PREPRINT to be submitted to *Geophys. J. Int.* (2018)

Stationary and Time-Dependent Benchmarks for Hybrid Finite Element/Particle-In-Cell Methods for Modeling 2D Incompressible Stokes Flow

Elbridge Gerry Puckett[1,*]
Harsha Lokavarapu[2]
Rene Gassmöller[2]
Wolfgang Bangerth[3]

[1] *Department of Mathematics, U. C. Davis, Davis, CA 95616, USA.*
[2] *Department of Earth and Planetary Science, U. C. Davis, Davis, CA 95616, USA.*
[3] *Department of Mathematics, Colorado State University, Fort Collins, CO 80523, USA.*
[*] *Corresponding author.*

September 7, 2018

SUMMARY

Combining finite element methods for the incompressible Stokes equations with particle-in-cell methods for approximating the solution of certain advection equations is an important technique in computational geodynamics that has been widely applied in mantle convection, lithosphere dynamics, and crustal-scale modeling. There are several benchmarks to measure the accuracy and convergence rate for these hybrid methods, although usually these benchmarks are instantaneous problems or have only qualitative comparison criteria. Thus, the convergence properties and accuracy of these hybrid finite element / particle-in-cell methods in real time-dependent problems remain uncertain. Moreover, the number of particles per cell that are necessary to achieve the design accuracy of the finite element is currently not well understood.

In this paper we modify two existing instantaneous benchmarks and present a new analytic benchmark for time-dependent incompressible Stokes flow in order to compare the convergence rate and accuracy of various combinations of finite element, particle advection, and particle interpolation methods. Using these benchmarks, we find that in order to retain the design accuracy of the finite element formulation, one needs to use a sufficiently accurate particle interpolation algorithm. Additionally, we observe that if the convergence rate of the particle interpolation algorithm is lower than the expected convergence rate of the finite element, one needs to increase the number of particles per cell as the mesh resolution increases (i.e., grid cell size decreases) in order to avoid a loss of accuracy in the numerical solution.

Our results will allow computational scientists to design new particle-in-cell methods with specific convergence rates. In addition, our new time-dependent benchmark provides a simple, standardized test for different software implementations, for comparison of existing algorithms, and for the assessment of new numerical methods for particle interpolation and advection. We provide a reference implementation of this benchmark in ASPECT ("Advanced Solver for Problems in Earth's ConvecTion"), an open source code for geodynamic modeling.

Key words: Particle-in-Cell Methods, Finite Element Method, Stokes Equations, Benchmark

1 INTRODUCTION

Computational geodynamic models have become important tools that are used by researchers to understand the dynamic processes observed in the solid Earth; for example, to model mantle convection, lithosphere dynamics, and crustal deformation.

Most of these models involve solving the Stokes equations with variable rock properties, viscosity, and density for the dependent variables velocity and pressure, coupled to the time-evolution of an advection-diffusion equation for the temperature, and, more generally, the advection of additional quantities that influence rock properties, such as chemical composition [Tackley(1998), McNamara & Zhong(2005), Dannberg & Gassmöller(2018)], grain size [Rozel et al.(2011), Thielmann et al.(2015), Dannberg et al.(2017), Mulyukova & Bercovici(2018)], or melt fraction and depletion [Fischer & Gerya(2016), Gassmöller et al.(2016)].

Consequently, a number of different techniques, with various advantages and disadvantages, have been developed to solve advection or advection-diffusion equations, such as particle-in-cell or marker-and-cell methods [Evans et al.(1957), Harlow & Welch(1965)], finite-difference or finite-element methods with various stabilization mechanisms [Brooks & Hughes(1982), Guermond & Pasquetti(2011)], and interface tracking methods, such as marker-chain or volume-of-fluid methods [Hirt & Nichols(1981)]. For a recent comparison see [Puckett et al.(2017)].

In particular, particle-in-cell (PIC) methods have a long history of being used by researchers in geodynamics computations [Tackley & King(2003), Poliakov & Podladchikov(1992), Moresi et al.(2003), Gerya & Yuen(2003), McNamara & Zhong(2004), Popov & Sobolev(2008)], since they are conceptually simple and do not require specialized algorithms or other techniques to stabilize the solution of the strongly advection dominated advection-diffusion equation. In PIC or related methods, the advected property is transferred to a set of discrete particles that are then advected with the flow. Since each particle's movement is independent of all of the other particles, this converts the partial differential equation for the advection of the quantity or quantities carried by the particles into a set of ordinary differential equations for each particle's movement. When the particles' properties are required for the solution of the Stokes equations for the next time step, they are interpolated or projected back onto the discrete grid and, if necessary, after this Stokes solve the properties on the particles are updated with the newly computed solution on the grid by interpolating this new solution back onto the particles.

Despite the long history of researchers using PIC methods in geodynamic codes, many challenges continue to exist in the implementation and application of these methods. In particular, PIC methods are difficult to combine with adaptively refined and dynamically changing meshes, since the number of particles per cell may vary widely during a computation and the numerical error and convergence properties of the method are difficult to determine precisely. Building on the algorithms described in [Gassmöller et al.(2018)], in Section 4 we will describe advances in determining the accuracy of particle-in-cell methods with finite element based Stokes solvers in a way that will allow other researchers to *quantitatively* compare the results obtained with their codes to ours. In particular, we quantify the influence of the following building blocks of PIC methods on the accuracy of the solution: (1) the number and distribution of particles, (2) the interpolation of particle properties to field based properties, and (3) the integration of the motion of the particles over time. In order to achieve this we start by reproducing the instantaneous benchmark results SolCx and SolKz [Duretz et al.(2011), Zhong(1996)], and discuss how the convergence rate of the computed solution depends on different finite element and interpolation algorithm combinations. In particular, we will extend the previous work of [Thielmann et al.(2014)] by investigating different numbers of particles per cell, different initial positions for the particles and examine the difference between the error caused by interpolating the density versus the the viscosity onto the grid. We demonstrate that in order to recover the intrinsic convergence rate of a given finite element for these benchmarks, we need a sufficiently accurate particle interpolation algorithm and sufficiently many particles per cell as a function of the cell size. In addition, if the particles only carry density information the particle interpolation algorithm can be of lower order than if the particles also carry viscosity information.

In Section 5 we then present a new, *time-dependent* benchmark (the derivation of which may be found in Appendix A) with a analytical solution that is independent of time. This problem allows us to evaluate the error and convergence rates for time-dependent computations of incompressible Stokes flow coupled to a PIC advection method. It also allows us to measure the influence of the method for integrating the particle positions in time on the overall error between the computed solution and the true solution. We present results for this benchmark as implemented in the open-source geodynamic modeling code ASPECT [Kronbichler et al.(2012), Heister et al.(2017)] and discuss the conditions under which the particle advection algorithm affects the accuracy of the Stokes solution. It is our intention that these results will act as reference results for future code comparison studies of time dependent PIC advection algorithms, and will allow researchers to design PIC methods that use a combination of techniques to ensure optimal accuracy of the numerical method as a whole. To our knowledge this is the first time-dependent benchmark for hybrid finite element / PIC methods in the computational geodynamic community.

2 GOVERNING EQUATIONS

Geologic deformation over long time scales is commonly modeled by the incompressible Stokes equations for a slow-moving fluid, using a spatially and temporally variable density that depends nonlinearly on both the strain rate of the fluid, as well as temperature, chemical composition, and other factors. The driving force for the flow is provided by a buoyancy term that results from the spatial variability of the density, again due to temperature and chemical composition differences.

The incompressible Stokes equations that describe this type of flow are given by a force balance and a mass continuity equation:

$$-\nabla \cdot \eta\, \varepsilon(\boldsymbol{u}) + \nabla p = \rho\, \boldsymbol{g}, \tag{1}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{2}$$

where $\boldsymbol{u}$ is the velocity, p the pressure, ρ the density, η the viscosity, and $\boldsymbol{g}$ the gravity. Furthermore, $\varepsilon(\boldsymbol{u}) = \frac{1}{2}(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)$ is the symmetric gradient of the velocity and denotes the strain rate within the fluid.

In more realistic applications, the mass continuity equation (2) has to be replaced by an equation that allows for compressible effects. However, as this is tangential to the purpose of the current paper, we will simply assume that the fluid is incompressible. In either case, the equations above are augmented by appropriate boundary conditions.

A complete description of mantle convection would couple the equations above to a set of advection-diffusion equations for the temperature and chemical compositions that are transported along with the velocity $\boldsymbol{u}$ (see [Schubert et al.(2001)]). If we denote any of these fields by $\phi = \phi(\boldsymbol{x}, t)$, then they typically satisfy an advection-diffusion equation of the form

$$\frac{D\phi}{Dt} + \kappa \Delta \phi = \frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi + \nabla \kappa \nabla \phi = H, \tag{3}$$

augmented by appropriate initial conditions $\phi(\boldsymbol{x}, 0) = \phi_0(\boldsymbol{x})$ and, if necessary, boundary conditions. In typical applications the equation is dominated by the advection term $\boldsymbol{u} \cdot \nabla \phi$, and the contributions by the diffusion term $\nabla \kappa \nabla \phi$ and source term H are rather small (if ϕ denotes the temperature) or are completely negligible (if ϕ denotes chemical compositions).

The importance of these additional fields lies in the fact that the viscosity and density coefficients in the Stokes equations above depend on strain rate $\varepsilon(\boldsymbol{u})$ and pressure p, but also on the temperature and chemical composition variables. Consequently, the resulting set of equations is coupled, nonlinear, and time dependent. An accurate solution of the complete model therefore requires an accurate way of advecting along these additional quantities.

3 NUMERICAL METHODOLOGY

The above equations can be solved by direct discretization via finite element, finite volume, or finite difference methods, or a variety of other methods (see, for example [Donea & Huerta(2003), Deubelbeiss & Kaus(2008), Gerya(2009), Ismail-Zadeh & Tackley(2010)].

However, discretizing advection-dominated problems such as (3) without introducing oscillations or excessive diffusion is not trivial. As discussed above, many mantle convection codes have instead used particle schemes to advect along properties of rocks. In these schemes, a number of particles $k = 1 \ldots N$ are characterized by their location $\boldsymbol{x}_k(t)$ and associated properties ϕ_k. Their location then evolves according to the ordinary differential equation

$$\frac{d}{dt}\boldsymbol{x}_k(t) = \boldsymbol{u}(\boldsymbol{x}_k(t), t), \qquad \boldsymbol{x}_k(0) = \boldsymbol{x}_{k,0}, \tag{4}$$

and coefficients such as the viscosity η and density ρ in (1) are computed at points $\boldsymbol{x}$ not only based on field quantities such as velocity and pressure at $\boldsymbol{x}$, but also based on the quantities ϕ_k of particles located "close" to $\boldsymbol{x}$.

While conceptually simple to implement, this approach requires (i) transferring data from field-based quantities to particle locations, (ii) advecting the particles according to (4), and finally (iii) transferring data back from particle locations to quadrature points when assembling matrices and right hand sides for the Stokes equation.

All of these three steps introduce errors into the solution process: In the first step, the exact solution $\boldsymbol{u}(t)$ is not available, and one has to use the numerical approximation $\boldsymbol{u}_h(t^n)$ that was found by solving the Stokes equation, and that is only a spatial approximation and typically only available at discrete time steps. This error therefore depends on the accuracy of the computed velocity field, and the time-stepping scheme. In the second step, the numerical integration of (4) yields a trajectory $\boldsymbol{x}_h(t)$ that is different from $\boldsymbol{x}(t)$ even if the velocity were known exactly, depending on the accuracy of the ODE solver scheme. Finally, no particle will typically be located on a quadrature point, and the required property $\phi(\boldsymbol{x})$ will need to be interpolated in one of many possible, approximate way from the properties ϕ_k of nearby particles.

We will assess these errors in Section 4 and Section 5 in a number of benchmarks, for different Stokes discretizations, initial particle patterns, ODE solvers, and particle interpolation methods, which we will describe in this section.

3.1 Discretization of the Stokes system

The advection of particles can only be as accurate as the underlying velocity field that is used to advect them. In this work, the velocity is obtained by using finite elements to discretize and solve the Stokes equations. Specifically, we will employ the common $Q_k \times Q_{k-1}$ "Taylor-Hood" element [Taylor & Hood(1973)] in which the velocity and pressure are discretized by continuous finite elements of degrees k and $k-1$ on quadrilaterals or hexahedra, respectively. For comparison to the existing results of [Thielmann et al.(2014)], we will also use $Q_k \times P_{-(k-1)}$ elements in which the pressure is discretized using discontinuous polynomials of (total) degree $k-1$. Based on finite-element theory we expect both the $Q_k \times Q_{k-1}$ and the $Q_k \times P_{-(k-1)}$ elements to show optimal convergence order [Bercovier & Pironneau(1979)], i.e. to show a decay of the velocity and pressure errors, when measured in the L_2 norm, as h^{k+1} and h^k, respectively, where h is the element size of the mesh. We show in Section 4 and 5 that this is indeed the case for our implementation and model setups. In all of our experiments we assume that the Stokes equation is solved either with a direct solver, or with a sufficiently tight tolerance on an iterative solver, so that the only remaining error stems from the spatial discretization of the flow field intrinsic to the used finite-element.

3.2 Generation of particles

In time dependent problems, particles are transported along with the flow; after some time, they will no longer be at specific locations. Therefore, algorithms that reconstruct coefficients from particles' properties need to be general and deal with both arbitrary particle numbers and locations on each cell. However, the test cases we will consider in Section 4, will only solve a single time step without advecting particles. Thus, the particles are located where they were created, and we need to make sure not to rely on a particular particle distribution that controls our results. On the other hand this experiment allows us to determine the influence of different particle distributions on the accuracy of the solution.

We will consider two strategies for choosing the initial particle locations $\boldsymbol{x}_k(0) = \boldsymbol{x}_{k,0}$:

(i) Create a number of particles N_K on a regular grid of points within the cell $\hat{K}$ in the reference domain, from where they are mapped to the corresponding points on each cell K of the triangulation.

(ii) Create a number of particles N_K within each cell K, with locations drawn from a uniform probability distribution on K; here, N_K is equal to the fraction of the volume occupied by cell K relative to the volume of the global domain Ω, times the global number of particles N.

The practical implementation of both algorithms in arbitrary geometries is described in [Gassmöller et al.(2018)]. Note, that approach (i) will lead to a constant particle count per cell, while approach (ii) will lead to a constant particle density per area.

For the time-dependent benchmark cases in Section 5 initial particle locations are less critical as they are moving from their starting positions, and for simplicity we always generate particles at regular grid locations (approach (i)).

3.3 Advection of particles

As described above the advection of particles involves solving (4) for their position, which we do using a Runge-Kutta method of second (RK2) or fourth order (RK4). As expected and as shown for our implementation before (see supporting information of [Gassmöller et al.(2018)]) the error of particle positions for a given static flow field reduces as Δt^2 and Δt^4 for RK2 and RK4 respectively. However, because we use a time-stepping scheme for our Stokes solution that is second order accurate at a finite time (BDF2) any particle advection method is limited for a time-varying velocity field to be second order accurate in time. Since the analytic solution of the benchmarks in Section 5 is time-independent, this will not be a limiting factor for our experiments. Nevertheless, this limitation has to be considered for realistic applications. We also note that our discrete velocity solutions are only divergence-free in an integral sense, and evaluating the velocity at the particle locations introduces a spurious velocity divergence that can lead to the clustering of particles in certain flow patterns. This phenomenon can be improved using velocity corrections known as conservative velocity interpolation [Wang et al.(2015), Pusok et al.(2017)]. We did not employ such a pattern in our benchmarks, as we did not observe this clustering, and we were mostly concerned with the optimal convergence rate possible with the unmodified advection schemes. Nevertheless, it would be an interesting future study to quantify the influence of such velocity modifications on the accuracy of the particle advection.

3.4 Interpolation of particle data

Since particles carry material properties ϕ_k that enter the assembly of the linear systems used to solve for the field-based quantities, we need to define how these material properties can be evaluated at quadrature points $\boldsymbol{x}$ that do not, in general, coincide with the location of any of the particles. In particular, let K be a cell, $I_K \subseteq [1, N]$ be the set of indices of those particles that are located on K, and $N_K = |I_K|$ be their number. Then we consider the following strategies to evaluate property ϕ at an arbitrary location $\boldsymbol{x}$ based on the information $\{\phi_k\}_{k\in I_K}$ that is available on K alone:

(i) Piecewise constant averages: To obtain this particle interpolation function we average the material properties among all particles located on cell K:

$$\phi|_K = \frac{1}{N_K} \sum_{k\in I_K} \phi_k. \tag{5}$$

The value $\phi(\boldsymbol{x})$ is then computed by finding the cell K within which $\boldsymbol{x}$ is located, and taking the local average on K. In theory one could use different averaging schemes than arithmetic averaging, e.g. harmonic or geometric averaging, but since it was shown before that these schemes converge with the same order (though varying absolute accuracy) to the correct solution [Thielmann et al.(2014)], we here limit ourselves to arithmetic averaging.

(ii) Least squares (bi-/tri-)linear interpolation: In this algorithm, we seek a function ϕ that is (bi-/tri-)linear on each cell K. We will allow it to be discontinuous between cells, and in that case it can be computed locally on each cell independently. Specifically, we seek $\phi|_K$ so that it minimizes the squared error,

$$\epsilon^2 = \sum_{k\in I_K} \left[\phi|_K(\boldsymbol{x}_k) - \phi_k\right]^2, \tag{6}$$

where $\boldsymbol{x}_k$ is the location of particle k with associated property ϕ_k.
The minimizer $\phi|_K$ is found by solving a 4×4 matrix in 2 dimensions, or an 8×8 matrix in three dimensions, for the coefficients of the (bi-/tri-)linear least-squares approximation.
To obtain material property values at an arbitrary $\boldsymbol{x}$ in K then only requires evaluating $\phi|_K(\boldsymbol{x})$, i.e., evaluating the (bi-/tri-)linear shape functions of the approximand times their corresponding coefficient values.

Our expectation for these interpolation methods is that the difference between analytic and approximated density and viscosity with interpolation option (i) will decrease linearly with cell-size, while it should decrease with h^2 for option (ii). We will test and confirm this hypothesis in Section 5, which allows us to interpret the error results for velocity and pressure in terms of their dependence on the error in viscosity and density.

4 INSTANTANEOUS BENCHMARKS

The first set of benchmarks we will consider only solve a single time step; thus, the positions of particles are known *exactly.* The benchmarks are therefore intended to test the influence of initial particle distributions, Stokes discretizations, and the transfer of information from particles to field-based quantities.

Specifically, we will consider the SolKz and SolCx benchmarks [Revenaugh & Parsons(1987), Zhong(1996)] that have previously been used to test the accuracy of

Stokes solvers in the presence of a spatially variable viscosity [Duretz et al.(2011), Kronbichler et al.(2012)]. For both benchmarks, an exact solution for the velocity and pressure fields is available. We can then compare the convergence order we obtain if (i) we use the exact density and viscosity when assembling the finite element matrix for the Stokes system, or (ii) we use viscosity and density values that are interpolated from a set of nearby particles that have each been initialized using the exact values at their respective location.

As we will show, in this benchmark the way we interpolate from nearby particles to quadrature points greatly matters in retaining (or not retaining) the convergence order of the finite element scheme. To assess this quantitatively, we will evaluate the difference between the known, exact solution and the computed, approximate solution in the L_2 norm, which is defined as:

$$\|f\|_{L_2} = \sqrt{\int \|f(x,y,z)\|^2 d\Omega} \tag{7}$$

Therefore, we compute $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2}$ for the velocity, and $\|p - p_h\|_{L_2}$ for the pressure, respectively. In both cases, the involved integrals are approximated through quadrature using a Gauss formula with two more quadrature points in each coordinate direction than the polynomial degree of the velocity element; this guarantees both an accurate evaluation of the integral and avoids inadvertent super-convergence effects.

We will defer to the next section a discussion of time dependent cases where we also have to deal with the additional error introduced by inexact advection of particle locations.

4.1 SolKz

The SolKz benchmark [Duretz et al.(2011)] uses a smoothly varying viscosity on a 2D square domain with height and width of one. It uses tangential boundary conditions on the boundary, a vertical gravity of 1, and chooses the density field in such a way that one can construct an exact solution for the Stokes equation with the given viscosity.

Specifically, the viscosity varies with depth y as

$$\eta(x,y) = e^{2By}, \tag{8}$$

where B is chosen such that the viscosity ratio between top and bottom is 10^6. The density is given by

$$\rho(x,y) = -\sin(2y)\cos(3\pi x). \tag{9}$$

We begin by investigating the influence of the initial particle locations on the convergence rate of the velocity and pressure solution for either of the two interpolation methods discussed before. We show these results in Table 1 for different resolutions, and remark that although both investigated methods converge with different rates (we will discuss this difference later in this section), the initial particle locations do not influence the convergence rate significantly. Nevertheless, the absolute errors are larger for random particle locations, likely because some cells receive unfavorable particle locations (e.g. a high particle density in only a small volume of the cell). We chose to present a large number of particles per cell to ensure the optimal convergence rate is reached, however we observed that for smaller numbers of particles per cell the difference between the two methods increases. This is intuitive, as for an infinite number of particles the two methods should generate similar particle locations, namely particles in every possible location, while for few particles all of them could be randomly generated in a very small part of the cell, leaving a large region unsampled. Having established that the choice of initial particle locations does not influence the achieved convergence rate, we will conduct all other experiments with a regular particle distribution, which delivers more reproducable model results.

Table 1. Velocity errors $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2}$ and pressure errors $\|p - p_h\|_{L_2}$ for the SolKz benchmark using the $Q_2 \times P_{-1}$ Stokes element, for arithmetic averaging and bilinear least squares interpolation methods, and for regular and random particle distributions as discussed in Section 3.2. "PPC" stands for particles per cell.

Arithmetic average

		regular		random	
h	PPC	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2}$	rate	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2}$	rate
$\frac{1}{8}$	100	$7.05 \cdot 10^{-6}$	-	$7.08 \cdot 10^{-6}$	-
$\frac{1}{16}$	100	$1.86 \cdot 10^{-6}$	1.92	$1.95 \cdot 10^{-6}$	1.86
$\frac{1}{32}$	100	$4.81 \cdot 10^{-7}$	1.95	$4.87 \cdot 10^{-7}$	2.00
$\frac{1}{64}$	100	$1.22 \cdot 10^{-7}$	1.98	$1.29 \cdot 10^{-7}$	1.92
$\frac{1}{128}$	100	$3.05 \cdot 10^{-8}$	2.00	$2.93 \cdot 10^{-8}$	2.13
$\frac{1}{256}$	100	$7.63 \cdot 10^{-9}$	2.00	$7.91 \cdot 10^{-9}$	1.89
		$\|p - p_h\|_{L_2}$		$\|p - p_h\|_{L_2}$	
$\frac{1}{8}$	100	$1.91 \cdot 10^{-2}$	-	$1.92 \cdot 10^{-2}$	-
$\frac{1}{16}$	100	$1.24 \cdot 10^{-2}$	0.62	$1.24 \cdot 10^{-2}$	0.63
$\frac{1}{32}$	100	$6.57 \cdot 10^{-3}$	0.92	$6.60 \cdot 10^{-3}$	0.91
$\frac{1}{64}$	100	$3.33 \cdot 10^{-3}$	0.98	$3.35 \cdot 10^{-3}$	0.98
$\frac{1}{128}$	100	$1.67 \cdot 10^{-3}$	1.00	$1.68 \cdot 10^{-3}$	1.00
$\frac{1}{256}$	100	$8.37 \cdot 10^{-4}$	1.00	$8.40 \cdot 10^{-4}$	1.00

Bilinear least squares

		regular		random	
h	PPC	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2}$	rate	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2}$	rate
$\frac{1}{8}$	100	$1.72 \cdot 10^{-6}$	-	$1.68 \cdot 10^{-6}$	-
$\frac{1}{16}$	100	$2.46 \cdot 10^{-7}$	2.81	$2.49 \cdot 10^{-7}$	2.75
$\frac{1}{32}$	100	$3.50 \cdot 10^{-8}$	2.81	$3.52 \cdot 10^{-8}$	2.82
$\frac{1}{64}$	100	$4.56 \cdot 10^{-9}$	2.94	$4.71 \cdot 10^{-9}$	2.90
$\frac{1}{128}$	100	$5.95 \cdot 10^{-10}$	2.94	$6.55 \cdot 10^{-10}$	2.85
$\frac{1}{256}$	100	$8.41 \cdot 10^{-11}$	2.82	$1.05 \cdot 10^{-10}$	2.64
		$\|p - p_h\|_{L_2}$		$\|p - p_h\|_{L_2}$	
$\frac{1}{8}$	100	$4.53 \cdot 10^{-3}$	-	$4.72 \cdot 10^{-3}$	-
$\frac{1}{16}$	100	$1.30 \cdot 10^{-3}$	1.80	$1.33 \cdot 10^{-3}$	1.83
$\frac{1}{32}$	100	$3.42 \cdot 10^{-4}$	1.93	$3.49 \cdot 10^{-4}$	1.93
$\frac{1}{64}$	100	$8.67 \cdot 10^{-5}$	1.98	$8.84 \cdot 10^{-5}$	1.98
$\frac{1}{128}$	100	$2.17 \cdot 10^{-5}$	2.00	$2.22 \cdot 10^{-5}$	1.99
$\frac{1}{256}$	100	$5.43 \cdot 10^{-6}$	2.00	$5.54 \cdot 10^{-6}$	2.00

Given that both viscosity and density in this benchmark are smooth, we expect the velocity and pressure fields to also be sufficiently smooth for a finite element method to obtain the optimal convergence order if the coefficients are evaluated exactly at each quadrature point during the assembly of linear systems. In accordance with earlier studies [Thielmann et al.(2014)] we will call this the "*direct method*". As discussed, for both the $Q_k \times Q_{k-1}$ and the $Q_k \times P_{-(k-1)}$ ele-

Table 2. Velocity errors $\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2}$ and pressure errors $\|p-p_h\|_{L_2}$ for the SolKz benchmark using the $Q_2 \times P_{-1}$ (top rows), and $Q_3 \times Q_2$ (bottom rows) Stokes elements. Here "PPC" stands for particles per cell.

$Q_2 \times P_{-1}$

$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2}$	direct method			arithmetic average			bilinear least squares	
h	error	rate	PPC	error	rate	PPC	error	rate
$\frac{1}{8}$	$1.51 \cdot 10^{-6}$	-	4	$6.32 \cdot 10^{-6}$	-	4	$2.24 \cdot 10^{-6}$	-
$\frac{1}{16}$	$2.50 \cdot 10^{-7}$	2.60	4	$1.61 \cdot 10^{-6}$	1.97	4	$3.61 \cdot 10^{-7}$	2.63
$\frac{1}{32}$	$3.52 \cdot 10^{-8}$	2.80	4	$4.15 \cdot 10^{-7}$	1.96	9	$4.62 \cdot 10^{-8}$	2.97
$\frac{1}{64}$	$4.53 \cdot 10^{-9}$	3.00	4	$1.05 \cdot 10^{-7}$	1.98	25	$5.3 \cdot 10^{-9}$	3.12
$\frac{1}{128}$	$5.7 \cdot 10^{-10}$	3.00	4	$2.63 \cdot 10^{-8}$	2.00	49	$6.75 \cdot 10^{-10}$	2.97
$\frac{1}{256}$	$7.23 \cdot 10^{-11}$	3.00	4	$6.58 \cdot 10^{-9}$	2.00	100	$8.41 \cdot 10^{-11}$	3.00
$\frac{1}{512}$	$9.14 \cdot 10^{-12}$	3.00	4	$1.64 \cdot 10^{-10}$	2.00	196	$1.05 \cdot 10^{-11}$	3.00
$\|p-p_h\|_{L_2}$								
$\frac{1}{8}$	$5.02 \cdot 10^{-3}$	-	4	$1.93 \cdot 10^{-2}$	-	4	$4.58 \cdot 10^{-3}$	-
$\frac{1}{16}$	$1.33 \cdot 10^{-3}$	1.90	4	$1.24 \cdot 10^{-2}$	0.64	4	$1.31 \cdot 10^{-3}$	1.80
$\frac{1}{32}$	$3.44 \cdot 10^{-4}$	2.00	4	$6.58 \cdot 10^{-3}$	0.92	9	$3.43 \cdot 10^{-4}$	1.94
$\frac{1}{64}$	$8.68 \cdot 10^{-5}$	2.00	4	$3.33 \cdot 10^{-3}$	0.98	25	$8.67 \cdot 10^{-5}$	1.98
$\frac{1}{128}$	$2.17 \cdot 10^{-5}$	2.00	4	$1.67 \cdot 10^{-3}$	1.00	49	$2.17 \cdot 10^{-5}$	2.00
$\frac{1}{256}$	$5.43 \cdot 10^{-6}$	2.00	4	$8.37 \cdot 10^{-4}$	1.00	100	$5.43 \cdot 10^{-6}$	2.00
$\frac{1}{512}$	$1.36 \cdot 10^{-6}$	2.00	4	$4.19 \cdot 10^{-4}$	1.00	196	$1.36 \cdot 10^{-6}$	2.00

$Q_3 \times Q_2$

$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2}$	direct method			arithmetic average			bilinear least squares	
h	error	rate	PPC	error	rate	PPC	error	rate
$\frac{1}{8}$	$3.1 \cdot 10^{-7}$	-	4	$5.78 \cdot 10^{-6}$	-	9	$1.26 \cdot 10^{-6}$	-
$\frac{1}{16}$	$2.48 \cdot 10^{-8}$	3.64	4	$1.36 \cdot 10^{-6}$	2.08	9	$1.64 \cdot 10^{-7}$	2.94
$\frac{1}{32}$	$1.59 \cdot 10^{-9}$	3.96	4	$3.34 \cdot 10^{-7}$	2.03	16	$2.09 \cdot 10^{-8}$	2.97
$\frac{1}{64}$	$9.9 \cdot 10^{-11}$	4.00	4	$8.27 \cdot 10^{-8}$	2.01	36	$2.27 \cdot 10^{-9}$	3.20
$\frac{1}{128}$	$6.23 \cdot 10^{-12}$	3.99	4	$2.06 \cdot 10^{-8}$	2.01	81	$2.52 \cdot 10^{-10}$	3.17
$\frac{1}{256}$			4	$5.13 \cdot 10^{-9}$	2.00	169	$3.01 \cdot 10^{-11}$	3.07
$\frac{1}{512}$			4	$1.28 \cdot 10^{-9}$	2.00	361	$3.66 \cdot 10^{-12}$	3.04
$\|p-p_h\|_{L_2}$								
$\frac{1}{8}$	$7.04 \cdot 10^{-4}$	-	4	$1.86 \cdot 10^{-2}$	-	9	$1.37 \cdot 10^{-3}$	-
$\frac{1}{16}$	$1.15 \cdot 10^{-4}$	2.61	4	$8.27 \cdot 10^{-3}$	1.17	9	$1.18 \cdot 10^{-3}$	0.21
$\frac{1}{32}$	$1.68 \cdot 10^{-5}$	2.78	4	$3.06 \cdot 10^{-3}$	1.43	16	$3.52 \cdot 10^{-4}$	1.74
$\frac{1}{64}$	$2.3 \cdot 10^{-6}$	2.89	4	$1.11 \cdot 10^{-3}$	1.47	36	$9.19 \cdot 10^{-5}$	1.94
$\frac{1}{128}$	$3.03 \cdot 10^{-7}$	2.92	4	$3.99 \cdot 10^{-4}$	1.48	81	$2.32 \cdot 10^{-5}$	1.98
$\frac{1}{256}$	$3.89 \cdot 10^{-8}$	2.96	4	$1.43 \cdot 10^{-4}$	1.48	169	$5.83 \cdot 10^{-6}$	2.00
$\frac{1}{512}$	$4.94 \cdot 10^{-9}$	2.98	4	$5.07 \cdot 10^{-5}$	1.49	361	$1.46 \cdot 10^{-6}$	2.00

ments, we expect that the velocity and pressure errors decay as h^{k+1} and h^k, respectively, where h is size of the elements of the mesh. Indeed, we show this experimentally in the left-most columns of Table 2 for $Q_2 \times P_{-1}$ (in the top rows), and for $Q_3 \times Q_2$ (in the bottom rows). These results – as well as those in the remainder of the paper – omit data points where the error is less than approximately 10^{-12}, since at that point round-off errors, ill-conditioning of the linear systems, and the finite tolerance of iterative solvers begin to dominate the overall error.

To investigate the case where the viscosity and density are not obtained from an exactly prescribed function, but instead sampled from nearby particles, let us again solve the same problem but use coefficients for the finite element discretization that at every quadrature point are obtained from particles in one of the two ways discussed in Section 3 (namely, cellwise constant averaging and a least squares projection onto a bilinear function).

The corresponding convergence orders for the velocity and pressure errors are shown the second and third set of columns in Table 2. For these results, we use between 4 and 361 particles per cell (PPC), distributed on a regular, equidistant grid. For models that show variable results with increasing PPC we always choose the smallest number of particles that reaches the largest possible convergence rate.

Table 2 shows that a cellwise arithmetic average interpolation for the $Q_k \times P_{-(k-1)}$ element reduces the convergence of the velocity error to second order, independent of the number of particles per cell. We have verified that this remains so if the number of particles per cell were larger than the one used in the table, and consequently only show results for a small number of particles per cell. In other words using a cell-wise constant averaging is suboptimal by one order no matter how many PPC are generated. Using a bilinear least-squares interpolation shows an interesting behavior that was briefly observed, but not fully explored before [Thielmann et al.(2014)]: At low resolutions and for a constant number of particles per cell the velocity error decreases with nearly the expected rate of the direct method, but then quickly degrades to second order convergence (not shown in the table, see Figure 1 and compare also Figure 6b of [Thielmann et al.(2014)]). However, we here show that increasing the number of particles per cell approximately linearly with increasing resolution recovers the expected convergence rate of the Stokes element (last set of columns in Table 2 and Figure 1, top). This is a behavior that to our knowledge has not been described in geodynamic benchmark results before, and is worth exploring in more detail. The pressure error for the $Q_k \times P_{-(k-1)}$ element shown in Table 2 behaves as expected, it is suboptimal by one order for the arithmetic averaging and is identical to the direct method for the bilinear least squares interpolation, both results are independent of PPC (not shown in the table).

Recomputing the results above for the $Q_3 \times Q_2$ Stokes element reveals some similarities, but also noteworthy variations. For the velocity, the direct method decreases with the expected fourth order. The arithmetic average interpolation method again achieves second order accuracy, which for this element is sub-optimal by two orders. The bilinear least-squares interpolation results in second order convergence with constant PPC (not shown in Table 2, but shown in Figure 1, bottom), and third order convergence with increasing PPC. However, it is impossible to recover the expected fourth order convergence rate with increasing PPC, except at very coarse resolutions (rightmost data points in Figure 1, bottom). This will be a significant difference of the SolKz benchmark to the benchmarks we will present in Section 4.2 and Section 5. The pressure error converges with the expected third order rate for the direct method, a second order rate for the bilinear least-squares interpolation – independent of PPC (not shown in the table) – and a rate of 1.5 for cell-wise constant averaging.

To further clarify the effect of the number of particles per cell on the convergence rate, Fig. 1 shows convergence data for the velocity error $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2}$ as a function of both the mesh size and the number of particles per cell. The plots show that the optimal convergence order can indeed be recovered for the $Q_2 \times P_{-1}$ – but not the $Q_3 \times Q_2$ – element, if one uses sufficiently many particles per cell. For both elements, the error is well described by the approximation $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2} = \mathcal{O}(h^3) + \mathcal{O}(h^2 PPC^{-1})$.

As a consequence, for a fixed number of particles per cell – i.e., the only case that can be considered scalable to large problems with fine meshes –, both elements only yield an asymptotic convergence rate of $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2} = \mathcal{O}(h^2)$. In addition, it is worth mentioning that using 196, 361, or even 4,096 particles per cell would make particle advection in time dependent problems far more expensive than solving the Stokes equation, and that using the corresponding $14^3 = 2,744$, $19^3 = 6,859$ or even $64^3 = 262,144$ particles per cell in three space dimensions is not a realistic option. Consequently, unless additional measures are taken (e.g. a higher order particle interpolation scheme) any practical use of particle methods combined with higher order finite-elements will necessarily be prohibitively expensive, or suffer from a sub-optimal convergence rate.

We here do not show corresponding figures for convergence data for the pressure error. This is because for a bilinear reconstruction, the pressure converges at a fixed rate and is essentially independent of the number of particles per cell. Increasing the number of particles therefore does not increase the accuracy of the pressure, unlike for the velocity.

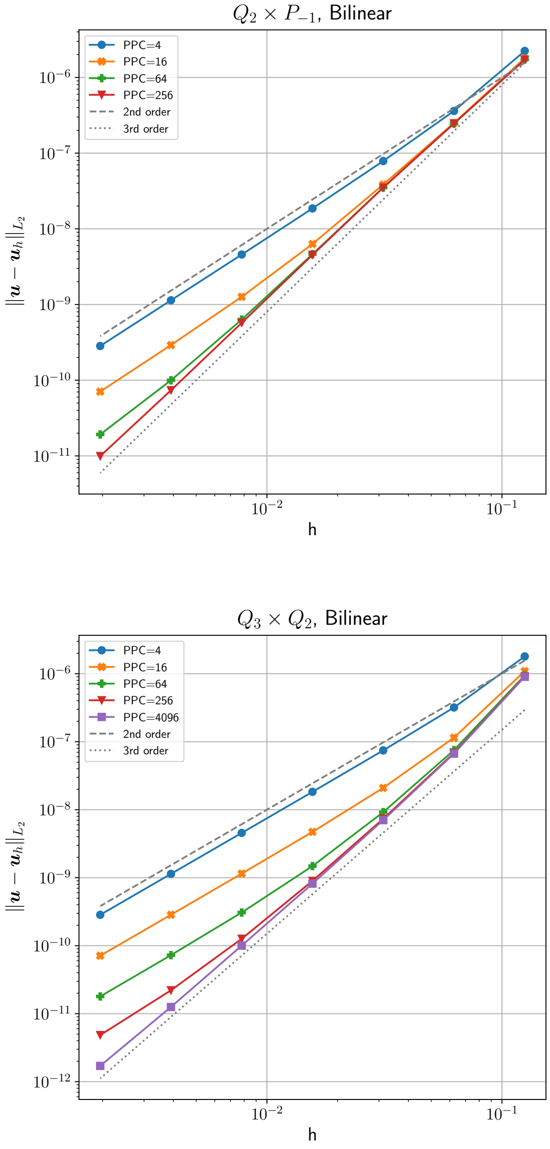

Figure 1. Velocity errors $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2}$ for the SolKz benchmark for the $Q_2 \times P_{-1}$ element (top) and for the $Q_3 \times Q_2$ element (bottom). The error is plotted as a function of both mesh size and number of particles per cell.

4.2 SolCx

The second instantaneous benchmark we investigate is SolCx, where the viscosity is described by

$$\eta(x, y) = \begin{cases} 1 & \text{if} \quad x < 0.5 \\ 10^6 & \text{if} \quad x \geq 0.5, \end{cases} \tag{10}$$

and the density by

$$\rho(x, y) = -\sin(\pi y)\cos(\pi x), \tag{11}$$

Table 3. Velocity errors $\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2}$ and pressure errors $\|p-p_h\|_{L_2}$ for the SolCx benchmark using the $Q_2 \times P_{-1}$ Stokes element (top rows), and the $Q_3 \times Q_2$ Stokes element (bottom rows). ("PPC" stands for particles per cell.)

$Q_2 \times P_{-1}$

$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2}$	direct method			arithmetic average			bilinear least squares	
h	error	rate	PPC	error	rate	PPC	error	rate
$\frac{1}{8}$	$1.32 \cdot 10^{-5}$	-	4	$3.16 \cdot 10^{-5}$	-	4	$1.36 \cdot 10^{-5}$	-
$\frac{1}{16}$	$1.66 \cdot 10^{-6}$	2.99	4	$7.30 \cdot 10^{-6}$	2.12	4	$1.93 \cdot 10^{-6}$	2.81
$\frac{1}{32}$	$2.08 \cdot 10^{-7}$	3.00	4	$1.79 \cdot 10^{-6}$	2.03	9	$2.36 \cdot 10^{-7}$	3.03
$\frac{1}{64}$	$2.60 \cdot 10^{-8}$	3.00	4	$4.44 \cdot 10^{-7}$	2.01	25	$2.79 \cdot 10^{-8}$	3.08
$\frac{1}{128}$	$3.26 \cdot 10^{-9}$	3.00	4	$1.11 \cdot 10^{-7}$	2.00	49	$3.50 \cdot 10^{-9}$	3.00
$\frac{1}{256}$	$4.08 \cdot 10^{-10}$	3.00	4	$2.77 \cdot 10^{-8}$	2.00	100	$4.39 \cdot 10^{-10}$	3.00
$\frac{1}{512}$	$5.13 \cdot 10^{-11}$	3.00	4	$6.92 \cdot 10^{-9}$	2.00	196	$5.87 \cdot 10^{-11}$	2.90
$\|p-p_h\|_{L_2}$								
$\frac{1}{8}$	$1.48 \cdot 10^{-3}$	-	4	$3.16 \cdot 10^{-3}$	-	4	$1.53 \cdot 10^{-3}$	-
$\frac{1}{16}$	$3.7 \cdot 10^{-4}$	2.00	4	$8.00 \cdot 10^{-4}$	1.99	4	$3.83 \cdot 10^{-4}$	2.00
$\frac{1}{32}$	$9.22 \cdot 10^{-5}$	2.00	4	$2.00 \cdot 10^{-4}$	2.00	9	$9.29 \cdot 10^{-5}$	2.05
$\frac{1}{64}$	$2.30 \cdot 10^{-5}$	2.00	4	$5.00 \cdot 10^{-5}$	2.00	25	$2.30 \cdot 10^{-5}$	2.01
$\frac{1}{128}$	$5.75 \cdot 10^{-6}$	2.00	4	$1.25 \cdot 10^{-5}$	2.00	49	$5.75 \cdot 10^{-6}$	2.00
$\frac{1}{256}$	$1.44 \cdot 10^{-6}$	2.00	4	$3.12 \cdot 10^{-6}$	2.00	100	$1.44 \cdot 10^{-6}$	2.00
$\frac{1}{512}$	$3.59 \cdot 10^{-7}$	2.00	4	$7.80 \cdot 10^{-7}$	2.00	196	$3.59 \cdot 10^{-7}$	2.00

$Q_3 \times Q_2$

$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2}$	direct method			arithmetic average			bilinear least squares	
h	error	rate	PPC	error	rate	PPC	error	rate
$\frac{1}{8}$	$6.04 \cdot 10^{-7}$	-	4	$3.15 \cdot 10^{-5}$	-	100	$9.10 \cdot 10^{-7}$	-
$\frac{1}{16}$	$4.03 \cdot 10^{-8}$	3.90	4	$7.29 \cdot 10^{-6}$	2.11	400	$5.84 \cdot 10^{-8}$	3.96
$\frac{1}{32}$	$2.60 \cdot 10^{-9}$	4.00	4	$1.79 \cdot 10^{-6}$	2.03	1600	$3.70 \cdot 10^{-9}$	3.98
$\frac{1}{64}$	$1.67 \cdot 10^{-10}$	4.00	4	$4.44 \cdot 10^{-7}$	2.01	6400	$2.34 \cdot 10^{-10}$	3.97
$\frac{1}{128}$	$1.98 \cdot 10^{-11}$	3.10	4	$1.11 \cdot 10^{-7}$	2.00	25600	$1.93 \cdot 10^{-11}$	3.60
$\frac{1}{256}$			4	$2.77 \cdot 10^{-8}$	2.00			
$\|p-p_h\|_{L_2}$								
$\frac{1}{8}$	$8.81 \cdot 10^{-3}$	-	4	$8.87 \cdot 10^{-3}$	-	100	$8.89 \cdot 10^{-3}$	-
$\frac{1}{16}$	$6.22 \cdot 10^{-3}$	0.50	4	$6.18 \cdot 10^{-3}$	0.52	400	$6.22 \cdot 10^{-3}$	0.51
$\frac{1}{32}$	$4.39 \cdot 10^{-3}$	0.50	4	$4.38 \cdot 10^{-3}$	0.50	1600	$4.39 \cdot 10^{-3}$	0.50
$\frac{1}{64}$	$3.1 \cdot 10^{-3}$	0.50	4	$3.10 \cdot 10^{-3}$	0.50	6400	$3.1 \cdot 10^{-3}$	0.50
$\frac{1}{128}$	$2.19 \cdot 10^{-3}$	0.50	4	$2.19 \cdot 10^{-3}$	0.50	25600	$2.19 \cdot 10^{-3}$	0.50
$\frac{1}{256}$			4	$1.55 \cdot 10^{-3}$	0.50			

all again on the unit square $\Omega = (0,1)^2$. The complete derivation of the analytical solution uses a propagator matrix method, and is described in [Zhong(1996)]. The defining property of this benchmark is that the discontinuous viscosity implies a nearly discontinuous pressure field and a velocity field that has a kink. Consequently, we can generally not expect optimal convergence rates unless (i) the mesh is aligned with the discontinuity and (ii) we use a pressure finite element that is discontinuous. While these properties reduce the usefulness of the benchmark for general problems, it is useful for our investigation for an unrelated reason: While the density of the benchmark problem can only be approximated with the expected accuracy of the particle interpolation methods mentioned in Section 3.4 (namely $O(h)$ for arithmetic averaging and $O(h^2)$ for the bilinear least squares method), the viscosity is cell-wise constant, for a mesh that is aligned with the interface, such as ours. The viscosity can therefore be interpolated exactly from particles to cells independent of the interpolation method. This allows us to separate influences from density and viscosity errors on the pressure and velocity solution.

Table 3 demonstrates convergence of the velocity and pressure for the $Q_2 \times P_{-1}$ element (top rows), and the $Q_3 \times Q_2$ element (bottom rows).

Starting with the $Q_2 \times P_{-1}$ element and the direct method, the velocity error decreases with $O(h^3)$ and the pressure error with $O(h^2)$ just as expected and as reported previously [Kronbichler et al.(2012)], although half an order higher than reported elsewhere [Thielmann et al.(2014)]. Using particles and arithmetic averaging with a constant PPC (the result is independent of PPC) achieves second order convergence for velocity and pressure, i.e. suboptimal by one order in the velocity; this is consistent with previous results [Thielmann et al.(2014)]. Using the bilinear least squares interpolation mirrors the results described in Section 4.1, the velocity error decays as $O(h^2)$ for constant PPC and $O(h^3)$ if PPC increases with h^{-1}. The pressure error is unremarkable and reaches the design error convergence rate

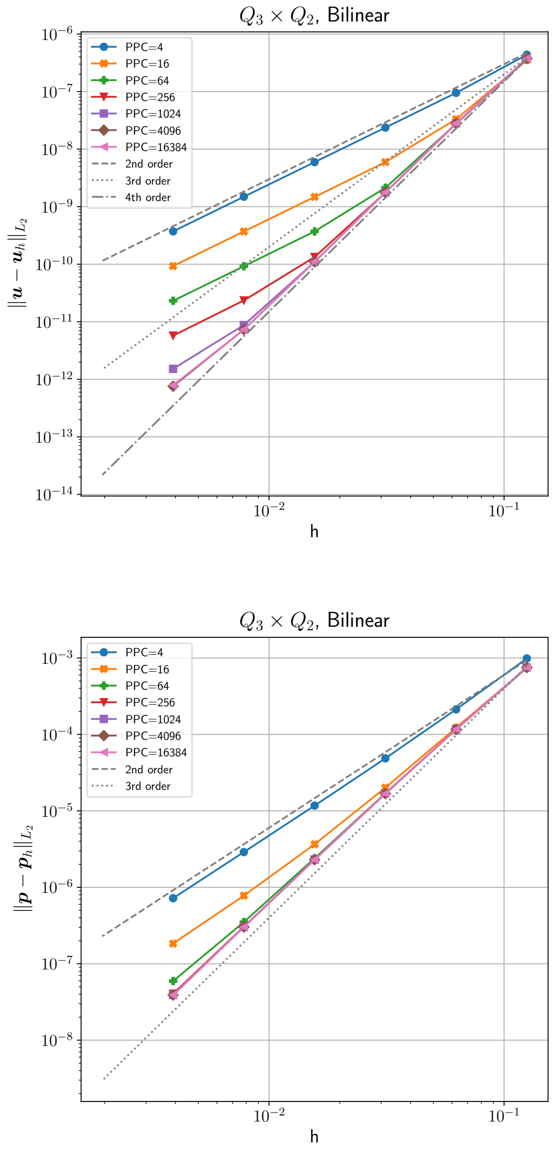

Figure 2. Velocity errors $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2}$ (top) and pressure errors $\|p - p_h\|_{L_2}$ (bottom) for the SolKz benchmark for the $Q_3 \times Q_2$ element. The error is plotted as a function of both mesh size and number of particles per cell. For this model, we interpolate only density from particles (that is, we use the exact viscosity in the assembly of the finite element linear system), and we recover 4th order convergence rate in velocity and 3rd order in pressure.

of 2 for all methods. So far the SolCx benchmark behaves exactly like SolKz, and the exact viscosity interpolation did not influence the convergence rates of the solution.

As described before [Kronbichler et al.(2012), Thielmann et al.(2014)], using a continuous pressure element like $Q_3 \times Q_2$ in general destroys the optimal convergence rates of the pressure error, because of the discontinuity in the pressure solution, and all methods (independently of PPC choice) now only show a pressure convergence rate of $O(h^{1/2})$. Nevertheless, as expected for this benchmark despite the suboptimal pressure solution, the velocity error is still able to converge with the expected rates for the direct method ($O(h^4)$) and the arithmetic averaging ($O(h^2)$, independent of PPC). Using the bilinear least-squares method however not only allows convergence of the velocity error of $O(h^2)$ for a constant PPC, and $O(h^3)$ for a PPC that scales with h^{-1} as for the $Q_2 \times P_{-1}$ element, but also $O(h^4)$ for a PPC that increases with h^{-2}. In other words, we can recover the expected velocity convergence order of the $Q_3 \times Q_2$ element if the number of particles per cell increases quadratically with increasing cell size, a result that is not reproducable for the SolKz benchmark with identical numerical methods. We therefore infer that if the error of the density interpolation converges with $O(h^k)$ by choosing PPC in a way that facilitates this, then the velocity error can converge with at least up to $O(h^{k+2})$. However, if the particles are also used to interpolate the viscosity (as for the SolKz benchmark), the velocity error converges with at most $O(h^{k+1})$.

To test this hypothesis we repeat the SolKz benchmark with a density that is interpolated from particles, and a prescribed viscosity (i.e., using the particles for density, but the direct method for viscosity). The $Q_2 \times P_{-1}$ element shows no difference to the computations with interpolated viscosity, as they already reached the designed convergence order. However, the $Q_3 \times Q_2$ element now also reaches the designed convergence order for velocity (namely, 4) and pressure (i.e., 3), as can be seen in Fig. 2. Moreover, we now also require an increase of PPC with h that goes as h^{-2}.

In summary, these experiments show the importance of the choice of PPC and particle interpolation method in practical applications, and that their optimal choices differ depending on whether the particles only carry density, or also viscosity information. In particular, we may need to increase the number of particles per cell as $O(h^{-1})$ or even $O(h^{-2})$ to retain the convergence order of the finite element scheme if the particle interpolation scheme converges with a lower order than the Stokes discretization. This requires choosing between one of three options: (i) One needs to use a potentially very large number of particles per cell to retain the accuracy of the Stokes discretization, in particular if high accuracy is required or the computations are in three space dimensions. This may be prohibitively expensive, however: for example, in the $Q_3 \times Q_2$ solution of the SolCx case with $h = \frac{1}{128}$ and $PPC = 25,600$ (see Table 3), all particle operations associated with the one time step we solve account for some 95% of the overall run time. (ii) One accepts the loss of accuracy by using too few particles per cell, although that then calls into question the use of higher order polynomial spaces in the Stokes discretization. (iii) One uses methods with higher accuracy to project properties from particle locations to fields. An alternative is to use field-based – instead of a particle-based – descriptions of the temperature, chemical composition, or other advected quantities as discussed in [Kronbichler et al.(2012)]; in that case, the effort for the Stokes solve and the advection solve is automatically balanced.

5 TIME DEPENDENT BENCHMARKS

The previous section presented benchmarks that evaluate the transfer of information from (stationary) particle locations back to the finite element mesh, along with the error that was introduced by this operation. On the other hand, in realistic applications, particles will be advected along, and consequently the overall error will contain contributions that are due to the transfer of particle information to the mesh, but also due to the fact that we only know particle locations up to the numerical error introduced in the integration of particle trajectories, as discussed in Section 3. We will here numerically test how large this overall error is, and what effect it has on the numerical solution of the Stokes equation when feeding information back to the Stokes solver.

To this end, we derive a time-independent solution to the Stokes equations (1)–(2) in an annulus in which the exact density ρ is constant on streamlines. As we noted before a spatially varying viscosity would limit the convergence rate we could achieve with our interpolation methods, and might obscure the error of the particle advection method, therefore we choose a constant viscosity. If one were to solve the Stokes equations with this setup, the solution would of course not change with time. However, if the density (as part of the right hand side) is interpolated from particles in each time step, and particles are advected along with the computed velocity, then the numerical solution *will* change with time, and we can assess the accuracy of the particle-in-cell algorithm using the difference between exact (time independent) and computed (time dependent) solution. In our experiments, we will evaluate this numerical error for different values of the (largest) grid size $h_{\max}$ and different numbers of particles per cell (PPC).

5.1 A time dependent benchmark in an annulus

For the concrete realization of the approach outlined above, we need to construct a testcase with a steady-state velocity field that depends on a spatially non-constant density that we can advect along either as a field or with particles. To make the situation not too trivial, we will choose the domain as a two-dimensional annulus with inner and outer radii $R_1 = 1$ and $R_2 = 2$, respectively.

In this situation, we can express the equations and the solution in a cylindrical coordinate system in terms of the radius r and the azimuthal angle θ. A solution of equations (1)–(2) can then be obtained by setting

$$\eta = 1, \qquad \rho(r,\theta) = 48r^5, \qquad \boldsymbol{g} = \frac{r^3}{384}\mathbf{e}_r + \mathbf{e}_\theta, \qquad (12)$$

where $\mathbf{e}_r$ and $\mathbf{e}_\theta$ are the radial and azimuthal unit vectors, respectively. Such a gravity vector is not the gradient of a gravity potential and consequently not physical, but this is of no importance here. The Stokes system can then be solved using a separation of variables approach and yields

$$\boldsymbol{u}(r,\theta) = 0\mathbf{e}_r - r^7\mathbf{e}_\theta, \qquad p(r,\theta) = \frac{r^9}{72} - \frac{512}{72}, \quad (13)$$

for the velocity and pressure. In other words, the flow field is circular around the center with a velocity that varies with radius. The friction induced by the variable velocity is countered by the tangential component of the gravity vector. Importantly, while all solution fields in question are polynomials in r and θ, their degrees are sufficiently high so as to not be in the finite element spaces we use. The benchmark is then completely defined by prescribing η and $\boldsymbol{g}$ as above, along with prescribed tangential velocity boundary values on the inner and outer boundaries of the annulus, and the initial distribution of ρ. Note, that while it seems unintuitive for a gravity in e_θ direction to cause a flow in $-e_\theta$ direction, one can think of this flow as being driven by the prescribed tangential velocity at the outer boundary, which is gradually reduced by the gravity with decreasing radius. A detailed derivation and visualization of this solution can be found in Appendix A and Fig. A1.

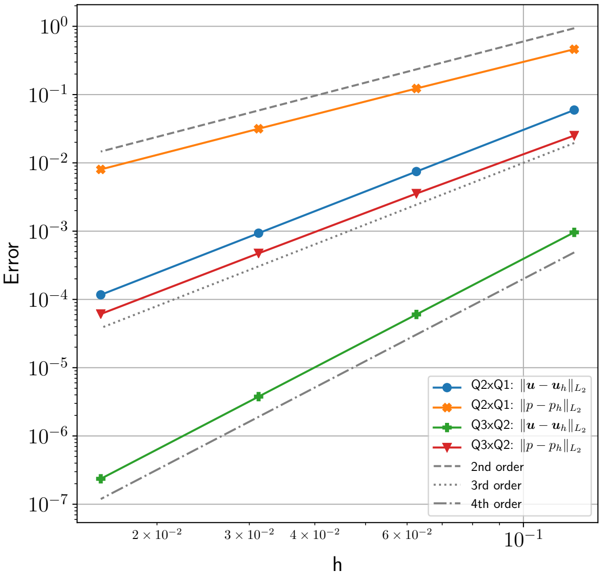

Figure 3. Convergence rates for the velocity $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2}$ and pressure $\|p - p_h\|_{L_2}$ for the time-dependent benchmark on the annulus using $Q_2 \times Q_1$ and $Q_3 \times Q_2$ element combinations, respectively. The results shown here use the exact density.

All experiments in this section show the error between the (stationary) exact solution $\boldsymbol{u}$, p, and ρ and the (time-dependent) numerical approximation $\boldsymbol{u}_h$, p_h, and ρ_h at time $t = \frac{4\pi}{2^7} \approx 0.0982$, which equals two complete revolutions of particles on the outer edge $r = R_2$.

5.2 Convergence results

If we use the exact (and unchanging) density when computing the numerical solution of the Stokes equation, one expects convergence to the exact solution with an appropriate power of the mesh size. We verify that our solver achieves the expected convergence orders in Figure 3 for both $Q_2 \times Q_1$ and $Q_3 \times Q_2$ elements.

On the other hand, if the density in each time step is interpolated from particles to quadrature points, then the solution will vary from time step to time step due to the fact that particle locations are advected along with the numerical approximation of the velocity field $\boldsymbol{u}$.

Figure 4 shows convergence results for the $Q_2 \times Q_1$ element for velocity, pressure, and density. As was shown in the instantaneous benchmarks above (Section 4), the order of convergence of the velocity and pressure error directly depends on the averaging scheme, which also determines the convergence order for the error in density. Calculating the

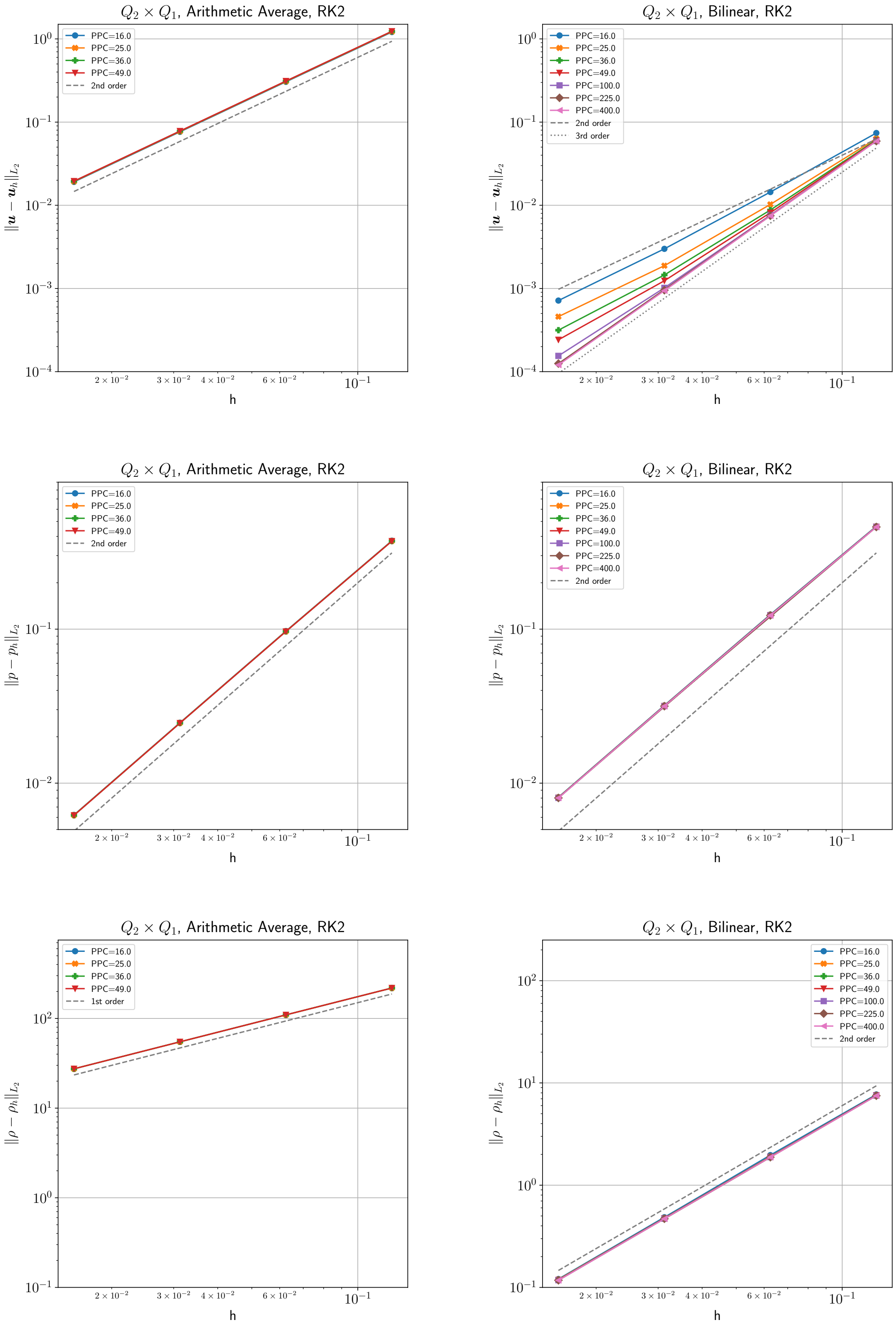

Figure 4. The convergence rate of $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2}$ (top), $\|p - p_h\|_{L_2}$ (middle), and $\|\rho - \rho_h\|_{L_2}$ (bottom) measured at $t = 4\pi/2^7$ for the time dependent benchmark. Density is carried on particles and is interpolated as cell-wise arithmetic average (left) and bilinear least-squares interpolation (right). All models use a $Q_2 \times Q_1$ element and RK2 to advect particles. Colored dots represent models with varying numbers of PPC. Note that only with bilinear least-squares interpolation and an increasing number of PPC the third order convergence rate of velocity is recovered. In all cases, $\|p - p_h\|_{L_2}$ converges at second-order rate with no apparent influence due to the number of PPC (i.e., all dots fall on each other), while the convergence rate of $\|\rho - \rho_h\|_{L_2}$ depends on the interpolation scheme, but not on PPC.

density as a cell-wise arithmetic average of particle densities yields a first-order convergence in density, second-order in pressure, and second-order in velocity. All errors are independent of the number of particles per cell (in the tested range). This means while the pressure solution reaches its designed convergence order, the velocity solution loses one order of convergence as seen in the instantaneous benchmarks. Using a bilinear least-squares fit significantly reduces the error in density and converges with a second-order rate, while the pressure error remains unaffected. Both errors are independent of PPC. Also, the error in velocity is significantly reduced, but remains at a second-order convergence rate for a constant PPC, unless one increases the number of particles per cell as h goes to zero. In other words the designed convergence rate is only recovered if the number of particles per cell is increased when the mesh is refined and the number of cells increased. This observation is consistent with our instantaneous benchmarks above, and the observation in [Thielmann et al.(2014)] that the convergence rate is reduced for constant PPC. All of these results are identical for the RK2 and RK4, advection schemes, which is why we only present the RK2 results.

Figure 5 shows the corresponding results for the $Q_3 \times Q_2$ element combination, and offers some interesting differences. For lack of any new information we omit the arithmetic averaging case and instead compare the RK2 integration scheme to the RK4 integrator. Starting with the density error we point out that the integration scheme (RK2 vs. RK4), the PPC (16 to 6400) and the finite element ($Q_2 \times Q_1$ vs. $Q_3 \times Q_2$) do not change the convergence rate of the density, it remains second-order accurate. However pressure and velocity show significant differences. The pressure errors for RK2 and RK4 integration schemes are essentially identical to each other and show a second-order convergence rate for a constant PPC and a third-order convergence rate (i.e., the design rate of the Q_2 finite element) for an increasing number of PPC. In other words the pressure error of these models shows the same dependence on PPC as the velocity error for the $Q_2 \times Q_1$ element. In combination this leads to the following hypothesis: *Whenever the design convergence rate of a finite element surpasses the convergence rate of the particle interpolation method of a particle-in-cell method it requires an increasing number of PPC to recover the design convergence rate of the solution.*

Investigating the velocity error of the $Q_3 \times Q_2$ element introduces another factor: When using a RK2 integration scheme (top left) in combination with this element the velocity error remains at the same second order convergence for constant PPC and third order convergence for increasing PPC, independent of the finite element degree (even though the errors are smaller than for the $Q_2 \times Q_1$ element). However, when using an RK4 integrator the convergence order for a constant PPC remains the same (second order), but the convergence rate for increasing PPC recovers the fourth order design rate of the Q_3 element.

Consequently the hypothesis has to be modified to: *Whenever the design convergence rate of a finite element surpasses the convergence rate of the particle interpolation method of a particle-in-cell method it requires an increasing number of PPC to recover the design convergence rate of the solution. However the convergence rate of the particle advection method creates an upper bound for the convergence rate that can be recovered.*

Figure 6 plots selected information from the two previous figures as velocity error over number of PPC for different finite elements, particle interpolation schemes, and mesh resolutions, which allows determining the influence of the particular choice of PPC on the error. In general all of the computations we made show a linear decrease of velocity error with increasing PPC, which eventually transitions into a constant error at a model-specific number of PPC when the finite element error begins to dominate the error of the particle interpolation. The number of PPC at which the transition occurs can be interpreted as *optimal*, in the sense that it recovers the design rate of the finite element with the minimum number of particles. As one can see from this figure the optimal number of PPC is dependent on the finite element type and in the case of the $Q_3 \times Q_2$ element also the particle integrator and in all configurations the mesh size. Most likely it will also depend on the problem one is solving as well. Therefore, the optimal number of PPC can not be accurately determined for practical applications except by performing a convergence series test with increasing PPC for the specific problem at the final resolution. Nevertheless, we hypothesize that the optimal values of PPC that we have found in this work will be close to optimal values for a variety of smooth problems.

6 CONCLUSIONS

In this manuscript we have developed new benchmarks to measure the accuracy and convergence rate of hybrid finite element/particle-in-cell methods and provided reference results for these benchmarks obtained with the geodynamic modeling code ASPECT. In particular, we have presented the first analytical benchmark that measures the accuracy and convergence order of a time-dependent flow problem in a 2D spherical annulus using particles to carry material properties. Since the benchmark is simple to derive and implement, it can be used as a convenient measure for the correctness of future implementaions of similar algorithms, or as a common model for code comparisons.

We have also investigated the influence of different interpolation algorithms for transfering information from the particles to the cells and determine that in order to retain the design convergence rate of high-order finite element formulations one needs to use a high-order particle interpolation algorithm. Additionally, we have observed that if the convergence rate of the particle interpolation algorithm is lower than the expected convergence rate of the velocity or pressure solution, one needs to increase the number of particles per cell along with increasing the underlying grid resolution in order to avoid a loss of convergence rate in the numerical solution. While the exact number of particles per cell necessary to achieve the designed convergence rate may be problem-dependent, the fact that it is resolution dependent to begin with raises the question of the scalability of the method, since either a loss of convergence rate (e.g., with a constant number of particles per cell) needs to be accepted or the number of particles will need to increase substantially faster than the number of cells. A remedy to this problem could be the implementation of higher order

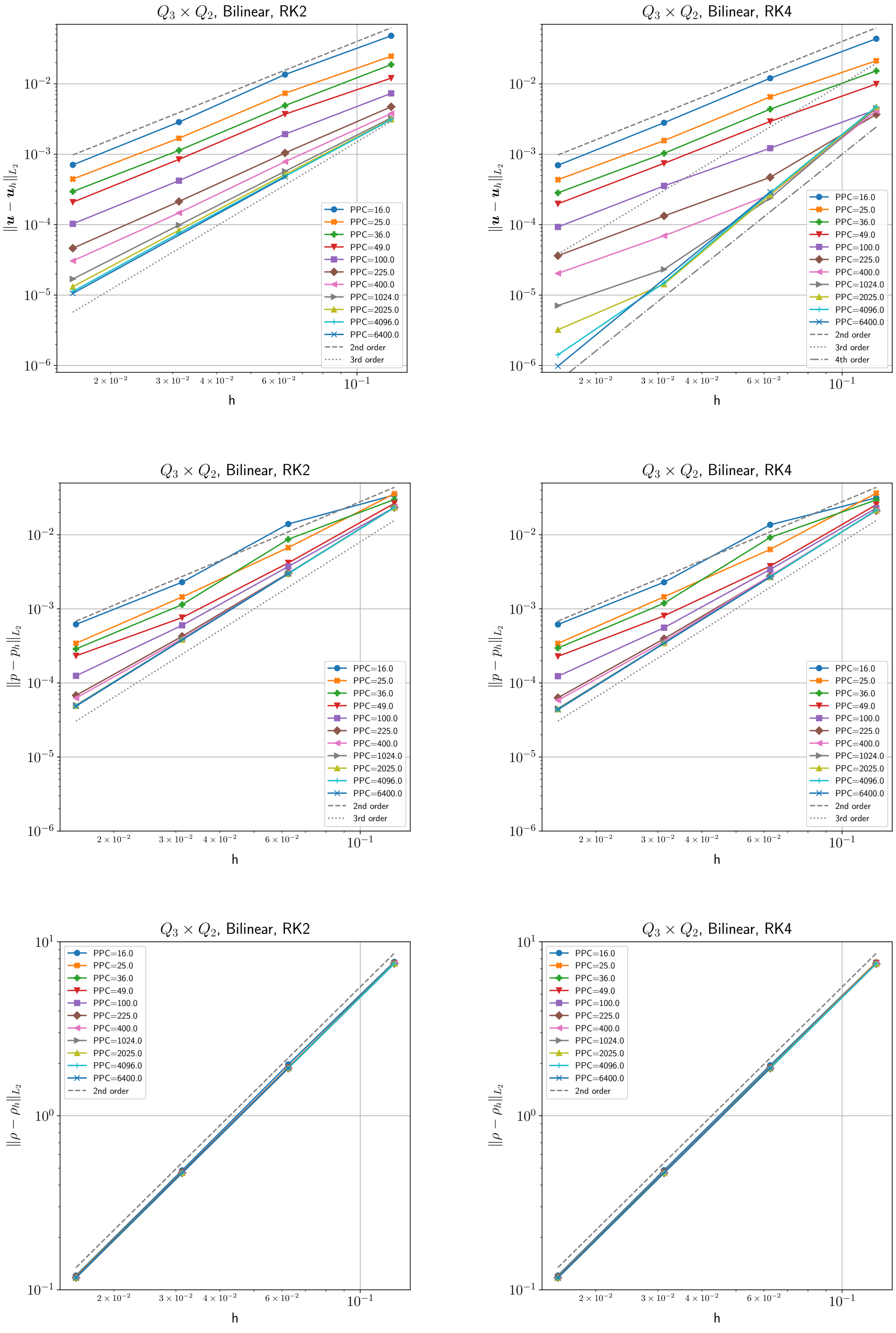

Figure 5. Panels as in Fig. 4, but for a $Q_3 \times Q_2$ element. All models use the bilinear least squares interpolation. Columns represent RK2 (left) and RK4 (right) particle integration. Note that only with RK4, bilinear least-squares interpolation and an increasing number of PPC is the fourth order convergence rate of the velocity recovered. All properties with a design convergence rate higher than 2 require an increasing PPC to reach their design rate, while constant PPC only allows for second order convergence. The density is limited to second-order accuracy due to the interpolation scheme we have chosen.

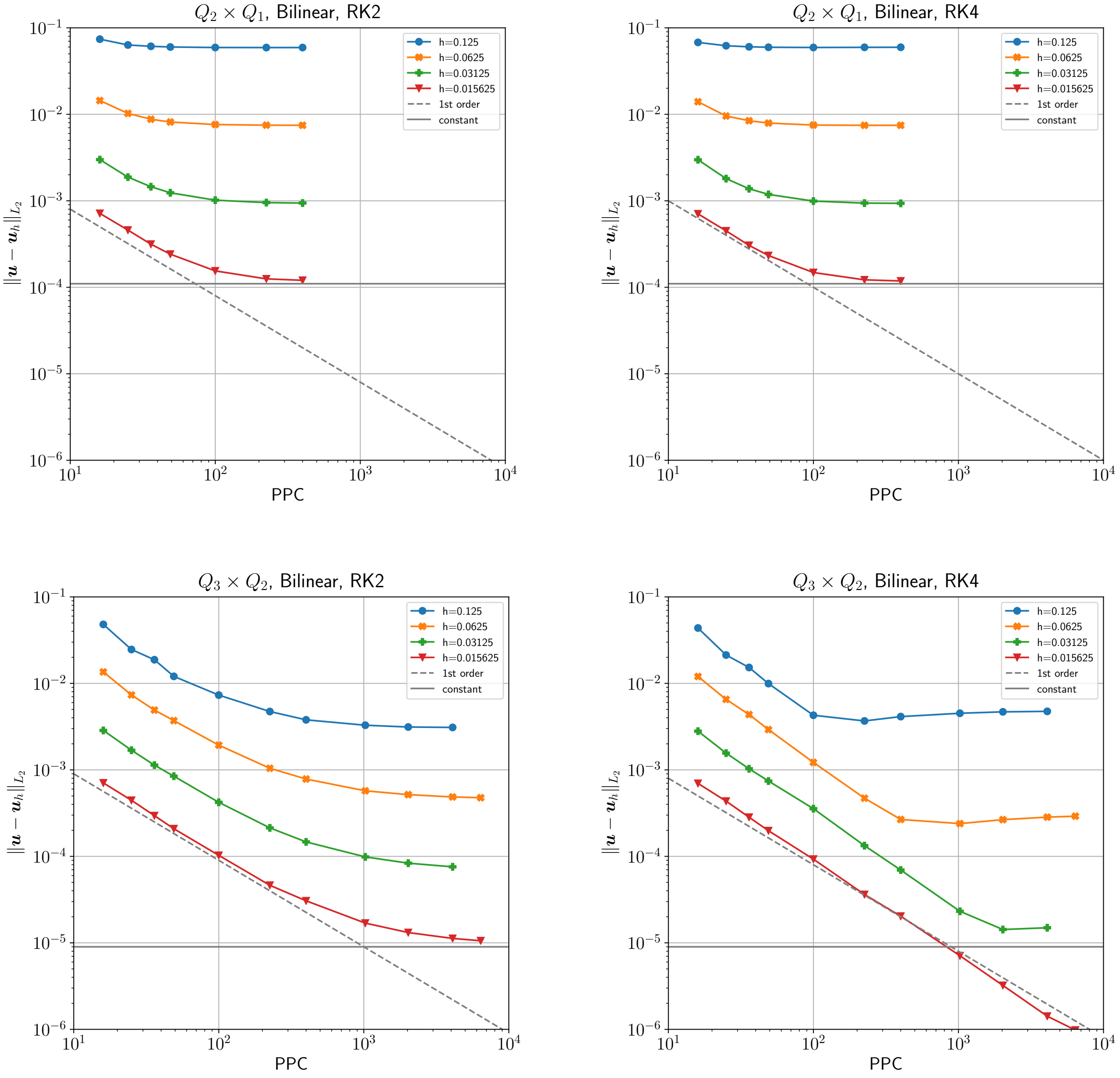

Figure 6. Convergence plots for velocity in the L_2 norm for the time dependent annulus benchmark in dependence of PPC. Models were computed using a $Q_2 \times Q_1$ finite element (top) and $Q_3 \times Q_2$ element (bottom) respectively and particles were advected using a RK2 integration scheme (left) and RK4 integration scheme (right). Colored dots represent models of different cell sizes. Note that the required PPC to reach the minimum error for a given mesh refinement depends on the finite element and the mesh refinement itself. The integration scheme only plays a role if its convergence rate is lower than the convergence rate of the velocity element.

interpolation algorithms with an expected convergence rate of at least that of the velocity. We will leave this to future work.

ACKNOWLEDGMENTS

All models were computed with the open source software ASPECT [Bangerth et al.(2018), `https://aspect.geodynamics.org`] published under the GPL2 license, and the necessary data to reproduce the models is included in the software. We thank the Computational Infrastructure for Geodynamics (http://geodynamics.org) – which is funded by the National Science Foundation under awards EAR-0949446 and EAR-1550901 – for supporting the development of ASPECT.

R. Gassmöller and W. Bangerth were partially supported by the National Science Foundation under award OCI-1148116 as part of the Software Infrastructure for Sustained Innovation (SI2) program; and by the Computational Infrastructure in Geodynamics initiative (CIG), through the National Science Foundation under Awards No. EAR-0949446, EAR-1550901 and The University of California – Davis. W. Bangerth was also supported by the National Science Foundation under Award OAC-1835673.

E. G. Puckett was supported by the National Science

Foundation under Award ACI-1440811 as part of the SI2 Scientific Software Elements (SSE) program.

The computational resources were provided by the Computational Infrastructure for Geodynamics on the Stampede2 system at the Texas Advanced Computing Centre (TACC).

REFERENCES

Bangerth, W., Dannberg, J., Gassmöller, R., Heister, T., et al., 2018. ASPECT: Advanced Solver for Problems in Earth's ConvecTion, User Manual, doi:10.6084/m9.figshare.4865333.

Bercovier, M. & Pironneau, O., 1979. Error estimates for finite element method solution of the stokes problem in the primitive variables, *Numerische Mathematik*, **33**(2), 211–224.

Brooks, A. N. & Hughes, T. J., 1982. Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations, *Computer methods in applied mechanics and engineering*, **32**(1-3), 199–259.

Dannberg, J. & Gassmöller, R., 2018. Chemical trends in ocean islands explained by plume–slab interaction, *Proceedings of the National Academy of Sciences*, **115**(17), 4351–4356.

Dannberg, J., Eilon, Z., Faul, U., Gassmöller, R., Moulik, P., & Myhill, R., 2017. The importance of grain size to mantle dynamics and seismological observations, *Geochemistry, Geophysics, Geosystems*, **18**(8), 3034–3061.

Deubelbeiss, Y. & Kaus, B., 2008. Comparison of eulerian and lagrangian numerical techniques for the stokes equations in the presence of strongly varying viscosity, *Physics of the Earth and Planetary Interiors*, **171**(1-4), 92–111.

Donea, J. & Huerta, A., 2003. *Finite element methods for flow problems*, John Wiley & Sons.

Duretz, T., May, D. A., Gerya, T. V., & Tackley, P. J., 2011. 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), Q07004.

Evans, M. W., Harlow, F. H., & Bromberg, E., 1957. The particle-in-cell method for hydrodynamic calculations, Tech. rep., LOS ALAMOS NATIONAL LAB NM.

Fischer, R. & Gerya, T., 2016. Early earth plume-lid tectonics: A high-resolution 3d numerical modelling approach, *Journal of Geodynamics*, **100**, 198–214.

Gassmöller, R., Dannberg, J., Bredow, E., Steinberger, B., & Torsvik, T. H., 2016. Major influence of plume-ridge interaction, lithosphere thickness variations, and global mantle flow on hotspot volcanism–the example of tristan, *Geochemistry, Geophysics, Geosystems*, **17**(4), 1454–1479.

Gassmöller, R., Lokavarapu, H., Heien, E., Puckett, E. G., & Bangerth, W., 2018. Flexible and scalable particle-in-cell methods with adaptive mesh refinement for geodynamic computations, *Geochemistry, Geophysics, Geosystems*, **19**(9), 3596–3604.

Gerya, T., 2009. *Introduction to numerical geodynamic modelling*, Cambridge University Press.

Gerya, T. V. & Yuen, D. A., 2003. Characteristics-based marker-in-cell method with conservative finite-differences schemes for modeling geological flows with strongly variable transport properties, *Physics of the Earth and Planetary Interiors*, **140**(4), 293–318.

Guermond, J.-L. & Pasquetti, R., 2011. Entropy viscosity method for high-order approximations of conservation laws, in Spectral and high order methods for partial differential equations, pp. 411–418, Springer.

Harlow, F. H. & Welch, J. E., 1965. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *The Physics of Fluids*, **8**(12), 2182–2189.

Heister, T., Dannberg, J., Gassmöller, R., & Bangerth, W., 2017. High accuracy mantle convection simulation through modern numerical methods – II: Realistic models and problems, *Geophys. J. Int.*, **210**(2), 833–851.

Hirt, C. W. & Nichols, B. D., 1981. Volume of fluid (VOF) method for the dynamics of free boundaries, *Journal of Computational Physics*, **39**(1), 201–225.

Ismail-Zadeh, A. & Tackley, P., 2010. *Computational methods for geodynamics*, Cambridge University Press.

Kronbichler, M., Heister, T., & Bangerth, W., 2012. High accuracy mantle convection simulation through modern numerical methods, *Geophysical Journal International*, **191**(1), 12–29.

McNamara, A. K. & Zhong, S., 2004. Thermochemical structures within a spherical mantle: Superplumes or piles?, *Journal of Geophysical Research*, **109**(B7), 1–14.

McNamara, A. K. & Zhong, S., 2005. Thermochemical structures beneath Africa and the Pacific Ocean, *Nature*, **437**(7062), 1136.

Moresi, L., Dufour, F., & Muhlhaus, H. B., 2003. A Lagrangian integration point finite element method for large deformation modeling of viscoelastic geomaterials, *Journal of Computational Physics*, **184**, 476–497.

Mulyukova, E. & Bercovici, D., 2018. Collapse of passive margins by lithospheric damage and plunging grain size, *Earth and Planetary Science Letters*, **484**, 341–352.

Poliakov, A. & Podladchikov, Y., 1992. Diapirism and topography, *Geophysical Journal International*, **109**(3), 553–564.

Popov, A. A. & Sobolev, S. V., 2008. SLIM3D: A tool for three-dimensional thermomechanical modeling of lithospheric deformation with elasto-visco-plastic rheology, *Physics of the Earth and Planetary Interiors*, **171**, 55–75.

Puckett, E. G., Turcotte, D. L., Kellogg, L. H., He, Y., Robey, J. M., & Lokavarapu, H., 2017. New numerical approaches for modeling thermochemical convection in a compositionally stratified fluid, *arXiv preprint arXiv:1703.02202*.

Pusok, A. E., Kaus, B. J., & Popov, A. A., 2017. On the quality of velocity interpolation schemes for marker-in-cell method and staggered grids, *Pure and Applied Geophysics*, **174**(3), 1071–1089.

Revenaugh, J. & Parsons, B., 1987. Dynamic topography and gravity anomalies for fluid layers whose viscosity varies exponentially with depth, *Geophysical Journal International*, **90**(2), 349–368.

Rozel, A., Ricard, Y., & Bercovici, D., 2011. A thermodynamically self-consistent damage equation for grain size evolution during dynamic recrystallization, *Geophysical Journal International*, **184**(2), 719–728.

Schubert, G., Turcotte, D. L., & Olson, P., 2001. *Mantle Convection in the Earth and Planets, Part 1*, Cambridge University Press.

Tackley, P. J., 1998. Thermo-chemical basal boundary layer: D"?, *The core-mantle boundary region*, p. 231.

Tackley, P. J. & King, S. D., 2003. Testing the tracer ratio method for modeling active compositional fields in mantle convection simulations, *Geochemistry, Geophysics, Geosystems*, **4**(4).

Taylor, C. & Hood, P., 1973. A numerical solution of the Navier-Stokes equations using the finite element technique, *Computers & Fluids*, **1**(1), 73–100.

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

Thielmann, M., Rozel, A., Kaus, B., & Ricard, Y., 2015. Intermediate-depth earthquake generation and shear zone formation caused by grain size reduction and shear heating, *Geology*, **43**(9), 791–794.

Wang, H., Agrusta, R., & Hunen, J., 2015. Advantages of a conservative velocity interpolation (CVI) scheme for particle-in-cell methods with application in geodynamic modeling, *Geochemistry, Geophysics, Geosystems*, **16**(6).

Zhong, S., 1996. Analytic solutions for Stokes' flow with lateral variations in viscosity, *Geophysical Journal International*, **124**(1), 18–28.

APPENDIX A: DERIVATION OF AN INCOMPRESSIBLE STOKES SOLUTION ON AN ANNULUS

In order to derive the solution of the Stokes problem discussed in Section 5.1, we consider the Stokes equations (1)–(2) in polar coordinates. Since we will impose Dirichlet boundary conditions along all boundaries, and since we only consider an isoviscous fluid with $\eta = 1$, the equations can be simplified to

$$-\Delta \boldsymbol{u} + \nabla p = \rho\, \boldsymbol{g}, \tag{A.1}$$

$$\nabla \cdot \boldsymbol{u} = 0. \tag{A.2}$$

In a polar coordinate system with $r = \sqrt{x^2+y^2}$ and $\theta = \arctan \frac{y}{x}$, we can express the Laplace operator, gradient, and divergence operators in terms of $\frac{\partial}{\partial r}$ and $\frac{\partial}{\partial \theta}$. The incompressible Stokes equations (A.1) and (A.2) then become

$$-\Big(\frac{\partial^2 u_r}{\partial r^2} + \frac{1}{r}\frac{\partial u_r}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u_r}{\partial \theta^2} - \frac{1}{r^2}u_r - \frac{2}{r^2}\frac{\partial u_\theta}{\partial \theta}\Big) + \frac{\partial p}{\partial r} = \rho\, g_r, \tag{A.3}$$

$$-\Big(\frac{\partial^2 u_\theta}{\partial r^2} + \frac{1}{r}\frac{\partial u_\theta}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u_\theta}{\partial \theta^2} - \frac{1}{r^2}u_\theta + \frac{2}{r^2}\frac{\partial u_r}{\partial \theta}\Big) + \frac{1}{r}\frac{\partial p}{\partial \theta} = \rho\, g_\theta, \tag{A.4}$$

$$\frac{1}{r}\frac{\partial (r u_r)}{\partial r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} = 0. \tag{A.5}$$

We can find a solution by introducing the "stream function" $\psi(r,\theta)$, and expressing the velocity through it:

$$u_r = \frac{1}{r}\frac{\partial \psi}{\partial \theta} \quad \text{and} \quad u_\theta = -\frac{\partial \psi}{\partial r}\,. \tag{A.6}$$

By this construction, the velocity field $\boldsymbol{u}$ then automatically satisfies the continuity equation (A.5).

We proceed by assuming that the the stream function is separable, i.e., that it can be expressed in the form $\psi(r,\theta) = F(r)G(\theta)$ for functions F, G still to be determined. This form then immediately implies $u_r = \frac{1}{r}F(r)G'(\theta)$ and $u_\theta = -F'(r)G(\theta)$. Thus, equations (A.3) and (A.4) become

$$-\Big(\frac{1}{r}F''G' + \frac{1}{r^2}F'G' + \frac{1}{r^3}FG' + \frac{1}{r^3}FG''' - \frac{1}{r^3}FG' + \frac{2}{r^2}F'G'\Big) = -\frac{\partial p}{\partial r} + \rho\, g_r, \tag{A.7}$$

$$-\Big(-F'''G - \frac{1}{r}F''G - \frac{1}{r^2}F'G'' + \frac{1}{r^2}F'G + \frac{2}{r^3}FG''\Big) = -\frac{1}{r}\frac{\partial p}{\partial \theta} + \rho\, g_\theta. \tag{A.8}$$

We can obtain a solution of this set of equations in the spirit of manufactured solutions by choosing $F(r) = \frac{1}{8c}r^8$ and $G(\theta) = c$ where c can be any nonzero constant. This corresponds to a flow field with no radial component $u_r = 0$ and a constant (but radially variable) angular velocity $u_\theta = -r^7$. Since F and G always appear as a product, c can be chosen arbitrarily and we will set it to $c = 1$.

Using this form then still requires us to find appropriate expressions for the pressure $p(r,\theta)$, the density $\rho(r,\theta)$, and the gravity vector $\boldsymbol{g} = (g_r, g_\theta)$ to satisfy the governing equations. Since ρ only appears in a product with the gravity vector, we set

$$\rho(r,\theta) = 48r^5, \tag{A.9}$$

ensuring that it is spatially variable but constant along streamlines.

Further substituting all of these expressions into (A.7)–(A.8) then yields

$$0 = -\frac{\partial p}{\partial r} + 48r^5\, g_r, \tag{A.10}$$

$$48r^5 = -\frac{1}{r}\frac{\partial p}{\partial \theta} + 48r^5\, g_\theta. \tag{A.11}$$

If we assume a radially outward gravity component $g_r = \frac{r^3}{384}$, this implies that

$$0 = -\frac{\partial p}{\partial r} + \frac{r^8}{8}. \tag{A.12}$$

Integrating with respect to r and normalizing the pressure such that at the outer boundary $r = R_2 = 2$ we have $p(r = R_2, \theta) = 0$, yields

$$p(r,\theta) = \frac{r^9}{72} - \frac{512}{72} \tag{A.13}$$

Given this pressure, the final remaining equation, (A.11), is

$$48r^5 = 48r^5 g_\theta. \tag{A.14}$$

This results in $g_\theta = 1$.

In summary, our constructed solution is as follows:

$$\boldsymbol{u} = \begin{bmatrix} 0 \\ -r^7 \end{bmatrix}, \qquad p = \frac{r^9}{72} - \frac{512}{72}, \qquad \rho = 48r^5, \qquad \boldsymbol{g} = \begin{bmatrix} \frac{r^3}{384} \\ 1 \end{bmatrix}. \tag{A.15}$$

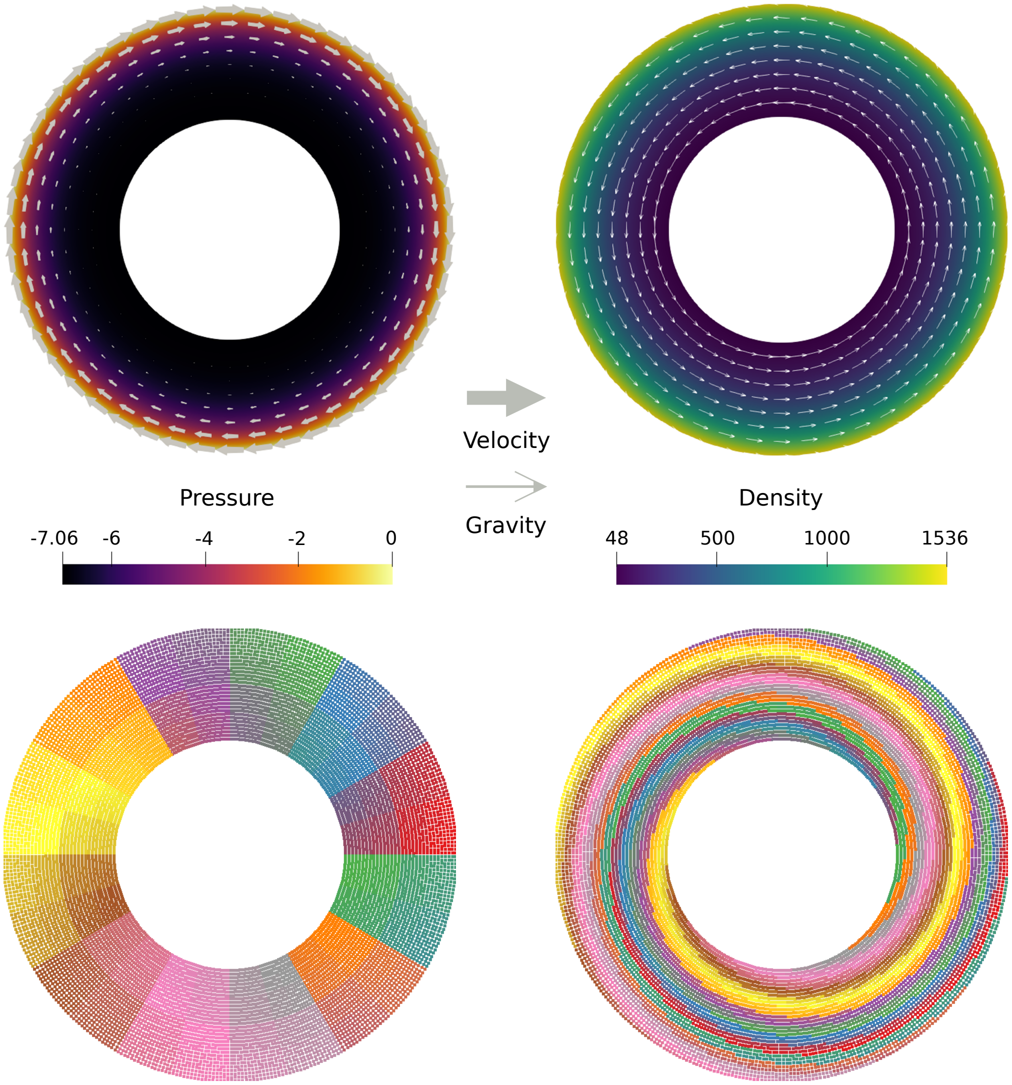

Figure A1. Solution of the annular flow benchmark. Top left: The velocity and pressure solution of the benchmark. Top right: Density and gravity fields that determine the right hand side of the Stokes system. Bottom row: Initial and final particle distributions after one full revolution of the outer edge, colored by particle index.