Geophysical Journal International

Geophys. J. Int. (2019) **00,** 1

doi: 10.1093/gji/ggz405

GJI Geodynamics and Tectonics

Evaluating the accuracy of hybrid finite element/particle-in-cell methods for modelling incompressible Stokes flow

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

[1] *Department of Earth and Planetary Sciences, U. C. Davis, Davis, CA 95616, USA. E-mail:* rene.gassmoeller@mailbox.org

[2] *Department of Mathematics, Colorado State University, Fort Collins, CO 80523, USA*

[3] *Department of Mathematics, U. C. Davis, Davis, CA 95616, USA*

Accepted 2019 September 5. Received 2019 February 14; in original form 2019 September 3

SUMMARY

Combining finite element methods for the incompressible Stokes equations with particle-in-cell methods is an important technique in computational geodynamics that has been widely applied in mantle convection, lithosphere dynamics and crustal-scale modelling. In these applications, particles are used to transport along properties of the medium such as the temperature, chemical compositions or other material properties; the particle methods are therefore used to reduce the advection equation to an ordinary differential equation for each particle, resulting in a problem that is simpler to solve than the original equation for which stabilization techniques are necessary to avoid oscillations.

On the other hand, replacing field-based descriptions by quantities only defined at the locations of particles introduces numerical errors. These errors have previously been investigated, but a complete understanding from both the theoretical and practical sides was so far lacking. In addition, we are not aware of systematic guidance regarding the question of how many particles one needs to choose per mesh cell to achieve a certain accuracy.

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

Our methods and results allow designing new particle-in-cell methods with specific convergence rates, and also provide guidance for the choice of common building blocks and parameters such as the number of particles per cell. In addition, our new time-dependent benchmark provides a simple test that can be used to compare different implementations, algorithms and for the assessment of new numerical methods for particle interpolation and advection. We provide a reference implementation of this benchmark in ASPECT (the ‘Advanced Solver for Problems in Earth’s ConvecTion’), an open source code for geodynamic modelling.

Key words: Mantle processes; Numerical approximations and analysis; Numerical modelling; Numerical solutions; Dynamics of lithosphere and mantle.

1 INTRODUCTION

Computational geodynamic models are important tools 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 (such as viscosity and density) for the velocity and pressure. These equations are then 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. Among these are techniques that directly solve the advection equations using stabilized finite element or finite difference methods (Brooks & Hughes 1982; Guermond & Pasquetti 2011; Kronbichler *et al.* 2012), volume-of-fluid methods (Hirt & Nichols 1981; Robey & Puckett 2019), but notably also ones in which 'particles' are used to describe the motion of the material with its associated properties. Among these latter methods are particle-in-cell (PIC) or marker-and-cell methods (Evans *et al.* 1957; Harlow & Welch 1965), and interface tracking methods such as marker chain (Poliakov & Podladchikov 1992). For a recent comparison, see Puckett *et al.* (2018).

PIC methods in particular have been widely used for geodynamic computations (Tackley & King 2003; Moresi *et al.* 2003; Gerya & Yuen 2003; McNamara & Zhong 2004; Popov & Sobolev 2008; Samuel 2018), since they are conceptually simple and do not require specialized algorithms or other techniques to stabilize the solution of the strongly advection-dominated equation. In PIC or related methods, the advected property is transported on a set of discrete particles that are 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 location and, possibly, the evolution of the quantity. 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. After the Stokes solve, the locations and properties of the particles are updated, for example, by interpolating the newly computed solution or an appropriately determined update 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. Among these are that PIC methods are difficult to combine with adaptively refined and dynamically changing meshes, since the number of particles per cell (PPC) may vary widely during a computation and the numerical error and convergence properties of the method are difficult to determine precisely (see also Gassmöller *et al.* 2018). At a more fundamental level, we are not aware of a systematic study that considers the different contributions to the overall numerical error in a PIC scheme. The excellent paper by Thielmann, May and Kaus (Thielmann *et al.* 2014) provides many answers in this regard, but leaves open others that relate, in particular, to the question of what convergence orders one can expect in time-dependent Stokes flow, and when appropriately varying the number of PPC. A separate recent study by Samuel (see Samuel 2018) is concerned with improvements of the PIC method for time-dependent shear flow and reducing the required number of particles, but does not quantify the influence of the accumulated particle error on the Stokes solution. We therefore consider the current study an extension of Thielmann *et al.* (2014) in which we provide both a theoretical analysis and numerical evidence that support each other, and complementary to Samuel (2018) in that we investigate the particle error contributions on time-dependent flow.

Specifically, we quantitatively determine the accuracy of PIC methods coupled to finite element-based Stokes solvers in order to untangle 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-based 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 (Zhong 1996; Duretz *et al.* 2011), and discuss how the convergence rate of the computed solution depends on different finite element and interpolation algorithm combinations. Our numerical results generally reproduce our theoretical predictions and demonstrate that in order to recover the intrinsic convergence rate of a given finite element, we need both a sufficiently accurate particle interpolation algorithm and sufficiently many PPC. Crucially, however, we also show that for the chosen algorithms the number of PPC needs to grow with the mesh resolution in order to retain the optimal convergence order for higher-order elements, leading to a method in which the cost of particle advection grows faster than the cost of the mesh-based computations if higher accuracy is required.

We then extend these considerations to the time-dependent case by developing two new benchmarks, and use them to evaluate the coupled finite element/PIC scheme. All of our results are implemented in the open-source geodynamic modelling code ASPECT (Kronbichler *et al.* 2012; Heister *et al.* 2017). It is our intention that these results will act as reference results for future code comparison studies of time-independent or 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.

This paper continues as follows: In Section 2, we present the continuous model we wish to solve. Section 3 then describes in detail how we compute numerical approximations to the solution of the model, and we end the section with a theoretical analysis of error contributions and the convergence orders one can predict using this analysis. Section 4 then uses stationary benchmarks to confirm that the theoretical analysis indeed correctly describes what one sees in practical computations. Section 5 extends these results to time-dependent problems: we present two new, *time-dependent* benchmarks, the derivations of which, including analytical solutions, may be found in Appendices A and B, respectively. These benchmarks then allow us to evaluate the error and convergence rates for time-dependent computations of incompressible Stokes flow coupled to a PIC advection method. We conclude in Section 6.

2 GOVERNING EQUATIONS

Geologic deformation over long timescales is commonly modelled by the incompressible Stokes equations for a slow-moving fluid, using a spatially and temporally variable viscosity that depends nonlinearly on both the strain rate and pressure of the fluid, as well as temperature, chemical composition and possibly 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, pressure and chemical composition differences.

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

$$-\nabla\cdot(2\eta\,\varepsilon(\boldsymbol{u}))+\nabla p=\rho\,\boldsymbol{g}, \tag{1}$$

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

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, as well as possibly other relevant quantities such as grain size distributions, frozen stress tensors, etc., all of which are transported along with the velocity $\boldsymbol{u}$ (see Schubert *et al.* 2001). If we denote (the components of) these fields by $\phi_c=\phi_c(\boldsymbol{x},t)$, $c=1,\dots,C$, then each such ϕ_c typically satisfies an advection-diffusion equation of the form

$$\frac{\mathrm{D}\phi_c}{\mathrm{D}t}-\nabla\cdot(\kappa_c\nabla\phi_c)=\frac{\partial\phi_c}{\partial t}+\boldsymbol{u}\cdot\nabla\phi_c-\nabla\,(\kappa_c\nabla\phi_c)=H_c, \tag{3}$$

augmented by appropriate initial conditions $\phi_c(\boldsymbol{x},0)=\phi_{c,0}(\boldsymbol{x})$ and, if necessary, boundary conditions. H_c is a source term that in general depends on both the flow variables and some or all of the other $\phi_{c'}$. For example, if ϕ_c denotes the temperature, then the source term might include contributions due to friction heating and adiabatic compression, while if ϕ_c represents a particular material type's volume fraction, it might increase its value at the cost of that of other materials.

The importance of these additional fields lies in the fact that in realistic descriptions of convection in the Earth, the viscosity η and density ρ in the Stokes equations above not only depend on strain rate $\varepsilon(\boldsymbol{u})$ and pressure p but also on these additional variables ϕ_c. 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.

In typical applications eq. (3) is dominated by the advection term $\boldsymbol{u}\cdot\nabla\phi_c$, and the contributions by the diffusion term $-\nabla\cdot(\kappa_c\nabla\phi_c)$ are rather small (if ϕ_c denotes the temperature) or are completely negligible (e.g. if ϕ_c denotes a chemical composition). In this paper, we are concerned with solving these equations for quantities for which the diffusion term can be neglected; in this case, the equation above simplifies to

$$\frac{\mathrm{D}\phi_c}{\mathrm{D}t}=\frac{\partial\phi_c}{\partial t}+\boldsymbol{u}\cdot\nabla\phi_c=H_c. \tag{4}$$

Consequently, this paper is devoted to solving the coupled set of Stokes and advection equations (1), (2) and (4), accurately. In particular, we will consider approximating the solution of eq. (4) using particle methods and how these methods affect the accuracy of solving eqs (1) and (2) using field-based finite element methods when the two approaches are coupled.

In the following sections, we will not make use of the fact that we may, in fact, have more than one additional property. As a consequence, we will drop the index c on the quantities ϕ_c. However, everything we will say below remains true for cases with multiple such properties.

3 NUMERICAL METHODOLOGY

Eqs (1), (2) and (4) can be solved by direct discretization via finite element, finite volume or finite difference methods, or a variety of other methods (see e.g. Donea & Huerta 2003; Deubelbeiss & Kaus 2008; Gerya 2009; Ismail-Zadeh & Tackley 2010).

However, discretizing advection problems such as eq. (4) 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 $k=1\dots N$ are characterized by their location $\boldsymbol{x}_k(t)$ and associated properties $\phi_k(t)$. Their location and value then evolve according to the ordinary differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}_k(t)=\boldsymbol{u}(\boldsymbol{x}_k(t),t),\qquad \boldsymbol{x}_k(0)=\boldsymbol{x}_{k,0}, \tag{5}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\phi_k(t)=H,\qquad \phi_k(0)=\phi_{k,0}, \tag{6}$$

where H is a function of both particle-based quantities $[\boldsymbol{x}_k(t),\phi_k(t)]$, field-based quantities $[\boldsymbol{u}(\boldsymbol{x}_k(t),t),\varepsilon(\boldsymbol{u}(\boldsymbol{x}_k(t),t)),p(\boldsymbol{x}_k(t),t)]$ and possibly other variables such as the time t. Conversely, coefficients in the Stokes system (1) and (2) such as the viscosity η and density ρ at arbitrary points $\boldsymbol{x}$ (e.g. at quadrature points) may depend not only on field quantities such as velocity and pressure at $\boldsymbol{x}$ but also 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 when evaluating the right-hand sides of eqs (5) and (6) at $\boldsymbol{x}_k$, (ii) integrating particle locations and properties in time according to eqs (5) and (6), and finally (iii) transferring data back from particle locations to quadrature points when evaluating coefficients of eqs (1) and (2) at arbitrary locations $\boldsymbol{x}$ during assembly of 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 numerical approximations $\boldsymbol{u}_h(t^n)$ that were found by approximating the solution of the Stokes equations at discrete times t^n. This error therefore depends on the accuracy of the spatial discretization used for the computed velocity field, and of the time-stepping scheme. In the second step, the numerical integration of eq. (5) 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 advection solver scheme; likewise, we obtain an approximation $\phi_{h,k}(t)$ different from the exact solution $\phi_k(t)$ of eq. (6). Finally, no particle will typically be located on a quadrature point $\boldsymbol{x}$ at time t, and the required property $\phi(\boldsymbol{x},t)$ will need to be interpolated in one of many possible approximate ways from the properties $\phi_k(t)$ of nearby particles.

We will assess these errors quantitatively in Sections 4 and 5 in a number of benchmarks, for different Stokes discretizations, different initial particle locations, advection solvers and particle interpolation methods, all of which we will describe in remainder of 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); that is, 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 Sections 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 finite element we have used.

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 particle 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 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 specific particle distribution that controls our results.

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. This method implies that N_K is the square of an integer number of particles per spatial coordinate direction N_{PPD}, that is, $N_K = (N_{\mathrm{PPD}})^d$, where d is the spatial dimension.

(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 roughly constant particle density per area.

Choosing between the two strategies allows us to determine the influence of different particle distributions on the accuracy of the solution. As we will see, for our benchmark models with uniform mesh resolution these differences are in fact pretty small, although they would become important for adaptive meshes, and after a finite amount of shear. Furthermore, for the time-dependent benchmark cases in Section 5, initial particle locations are less critical as particles are moving from their starting positions; for easier reproducibility, we therefore always generate particles at regular grid locations [approach (i)] in the time-dependent cases.

3.3 Advection of particles

As described above, the advection of particles involves solving eq. (5) 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 in 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 will use a second-order accurate BDF2 time-stepping scheme for our Stokes solution, any particle advection method is limited for a time-varying velocity field to be second-order accurate in time. Since the exact solution of the benchmark in Section 3.4 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 (Meyer & Jenny 2004; Wang *et al.* 2015; Pusok *et al.* 2017). However, even perfectly known and divergence free velocities can form shear patterns that lead to particle clustering; this can be addressed with appropriate particle weighting, splitting, and merging schemes (Samuel 2018). We did not employ such methods in our benchmarks, as we limited our benchmarks to moderate strain, 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, while ensuring that they do not affect the convergence rate.

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. This operation is often called 'interpolation' from particle locations to the mesh, though a better term may in fact be 'projection'; we will use the terms interchangeably. In particular, let K be a cell, $I_K \subseteq [1, N]$ be the set of those particle indices (among the overall N particles) that are located on K, and $N_K = |I_K|$ be their number. Then we consider the following two 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 $\phi(\boldsymbol{x})$ on cell K, we average the material properties among all particles located on K:

$$\phi|_K = \frac{1}{N_K} \sum_{k \in I_K} \phi_k. \tag{7}$$

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, for example, harmonic or geometric averaging. However, 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) (see also the related discussion in Heister *et al.* 2017), here we 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,

$$\epsilon^2 = \sum_{k \in I_K} [\phi|_K(\boldsymbol{x}_k) - \phi_k]^2, \tag{8}$$

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-D, or an 8×8 matrix in 3-D, 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 $\phi|_K(\boldsymbol{x})$, that is, evaluating the (bi-/tri-)linear shape functions of the approximant times their corresponding coefficient values. As observed before (Thielmann *et al.* 2014) this algorithm generates over-/undershooting close to strong property gradients, which need to be handled in some form, for example, by a strict limiter for the interpolated property. However, all of the benchmark results we show below are either sufficiently smooth or have property gradients aligned with the mesh, therefore we did not need to apply the limiter here. Note that in contrast to Thielmann *et al.* (2014) we include the mixed polynomial terms *xy* (and, in three space dimensions, *xz*, *yz*, *xyz*) in the interpolation function to stay consistent with the polynomial space of our pressure element. This modification potentially explains why our method performs better for lower number of PPC, as discussed in Section 4.1.

3.5 An error analysis

In this section, let us provide some theoretical considerations for how the particle-based scheme outlined above might affect the overall error in the finite element solution of the Stokes problem. Our goal here is to derive error convergence orders for the L_2 norm errors in velocity and pressure, that is, for

$$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2} = \left(\int |\boldsymbol{u}(\boldsymbol{x}, t) - \boldsymbol{u}_h(\boldsymbol{x}, t)|^2 \, \mathrm{d}x \right)^{1/2}, \tag{9}$$

$$\|p - p_h\|_{L_2} = \left(\int |p(\boldsymbol{x}, t) - p_h(\boldsymbol{x}, t)|^2 \, \mathrm{d}x \right)^{1/2}. \tag{10}$$

We will test the statements we will derive in computational experiments in the sections to follow.

Before stating concrete error inequalities, let us present the conceptual framework in which these are presented. In particular, in Section 4 we will consider the numerical approximation of the solution of stationary Stokes problems (1) and (2) using the finite element method in which we do not know the exact density ρ and viscosity η, but only have this information available at the locations of particles. (In Section 5, where we consider time-dependent benchmarks, we will in fact only know the exact density and viscosity at points $\boldsymbol{x}_k$ whose coordinates are only approximately known; we ignore this for the moment.) This can be stated as follows: In the numerical problem that we will solve using the finite element method, we will use a density $\rho_h = I_h R_h \rho$ and viscosity $\eta_h = I_h R_h \eta$, where the operator $R_h f$ *restricts* the values of a function f to the locations of particles, and the operator I_h *interpolates* the values of a function defined only at particle locations to the entire domain so that it can be evaluated at arbitrary quadrature points for use in the finite element method; I_h can be one of the two options discussed in the previous subsection. The question is how the replacement of ρ, η by ρ_h, η_h affects the accuracy with which we can compute numerical approximations $\boldsymbol{u}_h$, p_h via the finite element method.

Let us then concisely define what problem we solve. In particular, let $\mathcal{L}_\eta$ be the solution operator of the Stokes equations (1) and (2), that is, for a given right-hand side $\rho \boldsymbol{g}$ and viscosity η, we have that $\{\boldsymbol{u}, p\} = \mathcal{L}_\eta(\rho \boldsymbol{g})$ solves the Stokes equations. Furthermore, let $\mathcal{L}_\eta^h$ be the *discrete* solution operator, that is, $\{\boldsymbol{u}_h, p_h\} = \mathcal{L}_\eta^h(\rho \boldsymbol{g})$ is the finite element solution of these equations. The question we want to answer is how the exact solution $\mathcal{L}_\eta(\rho \boldsymbol{g})$ relates to the finite element approximation $\mathcal{L}_{\eta_h}^h(\rho_h \boldsymbol{g})$ in which we have replaced density and viscosity as discussed above. Specifically, we will measure this error in the 'energy norm':

$$|||\mathcal{L}_\eta(\rho \boldsymbol{g}) - \mathcal{L}_{\eta_h}^h(\rho_h \boldsymbol{g})|||^2 = \eta_0 \|\nabla(\boldsymbol{u} - \boldsymbol{u}_h)\|_{L_2}^2 + \|p - p_h\|_{L_2}^2, \tag{11}$$

where η_0 is a suitably chosen reference viscosity that ensures that the two terms are appropriately balanced and have matching physical units. We will later relate this norm to the L_2 norms of both the velocity and pressure errors (instead of the H^1 semi-norm of the velocity and the L_2 norm of the pressure).

To answer the question about the size of the error, let us first consider the following auxiliary problem: It is well known that replacing a sufficiently smooth function ρ or η by a suitable (i) piecewise constant or (ii) piecewise (bi-/tri-)linear approximation on a mesh of maximal mesh size h incurs an error proportional to h and h^2, respectively, when measuring the error in the L_2 norm. In other words, if we denote these approximants by $P_h \rho$ and $P_h \eta$, then

$$\begin{aligned} \|\rho - P_h \rho\|_{L_2} &= \mathcal{O}(h^r), \\ \|\eta - P_h \eta\|_{L_2} &= \mathcal{O}(h^r), \end{aligned} \tag{12}$$

where $r = 1$ for approximation option (i) and $r = 2$ for option (ii) of the previous subsection. Concisely, we define P_h locally on each cell K as follows, when applied to an arbitrary function f:

$$\begin{aligned} &\text{option (i):} \quad P_h f|_K = \frac{1}{|K|} \int_K f, \\ &\text{option (ii):} \quad P_h f|_K = \arg \min_{\varphi_h \in Q_1(K)} \frac{1}{2} \|f - \varphi_h\|_{L_2(K)}^2. \end{aligned} \tag{13}$$

Note that P_h is equal to $I_h R_h$ if one were to consider infinitely many particles equally distributed on each cell K because then the point-based least-squares approximations (7) and (8) agree with the integral-based least-squares approximations in (13). Below, we will also need estimates such as (12) in other norms, and consequently state the following results:

$$\begin{aligned} \|f - P_h f\|_{H^1} &= \mathcal{O}(h^{r-1}), \\ \|f - P_h f\|_{H^{-1}} &= \mathcal{O}(h^{r+1}), \end{aligned} \tag{14}$$

where the first denotes the error in the gradient of f.

Using this argument, we can now decompose the overall error into four components. Namely, we will write the error as follows:

$$\begin{aligned}
&\left(\eta_0\|\nabla(\boldsymbol{u}-\boldsymbol{u}_h)\|_{L_2}^2+\|p-p_h\|_{L_2}^2\right)^{1/2}\\
&\quad= |||\mathcal{L}_\eta(\rho\boldsymbol{g})-\mathcal{L}^h_{\eta_h}(\rho_h\boldsymbol{g})|||\\
&\quad\le \underbrace{|||\mathcal{L}_\eta(\rho\boldsymbol{g})-\mathcal{L}_\eta(P_h\rho\,\boldsymbol{g})|||}_{(1)}\\
&\qquad+\underbrace{|||\mathcal{L}_\eta(P_h\rho\,\boldsymbol{g})-\mathcal{L}_{P_h\eta}(P_h\rho\boldsymbol{g})|||}_{(2)}\\
&\qquad+\underbrace{|||\mathcal{L}_{P_h\eta}(P_h\rho\,\boldsymbol{g})-\mathcal{L}_{\eta_h}(\rho_h\boldsymbol{g})|||}_{(3)}\\
&\qquad+\underbrace{|||\mathcal{L}_{\eta_h}(\rho_h\boldsymbol{g})-\mathcal{L}^h_{\eta_h}(\rho_h\boldsymbol{g})|||.}_{(4)}
\end{aligned}\tag{15}$$

Here, the four norm terms on the right correspond, respectively, to (1) the error introduced by replacing ρ by the projection $P_h\rho$ when solving the continuous Stokes equations, (2) the error introduced by replacing η by the projection $P_h\eta$ when solving the continuous Stokes equations, (3) the error introduced by further substituting $P_h\rho$, $P_h\eta$ by $\rho_h = I_hR_h\rho$, $\eta_h = I_hR_h\eta$ when solving the continuous Stokes equations and (4) the error introduced by the finite element solution instead of the exact solution of two problems with the same coefficients. Let us determine the size of these terms individually, in increasing order of difficulty.

For the discretization error, (4), it is well known that when using either $Q_k \times Q_{k-1}$ or $Q_k \times P_{-(k-1)}$ finite elements, we have

$$|||\mathcal{L}_{\eta_h}(\rho_h\boldsymbol{g})-\mathcal{L}^h_{\eta_h}(\rho_h\boldsymbol{g})||| = \mathcal{O}(h^k),\tag{16}$$

where h is the diameter of the largest cell of the mesh. It is worth mentioning that this statement is only correct if the solution is sufficiently smooth (e.g. $\boldsymbol{u} \in H^{k+1}$ and $p \in H^k$).

The replacement error for the density (1) is also easy. To this end, one needs to know that the Stokes operator is linear and stable in the H^{-1} norm, that is, that

$$|||\mathcal{L}_\eta f_1-\mathcal{L}_\eta f_2||| = |||\mathcal{L}_\eta(f_1-f_2)||| \le C\,\|f_1-f_2\|_{H^{-1}}\,,\tag{17}$$

with some constant $C < \infty$. Since $f_1 = \rho$ and $f_2 = P_h\rho$, we can use (14) to show that the first error term satisfies

$$|||\mathcal{L}_\eta(\rho\boldsymbol{g})-\mathcal{L}_\eta(P_h\rho\,\boldsymbol{g})||| = \mathcal{O}(h^{r+1}).\tag{18}$$

The replacement error for the viscosity, (2), is more difficult to analyse. However, it is reasonable to assume that the solutions of two Stokes equations with viscosities η_1, η_2 differ by an amount proportional to $\|\eta_1-\eta_2\|_{L_2}$. This would here suggest, invoking (12), that

$$|||\mathcal{L}_{P_h\eta}(P_h\rho\,\boldsymbol{g})-\mathcal{L}_\eta(P_h\rho\,\boldsymbol{g})||| \le D\|\eta-P_h\eta\|_{L_2} = \mathcal{O}(h^r),\tag{19}$$

again with some constant D. We have no proof of this statement, though it seems reasonable using standard arguments in the analysis of elliptic partial differential equations (see e.g. Gilbarg & Trudinger 1983). The use of the L_2 norm—or maybe the L_∞ norm, for which one obtains the same estimate—seems natural when analysing solutions this way. One might be tempted to ask whether one could replace $\|\eta-P_h\eta\|_{L_2}$ by $\|\eta-P_h\eta\|_{H^{-1}}$ and thereby gain an order of convergence. But it will turn out, based on our numerical examples, that the estimate is indeed correct as stated regarding the order of convergence.

This leaves the error (3) due to replacing the projections $P_h\rho$, $P_h\eta$ by the interpolants $\rho_h = I_hR_h\rho$, $\eta_h = I_hR_h\eta$. Similar arguments as those for the errors (1) and (2) yield that

$$\begin{aligned}|||\mathcal{L}_{P_h\eta}(P_h\rho\,\boldsymbol{g}) - \mathcal{L}_{\eta_h}(\rho_h\boldsymbol{g})||| \le{}& C\|P_h\rho-I_hR_h\rho\|_{H^{-1}}\\ &+D\|P_h\eta-I_hR_h\eta\|_{L_2}.\end{aligned}\tag{20}$$

The exact size of these terms depends on how many particle locations we have on each cell, as well as how they are located. All we know is that if we increase the number of points, and if these points are uniformly distributed, then $I_hR_h \to P_h$ and consequently the entire error term goes to zero. For finite numbers of PPC, we will simply denote the right-hand side as $E(h, \mathrm{PPC})$ with the expectation that asymptotically $E(h, \mathrm{PPC}) \to 0$ as $\mathrm{PPC} \to \infty$ or $h \to 0$.

Taking all of this together then yields that we should expect the following error behaviour in the energy norm of the Stokes problem:

$$\begin{aligned}&\left(\eta_0\|\nabla(\boldsymbol{u}-\boldsymbol{u}_h)\|_{L_2}^2+\|p-p_h\|_{L_2}^2\right)^{1/2}\\ &\quad= \mathcal{O}(h^{r+1})+\mathcal{O}(h^r)+\mathcal{O}(E(h,\mathrm{PPC}))+\mathcal{O}(h^k).\end{aligned}\tag{21}$$

This immediately yields the desired behaviour of the pressure error in the L_2 norm:

$$\|p-p_h\|_{L_2} = \mathcal{O}(h^{r+1})+\mathcal{O}(h^r)+\mathcal{O}(E(h,\mathrm{PPC}))+\mathcal{O}(h^k).\tag{22}$$

The velocity error in the L_2 norm requires marginally more work. Using the standard Nitsche trick (Brenner & Scott 2007) to obtain the L_2 error from the H^1 error provides us with an extra power of h and then yields

$$\begin{aligned}\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2} ={}& \mathcal{O}(h^{r+2})+\mathcal{O}(h^{r+1})\\ &+\mathcal{O}(h\,E(h,\mathrm{PPC}))+\mathcal{O}(h^{k+1}).\end{aligned}\tag{23}$$

The next section of this paper is in essence an exploration of these last two relationships using concrete test cases.

4 INSTANTANEOUS BENCHMARKS

The first set of benchmarks we will consider only solves 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 the 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 linear system 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. In the first of these cases, only the contribution (4) of the errors considered in Section 3.5 is present, whereas in the second case, all four contributions [(1)–(4)] matter.

As we will show, and as anticipated in Section 3.5, 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 as defined in Section 3.5, considering both the velocity and pressure. The relevant 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 superconvergence effects.

We will defer to the next section a discussion of the 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 2-D square domain with height and width of one. It uses tangential boundary conditions, 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

$$\eta(x, y) = \mathrm{e}^{2By}, \tag{24}$$

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

$$\rho(x, y) = -\sin(2y)\cos(3\pi x). \tag{25}$$

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 mesh resolutions. The methods converge with different rates, and indeed at the rates predicted by eqs (22) and (23). The initial particle locations do not influence the convergence rate significantly, though the absolute errors are somewhat larger for random particle locations, likely because some cells receive unfavourable particle locations (e.g. a high particle density in only a small volume of the cell). We also observe that for smaller numbers of PPC than the one shown here, the difference between the results obtained using regular and random particle locations is larger. 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, as this delivers more reproducible model results.

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* and, in the notation of Section 3.5 and eq. (12), it corresponds to $r = \infty$ because the projection of the coefficients onto the function that is actually evaluated is the identity operation. The results of Section 3.5 then predict that, for both the $Q_k \times Q_{k-1}$ and the $Q_k \times P_{-(k-1)}$ elements, the velocity and pressure errors decay as h^{k+1} and h^k, respectively. Indeed, we show this experimentally in the leftmost 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.

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 ($k = 2$), for arithmetic averaging ($r = 1$) and bilinear least-squares ($r = 2$) interpolation methods for regular and random particle distributions as discussed in Section 3.2. PPC (particles per cell), k and r are as defined in Section 3.5.

Arithmetic average ($r = 1$)					
		Regular		Random	
h	PPC	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2}$	Rate	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2}$	Rate
$\frac{1}{8}$	100	7.05×10^{-6}	–	7.08×10^{-6}	–
$\frac{1}{16}$	100	1.86×10^{-6}	1.92	1.95×10^{-6}	1.86
$\frac{1}{32}$	100	4.81×10^{-7}	1.95	4.87×10^{-7}	2.00
$\frac{1}{64}$	100	1.22×10^{-7}	1.98	1.29×10^{-7}	1.92
$\frac{1}{128}$	100	3.05×10^{-8}	2.00	2.93×10^{-8}	2.13
$\frac{1}{256}$	100	7.63×10^{-9}	2.00	7.91×10^{-9}	1.89
		$\|p - p_h\|_{L_2}$		$\|p - p_h\|_{L_2}$	
$\frac{1}{8}$	100	1.91×10^{-2}	–	1.92×10^{-2}	–
$\frac{1}{16}$	100	1.24×10^{-2}	0.62	1.24×10^{-2}	0.63
$\frac{1}{32}$	100	6.57×10^{-3}	0.92	6.60×10^{-3}	0.91
$\frac{1}{64}$	100	3.33×10^{-3}	0.98	3.35×10^{-3}	0.98
$\frac{1}{128}$	100	1.67×10^{-3}	1.00	1.68×10^{-3}	1.00
$\frac{1}{256}$	100	8.37×10^{-4}	1.00	8.40×10^{-4}	1.00
Bilinear least squares ($r = 2$)					
		Regular		Random	
h	PPC	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2}$	Rate	$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2}$	Rate
$\frac{1}{8}$	100	1.72×10^{-6}	–	1.68×10^{-6}	–
$\frac{1}{16}$	100	2.46×10^{-7}	2.81	2.49×10^{-7}	2.75
$\frac{1}{32}$	100	3.50×10^{-8}	2.81	3.52×10^{-8}	2.82
$\frac{1}{64}$	100	4.56×10^{-9}	2.94	4.71×10^{-9}	2.90
$\frac{1}{128}$	100	5.95×10^{-10}	2.94	6.55×10^{-10}	2.85
$\frac{1}{256}$	100	8.41×10^{-11}	2.82	1.05×10^{-10}	2.64
		$\|p - p_h\|_{L_2}$		$\|p - p_h\|_{L_2}$	
$\frac{1}{8}$	100	4.53×10^{-3}	–	4.72×10^{-3}	–
$\frac{1}{16}$	100	1.30×10^{-3}	1.80	1.33×10^{-3}	1.83
$\frac{1}{32}$	100	3.42×10^{-4}	1.93	3.49×10^{-4}	1.93
$\frac{1}{64}$	100	8.67×10^{-5}	1.98	8.84×10^{-5}	1.98
$\frac{1}{128}$	100	2.17×10^{-5}	2.00	2.22×10^{-5}	1.99
$\frac{1}{256}$	100	5.43×10^{-6}	2.00	5.54×10^{-6}	2.00

Next, we investigate the case where the viscosity and density are not obtained from an exactly prescribed function, but are instead interpolated from nearby particles. The corresponding convergence orders for the velocity and pressure errors are shown in the second and third set of columns in Table 2. For these results, we use between 4 and 361 PPC, distributed on a regular, equidistant grid. For models in which results depend on increasing PPC we always choose the smallest, most efficient number of particles that reaches the largest possible convergence rate.

The table then shows that a cell-wise arithmetic average interpolation for the $Q_2 \times P_{-1}$ element reduces the convergence of the velocity error to second order. We have verified that this remains so if the number of PPC were larger than the one used in the table. In other words using a cell-wise constant averaging is suboptimal by one order no matter how many PPC are used, and this also makes sense in view of the discussion in Section 3.5 that suggests that the best order that can be achieved is $\min\{k + 1, r + 1\}$ for the velocity

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. PPC (particles per cell), k and r are as defined in Section 3.5.

				$Q_2 \times P_{-1}$ ($k=2$)				
$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2}$	Direct method ($r=\infty$)			Arithmetic average ($r=1$)			Bilinear least squares ($r=2$)	
h	Error	Rate	PPC	Error	Rate	PPC	Error	Rate
$\frac{1}{8}$	1.51×10^{-6}	–	4	6.32×10^{-6}	–	4	2.24×10^{-6}	–
$\frac{1}{16}$	2.50×10^{-7}	2.60	4	1.61×10^{-6}	1.97	4	3.61×10^{-7}	2.63
$\frac{1}{32}$	3.52×10^{-8}	2.80	4	4.15×10^{-7}	1.96	9	4.62×10^{-8}	2.97
$\frac{1}{64}$	4.53×10^{-9}	3.00	4	1.05×10^{-7}	1.98	25	5.3×10^{-9}	3.12
$\frac{1}{128}$	5.7×10^{-10}	3.00	4	2.63×10^{-8}	2.00	49	6.75×10^{-10}	2.97
$\frac{1}{256}$	7.23×10^{-11}	3.00	4	6.58×10^{-9}	2.00	100	8.41×10^{-11}	3.00
$\frac{1}{512}$	9.14×10^{-12}	3.00	4	1.64×10^{-10}	2.00	196	1.05×10^{-11}	3.00
$\|p-p_h\|_{L_2}$								
$\frac{1}{8}$	5.02×10^{-3}	–	4	1.93×10^{-2}	–	4	4.58×10^{-3}	–
$\frac{1}{16}$	1.33×10^{-3}	1.90	4	1.24×10^{-2}	0.64	4	1.31×10^{-3}	1.80
$\frac{1}{32}$	3.44×10^{-4}	2.00	4	6.58×10^{-3}	0.92	9	3.43×10^{-4}	1.94
$\frac{1}{64}$	8.68×10^{-5}	2.00	4	3.33×10^{-3}	0.98	25	8.67×10^{-5}	1.98
$\frac{1}{128}$	2.17×10^{-5}	2.00	4	1.67×10^{-3}	1.00	49	2.17×10^{-5}	2.00
$\frac{1}{256}$	5.43×10^{-6}	2.00	4	8.37×10^{-4}	1.00	100	5.43×10^{-6}	2.00
$\frac{1}{512}$	1.36×10^{-6}	2.00	4	4.19×10^{-4}	1.00	196	1.36×10^{-6}	2.00
				$Q_3 \times Q_2$ ($k=3$)				
$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2}$	Direct method ($r=\infty$)			Arithmetic average ($r=1$)			Bilinear least squares ($r=2$)	
h	Error	Rate	PPC	Error	Rate	PPC	Error	Rate
$\frac{1}{8}$	3.1×10^{-7}	–	4	5.78×10^{-6}	–	9	1.26×10^{-6}	–
$\frac{1}{16}$	2.48×10^{-8}	3.64	4	1.36×10^{-6}	2.08	9	1.64×10^{-7}	2.94
$\frac{1}{32}$	1.59×10^{-9}	3.96	4	3.34×10^{-7}	2.03	16	2.09×10^{-8}	2.97
$\frac{1}{64}$	9.9×10^{-11}	4.00	4	8.27×10^{-8}	2.01	36	2.27×10^{-9}	3.20
$\frac{1}{128}$	6.23×10^{-12}	3.99	4	2.06×10^{-8}	2.01	81	2.52×10^{-10}	3.17
$\frac{1}{256}$			4	5.13×10^{-9}	2.00	169	3.01×10^{-11}	3.07
$\frac{1}{512}$			4	1.28×10^{-9}	2.00	361	3.66×10^{-12}	3.04
$\|p-p_h\|_{L_2}$								
$\frac{1}{8}$	7.04×10^{-4}	–	4	1.86×10^{-2}	–	9	1.37×10^{-3}	–
$\frac{1}{16}$	1.15×10^{-4}	2.61	4	8.27×10^{-3}	1.17	9	1.18×10^{-3}	0.21
$\frac{1}{32}$	1.68×10^{-5}	2.78	4	3.06×10^{-3}	1.43	16	3.52×10^{-4}	1.74
$\frac{1}{64}$	2.3×10^{-6}	2.89	4	1.11×10^{-3}	1.47	36	9.19×10^{-5}	1.94
$\frac{1}{128}$	3.03×10^{-7}	2.92	4	3.99×10^{-4}	1.48	81	2.32×10^{-5}	1.98
$\frac{1}{256}$	3.89×10^{-8}	2.96	4	1.43×10^{-4}	1.48	169	5.83×10^{-6}	2.00
$\frac{1}{512}$	4.94×10^{-9}	2.98	4	5.07×10^{-5}	1.49	361	1.46×10^{-6}	2.00

and min $\{k, r\}$ for the pressure [see eqs (22) and (23)]. For the element used here, we have $k = 2$, and cell-wise constant interpolation implies $r = 1$, so we need to expect the observed reduction in order of convergence.

When we use the bilinear least-squares interpolation ($r = 2$) we find an interesting behaviour that was briefly observed, but not fully explored before (Thielmann *et al.* 2014); at low resolutions and for a constant number of PPC the velocity error decreases with nearly the expected rate of the direct method, but then degrades to second-order convergence (not shown in the table; however, see Fig. 1 and compare also fig. 6(b) of Thielmann *et al.* (2014)). Here we show that increasing the number of PPC approximately linearly with increasing resolution recovers the expected convergence rate of the Stokes element (see the last set of columns in Table 2 and the top plot in Fig. 1). This is a behaviour that to our knowledge has not been described using geodynamic benchmark results before. We also note that our implementation seems to be less sensitive to the number of PPC since our convergence rate remains optimal to $h = \frac{1}{512}$ for PPC = 256, while the implementation in (Thielmann *et al.* 2014) degrades to second order at $h \approx \frac{1}{128}$ for the same number of PPC. We speculate that this is caused by our use of a bilinear approximation, instead of a linear one, as discussed in Section 3.4. The pressure error for the $Q_2 \times P_{-1}$ 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). All of these results are of course consistent with the predictions of Section 3.5, if one assumes a specific relationship for $E(h, \text{PPC})$ as further discussed below.

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 the error with the expected fourth order. The arithmetic average interpolation method again achieves second-order accuracy, which for this element is sub-optimal by two

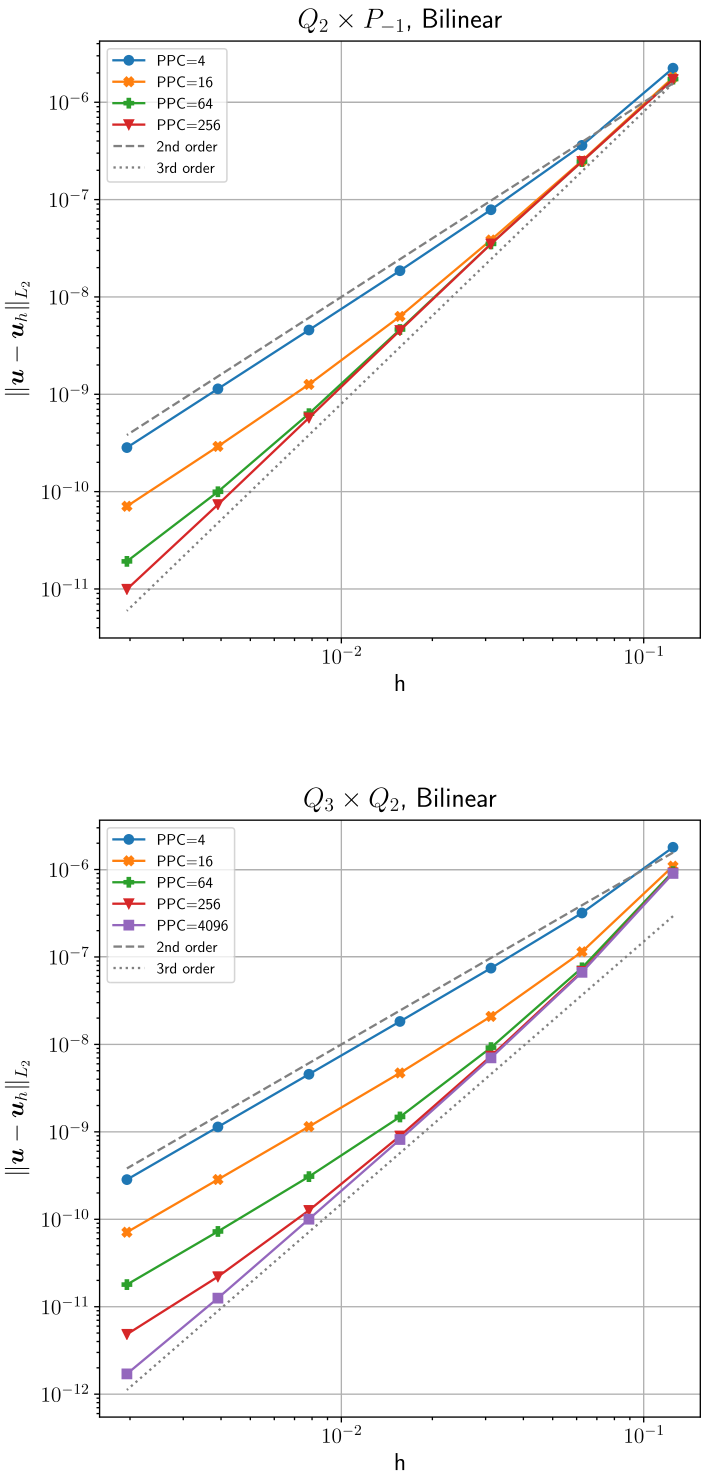

Figure 1. Velocity errors $\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2}$ for the SolKz benchmark for the $Q_2 \times P_{-1}$ element ($k = 2$, top) and for the $Q_3 \times Q_2$ element ($k = 3$, bottom), using bilinear interpolation ($r = 2$). The error is plotted as a function of both mesh resolution (h) and number of particles per cell (PPC).

orders. The bilinear least-squares interpolation results in second-order convergence with constant PPC (not shown in Table 2, but shown in the bottom plot in Fig. 1), and third-order convergence with increasing PPC. However, as expected it is impossible to recover the fourth-order convergence rate of the direct method with increasing PPC; this is consistent with the theoretical prediction that the velocity error converges at best with a rate of min $\{k+1, r+1\}$, for $k = 3$ and $r = 2$. As for the $Q_2 \times P_{-1}$ element, these results are all consistent with the predictions of eqs (22) and (23); the exception is that for arithmetic averaging, one would expect a first-order convergence rate for the pressure when in fact we observe order 1.5.

To further clarify the effect of the number of PPC on the convergence rate when using the bilinear interpolation scheme ($r = 2$), Fig. 1 shows convergence data for the velocity error $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2}$ as a function of both the mesh resolution (h) and the number of PPC. 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 PPC. For both elements, the velocity error is well described by the approximation $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2} = \mathcal{O}(h^3) + \mathcal{O}(h^2 \text{PPC}^{-1})$. This can be compared with (23), predicting $\mathcal{O}(h^{\min\{k+1,r+1\}}) + \mathcal{O}(h\, E(h, \text{PPC}))$, to postulate a specific form for $E(h, \text{PPC})$, namely $E(h, \text{PPC}) = h\, \text{PPC}^{-1}$. For the two parts of Fig. 1, we have $k = 2$ or 3 and $r = 2$.

Fig. 1 only shows velocity errors. We do not show corresponding figures for convergence data for the pressure error because for a bilinear reconstruction, the pressure converges at a fixed rate and is essentially independent of the number of PPC. Increasing the number of particles therefore does not increase the accuracy of the pressure, unlike for the velocity.

As a consequence of all of these considerations, for a fixed number of PPC—that is, 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 4096 PPC 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 $64^3 = 262,144$ PPC in three space dimensions is not a realistic option. Consequently, unless additional measures are taken, any practical use of particle methods combined with higher-order finite elements will be prohibitively expensive for high mesh resolutions, or suffer from a sub-optimal convergence rate.

4.2 SolCx

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

$$\eta(x, y) = \begin{cases} 1 & \text{if} \quad x < 0.5 \\ 10^6 & \text{if} \quad x \geq 0.5, \end{cases} \tag{26}$$

and the density by

$$\rho(x, y) = -\sin(\pi y)\cos(\pi x), \tag{27}$$

all again on the unit square $\Omega = (0, 1)^2$. The complete derivation of the exact 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 if one uses a mesh that is aligned with the interface, as we will do here. 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. Specifically, within the analysis of Section 3.5, this implies that the error contribution labelled (2) in eq. (15) above does not exist for this benchmark and that, consequently, eqs (22) and (23) can be replaced by

$$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2} = \mathcal{O}(h^{r+2}) + \mathcal{O}(h\, E(h, \text{PPC})) + \mathcal{O}(h^{k+1}), \tag{28}$$

$$\|p - p_h\|_{L_2} = \mathcal{O}(h^{r+1}) + \mathcal{O}(E(h, \text{PPC})) + \mathcal{O}(h^k). \tag{29}$$

In other words, as a function of the interpolation order r, the expected convergence order is one higher than in the general case represented by the SolKz benchmark discussed in the previous subsection.

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 ($k = 2$) and the direct method ($r = \infty$, left-most columns of the top half of the table), the velocity error decreases with $O(h^3)$ and the pressure error with $O(h^2)$ as expected and as reported previously (Kronbichler *et al.* 2012), although half an order higher than reported in Thielmann *et al.* (2014). Similarly, and as predicted by eqs (28) and (29) above, when using particles and bilinear reconstructions ($r = 2$, right-most columns of the table), we obtain the same convergence rates as for the direct method. The one exception that violates our theoretical predictions is when using particles and arithmetic averaging ($r = 1$, middle columns) where the theory predicts third and second-order convergence for velocity and pressure, respectively, but we only obtain second order for both. The table shows this for a constant number of PPC, suggesting that perhaps the term involving $E(h, \text{PPC})$ limits the convergence order; however, we have verified that even with large values of PPC, the convergence rate remains at two for the velocity. While we lack an understanding of why theory and practice do not agree here, we note that our data are consistent with previous results in Thielmann *et al.* (2014).

As described before (Kronbichler *et al.* 2012; Thielmann *et al.* 2014), using a continuous pressure element like $Q_3 \times Q_2$ ($k = 3$) in general does not result in the optimal convergence rate for the pressure error because of the discontinuity in the pressure solution. Indeed, all methods to evaluate coefficients (independently of PPC choice) now only reach a pressure convergence rate of $\mathcal{O}(h^{1/2})$ as shown in the bottom half of Table 3. 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 ($r = \infty$, left-most columns) and the bilinear least-squares method ($r = 2$, right-most columns), namely $\mathcal{O}(h^4)$. However, in order to obtain the latter result, we now need to increase $\text{PPC} \propto h^{-2}$: using a constant number of PPC yields a suboptimal convergence order of $\mathcal{O}(h^2)$, whereas using $\text{PPC} \propto h^{-1}$ results in $\mathcal{O}(h^3)$.

The outlier is again the velocity error when using the piecewise constant averaging ($r = 1$) where one would expect third-order convergence but we only observe second order.

The convergence orders predicted for the bilinear interpolation of the density—using $\text{PPC} \propto h^{-2}$—were one order higher than we saw for the SolKz benchmark when using $\text{PPC} \propto h^{-1}$. This conclusion followed from the fact that the viscosity interpolation for SolCx is exact, and remains unchanged if one tried to solve the benchmark with $\text{PPC} \propto h^{-2}$. In order to verify that this interpretation is in fact correct, we repeat the SolKz benchmark with a density that is interpolated from particles, but a viscosity that is exact (i.e. using the particles for density, but the direct method for viscosity)—see the results shown in Fig. 2. The $Q_2 \times P_{-1}$ element shows no difference in the computations with interpolated viscosity, as they already reached the convergence order implied by the discretization error

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 (particles per cell), k and r are as defined in Section 3.5.

			$Q_2 \times P_{-1}$ $(k=2)$					
$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2}$	Direct method $(r=\infty)$		Arithmetic average $(r=1)$			Bilinear least squares $(r=2)$		
h	Error	Rate	PPC	Error	Rate	PPC	Error	Rate
$\frac{1}{8}$	1.32×10^{-5}	–	4	3.16×10^{-5}	–	4	1.36×10^{-5}	–
$\frac{1}{16}$	1.66×10^{-6}	2.99	4	7.30×10^{-6}	2.12	4	1.93×10^{-6}	2.81
$\frac{1}{32}$	2.08×10^{-7}	3.00	4	1.79×10^{-6}	2.03	9	2.36×10^{-7}	3.03
$\frac{1}{64}$	2.60×10^{-8}	3.00	4	4.44×10^{-7}	2.01	25	2.79×10^{-8}	3.08
$\frac{1}{128}$	3.26×10^{-9}	3.00	4	1.11×10^{-7}	2.00	49	3.50×10^{-9}	3.00
$\frac{1}{256}$	4.08×10^{-10}	3.00	4	2.77×10^{-8}	2.00	100	4.39×10^{-10}	3.00
$\frac{1}{512}$	5.13×10^{-11}	3.00	4	6.92×10^{-9}	2.00	196	5.87×10^{-11}	2.90
$\|p-p_h\|_{L_2}$								
$\frac{1}{8}$	1.48×10^{-3}	–	4	3.16×10^{-3}	–	4	1.53×10^{-3}	–
$\frac{1}{16}$	3.7×10^{-4}	2.00	4	8.00×10^{-4}	1.99	4	3.83×10^{-4}	2.00
$\frac{1}{32}$	9.22×10^{-5}	2.00	4	2.00×10^{-4}	2.00	9	9.29×10^{-5}	2.05
$\frac{1}{64}$	2.30×10^{-5}	2.00	4	5.00×10^{-5}	2.00	25	2.30×10^{-5}	2.01
$\frac{1}{128}$	5.75×10^{-6}	2.00	4	1.25×10^{-5}	2.00	49	5.75×10^{-6}	2.00
$\frac{1}{256}$	1.44×10^{-6}	2.00	4	3.12×10^{-6}	2.00	100	1.44×10^{-6}	2.00
$\frac{1}{512}$	3.59×10^{-7}	2.00	4	7.80×10^{-7}	2.00	196	3.59×10^{-7}	2.00
				$Q_3 \times Q_2$ $(k=3)$				
$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2}$	Direct method $(r=\infty)$		Arithmetic average $(r=1)$			Bilinear least squares $(r=2)$		
h	Error	Rate	PPC	Error	Rate	PPC	Error	Rate
$\frac{1}{8}$	6.04×10^{-7}	–	4	3.15×10^{-5}	–	100	9.10×10^{-7}	–
$\frac{1}{16}$	4.03×10^{-8}	3.90	4	7.29×10^{-6}	2.11	400	5.84×10^{-8}	3.96
$\frac{1}{32}$	2.60×10^{-9}	4.00	4	1.79×10^{-6}	2.03	1600	3.70×10^{-9}	3.98
$\frac{1}{64}$	1.67×10^{-10}	4.00	4	4.44×10^{-7}	2.01	6400	2.34×10^{-10}	3.97
$\frac{1}{128}$	1.98×10^{-11}	3.10	4	1.11×10^{-7}	2.00	25600	1.93×10^{-11}	3.60
$\frac{1}{256}$			4	2.77×10^{-8}	2.00			
$\|p-p_h\|_{L_2}$								
$\frac{1}{8}$	8.81×10^{-3}	–	4	8.87×10^{-3}	–	100	8.89×10^{-3}	–
$\frac{1}{16}$	6.22×10^{-3}	0.50	4	6.18×10^{-3}	0.52	400	6.22×10^{-3}	0.51
$\frac{1}{32}$	4.39×10^{-3}	0.50	4	4.38×10^{-3}	0.50	1600	4.39×10^{-3}	0.50
$\frac{1}{64}$	3.1×10^{-3}	0.50	4	3.10×10^{-3}	0.50	6400	3.1×10^{-3}	0.50
$\frac{1}{128}$	2.19×10^{-3}	0.50	4	2.19×10^{-3}	0.50	25600	2.19×10^{-3}	0.50
$\frac{1}{256}$			4	1.55×10^{-3}	0.50			

(not shown in the figure). However, the $Q_3 \times Q_2$ element now also reaches the optimal convergence order for velocity (namely, 4) and pressure (i.e. 3). Moreover, to achieve this, we now also require PPC$\propto h^{-2}$ for the SolKz benchmark. All of this follows from the theoretical considerations of Section 3.5 and shows the usefulness of separating the total error into components that can be tested individually.

Finally, we have run additional tests in which the mesh cells are not aligned with the viscosity jump (by using an odd number of cells in each direction), and have confirmed previous results that a non-aligned jump limits the convergence order to $\mathcal{O}(h^1)$ for the velocity and $\mathcal{O}(h^{1/2})$ for the pressure (Kronbichler *et al.* 2012; Thielmann *et al.* 2014). The choice of finite element, particle method, and number of PPC does not influence this result and does not limit the convergence order any further.

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, and also depend on the continuity of the viscosity. In particular, we may need to grow the number of PPC as $O(h^{-1})$ or even $O(h^{-2})$ to retain the convergence order of the finite element scheme if the expected convergence order is better than $\mathcal{O}(h^2)$. This requires choosing between one of three options: (i) One needs to use a potentially very large number of PPC 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), the particle operations associated with the one time step we solve account for some 95 per cent of the overall run time. (ii) One accepts the loss of accuracy by using too few PPC, although that then calls into question the use of higher-order polynomial spaces in the Stokes discretization. (iii) One develops 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

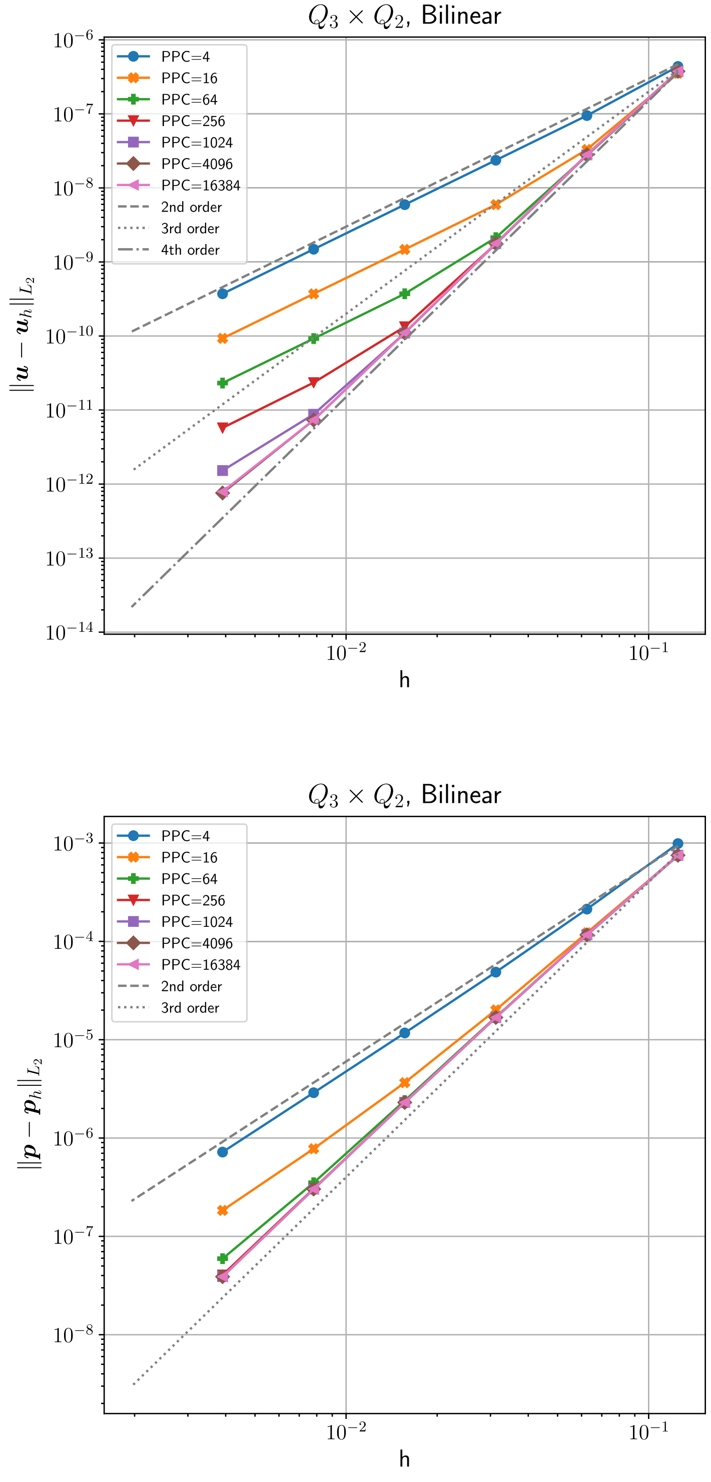

Figure 2. Velocity errors $\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L_2}$ (top) and pressure errors $\|p-p_h\|_{L_2}$ (bottom) for the SolKz benchmark for the $Q_3 \times Q_2$ element ($k = 3$) and bilinear interpolation ($r = 2$). The error is plotted as a function of both mesh resolution (h) and number of particles per cell (PPC). In contrast to Fig. 1, here we interpolate only the density from particles (i.e. we use the exact viscosity in the assembly of the finite element linear system), and we recover fourth-order convergence rate in velocity and third order in pressure.

discussed in Kronbichler *et al.* (2012); in that case, the effort for the Stokes solve and the advection solve is automatically balanced.

Finally, we want to emphasize that higher-order PIC schemes with a constant number of PPC have been successfully developed for other applications like the shallow-water equation, and the vortex formulation of the Navier–Stokes equations (Edwards & Bridson 2012). In other words, we do not argue that the dependence on PPC is an intrinsic property of *any* higher-order PIC schemes, but is rather a consequence of the algorithmic differences between our methods and those implemented in Edwards & Bridson (2012). It is apparent that determining the precise differences responsible provides a useful direction for future research.

5 TIME-DEPENDENT BENCHMARKS

The previous section presented benchmarks that assess different strategies for the transfer of information from (stationary) particle locations back to the finite element mesh, along with the error, which 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. Here we will 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 two different time-independent solutions to the Stokes equations (1) and (2), in an annulus and in a box, in which the exact density ρ is constant on streamlines. As we noted before a spatially varying viscosity could limit the convergence rate we are able to achieve with our interpolation methods, and might obscure the error of the particle advection method; consequently, we choose a constant viscosity. When one solves the Stokes equations with this setup, the solution will not change with time, since ρ is constant along streamlines, and ρ is advected along these streamlines. However, if the density (as part of the right-hand side) is inexactly interpolated from particles in each time step, and the particles are inexactly advected along with the computed velocity, then the numerical solution *will* change with time, and we can assess the accuracy of the PIC 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 PPC.

Given that we use a constant viscosity, the same considerations apply as for the SolCx benchmark in Section 4.2. Namely, one might expect that if the time discretization error is negligible, we could obtain the same convergence rates as shown in eqs (28) and (29):

$$\begin{aligned} \|\boldsymbol{u}(t) - \boldsymbol{u}_h(t)\|_{L_2} &= \mathcal{O}(h^{\min\{k+1,r+2\}}) + \mathcal{O}(h\, E(h, \mathrm{PPC}), \\ \|p(t) - p_h(t)\|_{L_2} &= \mathcal{O}(h^{\min\{k,r+1\}}) + \mathcal{O}(E(h, \mathrm{PPC})). \end{aligned} \tag{30}$$

5.1 A time-dependent benchmark in an annulus

For the first concrete realization of the approach outlined above, we need to construct a test case 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. We start by choosing the domain as a 2-D 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 eqs (1) and (2) can then be obtained by setting

$$\eta = 1, \quad \rho(r, \theta) = 48r^5, \quad \boldsymbol{g}(r, \theta) = \frac{r^3}{384}\mathbf{e}_r + \mathbf{e}_\theta, \tag{31}$$

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

$$\boldsymbol{u}(r, \theta) = 0\mathbf{e}_r - r^7\mathbf{e}_\theta, \qquad p(r, \theta) = \frac{r^9}{72} - \frac{512}{72}, \tag{32}$$

for the velocity and pressure. In other words, the flow field is circular around the centre with a velocity that varies with radius. 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 the $-e_\theta$ 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.

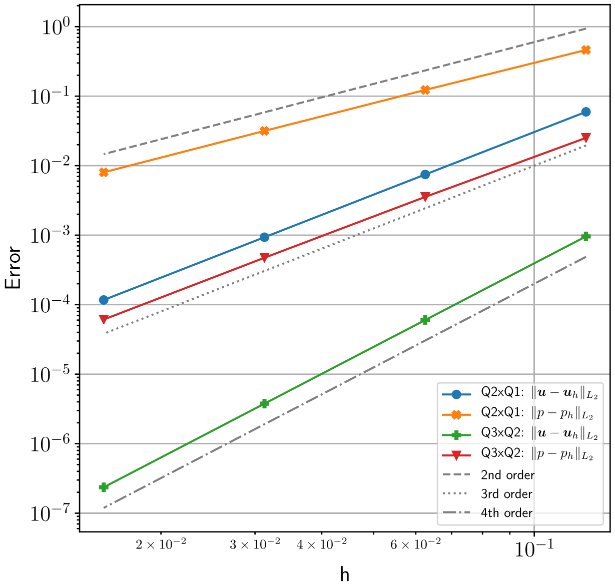

Figure 3. Convergence rates for the velocity $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2}$ and pressure $\|p - p_h\|_{L_2}$ for the time-dependent benchmark on the annulus using $Q_2 \times Q_1$ and $Q_3 \times Q_2$ element combinations, respectively. The results shown here use the exact density.

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 $t = \frac{4\pi}{27} \approx 0.0982$, which equals two complete revolutions of particles on the outer edge $r = R_2$.

5.2 Results of the time-dependent annulus benchmark

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 Fig. 3 for both $Q_2 \times Q_1$ and $Q_3 \times Q_2$ elements.

On the other hand, if the density at 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 $\boldsymbol{u}$.

Fig. 4 shows convergence results for the $Q_2 \times Q_1$ element ($k = 2$) for the velocity and pressure. As was shown in the instantaneous benchmarks above (Section 4), the orders of convergence of the

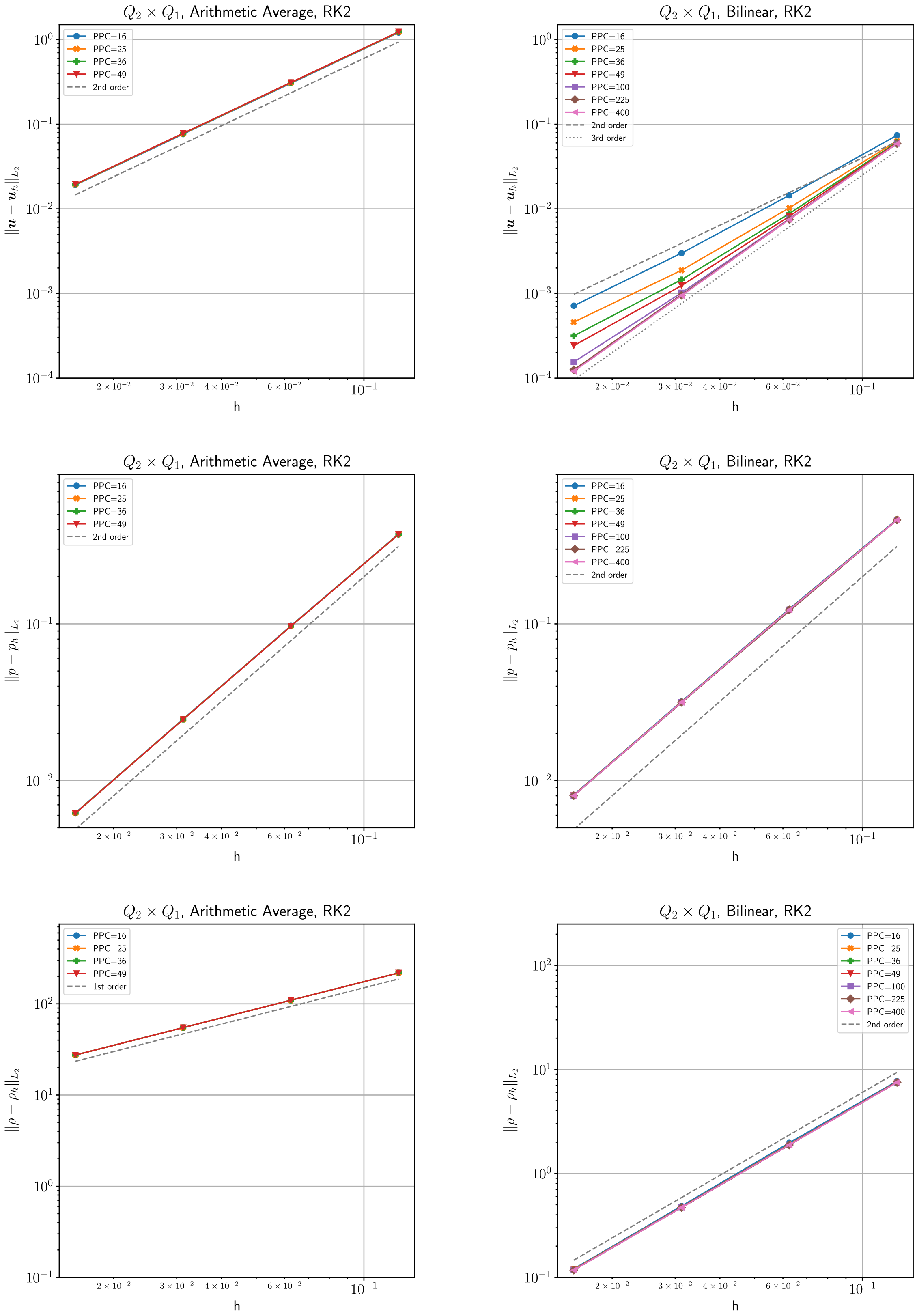

Figure 4. The convergence rate of $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2}$ (top), $\|p - p_h\|_{L_2}$ (middle) and $\|\rho - \rho_h\|_{L_2}$ (bottom) measured at $t = 4\pi/2^7$ for the time-dependent benchmark. Density is carried on particles and is interpolated as cell-wise arithmetic average ($r = 1$, left) and bilinear least-squares interpolation ($r = 2$, right). All models use a $Q_2 \times Q_1$ element ($k = 2$) and RK2 to advect particles. Note that only with bilinear least-squares interpolation and an increasing number of particles per cell (PPC) is the third-order convergence rate of velocity 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.

velocity and pressure error directly depend on the interpolation scheme, which also determines the convergence order for the error in density. The rates we observe in the figure exactly correspond to the predictions of (12) and (30) with one exception; for the velocity error with piecewise constant interpolation of the density (top left panel), we would have expected third-order convergence ($\min\{k+1, r+2\}=3$) if the temporal error were negligible, whereas we only observe second order. Furthermore, this result is independent of PPC. We are unsure about the reasons for this, but note that it is consistent with observing the same phenomenon for the SolCx benchmark which uses a similar setup (see Section 4.2).

As expected, for the bilinear interpolation ($r=2$), the optimal convergence rate is only recovered if the number of PPC is increased as the mesh is refined and the number of cells increases. This observation is consistent with our instantaneous benchmarks above, and the observation in Thielmann *et al.* (2014) that the convergence rate is suboptimal 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.

Fig. 5 shows the corresponding results for the $Q_3 \times Q_2$ element (k = 3). For lack of any new information we omit the arithmetic averaging case ($r=1$) and instead compare the RK2 integration scheme to the RK4 integrator. We start by pointing out that the integration scheme (RK2 versus RK4), the PPC (16–6400), and the finite element ($Q_2 \times Q_1$ versus $Q_3 \times Q_2$) do not change the convergence rate of the density: it remains second-order accurate. However pressure and velocity show significant differences as predicted by eq. (30). The only case where we obtain a lower convergence order than predicted by eq. (30) is the velocity error when using the RK2 integrator (top left panel), which only reduces with third-order where we would have expected fourth order ($\min\{k+1, r+2\}=4$). Interestingly, however, the expected order can be recovered by using the RK4 integrator and an increasing number of PPC (top right panel), suggesting that it is the temporal error that we neglected in deriving eq. (30) that is responsible for the reduced order.

Fig. 6 plots selected information from the two previous figures as velocity error over number of PPC for different finite elements, particle integration schemes, and mesh resolutions. In general all of the computations we made show a linear decrease of velocity error with increasing PPC (i.e. $E(h, \mathrm{PPC}) \propto (\mathrm{PPC})^{-1}$ for fixed h), which eventually transitions into a constant error at a model-specific number of PPC when the error sources (1), (2), and (4) of Section 3.5 begin to dominate over 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 can be seen 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. Therefore, the optimal number of PPC cannot be accurately determined for practical applications except by performing a convergence series test with increasing PPC for the specific problem at the final resolution.

However, we propose that it is possible to determine a nearly optimal number of PPC for most problems on a coarse resolution, and then appropriately scale this number to the target resolution, considering the convergence order of the finite element (k), the interpolation scheme (r), and the type of properties carried on the particle (density or viscosity). To illustrate this, consider the case presented in the top right panel of Fig. 4, which uses the default values for k (namely, 2), r (2), and the RK2 integration scheme of our reference implementation in ASPECT. The series of models with increasing PPC shows that when using PPC = 16, the error is already sufficiently close to the error when using larger numbers of PPC for $h=\frac{1}{8}$ to consider this number appropriate for this resolution. As determined above, the PPC-dependent error term discussed in Section 3.5 scales as $E(h, \mathrm{PPC}) = \mathcal{O}(h^2\mathrm{PPC}^{-1})$; consequently, we need to choose $\mathrm{PPC} \propto h^{-1}$ to achieve the expected velocity error convergence order of $\mathcal{O}(h^3)$. Thus, choosing PPC = 32 for $h=\frac{1}{16}$ is a natural choice, as is PPC = 64 for $h=\frac{1}{32}$. In fact, we would have done so for the figure, but our particle generation algorithm requires PPC to be the square of a natural number, which is why we chose numbers close to the natural choice. We hypothesize that the optimal values of PPC that we have found in this section will be close to optimal values for a variety of smooth problems, at least for the 2-D cases we have considered here. Therefore, while Section 3.5 provided the maximum possible convergence order one could expect, this section provided guidance on how to choose PPC to actually achieve this convergence order.

Concluding this section we want to emphasize that for higher-order methods and high mesh resolutions, choosing a higher PPC might be a more important and cheaper (though less visible) improvement in accuracy than a higher mesh resolution h. Conversely choosing a low PPC can result in a significant (but usually invisible) degradation of the accuracy of the solution.

5.3 A time-dependent benchmark in a box

For the second realization of the time-dependent benchmark approach outlined above, we choose the domain as the 2-D unit box $\Omega = (0, 1)^2$.

For this situation, we can express the equations and the solution in a Cartesian coordinate system. A solution of eqs (1) and (2) can then be obtained by setting

$$\eta = 1, \tag{33}$$

$$\rho(x, y) = \sin(\pi x)\sin(\pi y), \tag{34}$$

$$\mathbf{g}(x, y) = -4\pi^2 \frac{\cos(\pi x)}{\sin(\pi x)} \mathbf{e}_y, \tag{35}$$

where $\mathbf{e}_y$ is the vertical unit vector (pointing upwards). While the y-component of gravity becomes singular at $x=0$ and $x=1$, the right-hand side of eq. (1) only contains $\rho\mathbf{g}$ and consequently remains non-singular. We avoid accidental division by zero when assembling the equations by additionally computing $\rho\mathbf{g}$ on the particles before interpolating the product to the grid. For consistency with the annulus benchmark we also interpolate ρ when computing the interpolation error.

The Stokes system can then be solved and yields

$$\boldsymbol{u}(x, y) = \begin{bmatrix} \sin(\pi x)\cos(\pi y) \\ -\cos(\pi x)\sin(\pi y) \end{bmatrix}, \tag{36}$$

$$p(x, y) = 2\pi\cos(\pi x)\cos(\pi y) \tag{37}$$

for the velocity and pressure. The resulting flow field contains rotational and shear components and is tangential to all boundaries of the box. A detailed derivation and visualization of this solution can be found in Appendix B and Fig. B1.

All experiments for this benchmark show the error between the (stationary) exact solution $\mathbf{u}$, p, and ρ and the (time-dependent)

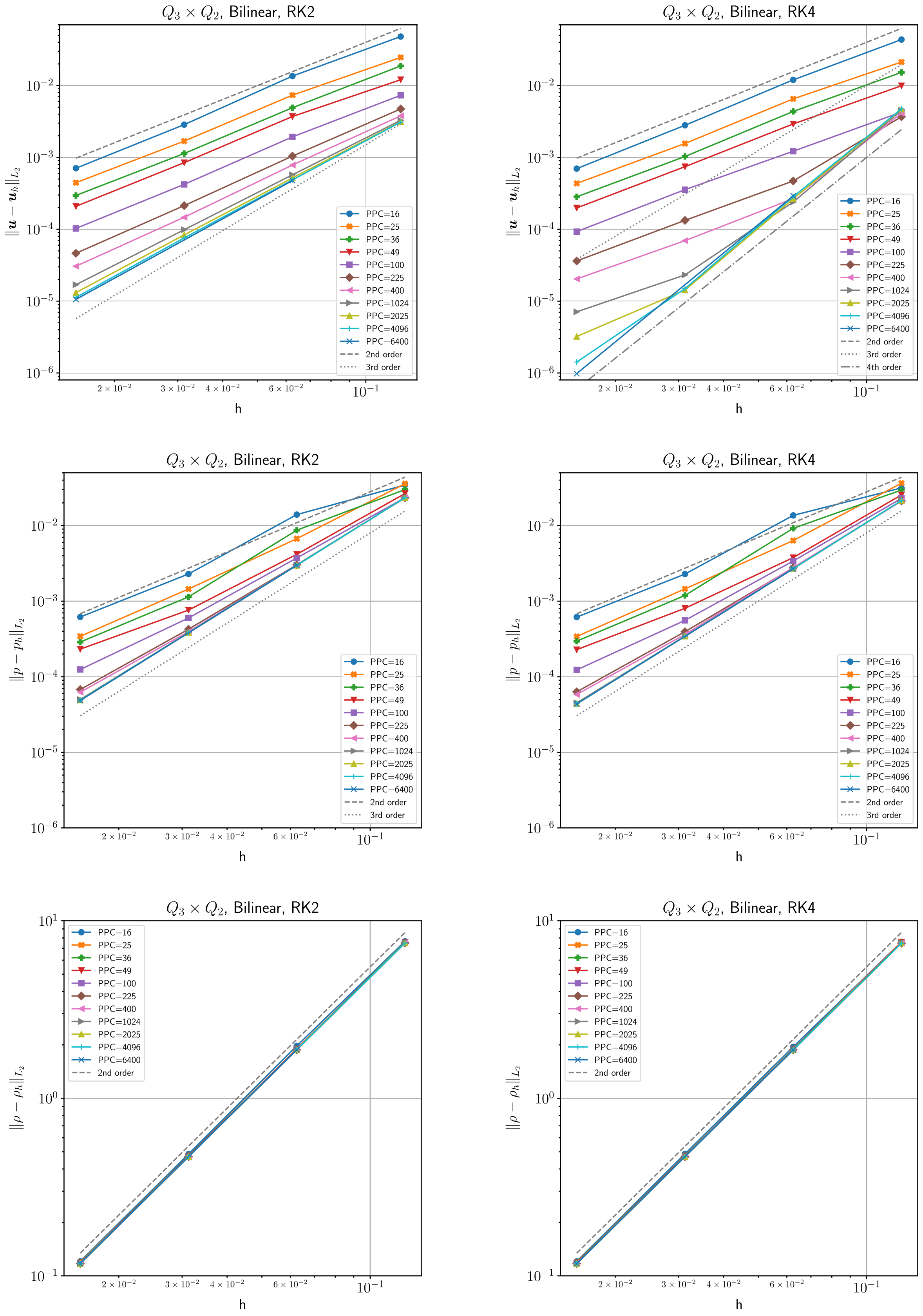

Figure 5. Panels as in Fig. 4, but for a $Q_3 \times Q_2$ element ($k = 3$). All models use the bilinear least-squares interpolation ($r = 2$). Columns represent RK2 (left) and RK4 (right) particle integration. Note that only with RK4, bilinear least-squares interpolation and an increasing number of particles per cell (PPC) is the fourth-order convergence rate of the velocity recovered. All properties with a design convergence rate higher than 2 require increasing PPC to reach their design rate, while constant PPC only allow for second-order convergence. The density is limited to second-order accuracy due to the chosen interpolation scheme ($r = 2$).

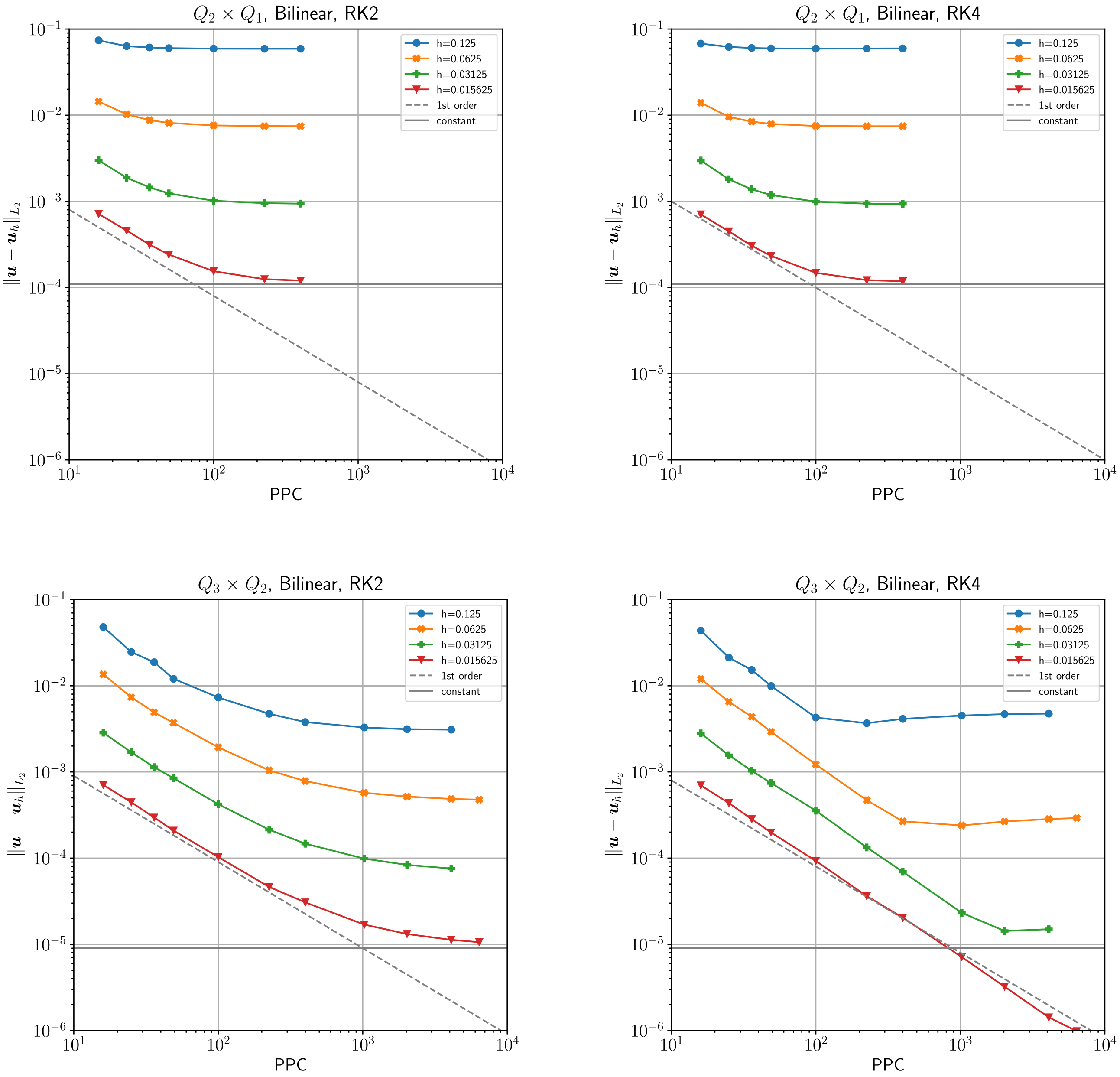

Figure 6. Convergence plots for velocity in the L_2 norm for the time-dependent annulus benchmark in dependence of the number of particles per cell (PPC). Models were computed using a $Q_2 \times Q_1$ finite element ($k = 2$, top) and a $Q_3 \times Q_2$ element ($k = 3$, bottom) respectively and particles were advected using an RK2 integration scheme (left) and an RK4 integration scheme (right). Note that the required PPC to reach the minimum error for a given mesh refinement depend on the finite element and the mesh resolution h itself. The time integration scheme only plays a role if its convergence rate is lower than the convergence rate of the velocity element.

numerical approximation $\boldsymbol{u}_h$, p_h, and ρ_h at time $t = 0.1$, which equals $\frac{1}{20}$ of a complete revolution of the centre of the model. We did not run the benchmark for a full revolution, because as described in an earlier study (Samuel 2018) the found flow field requires a particle rebalancing algorithm as regions of the model are sufficiently stretched to lose all particles. To avoid the complication of measuring the accuracy of particle splitting/merging algorithms we limited the model time.

The results of this benchmark setup are consistent with the results described for the annulus geometry in Section 5.2. The particle interpolation algorithm plays a crucial role in retaining the expected convergence order of the finite element, and the particle advection scheme can limit the convergence order if its convergence order is lower than the one of the interpolation scheme. For lack of new information the corresponding figures are presented in Appendix C. This experiment shows that the interpretations of Section 5.2 are independent of model geometry.

6 CONCLUSIONS

In this paper, we have used existing benchmarks and developed new benchmarks to measure the accuracy and convergence rate of hybrid finite element/PIC methods and provided reference results for these benchmarks obtained with the geodynamic modelling code ASPECT. In particular, we have presented the first analytical benchmarks in the computational geodynamics community that measure the accuracy and convergence order of a time-dependent flow problem in a 2-D spherical annulus or a 2-D unit box using particles to carry material properties. Since the two benchmarks are simple to derive and implement, they can be used as a convenient measure for the correctness of future implementations of similar algorithms, or as a common model for code comparisons.

Additionally, we have investigated the influence of different interpolation algorithms for transferring information from the particles to the cells and determined that in order to retain the optimal convergence rate of high-order finite element formulations, one needs to use a sufficiently high-order particle interpolation algorithm. Of course the overall convergence rate of a model is also bounded by the application in question: models with discontinuous material properties are limited to lower order accuracy if the mesh is not aligned with the discontinuities. This assertion is backed up by a theoretical analysis of the error contributions, predicting the observed convergence orders of the presented benchmark cases. Among the error contributions are (i) the discretization error due to using finite element methods on meshes of finite cell size, (ii) the error introduced by replacing the exact density and viscosity functions with ones obtained by interpolating information from particles to (low-order) polynomial spaces and (iii) the error introduced by using a finite number of PPC.

The design of better and more accurate methods than the ones we have presented here will need to address all of these error sources. For the first of the error contributions mentioned above, this might involve the use of higher-order finite element methods and/or finer meshes; both of these options are well understood and frequently used. The second error would involve interpolating data from particle locations onto polynomials of degree larger than one, for example, onto quadratic polynomials ($r = 3$) rather than the constant ($r = 1$) or linear ones ($r = 2$) used here. However, this has substantial drawbacks, for example, the fact that it is often difficult to determine in practice whether a quadratic function in two or three space dimensions is strictly positive, as one would hope the density and viscosity are; more generally, the question of minimizing unwanted variability of the interpolant needs to be addressed. For the third error source, the experiments we have shown suggest that one may need to increase the number of PPC as one refines the mesh, and we have provided guidance on how many PPC to choose for smooth problems to retain the intended convergence rate. Nevertheless, while the exact number of PPC 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 PPC) needs to be accepted; or the number of particles will need to increase substantially faster than the number of cells, resulting in computations in which operations on particles account for the vast majority of CPU cycles spent on a simulation. As shown by the error analysis, this error source does not disappear just because one uses a higher-order interpolation scheme to transfer data from particles to the mesh. As a consequence, we are not aware of a simple, cheap, and obvious method to reach high convergence rates using such PIC methods with higher-order finite elements, although it is quite possible that the methods we have presented yield an accuracy that is sufficient for practical geodynamic simulations.

ACKNOWLEDGEMENTS

All models were computed with the open source software ASPECT (Bangerth *et al.* 2018,https://aspect.geodynamics.org) version 2.1.0.pre (git commit hash a42335f33, the time-dependent box benchmark was added later in commit hash 67f2623f3). ASPECT is published under the GPL2 licence, and the necessary data to reproduce the models are included in the software. We thank the Computational Infrastructure for Geodynamics (https://geodynamics.org)—which is funded by the National Science Foundation under awards EAR-0949446 and EAR-1550901—for supporting the development of ASPECT.

WB, RG, HL and EGP jointly wrote the manuscript; EGP, RG and HL derived the time-dependent benchmark setups; HL and RG performed the computations; RG, EGP and HL analysed the benchmark results. WB developed the theoretical error analysis.

RG and WB were partially supported by the National Science Foundation under Award No. 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 Award Nos. EAR-0949446 and EAR-1550901 and The University of California—Davis. WB was also supported by the National Science Foundation under Award No. OAC-1835673.

EGP was supported by the National Science Foundation under Award No. ACI-1440811 as part of the SI2 Scientific Software Elements (SSE) program.

The computational resources were provided by the Computational Infrastructure for Geodynamics. Computations were run on the distributed computing cluster Peloton of the U.C. Davis Division of Mathematical and Physical Sciences, and the Stampede2 system at the Texas Advanced Computing Center (TACC).

REFERENCES

Bangerth, W. *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, *Numer. Math.,* **33**(2), 211–224.

Brenner, S. & Scott, R., 2007. *The Mathematical Theory of Finite Element Methods,* Vol. **15,** Springer Science & Business Media.

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, *Comput. Methods Appl. Mech. Eng.,* **32**(1-3), 199–259.

Dannberg, J. & Gassmöller, R., 2018. Chemical trends in ocean islands explained by plume–slab interaction, *Proc. Natl. Acad. Sci. USA,* **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, *Geochem. Geophys. Geosyst.,* **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, *Phys. Earth planet. Inter.,* **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, *Geochem. Geophys. Geosyst.,* **12**(7), Q07004.

Edwards, E. & Bridson, R., 2012. A high-order accurate particle-in-cell method, *Int. J. Numer. Methods Eng.,* **90**(9), 1073–1088.

Evans, M.W., Harlow, F.H. & Bromberg, E., 1957. The particle-in-cell method for hydrodynamic calculations, Tech. rep., Los Alamos National Laboratory.

Fischer, R. & Gerya, T., 2016. Early earth plume-lid tectonics: a high-resolution 3D numerical modelling approach, *J. Geodyn.,* **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, *Geochem. Geophys. Geosyst.,* **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, *Geochem. Geophys. Geosyst.,* **19**(9), 3596–3604.

Gerya, T., 2009. *Introduction to Numerical Geodynamic Modelling,* Cambridge Univ. 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, *Phys. Earth planet. Inter.,* **140**(4), 293–318.

Gilbarg, D. & Trudinger, N.S., 1983. *Elliptic Partial Differential Equations of Second Order,* 2nd edn., Springer.

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, eds Hesthaven, J. & Rønquist, E., Springer.

Harlow, F.H. & Welch, J.E., 1965. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *Phys. 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, *J. Comput. Phys.,* **39**(1), 201–225.

Ismail-Zadeh, A. & Tackley, P., 2010. *Computational Methods for Geodynamics,* Cambridge Univ. Press.

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

McNamara, A.K. & Zhong, S., 2004. Thermochemical structures within a spherical mantle: superplumes or piles? *J. geophys. Res.,* **109**(B7), 1–14.

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

Meyer, D. & Jenny, P., 2004. Conservative velocity interpolation for pdf methods, *Proc. Appl. Math. Mech.,* **4,** 466–467.

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

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

Poliakov, A. & Podladchikov, Y., 1992. Diapirism and topography, *Geophys. J. Int.,* **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, *Phys. Earth planet. Inter.,* **171,** 55–75.

Puckett, E.G., Turcotte, D.L., He, Y., Lokavarapu, H., Robey, J.M. & Kellogg, L.H., 2018. New numerical approaches for modeling thermochemical convection in a compositionally stratified fluid, *Phys. Earth planet. Inter.,* **276,** 10–35.

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

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

Robey, J.M. & Puckett, E.G., 2019. Implementation of a volume-of-fluid method in a finite element code with applications to thermochemical convection in a density stratified fluid in the earth's mantle, *Comput. Fluids,* **190**(2), 217–253.

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

Samuel, H., 2018. A deformable particle-in-cell method for advective transport in geodynamic modelling, *Geophys. J. Int.,* **214**(3), 1744–1773.

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

Tackley, P.J., 1998. Three-dimensional simulations of mantle convectionwith a thermo-chemical CMB boundary layer: D", in *The Core-MantleBoundary Region,Geodynamical Ser 28.,* pp. 231–253, eds , ,Gurnis, M., AGU .

Tackley, P.J. & King, S.D., 2003. Testing the tracer ratio method for modeling active compositional fields in mantle convection simulations, *Geochem. Geophys. Geosyst.,* **4**(4), doi:10.1029/2001GC000214.

Taylor, C. & Hood, P., 1973. A numerical solution of the Navier-Stokes equations using the finite element technique, *Comput. 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 appl. Geophys.,* **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.

van Keken, P.E., King, S.D., Schmeling, H., Christensen, U.R., Neumeister, D. & Doin, M.-P., 1997. A comparison of methods for the modeling of thermochemical convection, *J. geophys. Res.,* **102**(B10), 22 477–22 495.

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, *Geochem. Geophys. Geosyst.,* **16**(6), doi:10.1002/2015GC005824.

Zhong, S., 1996. Analytic solutions for Stokes' flow with lateral variations in viscosity, *Geophys. J. Int.,* **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) and (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

$$-\Delta \boldsymbol{u} + \nabla p = \rho\, \boldsymbol{g}, \tag{A1}$$

$$\nabla \cdot \boldsymbol{u} = 0. \tag{A2}$$

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 (A1) and (A2) then become

$$-\Big(\frac{\partial^2 u_r}{\partial r^2} + \frac{1}{r}\frac{\partial u_r}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u_r}{\partial \theta^2} - \frac{1}{r^2}u_r - \frac{2}{r^2}\frac{\partial u_\theta}{\partial \theta}\Big) + \frac{\partial p}{\partial r} = \rho\, g_r, \tag{A3}$$

$$-\Big(\frac{\partial^2 u_\theta}{\partial r^2} + \frac{1}{r}\frac{\partial u_\theta}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u_\theta}{\partial \theta^2} - \frac{1}{r^2}u_\theta + \frac{2}{r^2}\frac{\partial u_r}{\partial \theta}\Big) + \frac{1}{r}\frac{\partial p}{\partial \theta} = \rho\, g_\theta, \tag{A4}$$

$$\frac{1}{r}\frac{\partial (r u_r)}{\partial r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} = 0. \tag{A5}$$

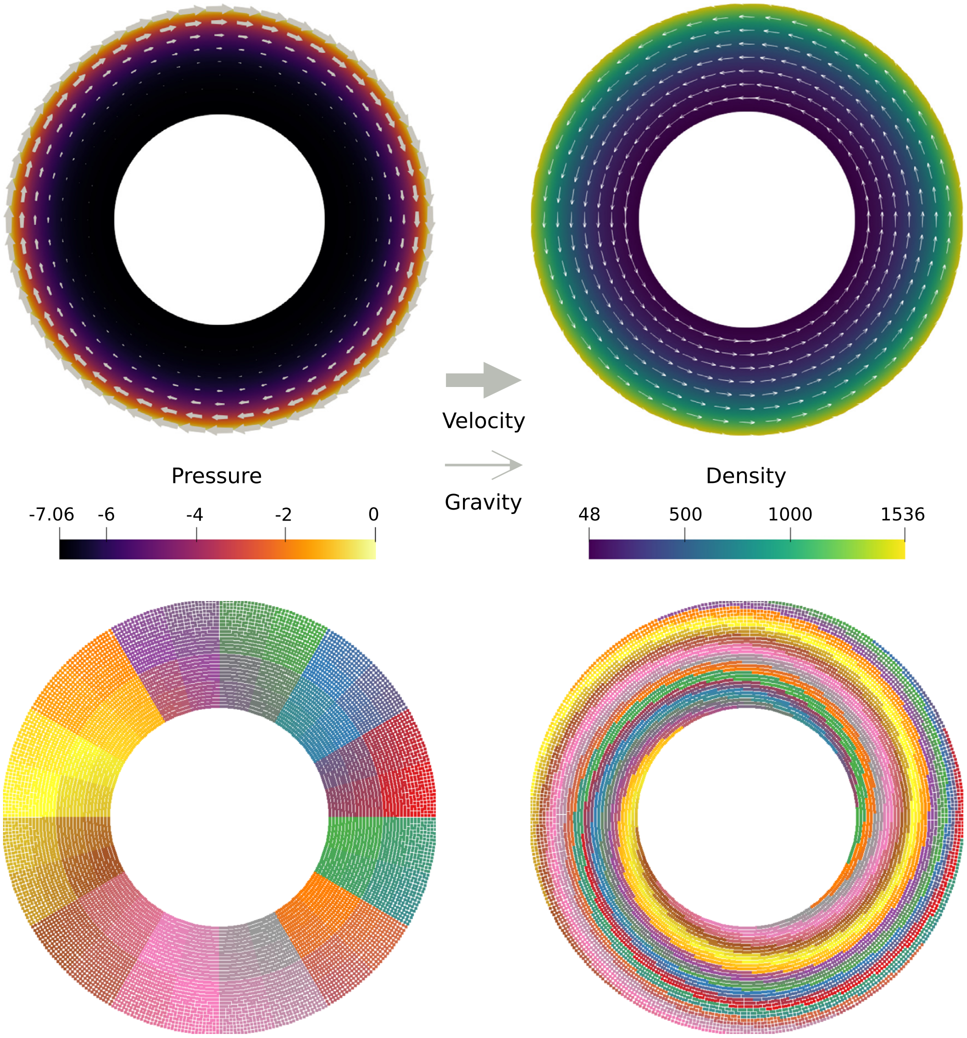

Figure A1. Solution of the annular flow benchmark. Top left: the velocity and pressure solution of the benchmark. Top right: density and gravity fields that determine the right-hand side of the Stokes system. Bottom row: initial and final particle distributions after one full revolution of the outer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

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

$$u_r = \frac{1}{r}\frac{\partial \psi}{\partial \theta} \quad \text{and} \quad u_\theta = -\frac{\partial \psi}{\partial r}. \tag{A6}$$

By this construction, the velocity field $\boldsymbol{u}$ then automatically satisfies the continuity equation (A5).

We proceed by assuming that the stream function is separable, that is, 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, eqs (A3) and (A4) become

$$-\Big(\frac{1}{r}F''G' + \frac{1}{r^2}F'G' + \frac{1}{r^3}FG' + \frac{1}{r^3}FG''' - \frac{1}{r^3}FG' + \frac{2}{r^2}F'G'\Big) = -\frac{\partial p}{\partial r} + \rho\, g_r, \tag{A7}$$

$$-\Big(-F'''G - \frac{1}{r}F''G - \frac{1}{r^2}F'G'' + \frac{1}{r^2}F'G + \frac{2}{r^3}FG''\Big) = -\frac{1}{r}\frac{\partial p}{\partial \theta} + \rho\, g_\theta. \tag{A8}$$

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

$$\rho(r, \theta) = 48r^5, \tag{A9}$$

ensuring that it is spatially variable but constant along streamlines.

Further substituting all of these expressions into eqs (A7) and (A8) then yields

$$0 = -\frac{\partial p}{\partial r} + 48r^5\, g_r, \tag{A10}$$

$$48r^5 = -\frac{1}{r}\frac{\partial p}{\partial \theta} + 48r^5\, g_\theta. \tag{A11}$$

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

$$0 = -\frac{\partial p}{\partial r} + \frac{r^8}{8}. \tag{A12}$$

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

$$p(r, \theta) = \frac{r^9}{72} - \frac{512}{72}. \tag{A13}$$

Given this pressure, the final remaining equation, eq. (A11), is

$$48r^5 = 48r^5 g_\theta. \tag{A14}$$

This results in $g_\theta = 1$.

In summary, our constructed solution is as follows:

$$\boldsymbol{u} = \begin{bmatrix} 0 \\ -r^7 \end{bmatrix}, \tag{A15}$$

$$p = \frac{r^9}{72} - \frac{512}{72}, \tag{A16}$$

$$\rho = 48r^5, \tag{A17}$$

$$\boldsymbol{g} = \begin{bmatrix} \frac{r^3}{384} \\ 1 \end{bmatrix}. \tag{A18}$$

APPENDIX B: DERIVATION OF AN INCOMPRESSIBLE STOKES SOLUTION IN A BOX

In order to derive the solution of the Stokes problem discussed in Section 5.1, we consider the Stokes equations (1) and (2) in Cartesian coordinates. As before, we only consider an isoviscous fluid with $\eta = 1$. The equations are then

$$-\Delta \boldsymbol{u} + \nabla p = \rho\, \boldsymbol{g}, \tag{B1}$$

$$\nabla \cdot \boldsymbol{u} = 0. \tag{B2}$$

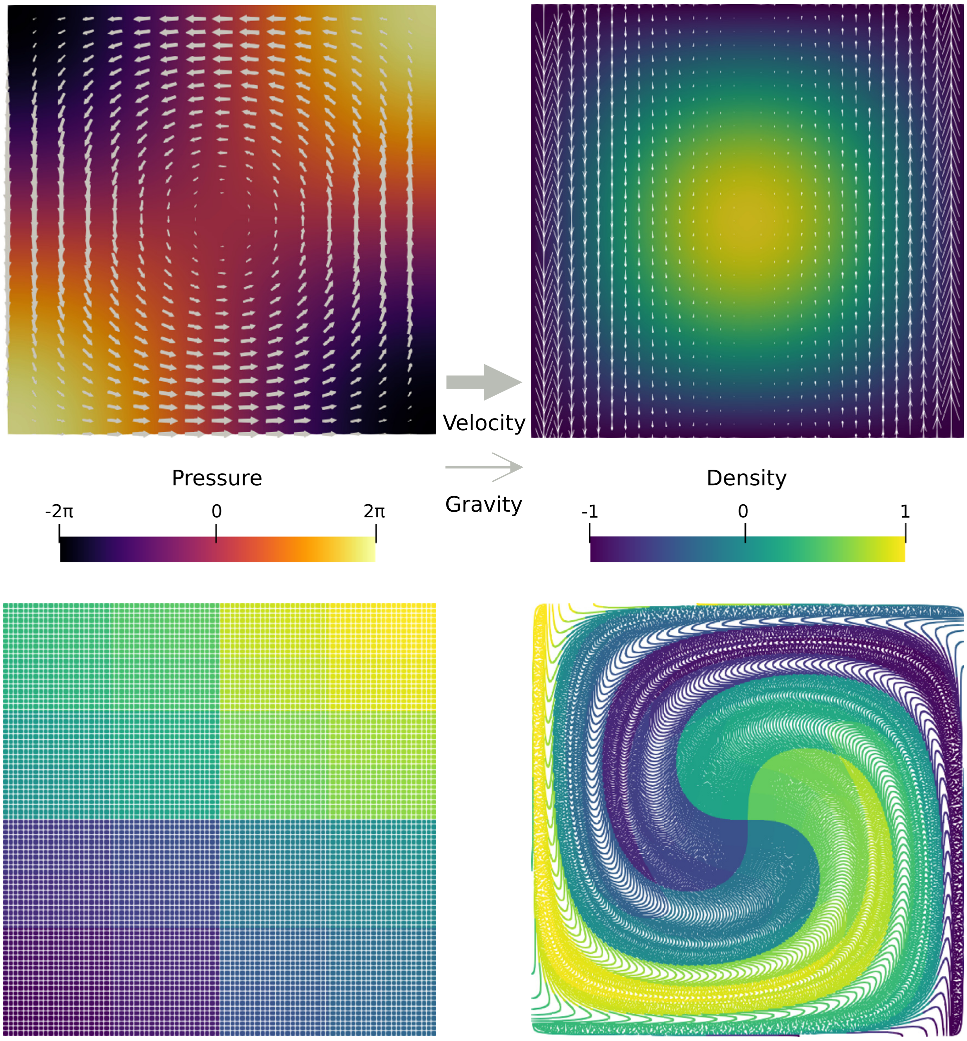

Figure B1. Solution of the rigid shear benchmark. Top left: the velocity and pressure solution of the benchmark. Top right: density and gravity fields that determine the right-hand side of the Stokes system. Bottom row: initial ($t = 0$) and final particle distributions after one full revolution of the centre ($t = 2$), coloured by particle index.

We find a solution by introducing a variation of a previously described stream function $\psi(x, y) = \frac{1}{\pi}\sin(\pi x)\sin(\pi x)$ (van Keken *et al.* 1997; Samuel 2018), and expressing the velocity through it:

$$u_x = \frac{\partial \psi}{\partial y} = \sin(\pi x)\cos(\pi y) \tag{B3}$$

$$u_y = -\frac{\partial \psi}{\partial x} = -\cos(\pi x)\sin(\pi y). \tag{B4}$$

Using this construction, the velocity field $\boldsymbol{u}$ automatically satisfies the continuity eq. (B2), is tangential to all boundaries of a unit box, and contains both shear and rotational components.

Completing the solution then requires us to find appropriate expressions for the pressure $p(x, y)$, the density $\rho(x, y)$ and the gravity vector $\boldsymbol{g} = (g_x, g_y)$ to satisfy the governing equations. Since there are two equations to satisfy (x and y components of eq. B1), but four functions to choose, we can choose two of these functions arbitrarily. As for the spherical case, because we want the benchmark to be stationary, we choose a density $\rho(x, y)$ that is constant along streamlines, and for convenience we choose $\rho(x, y) = \pi\psi(x, y) = \sin(\pi x)\sin(\pi y)$. Additionally, we arbitrarily set $g_x = 0$. Substituting all of these expressions into eq. (A1) then yields

$$2\pi^2\sin(\pi x)\cos(\pi y) + \frac{\partial p}{\partial x} = 0, \tag{B5}$$

$$-2\pi^2\cos(\pi x)\sin(\pi y) + \frac{\partial p}{\partial y} = \rho g_y, \tag{B6}$$

and integrating (B5) for x gives us the pressure:

$$p(x, y) = 2\pi\cos(\pi x)\cos(\pi y) + c. \tag{B7}$$

Similarly, differentiating (B7) and substituting in (B6) results in the y-component of gravity:

$$g_y = -4\pi^2\frac{\cos(\pi x)}{\sin(\pi x)}. \tag{B8}$$

In summary, our constructed solution is as follows:

$$\boldsymbol{u} = \begin{bmatrix} \sin(\pi x)\cos(\pi y) \\ -\cos(\pi x)\sin(\pi y) \end{bmatrix}, \tag{B9}$$

$$p = 2\pi\cos(\pi x)\cos(\pi y), \tag{B10}$$

$$\rho = \sin(\pi x)\sin(\pi y), \tag{B11}$$

$$\boldsymbol{g} = \begin{bmatrix} 0 \\ -4\pi^2\frac{\cos(\pi x)}{\sin(\pi x)} \end{bmatrix}. \tag{B12}$$

APPENDIX C: RESULTS OF THE TIME-DEPENDENT BOX BENCHMARK

Figs C1 and C2 present results for this second time-dependent benchmark, using an identical layout as for the spherical annulus case. Despite the changed geometry and different model solution, all measured convergence rates are consistent with the model of Section 5.1.

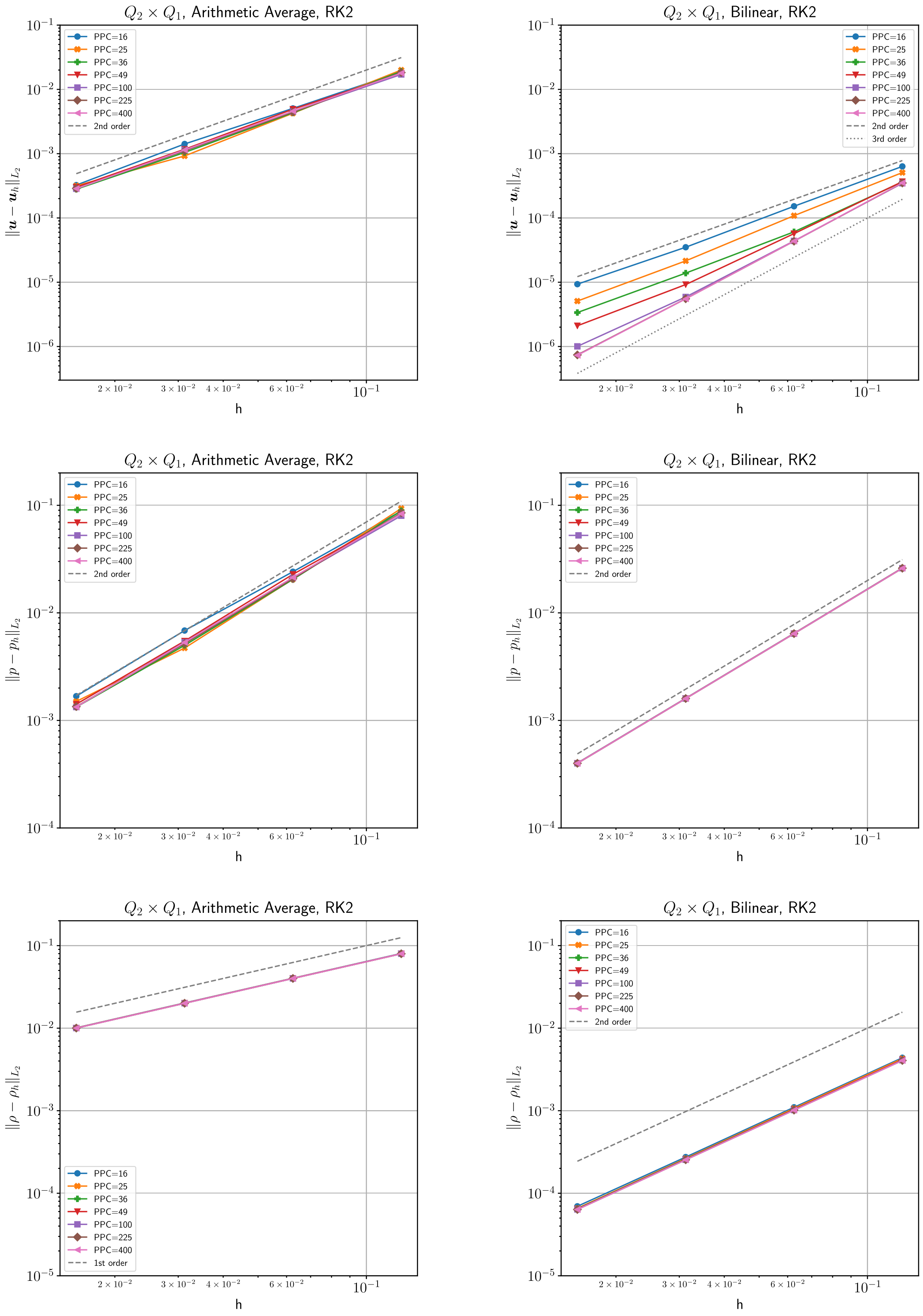

Figure C1. The convergence rate of $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L_2}$ (top), $\|p - p_h\|_{L_2}$ (middle), and $\|\rho - \rho_h\|_{L_2}$ (bottom) measured at $t = 0.1$ for the time-dependent box benchmark. Density is carried on particles and is interpolated as cell-wise arithmetic average ($r = 1$, left) and bilinear least-squares interpolation ($r = 2$, right). All models use a $Q_2 \times Q_1$ element ($k = 2$) and RK2 to advect particles. Note that only with bilinear least-squares interpolation and an increasing number of particles per cell (PPC) is the third-order convergence rate of velocity 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.

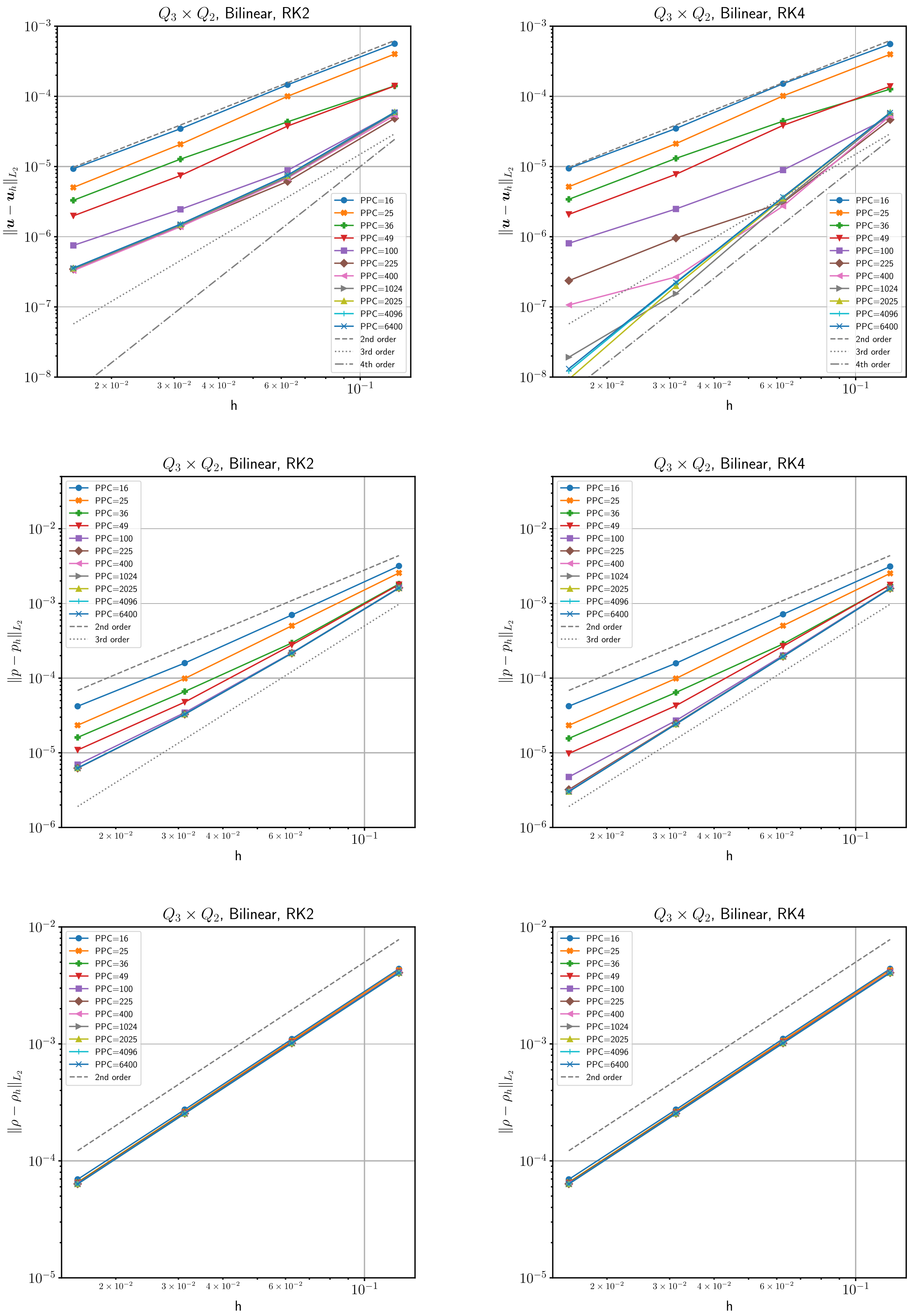

Figure C2. Panels as in Fig. C1, but for a $Q_3 \times Q_2$ element ($k = 3$). All models use the bilinear least-squares interpolation ($r = 2$). Columns represent RK2 (left) and RK4 (right) particle integration. Note that only with RK4, bilinear least-squares interpolation and an increasing number of particles per cell (PPC) is the fourth-order convergence rate of the velocity recovered. All properties with a design convergence rate higher than 2 require increasing PPC to reach their design rate, while constant PPC only allow for second-order convergence. The density is limited to second-order accuracy due to the chosen interpolation scheme ($r = 2$).