# Benchmark Results of Active Tracer Particles In The Open Source Code ASPECT For Modeling Convection In The Earth's Mantle

J. Jiang [3], A. Kaloti [3], H. R. Levinson [3], N. Nyguen [3], H. Lokavarapu [2], Y. He [1], E.G. Puckett [1]

[1] Department of Mathematics, [2] Computational Infrastructure for Geodynamics (CIG), [3] 2016 Summer Mathematics & CIG REU Students
University of California, Davis, CA, 95616

## Overview

- We tested the new active tracer particle algorithm recently implemented in the open source code **ASPECT** for modeling mantle convection.
- We implemented a different bilinear interpolation method that in [3], in which we designed a new algorithm to preserve the upper and lower bounds of the values carried on the particles in each cell
- Used three different benchmarks to test these algorithms and the performances of all the methods. These benchmarks has finite difference [2] and finite element [1] and are steady (time independent) solutions of incompressible Stokes equations.

## Motivation

- We used three different benchmarks to test these algorithms and the performances of all the methods
- These benchmarks has finite difference [2] and finite element [1] and are steady (time independent) solutions of incompressible Stokes equations
- We discretize the density and viscosity by initially placing the true values of density and viscosity on the particle at in [1,2]. The density and viscosity are then interpolated from the particles onto the finite element grid. The resulting Stokes system is solved for the velocity and pressure, which is then compared to the exact solution

## The Incompressible Stokes Equations:

$$\mu \Delta \boldsymbol{u} - \nabla P = \rho \boldsymbol{g}$$
$$\nabla \cdot \boldsymbol{u} = 0$$

where $\mu$ is the viscosity, $\boldsymbol{u}$ is the velocity, $P$ is the pressure, $\rho$ is the density and $\boldsymbol{g}$ is the acceleration of gravity.

## The Interpolation Algorithms

For each of the properties, we used two different interpolation algorithms:

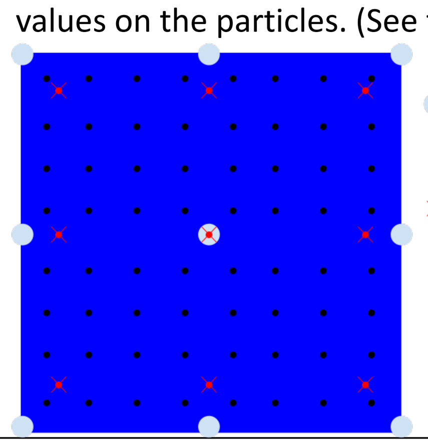
- **Arithmetic average:**

$$\bar{\phi}(x,y) = \frac{1}{N} * \sum_{i=1}^{N} \phi(x_i, y_i)$$

- **Bilinear:**

$$\phi(x,y) = c_0 + c_1 x + c_2 y$$

The coefficients of the bilinear function are obtained from the least squares fit of particle values and at their positions. A limiter is applied to ensure that the function value remains in the range of the maximum and minimum values on the particles. (See the algorithm described in the third column)

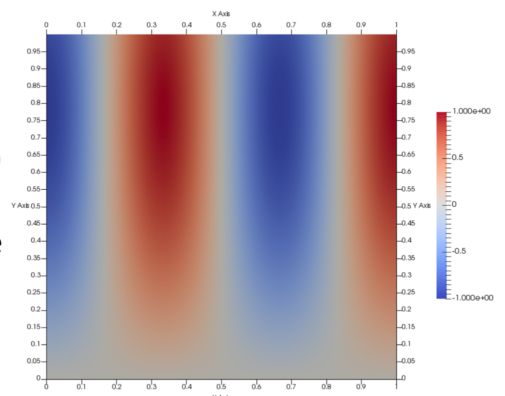



For each cell, the arithmetic average or the value of the (limited) bilinear function is interpolated onto the support points.

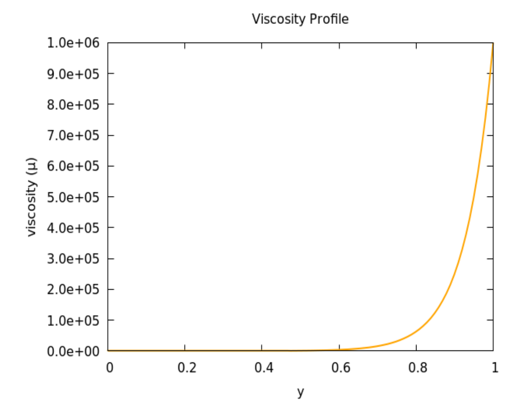
## SolKz

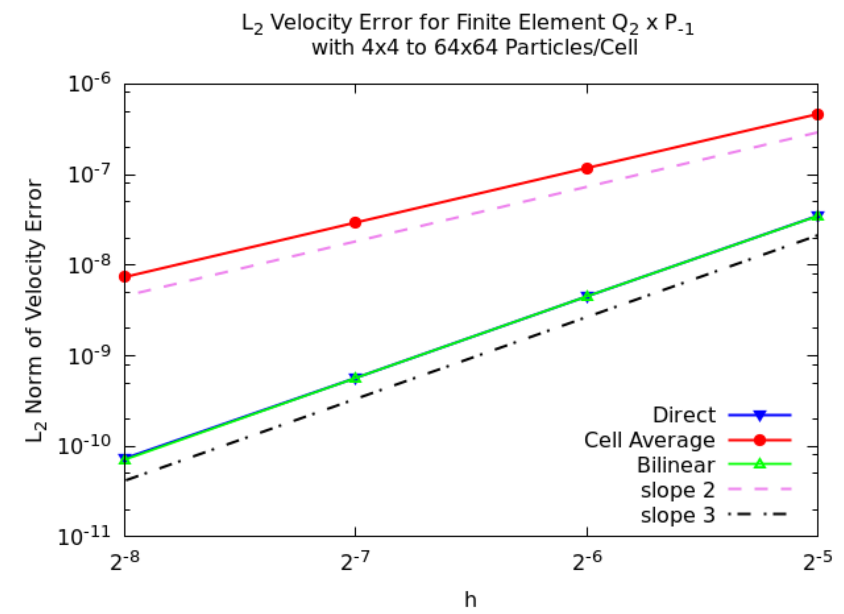
- SolKz is a smooth solution of the steady incompressible Stokes equations.
- The density is given by $\rho(x,y) = -sin(3\pi x) * cos(2y)$
- The viscosity is a smoothly varying function of the height: $\mu(y) = 10^{6y}$
- The optimal solution can be reached because the solution is smooth.

Density



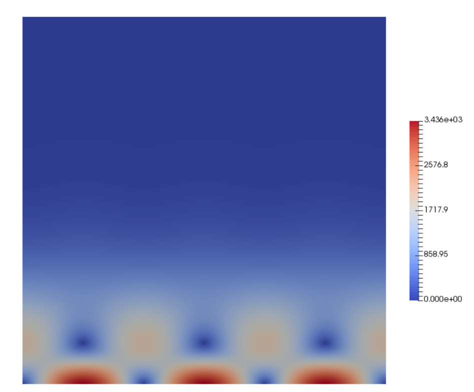
Viscosity



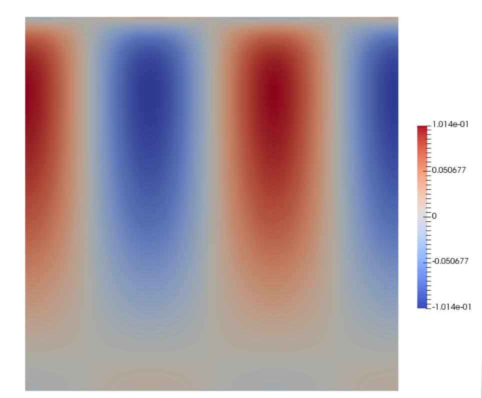




The convergence rate of the error in the velocity in the L2 norm on grid resolutions of h = 1 / 8 to 1 / 256

Velocity



Pressure



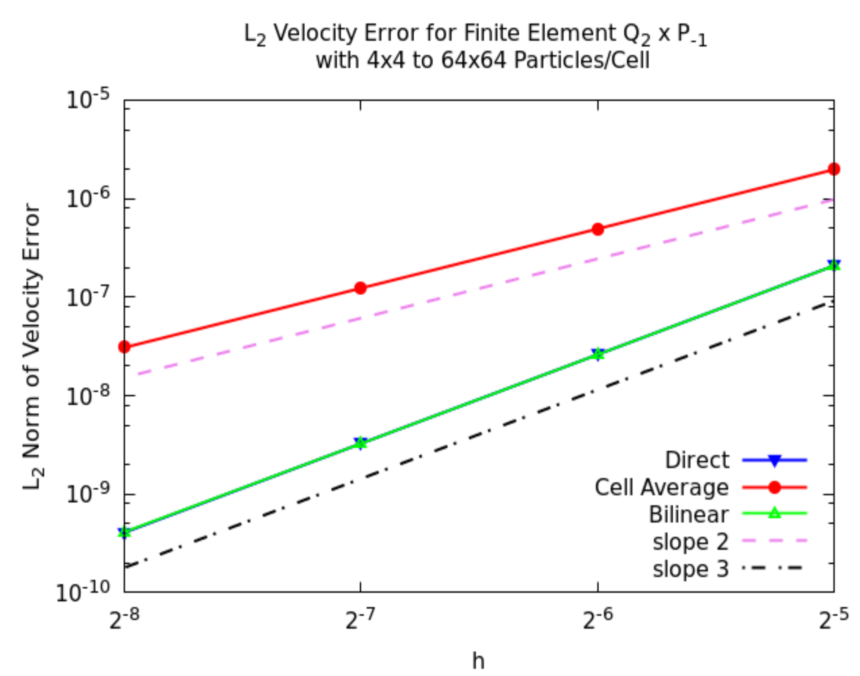
- Cell averaging is second-order accurate while bilinear interpolation is third-order accurate
- **Note:** The bilinear interpolation algorithm produces errors that are indistinguishable from the direct method

## SolCx

- In the SolCx benchmark, the domain is the unit square with $\rho(x,y) = -\sin(\pi y) * \cos(\pi x)$ and

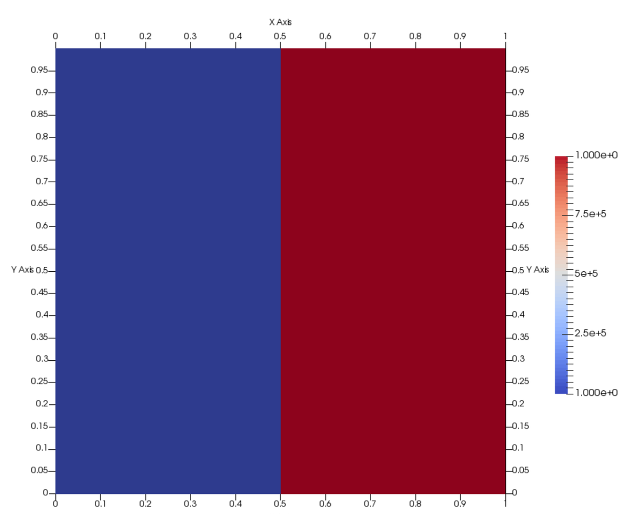
$$\mu(x) = \begin{cases} 1 & \text{if } x < 0.5 \\ 10^6 & \text{if } x \geq 0.5 \end{cases}$$

- Errors were highest when using particles with the cell average interpolation scheme
- Cell averaging is second-order accurate while bilinear interpolation is third-order accurate
- **Note:** The bilinear interpolation algorithm produces direct method

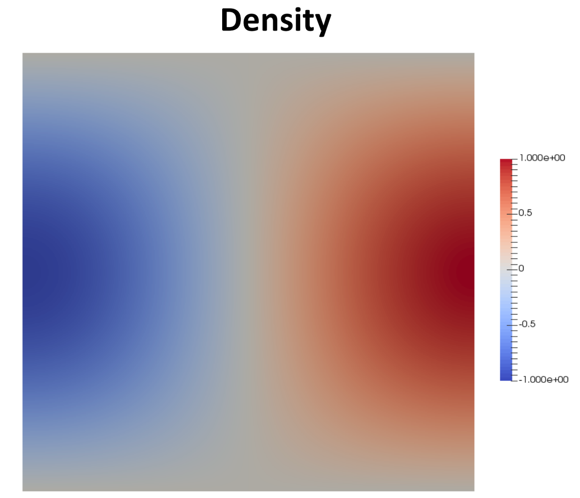



The convergence rate of the error in the velocity in the L2 norm on grid resolutions of h = 1 / 8 to 1 / 256

Viscosity



Density



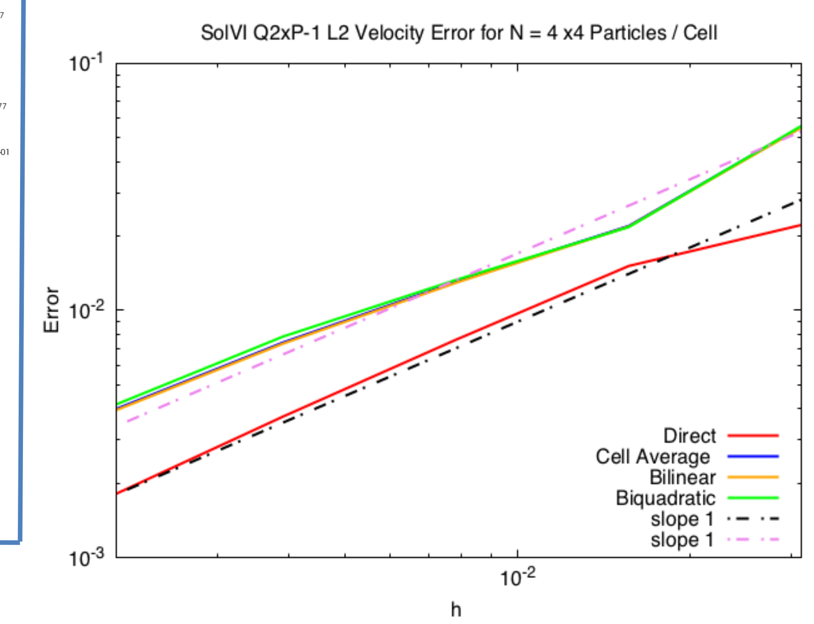
## SolVI (inclusion)

This benchmark focuses on a situation in which the viscosity occupies a circular region cell boundaries that can never be aligned with the interface between regions with different viscosities.

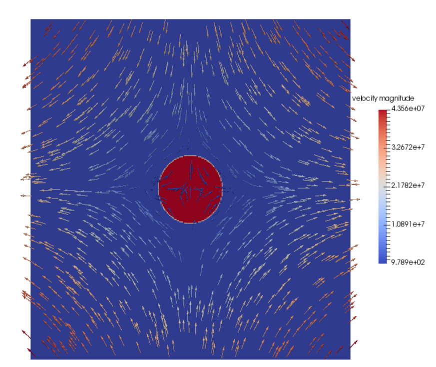
Therefore, in general, we should expect the error convergence rate be no better than the corresponding error convergence rate of SolCx benchmark when implemented with odd mesh. (a situation when viscosity jump is not aligned with cell boundaries).

- Regarding the SolVI benchmark, the convergence rates we obtain are only first-order, which is low compared with the other two benchmarks

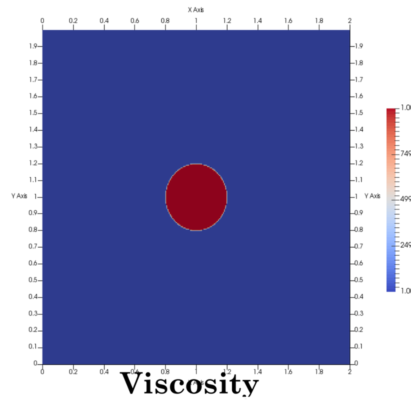
In addition, the boundary conditions for both SolKz and SolCx are free-slip condition, whereas the boundary condition of inclusion is Dirichlet boundary. This boundary condition requires certain shear forces so that the medium is compressed in vertical and stretched in horizontal direction.



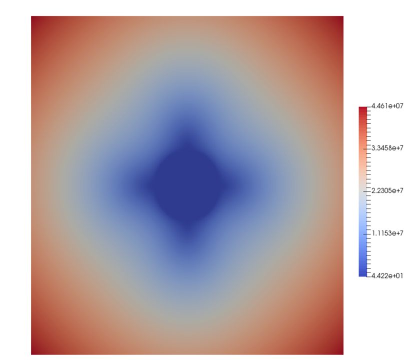

- All interpolation algorithms yield first-order convergence rates, the same as the SolCx odd mesh.
- The magnitude of the errors for the particle methods slightly larger errors than the direct method.
- All interpolation yield roughly the same error.



Velocity Field



Viscosity
μ1 : μ0 = 1000 : 1



Velocity Magnitude

## Algorithm to prevent overshoot and undershoot

bb

We have developed a new algorithm to maintain the bounds on the particle values when interpolating to Q_1 x P_0 elements.

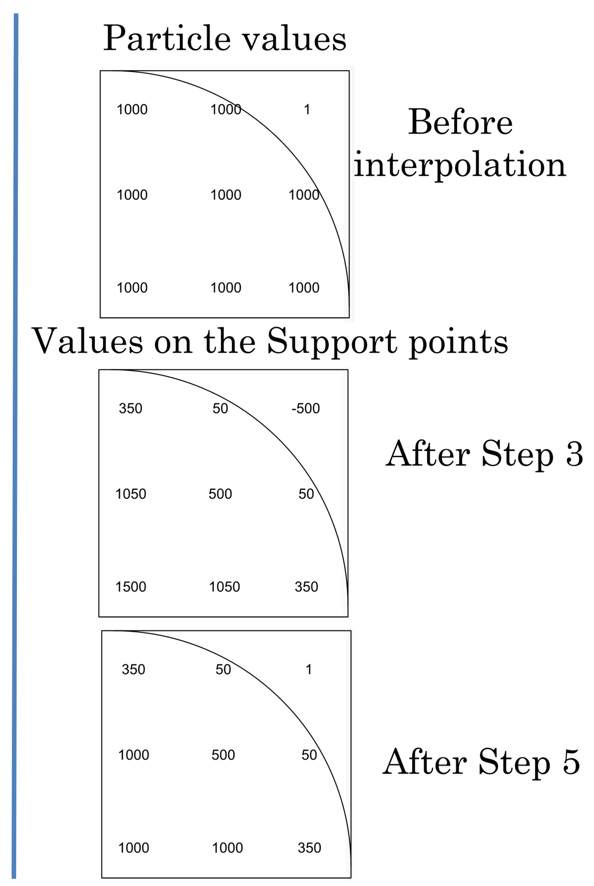
Step 1: Find the bilinear fit for all particles

Step 2: Compute max and min of all particles

Step 3: Calculate the approximation at the support points using the bilinear function found in Step 1.

Step 4: If a values at one of the support points is not between the max and min, modify the approximation at that support point to be either the max or min.

Step 5: return the approximation at the support points.




## Conclusions

- With enough particles and the appropriate polynomial degree for approximating the grid, our results have shown that the newly implemented active tracer particle algorithm converges to the exact solution of each of these benchmarks at the correct design rate
- For both SolKz and SolCx
  - The cell averaging method is second-order accurate and produces errors that are larger that the bilinear interpolation algorithm
  - The higher-order interpolation method bilinear produce convergence rates that are very close to the direct method
  - Error is reduced until we run either on the grid of 256x256, at which point the error for Bilinear fail to converge to its design rate
- Regarding SolVI benchmark, the convergence rates we obtain are only first-order, which is low compared with the other two benchmarks
  - This is due to the large viscosity jump that can not be aligned with the cell boundaries and large velocity gradient at the center of the shear flow
- We found that in **ASPECT** it was not possible to reduce the error in our approximations of the solutions of SolKz and SolCx below approximately $10^{-10}$ to $10^{-11}$
  - One can see this problem in Table 1 in [1] for the two norm of velocity with the $Q_3 x Q_2$ element combination

## REFERENCES

[1] M. Kronbichler, T. Heister, and W. Bangerth. High accuracy mantle convection simulation through modern numerical methods. Geophys. J. Int., 191(1):12-29, Oct 2012.

[2] T. Duretz, D. A. May, T. V. Gerya, and P. J. Tackley. Discretization errors and free surface stabilization in the finite difference and marker-in-cell method for applied geodynamics: A numerical study. Geochemistry, Geophysics, Geosystems, 12(7):1525-2027, 2011.

[3] M. Thielmann, D. A. May, and B. J. P. Kaus. Discretization errors in the hybrid finite element particle-in-cell method. Pure and Applied Geophysics., 171(9):2165-2184, 2014.

[4] S. Zhong. Analytic solutions for Stokes' flow with lateral variations in viscosity. Geophys. J. Int., 124(1):18-28, 1996.

Supported by the U. S. National Science Foundation under Award Numbers **ACI-1440811, EAR-0949446, EAR-1550901.**