3D modelling of non-linear visco-elasto-plastic crustal and lithospheric scale processes using LaMEM

Anton Popov, Boris Kaus, Adina Püsök, Tobias Baumann, Jianfeng Yang

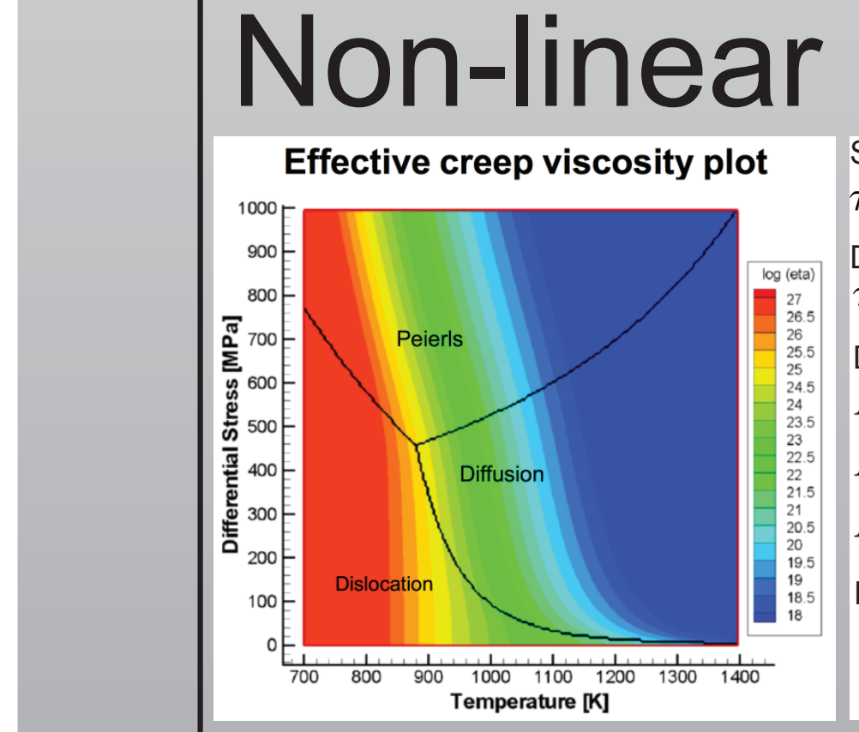
Johannes-Gutenberg University Mainz (Germany)

INTRODUCTION

LaMEM (Lithosphere and Mantle Evolution Model) is a 3D, open-source, thermo-mechanical numerical code to simulate crustal and lithospheric deformation. The code is written on top of PETSc and is based on a marker-and-cell approach combined with a staggered finite difference (FDSTAG) discretization in space, which is a stable and very efficient technique to solve the (nearly) incompressible Stokes equations that does not suffer from spurious pressure modes or artificial compressibility (a typical feature of low-order finite element techniques) and has the same accuracy as FE methods for typical geodynamic applications. Here we discuss some issues related to the accuracy of the method, convergence of non-linear solvers and show a few recent modelling applications.

Non-linear visco-elasto-plastic rheology

Effective creep viscosity plot



Stress update $\tau_{ij} = 2\eta^* \dot{\varepsilon}^*_{ij}$

Effective strain rate and invariant $\dot{\varepsilon}^*_{ij} = \dot{\varepsilon}_{ij} + \frac{\tau^*_{ij}}{2G\Delta t}$ $\dot{\varepsilon}^*_{II} = \left(\frac{1}{2}\dot{\varepsilon}^*_{ij}\dot{\varepsilon}^*_{ij}\right)^{1/2}$

Drucker-Prager $\tau_Y = \mu P + c$

Effective viscosity $\eta^* = \min\left[\left(\frac{1}{G\Delta t} + \frac{1}{\eta_l} + \frac{1}{\eta_n} + \frac{1}{\eta_p}\right)^{-1}, \frac{\tau_Y}{2\dot{\varepsilon}^*_{II}}\right]$

Diffusion, dislocation and Peierls constants

$$A_l = B_l \exp\left[-\frac{E_l + pV_l}{RT}\right],$$
$$A_n = B_n \exp\left[-\frac{E_n + pV_n}{RT}\right]$$
$$A_p = \frac{B_p}{(\gamma\tau_p)^s} \exp\left[-\frac{E_p + pV_p}{RT}(1-\gamma)^q\right]$$

Peierls effective exponent $s = \frac{E_p + PV_p}{RT}(1-\gamma)^{q-1} q\gamma$

Stress rotation terms $\tau^*_{ij} = \tau^n_{ij} + \Delta t\left(w_{ik}\tau^n_{kj} - \tau^n_{ik}w_{kj}\right)$

Deviatoric strain rate $\dot{\varepsilon}_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) - \frac{1}{3}\frac{\partial v_k}{\partial x_k}\delta_{ij}$

Spin tensor $\omega_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} - \frac{\partial v_j}{\partial x_i}\right)$

Effective creep viscosities

$$\eta_l = \tfrac{1}{2}(A_l)^{-1}$$
$$\eta_n = \tfrac{1}{2}(A_n)^{-\frac{1}{n}}(\dot{\varepsilon}^*_{II})^{\frac{1}{n}-1}$$
$$\eta_p = \tfrac{1}{2}(A_p)^{-\frac{1}{s}}(\dot{\varepsilon}^*_{II})^{\frac{1}{s}-1}$$

Finite element Jacobian derivation (matrix notation)

Stress and strain rate vectors

$$\dot{\varepsilon} = [\dot{\varepsilon}_{xx}\dot{\varepsilon}_{yy}\dot{\varepsilon}_{zz}\dot{\gamma}_{xy}\dot{\gamma}_{xz}\dot{\gamma}_{yz}]^T$$
$$\underline{\dot{\varepsilon}} = [\dot{\varepsilon}_{xx}\dot{\varepsilon}_{yy}\dot{\varepsilon}_{zz}\dot{\varepsilon}_{xy}\dot{\varepsilon}_{yx}\dot{\varepsilon}_{xz}\dot{\varepsilon}_{zx}\dot{\varepsilon}_{yz}\dot{\varepsilon}_{zy}]^T$$
$$\tau = [\tau_{xx}\tau_{yy}\tau_{zz}\tau_{xy}\tau_{xz}\tau_{yz}]^T$$
$$\underline{\tau} = [\tau_{xx}\tau_{yy}\tau_{zz}\tau_{xy}\tau_{yx}\tau_{xz}\tau_{zx}\tau_{yz}\tau_{zy}]^T$$

Projection matrix and projection

$$Q = \frac{1}{2}\begin{bmatrix} 2&0&0&0&0&0&0&0&0\\ 0&2&0&0&0&0&0&0&0\\ 0&0&2&0&0&0&0&0&0\\ 0&0&0&1&1&0&0&0&0\\ 0&0&0&0&0&1&1&0&0\\ 0&0&0&0&0&0&0&1&1 \end{bmatrix}$$

$\tau = Q\underline{\tau}, \quad \underline{\dot{\varepsilon}} = Q^T\dot{\varepsilon}$

Nonlinear residuals

$$r^U = \int_\Omega B^T(\tau - mP) - N_U^T \rho g \, d\Omega$$
$$r^P = -\int_\Omega N_P^T m^T \dot{\varepsilon} \, d\Omega$$

Jacobian iteration

$$\begin{bmatrix}\Delta U_k\\ \Delta P_k\end{bmatrix} = -\begin{bmatrix}J^{UU} & J^{UP}\\ J^{PU} & 0\end{bmatrix}^{-1}\begin{bmatrix}r^U_k\\ r^P_k\end{bmatrix}$$
$$\begin{bmatrix}U_{k+1}\\ P_{k+1}\end{bmatrix} = \begin{bmatrix}U_k\\ P_k\end{bmatrix} + \alpha\begin{bmatrix}\Delta U_k\\ \Delta P_k\end{bmatrix}$$

Jacobian blocks

$$J^{UU} = \int_\Omega B^T Q T Q^T B \, d\Omega$$
$$J^{UP} = -\int_\Omega B^T (m - \mu Q \underline{q}) N_P \, d\Omega$$
$$J^{PU} = -\int_\Omega N_P^T m^T B \, d\Omega$$

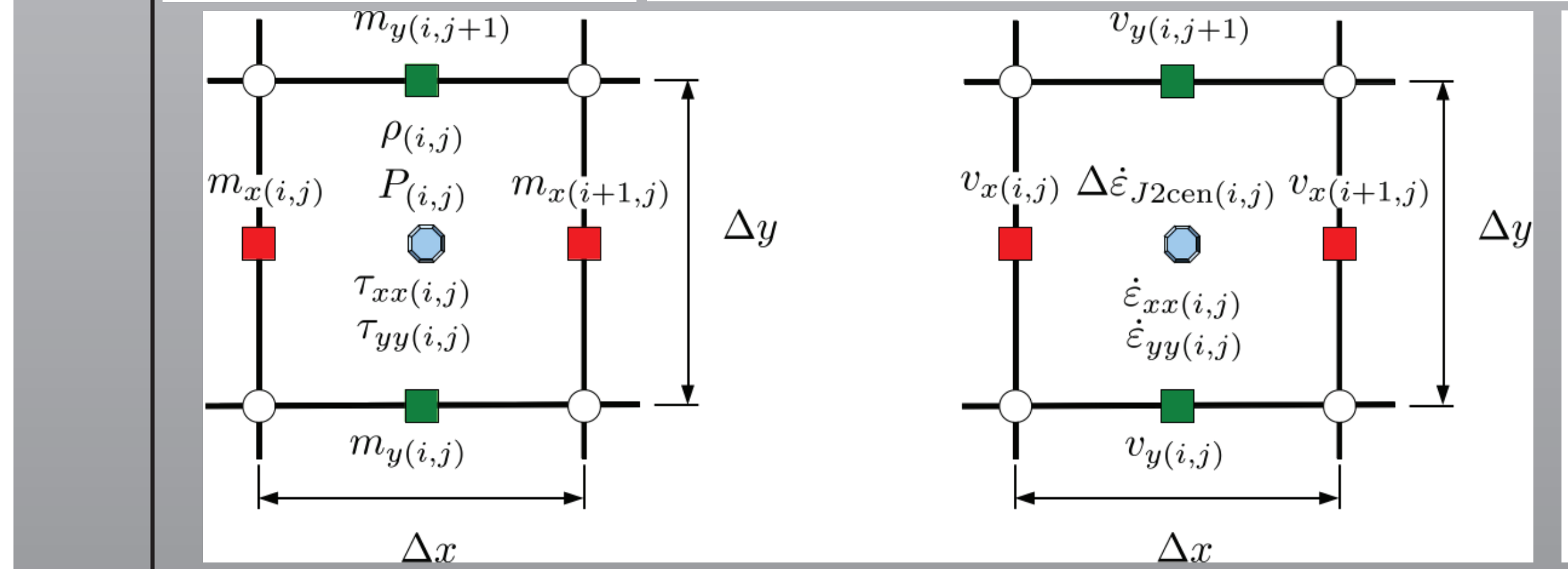
Tangent matrix $T = 2\eta^*(I + \beta \underline{q}\,\underline{q}^T)$

Visco-elastic nonlinear parameter $\beta = \frac{1}{2}\left(\frac{1}{n} - 1\right)\frac{\eta^*}{\eta_n} + \frac{1}{2}\left(\frac{1}{s} - 1\right)\frac{\eta^*}{\eta_p}$

Plastic nonlinear parameter $\beta = -\frac{1}{2}$

Normalized flow vector $\underline{q} = \frac{\underline{\dot{\varepsilon}}}{\dot{\varepsilon}^*_{II}}$

Pressure projection vector $m = [1\,1\,1\,0\,0\,0\,0\,0\,0]^T$


$m_{y(i,j+1)}$
$\rho_{(i,j)}$
$m_{x(i,j)}$ $P_{(i,j)}$ $m_{x(i+1,j)}$
$\tau_{xx(i,j)}$
$\tau_{yy(i,j)}$
$m_{y(i,j)}$
Δy
Δx
$v_{y(i,j+1)}$
$v_{x(i,j)}$ $\Delta\dot{\varepsilon}_{J2cen(i,j)}$ $v_{x(i+1,j)}$
$\dot{\varepsilon}_{xx(i,j)}$
$\dot{\varepsilon}_{yy(i,j)}$
$v_{y(i,j)}$
Δy
Δx

Staggered finite difference Jacobian derivation (single component example)

Momentum residual contributions

$$\Delta m_{x(i,j)} = +\frac{\tau_{xx(i,j)} - P_{(i,j)}}{\Delta x} + \frac{1}{2}\rho_{(i,j)} g_x$$
$$\Delta m_{x(i+1,j)} = -\frac{\tau_{xx(i,j)} - P_{(i,j)}}{\Delta x} + \frac{1}{2}\rho_{(i,j)} g_x$$

Residual derivatives (velocity)

$$\frac{\partial \Delta m_{x(i,j)}}{\partial v} = +\frac{1}{\Delta x}\frac{\partial \tau_{xx(i,j)}}{\partial v}$$
$$\frac{\partial \Delta m_{x(i+1,j)}}{\partial v} = -\frac{1}{\Delta x}\frac{\partial \tau_{xx(i,j)}}{\partial v}$$

Residual derivatives (pressure)

$$\frac{\partial \Delta m_{x(i,j)}}{\partial P} = +\frac{1}{\Delta x}\frac{\partial \tau_{xx(i,j)}}{\partial P} - \frac{1}{\Delta x}$$
$$\frac{\partial \Delta m_{x(i+1,j)}}{\partial P} = -\frac{1}{\Delta x}\frac{\partial \tau_{xx(i,j)}}{\partial P} + \frac{1}{\Delta x}$$

Stress derivatives

$$\frac{\partial \tau_{xx(i,j)}}{\partial v} = 2\frac{\partial \eta^*_{cen(i,j)}}{\partial v}\dot{\varepsilon}^*_{xx(i,j)} + 2\eta^*_{cen(i,j)}\frac{\partial \dot{\varepsilon}^*_{xx(i,j)}}{\partial v}$$
$$\frac{\partial \tau_{xx(i,j)}}{\partial P} = \mu\frac{\dot{\varepsilon}^*_{xx(i,j)}}{\dot{\varepsilon}^*_{II}}$$

Viscosity derivatives

$$\frac{\partial \eta^*}{\partial v} = \frac{\partial \eta^*}{\partial \dot{\varepsilon}_{II}}\frac{\partial \dot{\varepsilon}_{II}}{\partial \dot{\varepsilon}_{J2}}\frac{\partial \dot{\varepsilon}_{J2}}{\partial v} \qquad \frac{\partial \dot{\varepsilon}_{II}}{\partial \dot{\varepsilon}_{J2}} = \frac{1}{2}(\dot{\varepsilon}_{II})^{-1}$$
$$\frac{\partial \eta^*}{\partial \dot{\varepsilon}_{II}} = \left(\frac{\eta^*}{\eta_n}\right)^2\left(\frac{1}{n} - 1\right)\frac{\eta_n}{\dot{\varepsilon}^*_{II}} + \left(\frac{\eta^*}{\eta_p}\right)^2\left(\frac{1}{s} - 1\right)\frac{\eta_p}{\dot{\varepsilon}^*_{II}}$$
$$\frac{\partial \eta^*}{\partial \dot{\varepsilon}_{II}} = -\frac{1}{2}\frac{\eta^*}{\dot{\varepsilon}^*_{II}}$$

Strain rate derivatives

$$\dot{\varepsilon}_{xx(i,j)} = \frac{v_{x(i+1,j)} - v_{x(i,j)}}{\Delta x} \qquad \frac{\partial \dot{\varepsilon}_{yy(i,j)}}{\partial v_{y(i,j)}} = -\frac{1}{\Delta y} \qquad \frac{\partial \dot{\varepsilon}_{yy(i,j)}}{\partial v_{y(i,j+1)}} = \frac{1}{\Delta y}$$

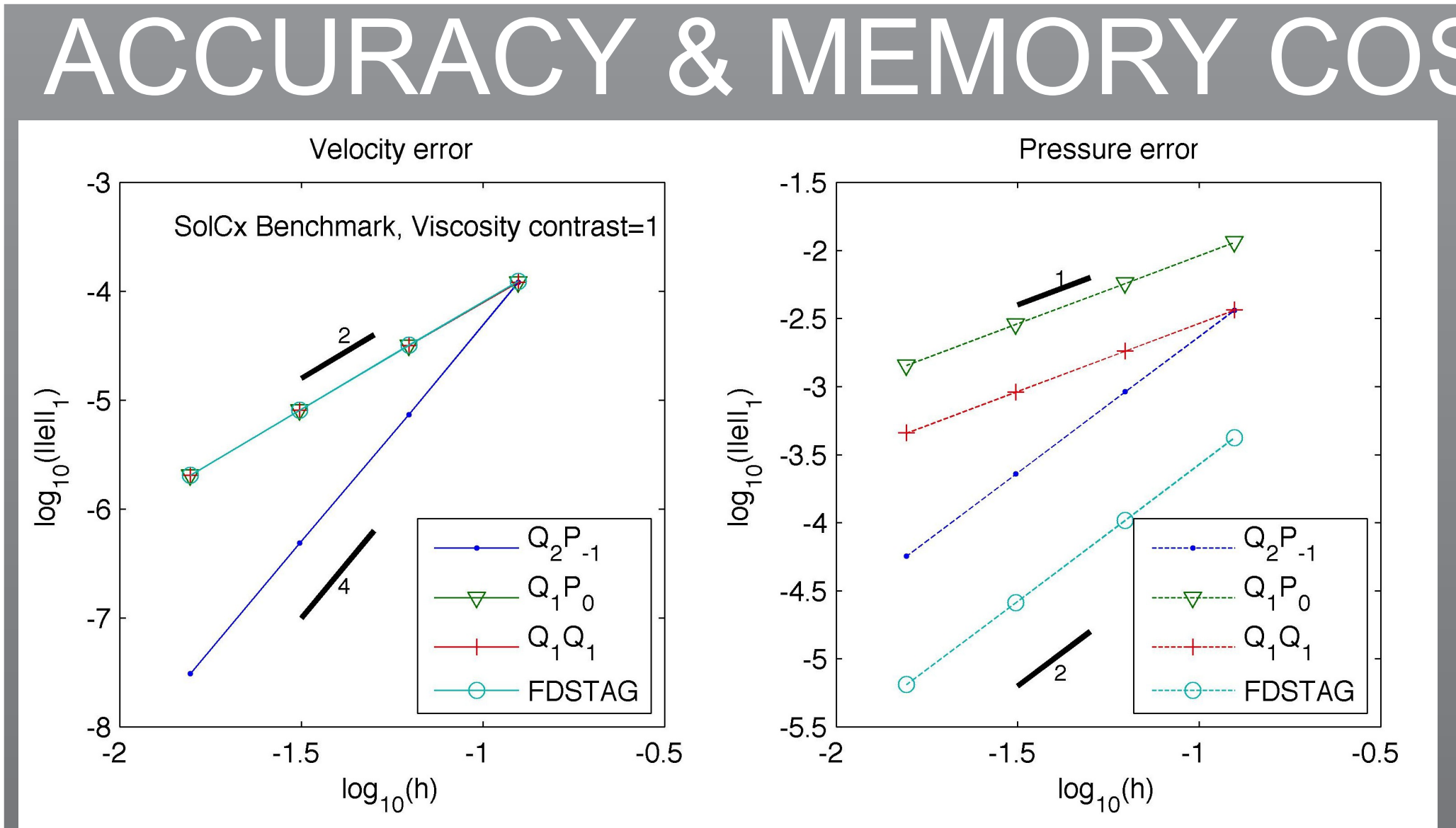
Second invariant contribution derivatives

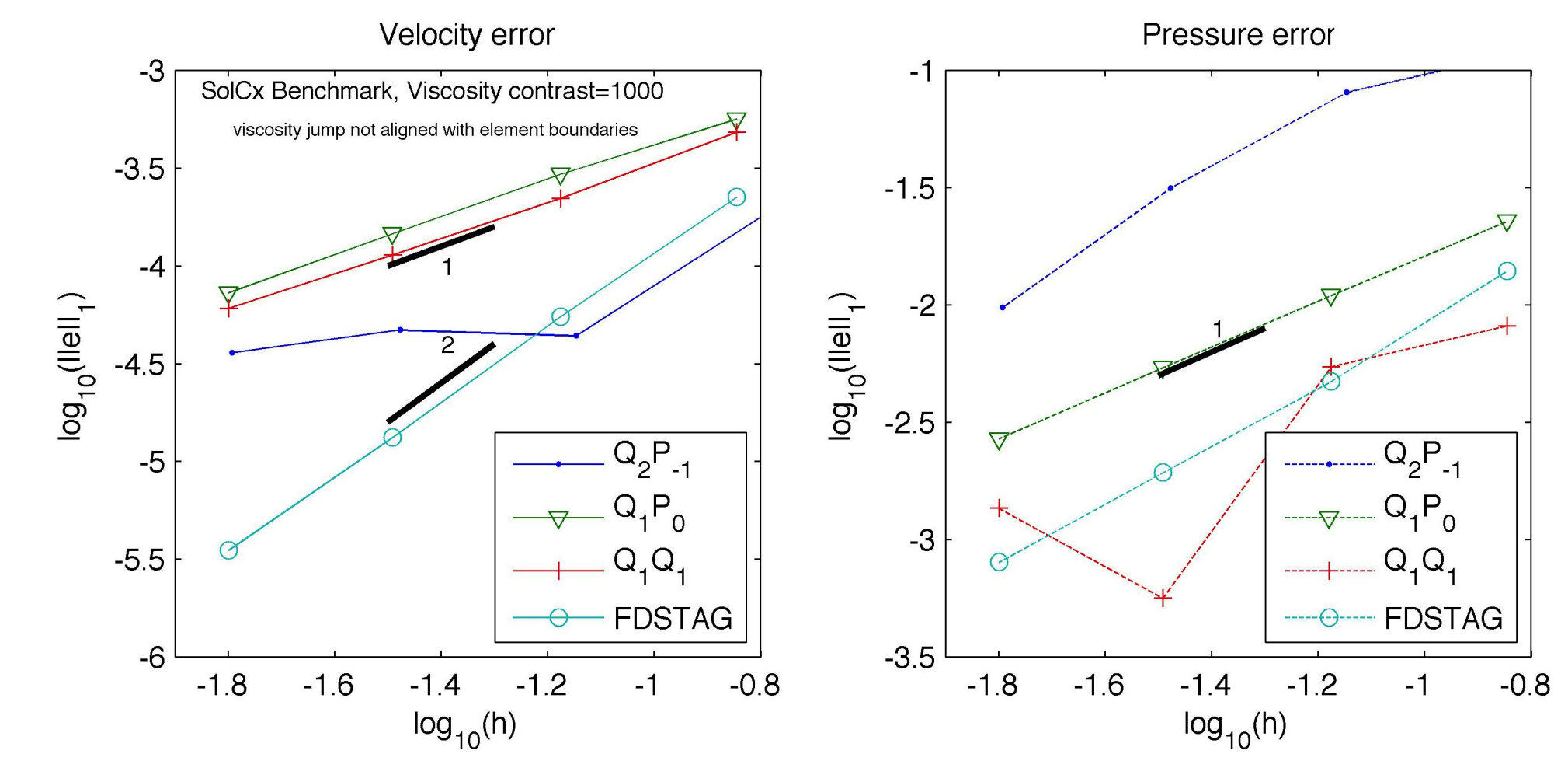
$$\Delta\dot{\varepsilon}_{J2cen(i,j)} = \frac{1}{2}\left(\dot{\varepsilon}^2_{xx(i,j)} + \dot{\varepsilon}^2_{yy(i,j)}\right)$$
$$\frac{\partial \Delta\dot{\varepsilon}_{J2cen(i,j)}}{\partial v_{x(i,j)}} = -\frac{1}{\Delta x}\dot{\varepsilon}_{xx(i,j)} \qquad \frac{\partial \Delta\dot{\varepsilon}_{J2cen(i,j)}}{\partial v_{x(i+1,j)}} = \frac{1}{\Delta x}\dot{\varepsilon}_{xx(i,j)}$$
$$\frac{\partial \Delta\dot{\varepsilon}_{J2cen(i,j)}}{\partial v_{y(i,j)}} = -\frac{1}{\Delta y}\dot{\varepsilon}_{yy(i,j)} \qquad \frac{\partial \Delta\dot{\varepsilon}_{J2cen(i,j)}}{\partial v_{y(i,j+1)}} = \frac{1}{\Delta y}\dot{\varepsilon}_{yy(i,j)}$$

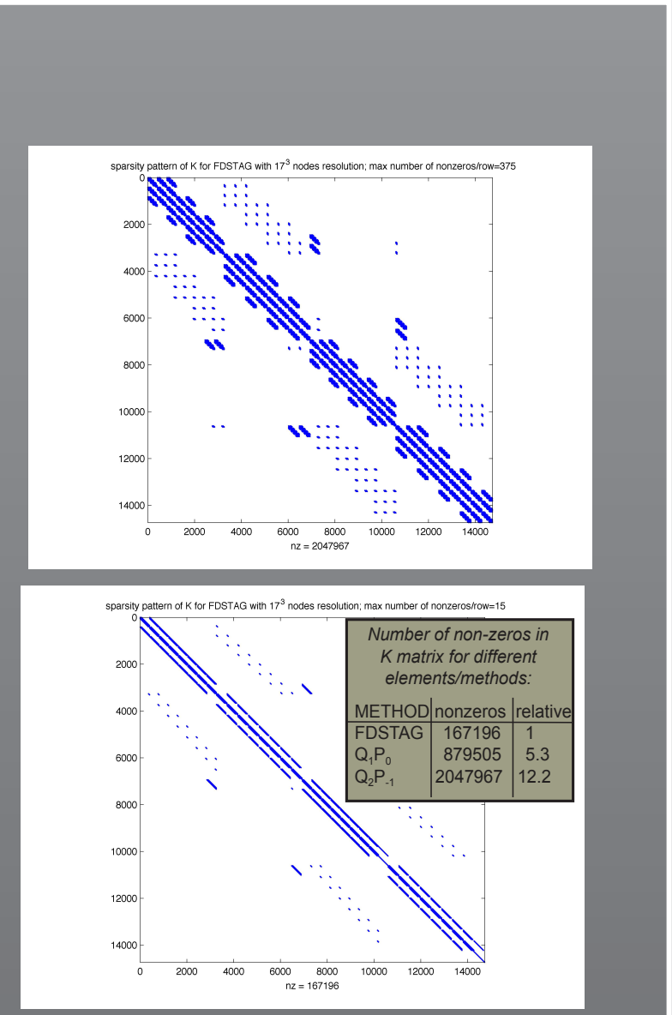
Contributions from neighbor points are assembled in the same way

The full, discretized, Jacobian increases the stencilwidth for FDSTAG. That is why we implement the action of the matrix-vector product in a matrix-free manner.

ACCURACY & MEMORY COSTS: FEM vs. FDSTAG



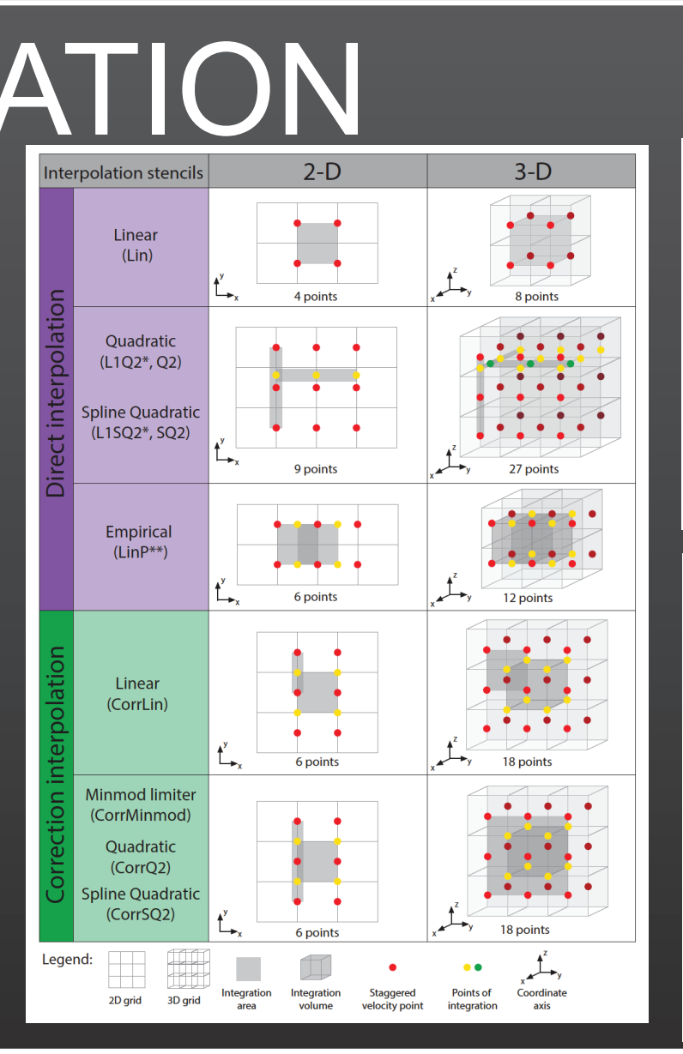


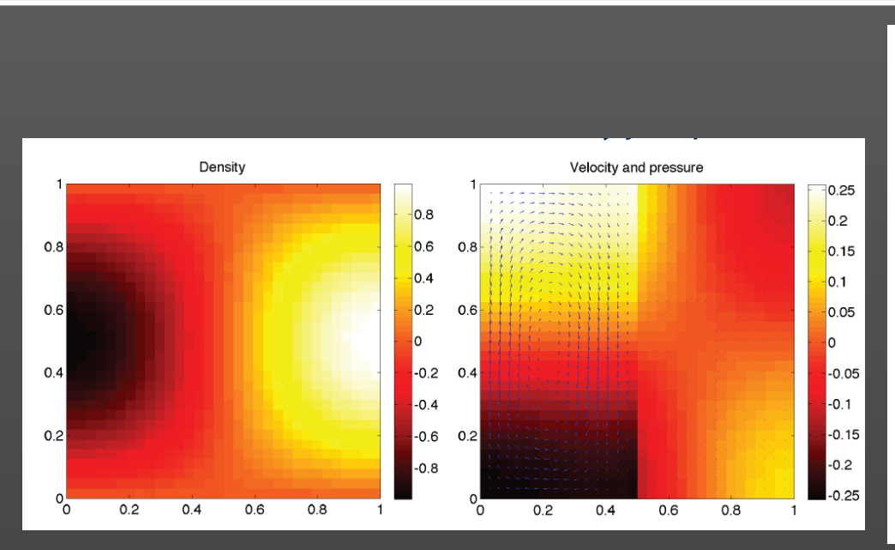


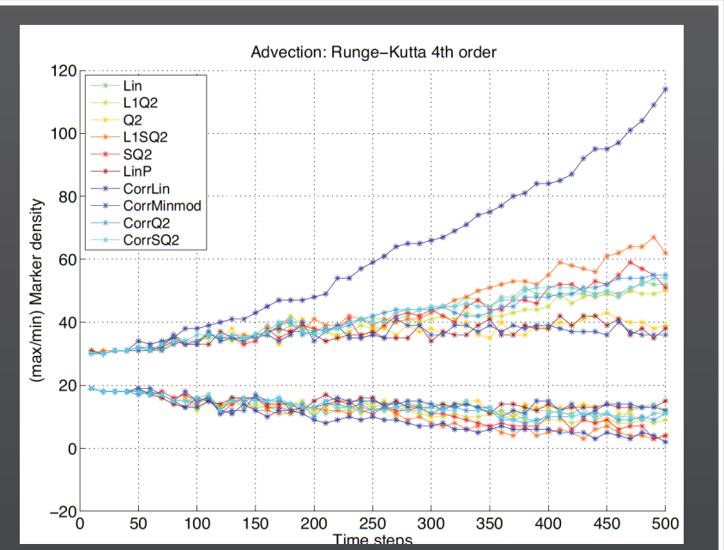
Some claim that higher-order finite element codes are more accurate than staggered finite difference discretizations. As the results show, this is only true if the jumps in viscosity are *exactly* alligned with the boundaries of the finite element. In typical evolving geodynamic setups (e.g., subduction), jumps do occur within an element either due to the marker distribution or due to non-linearities (e.g., plastic localization). Under these conditions, all methods are first order accurate [Deubelbeiss & Kaus, 2008; Thielmann et al. 2014]. Yet, low order finite elements (Q1P0) are unstable (wrong pressure), whereas the staggered finite difference is a stable discretization. The stabilized Q1Q1 element is compressible which is a problem in setups with a free surface. The Q2P1 element is stable but expensive (in 3D) as it requires up to 12 times more memory for the same number of nodes, which makes every matrix-vector product an order or magnitude slower. The staggered finite difference method is thus a cheap and stable low-order discretization method for typical geodynamic problems. A potential disadvantage is that the jacobian is more complicated to implement and increases the stencil width of the discretization. Yet, our results show it is typically not necessary to assemble the jacobian but that, instead, a matrix-free implementation is sufficient in many cases. Benchmarks show that a free surface can be succesfully represented by a sticky air layer [Crameri et al., GJI 2012], particularly if the 'air' is implemented by a stress-free internal boundary condition [Duretz et al.,GJI 2016].

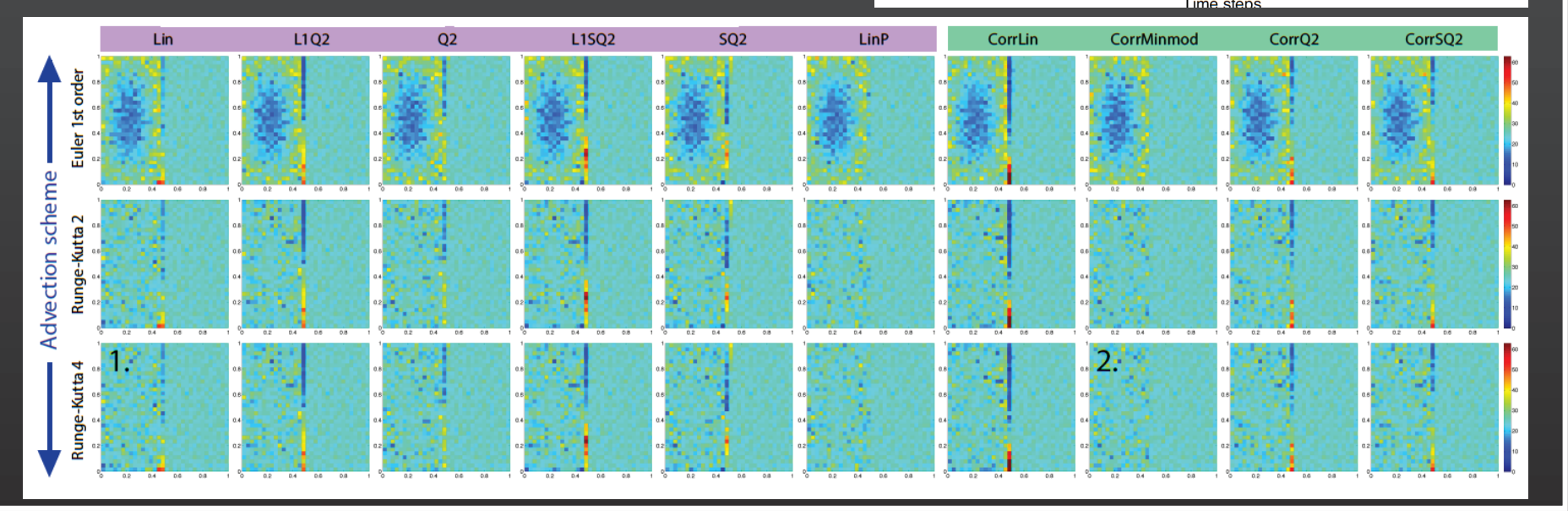
CONSERVATIVE MAC INTERPOLATION

The marker-and-cell approach involves frequent interpolations of velocities from nodal points to markers. This introduces interpolation errors as the velocity at markers is no longer guaranteed to be divergence-free which becomes noticable with time, as areas devoid of particles start appearing (which require particle injection/removal). A number of workarounds have been proposed recently, yet it is unclear which of those works best for a geodynamics application in particular combined with FDSTAG. We have adapted them for FDSTAG formulations and found the MINMOD method, combined with a 4th order Runga Kutta time integration scheme to be superior [Püsök et al., submitted].









Coupled vs. decoupled MG

$$P_c = \begin{pmatrix} K & G \\ D & -\frac{1}{\eta}I \end{pmatrix}$$

(COUPLED)
Galerkin MG applied to full matrix

$$P_u = \begin{pmatrix} K & G \\ & -\frac{1}{\eta}I \end{pmatrix}$$

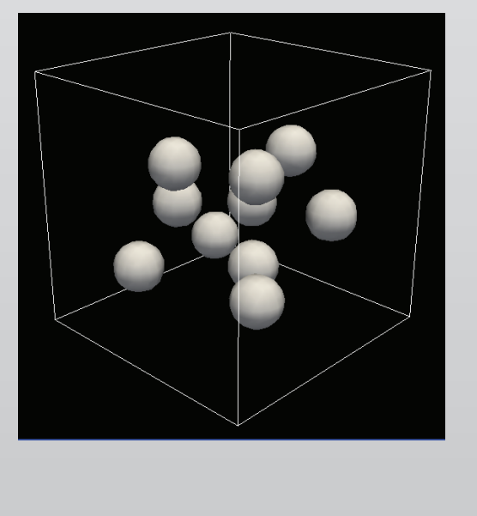
(UNCOUPLED)
Galerkin MG applied to K block only

$r_{coarse} = R\, r_{fine}$ - Restriction

$x_{fine} = P\, x_{coarse}$ - Prolongation

$A_{coarse} = R\, A_{fine}\, P$ - Coarsening

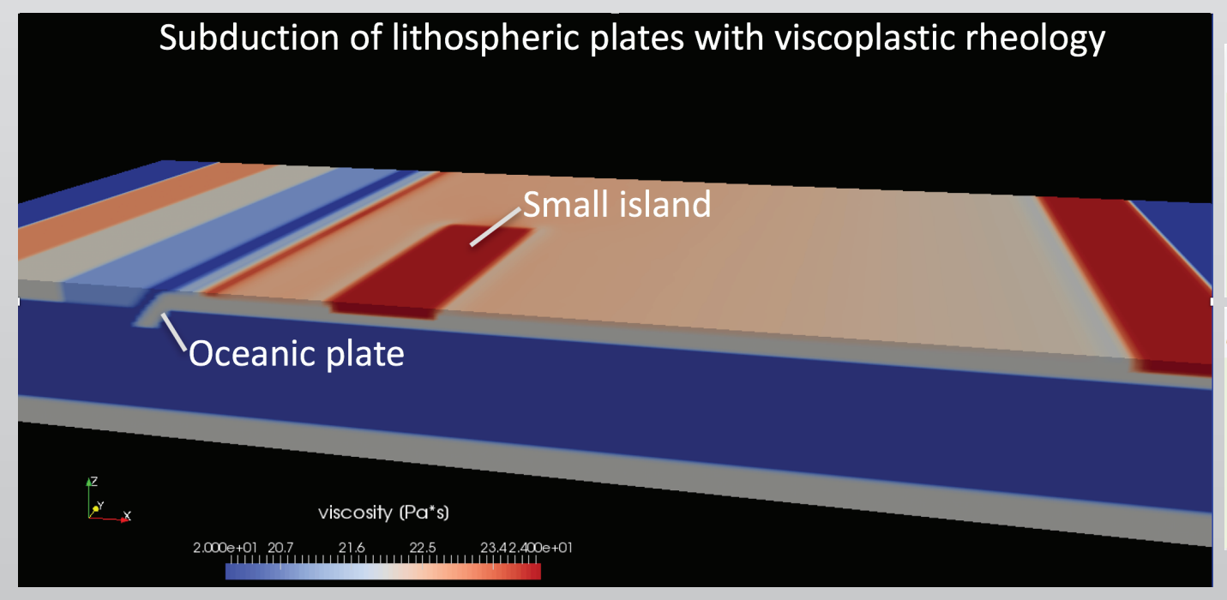
GALERKIN COARSENING



64³ nodes	Multiple spheres setup			
3 GMG levels	Coupled multigrid		Block MG	
Viscosity contrast	# outer KSP it	Time [s]	# outer KSP it	Total solve [s]
1	7	6.1	6	3.9
10	10	8	11	6
100	15	11.7	18	10.2
1000	36	27.6	45	23
10000	114	82	154	80.5
1.00E+05	378	266	585	297

Coupled: 3 GMG levels with FGMRES (rtol 1e-6), Jacobi(20,20) as smoothener; direct coarse grid, 4 cores
Block: FGMRES (rtol 1e-6) for full system with 1 V-cycle for the K-block, 3 GMG levels with Jacobi(20,20) as smoother and direct coarse grid

Subduction of lithospheric plates with viscoplastic rheology



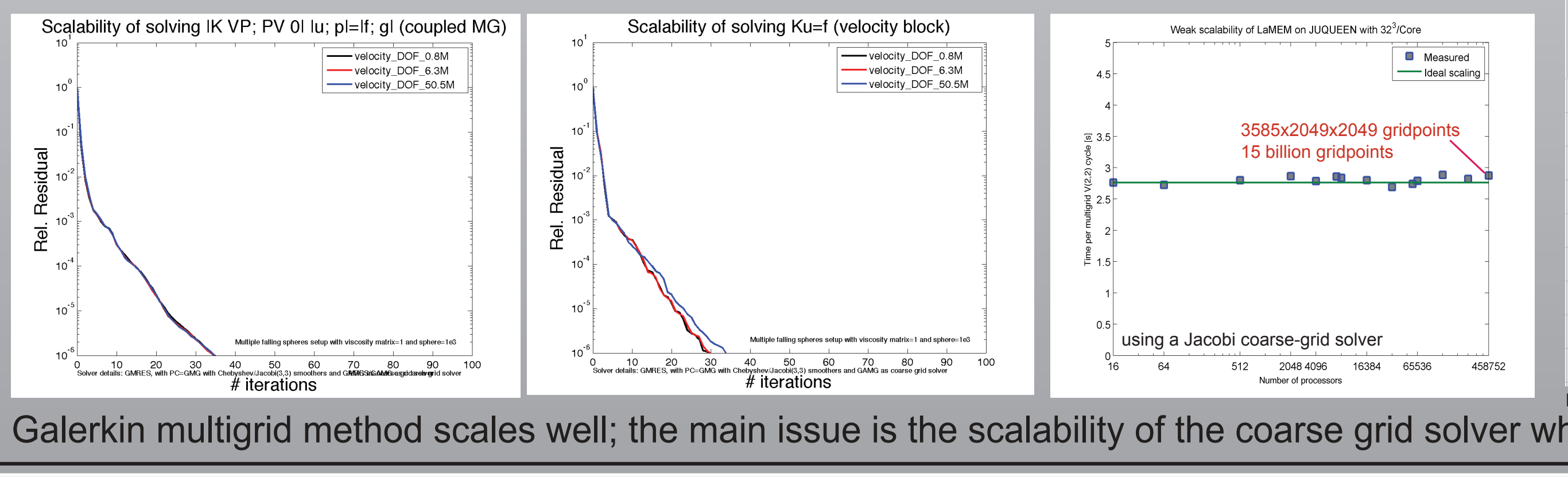
3 GMG levels	128x32x32		
Resolution	# SNES (nonlinear)	# KSP it (total)	Time per timestep [s]
Coupled MG	2	40	12
Block MG	2	100	24

4 GMG levels	256x64x64		
Resolution	# SNES (nonlinear)	# KSP it	Time [s]
Coupled MG	2	30	145
Block MG	5	250	759

Coupled: 3 GMG levels with FGMRES (rtol 1e-6), Jacobi(20,20) as smoothener; direct coarse grid, 4 cores
Block: FGMRES (rtol 1e-6) for full system with 1 V-cycle for the K-block, 3 GMG levels with Jacobi(20,20) as smoother and direct coarse grid

One can do Galerkin multigrid only for the velocity block, or for both velocity and pressure. Which one is best? For simple, linear viscous falling sphere setups there is little difference. For realistic viscoplastic setups, coupled wins.

PARALLEL SCALABILITY

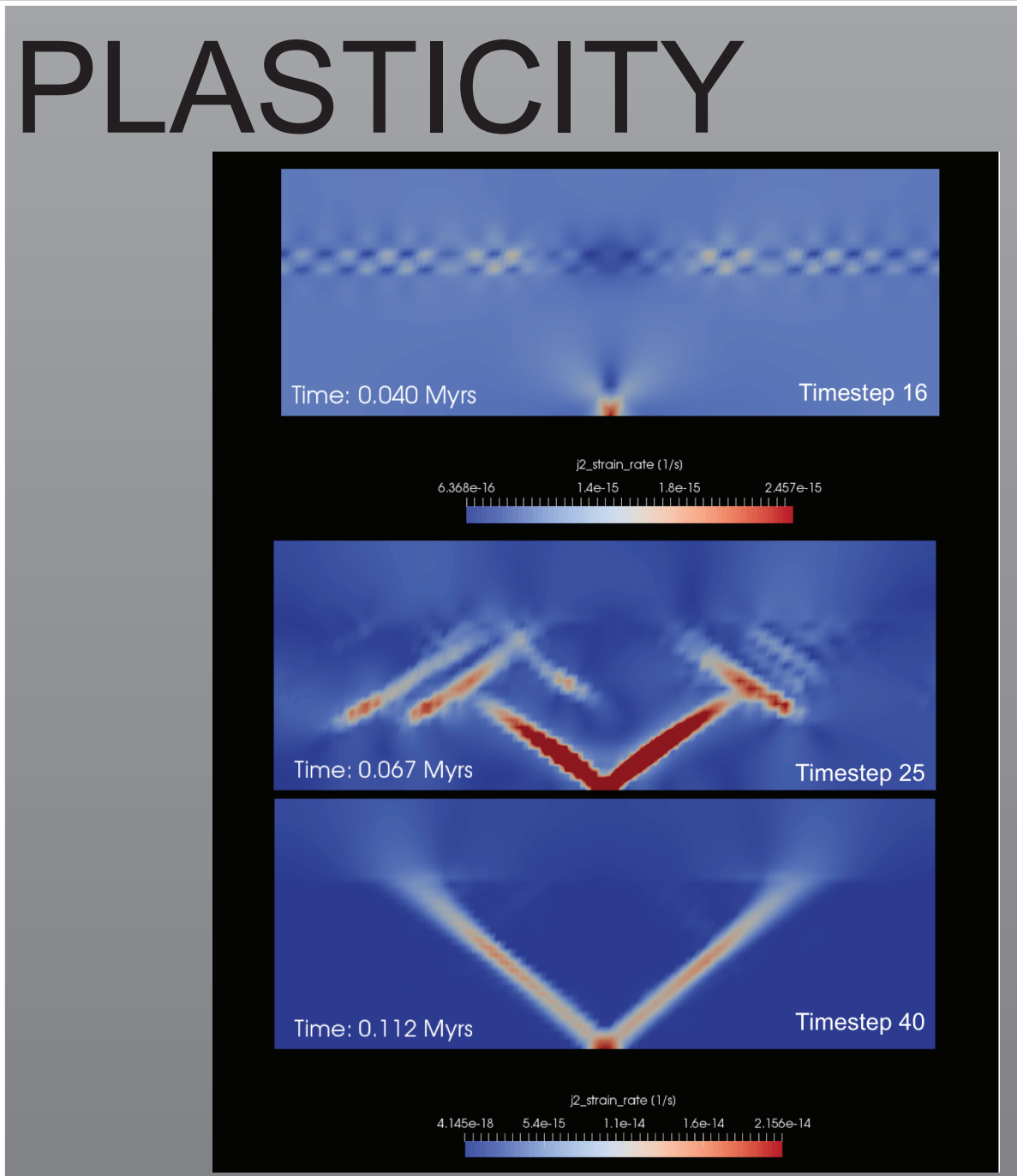


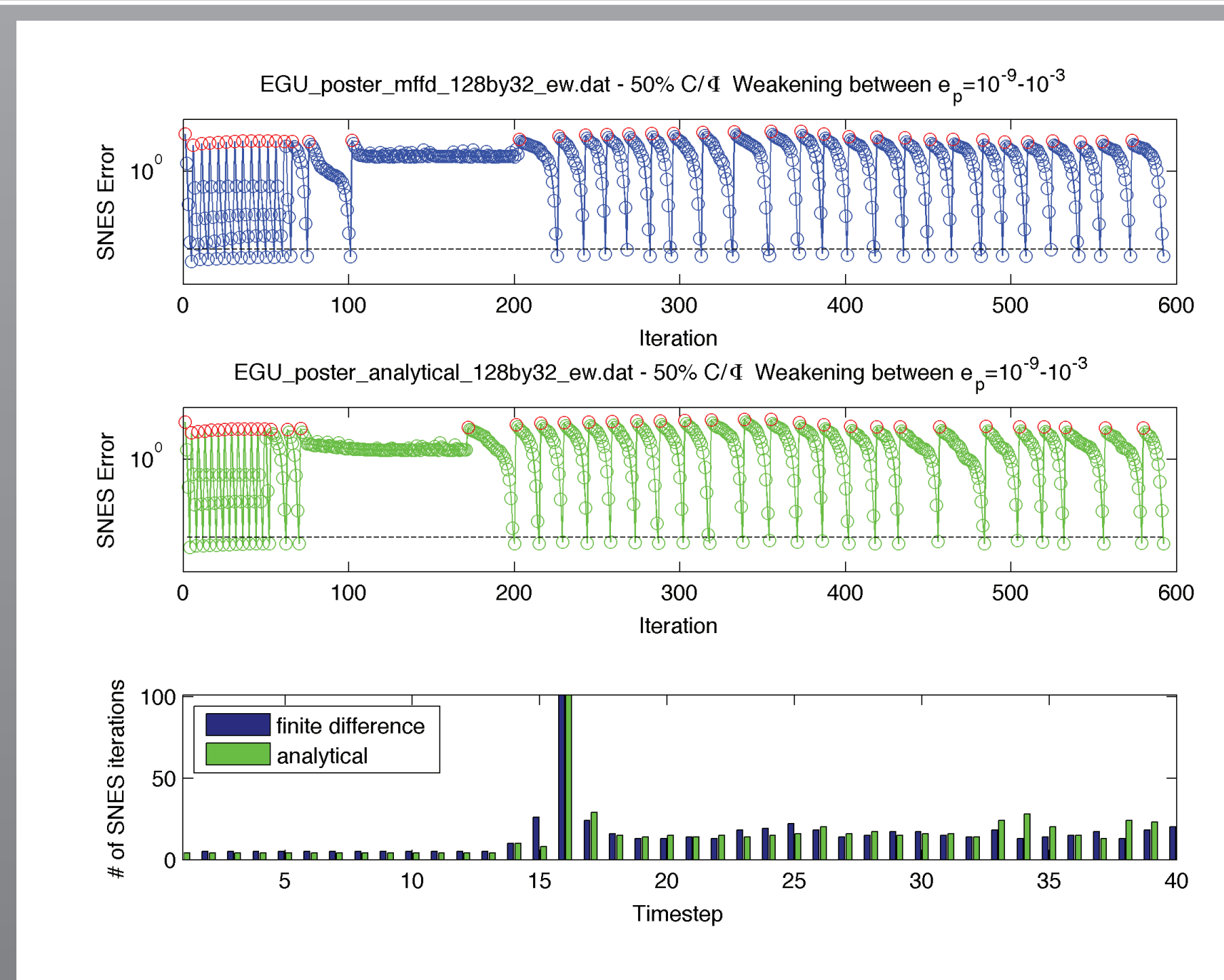
# Cores	Resolution	MG Levels	Coarse Level/core	Outer KSP it.	Main stage	Overall solver time [s]	Total time/tstep [s]	Coarse grid time [s]	MG time [w/out coarse solve]	MG time/it [s]
64	129x129x129	4	4x4x4	12	12.3	83.3	95.6	0.08	83.22	6.9
512	257x257x257	4	4x4x4	23	18.3	179.1	197.4	1.2	177.9	7.7
512	257x257x257	5	2x2x2	13	17.3	100.6	117.9	0.6	100	7.7
1024	513x257x257	5	2x2x2	18	17.6	140.8	158.4	3.1	137.7	7.7
2048	513x513x257	5	2x2x2	22	29.1	181.5	210.6	16.5	165	7.5
2048	513x513x257	6	1x1x1	14	27.4	116	143.4	9.7	106.3	7.6
4096	513x513x513	6	1x1x1	15	29.1	153.2	182.3	39.2	114	7.6
8192	1025x513x513	6	1x1x1	23	58.8	417	475.8	241	176	7.7
16384	1025x1025x513	6	1x1x1	26	71	1385	1456	1201	184	7.1

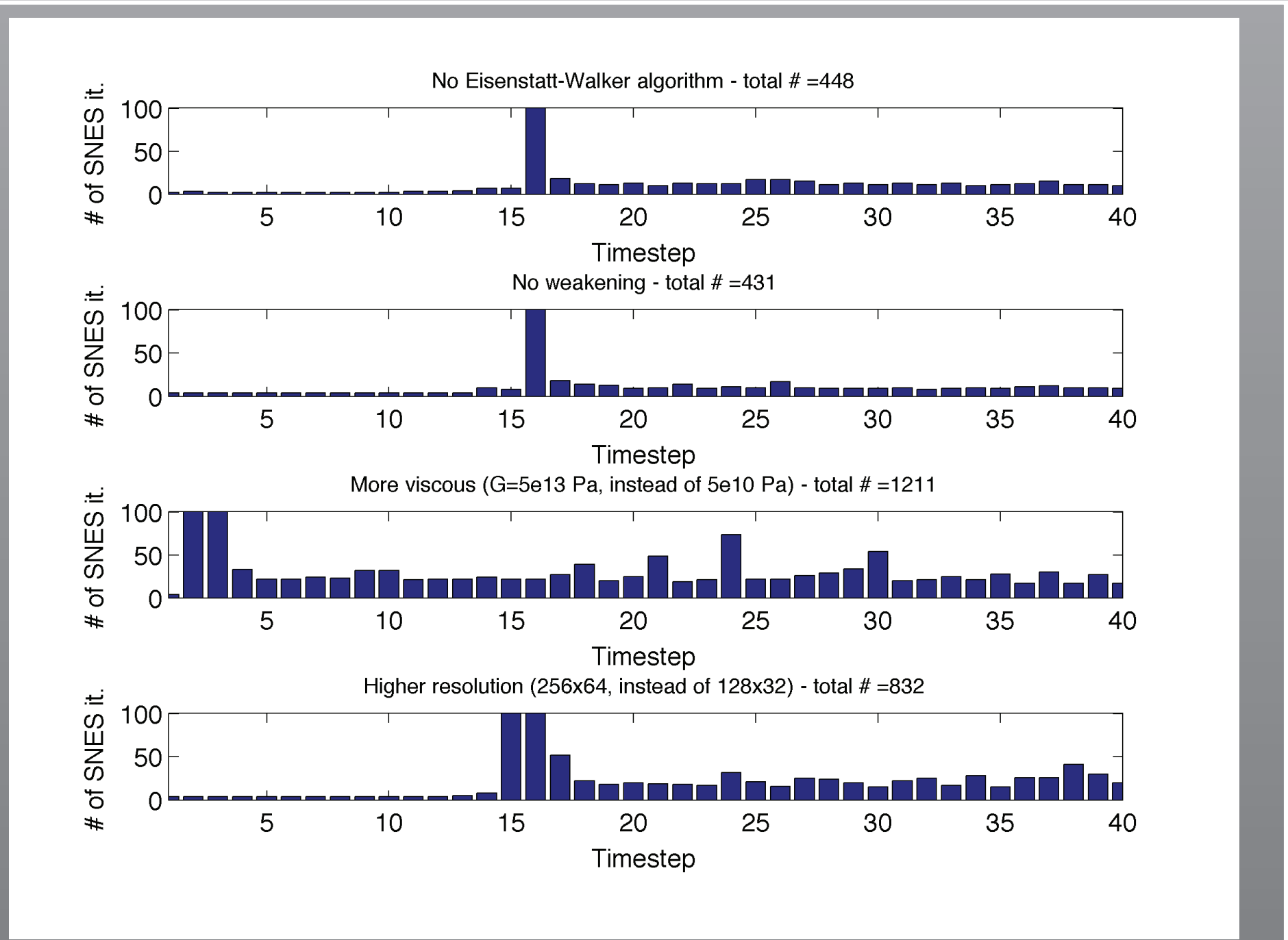
Isoviscous FB test with coupled MG, & SOR with 20 smoothening steps and relaxation factor 0.5 is used at every level with GAMG (algebraic MG) at the coarse level

Galerkin multigrid method scales well; the main issue is the scalability of the coarse grid solver which currently limits production runs to ~8191 cores (things slow down after that)

PLASTICITY







Viscoelastoplastic compressional setup with friction angle 30, Cohesion 50 MPa, viscosity 1e24 Pas, G=5e10 Pa, sticky air, weak inclusion (using direct inner solver). A typical feature is that non-convergence occurs during a few timesteps (when the pattern forms or changes), but that it does converge during subsequent timesteps. There is little difference between FD and the analytical matrix-free jacobian. Convergence is worse if the setup is more viscoplastic (higher G), and if the resolution increases.

ONGOING WORK

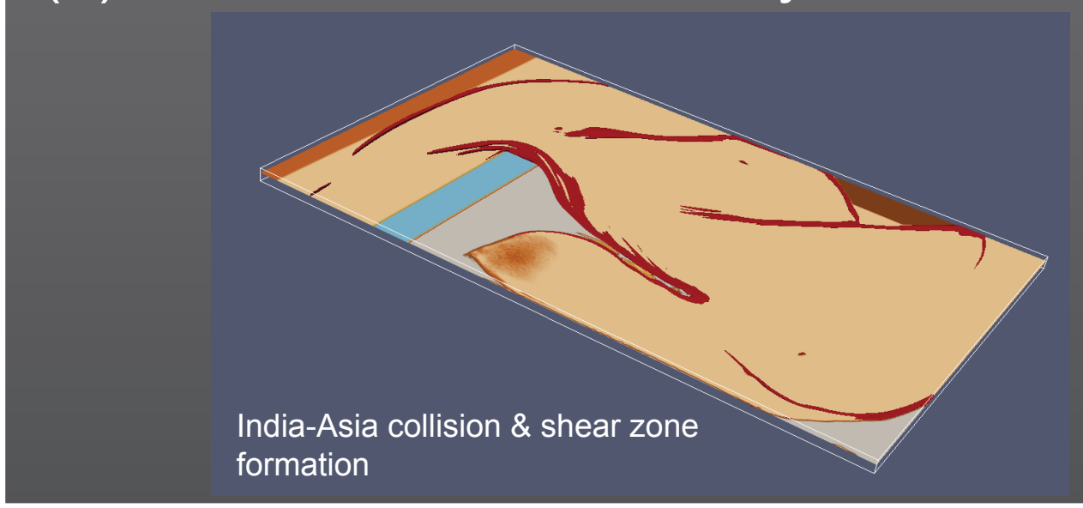
We are currently focussing on a few topics

(1) Better integrating the solvers with the PETSc framework, by using a DMSHELL for the FDSTAG

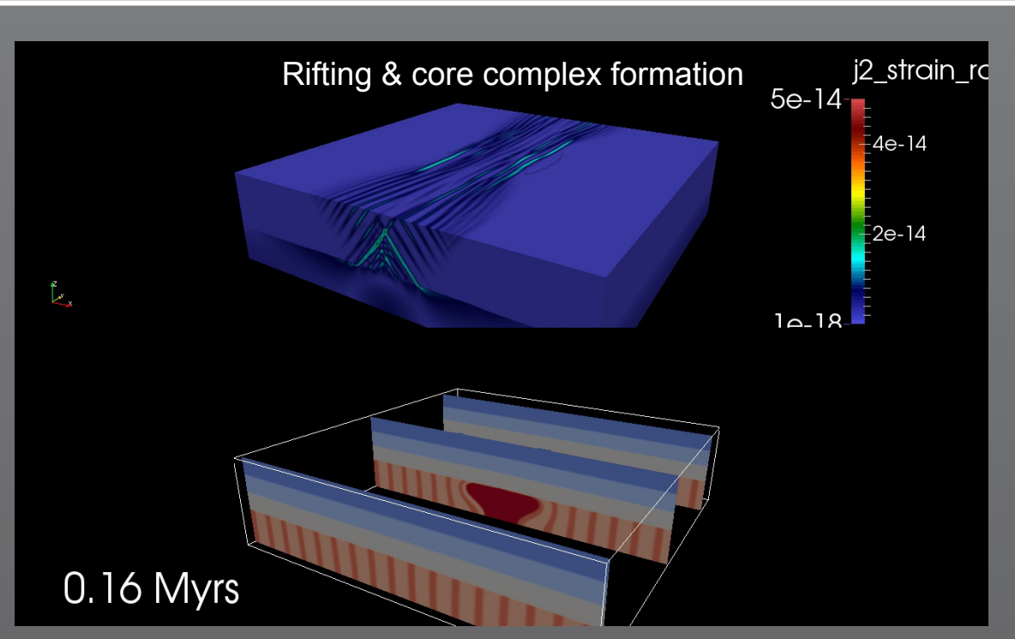
(2) Coupling of the vep Stokes code with two-phase melt migration.

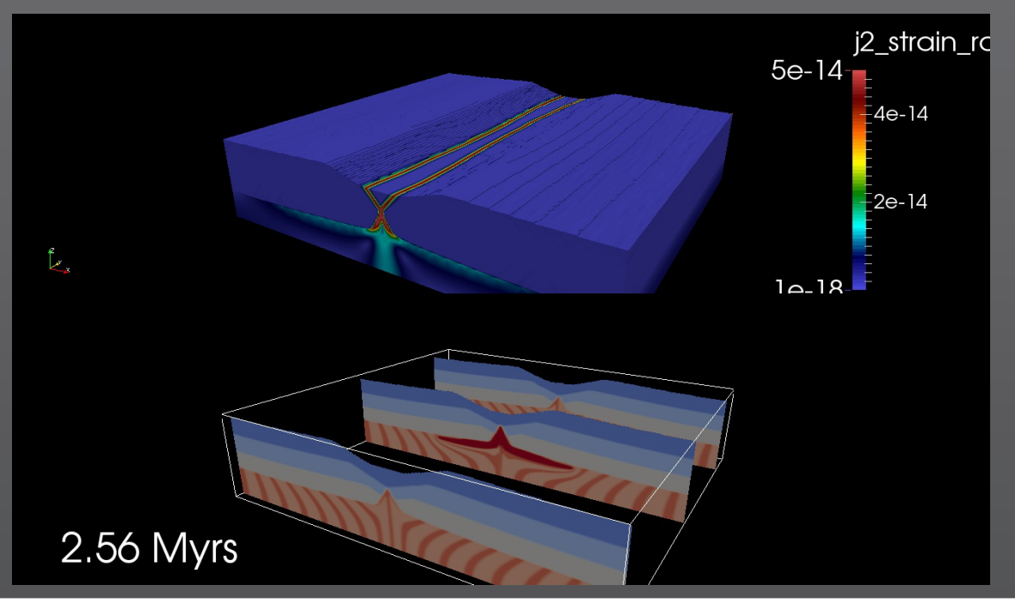
(3) Adding an explicit poro-visco-elasto-plastic solver to simulate hydrofracking

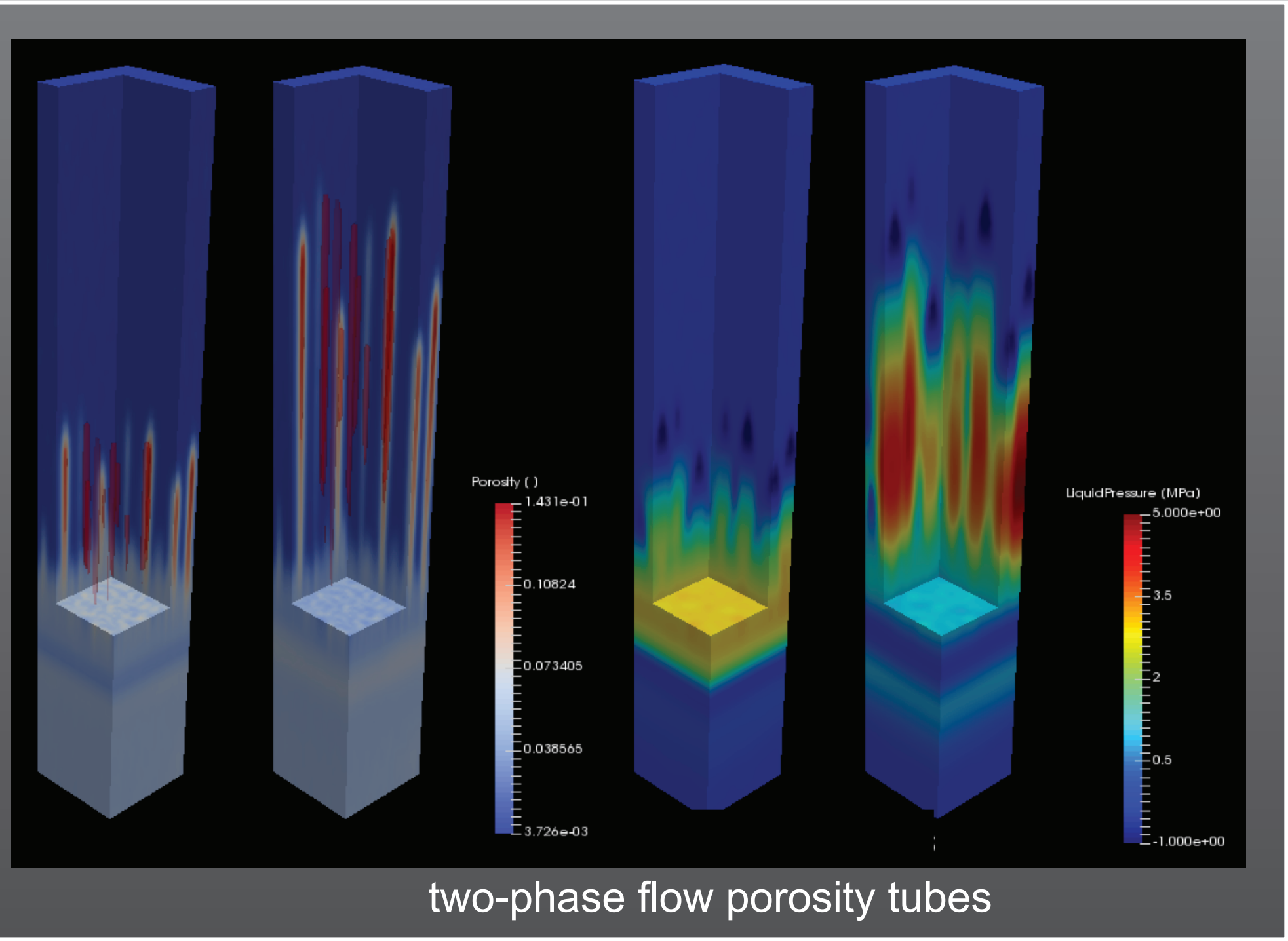
(4) Inflow/outflow boundary conditions.



India-Asia collision & shear zone formation







two-phase flow porosity tubes

CONCLUSIONS

- LaMEM is a parallel 3D marker-and-cell based deformation code suitable to model lithospheric processes.
- In typical geodynamic cases, FDSTAG is relative powerful (and cheap) alternative to finite-element calculations.
- The code can perform production runs on massively parallel machines.
- Conservative marker-in-cell interpolation was added & improves marker distributions.
- Coupled multigrid outperforms decoupled multigrid for nonlinear setups.
- Elasticity helps convergence for setups with brittle plastic rheology, even though more work on this topic is required to (either different stabilizing rheology or better convergent nonlinear algorithms).

ACKNOWLEDGEMENTS

Funding was provided by the European Research Council under the European Community's Seventh Framework program FP7/2007-2013) ERC Grant agreement #258830.

Download LaMEM at https://bitbucket.org/bkaus/lamem