1 Introduction

This document includes a description of the code FALCON, with a presentation of the features implemented followed by results of extensive benchmarks listed in Table 1, implemented to verify that the code solves correctly a large range of problems. Description and results of all benchmarks are shown in Section 10.

Feature	**Benchmark**	**Section**
Solver	Stokes flow	10.2
Markers advection	Zalesak disk	10.3
	Conservative Interpolation Velocity	10.4
Momentum equation	Poiseuille flow	10.5
	Instantaneous 2D sphere	10.6
	Rayleigh-Taylor experiment	10.7
	Falling block	10.8
Sticky air and free surface	2D Stokes sphere	10.9
	Stabilisation algorithm	10.10
	Topography relaxation	10.11
	Spontaneous subduction	10.12
Erosion and sedimentation	-	-
Non-linear visco-plasticity	Slab detachment	10.13
	Indenter	10.14
	Brick	10.15
Energy equation	Advection stabilisation	10.16
	Simple shear heating	10.17
	Shear and adiabatic heating	10.18
Energy + momentum	Mantle convection	??
	Viscoplastic mantle convection	??
	Thin layer entrainment	??
Phase changes and hydration	Hydrated sinking cylinder	??
Melting	Experimental melting curves	??

Table 1: List of benchmarks implemented for each feature of the code. Data of all benchmarks can be found at https://github.com/aleregorda/Benchmarks

The code is written in Fortran90 and uses the finite element method (FEM) with quadrilateral $Q_1 \times P_0$ elements (continuous bilinear velocity and discontinuous constant pressure), associated with MUMPS[1] (**??**) that is a software package for solving systems of linear equations of the form $A \cdot x = b$, where A is a square sparse matrix, by means of a direct method. Since $Q_1 \times P_0$ elements do not satisfy Ladyzhenskaya, Babuska and Brezzi (LBB) stability condition (**?**), elemental pressure is elaborated in post-processing to avoid spurious pressures by means of a double interpolation to smooth the pressure. Following this procedure, the elemental pressure is interpolated onto nodes and then back onto elements. After the smoothing procedure, the pressure field is calculated again on the nodes to be used in combination with the lithostatic pressure, which is calculated onto nodes. Correctness of the solution and performances of the solver in terms of time and memory usage are tested solving the Stokes flow with the analytical solution proposed by **?** (Section 10.2).

The thermo-mechanics of crust-mantle systems is described by means of the conservation of mass, momentum and energy equations, expressed as follows:

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}) = 0 \tag{1}$$

$$-\boldsymbol{\nabla} p + \boldsymbol{\nabla} \cdot \boldsymbol{\tau} + \rho \boldsymbol{g} = 0 \tag{2}$$

$$\rho C_p \left(\frac{\partial T}{\partial t} + \boldsymbol{u} \cdot \boldsymbol{\nabla} T \right) = \boldsymbol{\nabla} \cdot (k \boldsymbol{\nabla} T) + H_{tot} \tag{3}$$

where ρ is the density, $\boldsymbol{u}$ is the velocity, p is the pressure, $\boldsymbol{\tau}$ is the deviatoric stress, $\boldsymbol{g}$ is the gravity acceleration, C_p is the specific heat at constant pressure, T is the temperature, k is the thermal conductivity and H_{tot} is the total internal heat production. Density variations due to temperature are generally small enough to assume the density as constant ($\rho = \rho_0$) in Eq. 1 and in Eq. 3, while it must be treated as a variable in the buoyancy term of Eq. 2, such

[1] http://mumps.enseeiht.fr/

that

$$\rho = \rho_0(1 - \alpha(T - T_0))$$

where ρ_0 is the density at a reference temperature T_0 and α is the coefficient of thermal expansion. Eq. 1 then can be rewritten as

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0 \tag{4}$$

This simplification is known as the extended Boussinesq approximation.

The time step dt is chosen in according to Courant-Friedrichs-Lewy condition (**?**), such that

$$dt = C_n \ \min\left(\frac{\Delta x}{\max|\boldsymbol{u}|}, \frac{(\Delta x)^2}{\kappa}\right)$$

where $0 < C_n < 1$ is the Courant (CFL) number, Δx is the minimum dimension of smallest element, $\max|\boldsymbol{u}|$ is the maximum velocity calculated on the entire domain and $\kappa = \frac{k}{\rho C_p}$ is the thermal diffusivity (**?**).

2 Lagrangian markers

Elemental properties (density, viscosity, thermal conductivity, specific heat and thermal expansion) needed to solve Eqs. 4, 2 and 3 are related to the composition of each element, determined by means of Lagrangian markers that are characteristic of different materials in the domain. Elemental properties (with the sole exception of the viscosity) are calculated using an arithmetic mean, as

$$P_e = \frac{1}{n}\sum_{i=1}^{n} P_i$$

where P_e is the elemental property, P_i is the property characteristic of the material of each marker and n is the number of markers of the element. Differently, the average scheme for the elemental viscosity can be chosen between harmonic, geometric and arithmetic mean. Markers can be placed regularly or randomly at the beginning of the simulation and their advection is performed by either a 2nd-order or a 4th-order Runge-Kutta in space, interpolating the velocity field on each marker by means of the shape functions (see Section 10.3). The interpolated velocity is then corrected by means of the Conservative Velocity Interpolation (CVI), which introduces a corrective term to reduce dispersion and clustering of particles in both steady state and time-dependent (**?**) (see Section 10.4).

The initial distribution of the markers is created using the open-source code library Geodynamic World Builder (**?**). During the simulation, each marker carries memory of temperature, pressure and accumulated strain, which are determined interpolating the relative nodal parameter. In particular, pressure is interpolated onto each marker after the smoothing procedure. The number of markers contained in each element is maintained between n_{min} and n_{max}. When in an element there are less markers than n_{min} the code adds random markers to reach the n_{min}, while if the number is higher than n_{max} some of them are randomly deleted. When new markers are added, they assume the properties of the nearest marker. In this way elements are never empty and maintain a number of markers inside a prefixed range.

3 The momentum equation

The penalty method is implemented so that Eq. 4 can be written as:

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} + \frac{p}{\lambda} = 0 \tag{5}$$

where λ is the so-called penalty parameter, which should be 6-7 orders of magnitude larger than the shear viscosity to ensure that mass conservation is satisfied (**??**).

The deviatoric stress tensor in Eq. 2 can be written in terms of the strain rate tensor as $\boldsymbol{\tau} = 2\eta\dot{\boldsymbol{\epsilon}}$, with $\dot{\boldsymbol{\epsilon}} = \frac{1}{2}\left(\boldsymbol{\nabla v} + (\boldsymbol{\nabla v})^T\right)$. Therefore, Eq. 2 can be rewritten as:

$$-\boldsymbol{\nabla} p + \boldsymbol{\nabla} \cdot \left(\eta\left(\boldsymbol{\nabla v} + (\boldsymbol{\nabla v})^T\right)\right) + \rho \boldsymbol{g} = 0 \tag{6}$$

Finally, using pressure from Eq. 5, Eq. 6 can be rewritten as:

$$\lambda \boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \boldsymbol{v}) + \boldsymbol{\nabla} \cdot \left(\eta\left(\boldsymbol{\nabla v} + (\boldsymbol{\nabla v})^T\right)\right) + \rho \boldsymbol{g} = 0 \tag{7}$$

The penalty method is been associated to the iterative Uzawa method, as described in detail in **?** and **?** (see Section 10.5). Results of benchmarks performed to verify the correctness in the implementation of Eq. 7 are shown in Sections 10.6 and 10.7.

To support large viscosities variations the penalty parameter is related to the effective elemental viscosity by means of a dimensionless coefficient, so that $\lambda = \lambda_e(e)\eta_{eff}(e)$ (**???**). Benchmark of the falling block (**???**) is performed to verify that the code can correctly deal buoyancy driven flows with strong viscosity contrasts (Section 10.8).

4 Sticky air and free surface

The Earth's surface can be treated by means of either the so-called sticky-air or a true free surface method, both of them implemented in the code. In the sticky-air method the surface is approximated with the introduction of a buoyant layer with a viscosity at least four orders of magnitude lower than the crust (**??**) and the interface between lithosphere and air is defined using a chain of passive markers that are advected as the Lagrangian markers. The correctness of the evolution of the markers chain is tested with the experiment of a 2D time-dependent Stokes sphere below a free surface and compared with results from ASPECT[2] (**????**) (Section 10.9). In the true free surface case the top boundary is assumed stress-free and velocities are not fixed. In this case topography variations are described by vertical deformations of the mesh that depend on the velocity field of the nodes that identify the free surface, while horizontal deformations are not taken into account. This procedure is known as the Arbitrary Lagrangian-Eulerian (ALE) method and its implementation follows the technique described in **?**. However, although the implementation of a true free surface better reproduces laboratory experiment, extremely small time steps can be necessary to maintain stability (**???**). Therefore, the stability algorithm proposed by **?** is implemented to avoid instabilities due to high density differences at the free surface when using too large time steps. The implementation of this algorithm are tested by performing the experiment described by **?** (Section 10.10).

The topography relaxation benchmark proposed by **?** is performed to verify that the code correctly recovers topography variations in case of both the sticky-air and the true free surface method (Section 10.11). Finally, Section 10.12 show the results of the spontaneous subduction experiment described by **?**.

5 Erosion and sedimentation

Surface processes at either the lithosphere-air interface or in correspondence of the free surface have been implemented by means of the software Fastscape[3] (**????**).FastScapeLib is a set of routines that solve the stream power law (enriched by a sediment transport/deposition term) (**?**), hillslope diffusion and marine transport and deposition (**?**). The partial differential equation solved by FastScapeLib is:

$$\frac{\partial h}{\partial t} = U - K_f A^{m_e} S^{n_e} + \frac{G}{A}\int_A \left(U - \frac{\partial h}{\partial t}\right) dA + K_d \boldsymbol{\nabla}^2 h \tag{8}$$

where h is the topography, U is the uplift, K_f is the erodibility coefficient, A is the upstream drainage area, S is the slope, m_e and n_e are parameters that should be chosen to constrain the ratio m_e/n_e between 0.35 and 0.8 (**???**), G is the dimensionless deposition/transport coefficient for the enriched stream power law and K_d is the effective diffusivity. Terms

$$K_f A^{m_e} S^{n_e}$$

and

$$K_d \boldsymbol{\nabla}^2 h$$

consider long-range processes described by a stream power law equation (**???**) and short-range processes are proportional to slope (**???**).

In case of sticky air, slope is determined by means of the markers chain, which is vertically corrected considering erosion and sedimentation rates. After the correction, continental markers above the markers chain are transformed in air markers and air markers below the markers chain become sediments. Similarly, slope is determined considering the nodes on the top boundary in case of true free surface and they are vertically corrected as for the markers chain. In this case, continental markers above the top boundary are simply deleted.

6 Non-linear rheologies

Non-linear rheologies are implemented combining viscous creep (dislocation and diffusion) and plastic yielding. For each marker, diffusion and dislocation viscosity ($\eta_{df|ds}$) can be determined as follows (**??**):

$$\eta_{df|ds} = f_s \left(\frac{d^m}{A}\right)^{\frac{1}{n}} \frac{2^{\frac{1-n}{n}}}{3^{\frac{n+1}{2n}}} I_2^{\frac{(1-n)}{n}} \exp\left(\frac{Q+pV}{nRT}\right) \tag{9}$$

where f_s is a scaling factor used to represent lithologies that are stronger or weaker than the base set, d is the grain size, m is the grain size exponent, n is the stress exponent, A is the uniaxial pre-exponential factor, I_2 is the square root of the second invariant of the strain rate tensor, Q is the activation energy, p is the pressure, V is the activation volume, R is the gas constant and T is the temperature. Pressures and temperatures are determined by the interpolation of

[2] https://aspect.geodynamics.org/
[3] https://fastscape.org/

the nodal parameters, while strain rates are calculated by means of the derivative of the velocities on the nodes. In case of diffusion creep $n = 1$ and $m > 0$, in case of dislocation creep $n > 1$ and $m = 0$. The viscous creep η_{cp} is then calculated as the harmonic average between η_{df} and η_{ds}:

$$\eta_{cp} = \left(\frac{1}{\eta_{df}} + \frac{1}{\eta_{ds}} \right)^{-1} \tag{10}$$

The implementation of the non-linear viscous creep viscosity depending on the strain rate is tested by means of the slab detachment benchmark (Section 10.13).

Plastic yielding is implemented rescaling η_{cp} in order to limit the stress below the yield stress σ_y (**???**), obtaining

$$\eta_{pl} = \frac{\sigma_y}{2I_2} \tag{11}$$

where η_{pl} is the plastic viscosity and the yield stress is determined following the Drucker-Prager criterion, such as

$$\sigma_y = C \ \cos(\phi) + p \ \sin(\phi) \tag{12}$$

where C is the cohesion and ϕ is the internal friction angle. In case of negative pressure it is imposed equal to 0, so that negative yield stress are excluded. The correctness of the non-linear solution in case of plastic viscosity with variable internal friction angle is verified performing indenter and brick experiments (Sections 10.14 and 10.15, respectively).

Strain softening is taken into account for both viscous creep and plastic viscosity (**?????**) by means of the accumulated strain ϵ memorised by each marker. Softening in the viscous creep determines a linear decrease of η_{vc} by means of a viscous strain softening factor W_S that increases linearly from W_{S_0} to W_{S_∞} for $\epsilon_{S_0} < \epsilon_S < \epsilon_{S_\infty}$ (**?**). This viscous softening can be related to strain-induced grain size reduction (**?**). Differently, plastic softening is simulated with a linear decrease of internal friction angle $\phi(\epsilon)$ and cohesion $C(\epsilon)$ in according to

$$\phi(\epsilon) = \phi_0 + (\phi_\infty - \phi_0)\frac{\epsilon - \epsilon_0}{\epsilon_\infty - \epsilon_0} \tag{13}$$

$$C(\epsilon) = C_0 + (C_\infty - C_0)\frac{\epsilon - \epsilon_0}{\epsilon_\infty - \epsilon_0} \tag{14}$$

where ϕ_0, C_0 and ϕ_∞, C_∞ are internal friction angle and cohesion for ϵ_0 and ϵ_∞, respectively (**????**). Plastic softening approximates deformation-induced softening of faults and brittle shear zones (**?**).

Plastic yielding and viscous creep are then combined to obtain a viscoplastic viscosity η_{vp} as follows, assuming that they are independent processes (**??**):

$$\eta_{vp} = \min(\eta_{cp}, \eta_{pl}) \tag{15}$$

Finally, effective viscosity η_{eff} is capped by the minimum and the maximum viscosity (η_{min} and η_{max}, respectively) to avoid extremely low or high viscosity (**?**) as follows

$$\eta_{eff} = \min(\max(\eta_{vp}, \eta_{min}), \eta_{max}) \tag{16}$$

Viscous creep and plastic yielding are non-linear rheologies because of their dependence on the velocity field through pressure and strain rates. Therefore, the solution is determined by means of Picard-type iterations, until convergence of the velocity field (**?**). The convergence is verified at each iteration i via the nonlinear residual $\boldsymbol{\mathcal{R}}^i$ that can be determined as

$$\boldsymbol{\mathcal{R}}^i = \mathbb{K}(\eta_{eff}(\dot{\boldsymbol{\epsilon}}^{i-1}, p^{i-1})) \cdot \boldsymbol{v}^{i-1} - \boldsymbol{f}^i \tag{17}$$

where $\mathbb{K}$ is the Stiffness matrix and $\boldsymbol{f}$ is the right hand side vector (**??**). The L_2-norm is extracted from $\boldsymbol{\mathcal{R}}^i$ and it is normalised as follows

$$\frac{||\boldsymbol{\mathcal{R}}^i||_2}{||\boldsymbol{\mathcal{R}}^0||_2} \tag{18}$$

where $||\boldsymbol{\mathcal{R}}^0||_2$ is the L_2-norm of the first nonlinear iteration. Since often the normalised nonlinear residual drop very quickly over the first 2-3 iterations, also the L_2-norm of velocity and pressure residuals are calculated as

$$\frac{||\Delta \boldsymbol{v}||_2}{||\boldsymbol{v}||_2} = \frac{||\boldsymbol{v}_i - \boldsymbol{v}_{i-1}||_2}{||\boldsymbol{v}_i||_2} \tag{19}$$

and

$$\frac{||\Delta p||_2}{||p||_2} = \frac{||p_i - p_{i-1}||_2}{||p_i||_2} \tag{20}$$

respectively. Iterations are performed until either maximal number of non-linear iterations (it_{max}) is reached or all the normalised nonlinear residual, the velocity residual and the pressure residual converge under a defined tolerance tol.