fluids

Article

A Wavelet-Based Adaptive Finite Element Method for the Stokes Problems

Yury A. Mishin [1,*], Oleg V. Vasilyev [2] and Taras V. Gerya [1]

[1] Department of Earth Sciences, Swiss Federal Institute of Technology Zürich (ETH Zürich), CH-8092 Zürich, Switzerland; taras.gerya@erdw.ethz.ch

[2] Keldysh Institute of Applied Mathematics of Russian Academy of Sciences, Moscow 125047, Russia; oleg.v.vasilyev@gmail.com

* Correspondence: yury.mishin@gmail.com

Abstract: In this work, we present the mathematical formulation of the new adaptive multiresolution method for the Stokes problems of highly viscous materials arising in computational geodynamics. The method is based on particle-in-cell approach—the Stokes system is solved on a static Eulerian finite element grid and material properties are carried in space by Lagrangian material points. The Eulerian grid is adapted using the wavelet-based adaptation algorithm. Both bilinear (Q_1P_0, Q_1Q_1) and biquadratic (Q_2P_{-1}) mixed approximations for the Stokes system are supported. The proposed method is illustrated for a number of linear and nonlinear two-dimensional benchmark problems of geophysical relevance. The results of the adaptive numerical simulations using the proposed method are in an excellent agreement with those obtained on non-adaptive grids and with analytical solutions, while computational requirements are few orders of magnitude less compared to the non-adaptive simulations in terms of both time and memory usage.

Keywords: Stokes problem; adaptive mesh refinement; finite element method; wavelets; particle-in-cell

check for updates

Citation: Mishin, Y.A.; Vasilyev, O.V.; Gerya, T.V. A Wavelet-Based Adaptive Finite Element Method for the Stokes Problems. *Fluids* **2022**, *7*, 221. https://doi.org/10.3390/fluids7070221

Academic Editor: Mehrdad Massoudi

Received: 23 May 2022
Accepted: 25 June 2022
Published: 30 June 2022

1. Introduction

Practical problems of computational geodynamics require high-resolution numerical models. Computational requirements of such models can be very high, and thus powerful supercomputers are employed routinely nowadays. However, to utilize these machines more efficiently and to be able to model truly challenging problems like global mantle convection at numerical resolutions resolving 1 km scales, new advanced computational methods are needed (e.g., [1–3].

In this work we present the novel adaptive multiresolution method for mechanical Stokes problems arising in computational geodynamics. The Stokes system of equations is solved on a static Eulerian finite element grid which is adapted in space based on chosen adaptation criterion (viscosity field, strain rate field, etc.), and material properties and history are carried by Lagrangian particles. Fully automatic grid adaptation is implemented using an original algorithm based on wavelet analysis [4–8]. The adaptation is supported for both bilinear (Q_1P_0, Q_1Q_1) and biquadratic (Q_2P_{-1}) finite element grids.

The work is divided into 2 parts. In the first part (Sections 2–7), the mathematical formulation and implementation details of the method are given. In the second part (Section 8), we discuss the results of some numerical benchmarks.

2. Governing Equations

We are interested in very slow deformations of highly viscous materials, typical for mantle convection problems. In problems of geophysical relevance due to drastically different convection and viscous time scales, both material properties and scalar quantities are advected as described in Section 4, while viscous forces are assumed to be in equilibrium.

In this case all inertial forces can be neglected and such quasi-static deformation process can be described as an evolution through a succession of equilibria:

$$\frac{\partial \sigma_{ij}}{\partial x_j} + \rho f_i = 0 \tag{1}$$

where σ_{ij} is the total stress tensor component, f_i is the external force per unit mass vector component, x_j indicates the spatial coordinate and ρ is the density.

The medium is assumed to be incompressible. Thus, Equation (1) is solved subject to incompressibility condition:

$$\dot{\varepsilon}_{ii} = 0 \tag{2}$$

where $\dot{\varepsilon}_{ij}$ is the strain rate tensor component defined as:

$$\dot{\varepsilon}_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) \tag{3}$$

where v_i and v_j indicate the velocities.

The relation between dynamic (σ) and kinematic ($\dot{\varepsilon}$) quantities is given by rheological constitutive equation. For viscous incompressible medium it is expressed as (e.g., [9]):

$$\sigma_{ij} = -p\delta_{ij} + 2\eta\dot{\varepsilon}_{ij} \tag{4}$$

where p is the pressure, δ_{ij} is the Kronecker delta, η is the viscosity.

Equations (1)–(4) form the incompressible Stokes system. Even though the adaptive finite element method is generalizable to three dimensions, in this work both the method formulation and its applications are limited to two-dimensional problems. Thus we assume strain tensor components $\varepsilon_{zz}, \varepsilon_{xz}, \varepsilon_{zx}, \varepsilon_{yz}, \varepsilon_{zy}$ to be equal 0 (plane strain assumption) and the system we solve takes the form:

$$\begin{aligned} 2\frac{\partial}{\partial x}\left(\eta\frac{\partial v_x}{\partial x}\right) + \frac{\partial}{\partial y}\left(\eta\frac{\partial v_x}{\partial y} + \eta\frac{\partial v_y}{\partial x}\right) - \frac{\partial p}{\partial x} &= -\rho f_x \\ 2\frac{\partial}{\partial y}\left(\eta\frac{\partial v_y}{\partial y}\right) + \frac{\partial}{\partial x}\left(\eta\frac{\partial v_y}{\partial x} + \eta\frac{\partial v_x}{\partial y}\right) - \frac{\partial p}{\partial y} &= -\rho f_y \\ \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} &= 0 \end{aligned} \tag{5}$$

Viscoplastic rheology is implemented by using a non-linear effective viscosity in (5). A material deforms viscously when stresses fall below its yield strength σ_{yield}, but plastic deformations occur as soon as σ_{yield} is reached, and the effective viscosity is adjusted to ensure material strength σ_{yield} is never exceeded (e.g., [10–12]):

$$\eta_{\text{eff}} = \min\{\eta_{\text{creep}}, \frac{\sigma_{\text{yield}}}{2\dot{\varepsilon}_{II}}\} \tag{6}$$

Here η_{creep} is the creep viscosity, which is a function of strain rate and other material parameters in general cases, and is a constant if viscous deformations are linear (e.g., [9]), σ_{yield} is the yield strength of a material and $\dot{\varepsilon}_{II}$ is the second invariant of the strain rate tensor defined as:

$$\dot{\varepsilon}_{II} = \sqrt{\frac{1}{2}\dot{\varepsilon}_{ij}\dot{\varepsilon}_{ij}} \tag{7}$$

The yield strength σ_{yield} is calculated according to the Drucker-Prager plasticity model, which in general form is expressed as (e.g., [13]):

$$\sigma_{\text{yield}} = \alpha p + \beta \tag{8}$$

where p is the pressure, α and β are material parameters. The following relationships for α and β are used [14]:

$$\alpha = \sin(\phi), \qquad \beta = c\cos(\phi) \tag{9}$$

where c and ϕ are cohesion and friction angle of a material respectively.

3. Finite Element Discretization of the Stokes System

The Stokes system of Equation (5) is solved on an Eulerian grid using Galerkin mixed finite element discretization technique. In this way velocities and pressure are approximated as:

$$\begin{gathered} v_x(x,y) \simeq \sum_{i=1}^{n} N_i(x,y)v_{xi} \qquad v_y(x,y) \simeq \sum_{i=1}^{n} N_i(x,y)v_{yi} \\ p(x,y) \simeq \sum_{i=1}^{m} M_i(x,y)p_i \end{gathered} \tag{10}$$

where n and m are numbers of velocity and pressure nodes, N_i and M_i are locally defined basis functions, v_{xi}, v_{yi} and p_i are nodal values of velocities and pressure which have to be found.

In this work we employ 3 different quadrilateral element types (i.e., combinations of basis sets $\{N_i\}$ and $\{M_i\}$): bilinear velocity—constant discontinuous pressure $\mathrm{Q_1P_0}$, bilinear velocity—bilinear continuous pressure $\mathrm{Q_1Q_1}$ with stabilization, biquadratic velocity—linear discontinuous pressure $\mathrm{Q_2P_{-1}}$ (Figure 1; for exact expressions of basis functions see e.g., [15]). Probably the most popular in the community is the $\mathrm{Q_1P_0}$ element, primary due to its low computational costs. However, this approximation is unstable as it does not satisfy *inf-sup* condition and so can lead to highly inaccurate results under some circumstances [16,17]. The $\mathrm{Q_1Q_1}$ approximation is also inherently unstable, but it can be stabilized by very efficient and simple to implement method proposed recently [18,19]. Finally, the $\mathrm{Q_2P_{-1}}$ element is stable and known to be robust [17], but associated computational cost is significantly higher compared to bilinear approximations $\mathrm{Q_1P_0}$ or $\mathrm{Q_1Q_1}$.

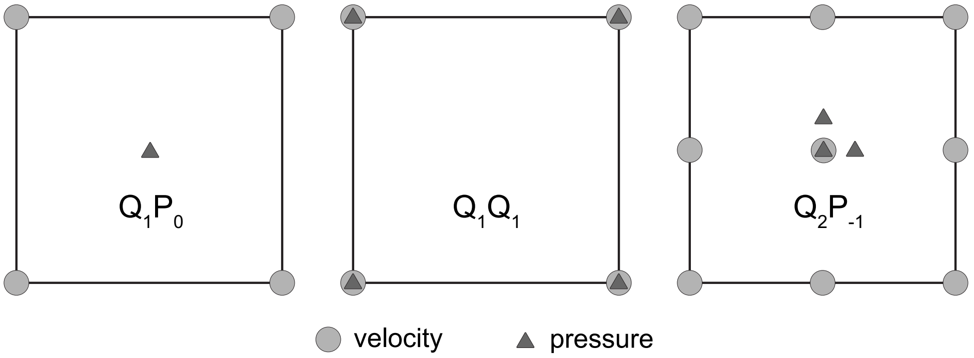

Figure 1. Mixed velocity-pressure elements.

Substitution of (10) into (5) and application of Galerkin discretization procedure yield the Stokes system of equations in the matrix form (e.g., [20]:

$$\begin{bmatrix} \mathbf{A} & \mathbf{Q} \\ \mathbf{Q}^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{0} \end{bmatrix} \tag{11}$$

where stiffness matrix $\mathbf{A}$, gradient operator matrix $\mathbf{Q}$, right hand side vector $\mathbf{F}$ and vectors of unknowns $\mathbf{v}$ and $\mathbf{p}$ are:

$$\begin{gathered} \mathbf{A} = \int_\Omega \mathbf{B}^{\mathrm{T}}\mathbf{D}\mathbf{B}dxdy \qquad \mathbf{Q} = -\int_\Omega \mathbf{K}^{\mathrm{T}}\mathbf{H}dxdy \qquad \mathbf{F} = \int_\Omega \mathbf{R}^{\mathrm{T}}dxdy \\ \mathbf{v} = \begin{bmatrix} v_{x1} & v_{y1} & v_{x2} & v_{y2} & \ldots & v_{xn} & v_{yn} \end{bmatrix}^{\mathrm{T}} \qquad \mathbf{p} = \begin{bmatrix} p_1 & p_2 & \ldots & p_m \end{bmatrix}^{\mathrm{T}} \end{gathered} \tag{12}$$

Here vectors and matrices **B**, **D**, **K**, **H** and **R** are:

$$\begin{aligned}
&\mathbf{B} = \begin{bmatrix} \Lambda N_1 & \Lambda N_2 & \dots & \Lambda N_n \end{bmatrix} \qquad \text{where} \quad \Lambda = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \\
&\mathbf{D} = \eta \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&\mathbf{K} = \begin{bmatrix} \nabla N_1 & \nabla N_2 & \dots & \nabla N_n \end{bmatrix} \qquad \text{where} \quad \nabla = \begin{bmatrix} \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix} \\
&\mathbf{H} = \begin{bmatrix} M_1 & M_2 & \dots & M_m \end{bmatrix} \\
&\mathbf{R} = \begin{bmatrix} N_1 X & N_2 X & \dots & N_n X \end{bmatrix} \qquad \text{where} \quad X = \rho \begin{bmatrix} f_x & f_y \end{bmatrix}
\end{aligned} \tag{13}$$

In case of Q_1Q_1 approximation, the system (11) has to be further modified to include stabilization matrix [18,19]:

$$\begin{bmatrix} \mathbf{A} & \mathbf{Q} \\ \mathbf{Q}^{\mathrm{T}} & -\mathbf{V} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{0} \end{bmatrix} \tag{14}$$

where stabilization matrix **V** is:

$$\mathbf{V} = \frac{1}{\eta} \int_{\Omega} \bar{\mathbf{H}}^{\mathrm{T}} \bar{\mathbf{H}} dx dy \tag{15}$$

where vector $\bar{\mathbf{H}}$ is:

$$\bar{\mathbf{H}} = \begin{bmatrix} M_1 - \frac{1}{4} & M_2 - \frac{1}{4} & \dots & M_m - \frac{1}{4} \end{bmatrix} \tag{16}$$

Since basis functions N_i and M_i are locally defined, matrices **A**, **Q** and **V** in (11) and (14) are sparse and in practise are built by assembling contributions from individual elements.

The problem (11) is solved by augmented Lagrangian iterative method (e.g., [21,22]). First, the system (11) is amended by addition of the term $-\frac{1}{k}\mathbf{M}$ to the both sides of the second equation, and velocity Schur complement is computed as:

$$\hat{\mathbf{A}} = \mathbf{A} + k\mathbf{Q}\mathbf{M}^{-1}\mathbf{Q}^{\mathrm{T}} \tag{17}$$

where k is the penalty parameter and **M** is the mass matrix:

$$\mathbf{M} = \int_{\Omega} \mathbf{H}^{\mathrm{T}} \mathbf{H} dx dy \tag{18}$$

Schur complement $\hat{\mathbf{A}}$ is then factorized and the iterative scheme is employed until sufficient level of incompressibility defined by ϵ_{inc} is reached. The matrix $\hat{\mathbf{A}}$ is symmetric positive definite, and so the Cholesky factorization is employed:

```
$\hat{\mathbf{A}} \to \mathbf{L}\mathbf{L}^{\mathrm{T}}$
DO
    $\mathbf{v}^{\mathrm{i}} = \mathbf{L}^{-\mathrm{T}}(\mathbf{L}^{-1}(\mathbf{F} - \mathbf{Q}\mathbf{p}^{\mathrm{i}}))$
    $\mathbf{d}^{\mathrm{i}} = \mathbf{M}^{-1}\mathbf{Q}^{\mathrm{T}}\mathbf{v}^{\mathrm{i}}$                    (19)
    $\mathbf{p}^{\mathrm{i}+1} = \mathbf{p}^{\mathrm{i}} + k\mathbf{d}^{\mathrm{i}}$
    $\mathrm{i} = \mathrm{i} + 1$
UNTIL $\|\mathbf{d}^{\mathrm{i}}\|_{\infty} > \epsilon_{\mathrm{inc}}$
```

Augmented Lagrangian method is very efficient, but it implies inversion of the mass matrix **M**. It is easy to compute the inverse $\mathbf{M}^{-1}$ when pressure approximation is discontinuous, which is true when Q_1P_0 or Q_2P_{-1} elements are employed. However, in case of Q_1Q_1 element we deal with the system (14), and it is not feasible to apply the method (19) since pressure approximation is continuous. The system (14) is symmetric but always indefinite, and in the current work we solve it at once using LDL factorization. Representing the problem (14) as $\mathbf{S}\mathbf{x} = \mathbf{b}$ this implies:

$$\begin{aligned} &\mathbf{S} \rightarrow \mathbf{L}\mathbf{D}\mathbf{L}^{\mathrm{T}} \\ &\mathbf{x} = \mathbf{L}^{-\mathrm{T}}(\mathbf{D}^{-1}(\mathbf{L}^{-1}\mathbf{b})) \end{aligned} \tag{20}$$

Both methods (19) and (20) rely on direct matrix factorization. Strong advantage of direct factorization methods is the possibility to use them as black boxes for ill-conditioned systems which typically arise in geodynamic models. On the other hand, CPU and memory requirements of direct methods are high and thus application of such methods to significantly large systems of linear equations is difficult and iterative schemes are used instead [23,24]. However, as will be shown in this work, introduction of dynamic mesh adaptation reduces the number of unknowns and thus the cost of factorization dramatically, and makes it possible to use direct methods for models of quite high efficient numerical resolution even in case of expensive $\mathrm{Q}_2\mathrm{P}_{-1}$ mixed approximation.

Non-linearities arising in case of plastic yielding are treated using the method of successive substitutions (also known as Picard's method, see e.g., [25]). At first, effective viscosities are calculated with some initial velocity and pressure fields. The stiffness matrix **A** is built then, and depending on element type the Stokes problem is solved either by (19) or (20). Then effective viscosities are recalculated using the new solution and matrix **A** is rebuilt. These steps are repeated until the maximum number of non-linear iterations is passed or convergence criterion is reached:

$$\frac{\|\mathbf{v}_{\mathrm{nl}}^{\mathrm{i}} - \mathbf{v}_{\mathrm{nl}}^{\mathrm{i}-1}\|_2}{\|\mathbf{v}_{\mathrm{nl}}^{\mathrm{i}}\|_2} < \epsilon_{\mathrm{nl}} \tag{21}$$

where $\mathbf{v}_{\mathrm{nl}}^{\mathrm{i}}$ is the velocity solution obtained at non-linear iteration i and ϵ_{nl} is the non-linear tolerance.

4. Particle-In-Cell Simulation Methodology

The material properties and history are carried by Lagrangian material points, which are advected using the solution of the Stokes system obtained on a static grid (Figure 2). Such a combination of Lagrangian and Eulerian frames is known as the particle-in-cell method and is widely used in computational geodynamics (e.g., [26–28]). The method allows accurate tracking of material interfaces, and can be used for modeling of very large deformations without necessity to perform re-gridding, which is required in case of alternative Arbitrary Lagrangian-Eulerian approach (e.g., [10]).

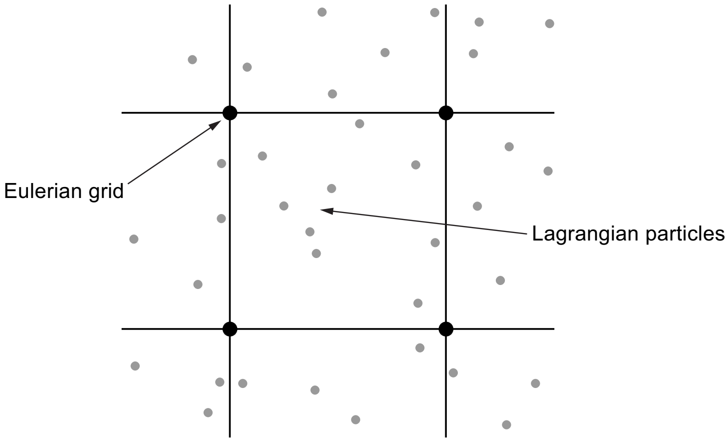

Figure 2. $\mathrm{Q}_1\mathrm{Q}_1$-based Eulerian grid combined with Lagrangian particles.

The PIC simulation algorithm can be described as a succession of the following actions performed at each time step:

- Interpolation of physical properties from Lagrangian particles to Eulerian grid.
- Assemblage of the Stokes system using interpolated physical properties.
- Solution of the system on an Eulerian grid (see Section 3).
- Interpolation of computed velocities to Lagrangian particles positions.
- Particles advection using the interpolated velocities.

Time integration is performed by the first-order forward Euler scheme. Thus each particle p is advected as:

$$\mathbf{x}_p(t+\Delta t) = \mathbf{x}_p(t) + \mathbf{v}_p(t)\Delta t \tag{22}$$

where $\mathbf{x}_p$ is the particle's coordinate vector, $\mathbf{v}_p$ is the interpolated velocity vector of a particle, and Δt is the time step calculated based on a given Courant number.

The accuracy of (22) is normally enough. However, if necessary, higher order integration methods can be used, for example the Runge-Kutta of fourth-order (e.g., [29]).

Initially, the Lagrangian particles are distributed randomly, but with uniform density. The initial uniform density, however, is not usually well preserved as simulation proceeds. The problem can be solved using the approach based on discrete Voronoi tessellation [30,31], which is described as follows. First, each element of computational Eulerian grid is discretized by a structured lattice. The lattice is then divided into subdomains (discrete Voronoi cells) in such a way that each sub-domain consists of all lattice cells closer to an associated particle than to any other particle residing in the element. The accuracy of such an approximate Voronoi diagram depends on a resolution of the lattice (Figure 3). Then, areas of discrete Voronoi cells are computed and compared to empirical thresholds AV_{min} and AV_{max}. Physically, the cell areas are considered as weights of associated Lagrangian particles, so particles whose weights are below AV_{min} are eliminated, and those with weights above AV_{max} are cloned. When cloning is performed, the new particle is added to the centroid of corresponding Voronoi cell (Figure 4). The centroid coordinate vector $\mathbf{x}_{\text{centroid}}$ is computed as:

$$\mathbf{x}_{\text{centroid}} = \frac{\sum_{i=1}^{n_{\text{cl}}} \mathbf{x}_{\text{cl}}^i}{n_{\text{cl}}} \tag{23}$$

where $\mathbf{x}_{\text{cl}}^i$ and n_{cl} are, respectively, the coordinate vectors and the total number of the lattice cells constituting the Voronoi cell.

Usually, some trial experiments are required to find appropriate values for AV_{min} and AV_{max}. Once these values are estimated, the algorithm works very efficiently to maintain uniform density of Lagrangian particle field.

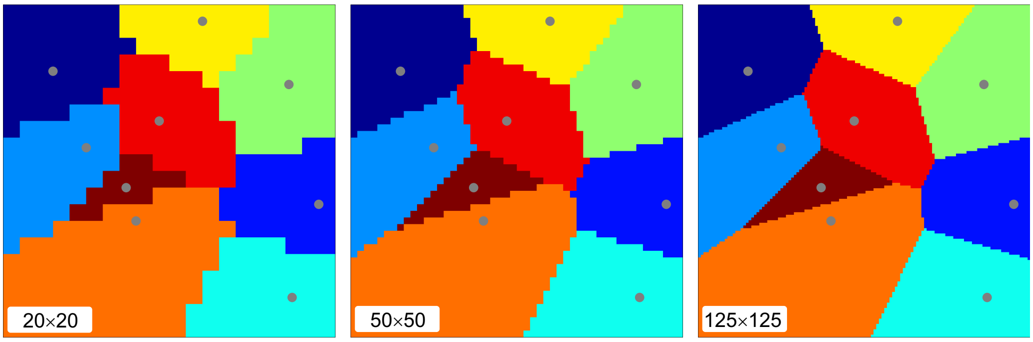

Figure 3. Approximate Voronoi diagrams. Note that accuracy depends on the lattice resolution.

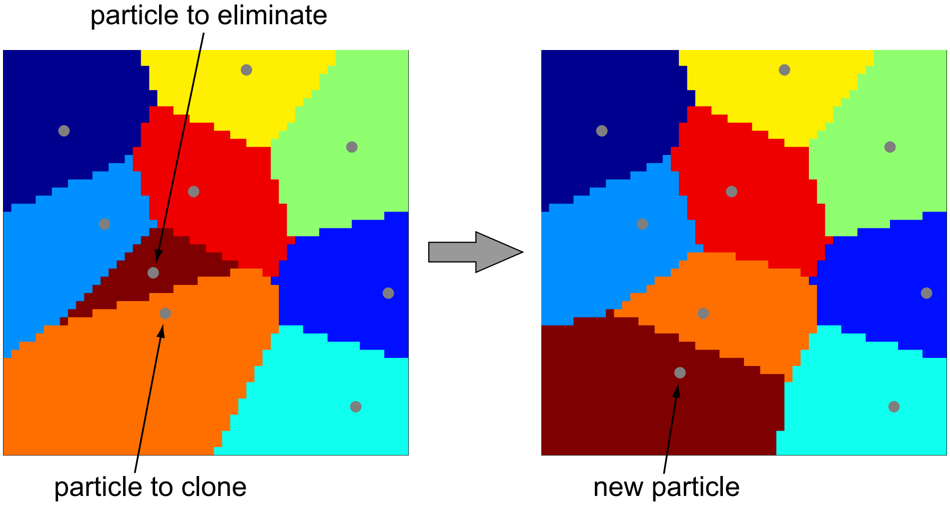

Figure 4. Elimination and cloning of Lagrangian particles.

5. Wavelet-Based Grid Adaptation

In this section we describe the implementation of the grid adaptation methodology based on wavelet analysis. First, the linear interpolating wavelet transform will be introduced. Next, the wavelet-based grid adaptation algorithm will be presented.

5.1. Linear Interpolating Wavelet Transform

In general, the wavelet transform is a very efficient methodology to construct the multilevel decomposition of some given function. Considering one dimensional case, a function $f(x)$ defined at $m2^{(n-1)}+1$ points (m is arbitrary integer number) can be decomposed to n levels from $J = n-1$ (finest) to 0 (coarsest) in such a way that at each level the function is approximated as f^j (e.g., [32]):

$$f^J(x) = \underbrace{\underbrace{f^0 + w^0}_{f^1} + \cdots + w^{j-1}}_{f^j} + \cdots + w^{J-1} \tag{24}$$

where f^0 is the function approximation at the coarsest level, and each w^j represents a difference between approximations at levels j and $j+1$:

$$\begin{aligned} f^0 &= \sum_{k\in\mathcal{K}^0} c_k^0 \phi_k^0(x) \\ w^j &= \sum_{l\in\mathcal{L}^j} d_l^j \psi_l^j(x) \end{aligned} \tag{25}$$

Further substitution of (25) into (24) yields:

$$f^J(x) = \sum_{k\in\mathcal{K}^0} c_k^0 \phi_k^0(x) + \sum_{j=0}^{J-1} \sum_{l\in\mathcal{L}^j} d_l^j \psi_l^j(x) \tag{26}$$

where $\mathcal{K}^0$ and $\mathcal{L}^j$ are index sets, $\phi_k^j(x)$ and $\psi_l^j(x)$ are scaling functions and wavelets respectively, and c_k^j and d_l^j are scaling function and wavelet coefficients respectively. This is the purpose of the forward wavelet transform to determine c_k^j and d_l^j.

Once the decomposition (26) is obtained, we can analyze wavelet coefficients d_l^j, which indicate differences between function approximations at neighbor levels, and leave only significant d_l^j with absolute values above some prescribed threshold ε. This way the compact representation $f_{\geqslant}^J(x)$ of the original function $f(x)$ is constructed:

$$f^J(x) = \underbrace{\sum_{k\in\mathcal{K}^0} c_k^0\phi_k^0(x) + \sum_{j=0}^{J-1}\sum_{\substack{l\in\mathcal{L}^j\\ |d_l^j|\geqslant\varepsilon}} d_l^j\psi_l^j(x)}_{\text{compact representation, } f_{\geqslant}^J(x)} + \underbrace{\sum_{j=0}^{J-1}\sum_{\substack{l\in\mathcal{L}^j\\ |d_l^j|<\varepsilon}} d_l^j\psi_l^j(x)}_{\text{insignificant, } f_{<}^J(x)} \tag{27}$$

Scaling functions $\phi_k^j(x)$ and wavelets $\psi_l^j(x)$ are spawned by dilations and translations of father wavelet $\phi(x)$ and mother wavelet $\psi(x)$ respectively:

$$\begin{aligned}\phi_k^j(x) &= \phi(2^jx-k)\\ \psi_k^j(x) &= \psi(2^jx-k)\end{aligned} \tag{28}$$

We use the linear interpolating transform in this work. This dictates the following expressions for $\phi(x)$ and $\psi(x)$ (see Figure 5 for the corresponding plots):

$$\begin{aligned}\phi(x) &= \max\{0, 1-|x|\}\\ \psi(x) &= 2\phi(2x-1) - \frac{1}{2}\phi(x) - \frac{1}{2}\phi(x-1)\end{aligned} \tag{29}$$

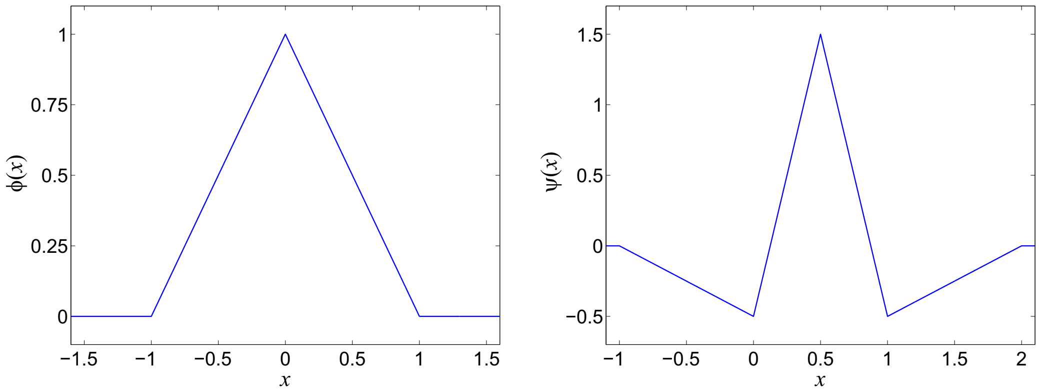

Figure 5. Father wavelet $\phi(x)$ and mother wavelet $\psi(x)$ for the linear interpolating transform.

Different techniques to perform forward wavelet transform exist. The very suitable one to work with interpolating wavelets is the lifting scheme [33], which is employed in this work. In case of linear interpolating $\phi(x)$ and $\psi(x)$, as in (29), the lifted transform from level $j+1$ to level j is (e.g., [34]):

$$\begin{aligned}d_k^j &= \frac{1}{2}(c_{2k+1}^{j+1} - \frac{1}{2}(c_{2k}^{j+1} + c_{2k+2}^{j+1}))\\ c_k^j &= c_{2k}^{j+1} + \frac{1}{2}(d_{k-1}^j + d_k^j)\end{aligned} \tag{30}$$

The scheme (30) is applied sequentially starting from the finest level and proceeding to the coarsest one. This produces values for c_k^j and d_l^j at all resolution levels which can be analyzed then in accordance with (27).

As an example consider the linear interpolating wavelet transform of the Gaussian function $f(x) = e^{-\frac{(x-0.5)^2}{(1/64)^2}}$ with further analysis for insignificant d_l^j coefficients shown on Figure 6. Scaling function coefficients c_k^0 represent the coarsest resolution level, and levels $1\ldots5$ are represented by wavelet coefficients $d_l^j, j = 0\ldots4$ respectively. Note how the

Gaussian curve is captured by significant d_l^j coefficients and how its approximation is sequentially refined at each level.

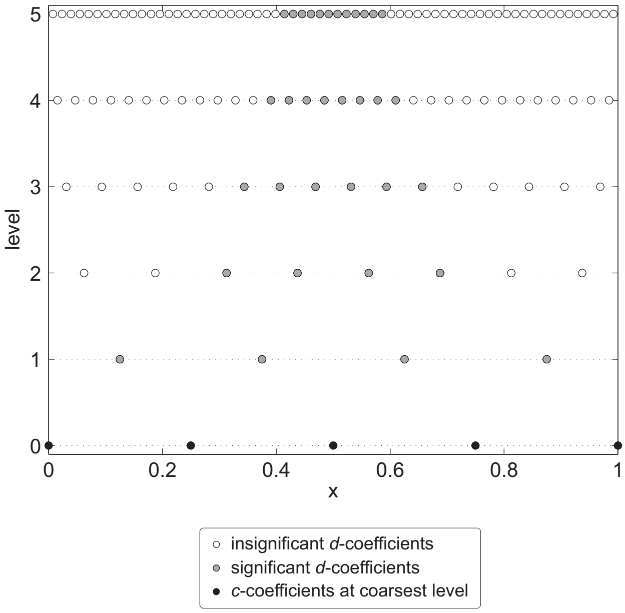

Figure 6. Wavelet coefficients for the Gaussian function $f(x) = e^{-\frac{(x-0.5)^2}{(1/64)^2}}$.

In two dimensional case, the 1D forward transform has to be performed in each dimension sequentially. Thus, given a function $f(x, y)$, the scheme (30) is applied first to x- and then to y-slices. The example of a two-dimensional function is considered in the next section.

5.2. Grid Adaptation Algorithm

The linear interpolating forward transform, described in the Section 5.1, can be applied to analyze a function representing some physical field of interest (viscosity η, density ρ, second strain rate invariant $\dot{\varepsilon}_{II}$, etc.). The result of such an analysis can be used then as a guidance to build a multilevel computational grid. To facilitate the process, the nodes of computational grid are associated with scaling function c_k^0 and wavelet coefficients d_l^j. Based on these ideas, the wavelet-based grid adaptation algorithm proceeds as follows (see publications [4,6] for further details):

- Perform the forward wavelet transform of a physical field which is considered as an adaptation criterion and get all c_k^0 and d_l^j coefficients. If a physical property field is defined on Lagrangian particles, the interpolation from particles to grid nodes is performed first.
- Analyze wavelet coefficients d_l^j at all levels and create a mask $\mathcal{M}$ containing grid nodes associated with significant d_l^j.
- Include into the mask $\mathcal{M}$ all grid nodes from the coarsest level, i.e. associated with coefficients c_k^0.
- Extend the mask $\mathcal{M}$ with grid nodes associated with adjacent to significant d_l^j. This is to ensure that the mask $\mathcal{M}$ includes all nodes whose coefficients can potentially become significant at the next simulation time step.
- Apply recursively the reconstruction check procedure to the mask $\mathcal{M}$. This is to guarantee that all wavelet coefficients d_l^j necessary to perform the forward transform at the next time step will be available.
- Using the adapted mask $\mathcal{M}$, construct a new multilevel finite element grid.

As an example, consider the linear interpolating wavelet transform of the 2-D Gaussian function $f(x,y) = e^{-\frac{x^2+y^2}{2(0.07)^2}}$. The mask $\mathcal{M}$ shown on Figure 7a was obtained by applying the two dimensional forward transform and including grid nodes associated with significant d_l^j and all c_k^0 into $\mathcal{M}$.

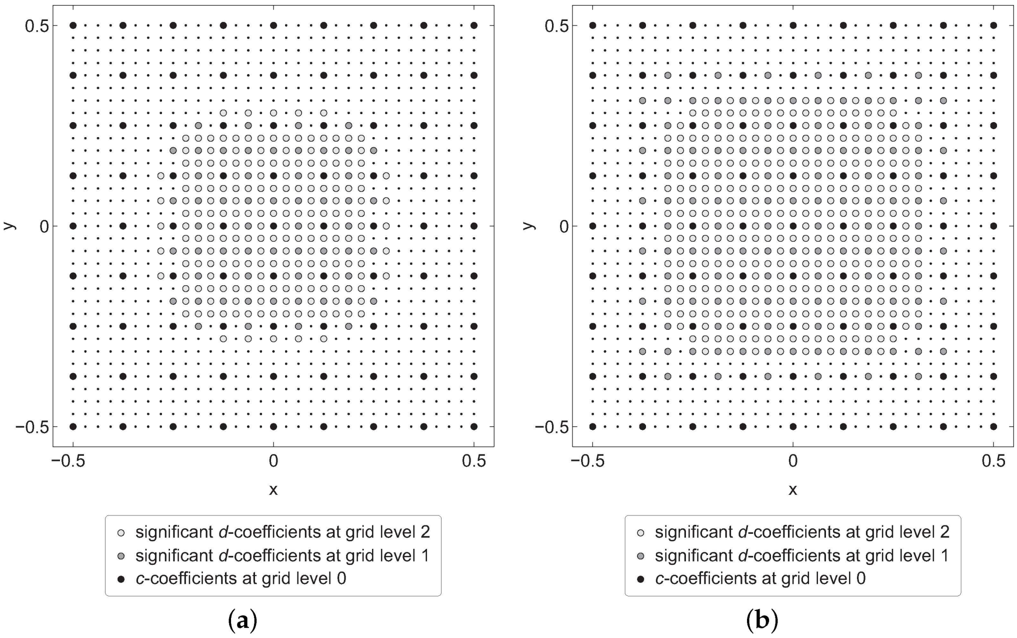

Figure 7. The linear interpolating wavelet transform of the 2-D Gaussian function $f(x,y) = e^{-\frac{x^2+y^2}{2(0.07)^2}}$: (**a**) the initial mask $\mathcal{M}$ after the wavelet transform and (**b**) the extended mask $\mathcal{M}$ used for mesh adaptation.

Further, extending $\mathcal{M}$ with grid nodes associated with adjacent wavelet coefficients and applying the reconstruction check procedure, the complete adapted mask $\mathcal{M}$ becomes as shown on Figure 7b. Based on the mask, the multilevel computational grid is built (Figure 8, bilinear discretization is assumed), which is then used to assemble and solve the Stokes system of equations.

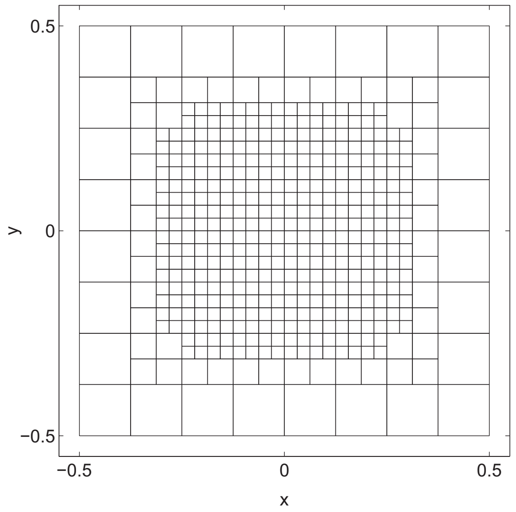

Figure 8. Multilevel bilinear finite element grid constructed using the adapted mask $\mathcal{M}$ on Figure 7b.

Normally, the new grid is constructed in this way at each simulation time step. Note that this does not impose any changes to the particle-in-cell simulation algorithm.

If necessary, several adaptation criteria can be used simultaneously. In such a case, each criterion is analyzed independently and the corresponding mask $\mathcal{M}_i$ is created. The nodewise OR combination of all $\mathcal{M}_i$ gives the common mask $\mathcal{M}$, which is then used for multilevel grid construction.

6. Dealing with Hanging Nodes

A multilevel grid, like one shown on Figure 8, is inherently non-conforming due to the presence of hanging nodes on boundaries between unequal grid levels. In order to obtain correct results on such a non-conforming grid, procedures of assemblage and solution of a corresponding system of equations have to be modified.

Bilinear (Q_1) and biquadratic (Q_2) basis functions, employed in this work, impose an approximated function to be continues across the element edges. Examining in this way the typical segment of Q_1 multilevel grid shown on Figure 9a, the approximation for an unknown function u has to be continues across the edge **ab**. This implies that for any point lying on the edge **ab** the following equality has to hold:

$$u_1 N_1(x,y) + u_2 N_2(x,y) + u_3 N_3(x,y) = u_1 N_4(x,y) + u_3 N_5(x,y) \tag{31}$$

where u_i are unknown nodal values of the function u, and N_i are bilinear basis functions.

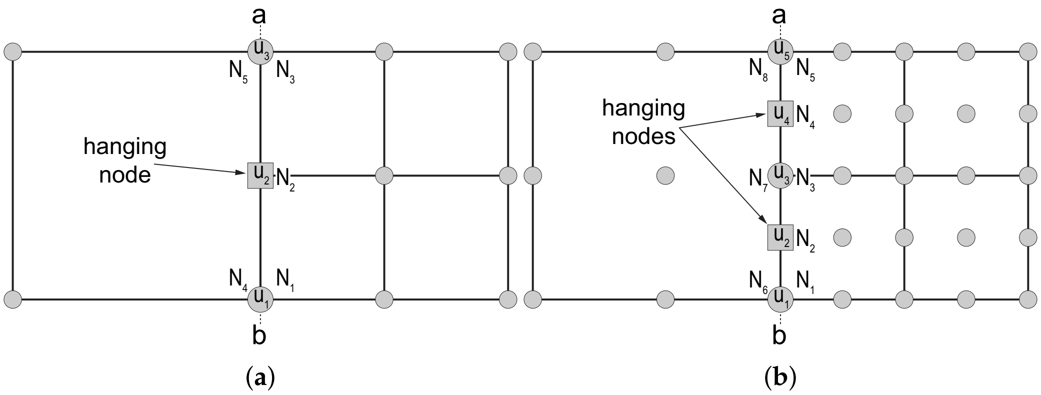

Figure 9. Typical segments of a multilevel grid for (**a**) bilinear (Q_1) and (**b**) biquadratic (Q_2) basis functions.

The degree of freedom corresponding to a hanging node has to be constrained to satisfy of the continuity requirement (31). Since bilinear approximation is considered, the constraint is expressed as:

$$u_2 = \frac{1}{2}(u_1 + u_3) \tag{32}$$

Generalizing, the relationship between unconstrained and constrained degrees of freedom defined on the edge **ab** can be expressed in the matrix form as:

$$\mathbf{u}^{\mathbf{ab}} = \mathbf{M}_{\mathbf{u}}^{\mathbf{ab}} \tilde{\mathbf{u}}^{\mathbf{ab}} \tag{33}$$

where $\mathbf{u}^{\mathbf{ab}}$ and $\tilde{\mathbf{u}}^{\mathbf{ab}}$ are unconstrained and constrained vectors of unknowns respectively, and $\mathbf{M}_{\mathbf{u}}^{\mathbf{ab}}$ is the transformation matrix:

$$\mathbf{u}^{\mathbf{ab}} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad \tilde{\mathbf{u}}^{\mathbf{ab}} = \begin{bmatrix} u_1 \\ u_3 \end{bmatrix} \quad \mathbf{M}_{\mathbf{u}}^{\mathbf{ab}} = \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{bmatrix} \tag{34}$$

In case of biquadratic basis functions, the similar to (31) continuity requirement across the edge **ab** for the typical segment of Q_2 multilevel grid shown on Figure 9b is formulated as:

$$\begin{aligned} & u_1 N_1(x,y) + u_2 N_2(x,y) + u_3 N_3(x,y) + u_4 N_4(x,y) + \\ & \qquad + u_5 N_5(x,y) = u_1 N_6(x,y) + u_3 N_7(x,y) + u_5 N_8(x,y) \end{aligned} \tag{35}$$

To satisfy the requirement (35), hanging degrees of freedom u_2 and u_4 have to be constrained by a second-order Lagrange polynomial constructed using values u_1, u_3 and u_5. Thus, the transformation between unconstrained and constrained vectors of unknowns can be expressed using the matrix relation (33), where $\mathbf{u^{ab}}$ and $\tilde{\mathbf{u}}^{\mathbf{ab}}$ vectors and matrix $\mathbf{M_u^{ab}}$ are:

$$\mathbf{u^{ab}} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} \quad \tilde{\mathbf{u}}^{\mathbf{ab}} = \begin{bmatrix} u_1 \\ u_3 \\ u_5 \end{bmatrix} \quad \mathbf{M_u^{ab}} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{3}{8} & \frac{6}{8} & -\frac{1}{8} \\ 0 & 1 & 0 \\ -\frac{1}{8} & \frac{6}{8} & \frac{3}{8} \\ 0 & 0 & 1 \end{bmatrix} \tag{36}$$

Apparently, the similar to (33) transformation between unconstrained and constrained degrees of freedom can be formulated for the whole system. Thereby, considering the Stokes problem on a multilevel grid, the velocity constraint relationship is expressed as:

$$\mathbf{v} = \mathbf{M_v} \tilde{\mathbf{v}} \tag{37}$$

where $\mathbf{v}$ and $\tilde{\mathbf{v}}$ are unconstrained and constrained velocity vectors respectively, and $\mathbf{M_v}$ is the transformation matrix. Note that $\mathbf{v}$ and $\tilde{\mathbf{v}}$ contain both x- and y-velocities and so $\mathbf{M_v}$ has to be constructed accordingly.

Thus, the Stokes system of equations, assembled on a multilevel grid, has to be solved subject to constraint defined by (37). Note that this is the case if $\mathrm{Q_1P_0}$ or $\mathrm{Q_2P_{-1}}$ elements are used, and element $\mathrm{Q_1Q_1}$ requires additional constraint on pressure due to the continuity of pressure approximation (see below). To impose constraint (37) on the Stokes system, the assembled stiffness matrix $\mathbf{A}$, gradient operator matrix $\mathbf{Q}$ and right hand side vector $\mathbf{F}$ are modified as [35,36]:

$$\begin{aligned} \tilde{\mathbf{A}} &= \mathbf{M_v^T A M_v} \\ \tilde{\mathbf{Q}} &= \mathbf{M_v^T Q} \\ \tilde{\mathbf{F}} &= \mathbf{M_v^T F} \end{aligned} \tag{38}$$

And then these modified matrices are used in the standard solution procedure. This way, the velocity Schur complement (17) is computed as:

$$\hat{\tilde{\mathbf{A}}} = \tilde{\mathbf{A}} + k \tilde{\mathbf{Q}} \mathbf{M}^{-1} \tilde{\mathbf{Q}}^{\mathrm{T}} \tag{39}$$

and augmented Lagrangian iterative method (19) is applied:

$$\begin{aligned} & \hat{\tilde{\mathbf{A}}} \rightarrow \tilde{\mathbf{L}} \tilde{\mathbf{L}}^{\mathrm{T}} \\ & \mathrm{DO} \\ & \quad \tilde{\mathbf{v}}^{\mathrm{i}} = \tilde{\mathbf{L}}^{-\mathrm{T}} (\tilde{\mathbf{L}}^{-1} (\tilde{\mathbf{F}} - \tilde{\mathbf{Q}} \mathbf{p}^{\mathrm{i}})) \\ & \quad \mathbf{v}^{\mathrm{i}} = \mathbf{M_v} \tilde{\mathbf{v}}^{\mathrm{i}} \\ & \quad \ldots \end{aligned} \tag{40}$$

Note, however, that unconstrained velocity vector $\mathbf{v}^{\mathrm{i}}$ is restored once the constrained solution $\tilde{\mathbf{v}}^{\mathrm{i}}$ is obtained.

In case if Q_1Q_1 element is employed, the additional constraint on pressure degrees of freedom is required:

$$\mathbf{p} = \mathbf{M_p}\tilde{\mathbf{p}} \tag{41}$$

where $\mathbf{p}$ and $\tilde{\mathbf{p}}$ are unconstrained and constrained pressure vectors respectively, and $\mathbf{M_p}$ is the transformation matrix defined in a similar way as $\mathbf{M_v}$.

To impose both velocity (37) and pressure constraint (41) on the Stokes system, the stiffness matrix $\mathbf{A}$ and right hand side vector $\mathbf{F}$ are modified in the same way as in (38), but gradient operator matrix $\mathbf{Q}$ and stabilization matrix $\mathbf{V}$ (which is required for Q_1Q_1 element) are modified as:

$$\begin{aligned} \tilde{\mathbf{Q}} &= \mathbf{M}_\mathbf{v}^\mathrm{T}\mathbf{Q}\mathbf{M_p} \\ \tilde{\mathbf{V}} &= \mathbf{M}_\mathbf{p}^\mathrm{T}\mathbf{V}\mathbf{M_p} \end{aligned} \tag{42}$$

The modified matrices are used to assemble the system which is then solved using LDL factorization (20). After that, the unconstrained velocity $\mathbf{v}$ and pressure $\mathbf{p}$ vectors are restored from the obtained $\tilde{\mathbf{v}}$ and $\tilde{\mathbf{p}}$ using (37) and (41) respectively.

7. Implementation Aspects

The numerical method described in this work is implemented in MATLAB programming language. To facilitate further development, the object-oriented programming capabilities introduced with MATLAB 2008a are extensively used. Input/output is performed using HDF5 data format, which makes possible to use external packages for postprocessing.

The Cholesky decomposition is performed using Cholmod solver [37], and for LDL factorization the MA57 solver is employed [38]. Both are distributed with MATLAB as external libraries, although up-to-date versions of both can be downloaded. The permutation to reduce fill-in is done using approximate minimum degree ordering method [39].

Where data structures permit, MATLAB Parallel Computing Toolbox is used for parallelization. This makes possible to run the code efficiently on modern multi-core architectures, as well as in distributed environments.

The code is available for download from http://sites.google.com/site/sdvigus/ (last accessed on 28 June 2022).

8. Numerical Benchmarks

In the second part of this work, the results of lateral viscosity variation, sinking block, brittle extension / compression and Rayleigh-Taylor instability benchmarks are presented and discussed. The list of the benchmark problems studied and the main aspects of the algorithm they test are summarized in Table 1. For each problem we compare performance and accuracy of adaptive and non-adaptive schemes, and evaluate applicability of bilinear (Q_1P_0, Q_1Q_1) and biquadratic (Q_2P_{-1}) mixed approximations in different conditions. All numerical experiments were performed using MATLAB R2009b.

Table 1. Benchmark problems studied.

Section	Benchmark Problem	Main Aspects of the Algorithm Tested by the Benchmark Problem
Section 8.1	Lateral viscosity variation	Comparison with the analytical solution
Section 8.2	Sinking block	Ability to handle large $O(10^7)$ viscosity contrasts
Section 8.3	Brittle extension/compression	Ability to capture and resolve spontaneously forming shear zones
Section 8.4	Incompressibility test	Influence of the artificial incompressibility
Section 8.5	Rayleigh-Taylor instability	Comparison with the analytical solution

8.1. Lateral Viscosity Variation Benchmark

This is the sharp viscosity variation benchmark which is often referred to as SolCx (e.g., [24,40]. The fully developed analytical solution exists, and thus the convergence of the proposed adaptive numerical method and correctness of the implementation can be evaluated.

8.1.1. Setup and Parameters

The model domain is defined as 1.0 × 1.0 box with free slip boundary conditions on all sides. The fluid flow is driven by density variation imposed by applying external force X as:

$$X = \rho(x,y)\begin{bmatrix} f_x \\ f_y \end{bmatrix}^{\mathrm{T}} \quad \text{where} \quad \begin{array}{l} \rho(x,y) = \sin(\pi y)\cos(\pi x) \\ f_x = 0,\ f_y = 1 \end{array} \tag{43}$$

The sharp lateral viscosity variation is defined as:

$$\eta(x,y) = \begin{cases} 1, & \text{if} \quad 0 \leqslant x \leqslant 0.5 \\ 10^6, & \text{if} \quad 0.5 < x \leqslant 1 \end{cases} \tag{44}$$

The plots showing initial density and viscosity profiles are provided on Figure 10.

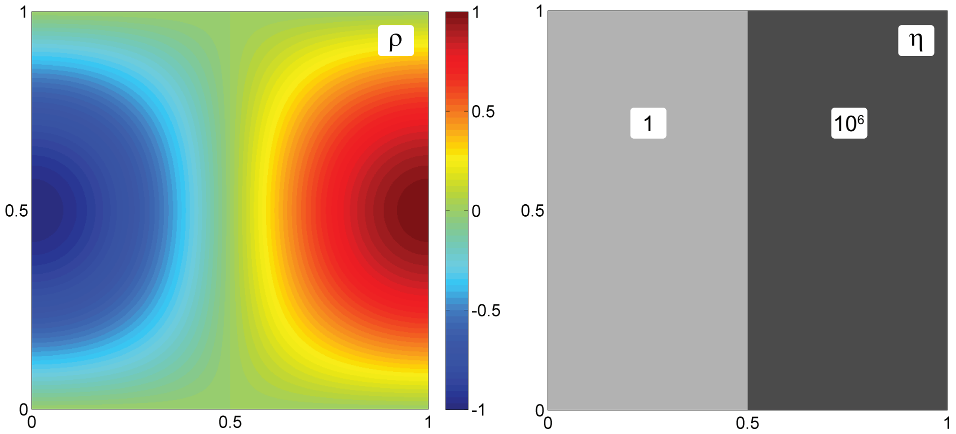

Figure 10. Initial density (**left**) and viscosity (**right**) profiles.

8.1.2. Convergence Test

The series of numerical experiments were performed on adaptive and non-adaptive Q_1Q_1-based grids of 128 × 128, 256 × 256, 384 × 384, 512 × 512 and 640 × 640 efficient numerical resolutions. In case of adaptive grid, the v_x- and v_y-velocity fields served as adaptation criteria with a threshold $\varepsilon = 10^{-5}$. Thus, the wavelet-based adaptation was performed independently for each of velocity fields, and the results were combined to form a common adapted mask.

The normalized error between numerical and analytical [41] solutions was computed as:

$$\|error\|_1 = \sum_{i=1}^{N}(|\bar{v}_{xi}^{\mathrm{an}} - \bar{v}_{xi}^{\mathrm{num}}| + |\bar{v}_{yi}^{\mathrm{an}} - \bar{v}_{yi}^{\mathrm{num}}|)V_i \tag{45}$$

where N is the total number of elements, $\bar{v}_{xi}^{\mathrm{an}}$, $\bar{v}_{xi}^{\mathrm{num}}$, $\bar{v}_{yi}^{\mathrm{an}}$ and $\bar{v}_{yi}^{\mathrm{num}}$ are arithmetic means of analytical and numerical v_x- and v_y-velocities of i-element, V_i is i-element volume.

Dependence of the normalized error on the number of elements for both adaptive and non-adaptive cases are shown in Figure 11. As it is clearly seen from the plots, both adaptive and non-adaptive methods converge with the same rate, but the convergence curve for the adaptive method is lower. Thus, the adaptive approach has a clear advantage—the higher accuracy for the same number of elements.

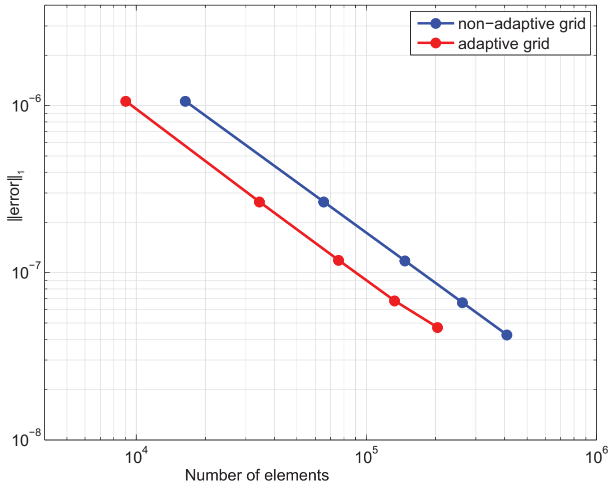

Figure 11. Dependence of the normalized error $\|error\|_1$ on the number of elements.

8.2. Sinking Block Benchmark

To evaluate the general applicability of the proposed adaptive numerical method, we performed the classical numerical benchmark of a sinking of a hard rectangular block into a medium with a lower viscosity [26]. Although corresponding analytical solution is not known, the numerical results can be evaluated in a qualitative sense.

8.2.1. Setup and Parameters

The model setup is shown on Figure 12, and employed physical and numerical parameters are provided in Table 2. The numerical simulations were performed on Q_1P_0 and Q_1Q_1 adaptive grids of $6 \times 12 \rightarrow 96 \times 192$ (5 levels) resolution, and with about 1.7×10^5 Lagrangian particles. The viscosity field was used as an adaptation criterion with an adaptation threshold $\varepsilon = 10^{-3}$.

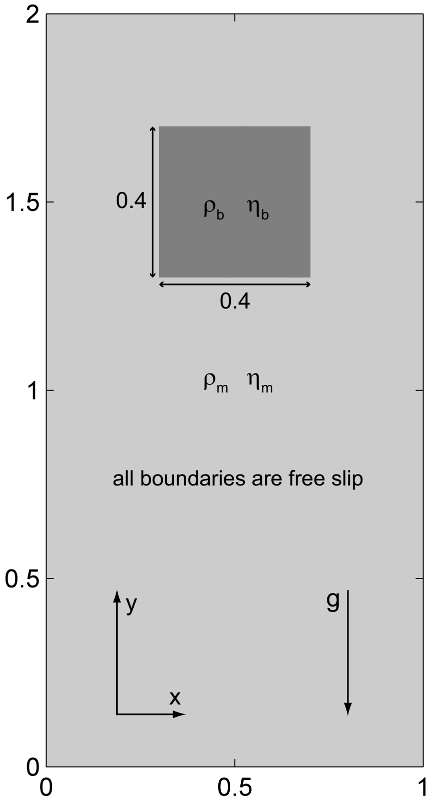

Figure 12. Sinking block model setup.

Table 2. Model parameters for the sinking block benchmark.

Parameter	Value
Block viscosity η_b	10^2–10^7
Medium viscosity η_m	1.0
Block density ρ_b	2.0
Medium density ρ_m	1.0
Gravitational acceleration g	−10.0
Time step Δt	0.1

Since the viscosity interface between the block and the medium exists a priori, the preliminary grid adaptation was performed before the actual numerical simulations were started (Figure 13).

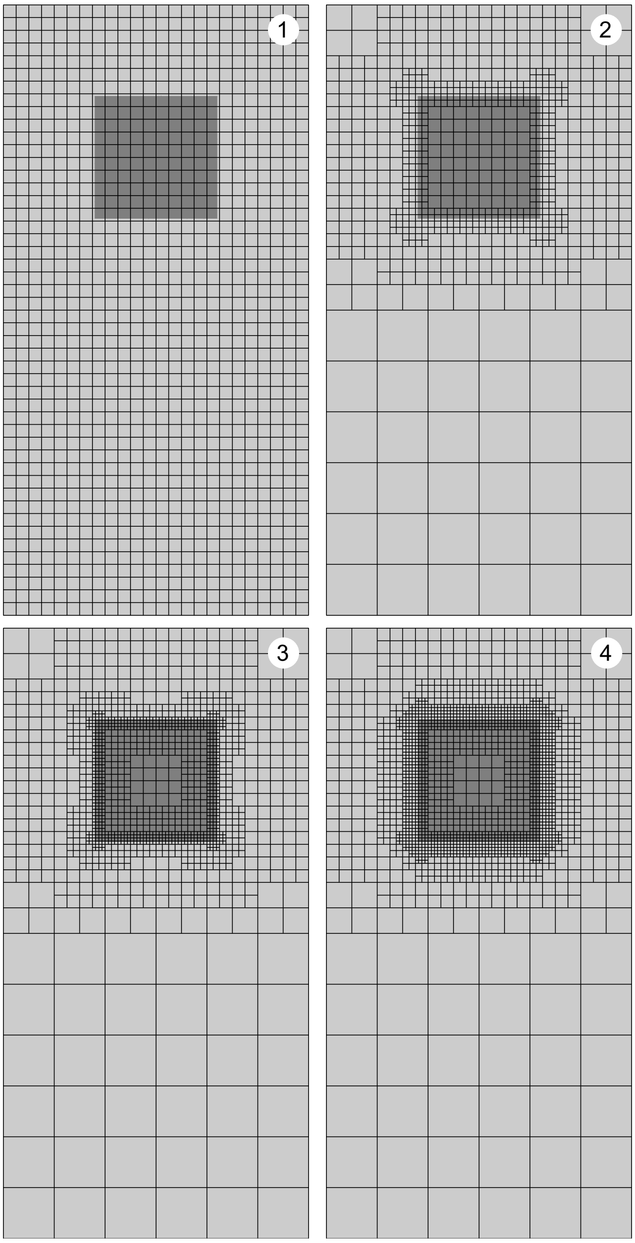

Figure 13. Preliminary grid adaptation.

8.2.2. Effect of Viscosity Contrast

The results of numerical simulations on Q_1Q_1 adaptive grid with 10^2 and 10^7 viscosity contrasts are shown on Figures 14 and 15 respectively. As it is seen from the figures, the interface between the block and the medium is accurately captured by a multilevel grid as simulation proceeds and the block sinks down. As expected from the physical point of view, the block undergoes deformations in case of moderate viscosity contrast (10^2), but these deformations vanish as viscosity contrast becomes high (10^7). The fact that the shape of the block is preserved at high viscosity contrast also verifies that the implemented numerical scheme is stress conservative.

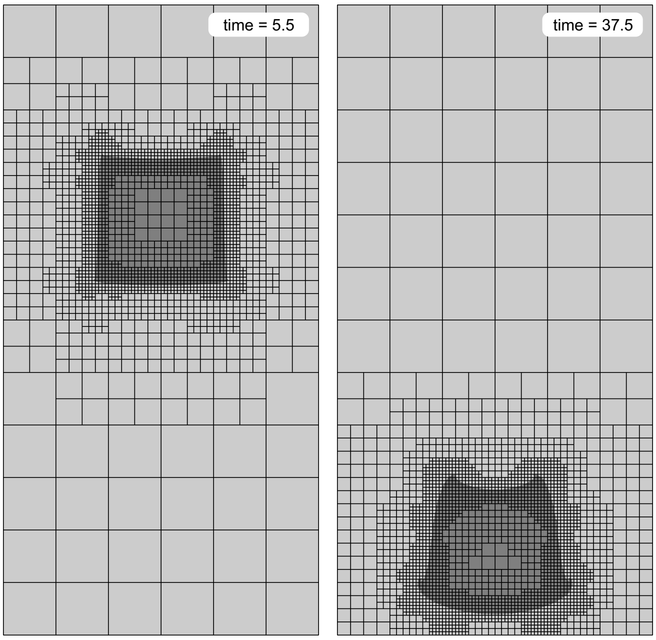

Figure 14. Results of sinking block benchmark with viscosity contrast $\frac{\eta_b}{\eta_m} = 10^2$. Material field with imposed numerical grid is shown.

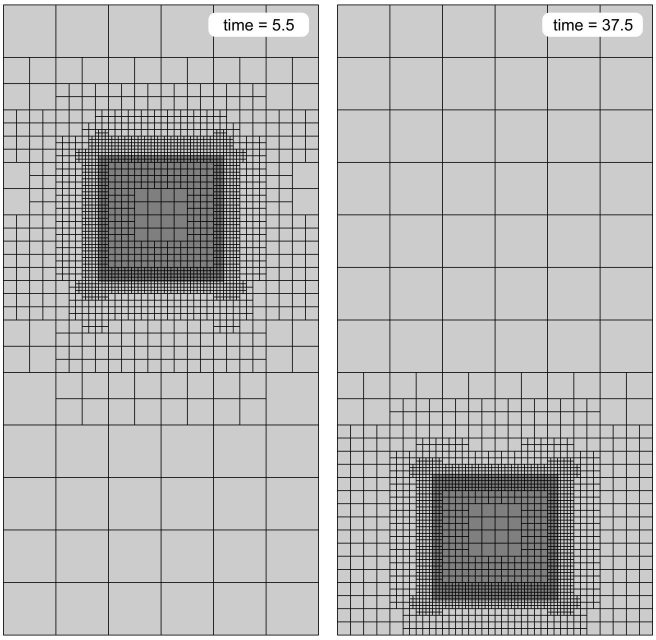

Figure 15. Results of sinking block benchmark with viscosity contrast $\frac{\eta_b}{\eta_m} = 10^7$. Material field with imposed numerical grid is shown.

8.2.3. Checkerboard Pressure Problem with Q_1P_0 Element

The benchmark was performed using Q_1P_0 mixed approximation to evaluate its applicability and make a comparison with Q_1Q_1 element.

While the general model evolution is similar for Q_1Q_1 and Q_1P_0 elements, the pressure solution obtained with Q_1P_0 clearly shows strong checkerboard effect (Figure 16). This problem is well known and is caused by the fact that Q_1P_0 is inherently unstable [16,17]. However, in our experience, the checkerboard effect is even more evident when multilevel grid is employed. Thus, Q_1P_0 element has to be used with care, especially when non-linear pressure-dependent rheology is used (see Section 8.3).

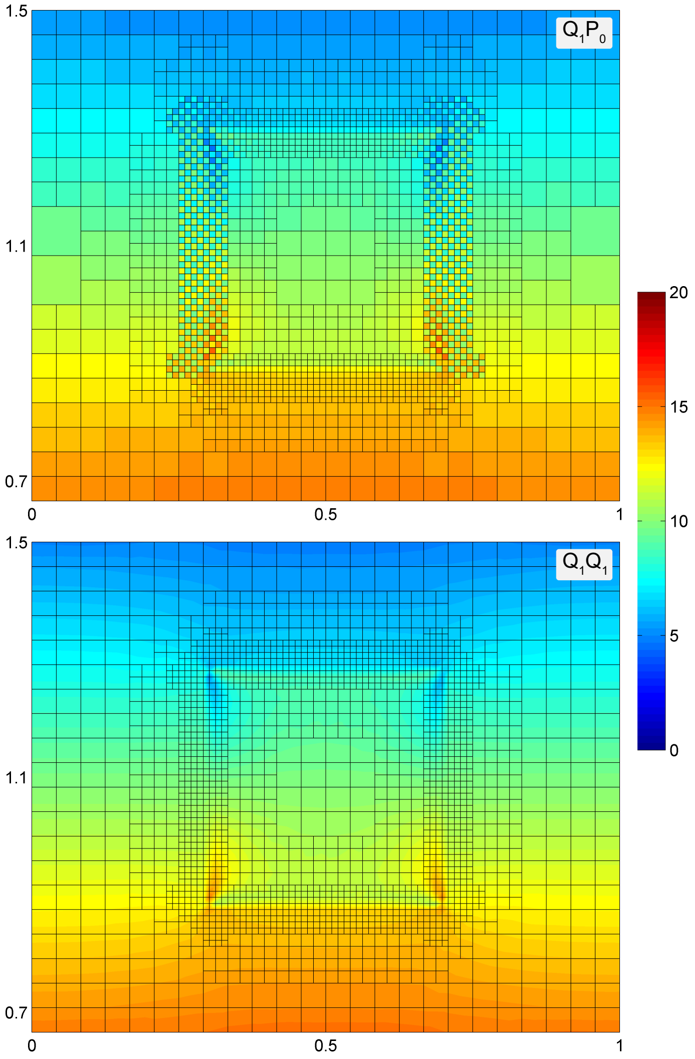

Figure 16. Pressure field around the block obtained with Q_1Q_1 and Q_1P_0 elements.

8.2.4. Performance Analysis

The comparisons of average performance between Q_1Q_1 adaptive and non-adaptive schemes were done at resolutions $6 \times 12 \rightarrow 384 \times 768$ (7 levels) in adaptive and 384×768 in non-adaptive case, and with about 2.7×10^6 Lagrangian particles in both cases. The results obtained in sequential (1 CPU) and parallel (8 CPUs) modes are shown on Figure 17.

As it is seen from the plots, there is a very significant performance gain when adaptation is employed, especially for the solution of the Stokes system. Due to the high cost of Voronoi tessellation, the reordering of Lagrangian particles is relatively expensive in both adaptive and non-adaptive cases. The potential solution to this problem is to implement the Voronoi tessellation functionality in an external library using a low-level language, such as `C++`.

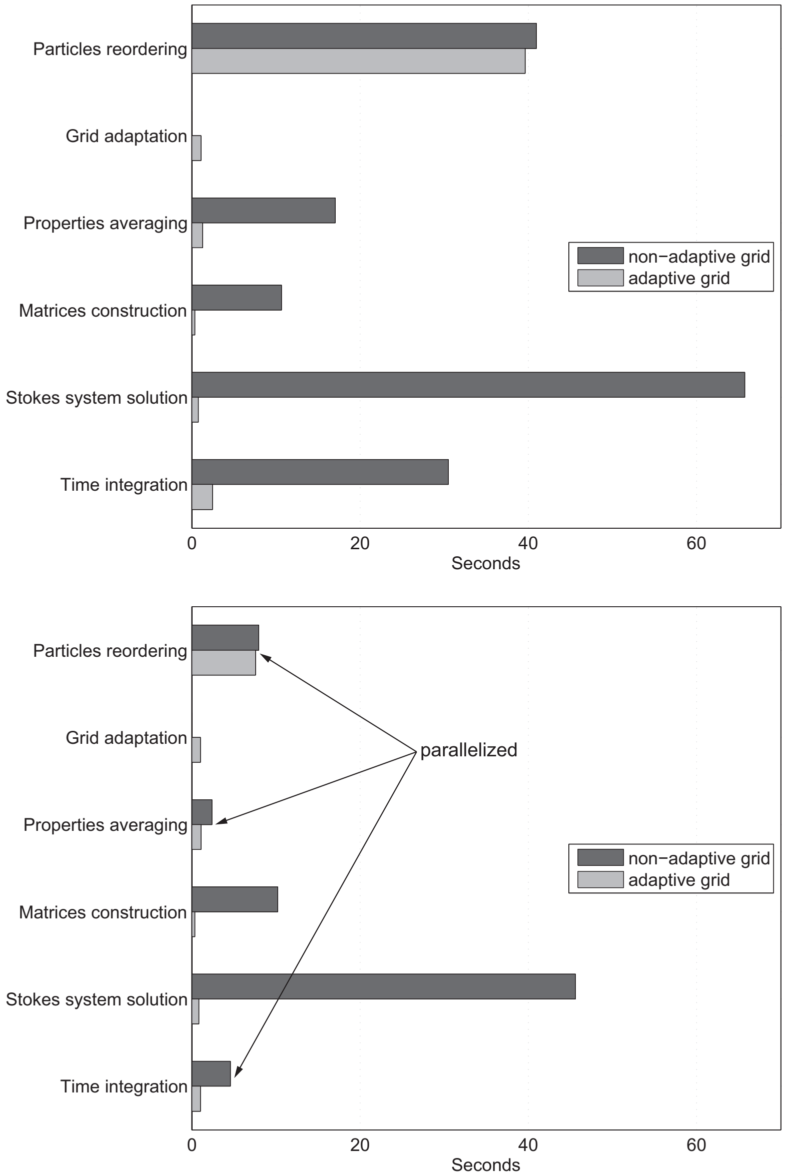

Figure 17. Performance comparisons between adaptive and non-adaptive numerical schemes for sinking block model; **top**: 1 CPU; **bottom**: 8 CPUs with MATLAB Parallel Computing Toolbox. Dual AMD Opteron 8380 system was used.

The parallel speedup from 1 to 8 CPUs is significant but less than linear. This can be explained by the fact that MATLAB Parallel Computing Toolbox is a relatively new product, and an efficient parallelization using an interpreted dynamic language, such as MATLAB, poses significant challenges.

8.3. Brittle Extension/Compression Benchmark

This benchmark was performed to test the capabilities of the proposed adaptive approach to capture and resolve spontaneously forming shear zones in non-linear brittle faulting models.

8.3.1. Setup and Parameters

The extension/compression model setup is shown on Figure 18, and model parameters are given in Table 3 (physical parameters are from [14]). The numerical experiments were performed on Q_1Q_1-based $12 \times 4 \rightarrow 768 \times 256$ (7 levels) adaptive and on 768×256 non-adaptive grids, and with about 1.8×10^6 Lagrangian particles in both cases. Adaptive simulations were started on 384 × 128 equidistant grid. The v_x- and v_y-velocity fields served as adaptation criteria with a threshold $\varepsilon = 10^{-3}$. Thus, the wavelet-based adaptation was performed independently for each of velocity fields, and the results were combined to form a common adapted mask. The multilevel grid, constructed from such a mask, efficiently captures the gradients of the strain rate field.

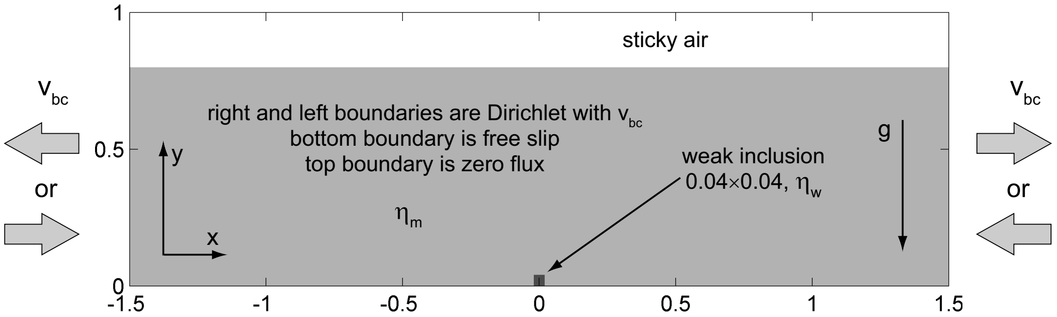
1
sticky air
V_bc
V_bc
right and left boundaries are Dirichlet with v_bc
bottom boundary is free slip
top boundary is zero flux
0.5
y
or
or
weak inclusion
0.04×0.04, η_w
g
η_m
x
0
-1.5
-1
-0.5
0
0.5
1
1.5

Figure 18. Extension/compression model setup.

Table 3. Model parameters for brittle extension/compression benchmark.

Parameter		Value
	Weak inclusion viscosity η_{w}	1.0
	Medium viscosity η_{m}	100.0
	Weak inclusion and medium density ρ	1.0
	Air viscosity η_{a}	0.01
	Air density ρ_{a}	0.0
	Gravitational acceleration g	−10.0
	Friction angle ϕ	0°–30°
	Strain values $\varepsilon_1/\varepsilon_2$	0.0 / 0.1
Cohesion c_1/c_2	Extension	4.0 / 1.0
	Compression	20.0 / 10.0
Boundary velocity v_{bc}	Extension	0.05
	Compression	−0.5
Time step Δt	Extension	0.02
	Compression	0.004
	Nonlinear tolerance ϵ_{nl}	10^{-3}

The strain softening/hardening behavior of brittle materials is modeled by adjusting the cohesion c at each time step as:

$$c = c_1 + (c_2 - c_1)\min\left\{1, \max\left\{0, \frac{\varepsilon_{\mathrm{pl}} - \varepsilon_1}{\varepsilon_2 - \varepsilon_1}\right\}\right\} \tag{46}$$

Thus, the cohesion value c is changed linearly from initial c_1 to final c_2 as accumulated plastic strain ε_{pl} changes from ε_1 to ε_2. The softening behavior is imposed when $c_2 < c_1$, and the hardening—when $c_2 > c_1$. The accumulated plastic strain $\varepsilon_{\mathrm{pl}}$ is defined on Lagrangian particles, and is integrated by the forward Euler scheme as other particles properties:

$$\varepsilon_{\mathrm{pl}}(t + \Delta t) = \varepsilon_{\mathrm{pl}}(t) + \dot{\varepsilon}_{\mathrm{pl}}(t)\Delta t \tag{47}$$

Here $\dot{\varepsilon}_{\mathrm{pl}}$ is the plastic strain rate computed as:

$$\dot{\varepsilon}_{\mathrm{pl}} = \dot{\varepsilon}_{II}\left(1 - \frac{\eta_{\mathrm{eff}}}{\eta_{\mathrm{creep}}}\right) \tag{48}$$

where $\dot{\varepsilon}_{II}$ is the second invariant of the strain rate tensor, and η_{creep} and η_{eff} and creep and viscoplastic effective viscosities respectively.

8.3.2. Shear Bands Formation

The brittle extension and compression numerical experiments were performed with friction angles $\phi = 0°$, $10°$, $20°$ and $30°$ on both adaptive and non-adaptive grids. As an example, the results of an extension numerical simulation with $\phi = 0°$ are shown on Figures 19 and 20, and the results of a compression simulation with $\phi = 30°$ are shown on Figures 21 and 22. As it is seen from the figures, the multilevel grid is progressively adapted to capture and better resolve appearing brittle faults as simulation proceeds.

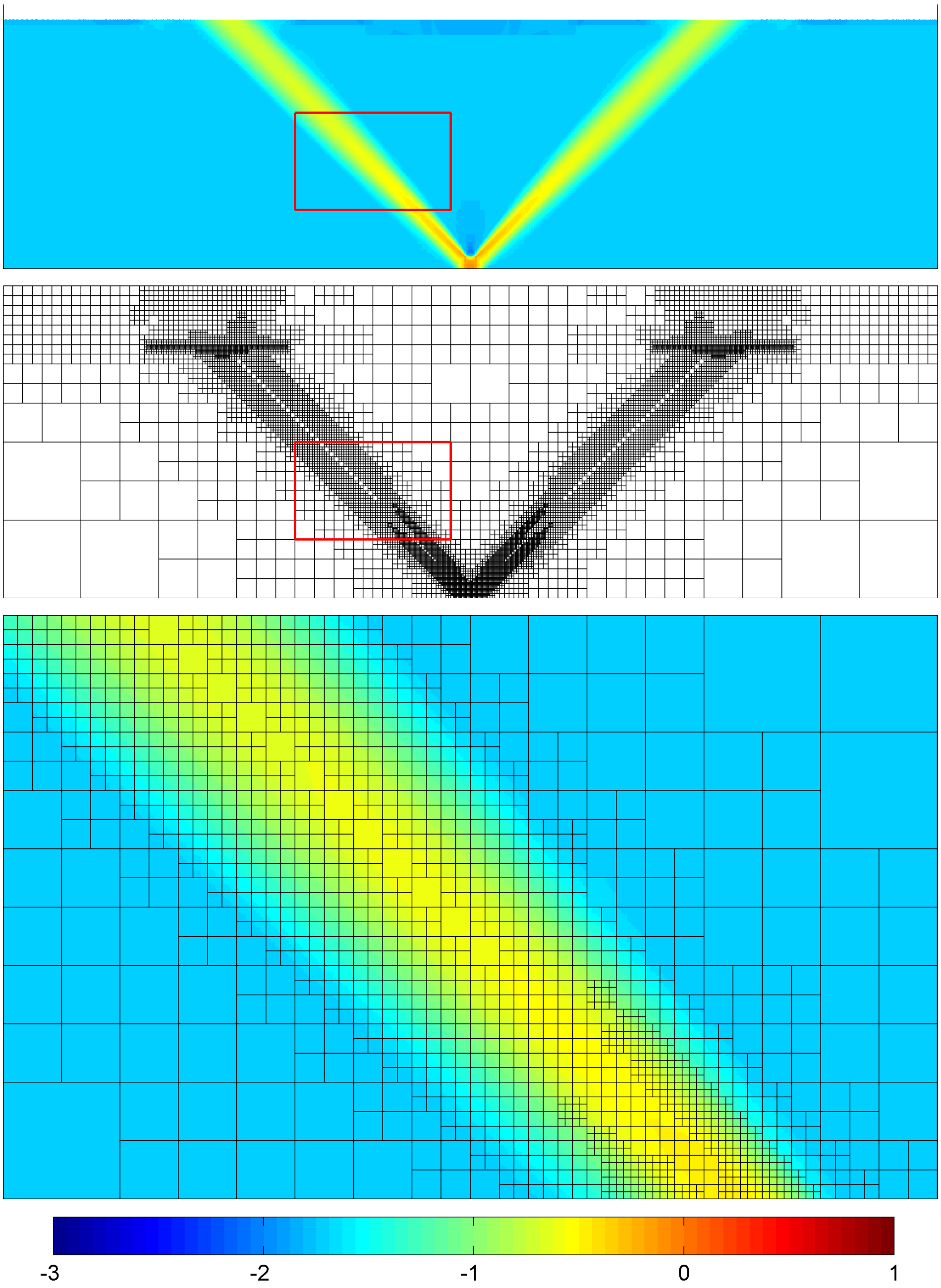

Figure 19. Extension model with $\phi = 0°$ after 0.07% strain; **top**: $\log(\dot{\varepsilon}_{II})$ plot; **middle**: numerical grid; **bottom**: zoom as marked by red rectangles.

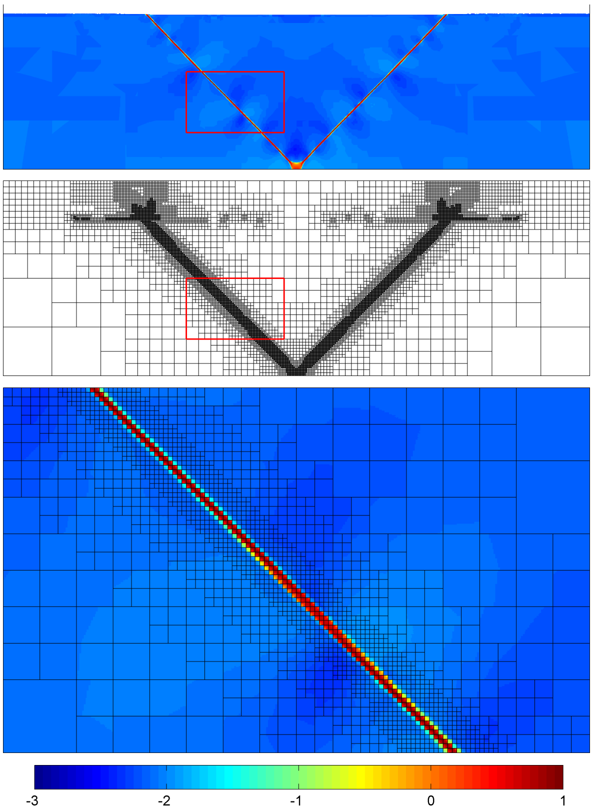

Figure 20. Extension model with $\phi = 0°$ after 0.53% strain; **top**: $\log(\dot{\varepsilon}_{II})$ plot; **middle**: numerical grid; **bottom**: zoom as marked by red rectangles.

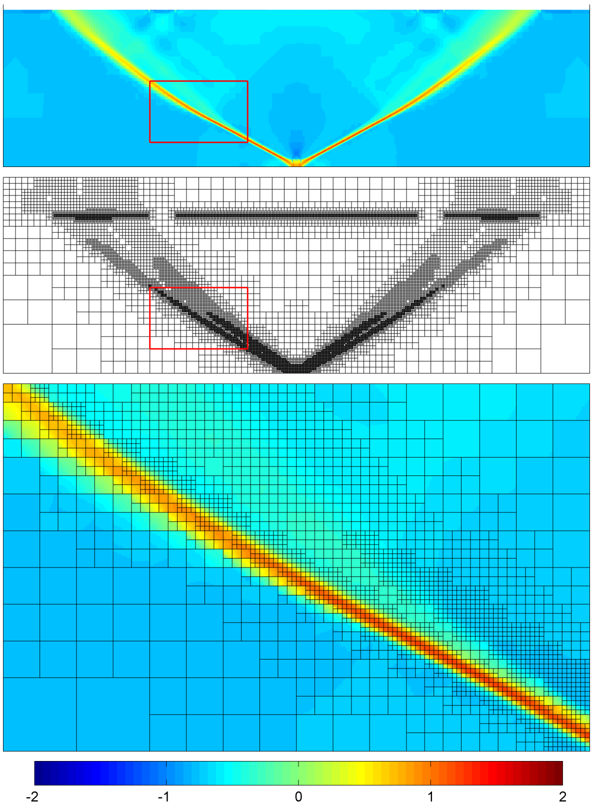

Figure 21. Compression model with $\phi = 0°$ after 0.13% strain; **top**: $\log(\dot{\varepsilon}_{II})$ plot; **middle**: numerical grid; **bottom**: zoom as marked by red rectangles.

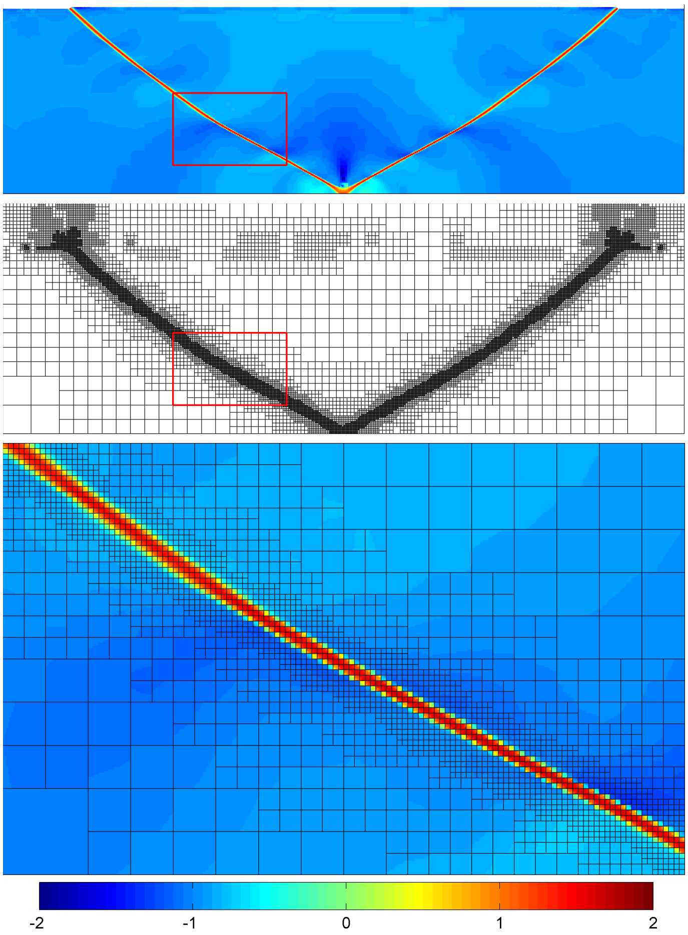

Figure 22. Compression model with $\phi = 0°$ after 1.33% strain; **top**: $\log(\dot{\varepsilon}_{II})$ plot; **middle**: numerical grid; **bottom**: zoom as marked by red rectangles.

The orientations of the formed shear bands with respect to horizontal were measured for both adaptive and non-adaptive models and compared against each other and against Coulomb orientations (Figure 23). The results obtained on adaptive and non-adaptive grids are very close, and show good agreement with Coulomb angles which are expected [14,42].

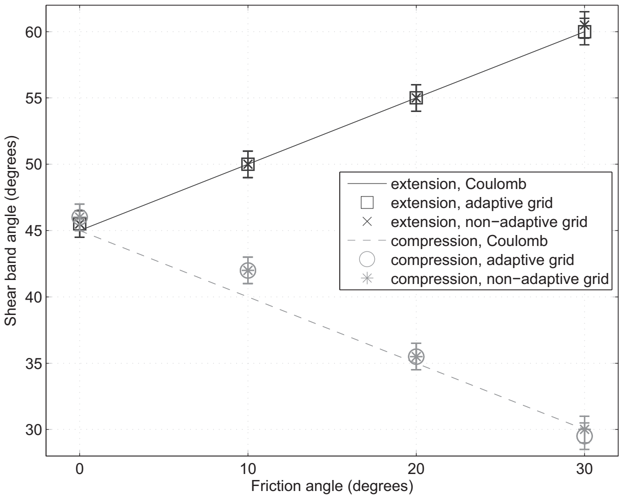

Figure 23. Orientations of shear bands in extension/compression simulations with different frictions angles.

8.3.3. Long-Term Brittle Extension

It is interesting to compare long-term brittle model evolution simulated on adaptive and non-adaptive grids. The material and strain rate fields from extension simulations with $\phi = 0^\circ$ after relatively high strains are shown on Figures 24–26. As it is seen from the figures, the results obtained on adaptive and non-adaptive grids are almost identical. Thus, for considered models, the adaptive scheme can be used instead of non-adaptive with confidence.

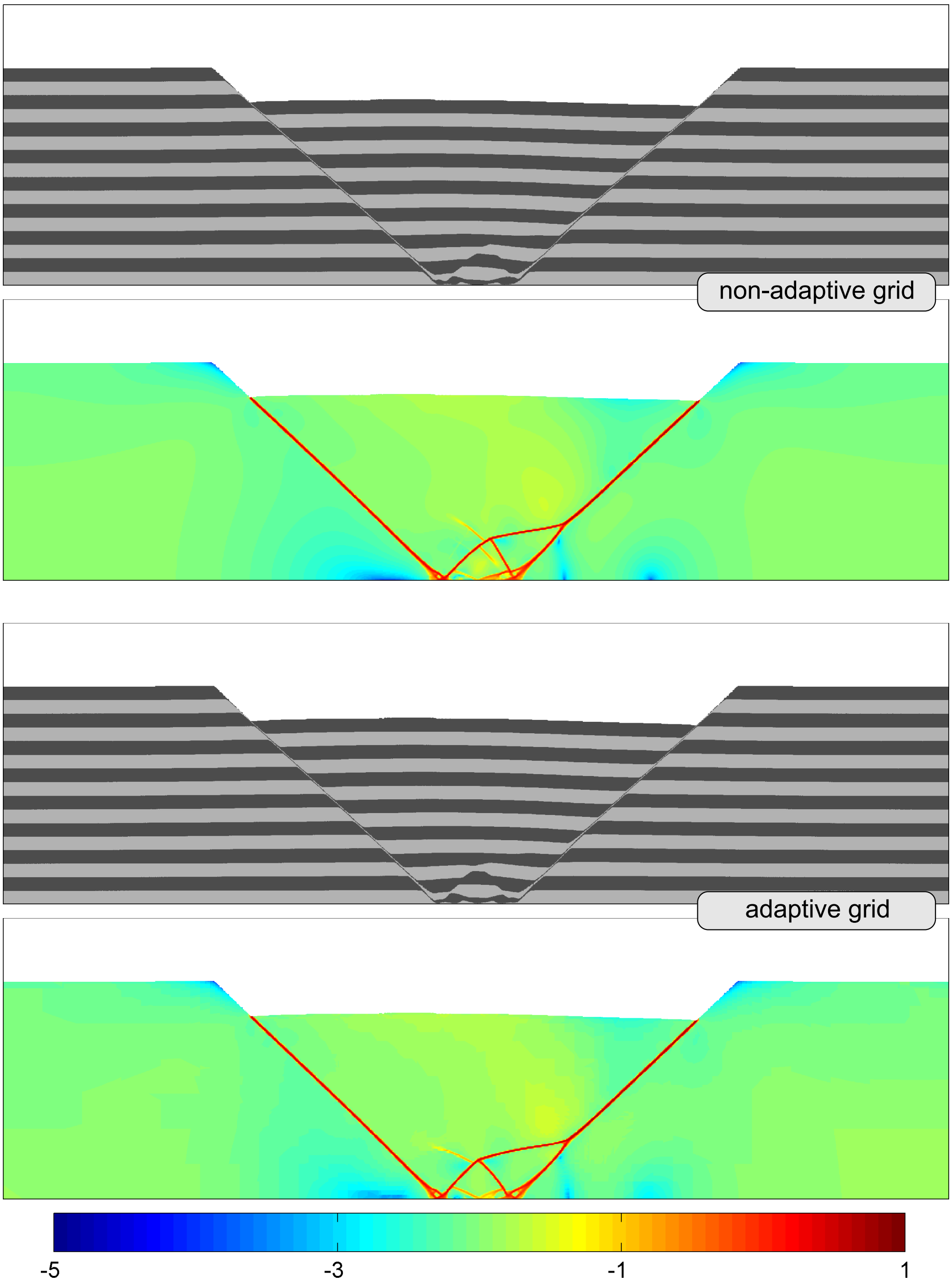

Figure 24. Material fields and $\log(\dot{\varepsilon}_{II})$ plots obtained with non-adaptive and adaptive grids for extension model with $\phi = 0^\circ$ after 12.33% strain.

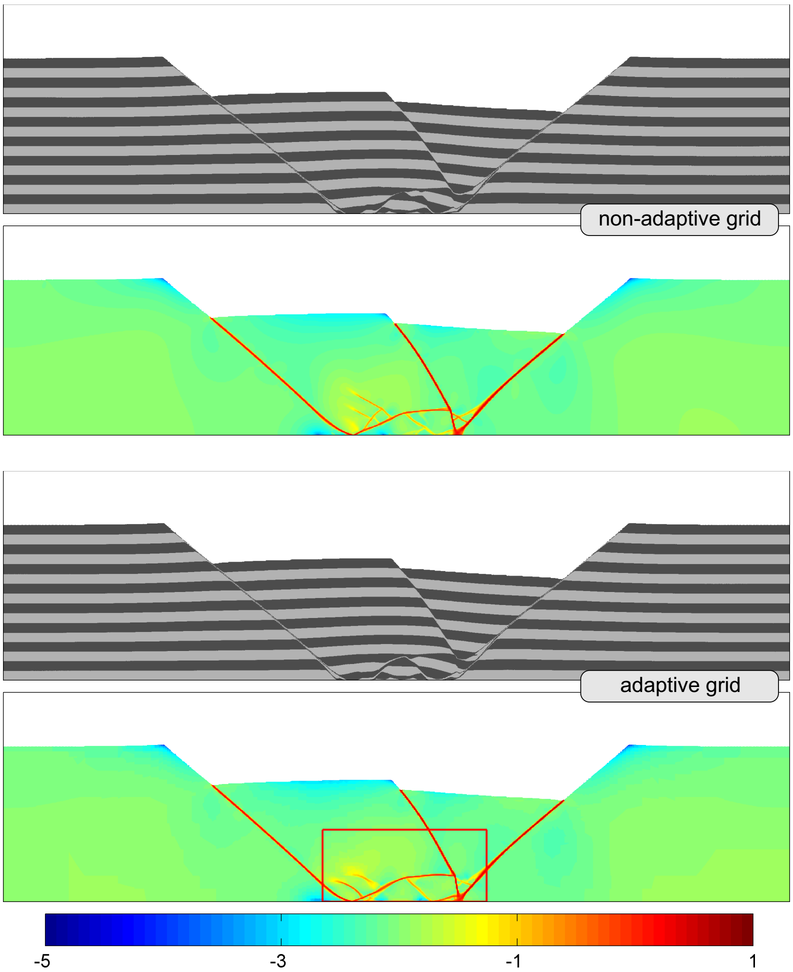

Figure 25. Material fields and $\log(\dot{\varepsilon}_{II})$ plots obtained with non-adaptive and adaptive grids for extension model with $\phi = 0°$ after 25.33% strain. See Figure 26 for zoom as marked by a red rectangle here.

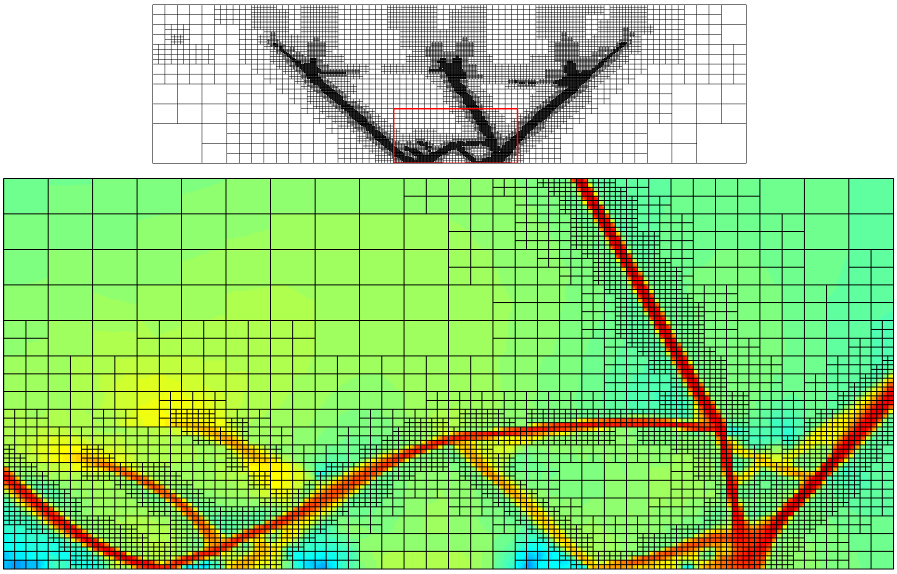

Figure 26. Results of adaptive extension simulation with $\phi = 0°$ after 25.33% strain. Top: numerical grid; bottom: zoom as marked by a red rectangle here and on Figure 25.

8.3.4. Performance Analysis

The comparison of average performance between adaptive and non-adaptive schemes is shown on Figure 27. As in the sinking block benchmark (Section 8.2), the performance gain for matrices construction and Stokes system solution is very significant when adaptive grid is employed. However, the considered problem is non-linear and the Stokes system has to be constructed and solved normally more than once at each time step. Thus, the overall performance advantage of the adaptive approach is even more dramatic in this case.

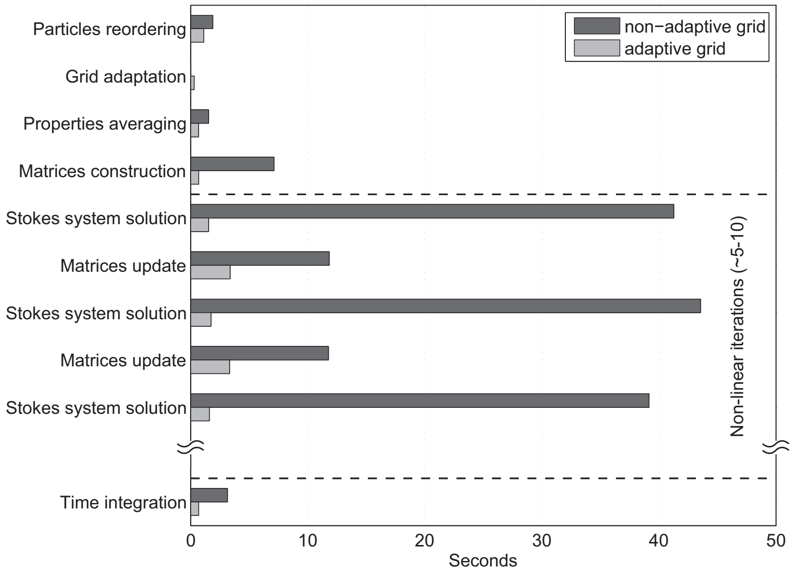

Figure 27. Performance comparison between adaptive and non-adaptive numerical schemes for brittle extension model. Dual AMD Opteron 8380 system was used.

8.4. Incompressibility Issue with Q_1Q_1 Element

As was demonstrated by previous numerical experiments (Sections 8.2 and 8.3), the results obtained with stabilized Q_1Q_1 element are adequate in many cases. However, in many gravity-driven geodynamic models, the Q_1Q_1 element behaves unreliably due to the issue of artificial incompressibility caused by the addition of the stabilization matrix **V** to the Stokes system. This problem is discussed in the original works [18,19] and was also confirmed in personal communications with P. Bochev.

To evaluate the issue, we performed the series of numerical simulations on Q_1Q_1- and Q_2P_{-1}-based regular grids of a model containing square 1.0 × 1.0 domain with uniform $\rho = 1.0$ and $\eta = 1.0$ and applied gravity $g = -10.0$. Since the media is assumed to be incompressible, one would expect v_y-velocity to be zero everywhere. However, the differences between normalized L^2-norms of v_y obtained with Q_1Q_1 and Q_2P_{-1} elements are almost 10 orders of magnitude (Table 4).

Table 4. Velocity norms $\|v_y\|_2$ normalized by the number of elements.

Element Type	**Resolution**			
	100 × 100	**200 × 200**	**400 × 400**	**800 × 800**
Q_1Q_1	8.35×10^{-7}	1.04×10^{-7}	1.30×10^{-8}	1.63×10^{-9}
Q_2P_{-1}	2.26×10^{-17}	2.31×10^{-17}	2.39×10^{-17}	2.89×10^{-17}

Thus, while the Q_1Q_1 element is stable and can be efficiently used in numerical experiments like brittle faulting (Section 8.3), it has to be avoided in models which are gravity-driven by a small density differences—such as Rayleigh-Taylor instability or subduction simulations.

8.5. Rayleigh-Taylor Instability Benchmark

This is the classical two-layer Rayleigh-Taylor instability benchmark. The corresponding analytical solution is available, and thus the accuracy of the employed computational scheme can be evaluated.

8.5.1. Setup and Parameters

The Rayleigh-Taylor model setup is shown on Figure 28, and model parameters are provided in Table 5. The Q_1Q_1 element can not be used for this benchmark due to incompressibility issue (see above), and thus biquadratic mixed approximation Q_2P_{-1} was used. The numerical simulations were performed on $25 \times 25 \rightarrow 400 \times 400$ (5 levels) adaptive and on 400×400 non-adaptive grids, and with 1.6×10^7 Lagrangian particles in both cases. The material density field was used as an adaptation criterion with a threshold $\varepsilon = 10^{-5}$. Since the density interface between the layers exists a priori, the preliminary grid adaptation was performed—as in the sinking block benchmark.

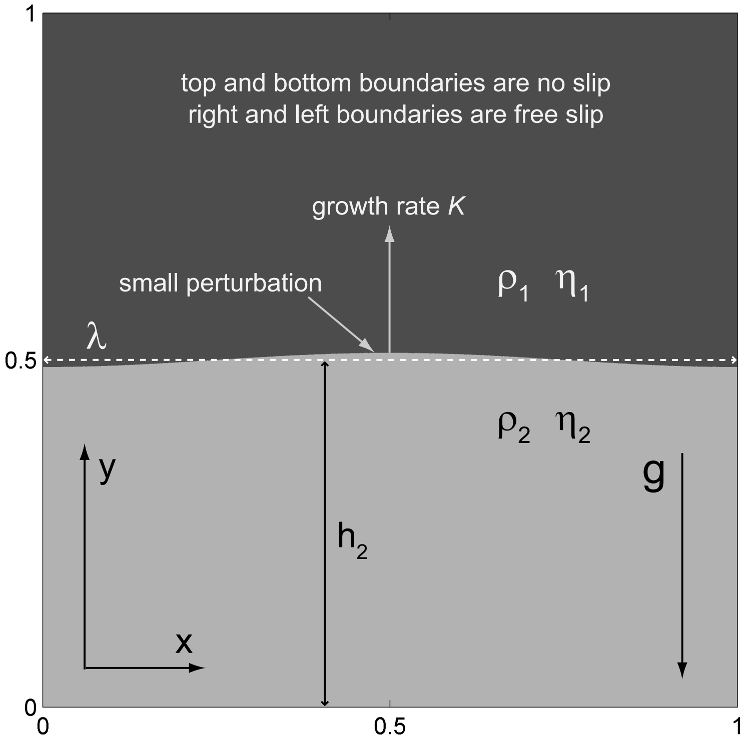
1
0.5
0
0
0.5
1
top and bottom boundaries are no slip
right and left boundaries are free slip
growth rate *K*
small perturbation
λ
ρ1 η1
ρ2 η2
y
x
h2
g

Figure 28. Rayleigh-Taylor model initial setup.

Table 5. Model parameters for the Rayleigh-Taylor benchmark.

Parameter	Value
Top layer viscosity η_1	1–2000
Bottom layer viscosity η_2	1–1000
Top layer density ρ_1	1.1
Bottom layer density ρ_2	1.0
Gravitational acceleration g	−10.0
Courant number	0.8

8.5.2. Growth of Diapirs

The series of Rayleigh-Taylor numerical simulations were performed varying viscosity contrast $\frac{\eta_1}{\eta_2}$ and thickness of bottom layer h_2 on both adaptive and non-adaptive grids. The growth factors were calculated after the first time step for both adaptive and non-adaptive models and compared against each other and against analytical solution [43,44], the results are provided on Figure 29. As it is seen from the plots, growth factors obtained on adaptive and non-adaptive grids are practically equal, and good agreement with analytical solution is observed.

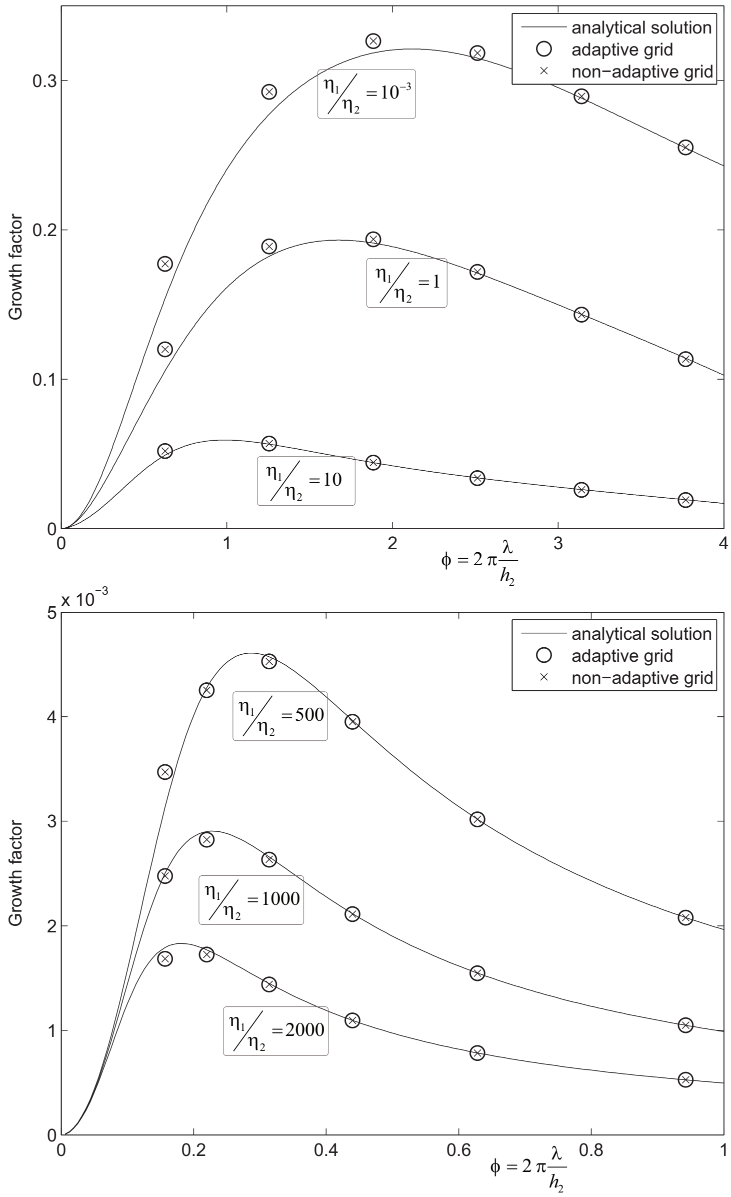

Figure 29. Comparison of numerical and analytical growth rates for Rayleigh-Taylor model.

To illustrate the following diapiric growth, some results for subsequent simulation time steps are shown on Figures 30–32. As it is seen from the figures, the interface between the layers remains accurately captured and resolved by an adaptive grid as simulation proceeds.

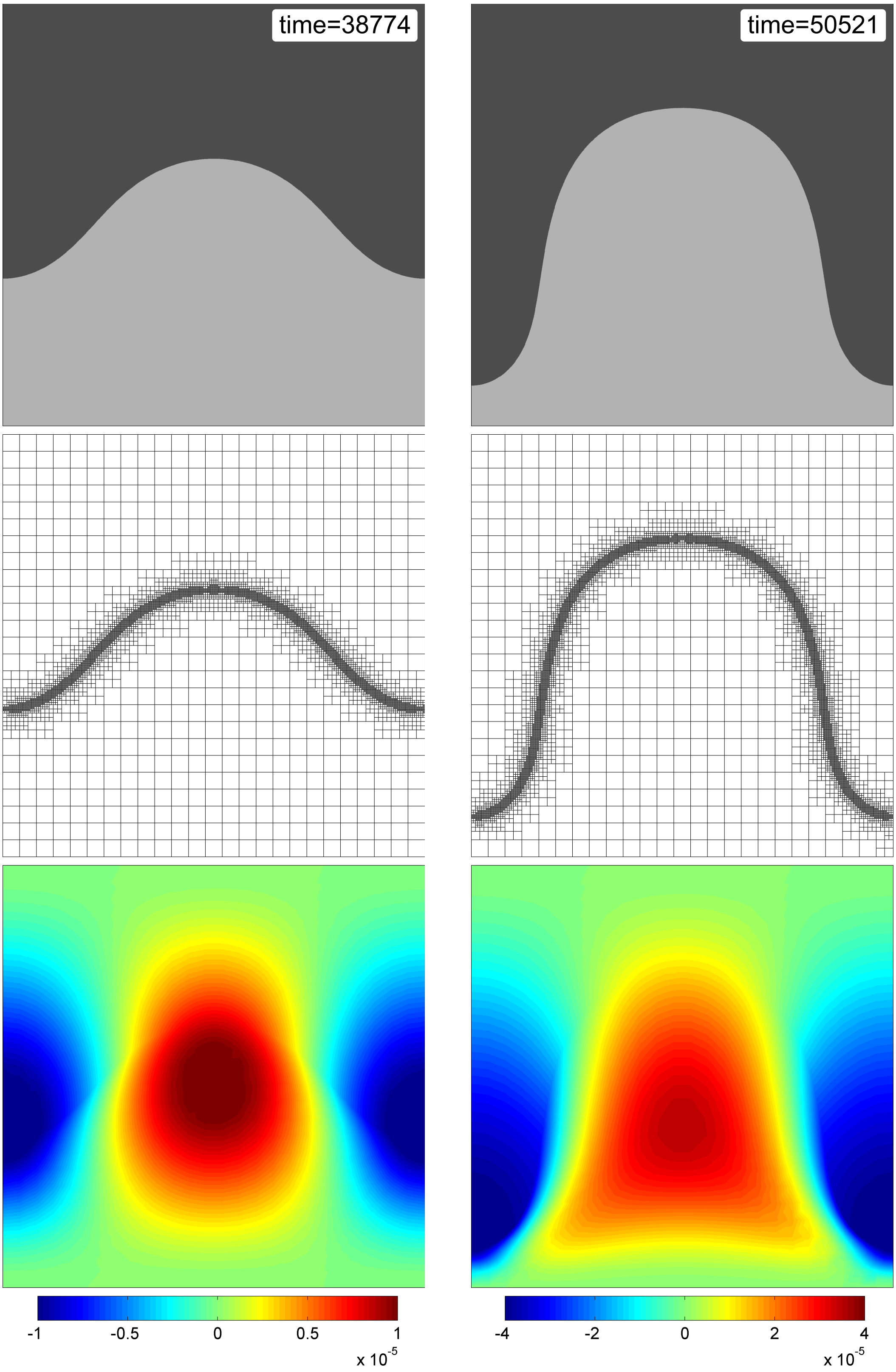

Figure 30. Results of Rayleigh-Taylor numerical experiment with $\frac{\eta_1}{\eta_2} = 10^3$. **top**: material plot; **middle**: numerical grid; **bottom**: v_y-velocity plot.

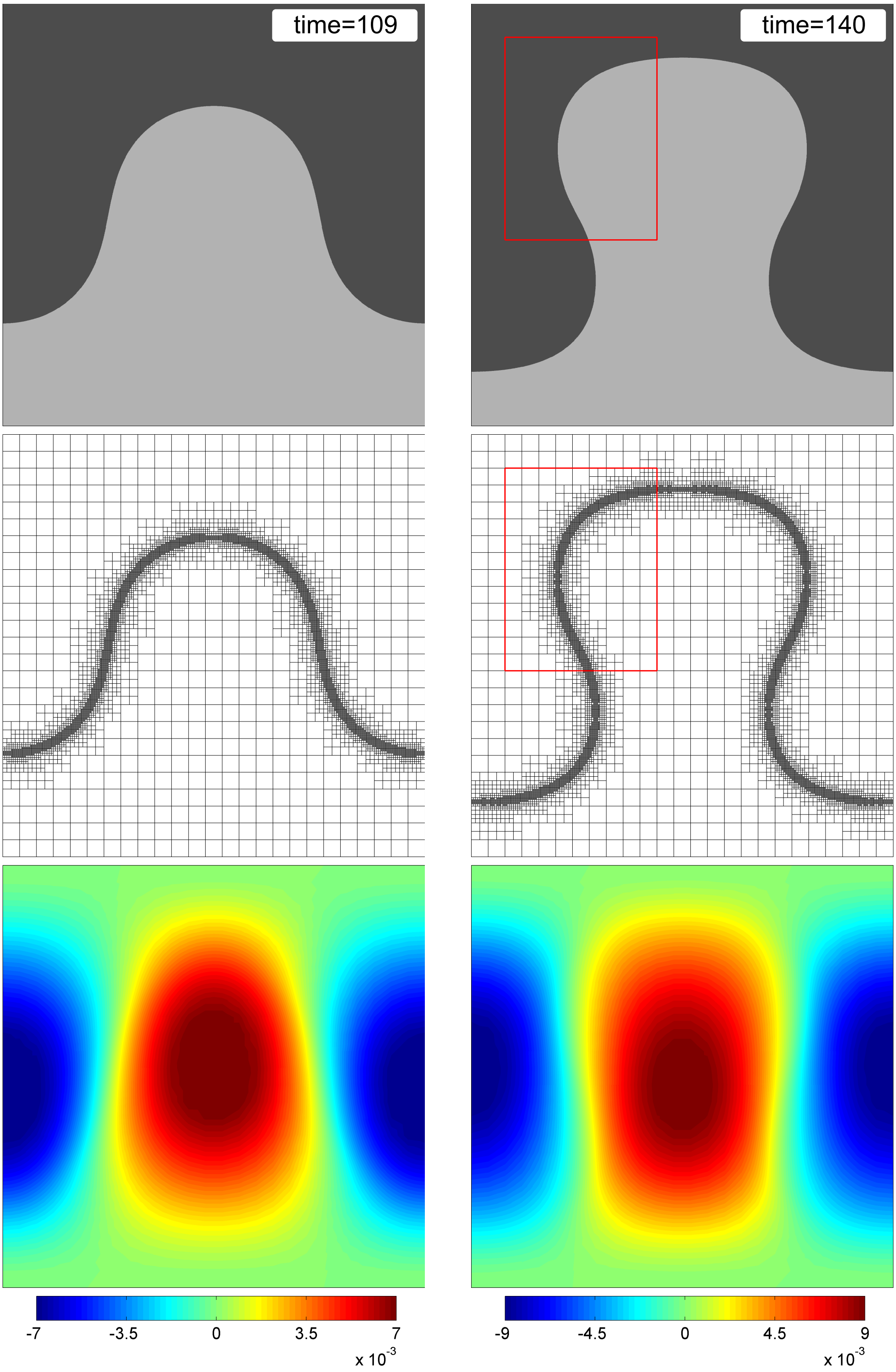

Figure 31. Results of Rayleigh-Taylor numerical experiment with $\frac{\eta_1}{\eta_2} = 1$. **top**: material plot; **middle**: numerical grid; **bottom**: v_y-velocity plot.

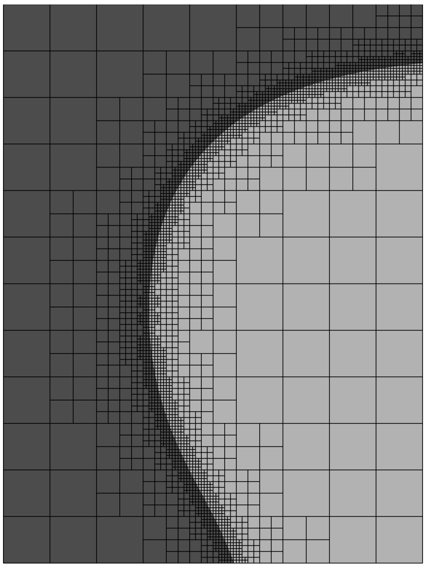

Figure 32. Zoom in Figure 31 at $t = 140$, as marked by red rectangles.

8.5.3. Performance Analysis

The plots provided on Figure 33 show the measured time and memory requirements for a single Cholesky factorization for numerical simulations on adaptive and non-adaptive grids of 100 × 100, 200 × 200, 400 × 400 and 800 × 800 efficient numerical resolutions. The power of adaptive approach is evident—both time and memory requirements can differ by as high as 3 orders of magnitude (the 800 × 800 case).

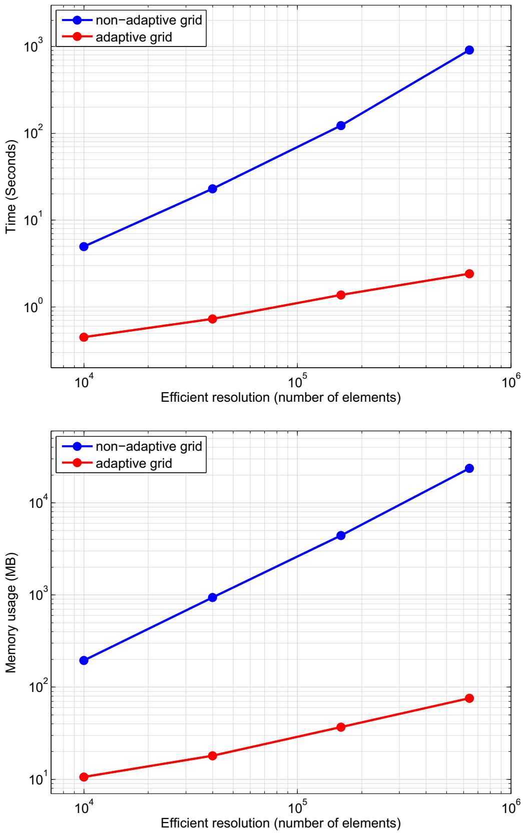

Figure 33. Time (**top**) and memory requirements (**bottom**) for Cholesky factorization depending on numerical resolution. Quad AMD Opteron 8220 system was used.

8.5.4. Effect of Voronoi Tessellation

Due to the relatively large deformations, the Rayleigh-Taylor instability benchmark serves as a good test to check the efficiency of the Voronoi tessellation algorithm to maintain uniform distribution of Lagrangian particles. The numerical simulations of diapiric growth with and without Voronoi algorithm were performed on Q_2P_{-1}-based $25 \times 25 \to 100 \times 100$ (3 levels) adaptive grid, and with 2.5×10^5 Lagrangian particles. For the experiment with enabled Voronoi tessellation, the minimum and maximum cell areas were $AV_{\min} = \frac{1}{50}$ and $AV_{\max} = \frac{1}{12}$. The effect of the Voronoi algorithm on a Lagrangian particle distribution is illustrated on Figure 34, and dependency plot of the number of particles on time is provided on Figure 35. As it is seen from the Figure 34, the Voronoi tessellation algorithm works very efficiently in keeping uniform particle density across the domain.

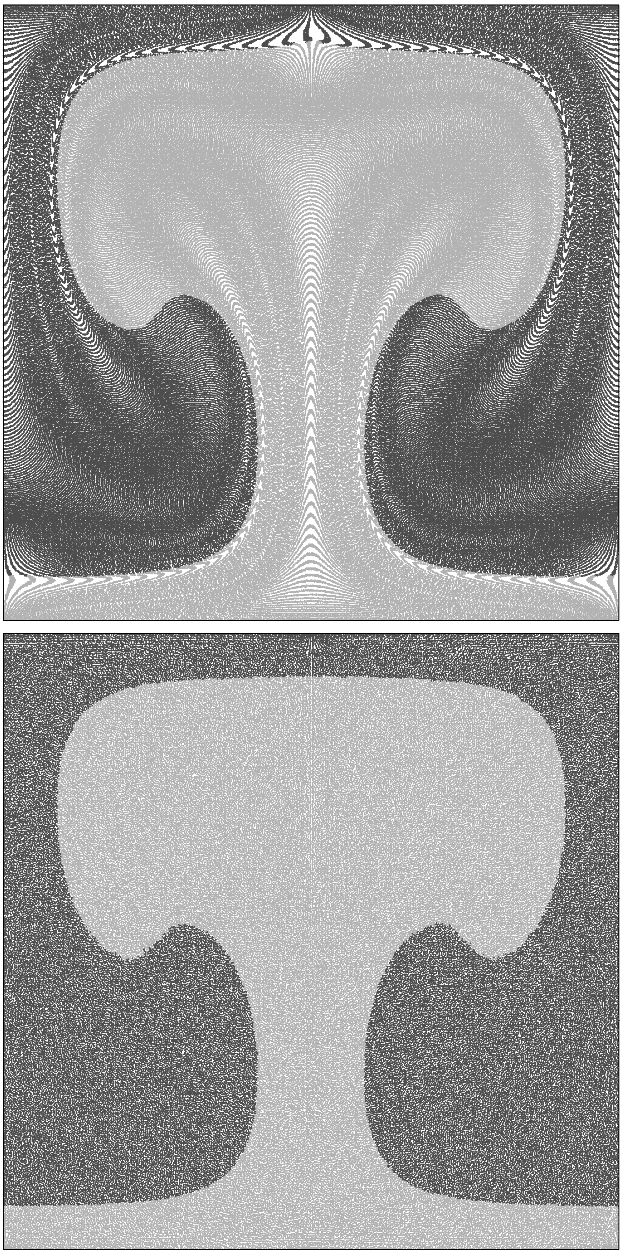

Figure 34. Effect of the Voronoi tessellation algorithm on a Lagrangian particle distribution. The tessellation is disabled (**top**)/enabled (**bottom**).

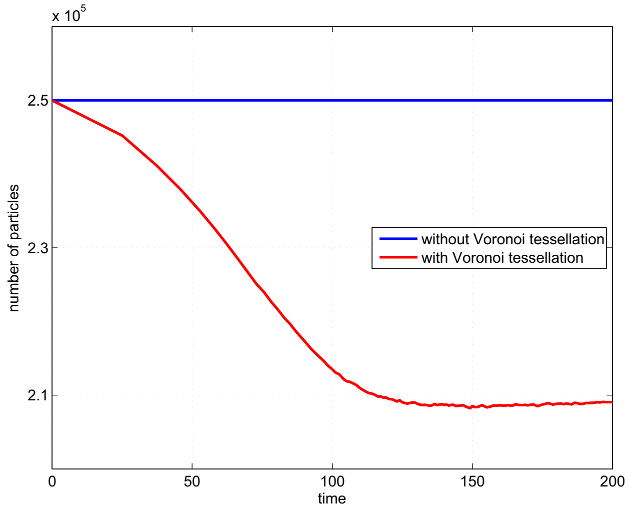

Figure 35. Number of Lagrangian particles as a function of time when Voronoi tessellation algorithm is disabled/enabled.

9. Conclusions

In this work, a new adaptive multiresolution finite element method for the Stokes flow problems is presented (the implementation is available for download from http://sites.google.com/site/sdvigus/ (last accessed on 28 June 2022)). The automatic grid adaptation is supported for bilinear and biquadratic mixed discretizations and is implemented using an original algorithm based on wavelet analysis [4,6]. The proposed method was applied to linear (lateral viscosity variation, sinking block and Rayleigh-Taylor instability benchmarks) and to non-linear (brittle extension/compression benchmark) problems. The obtained numerical results are shown to be in an excellent agreement with those obtained on non-adaptive grids and with analytical solutions. At the same time, computational requirements of the method are up to few orders of magnitude lower compared to the non-adaptive grid approach, in terms of both time and memory usage.

Author Contributions: Data curation, Y.A.M. and T.V.G.; investigation, Y.A.M., O.V.V. and T.V.G.; methodology, Y.A.M. and O.V.V.; resources, T.V.G.; supervision, O.V.V. and T.V.G.; validation, Y.A.M.; visualization, Y.A.M.; writing, original draft preparation, Y.A.M., O.V.V. and T.V.G.; writing, review and editing, Y.A.M., O.V.V. and T.V.G. All authors read and agreed to the published version of the manuscript.

Funding: Y.A.M. and T.V.G. were supported by ETH Research Grant 0-20437-07.

Acknowledgments: The authors appreciate fruitful discussions with Dave May and Boris Kaus. Jean-Pierre Burg is thanked for discussions and support of this study.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

FEM	finite element method
Q_1P_0	bilinear form of FEM
Q_1Q_1	bilinear form of FEM
Q_2P_{-1}	biquadratic form of FEM
CPU	central processing unit

References

1. Albers, M. A Local Mesh Refinement Multigrid Method for 3-D Convection Problems with Strongly Variable Viscosity. *J. Comput. Phys.* **2000**, *160*, 126–150. [CrossRef]
2. Stadler, G.; Gurnis, M.; Burstedde, C.; Wilcox, L.C.; Alisic, L.; Ghattas, O. The Dynamics of Plate Tectonics and Mantle Flow: From Local to Global Scales. *Science* **2010**, *329*, 1033–1038. [CrossRef] [PubMed]

3. Gerya, T.V.; May, D.A.; Duretz, T. An adaptive staggered grid finite difference method for modeling geodynamic Stokes flows with strongly variable viscosity. *Geochem. Geophys. Geosyst.* **2013**, *14*, 1200–1225. [CrossRef]
4. Vasilyev, O.V.; Bowman, C. Second-Generation Wavelet Collocation Method for the Solution of Partial Differential Equations. *J. Comput. Phys.* **2000**, *165*, 660–693. [CrossRef]
5. Vasilyev, O.V. Solving Multi-Dimensional Evolution Problems with Localized Structures Using Second Generation Wavelets. *Int. J. Comp. Fluid Dyn.* **2003**, *17*, 151–168. [CrossRef]
6. Vasilyev, O.V.; Gerya, T.V.; Yuen, D.A. The application of multidimensional wavelets to unveiling multi-phase diagrams and in situ physical properties of rocks. *Earth Planet. Sci. Lett.* **2004**, *223*, 49–64. [CrossRef]
7. Schneider, K.; Vasilyev, O.V. Wavelet Methods in Computational Fluid Dynamics. *Ann. Rev. Fluid Mech.* **2010**, *42*, 473–503. [CrossRef]
8. Nejadmalayeri, A.; Vezolainen, A.; Brown-Dymkoski, E.; Vasilyev, O.V. Parallel Adaptive Wavelet Collocation Method for PDEs. *J. Comp. Phys.* **2015**, *298*, 237–253. [CrossRef]
9. Ranalli, G. *Rheology of the Earth*; Chapman and Hall: London, UK, 1995.
10. Fullsack, P. An arbitrary Lagrangian-Eulerian formulation for creeping flows and its application in tectonic models. *Geophys. J. Int.* **1995**, *120*, 1–23. [CrossRef]
11. Moresi, L.; Solomatov, V. Mantle convection with a brittle lithosphere: Thoughts on the global tectonic styles of the Earth and Venus. *Geophys. J. Int.* **1998**, *133*, 669–682. [CrossRef]
12. Tackley, P.J. Self-consistent generation of tectonic plates in time-dependent, three-dimensional mantle convection simulations, 1, Pseudoplastic yielding. *Geochem. Geophys. Geosyst.* **2000**, *1*. [CrossRef]
13. Chakrabarty, J. *Theory of Plasticity*; Butterworth-Heinemann: Oxford, UK, 2006.
14. Lemiale, V.; Mühlhaus, H.B.; Moresi, L.; Stafford, J. Shear banding analysis of plastic models formulated for incompressible viscous flows. *Phys. Earth Planet. Inter.* **2008**, *171*, 177–186. [CrossRef]
15. Zienkiewicz, O.C.; Morgan, K. *Finite Elements and Approximation*; Wiley: New York, NY, USA, 1983.
16. Fortin, M.; Brezzi, F. *Mixed and Hybrid Finite Element Methods*; Springer: New York, NY, USA, 1991.
17. Elman, H.C.; Silvester, D.J.; Wathen, A.J. *Finite Elements and Fast Iterative Solvers*; Oxford University Press: Oxford, UK, 2005.
18. Dohrmann, C.R.; Bochev, P.B. A stabilized finite element method for the Stokes problem based on polynomial pressure projections. *Int. J. Num. Meth. Fluids* **2004**, *46*, 183–201. [CrossRef]
19. Bochev, P.B.; Dohrmann, C.R.; Gunzburger, M.D. Stabilization of low-order mixed finite elements for the Stokes equations. *SIAM J. Num. Anal.* **2006**, *44*, 82–101. [CrossRef]
20. Zienkiewicz, O.C.; Taylor, R.L.; Zhu, J.Z. *The Finite Element Method: Its Basis and Fundamentals*; Butterworth-Heinemann: Oxford, UK, 2005.
21. Zienkiewicz, O.C.; Vilotte, J.P.; Toyoshima, S.; Nakazawa, S. Iterative method for constrained and mixed approximation. An inexpensive improvement of FEM performance. *Comp. Meth. Appl. Mech. Eng.* **1985**, *51*, 3–29. [CrossRef]
22. Dabrowski, M.; Krotkiewski, M.; Schmid, D.W. MILAMIN: MATLAB-based finite element method solver for large problems. *Geochem. Geophys. Geosyst.* **2008**, *9*, Q04030. [CrossRef]
23. Schmid, D.W.; Dabrowski, M.; Krotkiewski, M. Evolution of large amplitude 3D fold patterns: A FEM study. *Phys. Earth Planet. Inter.* **2008**, *171*, 400–408. [CrossRef]
24. May, D.A.; Moresi, L. Preconditioned iterative methods for Stokes flow problems arising in computational geodynamics. *Phys. Earth Planet. Inter.* **2008**, *171*, 33–47. [CrossRef]
25. Cuvelier, C.; Segal, A.; van Steenhoven, A.A. *Finite Element Methods and Navier-Stokes Equations*; Reidel: Dordrecht, The Netherlands, 1986.
26. Gerya, T.V.; Yuen, D.A. Characteristics-based marker-in-cell method with conservative finite-differences schemes for modeling geological flows with strongly variable transport properties. *Phys. Earth Planet. Inter.* **2003**, *140*, 293–318. [CrossRef]
27. Gerya, T.V.; Yuen, D.A. Robust characteristics method for modelling multiphase visco-elasto-plastic thermo-mechanical problems. *Phys. Earth Planet. Inter.* **2007**, *163*, 83–105. [CrossRef]
28. O'Neill, C.; Moresi, L.; Müller, D.; Albert, R.; Dufour, F. Ellipsis 3D: A particle-in-cell finite-element hybrid code for modelling mantle convection and lithospheric deformation. *Comp. and Geosci.* **2006**, *32*, 1769–1779. [CrossRef]
29. Gerya, T.V. *Introduction to Numerical Geodynamic Modelling*; Cambridge University Press: Cambridge, UK, 2010.
30. OzBench, M.; Regenauer-Lieb, K.; Stegman, D.R.; Morra, G.; Farrington, R.; Hale, A.; May, D.A.; Freeman, J.; Bourgouin, L.; Mühlhaus, H.; et al. A model comparison study of large-scale mantle-lithosphere dynamics driven by subduction. *Phys. Earth Planet. Inter.* **2008**, *171*, 224–234. [CrossRef]
31. Velic, M.; May, D.; Moresi, L. A Fast Robust Algorithm for Computing Discrete Voronoi Diagrams. *J. Math. Model. Algorithm.* **2009**, *8*, 343–355. [CrossRef]
32. Boggess, A.; Narcowich, F.J. *A First Course in Wavelets with Fourier Analysis*; Prentice Hall: Upper Saddle River, NJ, USA, 2001.
33. Sweldens, W. The Lifting Scheme: A Construction of Second Generation Wavelets. *SIAM J. Math. Anal.* **1998**, *29*, 511–546. [CrossRef]
34. Sweldens, W. Wavelets and the lifting scheme: A 5 min tour. *Z. Angew. Math. Mech.* **1996**, *76*, 41–44.
35. Bangerth, W.; Kayser-Herold, O. Data structures and requirements for hp finite element software. *ACM Trans. Math. Softw.* **2009**, *36*, 1486529. [CrossRef]

36. Carey, G.F. *Computational Grids: Generation, Adaptation and Solution Strategies*; Taylor and Francis: Washington, DC, USA, 1997.
37. Chen, Y.; Davis, T.A.; Hager, W.W.; Rajamanickam, S. Algorithm 887: CHOLMOD, Supernodal Sparse Cholesky Factorization and Update/Downdate. *ACM Trans. Math. Softw.* **2008**, *35*, 1–14. [CrossRef]
38. Duff, I.S. MA57—A code for the solution of sparse symmetric definite and indefinite systems. *ACM Trans. Math. Softw.* **2004**, *30*, 118–144. [CrossRef]
39. Amestoy, P.R.; Enseeiht-Irit.; Davis, T.A.; Duff, I.S. Algorithm 837: AMD, an approximate minimum degree ordering algorithm. *ACM Trans. Math. Softw.* **2004**, *30*, 381–388. [CrossRef]
40. Duretz, T.; May, D.A.; Gerya, T.V.; Tackley, P.J. Discretization errors and free surface stabilization in the finite difference and marker-in-cell method for applied geodynamics: A numerical study. *Geochem. Geophys. Geosyst.* **2011**, *12*, Q07004. [CrossRef]
41. Zhong, S. Analytic solutions for Stokes' flow with lateral variations in viscosity. *Geophys. J. Int.* **1996**, *124*, 18–28. [CrossRef]
42. Kaus, B.J.P. Factors that control the angle of shear bands in geodynamic numerical models of brittle deformation. *Tectonophysics* **2010**, *484*, 36–47. [CrossRef]
43. Ramberg, H. Instability of layered systems in the field of gravity. I. *Phys. Earth Planet. Inter.* **1968**, *1*, 427–447. [CrossRef]
44. Ramberg, H. Instability of layered systems in the field of gravity. II. *Phys. Earth Planet. Inter.* **1968**, *1*, 448–474. [CrossRef]