinformation". In: *EASC '16 Proceedings of the Exascale Applications and Software Conference 2016*. ACM, Sept. 2016, 2:1–2:9.

[3] Carsten Uphoff et al. "Extreme Scale Multi-physics Simulations of the Tsunamigenic 2004 Sumatra Megathrust Earthquake". In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. SC '17. Denver, Colorado: ACM, 2017, 21:1–21:16.

Composition and filters



1. Reusable maps:
   - Constant map: $f_i(\boldsymbol{x}) = \boldsymbol{c}$
   - Affine map: $f_i(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{c}$
   - Polynomial map: ($m = 1$): $f_i(x) = \sum_{i=0}^{n} \boldsymbol{a}_i x^i$
   - JIT compilation map: $f_i(\boldsymbol{x}) =$ run-time compiled C-like function
   - ASAGI map: Trilinear interpolation from structured grid
2. Powerful through composition: $f = f_1 \circ \cdots \circ f_N$
3. Specified and composed in human readable configuration files.

Performance

Single-core tests on Intel Skylake Platinum 8174.
Test cases are available on github.com/SeisSol/easi.

Maximum time for 256^3 points $\in \mathbb{R}^m$:

Test case	n	t_{setup} [ms]	t_{point} [ns]	t_{total} [s]
0_constant	3	3.0	15	0.2
2_prem	3	5.2	95	1.6
3_layered_linear	3	1.8	52	0.9
33_layered_constant	3	1.7	52	0.9
5_function	3	1.8	153	2.6
26_function	11	2.9	203	3.4
f_120_sumatra	12	4.0	578	9.7
f_16_scec	14	3.2	257	4.3

Influence of number of points / evaluation: