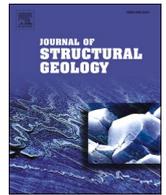




Contents lists available at ScienceDirect

## Journal of Structural Geology

journal homepage: [www.elsevier.com/locate/jsg](http://www.elsevier.com/locate/jsg)

# A simple computer program for calculating stress and strain rate in 2D viscous inclusion-matrix systems

William R. Halter<sup>\*</sup>, Emilie Macherel, Stefan M. Schmalholz

*Institute of Earth Sciences, University of Lausanne, Switzerland*

## ARTICLE INFO

### Keywords:

Numerical modelling  
Finite difference method  
Iterative method  
Viscous inclusion

## ABSTRACT

Computer-based numerical solutions of geomechanical problems are important to understand the processes forming rock structures as well as to quantify the associated pressure, stresses and strain rates. However, the development of such computer programs and the underlying mathematical methods are commonly not taught in a standard structural geology curriculum. Here, we present a simple computer program to calculate the stress, pressure, velocity and strain rate fields for two-dimensional (2D) viscous inclusion-matrix systems under pure shear and simple shear. The main aim of our contribution is to explain this computer program in a simple and transparent way, so that it can be used for introductory courses on geomechanical numerical modelling in structural geology. We present the governing equations of 2D viscous deformation and program the equations in the same order and style, so that the equations are still recognizable in the computer program. The computer program can treat stiff and weak inclusions of any shape and considers both linear and power-law viscous flow laws. We present numerical calculations for various inclusion-matrix scenarios. The program is written with the software MATLAB, is provided as supplementary material, and can also be run with the freely available software GNU Octave.

## 1. Introduction

John G. Ramsay made outstanding contributions to the science and teaching of structural geology. These contributions are manifested, for example, in scientific articles and particularly in four textbooks (Ramsay, 1967; Ramsay and Huber, 1983, 1987; Ramsay and Lisle, 2000). In the latest textbook, Ramsay and Lisle (2000) wrote: “We are convinced that the research advances which go on today and which will continue in the future will involve both detailed observations of naturally deformed rock systems together with a deeper appreciation of the theoretical application of mechanics. Both are essential to progress”. They further wrote: “We have considered carefully the part that computer-based numerical methods should play in the solutions of geomechanical problems and we have come to the conclusion that numerical solutions are more than justified”. In this study, we focus on such computer-based numerical methods to solve the mathematical equations of mechanics which quantify the deformation of natural rock systems.

University education in structural geology commonly focuses on detailed observations of naturally deformed rock systems, the geometrical classification of structures in deformed rock and the kinematics (i.e.

temporal evolution of geometry) associated with the generation of such structures. The mechanical, or dynamical, aspects of rock deformation, especially analytical and computer-based solutions, are commonly not treated to the same extent and detail (e.g. Mukherjee, 2019). However, several authors have emphasized the importance of considering the complete mechanical aspects and computer-based solutions for understanding the formation of geological structures (e.g. Fletcher and Pollard, 1999; Pollard and Fletcher, 2005; Allmendinger, 2011). One reason why analytical and computer-based solutions still often play a minor role only in a structural geology curriculum is likely that the mathematical treatment of supposedly simple mechanical problems, such as the two-dimensional (2D) deformation of elliptical inclusions embedded in a mechanically weaker or stronger matrix, is mathematically already quite complex (e.g. Schmid and Podladchikov, 2003; Pollard and Fletcher, 2005; Jaeger et al., 2007; Moulas et al., 2014). Furthermore, the application of computer-based solutions to mechanical problems of rock deformation requires a basic knowledge of numerical methods and programming, which is often not imparted in a standard structural geology curriculum. Therefore, during their studies many students of structural geology do often not acquire the skills to write a

<sup>\*</sup> Corresponding author.

E-mail address: [william.halter@unil.ch](mailto:william.halter@unil.ch) (W.R. Halter).

<https://doi.org/10.1016/j.jsg.2022.104617>

Received 30 November 2021; Received in revised form 25 April 2022; Accepted 28 April 2022

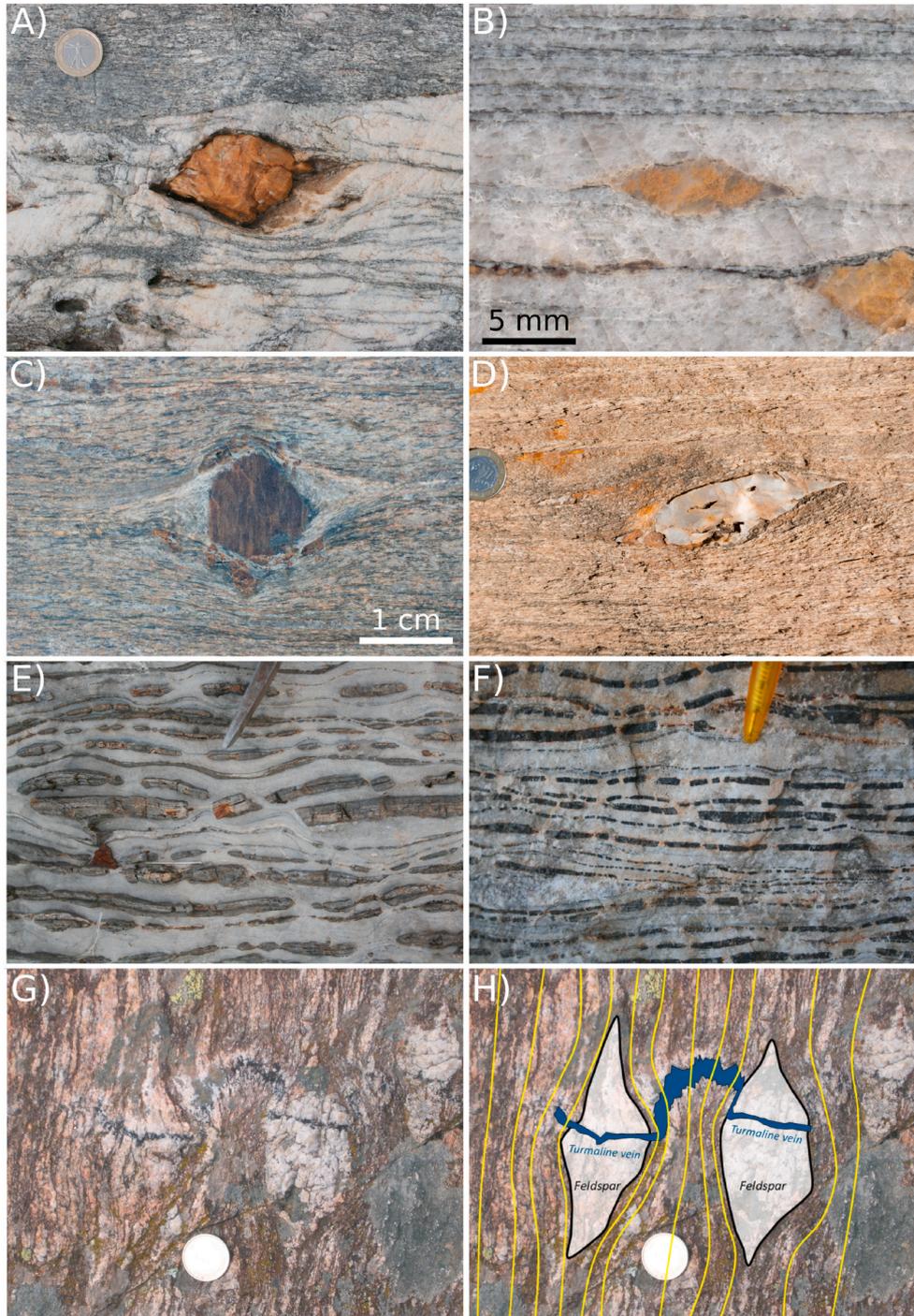
Available online 9 May 2022

0191-8141/© 2022 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

simple computer program which can, for example, calculate the pressure, velocity or strain rate fields for deforming inclusion-host systems. However, computer-based solutions for geomechanical problems are becoming more and more important in academia and industry to predict and quantify rock deformation processes (e.g. Kolditz et al., 2015; Pandey et al., 2018). Particularly, computer-based solutions are important for investigating the coupling between rock deformation, fluid flow and chemical reactions (e.g. Poulet et al., 2012; Kolditz et al., 2015; Pandey et al., 2018; Evans et al., 2020; Schmalholz et al., 2020). Understanding such coupled processes is essential for solving fundamental scientific questions, such as the interplay between metamorphic reactions and rock deformation or between fluid flow and rock deformation, and for solving applied problems concerning, for example,

geothermal energy exploitation or permanent storage of carbon dioxide in geological reservoirs (e.g. Matter and Kelemen, 2009).

Here, we present a simple computer program that numerically solves the mathematical equations describing the 2D viscous deformation of inclusion-matrix systems under pure shear and simple shear (Fig. 1). Such inclusion-matrix systems have also been considered in the textbook of Ramsay and Lisle (2000). An overview of many performed analytical, numerical and laboratory-experimental studies of deforming inclusion-matrix systems is given in Jessell et al. (2009). Typical examples of stiff inclusions in a weaker matrix are porphyroclasts or blocky boudins (Fig. 1) as well as crystal-melt mixtures (e.g. Yamato et al., 2015). Examples of weak inclusions in a stronger matrix are all kind of fluid inclusions, partially-molten or molten regions within



**Fig. 1.** Natural examples of inclusion-matrix systems. A) Calcite porphyroclasts in quartz, Neves area, Tauern window, Italy. B) Calcite porphyroclasts in mylonitic vein quartz, Neves area, Tauern window, Italy. C) Garnet in micaceous schist, Le Conquet, France. Width of garnet is 1 cm. D) Quartz in schist, Tinos, Greece. E) Boudinaged calc-silicate layers in limestone, Monte Frerone, Adamello, Italy. F) Boudinaged calc-silicate layers in limestone, Monte Frerone, Adamello, Italy. G) Gneiss with two feldspars transected by a turmaline vein, Gorges d'Héric, Montagne Noire, France. During deformation, the turmaline vein remained straight inside the strong feldspars, while it was shortened and folded in the matrix between the feldspars. H) Simplified sketch of photo shown in G). Photos A) and B) are courtesy of N. Mancktelow and G. Pennacchioni, photos C) and D) courtesy of N. Mancktelow and photos E), F) and G) and sketch in H) by S.M. Schmalholz. The calcite porphyroclasts in quartz are described in detail in Mancktelow and Pennacchioni (2010).

non-molten rock, such as migmatites (e.g. Saki et al., 2020) or magma chambers, or phyllosilicate-rich metasedimentary units within the continental crust (e.g. Petri et al., 2019). In engineering geology, the rock deformation around open fractures, bore holes or tunnels represents a typical geomechanical problem which can be also considered as an inclusion-matrix system with extremely weak inclusion (e.g. Schmid and Podladchikov, 2003; Jaeger et al., 2007; Moulas et al., 2014). The presented computer program can treat stiff and weak inclusions of any shape and considers linear and power-law viscous flow laws. For simplicity, we do not consider other, but also important, deformation mechanisms such as elastic deformation, or brittle-frictional deformation used to describe faulting by applying a yield criterion, such as the Mohr-Coulomb or the Drucker-Prager yield criterion (e.g. Jaeger et al., 2007). The main aim of our study is to present the mathematical equations, the numerical method and the computer program in a very simple and transparent way, so that our contribution might be used for introductory courses on geomechanical numerical modelling in structural geology.

A particular challenge in learning computer-based numerical methods is to understand the link between the original equations and the computer program that solves these equations, because these equations are typically completely obscured inside the program by a variety of numerical procedures. Even when the governing equations are understood by the students, the associated computer program solving these equations often appears to the students, in our experience, as non-transparent. In many numerical methods the equations are dismembered (e.g. separation into a coefficient matrix and a vector for the unknown variables) and/or various external computer codes (so-called libraries) are used which are often unknown to the student. In contrast, in the computer program presented here the original equations are still recognizable so that the link between the governing equations and the computer program is transparent and understandable. The computer program is based on the finite difference method (e.g. Gerya, 2019; Anderson et al., 2020) and we solve the finite difference equations with a simple iterative method (Frankel, 1950). The computer program is written in the programming language MATLAB and provided as supplementary material. This MATLAB program can also be run with the freely available software GNU Octave. It can, hence, be used on any computer without the need for specific software licences or additional software (e.g. compiler).

## 2. Mathematical model

### 2.1. Governing equations of continuum mechanics

The investigation of the mechanical aspects of rock deformation with mathematical models requires a system of equations with an equal number of equations and unknown variables which need to be determined. Such a system, also called a closed system of equations, is provided by the concepts of continuum mechanics (e.g. Mase, 1970; Turcotte and Schubert, 2014; Pollard and Fletcher, 2005; Gerya, 2019). In continuum mechanics the deformation of materials is described by partial differential equations because it is assumed that the unknown variables vary in space in a continuous manner (e.g. Mase, 1970; Turcotte and Schubert, 2014; Pollard and Fletcher, 2005; Gerya, 2019). These equations can be distinguished in three sets of equations: (1) Kinematic equations which describe the motion and deformation of a material, for example, the relation between velocities and strain rates in a fluid. (2) Constitutive equations describing the particular mechanical characteristics of the considered material, for example, the behaviour of a viscous fluid or an elastic solid. These constitutive equations include the material properties such as the viscosity. (3) Conservation equations describing the conservation of mass, linear momentum (representing the force balance) and energy. These conservation equations are independent of the considered material and are valid for solids, fluids and gases.

### 2.2. Kinematics

The flow of a viscous material in 2D is quantified by the velocities in the horizontal,  $V_x$ , and vertical,  $V_y$ , direction. The deformation of this viscous material is quantified by three strain rates which are calculated from the spatial gradients of the velocity field (e.g. Mase, 1970)

$$D_{xx} = \frac{\partial V_x}{\partial x} \quad (1)$$

$$D_{yy} = \frac{\partial V_y}{\partial y} \quad (2)$$

$$D_{xy} = \frac{1}{2} \left( \frac{\partial V_x}{\partial y} + \frac{\partial V_y}{\partial x} \right) \quad (3)$$

The horizontal dimension is represented by the x-coordinate and the vertical direction by the y-coordinate. Hence,  $D_{xx}$ ,  $D_{yy}$  and  $D_{xy}$  are the strain rates in the horizontal and vertical direction and the shear strain rate, respectively. We assume here an incompressible flow without any volumetric deformation. Therefore, the strain rates above are termed deviatoric strain rates which do not cause any volumetric deformation, only shear deformation. Strain rates are spatial gradients of the velocity field and have, hence, physical units of  $s^{-1}$ .

### 2.3. Constitutive equations

The constitutive equation, or flow law, for a viscous material is given by a mathematical relationship between deviatoric stress and deviatoric strain rate. Such stress-strain rate relationships for natural rocks and minerals are typically derived from rock deformation experiments (e.g. Karato, 2008). The three deviatoric stresses are (e.g. Turcotte and Schubert, 2014; Gerya, 2019)

$$T_{xx} = 2\eta D_{xx} \quad (4)$$

$$T_{yy} = 2\eta D_{yy} \quad (5)$$

$$T_{xy} = 2\eta D_{xy} \quad (6)$$

where  $\eta$  is the effective viscosity of the fluid. The stresses have units of Pa and the effective viscosity has units of Pa s. We assume that  $T_{xy} = T_{yx}$ , or in other words that the stress tensor is symmetric. The symmetry of the stress tensor can be derived from the balance of angular momentum (e.g. Mase, 1970).

To formulate the force balance equations, or the conservation equation for linear momentum, we need to know the horizontal and vertical forces. Forces are related to total stresses which in fluid dynamics are typically composed of a mean stress, or pressure, and the deviatoric stresses described above. For example, in 2D the horizontal total stress is the mean stress (average of the horizontal and vertical total stresses) plus the horizontal deviatoric stress. The total horizontal and vertical stresses are (e.g. Turcotte and Schubert, 2014; Gerya, 2019)

$$S_{xx} = -P + T_{xx} \quad (7)$$

$$S_{yy} = -P + T_{yy} \quad (8)$$

where  $P = -(S_{xx} + S_{yy})/2$  is the negative mean stress. The definition of pressure as negative mean stress is simply a convention often applied in fluid dynamics, but pressure could also be defined as positive mean stress. Only so-called normal stresses acting in the horizontal and vertical direction (having subscripts xx and yy) have a pressure component, because their orientation is parallel to the hydrostatic, or the principal, stress directions. The reason is that fluids at rest (i.e. static situation with no strain rates) can experience a pressure, for example due to gravity, and this pressure only depends on the total normal stresses. Fluids at rest experience no shear stress and no deviatoric stresses because such

stresses would cause strain rates (see equations (4)–(6)) and, hence, deformation.

#### 2.4. Conservation equations

We assume incompressible deformation, as mentioned above, and the equation for the conservation of mass can then be written as (e.g. Turcotte and Schubert, 2014; Gerya, 2019)

$$0 = \frac{\partial V_x}{\partial x} + \frac{\partial V_y}{\partial y} \quad (9)$$

The right-hand side of the equation above is also termed the divergence of the velocity field. The two equations for the conservation of linear momentum in the absence of gravity and inertial forces are (e.g. Turcotte and Schubert, 2014; Gerya, 2019)

$$0 = \frac{\partial S_{xx}}{\partial x} + \frac{\partial T_{xy}}{\partial y} \quad (10)$$

$$0 = \frac{\partial S_{yy}}{\partial y} + \frac{\partial T_{xy}}{\partial x} \quad (11)$$

Equations (10) and (11) represent the force balance in the horizontal x- and vertical y-direction, respectively.

#### 2.5. Closed system of equations

Equations (1)–(11) constitute a closed system of equations with as many equations as unknown variables so that the system of equations can be solved. The 11 unknown variables are:  $P$ ,  $V_x$ ,  $V_y$ ,  $D_{xx}$ ,  $D_{yy}$ ,  $D_{xy}$ ,  $T_{xx}$ ,  $T_{yy}$ ,  $T_{xy}$ ,  $S_{xx}$ ,  $S_{yy}$ . Only equations (9)–(11) are partial differential equations. These three partial differential equations represent the equations that we will solve numerically for the three unknowns  $P$ ,  $V_x$  and  $V_y$ . If these three unknowns are calculated, then all the remaining 8 unknowns can be calculated using equations (1)–(8). To calculate these three unknowns we will apply, first, a numerical method which approximates all the partial derivatives with ratios of differences (so-called finite differences) and then, second, a numerical solution technique which provides an explicit expression for the unknowns. Selecting pressure and velocities as unknowns, which must be determined, is a typical approach in computational fluid dynamics (e.g. Hughes, 2012; Zienkiewicz et al., 2014). The numerical method and solution technique is explained further below in sections 3.1 to 3.3.

#### 2.6. Effective viscosities

Generally, for flow laws the term effective viscosity refers to the ratio of stress to strain rate (e.g. Karato, 2008). The effective viscosity,  $\eta$ , used in equations (4)–(6) can represent different types of viscous flow. If  $\eta$  is constant, then it represents linear viscous (Newtonian) flow and we term here such linear viscosity  $\eta_L$ . Such linear viscous flow is often considered as representative for diffusion creep (e.g. Turcotte and Schubert, 2014).

For a non-linear, power-law viscous flow, the effective viscosity depends on the strain rate, or alternatively on the stress. The effective viscosity for a power-law viscous fluid, termed here  $\eta_{PL}$ , can be written as (e.g. Fletcher, 1974; Schmalholz and Schmid, 2012)

$$\eta_{PL} = \eta_L \left( \frac{T_{II}}{T_R} \right)^{1-n} \quad (12)$$

where  $\eta_L$  is the linear viscosity,  $T_R$  is a constant reference stress,  $n$  is the stress exponent, which for rocks is  $\geq 1$ , and

$$T_{II} = \sqrt{\frac{1}{2} (T_{xx}^2 + T_{yy}^2 + 2T_{xy}^2)}. \quad (13)$$

$T_{II}$  is called the square root of the second invariant of the deviatoric stress tensor. This invariant is used because its value is independent on

the chosen coordinate system and is, hence, a suitable quantity to describe the stress magnitude in the deforming fluid. If the value of  $T_{II}$  increases, then the value of  $\eta_{PL}$  decreases. Therefore, regions in the fluid with higher stresses have a smaller effective viscosity than regions in the fluid with smaller stresses. If the stress,  $T_{II}$ , in the fluid is equal to  $T_R$  then  $\eta_{PL} = \eta_L$ . If the stress exponent  $n = 1$  then  $\eta_{PL}$  reduces to  $\eta_L$  everywhere in the fluid and the effective viscosity is linear viscous and constant. A power-law viscous flow is often considered as representative for dislocation creep (e.g. Turcotte and Schubert, 2014).

Often, a combination of linear viscous and power-law viscous flow is assumed for ductile rocks. The viscosity for such combined flow law can be given by the pseudo-harmonic mean of  $\eta_L$  and  $\eta_{PL}$  (e.g. Gerya, 2019)

$$\eta_C = \frac{1}{\frac{1}{\eta_L} + \frac{1}{\eta_{PL}}} \quad (14)$$

The effective viscosity,  $\eta$ , in equations (4)–(6) can, hence, either be equal to  $\eta_L$ ,  $\eta_{PL}$  or  $\eta_C$  depending on whether a linear viscous, power-law viscous or combined flow law, respectively, should be modelled.

#### 2.7. Geological magnitudes of considered quantities and parameters

For applicability reasons, we provide here some typical magnitudes of the involved mechanical quantities in a tectonic context. We consider here viscous deformation only and ignore, for example, visco-elastic effects, fracturing and inertial forces (e.g. generating seismic waves). For such viscous deformation, strain rates vary typically between  $10^{-15}$  to  $10^{-12} \text{ s}^{-1}$ , but can potentially reach values up to  $10^{-10} \text{ s}^{-1}$  (e.g. Pfiffner and Ramsay, 1982; Chernak and Hirth, 2010; Fagereng and Biggs, 2019; Bose and Mukherjee, 2020). Deviatoric stresses vary typically between 1 and few hundreds of MPa, but could locally likely reach values up to 1 GPa (e.g. Karato, 2008, Table 19.2 therein), especially on the mineral scale (e.g. Angel et al., 2015). Typical effective viscosities for melt-free rock range from  $10^{18}$  to  $10^{23} \text{ Pa s}$  (e.g. Karato, 2008; Liu and Hasterok, 2016). The stress exponent  $n$  for power-law viscous flow (equation (12)) is typically 1 for diffusion creep (i.e. linear viscous flow) and between 3 and 5 for dislocation creep (e.g. Karato, 2008, Table 19.1 therein).

### 3. Computer-based numerical model

#### 3.1. Finite difference method

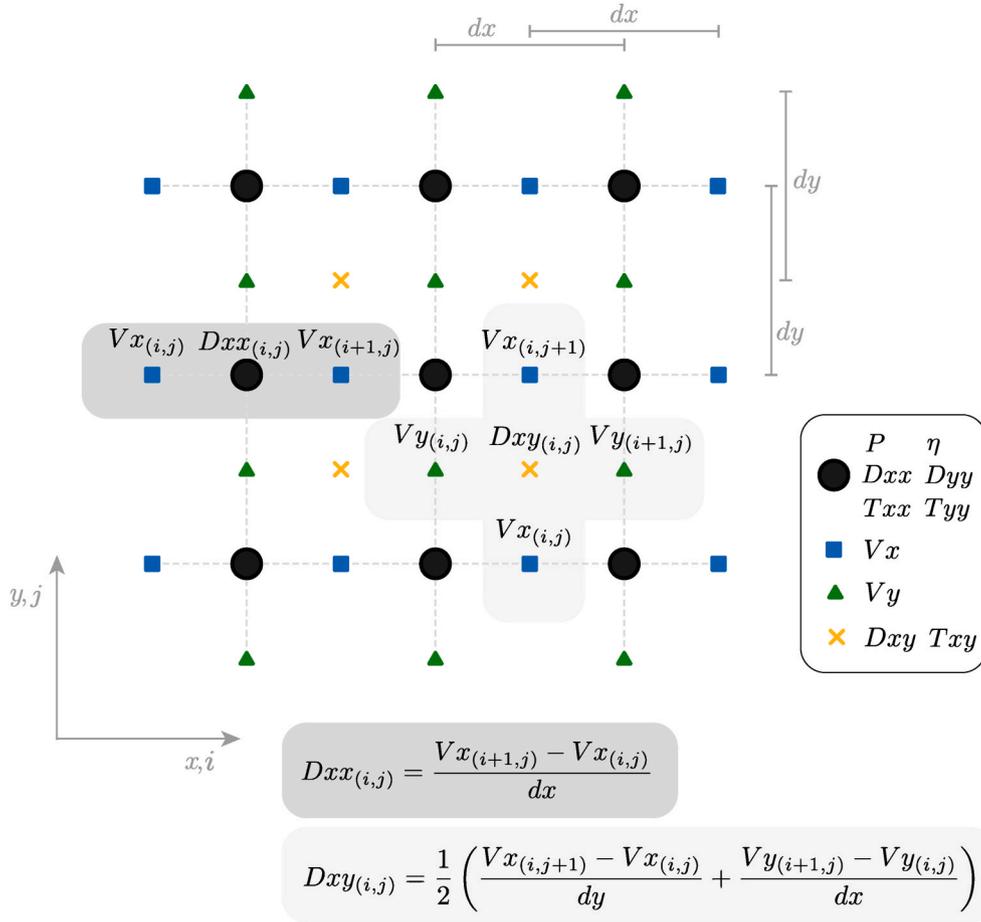
The presented computer program employs the finite difference method. Other numerical methods used in continuum mechanics include, for example, the finite element method (e.g. Hughes, 2012; Zienkiewicz et al., 2014), the discrete element method (e.g. Hardy and Finch, 2005; Schöpfer et al., 2009), and the spectral method (e.g. Schmalholz et al., 2001).

In the finite difference method the model domain is subdivided, or discretized, into a so-called numerical grid, or mesh (Fig. 2). In the so-called staggered finite difference method the unknown variables, here velocities and pressure, are calculated at different positions (e.g. Gerya, 2019). We explain further below why a staggered grid is useful.

The finite difference method approximates a partial derivative, with respect to the x- or y-direction, of any variable by the difference of this variable between two neighboring grid points, divided by the distance between these points. For example, the finite difference approximation of the horizontal spatial derivative of the horizontal velocity is

$$\frac{\partial V_x}{\partial x} = \frac{V_{x(i+1,j)} - V_{x(i,j)}}{dx} \quad (15)$$

The index  $i$  represents the number of a grid point (on which a value of the horizontal velocity is stored) along the x-direction. Hence,  $i$  varies from 1 to  $nx$ , whereby  $nx$  is the number of points storing values of the horizontal velocities along the horizontal direction. These grid points



**Fig. 2.** The applied staggered finite difference grid. Here a representative section with 3x3 grid points (black-filled circles) is shown. Not all variables are located on the grid points.  $V_x$ ,  $V_y$ ,  $D_{xy}$ , and  $T_{xy}$  are located on staggered points (colored shapes) in-between grid points. Equations (1) and (3) in their discretized form are highlighted in dark and light gray, respectively, below the grid. The relative positioning between the calculated strain rates and the used velocities is indicated on the grid.

are indicated with a square in Fig. 2. The value of  $dx$  specifies the horizontal distance between two horizontally neighboring points. Similarly, the index  $j$  represents the number of a point along the  $y$ -direction and  $dy$  specifies the vertical distance between two vertically neighboring points. The employed finite difference method uses an evenly spaced grid so that  $dx$  and  $dy$  have the same values everywhere in the grid (Fig. 2). Smaller  $dx$  and  $dy$  increases the spatial resolution and accuracy of the numerical model, but in turn leads to longer calculation times (e.g. Anderson et al., 2020). The spacial resolution needs to be large enough to resolve the geometry of the studied mechanical process (e.g. Anderson et al., 2020).

Every partial derivative appearing in the governing equations above is approximated by finite differences. These approximations of the spatial derivatives represent the spatial slope, or gradient, between two neighboring points. This spatial gradient, hence, represents the gradient at a position between two neighboring points. For example, in Fig. 2 the horizontal gradient of the horizontal velocity, defined at the points represented by the squares, is representative for the point horizontally between the squares, which is represented by the black-filled circles (Fig. 2). The horizontal gradient of the horizontal velocity determines the horizontal strain rate,  $D_{xx}$  (equation (1)), and hence the values of  $D_{xx}$  are stored at the point indicated by the black-filled circles (Fig. 2). The value of  $D_{xx}$  multiplied by the value of  $\eta$  determines the value of  $T_{xx}$  (equation (4)), which is the reason why values of  $\eta$  and  $T_{xx}$  are stored at the same points (indicated by black-filled circles) as  $D_{xx}$ . The same reasoning applies for vertical gradients of vertical velocities (stored at points indicated by triangles in Fig. 2) defining the vertical strain rates.

Furthermore, horizontal gradients of the vertical velocities and vertical gradients of the horizontal velocities define the shear strain rate,  $D_{xy}$  (equation (3)) and, consequently, values of  $D_{xy}$  are stored in the center points between the points storing the horizontal and vertical velocities, which are indicated with crosses in Fig. 2. Therefore, the fundamental reason for staggering (i.e. storing different quantities at different positions of the numerical grid) is that the governing equations, for example the horizontal force balance equation (10), involves both horizontal and vertical derivatives of different variables. The staggering assures that all spatial derivatives used in an equation are representative for exactly the same position in the numerical grid.

### 3.2. Approximate equations and residuals

Using the finite difference approximation, the three conservation equations (9)–(11) can be discretized in the following way (e.g. Richardson, 1911; Frankel, 1950; Duretz et al., 2019)

$$RES_p = \frac{V_{x(i+1,j)} - V_{x(i,j)}}{dx} + \frac{V_{y(i,j+1)} - V_{y(i,j)}}{dy} \quad (16)$$

$$RES_{\dot{\kappa}} = \frac{S_{xx(i+1,j)} - S_{xx(i,j)}}{dx} + \frac{T_{xy(i,j+1)} - T_{xy(i,j)}}{dy} \quad (17)$$

$$RES_{\dot{\gamma}} = \frac{S_{yy(i,j+1)} - S_{yy(i,j)}}{dy} + \frac{T_{xy(i+1,j)} - T_{xy(i,j)}}{dx} \quad (18)$$

The partial derivatives are approximated by finite differences and,

hence, the conservation equations are not fulfilled exactly so that the left-hand side of the equations is not zero but has an approximation error, which is typically called a residual, for example  $RES_{V_x}$  in equation (17). We term the residuals for the three conservation equations,  $RES_P$ ,  $RES_{V_x}$ , and  $RES_{V_y}$ , respectively.

### 3.3. Iterative solution method

The aim of every numerical method is to find such values for the unknown variables so that the residuals of the approximate equations become as small as possible. There is a large variety of different numerical solutions methods (e.g. Anderson et al., 2020). Here, we solve the approximate, or discretized, system of equations iteratively (Frankel, 1950). This means that we start with an initial guess for the unknown velocities and pressures on all grid points. Then, we iteratively (i.e. subsequently step by step) modify the values for the velocities and pressures so long until we found values which provide small enough residuals so that we can consider the numerical solution as accurate for our purpose. The iteration steps, or cycles, are done in the computer program using a so-called “while [...] end” loop, in which a computer operation is continuously repeated until a certain criterion is reached, in our case a sufficiently small value of the residuals (Fig. 3). The iterative method we use here is similar to the Richardson iterative method (Richardson, 1911; Frankel, 1950) which has been developed more than hundred years ago, before the existence of computers. Originally, the calculations for such iterative methods were done by hand.

The residuals in the approximate equations (16)–(18) depend on the values of the three unknown variables. With every new iteration step these values will be modified and the residual becomes smaller and smaller with more and more iteration steps. The values for the new iteration step can be found by assuming that the difference between the values of the new iteration step ( $P^{it}$ ,  $V_x^{it}$  and  $V_y^{it}$ ) and the old iteration step ( $P^{it-1}$ ,  $V_x^{it-1}$  and  $V_y^{it-1}$ ) is proportional to the respective residuals and scaled by some scalar numbers ( $dpt_P$ ,  $dpt_{V_x}$  and  $dpt_{V_y}$ ) (Richardson, 1911). We choose to use the same scaling  $dpt_V$  for both velocities and express the residuals as (e.g. Duretz et al., 2019)

$$RES_P = \frac{P^{it} - P^{it-1}}{dpt_P} \quad (19)$$

$$RES_{V_x} = \frac{V_x^{it} - V_x^{it-1}}{dpt_V} \quad (20)$$

$$RES_{V_y} = \frac{V_y^{it} - V_y^{it-1}}{dpt_V} \quad (21)$$

where the superscript  $it$  indicates the number of the iteration step. One

can imagine that the scalars  $dpt_P$  and  $dpt_V$  represent the “virtual”, or “pseudo”, time interval between two consecutive iteration steps. The right-hand sides of equations (19)–(21) have then the finite difference form of discretized “pseudo-time” derivatives. Therefore, we decided to term  $dpt_P$  and  $dpt_V$  pseudo-time steps. However, the nomenclature of pseudo-time steps is done here only for illustrative purposes and one could refer to the scalars  $dpt_P$  and  $dpt_V$  simply as iteration parameters. When the values of the unknown variables stop varying between two consecutive iteration steps, then the pseudo-time derivatives of these unknowns become zero and, consequently, the residuals become zero so that the approximate equations are solved. The values of the pseudo-time steps depend on the considered model configuration and material parameters. If  $dpt_P$  and  $dpt_V$  are too large, the residuals do not converge to small values. If  $dpt_P$  and  $dpt_V$  are too small, the residuals converge to small values very slowly. The values suitable for the presented computer simulations are specified in the computer program (Figs. 12 and 13). Each of the three residuals depends on one of the three unknown variables. Solving next the equations above for the new (with respect to a new iteration step) variables  $P^{it}$ ,  $V_x^{it}$ , and  $V_y^{it}$  we obtain (e.g. Richardson, 1911; Frankel, 1950; Duretz et al., 2019)

$$P^{it} = P^{it-1} + dpt_P \cdot RES_P \quad (22)$$

$$V_x^{it} = V_x^{it-1} + dpt_V \cdot RES_{V_x} \quad (23)$$

$$V_y^{it} = V_y^{it-1} + dpt_V \cdot RES_{V_y} \quad (24)$$

Equations (22)–(24) are the explicit expressions for  $P$ ,  $V_x$ , and  $V_y$ , which were missing in section 2.5. During each iteration step, all the values for the pressure and velocities on all the numerical grid points are modified, or updated, according to equations (22)–(24). The iteration steps above are performed so often until the values of the three residuals are smaller than a specified tolerance, or error. When this tolerance is achieved, the current values of the velocities and pressures at the numerical grid points are considered as numerical solution of the 2D viscous flow equations.

### 3.4. Computer algorithm

A computer program to solve the differential equations typically consists of three parts: (1) a pre-processor, which defines the model configuration (i.e. geometry, numerical grid etc.), the material properties (e.g. viscosity) and a variety of required numerical parameters (e.g. tolerance for numerical solution etc.), (2) a processor, which approximately solves the system of equations up to a specified numerical accuracy, and (3) a post-processor, which visualizes and saves the numerical results.

```

1 while error>tol; iter = iter+1; % START of iteration loop
2   DXX = diff(VX,1,1)/dx; % Eq (1)
3   DYY = diff(VY,1,2)/dy; % Eq (2)
4   DXY = 1/2*( diff(VX(2:end-1,:),1,2)/dy ...
5           + diff(VY(:,2:end-1),1,1)/dx ); % Eq (3)
6   TXX = 2.*ETA.*DXX; % Eq (4)
7   TYY = 2.*ETA.*DYY; % Eq (5)
8   TXY = 2.*n2c(ETA).*DXY; % Eq (6)
9   SXX = -P(:,2:end-1)+TXX(:,2:end-1); % Eq (7)
10  SYY = -P(2:end-1,:)+TYY(2:end-1,:); % Eq (8)
11  RES_P = -( diff(VX,1,1)/dx + diff(VY,1,2)/dy ); % Eq (16)
12  RES_VX = diff(SXX,1,1)/dx + diff(TXY,1,2)/dy; % Eq (17)
13  RES_VY = diff(SYY,1,2)/dy + diff(TXY,1,1)/dx; % Eq (18)
14  P = P + dpt_P*RES_P; % Eq (22)
15  VX(2:end-1,2:end-1) = VX(2:end-1,2:end-1) + dpt_V*RES_VX; % Eq (23)
16  VY(2:end-1,2:end-1) = VY(2:end-1,2:end-1) + dpt_V*RES_VY; % Eq (24)
17  TII = sqrt(0.5*(TXX.^2 + TYY.^2 + 2*c2n(TXY).^2)); % Eq (13)
18 end % END of iteration loop

```

Fig. 3. The iterative Matlab solver of the closed system of governing equations. The number of each programmed equation corresponds to the respective equation with the same number presented in sections 2 and 3.

We focus here on the processor, which consists of the iterative solver. The values for the velocities and pressures at every grid point are stored on the computer in so-called 2D arrays, similar to a matrix. For example, the values of the horizontal velocities are stored in the array termed VX (Fig. 3). This array has two dimensions: Along the first dimension the values along the horizontal x-direction are stored, which are numbered with index  $i$ ; along the second dimension the values along the vertical y-directions are stored, which are numbered with index  $j$ , that is  $VX(i = 1, \dots, nx; j = 1, \dots, ny)$ . In the software MATLAB and GNU octave, the difference between neighboring values stored in a 2D array can be simply calculated with the command termed “diff” (Fig. 3). The advantage of this command is that it can calculate the difference of neighboring values, say along the x-direction, in a single computer operation. For example, the differences of all horizontal velocity values in array VX along the horizontal, x-direction, is

$$\text{diff}(VX, 1, 1) = VX_{(i+1,j)} - VX_{(i,j)} \quad (25)$$

This “diff” operation is applied for all points with numbers  $i = 1, \dots, nx-1$  and  $j = 1, \dots, ny$  simultaneously. The maximal value of index  $i$  is  $nx-1$  because in the equation above the grid point with number  $i+1$  is used, which is the final grid point when  $i = nx-1$ . Similarly, the difference of neighboring horizontal velocity values along the vertical, y-direction is

$$\text{diff}(VX, 1, 2) = VX_{(i,j+1)} - VX_{(i,j)} \quad (26)$$

which is applied for all  $i = 1, \dots, nx$  and  $j = 1, \dots, ny-1$  simultaneously. Computer operations as performed by the command “diff” which affect an entire array are called a vectorized operation. The entire iterative solver for linear viscous flow (i.e. constant viscosity) is displayed in Fig. 3. The finite difference equations are programmed in the same order and style as the exact governing equations having partial derivatives. For transparency, the approximate equations in the computer program have the same number as the exact equations presented above. The computer program involving the (i) iteration loop (“while [...] end” loop), (ii) the approximate equations, (iii) the calculation of the residuals and (iv) the iterative update of the three unknown variables is just 18 lines long (Fig. 3).

### 3.5. Model configuration

We consider a quadratic model domain with dimensionless parameters (Fig. 4). All quantities are made dimensionless with a characteristic length scale,  $Lx$ , which is the width of the model, a characteristic time scale, defined by  $1/D_B$ , which is the inverse of the applied far-field pure shear or simple shear strain rate, and a characteristic viscosity,  $\eta_B$ , which is the viscosity of the matrix. All physical quantities in the model can be expressed as a function of these three characteristic values. For example, the x- and y-coordinates are displayed as dimensionless values,  $x/Lx$ , which means that the dimensional x-coordinate (having units of m) is divided by the dimensional model width (having units of m). Also, all values of the pressure,  $P$ , and stresses (having units of Pa), are divided by the product  $2\eta_B D_B$  (having also units of Pa). The quantity of  $2\eta_B D_B$  corresponds to the constant deviatoric stress in a homogeneous material, having viscosity  $\eta_B$ , under far-field pure shear or simple shear with strain rate  $D_B$ . In other words, all pressures and stresses are measured in units of the deviatoric stress for homogeneous shear. After the simulation, the calculated dimensionless results can be converted back to dimensional values by multiplying the dimensionless values by their characteristic scales. Working with dimensionless parameters has the advantage that the numerical result of a single simulation can be applied to a variety of physical parameters. For example, if the dimensionless pressure has a value of 3 and the result should be applied to a natural situation with a viscosity of  $10^{21}$  Pa s and a strain rate of  $10^{-14}$  s $^{-1}$ , then the corresponding dimensional pressure would be 60 MPa ( $3 \cdot 2\eta_B D_B$ ).

To solve the system of equations numerically, we have to specify the

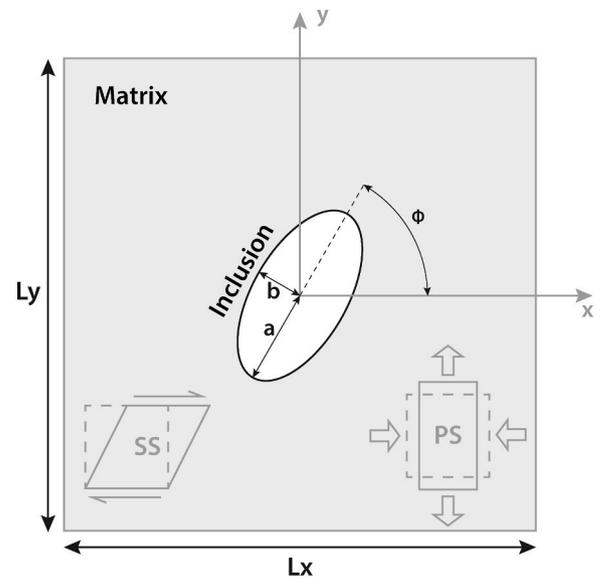


Fig. 4. Model configuration and boundary conditions. An elliptical inclusion of different viscosity than its surrounding matrix is located in the center of a square box of size  $Lx$  by  $Ly$ . The ellipse has a semi-major axis  $a$  and semi-minor axis  $b$  and is rotated by angle  $\phi$  from the horizontal axis. Both simple (SS) and pure (PS) shear boundary conditions can be applied.

values of the unknown variables at the model boundaries, that is, we have to specify so-called boundary conditions (e.g. Gerya, 2019). We apply either a far-field pure shear velocity field at the model boundaries or a simple shear velocity field (e.g. Ramsay and Huber, 1983; Ragan, 2009). Hence, both the horizontal and vertical velocities are defined as boundary conditions at the model boundaries. The model domain consists of a matrix in which stiffer or weaker inclusions are embedded. As initial guess for the unknown velocities, which is required for the first iteration step, we assume that the velocity field in the model domain corresponds to either a homogeneous pure shear or homogeneous simple shear velocity field, depending which far-field velocity boundary conditions are applied. For the pressure, we assume that all values are zero initially. The pressure values at the model boundaries remain zero. Results for specific model configurations are discussed in the following section.

## 4. Results

In this section we present several applications of the presented computer program to show the impact of the viscosity ratio between inclusion and matrix, the inclusion geometry and the pure shear or simple shear boundary conditions on the resulting stress, pressure and strain rate fields. There exist a large number of numerical studies focusing on such inclusion-matrix systems (e.g. Ramsay and Lisle, 2000; Jessell et al., 2009; Marques et al., 2014), and we show here some representative examples using the presented computer program.

### 4.1. Circular inclusion and comparison with analytical solution and finite element model

To test the correctness of our iterative finite difference (FD) program, we compare the obtained results with an analytical solution as well as with a tested finite element method (FEM) program.

First, we compare the presented FD program with the analytical solution for a stressed material including a circular hole (Timoshenko and Goodier, 1970). The FD program approximates the hole as a weak inclusion with a viscosity that is 1000 times smaller than the viscosity of the matrix (Schmid and Podladchikov, 2003). We apply horizontal pure shear extension. The computed normalized horizontal total stress fields

agree with each other (Fig. 5A and B) with both reaching the analytically predicted maximum stress of 3 times the far-field stress above and below the hole (Timoshenko and Goodier, 1970, equation 61 therein). The analytical and numerical solutions are not exactly comparable since the analytical solution is derived for a material with infinite dimension, whereas the numerical solution involves some boundary conditions and a finite size model domain. Nevertheless, the magnitude and distribution of the analytically and numerically calculated stress fields is essentially identical, indicating the correctness of the numerical solution.

Second, we compare the presented FD program against a tested FEM program (Schmalholz and Schmid, 2012) for two model configurations, a weak and a strong circular inclusion, for linear viscous flow. The weak inclusion has a viscosity that is 100 times smaller and the strong inclusion a viscosity that is 100 times larger than the viscosity of the matrix. We apply horizontal pure shear shortening. We compare the calculated pressure fields of our program (Fig. 5C and E) with results of the FEM program (Fig. 5D and F). The FEM program uses a so-called unstructured numerical mesh made of triangular cells of different size and orientation. Understanding and programming this FEM program is less obvious, the governing equations are separated into a stiffness matrix and a vector of unknown variables and external libraries are used, e.g. this FEM program needs an additional program to generate the triangular numerical mesh. The pressure field calculated with our FD program agrees with the pressure field resulting from the FEM program.

The pressure field in the matrix changes the sign when changing from a strong to a weak inclusion which means that regions showing an increased pressure (positive values) for a strong inclusion show a decreased pressure (negative values) for a weak inclusion, and vice versa. For a strong inclusion the pressure variations around the inclusion are larger than for a weak inclusion. This is due to the abrupt changes of the strain rate at the matrix-inclusion boundary for a strong inclusion where the deformation is close to zero. This boundary essentially behaves like a rigid internal boundary with no-slip. For a weak inclusion the pressure variations are smoother. The inclusion-matrix boundary

behaves effectively like an internal free surface. By comparing the results of our program with results from the FEM program we also test the correctness of the boundary conditions since both programs show the same pressure values and distribution along the model boundaries.

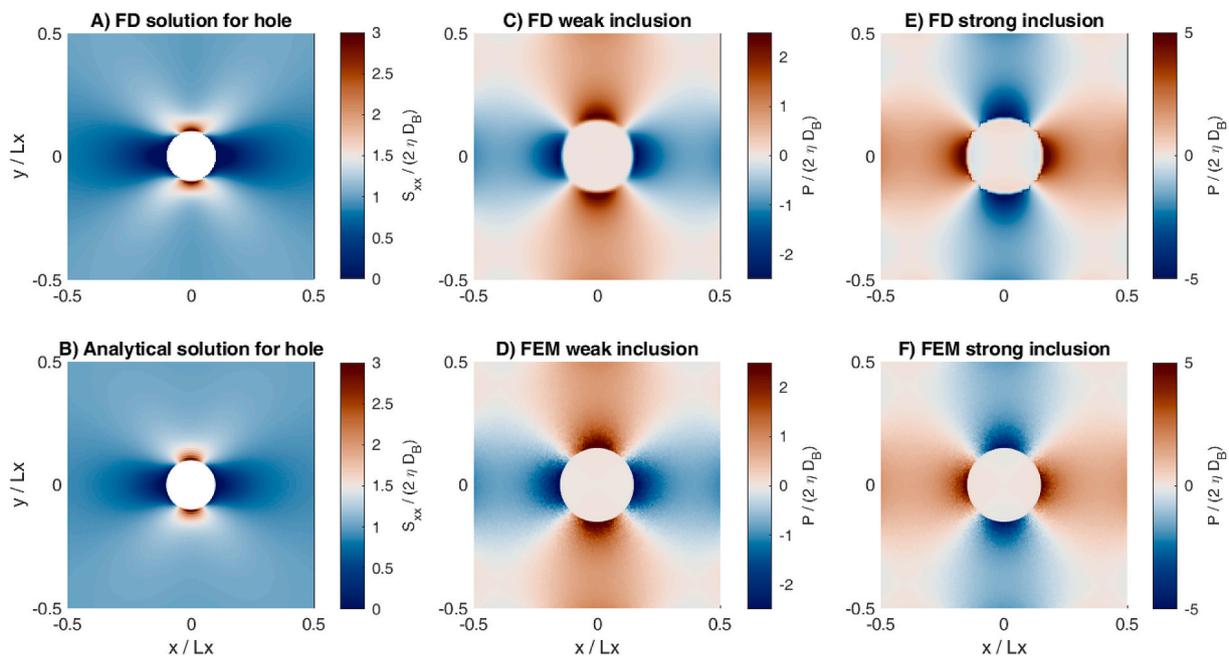
Testing a computer program with analytical solutions or comparing it with results from other computer programs is extremely important, because errors can happen always during programming and often errors in the computer program are not easily recognizable. For example, if there would be a factor 2 error in the numerical results presented in Fig. 5A, then the stress field would look reasonable, and the factor 2 error can only be detected by the comparison with an analytical solution or the comparison with another program.

#### 4.2. Rectangular inclusion under pure shear and simple shear

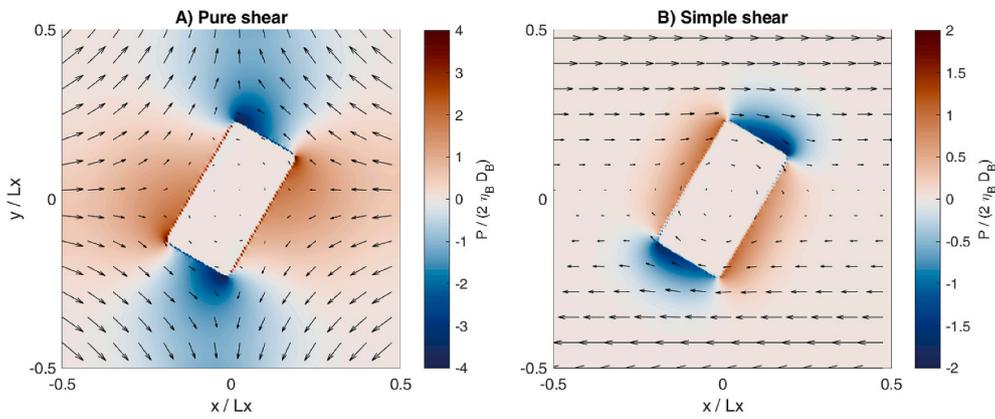
With the presented computer program both pure shear and simple shear boundary conditions can be applied. For illustration purposes, we calculated the pressure fields for a strong and inclined rectangular inclusion for pure shear and simple shear (Fig. 6). For simple shear, the pressure variations are more localized around the inclusion compared to the pressure variations for pure shear. Also, for simple shear the velocity field indicates a rotational flow of the inclusion with a clock-wise sense, in agreement with the applied simple shear (Fig. 6B).

#### 4.3. Power-law viscous matrix and weak elliptical inclusion

We apply a power-law viscous flow law for the matrix which is described by a combined linear and power-law viscous flow law. The boundary conditions impose a horizontal pure shear shortening. To calculate a power-law viscous flow, we have to add a few lines within the iterative solver of our program (Figs. 7, 12 and 13). These lines include the equation for the effective power-law viscosity (equation (12)) and for the effective viscosity of a combined flow law (equation (14)). The additional line numbered 6 (Fig. 7) is included to model a combined



**Fig. 5.** Panels A) and B): Comparison of the normalized horizontal total stress field generated with the presented iterative finite difference (FD) program against an analytical solution for a circular hole within a matrix subjected to horizontally extensional pure shear. The hole in the FD program is approximated by a viscosity contrast of 1000. The hole radius is 0.1 times the model width. Panels C) to F): Comparison of the normalized pressure field generated with the presented iterative FD program against a tested finite element method (FEM) program (Schmalholz and Schmid, 2012) for both a weak and a strong circular inclusion under horizontally compressive pure shear. The regions of high pressure (red color) and the low pressure (blue color) are inverted from the weak inclusion to the strong inclusion setup. The viscosity of the inclusion is 100 times lower, respectively higher, than the surrounding matrix viscosity. The circle radius is 0.15 times the model width. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)



**Fig. 6.** Comparison between horizontally compressive pure shear and horizontal simple shear boundary conditions for a strong (viscosity contrast of 1000) rectangular inclusion. Regions of pressure accumulation are depicted in red, pressure shadow zones in blue. The arrows display the velocity field, they are not to scale and only for direction. The long side of the rectangle is twice the size of its short side and corresponds to 0.4 times the model width. The long side of the rectangle is rotated by an angle of 60° from the horizontal axis. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

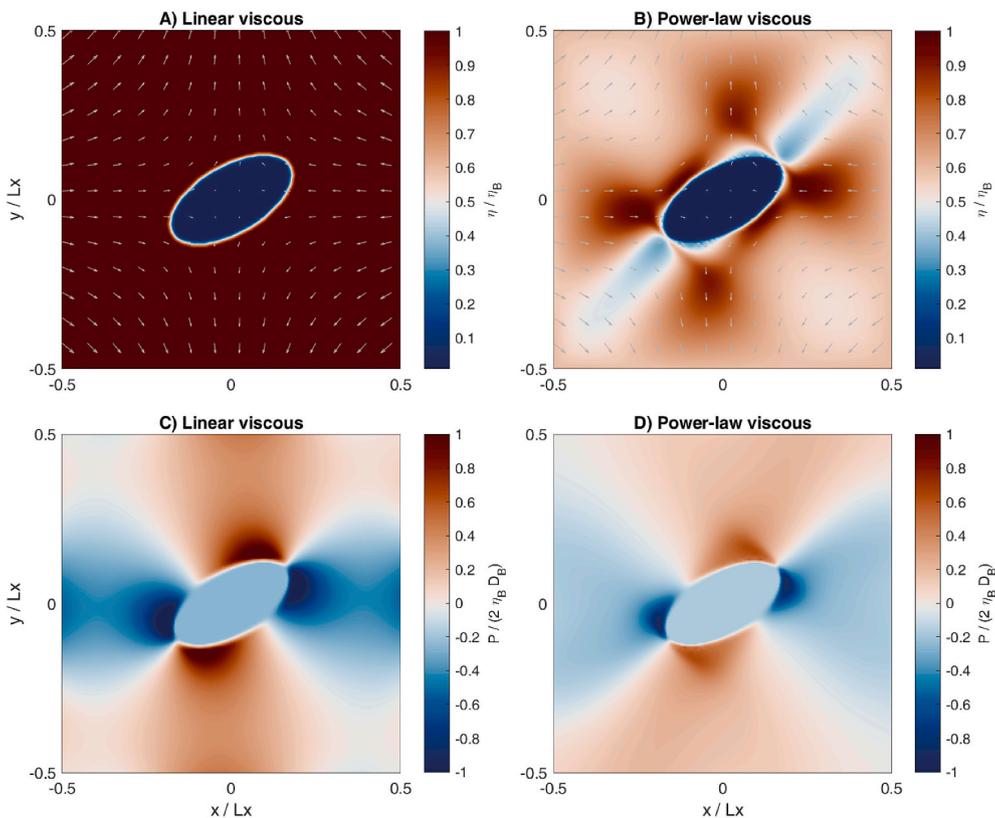
```

1  if n_exp>1 % Power-law viscosity
2      ETA_PL_it      = ETA_PL;           % Viscosity of previous iteration step
3      ETA_PL        = ETA.*(TII/s_ref).^(1-n_exp); % Eq (12)
4      ETA_PL        = exp(log(ETA_PL)*0.5+log(ETA_PL_it)*0.5);
5      ETA           = 1./( 1./ETA_L + 1./ETA_PL ); % Eq (14)
6      ETA(inside)   = ETA_L(inside); % Power-law viscosity only applied to matrix
7      ETA(ETA<etamin) = etamin; % Minimum viscosity
8  end
    
```

**Fig. 7.** The Matlab implementation of the effective viscosity calculation based on a power-law viscosity. The number of each programmed equation corresponds to the respective equation with the same number presented in section 2.6.

power-law viscous flow only inside the matrix while the viscosity inside the inclusion is linear viscous. In line numbered 7 it is possible to define a minimum viscosity to limit the viscosity reduction due to the power-law behaviour. The resulting effective viscosity in the matrix varies now as a function of the stress, which is quantified by the value of  $T_{II}$

(Fig. 8B). For comparison, we also show results for the same model configuration, but for linear viscous flow (Fig. 8A and C). The power-law viscous flow law has an effect on the pressure field which shows more abrupt changes (Fig. 8D) compared to the pressure field for a linear viscosity (Fig. 8C).



**Fig. 8.** Viscosity and pressure field of a weak (viscosity ratio of 1000) elliptical inclusion under pure shear. A model with a purely linear viscosity (left panels) is compared against a model with an effective viscosity including power-law viscosity (right panels). The power-law viscosity is calculated using a stress exponent of 5 and a constant reference stress of 1.5 (see equation (12)). The arrows display the velocity field, they are not to scale and only for direction. The ellipse's semi-major axis is twice as long as its semi-minor axis and corresponds to 0.2 times the model width. The semi-major axis is rotated by 30° from the horizontal axis.

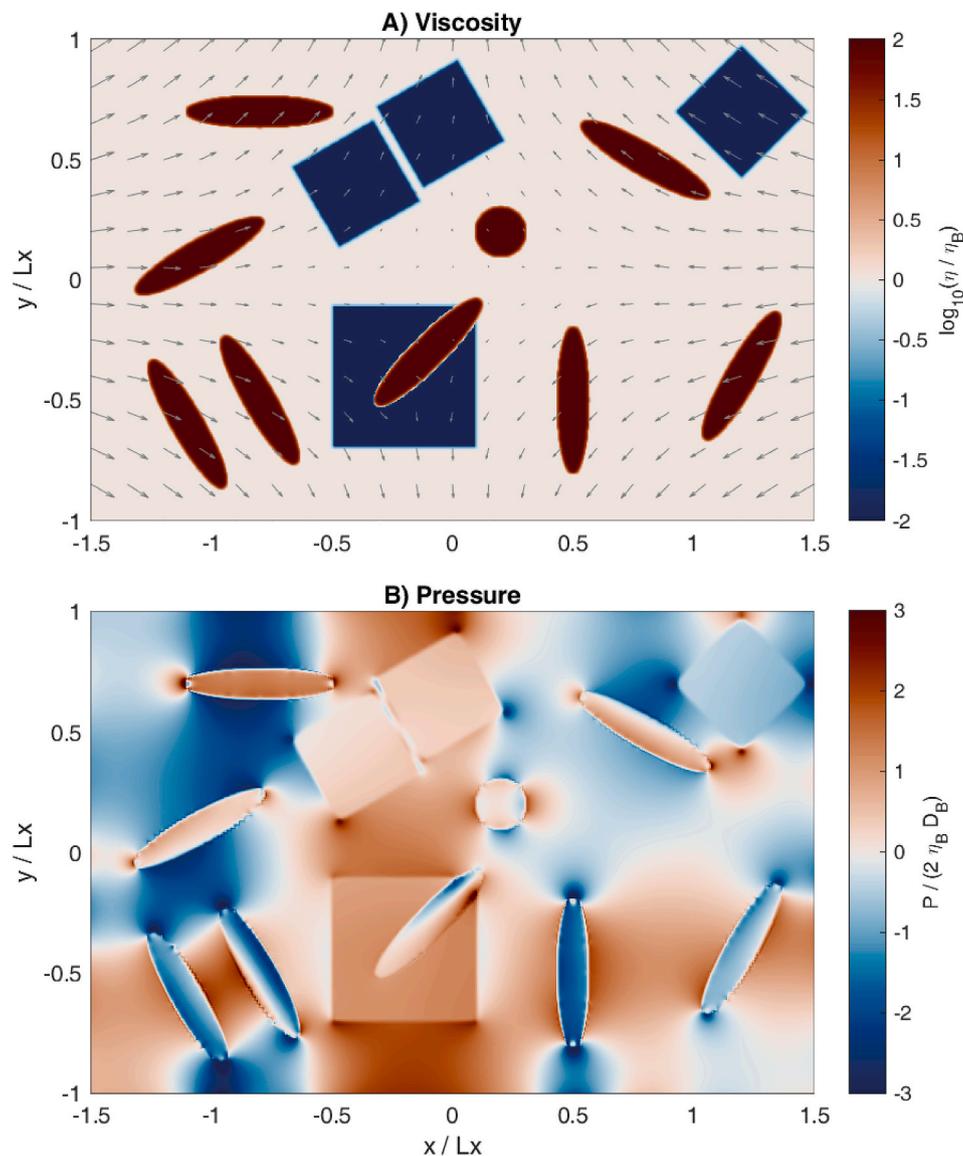
#### 4.4. Multiple inclusions of different shape

To show that our program can model multiple inclusions of various shape and strength we present numerical results for a model configuration including circular, elliptical and quadratic inclusions of different orientation (Fig. 9). We apply horizontal pure-shear shortening and linear viscous flow. The elliptical and circular inclusions have a viscosity 100 times larger than the one of the matrix and the quadratic inclusions have a viscosity 100 times smaller than the one of the matrix (Fig. 9A). The viscosity contrast of 100 represents an inclusion which is not entirely rigid with respect to the matrix (e.g. Schmid and Podladchikov, 2003). The pressure inside the elliptical inclusions depends on their orientation with respect to the shortening direction (Fig. 9B). The elliptical inclusions can exhibit both higher or lower pressure than the corresponding pressure in a homogeneous matrix without inclusions (which would have a dimensionless value of zero; see also Moulas et al., 2014). Pressure variations are largest around the corners of the quadratic inclusions or the boundaries with largest curvature for the elliptical inclusions. In a matrix with more than one inclusion the

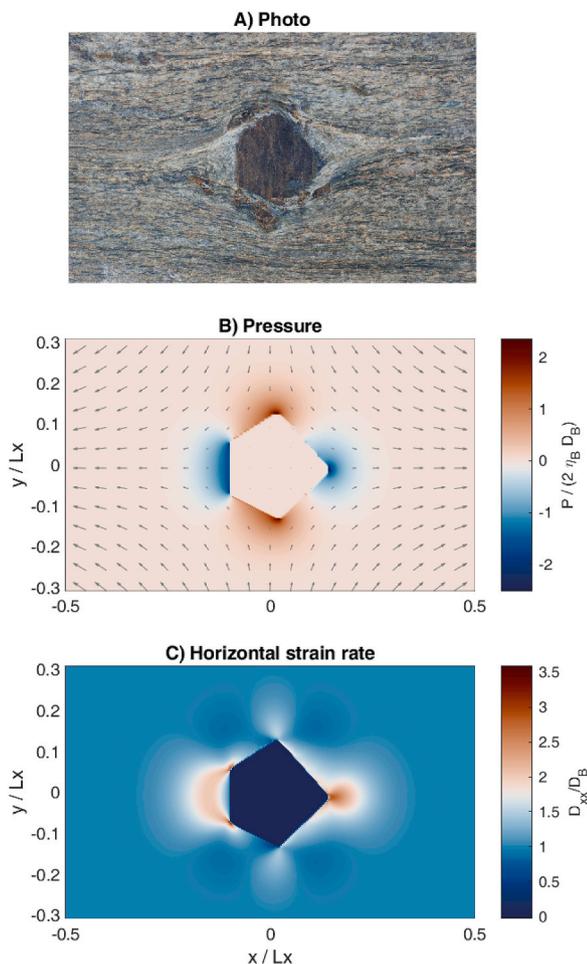
pressure variations caused by the inclusions can interact with each other. For example, high pressure is focused in the region between the two inclined ellipses in the bottom left region of the model (Fig. 9B). Similarly, the entire vertical column close to the horizontal center of the model, passing through three weak square inclusions, also constitutes one single connected high pressure zone (Fig. 9B). Interaction between inclusions is further visible by the variation of pressure inside the inclusions. For example, the elliptical inclusion partly embedded in the quadratic inclusion around the model centre shows a strong variation in pressure inside the elliptical inclusion (Fig. 9B).

#### 4.5. Inclusion with a more natural shape

The presented computer program is not limited to simple inclusion geometries such as circles, ellipses and rectangles, and can consider any geometrical shape, for example, more natural inclusion shapes like the one of the garnet shown in Fig. 10. The garnet is assumed to be nearly rigid compared to the surrounding matrix. Hence, a viscosity contrast of 1000 can reasonably well approximate the contrast of a rigid inclusion



**Fig. 9.** Viscosity and pressure for a combination of different geometrical weak and strong inclusions subjected to horizontally compressive pure shear. The arrows in panel A) display the velocity field, they are not to scale and only for direction. The ellipses are 100 times more viscous, the squares are 100 times less viscous than the surrounding matrix. Except for the circular inclusion, all ellipses possess the same aspect ratio.



**Fig. 10.** Numerical modelling of the pressure and strain rate field around a garnet inclusion using a photo from an actual inclusion (Fig. 1 C). In the model the strong inclusion has a viscosity contrast of 1000 and is subjected to horizontally extensional pure shear. The garnet was redrawn as a polygon using the Matlab `ginput` function. The arrows in panel B) display the velocity field, they are not to scale and only for direction.

in a matrix (e.g. Schmid and Podladchikov, 2003). We apply a horizontal pure shear extension and consider linear viscous flow. The pressure variation around the vertical, left side of the inclusion has a different shape than the pressure variations around the corner-like, right side of the inclusion (Fig. 10B). This difference is even more visible in the variation of the horizontal strain rate (Fig. 10C).

To better interpret the calculated result, the non-dimensional values can be rescaled (section 3.5) using the characteristic values for length scale ( $Lx$ ), background strain rate ( $D_B$ ) and background viscosity ( $\eta_B$ ). For example, the pressure field (Fig. 10B) ranges from approximately  $-2$  to  $2$  in the non-dimensional unit  $P/(2\eta_B D_B)$ . Assuming geologically realistic values for the characteristic values, e.g.  $Lx = 0.05$  m,  $\eta_B = 10^{20}$  Pa s and  $D_B = 10^{-13}$  s $^{-1}$ , the pressure field can be rescaled to range from approximately  $-40$  MPa to  $40$  MPa. These pressure values correspond to the so-called dynamic, or tectonic, pressure which needs to be added to the lithostatic pressure to obtain the total pressure for the natural situation (e.g. Mancktelow, 2008). The same calculation can be done for the horizontal deviatoric strain rate (Fig. 10C) which has a maximal value of approximately  $3.5 D_{xx}/D_B$ . This value is then rescaled to approximately  $3.5 \cdot 10^{-13}$  s $^{-1}$ . The value of  $D_{xx}$  inside the garnet inclusion is zero, indicating that the inclusion is rigid and not deforming.

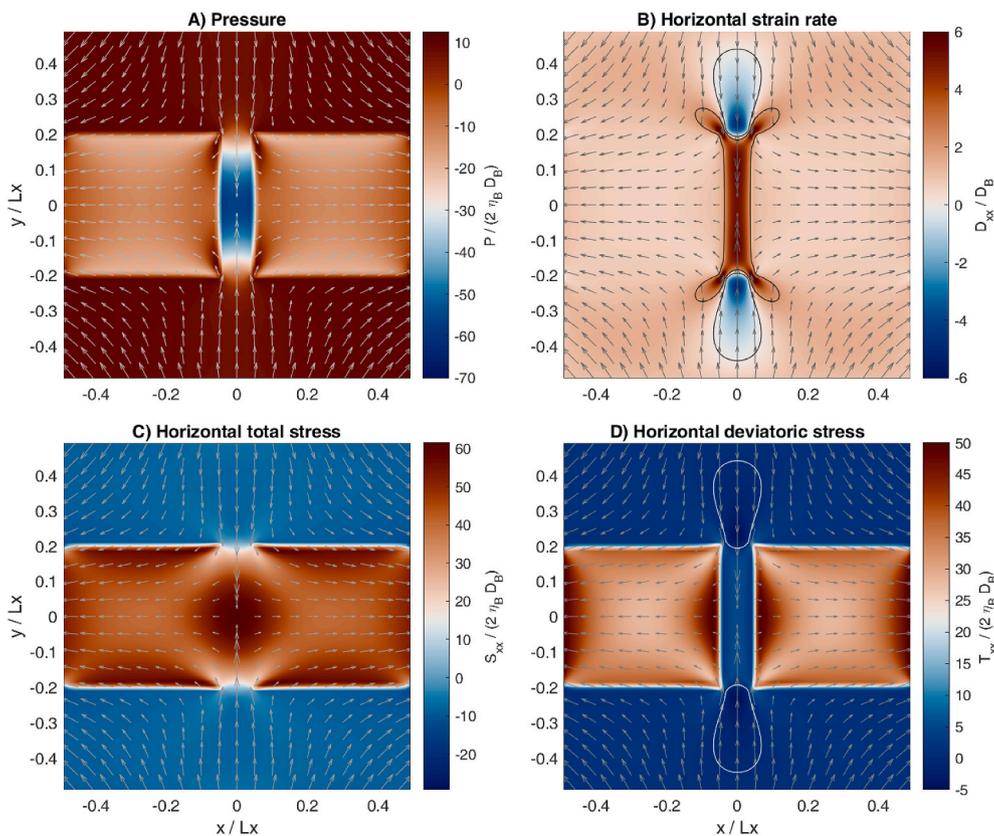
#### 4.6. Blocky boudinage

The last example applicable to blocky boudinage is motivated by the natural boudinage structures shown in Fig. 1E and F and by numerical simulations performed by Mancktelow (2008) (Fig. 6 therein). We apply horizontal pure shear extension, consider linear viscous flow and the viscosity of the boudins is 50 times larger than the ones of the matrix. The vertical gap between the boudins has an aspect ratio of 4 (height to width, Fig. 11). Such calculations of blocky boudinage are among the first numerical calculations applied to geomechanical problems in structural geology (Strömgård, 1973; Selkman, 1978; Mancktelow, 2008). The calculated pressure, stress, and strain rate fields are not intuitive (at least for beginners in mechanical modelling of rock deformation structures). There is an underpressure in the gap between the boudins which generates a hydrodynamic suction force (Fig. 11A). This force sucks matrix material into the gap. As a consequence, the horizontal strain rate in the matrix, above and below the gap, shows horizontal compression although the matrix is under far-field extension (Fig. 11B). The gap is filled with less viscous matrix material and, hence, the horizontal deviatoric stress is much lower in the gap than in the blocky boudins (Fig. 11D). However, the horizontal total stress, horizontally-along the boudins and across the weak gap, is more or less continuous (Fig. 11C). This continuation of the horizontal total stress across a weak gap is a consequence of the horizontal force balance across the gap. Such force balance effects have been also proposed as a mechanism to generate tectonic overpressure in weak crustal shear zones (Schmalholz and Podladchikov, 2013). Because the magnitude of the horizontal deviatoric stress in the gap is small (Fig. 11D), the magnitude of the horizontal total stress, and hence of the horizontal force, is controlled by the (absolute) magnitude of the pressure, which is significantly larger than the magnitude of the deviatoric stress (Fig. 11A).

#### 5. Discussion

The presented calculations show that computer-based solutions are useful to understand the distribution of deformation, stress and pressure in inclusion-matrix systems because even simple scenarios of inclusion-matrix systems yield counter-intuitive results with respect to, for example, pressure perturbations. For example, the orientation of elliptical strong inclusions has a major impact on the pressure inside the inclusion (Fig. 9). Depending on the orientation, the pressure in the strong elliptical inclusion can be larger (tectonic overpressure when long axis is parallel to shortening direction; red colored ellipse in top left region of Fig. 9) or smaller (tectonic underpressure when long axis is orthogonal to shortening direction; blue colored ellipse in bottom right region of Fig. 9) than the ambient pressure (see also Moulas et al., 2014). Such pressure variations have been considered as potential explanation for observed variation in peak metamorphic pressure inside structurally coherent tectonic nappes (Luisier et al., 2019; Vaughan-Hammon et al., 2021). Generally, the quantification of stresses and pressure during rock deformation is important because stresses and pressure significantly impact, for example, metamorphic reactions, fluid flow or pressure solution in natural rock (e.g. Wheeler, 2018; Moulas et al., 2019). Furthermore, the calculations involving blocky boudins show that under far-field horizontal pure shear extension the horizontal deviatoric strain rates and stresses can be compressive above and below the gaps separating blocky boudins. These compressive strain rates and stresses are associated with an underpressure in the gap which causes a hydrodynamic suction force, which also explains the difficulty in separating blocky boudins separated by a thin gap (Mancktelow, 2008).

Such pressure and stress variations, as mentioned above, are associated with the formation of essentially all structures in deformed rocks (e.g. Ramsay, 1967; Schmalholz and Podladchikov, 1999; Ramsay and Lisle, 2000; Pollard and Fletcher, 2005; Mancktelow, 2008; Schmalholz and Mancktelow, 2016). For simple geometries, such as circular or



**Fig. 11.** Numerical modelling of the pressure, strain rate, total and deviatoric stress distribution around two separated blocky boudins. The blocky boudins are 50 times more viscous than the surrounding matrix and subjected to horizontally extensional pure shear. The arrows display the velocity field, they are not to scale and only for direction. The light bulb-shaped contours in panels B) and D) above and below the gap indicate the contours of zero horizontal strain rate and zero horizontal deviatoric stress, respectively. The configuration of this numerical model is similar to the model configuration used by [Mancktelow \(2008\)](#).

elliptical inclusions, also analytical solutions exist and a numerical solution is in principle not required (e.g. [Jeffery and Filon, 1922](#); [Eshelby and Peierls, 1957](#); [Timoshenko and Goodier, 1970](#); [Pollard and Fletcher, 2005](#)). However, for more natural and irregular geometries and for more complicated flow laws, such as combined linear and power-law viscous flow, numerical solutions are required. Considering more irregular geometries is important, because “*the geometrical form of the competent inclusion clearly has an influence on the pattern of stress orientations in the surrounding medium*” ([Ramsay and Lisle, 2000](#)). The same applies, of course, for weak inclusions. Therefore, we find that simple computer programs, as presented here, are very useful for students in structural geology in order to better understand the mechanical aspects of rock deformation which control the formation of rock structures.

The reason why we chose to present an iterative solver over a direct solver is exclusively for didactic reasons because it is simple and transparent. This simplicity, on the other hand, is the main reason for several disadvantages of the applied solver. These disadvantages are that (i) the solver is not very efficient and needs many iterations to achieve a small tolerance and (ii) it can diverge for significantly different material parameters as applied here so that no solution is found and the applied iterative pseudo-time steps need to be adjusted. There are, of course, more elaborated and efficient iterative solvers (e.g. [Duretz et al., 2019](#); [Räss et al., 2019](#); [Schmalholz et al., 2020](#); [Wang et al., 2021](#)), but these are also more difficult to explain to beginners in numerical modelling. An iterative method which is an elaboration of the presented iterative method is the so-called pseudo-transient method (e.g. [Räss et al., 2020](#); [Wang et al., 2021](#)). This pseudo-transient method is more robust and efficient, especially when the computer program is run on graphical processing units (GPUs) ([Räss et al., 2020](#)). The presented iterative solver is a good basis to continue with such more elaborated numerical solution techniques. Furthermore, for simplicity we consider here only the instantaneous situation of a deforming inclusion-matrix system and we do not calculate the time evolution of the system under progressive deformation. Such progressive deformation has been numerically

investigated in several studies, for example, to understand the formation of S–C and S–C’ structures ([Jessell et al., 2009](#); [Dabrowski et al., 2012](#)) or the crystal-melt segregation in magma ([Yamato et al., 2015](#)).

## 6. Conclusion

We present a simple computer program to quantify velocities, strain rates, stresses, pressure and the effective power-law viscosity in 2D inclusion-matrix systems under pure shear and simple shear. The advantage of the applied finite difference method and the iterative solver is that the equations describing viscous flow are still recognizable in the computer program because these equations are programmed in the same order and style as the mechanical equations are presented in this article. Therefore, the presented computer program is useful to introduce students of structural geology without background in numerical modelling to the governing equations of viscous flow, the finite difference method, the numerical solution technique and to scientific programming. We hope that this article can motivate students to learn how to generate computer-based numerical solutions of geomechanical problems, which are useful to understand and quantify the physical processes that control natural rock deformation and the associated formation of rock structures.

## Author statement

All authors had the idea to write a manuscript which describes a simple computer program to model 2D viscous inclusion-matrix systems, which can be used for teaching in Bachelor and Master courses in structural geology. All authors contributed to the development of the computer program. W.R. Halter and S.M. Schmalholz wrote the manuscript and all authors contributed to the generation of the figures.

```

1 % Iterative finite difference program for 2D viscous deformation, Matlab version
2 % This code is free software under the Creative Commons CC-BY-NC-ND license
3 % Published in W.R.Halter, E.Macherel, and S.M.Schmalholz (2022) JSG, https://doi.org/10.1016/j.jsg.2022.104617
4 clear all, close all, clc
5 ps = 1; % ps = 1 models pure shear; ps = 0 models simple shear
6 etaRatio = 50; % Viscosity ratio (matrix/inclusion)
7 a = 0.2; b = a/2; phi = 30/180*pi; % Parameters defining elliptical inclusion
8 % Numerical parameters
9 nx = 151; ny = nx; % Numerical resolution
10 Lx = 1; Ly = Lx; % Model dimension
11 dx = Lx/(nx-1); dy = Ly/(ny-1); % Grid spacing
12 x = [-Lx/2:dx:Lx/2]; y = [-Ly/2:dy:Ly/2]; % Coordinate vectors
13 x_vx = [x(1)-dx/2, (x(1:end-1) + x(2:end))/2, x(end)+dx/2]; % Horizontal vector for Vx which is one more than basic grid
14 y_vy = [y(1)-dy/2, (y(1:end-1) + y(2:end))/2, y(end)+dy/2]; % Vertical vector for Vy which is one more than basic grid
15 % 3 numerical grids due to staggered grid
16 [X,Y] = ndgrid(x,y); [X_vx,Y_vx] = ndgrid(x_vx,y); [X_vy,Y_vy] = ndgrid(x,y_vy);
17 % Physical parameters
18 eta_B = 1; n_exp = 3;
19 s_ref = 1.5; D_B = 1;
20 % Initialization
21 P = zeros(nx, ny); ETA = eta_B*ones(nx, ny);
22 % Define elliptical inclusion
23 inside = ((X*cos(phi)+Y*sin(phi)).^2/a^2 + (X*sin(phi)-Y*cos(phi)).^2/b^2 < 1);
24 ETA(inside) = eta_B/etaRatio; etamin = 0.1*min(ETA(:));
25 for smo=1:2; Ii = [2:nx-1]; Ij = [2:ny-1]; % Smoothing of the initial viscosity field
26 ETA(Ii,:) = ETA(Ii,:) + 0.4*(ETA(Ii+1,:)-2*ETA(Ii,:)+ETA(Ii-1,:));
27 ETA(:,Ij) = ETA(:,Ij) + 0.4*(ETA(:,Ij+1)-2*ETA(:,Ij)+ETA(:,Ij-1));
28 end
29 ETA_L = ETA; ETA_PL = ETA;
30 % Boundary condition
31 if ps==1, VX = -D_B*X_vx; VY = D_B*Y_vy;
32 elseif ps==0, VX = D_B*Y_vx; VY = 0*Y_vy; end
33 % Parameters for pseudo-transient iterations
34 tol = 5e-6; error = 10*tol;
35 dpt_P = 50 *min(dx,dy)^2/max(max(ETA)); % Pseudo time step pressure
36 dpt_V = 0.05*min(dx,dy)^2/max(max(ETA)); % Pseudo time step velocity
37 PolD = P; iter = 0;
38 while error>tol; iter = iter+1; % START of iteration loop
39 DXX = diff(VX,1,1)/dx; % Eq (1)
40 DYY = diff(VY,1,2)/dy; % Eq (2)
41 DXY = 1/2*(diff(VX(2:end-1,:),1,2)/dy ...
42 + diff(VY(:,2:end-1),1,1)/dx); % Eq (3)
43 TXX = 2.*ETA.*DXX; % Eq (4)
44 TYY = 2.*ETA.*DYY; % Eq (5)
45 TXY = 2.*n2c(ETA).*DXY; % Eq (6)
46 SXX = -P(:,2:end-1)+TXX(:,2:end-1); % Eq (7)
47 SYX = -P(2:end-1,:)+TYY(2:end-1,:); % Eq (8)
48 RES_P = -(diff(VX,1,1)/dx + diff(VY,1,2)/dy); % Eq (16)
49 RES_VX = diff(SXX,1,1)/dx + diff(TXY,1,2)/dy; % Eq (17)
50 RES_VY = diff(SYX,1,2)/dy + diff(TXY,1,1)/dx; % Eq (18)
51 P = P + dpt_P*RES_P; % Eq (22)
52 VX(2:end-1,2:end-1) = VX(2:end-1,2:end-1) + dpt_V*RES_VX; % Eq (23)
53 VY(2:end-1,2:end-1) = VY(2:end-1,2:end-1) + dpt_V*RES_VY; % Eq (24)
54 TII = sqrt(0.5*(TXX.^2 + TYY.^2 + 2*c2n(TXY).^2)); % Eq (13)
55 if n_exp>1 % Power-law viscosity
56 ETA_PL_it = ETA_PL; % Viscosity of previous iteration step
57 ETA_PL = ETA.*(TII/s_ref)^(1-n_exp); % Eq (12)
58 ETA_PL = exp(log(ETA_PL)+0.5*log(ETA_PL_it)+0.5);
59 ETA = 1./(1./ETA_L + 1./ETA_PL); % Eq (14)
60 ETA(inside) = ETA_L(inside); % Power-law viscosity only applied to matrix
61 ETA(ETA<etamin) = etamin; % Minimum viscosity
62 end
63 error = max([max(abs(dpt_V*RES_VX(:))), max(abs(dpt_V*RES_VY(:))), max(abs(dpt_P*RES_P(:)))]);
64 if mod(iter,2000)==0 % Visualization during calculation
65 subplot(221),pcolor(X,Y,P/(2*eta_B*D_B)),shading interp,axis equal,axis tight,colorbar,colormap(jet)
66 title({'A') Pressure [ ], P / (2 \eta_B D_B)'); ['Iteration: ',num2str(iter),'; error: ',num2str(error)]},ylabel('Height [ ]')
67 subplot(222),pcolor(X,Y,log10(ETA/eta_B)),shading interp, colormap(jet), hold on
68 title({'B') Viscosity [ ], log_{10}(\eta / \eta_B), 'and velocity arrows')
69 st = 15; VXC = (VX(1:end-1,:)+VX(2:end,:))/2; VYC = (VY(:,1:end-1)+VY(:,2:end))/2;
70 quiver(X(1:st:end,1:st:end),Y(1:st:end,1:st:end),VXC(1:st:end,1:st:end),VYC(1:st:end,1:st:end),'w');
71 axis equal,axis tight,colorbar,axis([-Lx/2,Lx/2,-Ly/2,Ly/2]),hold off
72 subplot(223),pcolor(X,Y,TII/(2*eta_B*D_B)),shading interp,axis equal,axis tight,colorbar,colormap(jet)
73 title({'C') 2nd stress invariant [ ], T_{II} / (2 \eta_B D_B)'); xlabel('Width [ ]'), ylabel('Height [ ]')
74 subplot(224),pcolor(X,Y,DXX/(2*eta_B*D_B)),shading interp,axis equal,axis tight,colorbar,colormap(jet)
75 title({'D') Horizontal strain rate [ ], D_{xx}/D_B'); xlabel('Width [ ]')
76 set(gcf,'position',[281 47.4000 975.2000 724]), drawnow
77 end
78 end % END of iteration loop
79 % Additional functions performing interpolations on the numerical grid
80 function A1 = n2c(A0) % Interpolation of nodal points to center points
81 A1 = (A0(2:end,:) + A0(1:end-1,:))/2;
82 A1 = (A1(:,2:end) + A1(:,1:end-1))/2;
83 end
84 function A2 = c2n(A0) % Interpolation of center points to nodal points
85 A1 = zeros(size(A0,1)+1,size(A0,2));
86 A1(:,1) = [1.5*A0(1,:)-0.5*A0(2,:); (A0(2:end,:)+A0(1:end-1,:))/2; 1.5*A0(end,:)-0.5*A0(end-1,:)];
87 A2 = zeros(size(A1,1),size(A1,2)+1);
88 A2(:,1) = [1.5*A1(:,1)-0.5*A1(:,2); (A1(:,2:end)+A1(:,1:end-1))/2; 1.5*A1(:,end)-0.5*A1(:,end-1)];
89 end

```

**Fig. 12.** The complete Matlab code. The definition of numerical and physical parameters is followed by an initialization of initial and boundary conditions. All instantaneous quantities are calculated iteratively within the iterative loop until the error becomes smaller than the tolerance. Selected quantities are displayed every 2000th iteration, i.e. the pressure, the effective viscosity, the velocity, the 2nd stress invariant, and the horizontal normal strain rate fields.

```

1 % Iterative finite difference program for 2D viscous deformation, GNU Octave version
2 % This code is free software under the Creative Commons CC-BY-NC-ND license
3 % Published in W.R.Halter, E.Macherel, and S.M.Schmalholz (2022) JSG, https://doi.org/10.1016/j.jsg.2022.104617
4 clear all, close all, clc
5 % Additional functions performing interpolations on the numerical grid
6 function A1 = n2c(A0) % Interpolation of nodal points to center points
7     A1 = (A0(2:end,:) + A0(1:end-1,:))/2;
8     A1 = (A1(:,2:end) + A1(:,1:end-1))/2;
9 end
10 function A2 = c2n(A0) % Interpolation of center points to nodal points
11     A1 = zeros(size(A0,1)+1,size(A0,2));
12     A1(:, :) = [1.5*A0(1,:)-0.5*A0(2,:); (A0(2:end,:) + A0(1:end-1,:))/2; 1.5*A0(end,:)-0.5*A0(end-1,:)];
13     A2 = zeros(size(A1,1),size(A1,2)+1);
14     A2(:, :) = [1.5*A1(:,1)-0.5*A1(:,2), (A1(:,2:end)+A1(:,1:end-1))/2, 1.5*A1(:,end)-0.5*A1(:,end-1)];
15 end
16 % Beginning of the actual Script
17 ps = 1; % ps = 1 models pure shear; ps = 0 models simple shear
18 etaRatio = 50; % Viscosity ratio (matrix/inclusion)
19 a = 0.2; b = a/2; phi = 30/180*pi; % Parameters defining elliptical inclusion
20 % Numerical parameters
21 nx = 151; ny = nx; % Numerical resolution
22 Lx = 1; Ly = Lx; % Model dimension
23 dx = Lx/(nx-1); dy = Ly/(ny-1); % Grid spacing
24 x = [-Lx/2:dx:Lx/2]; y = [-Ly/2:dy:Ly/2]; % Coordinate vectors
25 x_vx = [x(1)-dx/2, (x(1:end-1) + x(2:end))/2, x(end)+dx/2]; % Horizontal vector for Vx which is one more than basic grid
26 y_vy = [y(1)-dy/2, (y(1:end-1) + y(2:end))/2, y(end)+dy/2]; % Vertical vector for Vy which is one more than basic grid
27 % 3 numerical grids due to staggered grid
28 [X,Y] = ndgrid(x,y); [X_vx,Y_vx] = ndgrid(x_vx,y); [X_vy,Y_vy] = ndgrid(x,y_vy);
29 % Physical parameters
30 eta_B = 1; n_exp = 3;
31 s_ref = 1.5; D_B = 1;
32 % Initialization
33 P = zeros(nx, ny); ETA = eta_B*ones(nx, ny);
34 % Define elliptical inclusion
35 inside = ((X*cos(phi)+Y*sin(phi)).^2/a^2 + (X*sin(phi)-Y*cos(phi)).^2/b^2 < 1);
36 ETA(inside) = eta_B/etaRatio; etamin = 0.1*min(ETA(:));
37 for smo=1:2; Ii = [2:nx-1]; Ij = [2:ny-1]; % Smoothing of the initial viscosity field
38     ETA(Ii,:) = ETA(Ii,:) + 0.4*(ETA(Ii+1,:)-2*ETA(Ii,:)+ETA(Ii-1,:));
39     ETA(:,Ij) = ETA(:,Ij) + 0.4*(ETA(:,Ij+1)-2*ETA(:,Ij)+ETA(:,Ij-1));
40 end
41 ETA_L = ETA; ETA_PL = ETA;
42 % Boundary condition
43 if ps==1, VX = -D_B*X_vx; VY = D_B*Y_vy;
44 else if ps==0, VX = D_B*Y_vx; VY = 0*Y_vy; end
45 % Parameters for pseudo-transient iterations
46 tol = 5e-6; error = 10*tol;
47 dpt_P = 50 *min(dx,dy)^2/max(max(ETA)); % Pseudo time step pressure
48 dpt_V = 0.05*min(dx,dy)^2/max(max(ETA)); % Pseudo time step velocity
49 PolD = P; iter = 0;
50 while error>tol; iter = iter+1; % START of iteration loop
51     DXX = diff(VX,1,1)/dx; % Eq (1)
52     DYY = diff(VY,1,2)/dy; % Eq (2)
53     DXY = 1/2*(diff(VX(2:end-1,:),1,2)/dy ...
54         + diff(VY(:,2:end-1),1,1)/dx); % Eq (3)
55     TXX = 2.*ETA.*DXX; % Eq (4)
56     TYY = 2.*ETA.*DYY; % Eq (5)
57     TXY = 2.*n2c(ETA).*DXY; % Eq (6)
58     SXX = -P(:,2:end-1)+TXX(:,2:end-1); % Eq (7)
59     SYX = -P(2:end-1,:)+TXY(2:end-1,:); % Eq (8)
60     RES_P = -(diff(VX,1,1)/dx + diff(VY,1,2)/dy); % Eq (16)
61     RES_VX = diff(SXX,1,1)/dx + diff(TXY,1,2)/dy; % Eq (17)
62     RES_VY = diff(SYX,1,2)/dy + diff(TXX,1,1)/dx; % Eq (18)
63     P = P + dpt_P*RES_P; % Eq (22)
64     VX(2:end-1,2:end-1) = VX(2:end-1,2:end-1) + dpt_V*RES_VX; % Eq (23)
65     VY(2:end-1,2:end-1) = VY(2:end-1,2:end-1) + dpt_V*RES_VY; % Eq (24)
66     TII = sqrt(0.5*(TXX.^2 + TYY.^2 + 2*c2n(TXY).^2)); % Eq (13)
67     if n_exp>1 % Power-law viscosity
68         ETA_PL_it = ETA_PL; % Viscosity of previous iteration step
69         ETA_PL = ETA.*(TII/s_ref).^(1-n_exp); % Eq (12)
70         ETA_PL = exp(log(ETA_PL)+0.5*log(ETA_PL_it)*0.5);
71         ETA = 1./(1./ETA_L + 1./ETA_PL); % Eq (14)
72         ETA(inside) = ETA_L(inside); % Power-law viscosity only applied to matrix
73         ETA(ETA-etamin) = etamin; % Minimum viscosity
74     end
75     error = max([max(abs(dpt_V*RES_VX(:))), max(abs(dpt_V*RES_VY(:))), max(abs(dpt_P*RES_P(:)))]);
76     if mod(iter,2000)==0 % Visualization during calculation
77         subplot(221), pcolor(X,Y,P/(2*eta_B*D_B)), shading interp, axis equal, axis tight, colorbar, colormap(jet)
78         title({'A') Pressure [ ], P / (2 \eta_B D_B)}; ['Iteration: ', num2str(iter), '; error: ', num2str(error)]), ylabel('Height [ ]')
79         subplot(222), pcolor(X,Y,log10(ETA/eta_B)), shading interp, colormap(jet), hold on
80         title({'B') Viscosity [ ], log_{10}(\eta / \eta_B)}; and velocity arrows')
81         st = 15; VXC = (VX(1:end-1,:)+VX(2:end,:))/2; VYC = (VY(:,1:end-1)+VY(:,2:end))/2;
82         quiver(X(1:st:end,1:st:end), Y(1:st:end,1:st:end), VXC(1:st:end,1:st:end), VYC(1:st:end,1:st:end), 'w');
83         axis equal, axis tight, colorbar, axis([-Lx/2,Lx/2,-Ly/2,Ly/2]), hold off
84         subplot(223), pcolor(X,Y,TII/(2*eta_B*D_B)), shading interp, axis equal, axis tight, colorbar, colormap(jet)
85         title({'C') 2nd stress invariant [ ], T_{II} / (2 \eta_B D_B)}; xlabel('Width [ ]'), ylabel('Height [ ]')
86         subplot(224), pcolor(X,Y,DXX/(2*eta_B*D_B)), shading interp, axis equal, axis tight, colorbar, colormap(jet)
87         title({'D') Horizontal strain rate [ ], D_{xx}/D_B)}; xlabel('Width [ ]')
88         set(gcf,'position',[281 47.4000 975.2000 724]), drawnow
89     end
90 end % END of iteration loop

```

Fig. 13. The complete GNU Octave code. The Octave version is identical to the Matlab version of the code with the only exception, that the two interpolation functions (lines 79–89 in the Matlab version) are moved to the beginning of the code (lines 5–15 in the Octave version).

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

We thank Y. Podladchikov, L. Räss and T. Duretz for very helpful and stimulating discussions. We thank N. Mancktelow and G. Pennacchioni for providing some of the photos presented in Fig. 1. E. Macherel and W. R. Halter are supported by SNSF grant No. 200020197218. This work was supported by the University of Lausanne.

## Appendix A. Supplementary data

The MATLAB (Fig. 12) and the GNU Octave program (Fig. 13) are provided as supplementary data to this article and can be found online at <https://doi.org/10.1016/j.jsg.2022.104617> and on GitHub at [http://github.com/halterw/A\\_simple\\_computer\\_program](http://github.com/halterw/A_simple_computer_program).

## References

- Allmendinger, R., 2011. *Structural Geology Algorithms: Vectors and Tensors*, first ed. ed. Cambridge University Press.
- Anderson, D.A., Tannehill, J.C., Pletcher, R.H., Ramakanth, M., Shankar, V., 2020. *Computational Fluid Mechanics and Heat Transfer*. CRC Press.
- Angel, R.J., Nimis, P., Mazzucchelli, M.L., Alvaro, M., Nestola, F., 2015. How large are departures from lithostatic pressure? constraints from host-inclusion elasticity. *J. Metamorph. Geol.* 33, 801–813. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/jmg.12138>, doi:<https://doi.org/10.1111/jmg.12138>, arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/jmg.12138>.
- Bose, N., Mukherjee, S., 2020. Estimation of deformation temperatures, flow stresses and strain rates from an intra-continental shear zone: the main boundary thrust, nw himalaya (uttarakhand, India). *Mar. Petrol. Geol.* 112, 104094. URL: <https://www.sciencedirect.com/science/article/pii/S0264817219305306>, doi:<https://doi.org/10.1016/j.marpetgeo.2019.104094>.
- Chernak, L.J., Hirth, G., 2010. Deformation of antigorite serpentinite at high temperature and pressure. *Earth Planet Sci. Lett.* 296, 23–33. URL: <https://www.sciencedirect.com/science/article/pii/S0012821X10002815>, doi:<https://doi.org/10.1016/j.epsl.2010.04.035>.
- Dabrowski, M., Schmid, D., Podladchikov, Y., 2012. A two-phase composite in simple shear: effective mechanical anisotropy development and localization potential. *J. Geophys. Res. Solid Earth* 117.
- Duretz, T., Räss, L., Podladchikov, Y., Schmalholz, S., 2019. Resolving thermomechanical coupling in two and three dimensions: spontaneous strain localization owing to shear heating. *Geophys. J. Int.* 216, 365–379.
- Eshelby, J.D., Peierls, R.E., 1957. The determination of the elastic field of an ellipsoidal inclusion, and related problems. *Proc. Roy. Soc. Lond. Math. Phys. Sci.* 241, 376–396. URL: <https://doi.org/10.1098/rspa.1957.0133>, URL: <https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1957.0133>, arXiv:<https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.1957.0133>.
- Evans, O., Spiegelman, M., Kelemen, P.B., 2020. Phase-field modeling of reaction-driven cracking: determining conditions for extensive olivine serpentinization. *J. Geophys. Res. Solid Earth* 125, e2019JB018614.
- Fagereng, A., Biggs, J., 2019. New perspectives on 'geological strain rates' calculated from both naturally deformed and actively deforming rocks. *J. Struct. Geol.* 125, 100–110. URL: <https://www.sciencedirect.com/science/article/pii/S0191814118300543>, doi:<https://doi.org/10.1016/j.jsg.2018.10.004>, back to the future.
- Fletcher, R.C., 1974. Wavelength selection in the folding of a single layer with power-law rheology. *Am. J. Sci.* 274, 1029–1043.
- Fletcher, R.C., Pollard, D.D., 1999. Can we understand structural and tectonic processes and their products without appeal to a complete mechanics? *J. Struct. Geol.* 21, 1071–1088. URL: <https://www.sciencedirect.com/science/article/pii/S0191814199000565>, doi:[https://doi.org/10.1016/S0191-8141\(99\)00056-5](https://doi.org/10.1016/S0191-8141(99)00056-5).
- Frankel, S.P., 1950. Convergence rates of iterative treatments of partial differential equations. *Math. Comput.* 4, 65–75.
- Gerya, T., 2019. *Introduction to Numerical Geodynamic Modelling*. Cambridge University Press.
- Hardy, S., Finch, E., 2005. Discrete-element modelling of detachment folding. *Basin Res.* 17, 507–520. URL: <https://www.earthdoc.org/content/journals/10.1111/j.1365-2117.2005.00280.x>, doi:<https://doi.org/10.1111/j.1365-2117.2005.00280.x>.
- Hughes, T.J., 2012. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications.
- Jaeger, J.C., Cook, N.G.W., Zimmerman, R., 2007. *Fundamentals of Rock Mechanics*, fourth ed. Wiley-Blackwell.
- Jeffery, G.B., Filon, L.N.G., 1922. The motion of ellipsoidal particles immersed in a viscous fluid. In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, vol. 102, pp. 161–179. <https://doi.org/10.1098/rspa.1922.0078>. URL: <https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1922.0078>, arXiv:<https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.1922.0078>.
- Jessell, M.W., Bons, P.D., Griera, A., Evans, L.A., Wilson, C.J., 2009. A tale of two viscosities. *J. Struct. Geol.* 31, 719–736.
- Karato, S.I., 2008. *Deformation of Earth Materials: an Introduction to the Rheology of Solid Earth*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511804892>.
- Kolditz, O., Shao, H., Wang, W., Bauer, S. (Eds.), 2015. *Thermo-Hydro-Mechanical-Chemical Processes in Fractured Porous Media: Modelling and Benchmarking*, first ed. ed. Springer. <https://doi.org/10.1007/978-3-319-11894-9>.
- Liu, L., Hasterok, D., 2016. High-resolution lithosphere viscosity and dynamics revealed by magnetotelluric imaging. *Science* 353, 1515–1519. URL: <https://doi.org/10.1126/science.aaf6542>. URL: <https://www.science.org/doi/abs/10.1126/science.aaf6542>, arXiv:<https://www.science.org/doi/pdf/10.1126/science.aaf6542>.
- Luisier, C., Baumgartner, L., Schmalholz, S.M., Siron, G., Venemann, T., 2019. Metamorphic pressure variation in a coherent alpine nappe challenges lithostatic pressure paradigm. *Nat. Commun.* 10, 1–11.
- Mancktelow, N.S., 2008. Tectonic pressure: theoretical concepts and modelled examples. *Lithos* 103, 149–177.
- Mancktelow, N.S., Pennacchioni, G., 2010. Why calcite can be stronger than quartz. *J. Geophys. Res. Solid Earth* 115.
- Marques, F.O., Mandal, N., Taborda, R., Antunes, J.V., Bose, S., 2014. The behaviour of deformable and non-deformable inclusions in viscous flow. *Earth Sci. Rev.* 134, 16–69. URL: <https://doi.org/10.1016/j.earscirev.2014.03.007>. URL: <https://www.sciencedirect.com/science/article/pii/S0012825214000555>.
- Mase, G.E., 1970. *Continuum Mechanics*, vol. 970. McGraw-Hill, New York.
- Matter, J.M., Kelemen, P.B., 2009. Permanent storage of carbon dioxide in geological reservoirs by mineral carbonation. *Nat. Geosci.* 2, 837–841.
- Moulas, E., Burg, J.P., Podladchikov, Y., 2014. Stress field associated with elliptical inclusions in a deforming matrix: mathematical model and implications for tectonic overpressure in the lithosphere. *Tectonophysics* 631, 37–49.
- Moulas, E., Schmalholz, S.M., Podladchikov, Y., Tajčmanová, L., Kostopoulos, D., Baumgartner, L., 2019. Relation between mean stress, thermodynamic, and lithostatic pressure. *J. Metamorph. Geol.* 37, 1–14.
- Mukherjee, S., 2019. *Teaching Structural Geology in Indian Context*. Springer Singapore, Singapore, pp. 221–232. URL: [https://doi.org/10.1007/978-981-13-2781-0\\_8](https://doi.org/10.1007/978-981-13-2781-0_8), doi:10.1007/978-981-13-2781-0\_8.
- Pandey, S., Vishal, V., Chaudhuri, A., 2018. Geothermal reservoir modeling in a coupled thermo-hydro-mechanical-chemical approach: a review. *Earth Sci. Rev.* 185, 1157–1169. URL: <https://www.sciencedirect.com/science/article/pii/S0012825218303301>, doi:<https://doi.org/10.1016/j.earscirev.2018.09.004>.
- Petri, B., Duretz, T., Mohn, G., Schmalholz, S.M., Karner, G.D., Müntener, O., 2019. Thinning mechanisms of heterogeneous continental lithosphere. *Earth Planet Sci. Lett.* 512, 147–162.
- Pfiffner, O.A., Ramsay, J.G., 1982. Constraints on geological strain rates: arguments from finite strain states of naturally deformed rocks. *J. Geophys. Res. Solid Earth* 87, 311–321. URL: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/JB087iB01p00311>, doi:<https://doi.org/10.1029/JB087iB01p00311>, arXiv:<https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/JB087iB01p00311>.
- Pollard, D., Fletcher, R.C., 2005. *Fundamentals of Structural Geology*. Cambridge University Press.
- Poulet, T., Karrech, A., Regenauer-Lieb, K., Fisher, L., Schaub, P., 2012. Thermal-hydraulic-mechanical coupling with damage mechanics using escript and abaqus. *Tectonophysics* 526, 124–132.
- Ragan, D.M., 2009. *Structural Geology, an Introduction to Geometrical Techniques*. Cambridge University Press, Cambridge.
- Ramsay, J.G., 1967. *Folding and Fracturing of Rocks*, vol. 568. Mc Graw Hill Book Company.
- Ramsay, J.G., Huber, M.I., 1983. *The Techniques of Modern Structural Geology: Folds and Fractures*, vol. 1. Academic press.
- Ramsay, J.G., Huber, M.I., 1987. *Modern Structural Geology*, vol. 2. Academic Press, London.
- Ramsay, J.G., Lisle, R.J., 2000. *Applications of Continuum Mechanics in Structural Geology (Techniques of Modern Structural Geology, vol. 3)*. Academic Press vol. 3.
- Räss, L., Duretz, T., Podladchikov, Y., 2019. Resolving hydromechanical coupling in two and three dimensions: spontaneous channelling of porous fluids owing to decompression weakening. *Geophys. J. Int.* 218, 1591–1616.
- Räss, L., Licul, A., Herman, F., Podladchikov, Y.Y., Suckale, J., 2020. Modelling thermomechanical ice deformation using an implicit pseudo-transient method (fastice v1. 0) based on graphical processing units (gpus). *Geosci. Model Dev. (GMD)* 13, 955–976.
- Richardson, L.F., 1911. IX. the approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philos. Trans. R. Soc. Lond. - Ser. A Contain. Pap. a Math. or Phys. Character* 210, 307–357.
- Saki, A., Miri, M., Oberhänsli, R., 2020. High temperature-low pressure metamorphism during subduction of neo-tethys beneath the Iranian plate: evidence for mafic migmatite formation in the alvand complex (western Iran). *Mineral. Petrol.* 114, 539–557.
- Schmalholz, S., Podladchikov, Y., 1999. Buckling versus folding: importance of viscoelasticity. *Geophys. Res. Lett.* 26, 2641–2644.
- Schmalholz, S., Podladchikov, Y., Schmid, D., 2001. A spectral/finite difference method for simulating large deformations of heterogeneous, viscoelastic materials. *Geophys. J. Int.* 145, 199–208. URL: <https://doi.org/10.1046/j.0956-540x.2000.01371.x>.

- Schmalholz, S.M., Mancktelow, N.S., 2016. Folding and necking across the scales: a review of theoretical and experimental results and their applications. *Solid Earth* 7, 1417–1465.
- Schmalholz, S.M., Moulas, E., Plümper, O., Myasnikov, A.V., Podladchikov, Y.Y., 2020. 2d hydro-mechanical-chemical modeling of (de) hydration reactions in deforming heterogeneous rock: the periclase-brucite model reaction. *G-cubed* 21, e2020GC009351.
- Schmalholz, S.M., Podladchikov, Y.Y., 2013. Tectonic overpressure in weak crustal-scale shear zones and implications for the exhumation of high-pressure rocks. *Geophys. Res. Lett.* 40, 1984–1988.
- Schmalholz, S.M., Schmid, D.W., 2012. Folding in power-law viscous multi-layers. *Phil. Trans. Math. Phys. Eng. Sci.* 370, 1798–1826.
- Schmid, D.W., Podladchikov, Y.Y., 2003. Analytical solutions for deformable elliptical inclusions in general shear. *Geophys. J. Int.* 155, 269–288.
- Schöpfer, M.P., Abe, S., Childs, C., Walsh, J.J., 2009. The impact of porosity and crack density on the elasticity, strength and friction of cohesive granular materials: insights from dem modelling. *Int. J. Rock Mech. Min. Sci.* 46, 250–261. URL: <https://www.sciencedirect.com/science/article/pii/S1365160908000695>. <https://doi.org/10.1016/j.jrmms.2008.03.009>.
- Selkman, S., 1978. Stress and displacement analysis of boudinages by the finite-element method. *Tectonophysics* 44, 115–139.
- Strömgård, K.E., 1973. Stress distribution during formation of boudinage and pressure shadows. *Tectonophysics* 16, 215–248.
- Timoshenko, S., Goodier, J., 1970. *Theory of Elasticity*, third ed. ed. McGraw-Hill Publishing company.
- Turcotte, D.L., Schubert, G., 2014. *Geodynamics*, third ed. Cambridge university press.
- Vaughan-Hammon, J.D., Luisier, C., Baumgartner, L.P., Schmalholz, S.M., 2021. Alpine peak pressure and tectono-metamorphic history of the monte rosa nappe: evidence from the cirque du véraz, upper ayas valley, Italy. *Swiss J. Geosci.* 114, 1–26.
- Wang, L.H., Yarushina, V.M., Alkhimenkov, Y., Podladchikov, Y., 2021. Physics-inspired pseudo-transient method and its application in modelling focused fluid flow with geological complexity. *Geophys. J. Int.* 229, 1–20. <https://doi.org/10.1093/gji/ggab426>. URL: <https://doi.org/10.1093/gji/ggab426>, arXiv:<https://academic.oup.com/gji/article-pdf/229/1/1/41781930/ggab426.pdf>.
- Wheeler, J., 2018. The effects of stress on reactions in the earth: sometimes rather mean, usually normal, always important. *J. Metamorph. Geol.* 36, 439–461.
- Yamato, P., Duret, T., May, D.A., Tartese, R., 2015. Quantifying magma segregation in dykes. *Tectonophysics* 660, 132–147.
- Zienkiewicz, O., Taylor, R., Fox, D. (Eds.), 2014. *The Finite Element Method for Solid and Structural Mechanics*, seventh ed. Butterworth-Heinemann, Oxford. URL: <https://www.sciencedirect.com/science/article/pii/B978185617634700017X>. doi: <https://doi.org/10.1016/B978-1-85617-634-7.00017-X>.

**Update**

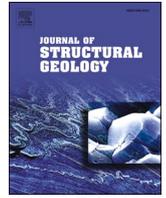
**Journal of Structural Geology**

Volume 161, Issue , August 2022, Page

DOI: <https://doi.org/10.1016/j.jsg.2022.104665>

Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

## Journal of Structural Geology

journal homepage: [www.elsevier.com/locate/jsg](http://www.elsevier.com/locate/jsg)

## Corrigendum to “A simple computer program for calculating stress and strain rate in 2D viscous inclusion-matrix systems” [J. Struct. Geol. 160 (2022) 104617]

William R. Halter<sup>\*</sup>, Emilie Macherel, Stefan M. Schmalholz

*Institute of Earth Sciences, University of Lausanne, Switzerland*

The authors found an error in the computer code presented in Halter et al., 2022 (termed H22 in the following). The error affects only the calculation of the effective power-law viscosity, and all results for a linear viscosity are correct. Furthermore, the incorrect results for a power-law viscosity, presented in figure 8B and 8D in H22, are very similar to the results calculated with the correct code (Fig. 1 of this corrigendum).

In the Matlab code (Fig. 7, code line 3, and Fig. 12, code line 57, in H22) and in the Octave code (Fig. 13, code line 69, in H22) the variable **ETA** in the published code line

```
ETA_PL = ETA.*(TII/s_ref)^(1-n_exp);
```

must be changed to the variable **ETA\_L** to obtain the correct code line

```
ETA_PL = ETA_L.*(TII/s_ref)^(1-n_exp);
```

The corrected Matlab and Octave codes are available online under [https://github.com/halterw/A\\_simple\\_computer\\_program](https://github.com/halterw/A_simple_computer_program).

DOI of original article: <https://doi.org/10.1016/j.jsg.2022.104617>.

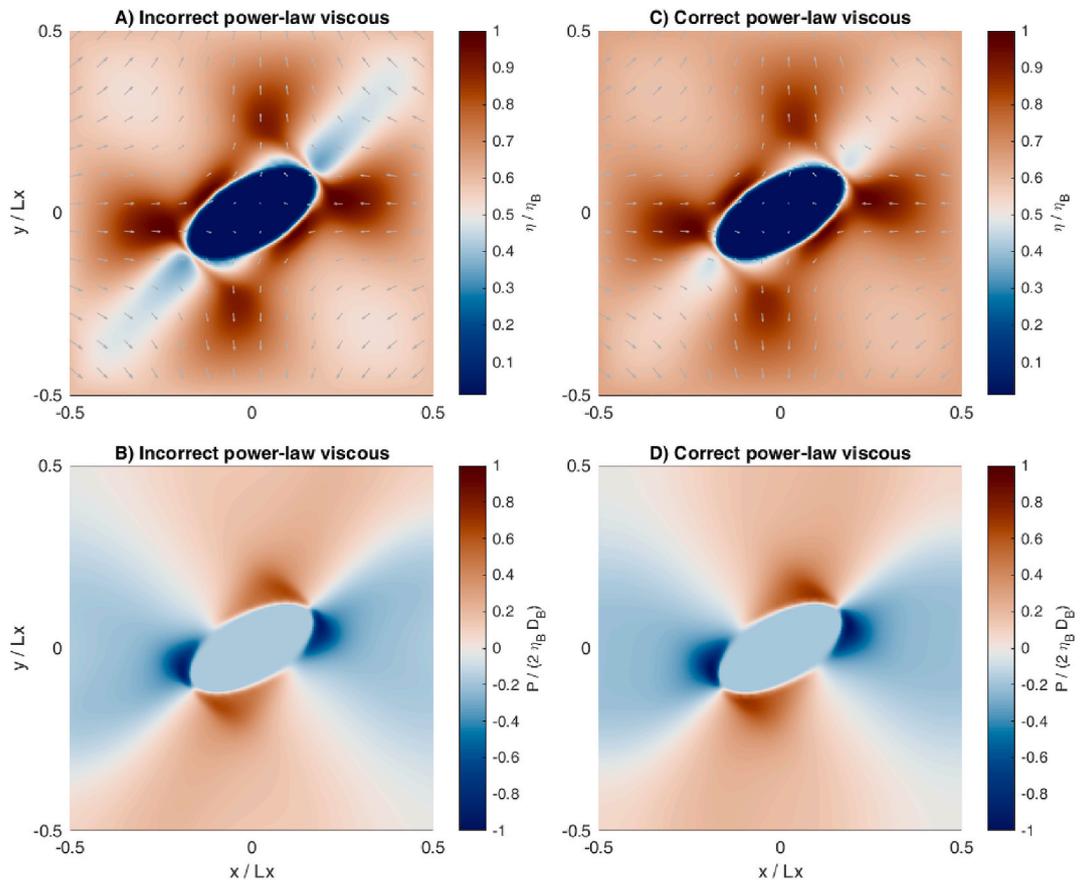
<sup>\*</sup> Corresponding author.

*E-mail address:* [william.halter@unil.ch](mailto:william.halter@unil.ch) (W.R. Halter).

<https://doi.org/10.1016/j.jsg.2022.104665>

Available online 25 June 2022

0191-8141/© 2022 The Author(s). Published by Elsevier Ltd. All rights reserved.



**Fig. 1.** A) and B): Incorrect results for the inclusion-matrix system with power-law viscosity, published in H22. C) and D): Correct results for the inclusion-matrix system with power-law viscosity. The general patterns of the effective viscosity (A and C) and pressure (B and D) are the same, but the absolute values of the effective viscosity and pressure are slightly different.

The authors apologise for any inconveniences caused.