2. Numerical implementation

The following sections show how the numerical methods were implemented to solve the equations of conservation of mass, momentum and energy that were presented in the Basic theory section.

2.1. Mass and momentum equations

For a solution domain Ω, finding the flow velocity $u_i = g_i + v_i$ and pressure P states the Galerkin weak formulation for Stokes' flow, where g_i is a specified boundary velocity, v_i belongs to a set of functions $\mathcal{V}$ in which every function is equal to zero on the boundary where the ith components of the velocity is g_i, and P belongs to a set of functions $\mathcal{P}$ such that the equations of conservation of mass and momentum can be written as follows [1]:

$$\int_\Omega w_{i,j}\sigma_{ij}d\Omega - \int_\Omega qu_{i,i}d\Omega = \int_\Omega w_i f_i d\Omega + \sum_{i=1}^{n_{sd}} \int_{\Gamma_{h_i}} w_i h_i d\Gamma \qquad (2.1)$$

where $w_i \in \mathcal{V}$ and $q \in \mathcal{P}$ are weighting functions and the boundary conditions are defined:

$$u_i = g_i \text{ on } \Gamma_{g_i}, \sigma_{ij}n_j = h_i \text{ on } \Gamma_{h_i} \qquad (2.2)$$

where Γ_{h_i} is the boundary where the ith components of the forces are set to be h_i, n_j is the normal vector at the boundary Γ_{h_i}, and $f_i = g\rho_0\alpha\delta_{i3}$ [6]. The weak formulation can be re-written as below:

$$\int_\Omega w_{i,j}c_{ijkl}v_{k,l}d\Omega - \int_\Omega qv_{i,i}d\Omega - \int_\Omega w_{i,i}Pd\Omega = $$
$$\int_\Omega w_i f_i d\Omega + \sum_{i=1}^{n_{sd}} \int_{\Gamma_{h_i}} w_i h_i d\Gamma - \int_\Omega w_{i,j}c_{ijkl}g_{k,l}d\Omega \qquad (2.3)$$

where $c_{ijkl} = \eta(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk})$ is obtained from the stress tensor equation (see Eq. 1.4 in the basic theory section).

The velocity field, the pressure field and the weighting functions shape functions that interpolate the grid points at every node are given:

$$\mathbf{v} = v_i\mathbf{e}_i = \sum_{A\in\Omega^v-\Gamma^v_{g_i}} N_A v_{iA}\mathbf{e}_i \qquad (2.4)$$

$$\mathbf{w} = w_i\mathbf{e}_i = \sum_{A\in\Omega^v-\Gamma^v_{g_i}} N_A w_{iA}\mathbf{e}_i \qquad (2.5)$$

$$\mathbf{g} = \sum_{A\in\Omega^v-\Gamma^v_{g_i}} N_A g_{iA}\mathbf{e}_i \qquad (2.6)$$

$$P = \sum_{B\in\Omega^p} M_B P_B \qquad (2.7)$$

$$q = \sum_{B \in \Omega^p} M_B q_B \qquad (2.8)$$

where N_A is the shape function for the velocity at node A, M_B is the shape function for the pressure at node B, Ω^v is the velocity nodes set, Ω^p is the pressure nodes set and $\Gamma^g_{g_i}$ is the velocity nodes set along the boundary Γ_{g_i}. *Mandyoc* defines Ω^p at the center of each element, while Ω^v is defined at every element vertex. This avoids spurious flow solutions and numerical instabilities [1] and it keeps the velocity shape functions one order higher than the pressure shape functions, a common strategy used in finite element modeling of incompressible media [6].

From the shape functions above and the Galerkin weak formulation (Eq. 2.3), the following expression can be obtained:

$$\sum_{B \in \Omega^v - \Gamma^v_{g_j}} \left(\mathbf{e}_i^T \int_\Omega B_A^T D B_B d\Omega \mathbf{e}_j v_{jB} \right) - \sum_{B \in \Omega^p} \left(\mathbf{e}_i \int_\Omega N_{A,i} M_B d\Omega P_B \right) = \qquad (2.9)$$
$$\int_\Omega N_A \mathbf{e}_i f_i d\Omega + \sum_{i=1}^{n_{sd}} \int_{\Gamma_{h_i}} N_A \mathbf{e}_i h_i d\Gamma - \sum_{B \in \Gamma^v_{g_j}} \left(\mathbf{e}_i^T \int_\Omega B_A^T D B_B^T d\Omega \mathbf{e}_j g_{jB} \right)$$

and also:

$$\sum_{B \in \Omega^v - \Gamma^v_{g_j}} \int_\Omega M_A N_{Bj} d\Omega \mathbf{e}_j v_{jB} = 0 \qquad (2.10)$$

The matrix representation of these two equations above can be presented:

$$\begin{bmatrix} K & G \\ G^T & 0 \end{bmatrix} \begin{Bmatrix} V \\ P \end{Bmatrix} = \begin{Bmatrix} F \\ 0 \end{Bmatrix} \qquad (2.11)$$

where V is the vector of velocity values at Ω^v, P is the vector of pressure values at Ω^p, F is the resulting vector of the rigth-hand side of equations Eq. 2.9 or Eq. 2.10, K is the stiffness matrix, G is the discrete gradient operator, and G^T is the discrete divergence operator [1]. K, G and G^T are derived from the first and second terms of Eq. 2.9 and Eq. 2.10.

The matrix operator B from Eq. 2.9 contains the spatial derivatives of the shapes function N and, for a 2-D plane, can be written as:

$$B_A = \begin{bmatrix} N_{A,1} & 0 \\ 0 & N_{A,2} \\ N_{A,2} & N_{A,1} \end{bmatrix} \qquad (2.12)$$

Again from Eq. 2.9 and for 2-D plane strain problems, the effective viscosity matrix D can be written:

$$D = \begin{bmatrix} 2\eta & 0 & 0 \\ 0 & 2\eta & 0 \\ 0 & 0 & \eta \end{bmatrix} \qquad (2.13)$$

The stiffness matrix K and the gradient operator G can be written:

$$K_{lm} = \mathbf{e}_i^T \int_\Omega B_A^T D B_B d\Omega \mathbf{e}_j \tag{2.14}$$

$$G_{lm} = \mathbf{e}_i \int_\Omega N_A M_B d\Omega \mathbf{e}_j \tag{2.15}$$

where the subscripts A and B are the global velocity node numbers, i and j are the degree of freedom per grid node, ranging from 1 to n_{sd}, l and m are global equation numbers for the velocity ranging from 1 to $n_v n_{sd}$, where n_v is the number of velocity nodes in the grid.

2.2. Energy equation

It is possible to represent the energy conservation equation (Eq. 1.3) as a finite element problem for a solution domain Ω_V as follow:

$$\mathbf{M}\dot{\mathbf{a}}_T + (\mathbf{K_a} + \mathbf{K_c})\mathbf{a}_T = \mathbf{F} \tag{2.16}$$

where $\mathbf{M}$, $\mathbf{K}_a$, $\mathbf{K}_c$ and $\mathbf{F}$ are written below:

$$\mathbf{M} = \int_{\Omega_V} \mathbf{N}_V^T \rho_0 c_p \mathbf{N}_V d\Omega_V \tag{2.17}$$

$$\mathbf{K}_a = \int_{\Omega_V} \mathbf{N}_V^T \rho_0 c_p \mathbf{v} \cdot \mathbf{B}_V d\Omega_v \tag{2.18}$$

$$\mathbf{K}_c = \int_{\Omega_V} \mathbf{B}_V^T \rho_0 c_p \mathbf{v} \cdot \mathbf{B}_v d\Omega_v \tag{2.19}$$

$$\mathbf{F} = \int_{\Omega_V} \mathbf{N}_V^T \left(\frac{H}{c_p} - \frac{\alpha T g u_3}{c_p} \right) d\Omega_v \tag{2.20}$$

where $\mathbf{N}_V$ is a row vector of shape functions, $\mathbf{a}_T$ is a column vector of the unknown temperature parameters, $\dot{\mathbf{a}}_T$ is its time derivative, and $\mathbf{B}_V \equiv \nabla \mathbf{N}_V$.

> ❶ Note
>
> The superscript T represents the transpose of the matrix, while the non-superscript T represents the temperature.

$\mathbf{M}$ and $\mathbf{K}_c$ are symmetric, but $\mathbf{K}_a$ is not. This asymmetry decreases the accuracy of the solution when advection is more dominant than conduction ([7], chapter 2). To increase numerical accuracy and stability, *Mandyoc* uses the streamline upwind Petrov-Galerkin process to modify $\mathbf{K}_a$ to $\mathbf{K}_a^*$ [7, 8, 9]:

$$\mathbf{K}_a^* = \int_{\Omega_V} \mathbf{N}_V^{*T} \rho_0 c_p \mathbf{v} \cdot \mathbf{B}_V d\Omega_V \tag{2.21}$$

where N_{Vi}^* is:

$$N_{Vi}^* = N_{Vi} + \frac{\alpha_{opt} h^e \mathbf{v} \cdot \nabla N_{Vi}}{2|\mathbf{v}|} \tag{2.22}$$

where h^e is the characteristic element size in the advection velocity direction ($\mathbf{v}$), and α_{opt} is:

$$\alpha_{opt} = \coth Pe - \frac{1}{Pe}, \text{ where } Pe = \frac{|\mathbf{v}|h^e}{2\kappa\rho_0 c_p} \tag{2.23}$$

The time discretization is made by an implicit scheme [3]:

$$\frac{\mathbf{a}_T(t + \Delta t) - \mathbf{a}_T(t)}{\Delta t} = \theta\dot{\mathbf{a}}_T(t + \Delta t) + (1 - \theta)\mathbf{a}_T(t) \tag{2.24}$$

where $\theta = 0.5$ is a weighting parameter. Multiplying both sides of Eq. 2.24 by $\mathbf{M}(t + \Delta t)$ and assuming $\mathbf{M}(t + \Delta t) \approx \mathbf{M}(t)$:

$$\mathbf{M}(t + \Delta t)\frac{\mathbf{a}_T(t + \Delta t) - \mathbf{a}_T(t)}{\Delta t} = \theta[\mathbf{F}(t + \Delta t) - \mathbf{K}_T(t + \Delta t)\mathbf{a}_T(t + \Delta t)] + \\ (1 - \theta)[\mathbf{F}(t) - \mathbf{K}_T(t)\mathbf{a}_T(t)] \tag{2.25}$$

where $\mathbf{K}_T = \mathbf{K}_a^* + \mathbf{K}_c$.

Rearranging Eq. 2.25 allows to rewrite it in a numerical form to solve the energy equation.

$$[\mathbf{M}(t + \Delta t) + \Delta t\theta\mathbf{K}_T(t + \Delta t)]\mathbf{a}_T(t)(t + \Delta t) = \\ [\mathbf{M}(t + \Delta t) - \Delta t(1 - \theta)\mathbf{K}_T(t)]\mathbf{a}_T(t) + \\ \Delta t[\theta\mathbf{F}(t + \Delta t) + (1 - \theta)\mathbf{F}(t)] \tag{2.26}$$

2.3. Free surface

Mandyoc uses the *Free Surface Stabilization Algorithm* [10] to modify the Stokes equation and avoid numerical instabilities that can occur on the surface of the model. The up-and-down oscillations around the steady state ("sloshing instability" or "drunken sailer effect") would require small time steps, which would increase running time by unfeasible amounts.

The modification is done on the stiffness matrix K_e, such that $\tilde{K}_e = K_e + L_e$, where the correction L_e is evaluated at the boundary Γ_e of each finite element. The correction is given by the Eq. 2.27 below:

$$L_e = \int_{\Gamma_e} \mathbf{N}\Theta\Delta\rho\Delta t\mathbf{g}\mathbf{n}d\Gamma \tag{2.27}$$

where $\mathbf{N}$ is the element shape function, $0 \leq \Theta \leq 1$ is a wight factor of the correction term, $\Delta\rho$ is the density contrast between the two mediums, Δt is the numerical integration time step, $\mathbf{g}$ is the gravity acceleration vector, and $\mathbf{n}$ is the normal vector to the element.

2.4. Rheology

Considering a visco-plastic model, the effective viscosity η follows the formulation described by Moresi and Solomatov (1998) [11], which combines plastic deformation and viscous deformation:

$$\eta = \min\left(\eta_{plas}, \eta_{visc}\right) = \min\left(\frac{\tau_{yield}}{2\dot{\varepsilon}_{II}}, \eta_{visc}\right) \tag{2.28}$$

where τ_{yield} is the rupture tension and $\dot\varepsilon_{II} = (\dot\varepsilon'_{ij}\dot\varepsilon'_{ij}/2)^{1/2}$ is the second invariant of the deviatoric strain rate tensor, and η_{plas} and η_{visc} are the plastic and ductile viscosities.

2.4.1. Plastic deformation

The plastic deformation can be calculated using the Byerlee Law [12] to compute τ_{yield} and η_{plas}. Eq. 2.29 shows the relationship implemented in *Mandyoc*.

$$\tau_{yield} = c_0 + \mu\rho gz \tag{2.29}$$

where c_0 in the internal cohesion, μ is the friction coefficient, ρ is the density and z is the depth.

Alternatively, the user can choose to use the the Druker-Prager criterion [13], which is presented in Eq. 2.30.

$$\tau_{yield} = c_0 \cos\varphi + P\sin\varphi \tag{2.30}$$

where φ is the internal angle of friction.

2.4.2. Viscous deformation

Mandyoc contains several rheology models that the user can choose for viscous deformation. Among them, two will be discussed here.

The ductile rheology can be simulated using the Frank-Kamenetskii approximation, following the formulation described by Solomatov and Moresi (2000) [14], where the viscosity is a function of the temperature T as in the equation below. Such formulation was also used by Sacek (2017) [15] in an earlier *Mandyoc* version.

$$\eta_{visc}(T) = C\eta_r b^* \exp\left(-\gamma T\right) \tag{2.31}$$

where η_r is the reference viscosity, C and $\gamma = E_a/RT_b^2$ are constants, which in turn, E_a is the activation energy, R is the gas constant, and T_b is the basal temperature.

Additionally, the rheology can also be considered to follow a power law, as a function of the temperature T, compositional factor C, pressure P and strain rate ε as follows:

$$\eta_{visc} = CA^{\frac{-1}{n}}\dot\varepsilon^{\frac{1-n}{2n}}\exp\frac{Q+VP}{nRT} \tag{2.32}$$

where A is a pre-exponential scale factor, n is the power law exponent, $\dot\varepsilon$ is the second invariant of the strain rate tensor, Q is the activation energy, and V is the activation volume. The values of A, n, Q, and V are measured under laboratory conditions [16, 17].