A Meshfree Approach to Non-Newtonian Free Surface Ice Flow: Application to the Haut Glacier d'Arolla

Josefin Ahlkrona* and Victor Shcherbakov†

Division of Scientific Computing, Department of Information Technology, Uppsala University, Uppsala, Sweden

April 18, 2016

Abstract

Numerical models of glacier and ice sheet dynamics traditionally employ finite difference or finite element methods. Although these are highly developed and mature methods, they suffer from some drawbacks, such as inability to handle complex geometries (finite differences) or a costly assembly procedure for nonlinear problems (finite elements). Additionally, they are mesh-based, and therefore moving domains become a challenge. In this paper, we introduce a novel meshfree approach based on a radial basis function (RBF) method. The meshfree nature of RBF methods enables efficient handling of moving margins and free ice surface. RBF methods are also highly accurate, easy to implement, and allow for reduction the computational cost associated with the linear system assembly, since stated in strong form. To demonstrate the global RBF method we model the velocity field of ice flow in the Haut Glacier d'Arolla, which is governed by the nonlinear Stokes equations. We test the method for different basal conditions and for a free moving surface. We also compare the global RBF method with its localized counterpart—the RBF partition of unity method (RBF–PUM)—that allows for a significant gain in the computational efficiency. We find that the RBF methods are well suited for ice dynamics modelling, especially the partition of unity approach.

Keywords: **Ice sheet modeling, Non-Newtonian, Free surface flow, Meshfree method, Radial basis function, Partition of unity**

1 Introduction

Glaciers and ice sheets interact with the global climate system and are two of the main contributors to sea level rise (Church et al., 2013). The dynamics of past ice masses also shaped many of our landscapes. Numerical modeling of ice sheets and glaciers is a crucial tool both to predict future evolution of ice masses, and to understand past configurations.

*Email: josefin.ahlkrona@it.uu.se.

†Email: victor.shcherbakov@it.uu.se. The list of authors is organized in the alphabetical order. Both authors contributed equally to the study.

Glacial ice slowly moves and deforms under its own weight, creeping down a valley or spreading over a continent. In this context, glacial ice can be described as a non-Newtonian, incompressible, very viscous fluid. The flow can be modeled by a set of nonlinear Stokes equations, and the movement of the ice-atmosphere interface constitutes a free surface problem. The nonlinearities, sensitivity of the free surface, large domains, and typically long time spans make numerical simulation computationally challenging.

Early ice sheet models were based on crude approximations of the governing equations, which where typically discretized using the finite difference (FD) method (Huybrechts, 1990; Greve, 1997; Calov and Marsiat, 1998). As computer power increased, more complex models gained popularity, such as the first order (FO) model, also called the Blatter–Pattyn model (Blatter, 1995; Pattyn, 2003), or even the exact model. Also the discretization schemes have changed, and an increasing number of models now use finite element (FE) methods (Zhang et al., 2011; Larour et al., 2012; Petra et al., 2012; Gagliardini et al., 2013; Brinkerhoff and Johnson, 2013; Petra et al., 2014), as these allow for complex geometries.

The finite element method, as it is implemented in ice sheet models today, does however suffer from some drawbacks. Since glaciers and ice sheets have a free surface that may change under climate conditions or due to motions of ice masses, the finite element discretization requires constant remeshing as the ice moves. This remeshing should account for both vertical movement of the surface, as well as changes in the marginal position of the ice sheet or glacier. Due to computational costs, the latter is often omitted. Another disadvantage of the finite element method is that the nonlinearity of the problem requires not only a repeated solution of a linear system (inside the nonlinear solver), but also a repeated assembly of a linear system. The cost of the assembly may severely dominate the simulation time (Ahlkrona et al., 2016).

We can avoid these issues by employing a meshfree numerical method, which does not require remeshing of the entire computational domain and avoids part of the repeated assembly phase. Rather than remeshing, a meshfree method requires placing (removing) additional computational nodes in the regions, which appear (disappear) due to displacement of the surface. For this purpose we exploit a radial basis function (RBF) method. An RBF approximation can be constructed on a set of scattered nodes and depends only upon the distances between the computational nodes. This makes the RBF approach very flexible with respect to the domain geometry and is suitable for problems defined on evolving domains.

Another valuable property of RBF methods, besides their meshfree nature, is an exponential convergence rate, if certain basis functions are used (Kansa, 1990a; Rieger and Zwicknagl, 2010). This is advantageous since computational domains for ice sheets are large. That is, in order to achieve a satisfactory resolution a large number of computational nodes has to be used with a standard method. An RBF approach can reduce the computational effort, since it requires significantly fewer nodes than standard FD or FE methods (Shcherbakov and Larsson, 2016) to achieve a similar accuracy. However, the approximation by RBF methods, when global collocation is used, results in a system of equations with a, in contrast to FD and FE methods, dense coefficient matrix. In order to overcome this issue we implement a radial basis function partition of unity method (RBF–PUM). The partition of unity based approach allows for a substansial sparsification of the coefficient matrix. This reduces the computational effort, while maintaining a similarly high convergence rate.

The first use of RBF based methods was in cartography, geodesy and digital terrain models in order to reduce errors in data interpolation (Hardy, 1971, 1990). In glaciology

radial basis functions have been used to interpolate e.g. radar data and surface elevation data (Hindmarsh et al., 2011; Sutterley et al., 2014). Later RBF methods were used for solving partial differential equations (Kansa, 1990b), which now have applications in a wide range of areas, such as fluid dynamics (Kansa, 1990b; Wendland, 2009; Fuselier et al., 2016), finance (Fasshauer et al., 2004; Safdari-Vaighani et al., 2015; Shcherbakov and Larsson, 2016), quantum mechanics (Dehghan and Shokri, 2007; Kormann and Larsson, 2012), etc. However, to the authors' knowledge, it has not been applied to ice sheet modeling so far. In this paper we aim to construct an RBF method to solve the FO model for the Haut Glacier d'Arolla, which is situated in Southern Switzerland. The method is implemented in MATLAB. Full scripts can be downloaded from `http://www.it.uu.se/research/project/rbf/software/rbf_ice`.

The paper is structured as follows. In Section 2, we present the equations governing glacial flow, both the exact form and the FO model, together with a discussion of computational aspects. We also present the Haut Glacier d'Arolla in this section. In Section 3, we introduce the RBF method, and explain the specific discretization of the Haut Glacier d'Arolla. In Section 4, we describe the setup and results for five numerical experiements, three of which test different glaciological scenarios such as frozen basal condtion, partially sliding boundary conditions and a moving ice surface, and two of which regard numerical aspects of the RBF method. We draw conclusions in Section 5.

2 Glacier Dynamics

2.1 Governing Equations

We consider a glacier on a rigid bedrock topography $z = b(x, y)$, see Fig. 1. The ice surface position $z = h(x, y, t)$ is changing in time according to the glacier velocity $\mathbf{v} = (v_x, v_y, v_z)$, which is determined by the (nonlinear) Stokes equations,

$$-\nabla p + \nabla \cdot \left(\eta(\mathbf{v})(\nabla \mathbf{v} + (\nabla \mathbf{v})^T)\right) + \rho \mathbf{g} = \mathbf{0}, \tag{1}$$

$$\nabla \cdot \mathbf{v} = 0, \tag{2}$$

where p is the pressure, ρ is the density, $\rho\mathbf{g}$ is the force of gravity, and η the viscosity. The acceleration term that would have been included in the more general Navier–Stokes equations is neglected as the Reynolds number of ice is very low. For a Newtonian fluid the viscosity is constant and then (1) would be linear. For ice, the viscosity is determined by Glen's flow law,

$$\eta = \frac{1}{2}\mathcal{A}^{-1/n} d_{\text{eff}}^{-(1-1/n)}, \tag{3}$$

such that η depends on the velocity through the second invariant d_{eff} of the strain rate tensor $\mathbf{D}$

$$d_{\text{eff}} = \sqrt{\frac{1}{2}\text{tr}\mathbf{D}^2}, \quad \mathbf{D} = \frac{1}{2}\left(\nabla \mathbf{v} + (\nabla \mathbf{v})^T\right). \tag{4}$$

We assume the so-called Glen parameter, n, to take the standard value $n = 3$. The rate factor $\mathcal{A}$ depends on temperature via an exponential function. However for simplicity we consider isothermal conditions, i.e., $\mathcal{A}$ is constant. Even for a constant viscosity, the Stokes problem is computationally challenging. For a d-dimensional problem the equation system (1)-(2) has $d+1$ unknown variables, and as the pressure is absent from (2), it becomes a saddle point problem,

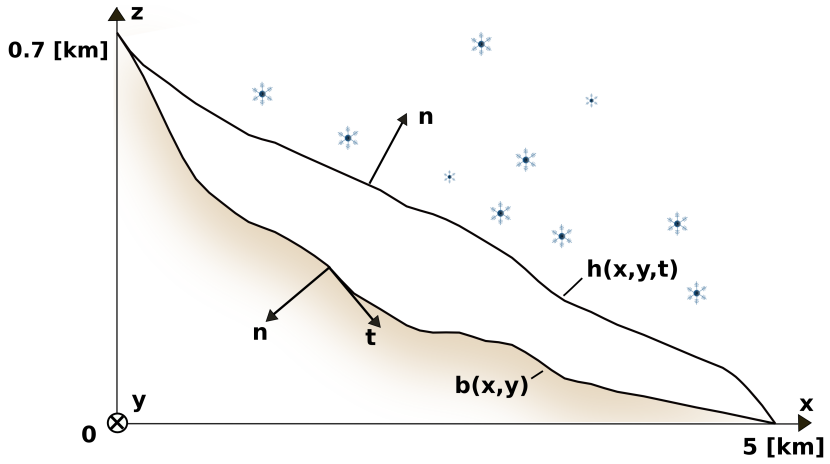

Figure 1: The Haut Glacier d'Arolla in a Cartesian coordinate system. For aesthetic reasons, the coordinates are exaggerated in the vertical direction. The ice surface position is described by $h(x, y, t)$ and the bedrock underneath is given by $b(x, y)$. The origin of the coordinate system is 2500m above sea level. The normal vector $\mathbf{n}$, is pointing outwards from the ice. In two dimensions there is one tangential vector $\mathbf{t}$, and in three dimensions two orthogonal vectors span a tangential plane.

requiring careful treatment. The nonlinear case is even more complicated, since the viscosity is dependent on the velocity field, and a nonlinear solver has to be employed. Typically the nonlinearity is resolved by fixed point or Newton iterations, repeatedly assembling and solving a linear Stokes system, and updating the viscosity in each iteration. As mentioned in the introduction, the time spent in the assembly phase in standard finite element models can dominate the solution phase (Valen-Sendstad et al., 2012; Ahlkrona et al., 2016). This is because in the finite element method, the equations are stated in weak form such that the equations are multiplied by a test function $\mathbf{q}$, and integrated over the domain Ω. The non-linear term in (1) in weak form is

$$\int_\Omega \frac{1}{2}\eta(\mathbf{v}) \left(\nabla \mathbf{v} + (\nabla \mathbf{v})^T\right) \left(\nabla \mathbf{q} + (\nabla \mathbf{q})^T\right) dV. \tag{5}$$

Since the viscoscity is inside the integral, it is inseparable from the divergence operator and symmetric gradient, such that the entire term has to be assembled together at once in each nonlinear iteration. In the RBF method, on the other hand, the equations are stated in strong form, and thus the discrete divergence operator and the symmetric gradient operator in the nonlinear term can be represented by matrices, and the viscosity by a vector. The two matrices are independent of the viscosity, such that they can be assembled outside the nonlinear iterations and only multiplied with the viscosity vector inside the nonlinear iterations.

Another challenge is the singular nature of the constitutive law (3). For zero strain rate, d, the viscosity is infinite, which may lead to ill-conditioned matrices and slowly converging iterative solvers. The problem sizes are typically large, for example, a finite element discretization of Greenland requires a mesh of at least 300 000 elements (Seddik et al., 2012). Therefore, it is crucial to address these issues efficiently. We do this partly by employing RBF methods and partly by using a higher order approximation to the Stokes model.

After the velocity field is computed, the ice surface position is updated by solving the free surface equation

$$\partial_t h + v_x|_{z=h}\partial_x h + v_y|_{z=h}\partial_y h = v_z + a_s, \tag{6}$$

where a_s is the net accumulation/ablation normal to the surface of the ice sheet, which depends on precipitation and surface air temperature. The solution of the free surface problem is not computationally demanding in itself, as it is only of dimensionality $d-1$. However, as the velocities enters as coefficients in (6) and are dependent on the geometry, and, thereby, h, the equation is nonlinear and sensitive to large time steps. Practical time steps span from weeks to a few years. Meanwhile, desired simulation time spans are large. For paleo-simulations, i.e., simulations performed in order to understand the past, the time span of interest is often a full glacial cycle ($\sim$ 100 000 years). In order to predict future sea level rise, simulations spanning a few centuries are sufficient, even though the initialization of the model often requires a prior paleo-simulation. Moreover, the mesh needs to be updated as the surface moves. Updating the vertical position of the ice surface can be done in a relatively efficient way in finite element models if extruded meshes are employed, while updating the geometry in the horizontal direction (the ice margins) is expensive. As a result, most of the ice sheet models keep margins fixed. However, by employing an RBF method we can avoid the constant remeshing of the domain, hence, significantly facilitating calculations.

2.2 Boundary Conditions

At the ice surface, h, the atmospheric stresses are negligible, implying

$$(-p\mathbf{I} + 2\eta\mathbf{D}) \cdot \mathbf{n} = 0, \tag{7}$$

where $\mathbf{I}$ is the identity matrix. The conditions at the ice-bedrock interface, b, depend on geological, hydrological and thermal conditions which are often unknown or uncertain. If the ice base is frozen, a no slip condition

$$\mathbf{v}|_b = 0 \tag{8}$$

is appropriate. If the temperature at the ice base is above the pressure melting point and there are sediments or water present at the interface the glacier will slide. For such scenarios we employ a sliding law on the form,

$$(\mathbf{v} \cdot \mathbf{t}_i)|_b = -(\mathbf{t}_i \cdot (-p\mathbf{I} + 2\eta\mathbf{D}) \cdot \mathbf{n})|_b/\beta, \quad i = 1, 2, \tag{9}$$

$$(\mathbf{v} \cdot \mathbf{n})|_b = 0, \tag{10}$$

where β describes the friction at the base. As the conditions under an ice sheet or glacier are difficult to observe, β is often determined by inverse modeling.

2.3 The First Order Stokes Model

Due to limited computer resources it has not been possible to perform simulations using the exact Stokes equations for a whole ice sheet until recently (Zhang et al., 2011; Gillet-Chaulet et al., 2012; Larour et al., 2012; Seddik et al., 2012; Petra et al., 2012, 2014). It is not yet possible to simulate very long time spans and large domains. Even on supercomputers we are limited to a couple of hundred years for Greenland, and even shorter times for Antarctica. To

avoid large systems of equations and circumvent any issues related to the saddle point nature of the Stokes problem we employ a high order approximation to the Stokes equations—the First Order (FO) Stokes model. The FO model, also called the Blatter–Pattyn model, is a high order approximation that was originally derived by Blatter (1995) and further refined by Pattyn (2003). It exploits the fact that the ice body is thin, such that the ratio between the typical thickness of the ice, $[H]$, and the typical width, $[L]$, is small. Under this assumption, $\partial_x v_z$ and $\partial_y v_z$ are neglected, and $\partial_x(\eta(\partial_z v_x + \partial_x v_z))$ and $\partial_y(\eta(\partial_z v_y + \partial_y v_z))$ are considered negligible in comparison to $\partial_z(2\eta\partial_z v_z)$. These simplifications allows for decoupling the vertical component of the momentum equation in (1) from the horizontal components. Then the horizontal velocity components (v_x, v_y) can be obtained by solving the following system of equations in only $d-1$ unknowns

$$\partial_x\left(\eta(2\partial_x v_x + \partial_y v_y)\right) + \frac{1}{2}\partial_y\left(\eta(\partial_x v_y + \partial_y v_x)\right) + \frac{1}{2}\partial_z\left(\eta\partial_z v_x\right) = \rho g \partial_x h, \tag{11}$$

$$\frac{1}{2}\partial_x\left(\eta(\partial_x v_y + \partial_y v_x)\right) + \partial_y\left(\eta(2\partial_y v_y + \partial_x v_x)\right) + \frac{1}{2}\partial_z\left(\eta\partial_z v_y\right) = \rho g \partial_y h. \tag{12}$$

Once the horizontal velocity is obtained, the vertical velocity is given by the mass conservation equation (2) and the pressure is given by

$$p = -2\eta(\partial_x v_x + \partial_y v_y) + \rho g(h - z). \tag{13}$$

The FO model was shown theoretically to be second order accurate in the aspect ratio $\delta = [H]/[L]$ by Schoof and Hindmarsh (2010). It was also compared to the exact Stokes equations in numerical experiments on the Greenland Ice Sheet by Larour et al. (2012), proving it to be highly accurate.

2.4 Haut Glacier d'Arolla

We apply the RBF method to a two dimensional simulation along the 5 km long central flow line of the Haut Glacier d'Arolla in the Swiss Alps, as it was during the Little Ice Age in 1930, see Fig. 1. This geometry was one of the test cases in a benchmark experiment designed for evaluating the accuracy of ice sheet models, the ISMIP–HOM benchmark (Pattyn et al., 2008). The initial surface and bedrock topography is available on the ISMIP–HOM website (Pattyn, 2016), with a resolution of $\delta x = 100$m. The benchmark case consists of two diagnostic simulations, one with no slip conditions at the base according to (8), and one with partial slip, such that β in (9)-(10) is

$$\begin{cases} \beta = 0, & \text{if } 2200 \leq x \leq 2500 \text{ m}, \\ \beta >> 1, & \text{elsewhere}, \end{cases} \tag{14}$$

that is, no slip condition is applied except for between $x = 2200$ and $x = 2500$ where free slip occurs. This is a challenging boundary condition, as the transition between no slip and free slip is discontinuous. In addition to these simulations, we will also let the free surface evolve in time by solving (6). The free surface will move purely due to transport, as we are not aware of any dataset describing the accumulation and ablation at Haut Glacier d'Arolla during the 1930's and therefore set $a_s = 0$, without loss of generality. The two dimensional

Table 1: Parameter values for the Haut Glacier d'Arolla simulations.

Parameter	Value
Rate factor $\mathcal{A}$	10^{-16} Pa^{-n}yr^{-1}
Density ρ	910 kg m^{-3}
Gravitational constant g	9.81 m s^{-2}
Glen's parameter n	3

flow line version of the FO equations determining the velocity field and ice surface position of the Haut Glacier d'Arolla, is

$$2\partial_x\left(\eta\partial_x v_x\right) + \frac{1}{2}\partial_z\left(\eta\partial_z v_x\right) = \rho g\partial_x h, \tag{15}$$

$$\partial_z v_z = -\partial_x v_x, \tag{16}$$

$$\partial_t h + v_x|_{z=s}\partial_x h = v_z, \tag{17}$$

where the viscosity is

$$\eta = \frac{1}{4}\mathcal{A}^{-1/3}\left((\partial_x v_x)^2 + (\partial_z v_z)^2 + \frac{1}{2}(\partial_z v_x + \partial_x v_z)^2\right)^{-1/3}. \tag{18}$$

Here the Glen parameter was set to the standard value $n = 3$. The values of all problem parameters for the Haut Glacier d'Arolla experiment follow ISMIP–HOM and are given in Table 1. The boundary conditions at the surface and the base are also simplified. By considering the FO approximation to the strain rate tensor $\mathbf{D}$ and expressing the normal and tangential vector as functions of the surface gradient $\partial_x h$ and bedrock gradient $\partial_x b$, the stress free boundary condition (7) in two dimensions reduces to

$$\eta\left(2\partial_x v_x\partial_x h - \frac{1}{2}\partial_z v_x\right) = 0. \tag{19}$$

Similarly, the slip boundary condition (9)-(10) reduces to,

$$2\eta\left(2\partial_x v_x\partial_x b - \frac{1}{2}\partial_z v_x\right) + \sqrt{(\partial_x b)^2 + 1}\beta v_x = 0. \tag{20}$$

The no slip condition is trivially $v_x = v_z = 0$.

3 Radial Basis Function Methods

In this section we introduce the concept of radial basis function (RBF) methods. In RBF methods, the domain is discretised by scattering a set of nodes. Each node is associated with a radial basis function, whose value depends only on the distance from its center. This makes RBF methods very flexible with respect to the geometry of the domain. The solution is sought as a linear combination of the basis function. Below follows the mathematical description of the RBF method in terms of general PDE problems.

Given N distinct scattered nodes $\underline{x} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$, $\mathbf{x}_i \in \Omega \subset \mathbf{R}^d$, the RBF interpolant, $\mathcal{J}_u$, of a function with values $\underline{u} = [u(\mathbf{x}_1), u(\mathbf{x}_2), \ldots, u(\mathbf{x}_N)]$ defined at those nodes takes the form

$$\mathcal{J}_u(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \phi(\|\mathbf{x} - \mathbf{x}_j\|), \quad \mathbf{x} \in \Omega, \tag{21}$$

where α_j are unknown coefficients, $\|\cdot\|$ is the Euclidean norm and $\phi(r)$ is a real-valued radial basis function (RBF). A few commonly used RBFs are presented in Table 2. In all our experiments we use Multiquadric (MQ) basis functions.

In order to determine the coefficients α_j we enforce the interpolation conditions by global collocation at the node points

$$\mathcal{J}_u(\mathbf{x}_j) = u(\mathbf{x}_j), \quad j = 1, 2, \ldots N. \tag{22}$$

As a result we obtain a linear system

$$A\underline{\alpha} = \underline{u}, \tag{23}$$

where $A_{ij} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$. For the RBFs presented in Table 2 the matrix A is always invertible, that is, a solution to system (23) can always be found. Such an RBF approximation yields exponential convergence for the basis functions from Table 2 for smooth problems (Kansa, 1990a; Rieger and Zwicknagl, 2010). However, it is important to mention that approximation properties of RBF based methods strongly depend upon the shape parameter ε, that governs the width of the basis function. A better approximation is usually observed when ε is a relatively small value. However, in this case the RBF matrix A becomes ill-conditioned, which leads to inaccurate calculations. In order to avoid this issue a stable evaluation technique, such as the Contour–Padé (Fornberg and Wright, 2004) or the RBF–QR method (Fornberg et al., 2011), can be applied.

Table 2: Commonly used radial basis functions.

RBF	$\phi(r)$
Multiquadric (MQ)	$(1 + (\varepsilon r)^2)^{1/2}$
Inverse Multiquadric (IMQ)	$(1 + (\varepsilon r)^2)^{-1/2}$
Inverse Quadratic (IQ)	$(1 + (\varepsilon r)^2)^{-1}$
Gaussian (GS)	$e^{-(\varepsilon r)^2}$

3.1 Kansa's Method

The global RBF method, also known as Kansa's method, discussed in the previous section can easily be extended to the solution of linear boundary value problems (BVPs) (Kansa, 1990b). Consider an elliptic BVP

$$\begin{cases} \mathcal{L}u(\mathbf{x}) = f(\mathbf{x}), & \mathbf{x} \in \Omega, \\ \mathcal{F}u(\mathbf{x}) = g(\mathbf{x}), & \mathbf{x} \in \partial\Omega, \end{cases} \tag{24}$$

where $\mathcal{L}$ is the interior differential operator, $\mathcal{F}$ is the boundary differential operator, and f, g are some functions. To attack this problem numerically, we scatter N discretisation nodes in Ω. Without loss of generality, we assume that the first N_I nodes belong to the interior of Ω and the last $N_B = N - N_I$ nodes belong to the boundary $\partial\Omega$. We seek a solution to system (24) in the form of the RBF interpolant (21). Collocating at the node points we obtain the following linear system

$$C\underline{\alpha} := \begin{bmatrix} L \\ F \end{bmatrix} \underline{\alpha} := \begin{bmatrix} L_{II} & L_{IB} \\ F_{BI} & F_{BB} \end{bmatrix} \begin{bmatrix} \underline{\alpha}_I \\ \underline{\alpha}_B \end{bmatrix} = \begin{bmatrix} \underline{f}_I \\ \underline{g}_B \end{bmatrix}, \tag{25}$$

where L, F, $\underline{f}$, and $\underline{g}$ are discrete representations of the continuous quantities, and the subscripts I and B denote that the quantities are evaluated in the interior and the boundary nodes, respectively. The matrices L and F consist of elements $L_{ij} = \mathcal{L}\phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$ and $F_{ij} = \mathcal{F}\phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$. For convenience, we switch from solving for $\underline{\alpha}$ to directly solving for the sought solution $\underline{u}$. By rewriting equation (23) we find

$$\underline{\alpha} = A^{-1}\underline{u}. \tag{26}$$

As it was already mentioned, A^{-1} exists for the choices of basis functions as in Table 2. Consequently, system (25) now takes the form

$$\begin{bmatrix} L_{II} & L_{IB} \\ F_{BI} & F_{BB} \end{bmatrix} \begin{bmatrix} \underline{\alpha}_I \\ \underline{\alpha}_B \end{bmatrix} = \begin{bmatrix} L_{II} & L_{IB} \\ F_{BI} & F_{BB} \end{bmatrix} A^{-1} \begin{bmatrix} \underline{u}_I \\ \underline{u}_B \end{bmatrix} = \begin{bmatrix} \underline{f}_I \\ \underline{g}_B \end{bmatrix}. \tag{27}$$

In the same manner solutions of nonlinear BVPs can be approximated by the global RBF method. Consider a nonlinear BVP

$$\mathcal{P}\left[\mathbf{x}, u(\mathbf{x}), \mathcal{D}u(\mathbf{x})\right] = 0 \quad \Rightarrow \quad \begin{cases} \mathcal{P}_1 = 0, & \mathbf{x} \in \Omega, \\ \mathcal{P}_2 = 0, & \mathbf{x} \in \partial\Omega, \end{cases} \tag{28}$$

where $\mathcal{P}_1$ is the interior nonlinear operator, $\mathcal{P}_2$ is the boundary nonlinear operator, and $\mathcal{D}$ is a shorthand notation for differential operators, such as ∂_x, ∂_y, ∇. Collocating (28) based on (21), we obtain a nonlinear system of equations

$$P(\underline{\alpha}) := \mathcal{P}\left[\underline{x}, \mathcal{J}_u(\underline{x}), \mathcal{D}\mathcal{J}_u(\underline{x})\right] = 0. \tag{29}$$

The RBF solution to the BVP (28) can then be found as $\mathcal{J}_u(\mathbf{x}; \underline{\alpha}^*)$, where $\underline{\alpha}^*$ is a root of the nonlinear system (29) that is sought by a nonlinear solver. In our particular case the nonlinear solver is a fixed point iteration method.

3.2 The Radial Basis Function Partition of Unity Method

The global RBF approximation results in a system of equations with a dense coefficient matrix. This implies that a significant computational effort is required to solve the system. In order to bypass this issue we employ a partition of unity technique, which was originally introduced for finite element methods in (Babuska and Melenk, 1997) and was later adopted to RBF methods (Cavoretto and Rossi, 2012; Safdari-Vaighani et al., 2015; Shcherbakov and Larsson, 2016). The partition of unity approach allows for significant sparsification of the coefficient

matrix. Thereby, the high computational cost associated with the global method is overcome, while a similarly high accuracy is maintained. Furthermore, a partition based formulation is well suited for parallel implementations.

We construct an open cover $\{\Omega^i\}_{i=1}^M$ of Ω, where Ω^i are overlapping patches, such that

$$\Omega \subset \bigcup_{i=1}^{M} \Omega^i. \tag{30}$$

In each patch we define a local interpolant similarly to (21)

$$\mathcal{J}_u^i(\mathbf{x}) = \sum_{j=1}^{N^i} \alpha_j^i \phi(\|\mathbf{x} - \mathbf{x}_j^i\|), \quad \mathbf{x} \in \Omega, \tag{31}$$

where N^i is the number of node points, which fall inside the i-th patch. The local interpolants are combined into a global interpolant

$$\mathcal{J}_u(\mathbf{x}) = \sum_{i=1}^{M} w^i(\mathbf{x}) \mathcal{J}_u^i(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{32}$$

where $\{w^i\}_{i=1}^M$ is a partition of unity subordinated to the open cover $\{\Omega^i\}_{i=1}^M$, that is, w^i is compactly supported on Ω^i and

$$\sum_{i=1}^{M} w^i(\mathbf{x}) = 1, \quad \mathbf{x} \in \Omega. \tag{33}$$

The partition of unity weight functions can be constructed using Shepard's method (Shepard, 1968)

$$w^i(\mathbf{x}) = \frac{\varphi^i(\mathbf{x})}{\sum_{k=1}^{M} \varphi^k(\mathbf{x})}, \quad i = 1, 2, \ldots, M, \tag{34}$$

where $\varphi^i(\mathbf{x})$ is compactly supported on Ω^i. We choose $C^2(\Omega)$ compactly supported Wendland functions (Wendland, 1995)

$$\varphi(r) = \begin{cases} (1-r)^4(4r+1), & \text{if } 0 \le r \le 1, \\ 0, & \text{if } r > 1. \end{cases} \tag{35}$$

We need at least C^2 continuity of the partition of unity weights, since the model for the glacier dynamics requires existence of the second derivative of the solution.

Collocation on (24) and (28) using (32) leads to a system of equations which, in contrast to the global method, has a sparse coefficient matrix. It is useful to bear in mind the following formulas for derivatives of the interpolant, while constructing discrete representations of differential operators.

$$\partial_x \mathcal{J}_u(\mathbf{x}) = \sum_{i=1}^{M} \left[\partial_x w^i(\mathbf{x}) \mathcal{J}_u^i(\mathbf{x}) + w^i(\mathbf{x}) \partial_x \mathcal{J}_u^i(\mathbf{x}) \right], \tag{36}$$

$$\begin{aligned} \partial_{xy}^2 \mathcal{J}_u(\mathbf{x}) = \sum_{i=1}^{M} \big[& \partial_{xy}^2 w^i(\mathbf{x}) \mathcal{J}_u^i(\mathbf{x}) + \partial_x w^i(\mathbf{x}) \partial_y \mathcal{J}_u^i(\mathbf{x}) + \\ & \partial_y w^i(\mathbf{x}) \partial_x \mathcal{J}_u^i(\mathbf{x}) + w^i(\mathbf{x}) \partial_{xy}^2 \mathcal{J}_u^i(\mathbf{x}) \big]. \end{aligned} \tag{37}$$

If (32), (36), and (37) are evaluated on a set of discrete nodes $\underline{x}$ we obtain

$$\mathcal{J}_u(\underline{x}) = \sum_{i=1}^{M} R^i W^i A^i \underline{\alpha}^i = \sum_{i=1}^{M} R^i W^i \underline{u}^i, \tag{38}$$

$$\partial_x \mathcal{J}_u(\underline{x}) = \sum_{i=1}^{M} R^i \left[W_x^i A^i + W^i A_x^i \right] \underline{\alpha}^i = \sum_{i=1}^{M} R^i \left[W_x^i A^i + W^i A_x^i \right] (A^i)^{-1} \underline{u}^i, \tag{39}$$

$$\partial_{xy}^2 \mathcal{J}_u(\underline{x}) = \sum_{i=1}^{M} R^i \left[W_{xy}^i A^i + W_x^i A_y^i + W_y^i A_x^i + W^i A_{xy}^i \right] \underline{\alpha}^i = \sum_{i=1}^{M} R^i \left[W_{xy}^i A^i + W_x^i A_y^i + W_y^i A_x^i + W^i A_{xy}^i \right] (A^i)^{-1} \underline{u}^i. \tag{40}$$

Here, R^i is a permutation operator that maps the local index set $\Xi^i = \{1, 2, \ldots, N^i\}$ corresponding to the nodes in the i-th partition into the global index set $\Xi = \{1, 2, \ldots, N\}$, and W^i, W_x^i, and W_{xy}^i are diagonal matrices with elements $w^i(\mathbf{x}_j)$, $\partial_x w^i(\mathbf{x}_j)$, and $\partial_{xy}^2 w^i(\mathbf{x}_j)$, respectively, on the diagonal. The local RBF matrix is denoted by A^i, and A_x^i and A_{xy}^i are local derivative RBF matrices with elements $\partial_x \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$ and $\partial_{xy}^2 \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$, respectively.

4 Numerical Experiments

In this section five numerical experiments are conducted. The first experiment, Experiment 0, is a pre-study performed in order to determine the optimal value of the shape parameter ε that will be used in the subsequent experiments. Experiment 1 and Experiment 2 are identical to the ISMIP–HOM E experiment (Pattyn et al., 2008). That is, the FO equations are solved for a fixed ice surface with no slip conditions (Experiment 1) or partial slip conditions (Experiment 2) at the base. In Experiment 3, a transient simulation with a moving free surface is carried out. This experiment demonstrates the advantage of applying the mesh free approach to problems which are defined in domains with moving boundaries. For these experiments the global RBF method is used. In the last experiment, the RBF–PUM is applied to the problem with a fixed surface and no slip boundary condition at the base. Then, it is compared with the global RBF method in terms of computational efficiency.

We summarize the settings for Experiment 1–Experiment 4, including the details about discretization and parameter values in Table 3.

Full versions of the MATLAB code can be downloaded from
`http://www.it.uu.se/research/project/rbf/software/rbf_ice`.

4.1 Domain Discretization

The domain discretization used for the Haut Glacier d'Arolla is presented in Fig. 2. To construct the set of computational nodes we start by defining a background grid. In general, it can be a set of scattered nodes. However, uniform or quasi-uniform node sets are typically

Table 3: Experimental setting.

	Experiment 1	Experiment 2	Experiment 3	Experiment 4
Surface BC	Fixed	Fixed	Moving	Fixed
Base BC	No Slip	Partial Slip	No Slip	No Slip
h_{fill}	0.028	0.028	0.028	0.028
Indist. threshold	0.016	0.016	0.016	0.016
Nodes	575	575	≈ 575	575
RBF	MQ	MQ	MQ	MQ
ε	5	7	5	5
Method	Global RBF	Global RBF	Global RBF	RBF-PUM
Partitions	-	-	-	20
Overlap	-	-	-	$0.26H$
Tolerance τ_{tol}	10^{-5}	10^{-5}	10^{-5}	10^{-5}
Time step	-	-	1 month	-

more suitable than purely random ones. The density of the background grid is defined by the so-called fill distance h_{fill}. In this context the fill distance is defined as the minimum distance between two points in the background grid. Since the domain is thin, the number of computational nodes required in the x-direction are larger than the number of nodes required in the z-dimension. For our experiments we found it to be optimal to use N_x nodes in the x-dimension and $N_z = N_x/4$ nodes in the z-dimension. After the background nodes are defined, we add the boundary nodes, select the background points which fall within the domain and remove external points. Also, we remove all indistinct nodes leaving just one copy (the internal nodes may coincide with the boundary nodes), as illustrated in Fig. 2. We identify the indistinct nodes as the nodes which lie in a small neighborhood of each other. In this study we define this small neighborhood as a disk with radius $\approx h_{\text{fill}}/2$ (for the values see Indisct. threshold in Table 3). Finally, the nodes that are included in the computations are the remaining internal nodes combined with the boundary nodes. This combination does not any more represent a grid and is a non-uniformly scattered node set. We use the fill distance $h_{\text{fill}} = 0.028$ throughout this paper. The background mesh contains 2626 nodes, resulting in 575 computational nodes for the geometry in Experiment 1, 2, and 4. Since the geometry in Experiment evolves, the number of nodes changes.

For the partition of unity approach we employ the same discretization, but the domain now is subdivided into partitions, which we select to be of a disk shape, see Fig. 3. We find that a partitioning with 20 patches gives sufficiently accurate results. There is a trade off between the number of partitions and the accuracy of the numerical method (Shcherbakov and Larsson, 2016). Typically, a low number of partitions leads to a higher accuracy, but also to more costly calculations because of a more dense structure of the system of equations. On the other hand, a large number of partitions leads to a worse approximation, but a higher computational efficiency. The overlap size is an extra degree of freedom, which can be flexibly chosen. However, it should not be too large as well as not too small. We find that 26% of the distance between the partition centers, H, is an appropriate ratio.

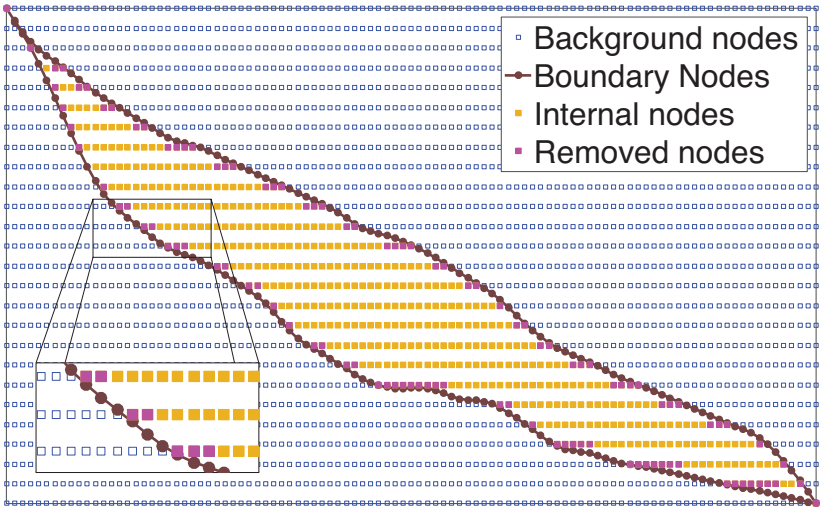

Figure 2: The discretisation of the Haut Glacier d'Arolla domain.

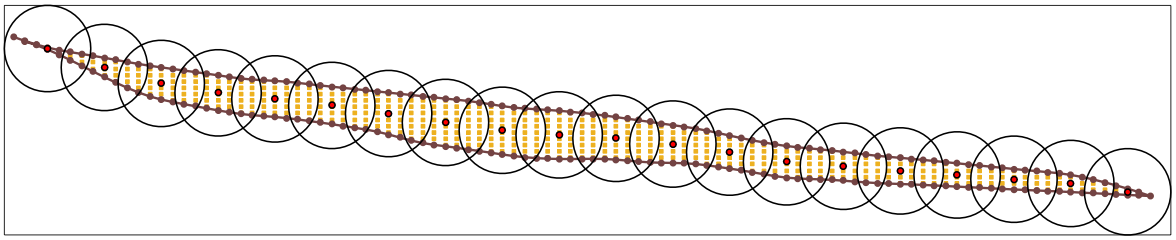

Figure 3: The partitioning of the Haut Glacier d'Arolla domain. Unlike the other figures in this paper, the coordinates are not exaggerated in the vertical z direction in order to reflect that the patches have circular form.

4.2 Experiment 0: Shape Parameter

The accuracy of an RBF approximation strongly depends upon the size of the shape parameter ε, that determines the width of the basis functions, as well as upon the fill distance h_{fill} (Fasshauer, 2007). Typically, the best approximation is achieved when the shape parameter is relatively small. However, in this case, the basis functions become flat, which leads to ill-conditioned systems of equations and unreliable solutions. Therefore, the value of the shape parameter should be chosen with a special care. Unfortunately, the question of how to correctly select the shape parameter still remains unanswered. Commonly, the following relation is used to define the shape parameter value

$$\varepsilon = C/h_{\text{fill}}, \tag{41}$$

where C is some constant. The constant C is often chosen empirically. By assuming that ε obeys (41) we make a trade-off between conditioning and accuracy such that we can expect a robust performance (for the range of problems we consider). If there exists an analytical or an accurate reference solution, one can empirically investigate the variation of the error

with respect to the shape parameter. If there is no reference solution, one can investigate the variation of the residual with respect to the shape parameter (Cheng et al., 2003). The value of the residual does not necessarily represent the qualitative characteristics of a method. However, it gives insight into which range of values of the shape parameter is suitable.

Let us consider the problem setting from Experiment 1 (see Section 4.3) with a fixed ice surface and no slip boundary condition at the base. In Fig. 4, we plot the dependence of the residual on the shape parameter ε for three different values of the fill distance h_{fill}. The plot indicates that a reasonable value for the constant $C = 0.14$. Thus, for Experiment 1 we proceed with the value of the shape parameter $\varepsilon = 0.14/h_{\text{fill}}$. For all other experiments we carry out similar tests to determine an appropriate shape parameter value. The values can be found in Table 3.

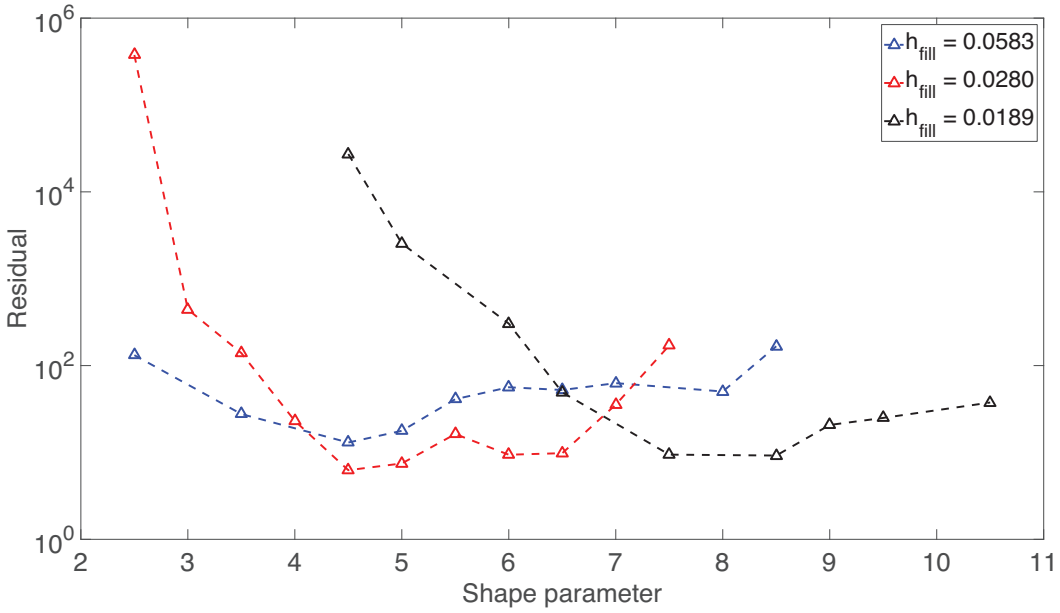

Figure 4: Experiment 0: A variation of the residual with respect to the shape parameter ε for different values of the fill distance.

We feel obliged to mention that there exist stabilisation methods, such as the Contour–Padé (Fornberg and Wright, 2004), the RBF–QR method (Fornberg et al., 2011), which allow for stable computations for small values of the shape parameter. However, these methods require an extra computational effort in order to find a stable basis, which becomes a significant part of the total effort for small problems ($<$1000 degrees of freedom). Nevertheless, stable methods are an essential aid when problems with more than 5000 degrees of freedom are solved, since otherwise the RBF matrix becomes too ill-conditioned even for large values of the shape parameter. Because we deal with a relatively small problem, the application of stable methods goes beyond the scope of our study. Note that the shape parameter optimization or the stabilization techniques do not play a dominant role in long term simulations.

4.3 Experiment 1: No Slip Boundary Conditions

The horizontal and vertical velocity fields v_x and v_z are computed by the global RBF method for a fixed ice surface and no slip boundary conditions at the base (see Fig. 5). The results are compared to a solution to the full nonlinear Stokes equations computed by the community finite element model Elmer/Ice (Gagliardini et al., 2013), see Fig. 5. In Elmer/Ice, the discretization is performed using linear elements on a triangular mesh with 1151 nodes, using a GLS type stabilization (Franca and Frey, 1992) to handle the saddle point nature of the Stokes equations. In the RBF method we avoid this issue by employing the FO model. Overall, the velocity field computed by the global RBF method is similar to the velocity field computed by Elmer/ICE. The horizontal velocity is slightly lower over the main trough around $x = 2.3$ km, and the vertical velocity shows more variance than in Elmer/Ice. However, the RBF solution remains essentially the same under a mesh refinement. Therefore we relate the difference in the solutions to the difference in the models used. For brevity of presentation, we display only solutions obtained by the RBF methods in the remaining experiments, and do not display their counterparts obtained by Elmer/Ice.

The main implementation steps are highlighted in Algorithm 1. The nonlinear material law is resolved through a nonlinear iteration (Step 9-16). Note that the main assembly is conducted outside the nonlinear solver (Step 6) in contrast to a finite element setting, where the entire assembly is executed within the nonlinear iteration since the equations are stated in weak form.

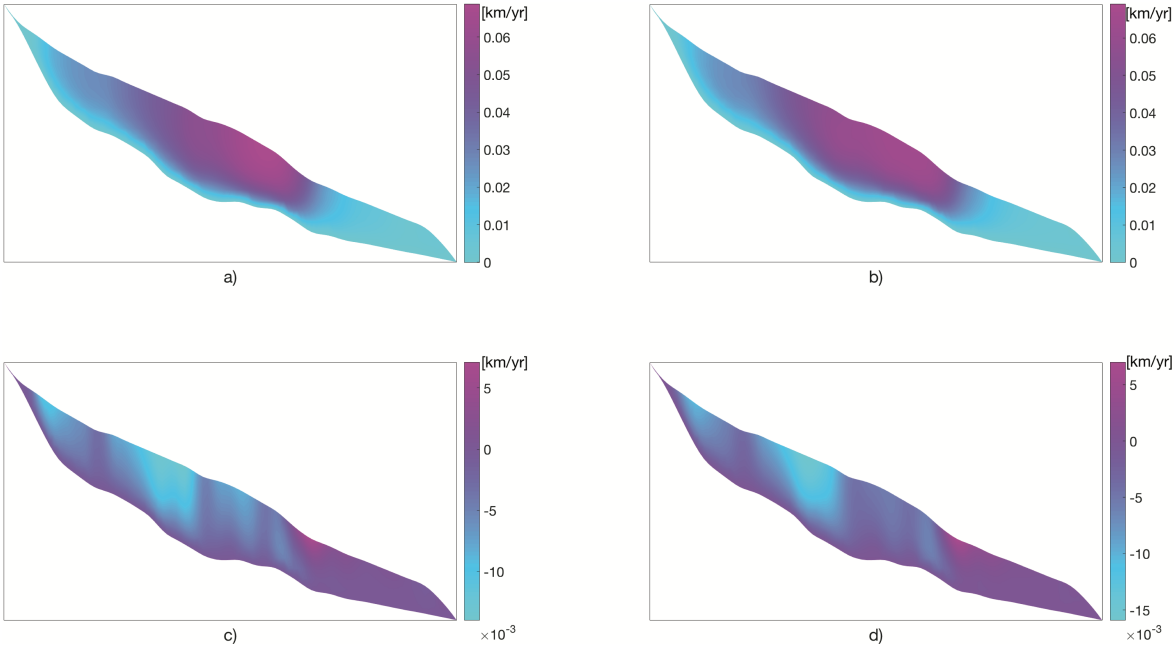

Figure 5: Experiment 1: a) Horizontal velocity v_x obtained by the global RBF method. b) Horizontal velocity v_x obtained by Elmer/Ice. c) Vertical velocity v_z obtained by the global RBF method. d) Vertical velocity v_z obtained by Elmer/Ice. No slip conditions are applied at the base.

Algorithm 1 Experiments 1 & 2

Define background node set
Select nodes which fall inside glacier domain
Remove external nodes
Find and erase indistinct nodes; leave only one
Add boundary nodes $\underline{x}_B$ to internal nodes $\underline{x}_I$
Compute distances between nodes
Assemble RBF derivative matrices
Identify initial guess $v_{x,\text{old}}$
while err $\geq \tau_{\text{tol}}$ **do** fixed point iterations
    Evaluate η as in (3)
    Define differential operator as in (15)
    Incorporate boundary condition
    Solve system of equations to obtain v_x
    Compute error err $= \|v_x - v_{x,\text{old}}\|_2$
    Update $v_{x,\text{old}} = v_x$
end while

4.4 Experiment 2: Partial Slip Boundary Conditions

In this experiment, the horizontal velocity field v_x is computed by the global RBF method when a discontinuous partial slipping at the base was permitted. The slip condition is governed by the barrier function (14). The discontinuous partial slip condition is challenging from the numerical approximation point of view. However, the RBF method allows for a sufficiently good resolution. Other types of slip boundary conditions can easily be implemented in the RBF framework.

The solution is displayed in Fig. 6, and it goes in line with the results of other studies, e.g., Perego et al. (2012) and with results obtained by Elmer/Ice. As in Experiment 1, the main implementation steps follow Algorithm 1.

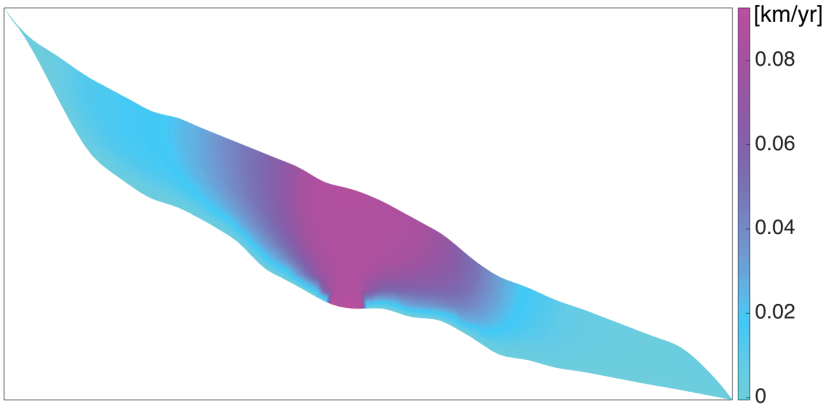

Figure 6: Experiment 2: Horizontal velocity v_x computed with the Global RBF method. Partial sliding is applied at the base.

4.5 Experiment 3: Moving Ice Surface

In Experiment 3, we consider a setting with no slip boundary condition at the base and a moving ice surface. The ice surface evolution is computed in a transient simulation over two years, from 1930 to 1932. The position of the ice surface h is updated according to (6). We use a backword Euler scheme for discretizing in time, with a time step of one month. In each time step, the surface boundary nodes are displaced while the background mesh remains fixed, see panel a) in Fig. 7. As the surface moves, some nodes that were previously outside the domain are included (marked green), and others are excluded (marked red). That is, we do not need to remesh the entire domain in every time step. The only supplementary calculations, which we have to execute, are to find distances between the displaced boundary nodes, the newly added nodes and the remaining internal nodes, and assembly the RBF matrices corresponding to those distances. This is a relatively small effort compared with a complete remeshing and full matrix assembly. Thus, such an approach significantly facilitates computations. The horizontal velocity field v_x after two years (in 1932) is presented in panel

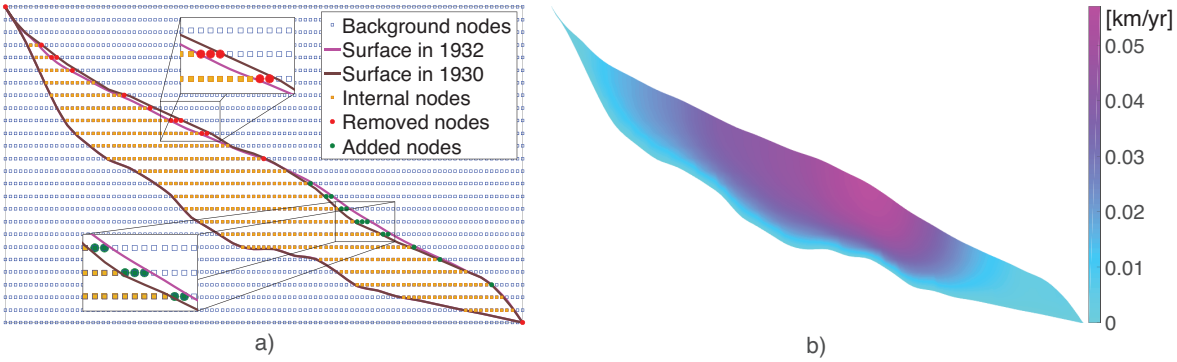

Figure 7: Experiment 3: a) The original (brown) and updated (magenta) free surface position. Green nodes are added and red removed. Note that indistinct nodes are not indicated in the figure for aesthetical reasons but they are treated as described in Section 4.1. b) The velocity field for the updated domain.

b) in Fig. 7. This test case was not a part of the original ISMIP–HOM E experiment (Pattyn et al., 2008), therefore we do not have a reference to compare with. Nevertheless, the result seems sound from the physical point of view and aligns with our intuition about the velocity distribution for this type of geometry. The key steps of the implementation are highlighted in Algorithm 2. As mentioned above, we benefit substantially from using a meshfree approach, because the assembly of the RBF derivative matrices for the immobile background nodes can conveniently be placed outside the time step iteration (Step 2-3). Within the time step iteration only a small part, which corresponds to the displaced boundary nodes and newly added internal nodes, has to be computed (Step 9-11).

4.6 Experiment 4: RBF Partition of Unity Method

In Experiment 4, we consider the same settings with no slip boundary condition and fixed surface as in Experiment 1. However, the solution method is now RBF–PUM. The horizontal

Algorithm 2 Experiment 3

1: Define background node set
2: Compute distances between nodes
3: Assemble RBF derivative matrices
4: Initialise surface position h
5: **for** each time step **do**
6: Select nodes which fall inside glacier domain
7: Find and erase indistinct nodes; leave only one
8: Pick part of matrices corresponding to internal nodes
9: Add boundary nodes to internal nodes $\underline{x} = [\underline{x}_I, \underline{x}_B]$
10: Compute distances between $\underline{x}_B$ and $\underline{x}$
11: Assemble RBF matrices for boundary nodes
12: Add RBF boundary matrices to matrices from step 8
13: Repeat steps 6-15 from Algorithm 1
14: Evaluate (16) to find vertical velocity v_z
15: Update surface position by solving (6)
16: **end for**

velocity v_x is shown in Fig. 8. If we compare this result to the solution obtained by the global RBF method, no qualitative differences can be observed. The purpose of this experiment is to

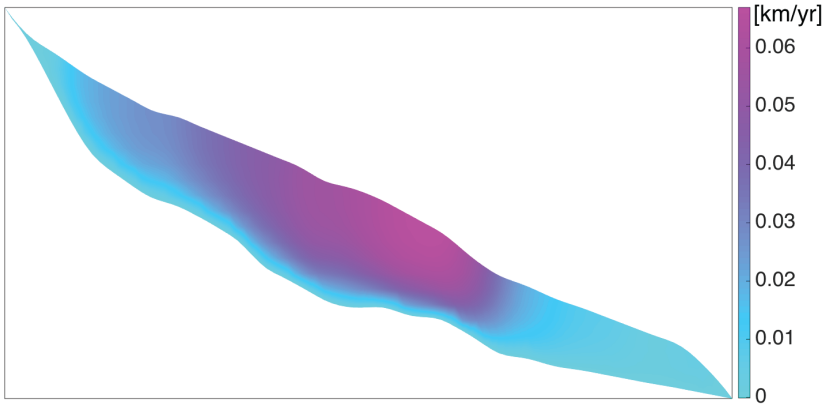

Figure 8: Experiment 4: Horizontal velocity v_x computed with the RBF-PUM method. No slip conditions are applied at the base.

illustrate the advantage of RBF–PUM over the global RBF method in terms of computational efficiency. Since RBF–PUM is a localized method, the computational effort spent on solving the system of equations is significantly less due to the sparse coefficient matrix. In Table 4 we present the execution times for the global RBF method and RBF–PUM. In order to eliminate the computer system bias we ran each code five times. From these five runs, the shortest and the longest times were taken out of consideration, and an average over the remaining three times is computed. Note that only the time of solving the nonlinear system is presented, while the time of the matrix assembly is disregarded for both methods. However, we emphasize that,

in contrast to the global RBF method, the matrix assembly in the RBF–PUM case can be implemented in parallel, since all local RBF matrices can be computed independently. In the experiment, the domain partitioning is performed such that the number of nodes per partition is kept nearly constant under refinement. That is, when the fill distance $h_{\text{fill}} = 0.0280$ the number of partitions $M = 20$, while for $h_{\text{fill}} = 0.0140$ the number of partitions $M = 40$. From

Table 4: Experiment 4: Execution times of the global RBF method and RBF–PUM for the same number of degrees of freedom.

		Run Time (s)	
h_{fill}	N_{dof}	RBF	RBF–PUM
0.0583	188	0.2076	0.1296
0.0280	575	0.8198	0.4800
0.0189	1131	4.1728	1.5946
0.0140	1905	78.851	10.313
0.0113	2814	234.90	36.824

the data presented in Table 4 we observe a substantial speedup of RBF–PUM compared to the global RBF method. Moreover, the advantage increases with the number degrees of freedom. The main implementation steps are presented in Algorithm 3.

Algorithm 3 Experiment 4

1: Repeat steps 1-4 from Algorithm 1
2: Identify partitioning
3: **for** 1 to number of partitions M **do**
4: Ascertain which nodes fall within i-th partition
5: Compute distances between nodes
6: Assemble local RBF derivative matrices
7: Compute partition of unity weight
8: Insert weighted local matrices into global matrices
9: **end for**
10: Repeat steps 6-15 from Algorithm 1

5 Conclusions

We have implemented the meshfree global RBF method and RBF–PUM for modeling of the dynamics of Haut Glacier d'Arolla. The underlying model was a higher order approximation to the governing nonlinear Stokes equations, the FO model. We considered several different glaciological scenarios: a transient simulation with a moving ice surface, frozen basal conditions and discontinuous partial slip basal conditions. We also compared the global RBF method and RBF–PUM in terms of computational efficiency and modeling capability.

We found that the RBF method is well suited for applications in glacier and ice sheet modeling, due to its meshfree nature which allows for swift handling of the free surface. In

a three dimensional simulation, the moving margin could be handled in the exact same way, which would be a major advantage as moving margins are one of the obstacles in modern ice sheet models. Moreover, the assembly phase, which is the dominant part of the computational effort in some finite element ice sheet models, may be accelerated. This is thanks to the possibility to move parts of the matrix construction outside both the nonlinear iterations, resolving the non-Newtonian nature of ice, and outside the time step iteration in a transient simulation. We also would like to stress that RBF methods can be favorable due to the ease of implementation, which may be of benefit to the glaciology community.

Another significant advantage of the RBF method is its high accuracy properties. This can be beneficial especially for larger problem simulations than the Haut Glacier d'Arolla. The cost associated with the global RBF method can be high due to dense coefficient matrix structure. This issue was overcome by RBF–PUM which is a localized method. Application of RBF–PUM leads to a sparse coefficient matrix resulting in more efficient computations. Furthermore, the RBF–PUM allows for easy implementation of parallel algorithms.

As a final remark we see no theoretical obstacles to extending the RBF models to three dimensional continental scale parallelized ice sheet simulations.

6 Acknowledgments

The authors thank Grady Wright for fruitful discussions, and Elisabeth Larsson and Per Lötstedt for reading and commenting on this manuscript. The authors also want to thank Slobodan Milovanović for help with visualizing the results. Josefin Ahlkrona was supported by the Swedish strategic research programme eSSENCE.

References

J. Ahlkrona, P. Lötstedt, N. Kirchner, and T. Zwinger. Dynamically coupling the non-linear stokes equations with the shallow ice approximation in glaciology: Description and first applications of the ISCAL method. *J. Comput. Phys.*, 308:1 –19, 2016.

I. Babuska and J. M. Melenk. The partition of unity method. *Internat. J. Numer. Methods Engrg.*, 40(4):727–758, 1997.

H. Blatter. Velocity and stress fields in grounded glaciers: a simple algorithm for including deviatoric stress gradients. *J. Glaciol.*, 41:333–344, 1995.

D. J. Brinkerhoff and J. V. Johnson. Data assimilation and prognostic whole ice sheet modelling with the variationally derived, higher order, open source, and fully parallel ice sheet model VarGlaS. *Cryosphere*, 7:1161–1184, 2013.

R. Calov and I. Marsiat. Simulations of the northern hemisphere through the last glacial-interglacial cycle with a vertically integrated and a three-dimensional thermomechanical ice-sheet model coupled to a climate model. *Ann. Glaciol.*, 27:169–176, 1998.

R. Cavoretto and A. De Rossi. Spherical interpolation using the partition of unity method: an efficient and flexible algorithm. *Appl. Math. Lett.*, 25(10):1251–1256, 2012.

A. H. D. Cheng, M. A. Golberg, E. J. Kansa, and G. Zammito. Exponential convergence and hc multiquadric collocation method for partial differential equations. *Numer. Meth. Part. D. E.*, 19(5):571–594, 2003.

J.A. Church, P.U. Clark, A. Cazenave, J.M. Gregory, S. Jevrejeva, A. Levermann, M.A. Merrifield, G.A. Milne, R.S. Nerem, P.D. Nunn, A.J. Payne, W.T. Pfeffer, D. Stammer, and A.S. Unnikrishnan. *Sea Level Change: in Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, book section 13, pages 1137–1216. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2013.

M. Dehghan and A. Shokri. A numerical method for two-dimensional Schrödinger equation using collocation and radial basis functions. *Comput. Math. Appl.*, 54(1):136–146, 2007. ISSN 0898-1221.

G. E. Fasshauer. *Meshfree approximation methods with MATLAB*, volume 6 of *Interdisciplinary Mathematical Sciences*. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2007.

G. E. Fasshauer, A. Q. M. Khaliq, and D. A. Voss. Using mesh free approximation for multi-asset American option problems. *J. Chinese Inst. Engrs.*, 27(4):563–571, 2004.

B. Fornberg and G. Wright. Stable computation of multiquadric interpolants for all values of the shape parameter. *Comput. Math. Appl.*, 48(5-6):853–867, 2004.

B. Fornberg, E. Larsson, and N. Flyer. Stable computations with Gaussian radial basis functions. *SIAM J. Sci. Comput.*, 33(2):869–892, 2011.

L. P. Franca and S. L. Frey. Stabilized finite element methods. ii: The incompressible navier-stokes equations. *Comput. Methods Appl. Mech. Eng.*, 99(2-3):209–233, September 1992.

E. J. Fuselier, V. Shankar, and G. Wright. A high-order radial basis function (RBF) Leray projection method for the solution of the incompressible unsteady Stokes equations. *Comput. Fluids*, 128:41 – 52, 2016. ISSN 0045-7930.

O. Gagliardini, T. Zwinger, F. Gillet-Chaulet, G. Durand, L. Favier, B. de Fleurian, R. Greve, M. Malinen, C. Martín, P. Råback, J. Ruokolainen, M. Sacchettini, M. Schäfer, H. Seddik, and J. Thies. Capabilities and performance of Elmer/Ice, a new generation ice-sheet model. *Geosci. Model Dev.*, 6:1299–1318, 2013.

F. Gillet-Chaulet, O. Gagliardini, H. Seddik, M. Nodet, G. Durand, C. Ritz, T. Zwinger, R. Greve, and D. G. Vaughan. Greenland ice sheet contribution to sea-level rise from a new-generation ice-sheet model. *Cryosphere*, 6:1561–1576, 2012.

R. Greve. A continuum-mechanical formulation for shallow polythermal ice sheets. *Phil. Trans. R. Soc. Lond. A*, 355:921–974, 1997.

R. Hardy. Multiquadric equations of topography and other irregular surfaces. *J. Geophyss Res.*, 76(8):1905–1915, 1971.

R. Hardy. Theory and applications of the multiquadric-biharmonic method 20 years of discovery 1968–1988. *Computers & Mathematics with Applications*, 19(8):163–208, 1990.

R. C. A. Hindmarsh, E. C. King, R. Mulvaney, H. F. J. Corr, Gisela G. Hiess, and F. Gillet-Chaulet. Flow at ice-divide triple junctions: 2. three-dimensional views of isochrone architecture from ice-penetrating radar surveys. *J. Geophys. Res.*, 116(F2):1–14, 2011.

P. Huybrechts. A 3-d model for the antarctic ice sheet: a sensitivity study on the glacial-interglacial contrast. *Clim. Dyn.*, 5(2):79–92, 1990.

E. J. Kansa. Multiquadrics—A scattered data approximation scheme with applications to computational fluid-dynamics—I surface approximations and partial derivative estimates. *Comput. Math. Appl.*, 19(8-9):127–145, 1990a. ISSN 0898-1221.

E. J. Kansa. Multiquadrics—A scattered data approximation scheme with applications to computational fluid-dynamics—II solutions to parabolic, hyperbolic and elliptic partial differential equations. *Comput. Math. Appl.*, 19(8-9):147–161, 1990b. ISSN 0898-1221.

K. Kormann and E. Larsson. An RBF-Galerkin approach to the time-dependent Schrödinger equation. Technical Report 2012-024, Department of Information Technology, Uppsala University, 2012.

E. Larour, H. Seroussi, M. Morlighem, and E. Rignot. Continental scale, high order, high spatial resolution, ice sheet modeling using the Ice Sheet System Model (ISSM). *J. Geophys. Res.*, 117:F01022, 2012.

F. Pattyn. A new three-dimensional higher-order thermomechanical ice sheet model: Basic sensitivity, ice stream development, and ice flow across subglacial lakes. *J. Geophys. Res.*, 108:2382, 2003.

F. Pattyn. Website for the Ice Sheet Model Intercomparison Project for Higher-Order ice sheet Models (ISMIP-HOM). http://homepages.ulb.ac.be/fpattyn/ismip/, January 2016.

F. Pattyn, L. Perichon, A. Aschwanden, B. Breuer, B. de Smedt, O. Gagliardini, G. H. Gudmundsson, R. Hindmarsh, A. Hubbard, J. V. Johnson, T. Kleiner, Y. Konovalov, C. Martin, A. J. Payne, D. Pollard, S. Price, M. Rückamp, F. Saito, O. Souček, S. Sugiyama, and T. Zwinger. Benchmark experiments for higher-order and full-Stokes ice sheet models (ISMIP-HOM). *Cryosphere*, 2:95–108, 2008.

M. Perego, M. Gunzburger, and J. Burkardt. Parallel finite-element implementation for higher-order ice-sheet models. *J. Glaciol.*, 58(207):76–88, 2012.

N. Petra, H. Zhu, G. Stadler, T. J. R. Hughes, and O. Ghattas. An inexact Gauss-Newton method for inversion of basal sliding and rheology parameters in a nonlinear Stokes ice sheet model. *J. Glaciol.*, 58:889–903, 2012.

N. Petra, J. Martin, G. Stadler, and O. Ghattas. A computational framework for infinite-dimensional Bayesian inverse problems, Part II: Newton MCMC with application to ice sheet flow inverse problems. *SIAM J. Sci. Comput.*, 36:A1525–A1555, 2014.

G. Rieger and B. Zwicknagl. Sampling inequalities for infinitely smooth functions, with applications to interpolation and machine learning. *Adv. Comput. Math.*, 32(1):103–129, 2010.

A. Safdari-Vaighani, A. Heryudono, and E. Larsson. A radial basis function partition of unity collocation method for convection-diffusion equations. *J. Sci. Comput.*, 64(2):341–367, 2015.

C. Schoof and R. Hindmarsh. Thin-film flows with wall slip: An asymptotic analysis of higher order glacier flow models. *Quart. J. Mech. Appl. Math.*, 63:73–114, 2010.

H. Seddik, R. Greve, T. Zwinger, F. Gillet-Chaulet, and O. Gagliardini. Simulations of the Greenland ice sheet 100 years into the future with the full Stokes model Elmer/Ice. *J. Glaciol.*, 58:427–440, 2012.

V. Shcherbakov and E. Larsson. Radial basis function partition of unity methods for pricing vanilla basket options. *Comput. Math. Appl.*, 71(1):185–200, 2016.

D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM National Conference*, ACM '68, pages 517–524, New York, NY, USA, 1968. ACM.

T. C. Sutterley, I. Velicogna, E. Rignot, J. Mouginot, T. Flament, M. R. van den Broeke, J. M. van Wessem, and C. H. Reijmer. Mass loss of the amundsen sea embayment of west antarctica from four independent techniques. *Geophys. Res. Lett.*, 41(23):8421–8428, 2014. 2014GL061940.

K. Valen-Sendstad, A. Logg, K-A. Mardal, H. Narayanan, and M. Mortensen. *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*, chapter A comparison of finite element schemes for the incompressible Navier–Stokes equations, pages 399–420. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Adv. Comput. Math.*, 4(4):389–396, 1995.

H. Wendland. Divergence-free kernel methods for approximating the Stokes problem. *SIAM J. Numer. Anal.*, 47(4):3158–3179, 2009. ISSN 0036-1429.

H. Zhang, L. Ju, M. Gunzburger, T. Ringler, and S. Price. Coupled models and parallel simulations for three-dimensional full-Stokes ice sheet modeling. *Numer. Math. Theor. Meth. Appl.*, 4:359–381, 2011.