A
User's Guide
to
I3ELVIS
in Subduction and Collision setup

by Ria Fischer
Geophysical Fluid Dynamics
Department of Earth Sciences
ETH Zurich

Zurich
January 2018

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Contents

1 Physics

1.1 Basic physical principals

1.1.1 The continuity equation

The continuity equation describes the conservation of mass, while it is displaced in a continuous medium. In its Lagrangian form it reads the following,

$$\frac{D\rho}{Dt} + \rho\nabla \cdot \vec{v} = 0, \tag{1.1}$$

where ρ denotes material density, $\vec{v}$ denotes displacement velocity and $\frac{D}{Dt}$ denotes the Lagrangian time derivative.

For many geological media like the crust or the mantle, where temperature and pressure are not too large and no phase changes occur, which would lead to larger volume changes, one can assume the following *incompressibility condition*,

$$\frac{D\rho}{Dt} = \frac{\partial\rho}{\partial t} + \vec{v}\nabla\rho = 0, \tag{1.2}$$

which means that the density of material points does not change in time.

This leads us to the following *incompressible continuity equation*, which is the same in its Eulerian and Lagrangian form,

$$\nabla \cdot \vec{v} = 0, \tag{1.3}$$

The *incompressible continuity equation* very widely used in numerical geodynamic modelling, although it is often a rather strong simplification.

1.1.2 The Navier-Stokes Euqation

The Navier-Stokes equation of motion in its full form reads the following,

$$\frac{\partial\sigma'_{ij}}{\partial x_j} - \frac{\partial P}{\partial x_i} + \rho g_i = \rho\frac{Dv_i}{Dt}, \tag{1.4}$$

where σ_{ij} is the strain-rate and $\vec{g} = (g_x, g_y, g_z)$ is the gravity vector.

In highly viscous flows the right-hand side of (1.4), the inertial forces $\rho\frac{Dv_i}{Dt}$, is much smaller compared to the gravitational force and therefore, be neglected. This leads to the *Stokes equation for creeping flow*,

$$\frac{\partial\sigma'_{ij}}{\partial x_j} - \frac{\partial P}{\partial x_i} + \rho g_i = 0. \tag{1.5}$$

Under Boussinesq approximation the density is assumed to be constant, except in the buoyancy force term, where temperature and volatile content play an important role (**?**). Taking into account the Boussinesq approximation, density $\rho(T, P, c)$ in the buoyancy term ρg_i may vary locally as a function of temperature T, pressure P and composition c,

$$\frac{\partial\sigma'_{ij}}{\partial x_j} - \frac{\partial P}{\partial x_i} = -\rho(T, P, c)g_i. \tag{1.6}$$

1.1.3 Heat conservation equation

The heat conservation equation, also called temperature equation, describes the heat balance in a convective medium, taking into account changes due to internal heat generation, advection and conduction. The Lagrangian heat conservation equation reads as follows,

$$\rho C_p \left(\frac{DT}{Dt}\right) = -\nabla\cdot\vec{q} + H_r + H_a + H_s + H_L, \tag{1.7}$$

with $\vec{q} = -k(T, p, c)\nabla T$, where thermal conductivity $k(T, P, c)$ depends on temperature, pressure and rock composition c. H_r, H_a, H_s, H_L denote radioactive, adiabatic, shear and latent heating.

Adiabatic and shear heating have shown to be important in many tectonic situations, which is why they are not taken as constant.

$$H_r = const. \tag{1.8a}$$

$$H_a = T\alpha\vec{v}\nabla P \tag{1.8b}$$

$$H_s = \sigma'_{ij}\left(\dot{\epsilon}'_{ij} - \dot{\epsilon}_{ij(elastic)}\right) \tag{1.8c}$$

$$H_L = const. \tag{1.8d}$$

The resulting set of the above equations, together with equations (1.6) and (1.8), is called the extended Boussinesq approximations.

1.1.4 Thermodynamics

Equation of State Different implementation depending on the value of the variable `densimod` are available.

- 0 : constante viscosity $\rho = \rho_0$.
- 1 : P,T dependant density $\rho = (1 - b_\rho * (T_K - 298.15)) * (1 + a_\rho * (P_{kbar} - 0.001))$
- 2 : add the effect of thermodynamic database on density and water content. $\rho = \rho/(1+\Delta H_2O*(\frac{\rho}{1050}-1)*10^{-2})$ || ΔH_2O is the difference in water content introduced by recalculation form the database.
- 3 : only get water content from thermodynamic database

Furthermore the effect of mantle depletion is taken in account in density calculation : $\rho = 1.0-0.4*ex$. ex being the amount of melt extracted from the marker saved in `markex` variable.

Thermal conductivity Thermal conductivity is by default calculated as dependant of pressure and temperature. An enhanced conductivity is calculated for based on equations from Gregg et al 2009.

$$K_T = K_{T0} + \frac{K_{Tkoef}}{T_K + 77.0} * exp(K_{Pkoef} * P_{bar}) \tag{1.9}$$

K_{TO}, K_{Tkoef} and K_{Pkoef} are fixed for each rocktype.

Enhanced conductivity is calculated in oceanic lithosphere when in are in an area defined by the parameters: waterlev, circdepth and circtemp.

$$K_T = 3 * (Nu - 1) * exp(\frac{3}{4}[2 - \frac{T_k - 273.15}{T_{circ} - 273.15} - \frac{y - markto}{y_{circ}}]) \tag{1.10}$$

Nu is the nusselt number.

1.1.5 Rheology

Viscosity

Plastic yield strength

$$\sigma_{yield} = C + sin(\phi_{dry})(1 - \lambda)P \tag{1.11}$$

where σ_{yield} denotes the shear stress limit after which plastic yielding occurs, C is the cohesion, ϕ_{dry} is the effective internal friction angle in dry rock, $\lambda = 1 - \frac{P_{fluid}}{P_{solid}}$ is the pore fluid pressure factor and $P = P_{solid}$ is the mean stress of the solid.

$$\begin{aligned} C = a_1, sin(\phi_{dry}) = b_1, \forall \varepsilon_{II} > \varepsilon_1 \\ C = a_0, sin(\phi_{dry}) = b_0, \forall \varepsilon_{II} < \varepsilon_0 \\ C = a_0 + \frac{\varepsilon_{II} - \varepsilon_0}{\varepsilon_1 - \varepsilon_0}(a_1 - a_0) \\ sin(\phi_{dry}) = b_0 + \frac{\varepsilon_{II} - \varepsilon_0}{\varepsilon_1 - \varepsilon_0}(b_1 - b_0) \end{aligned} \tag{1.12}$$

Peirl's creep (Katayamo & Karato, 2008)

$$\dot{\epsilon}_{II} = A_{Peirl}\sigma_{II}^2 exp\left\{-\frac{E_a + PV_a}{RT}\left[1 - \left(\frac{\sigma_{II}}{\sigma_{Peirl}}\right)^k\right]^q\right\} \tag{1.13}$$

(TODO: No! Visco-plastic!) A visco-elasto-plastic rheology is employed, with the deviatoric strain-rate $\dot{\epsilon}'_{ij}$ being composed of the following components,

$$\dot{\epsilon}'_{ij} = \dot{\epsilon}'_{ij(viscous)} + \dot{\epsilon}'_{ij(elastic)} + \dot{\epsilon}'_{ij(plastic}, \tag{1.14}$$

where

$$\dot{\epsilon}'_{ij(viscous)} = \frac{1}{2\eta}\sigma'_{ij}, \tag{1.15a}$$

$$\dot{\epsilon}'_{ij(elastic)} = \frac{1}{2\mu}\frac{D\sigma'_{ij}}{Dt}, \tag{1.15b}$$

$$\dot{\epsilon}'_{ij(plastic)} = \chi\frac{\partial G}{\partial \sigma'_{ij}} = \chi\frac{\sigma'_{ij}}{2\sigma_{II}} \quad \text{for } G = \sigma_{II} = \sigma_{yield}. \tag{1.15c}$$

where η denotes viscosity, σ'_{ij} denotes the deviatoric stress tensor, μ denotes the shear modulus, G is the plastic potential, σ_{yield} is yield strength, $\sigma_{II} = \sqrt{\frac{1}{2}\sigma'^2_{ij}}$ is second deviatoric stress invariant and χ is plastic potential.

Generally the strain tensor ϵ_{ij} can be defined as a function of displacement $\vec{u} = (u_x, u_y, u_z)$,

$$\epsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{x_i}\right). \tag{1.16}$$

and the second strain rate invariant is given by $\dot{\epsilon}_{II} = \sqrt{\frac{1}{2}\dot{\epsilon}'^2_{ij}}$

The viscosity η is defined as follows,

$$\eta = \left(\frac{2}{\sigma_{II}}\right)^{(n-1)} \frac{F^n}{A_D} exp\left(\frac{E+PV}{RT}\right), \tag{1.17}$$

where A_D, E, V and n are experimentally defined flow parameters, R is the gas constant and F is a dimensionless factor depending on the type of experiment (triaxial compression, simple shear).

The viscous constitutive relationship relates stress σ_{ij} with strain ϵ_{ij},

$$\sigma'_{ij} = 2\eta\dot{\epsilon}'_{ij} + \delta_{ij}\eta_{bulk}\dot{\epsilon}_{kk}, \tag{1.18}$$

where σ'_{ij} is the deviatoric stress, $\dot{\epsilon}'_{ij}$ is the deviatoric strain-rate, $\dot{\epsilon}_{kk}$ is the bulk strain-rate, and η and η_{bulk} are shear and bulk viscosity.

1.1.6 Impact treatment

The actual impact is not part of the model. The model only starts after the intrusion of the impactor into the parent body. Processes like crater excavation, redistribution of impactor and parent body material around the planet or decompression melting are not considered. A simplified model takes into account the thermal anomaly created by the impactor. A region called the isobaric core, of uniform temperature increase and shock pressure around the impactor can be found (**?**).

$$R_{ic} = 3\frac{1}{3} r_{imp} \tag{1.19}$$

where R_{ic} is the radius of the isobaric core and r_{imp} is the radius of the impactor.

Thermal anomaly in the isobaric core has been approximated by **?** in the following way,

$$\Delta T = \frac{4\pi}{9}\frac{\psi}{F}\frac{\rho_P G R_P^2}{c_P} \tag{1.20}$$

where ψ is the efficiency of conversion of kinetic energy to thermal energy and in this thesis assumed to be 0.3.

Outside the isobaric core, for $r > R_{ic}$, the thermal anomaly ΔT is decaying exponentially, according to the following rule (**??**),

$$T(r) = \Delta T \left(\frac{R_{ic}}{r} \right)^{4.4} \tag{1.21}$$

1.1.7 Computation of crust

This algorithm is only implemented in the 3D code.

Silicate melt within a certain depth is positively buoyant ($d_{depthmelt} = 2e5$) and rises up to the surface (**?**). Only markers with a melt fraction between 1% and 20% are considered, as this corresponds roughly to the pyroxene fraction in a fertile mantle (**?**). Silicate melt on markers fulfilling these criteria now percolates upwards through the mantle and at the surface crust is formed by freezing of the silicate melt.

1.1.8 Phase transitions, melting and hydration reactions

Melting +20

Melting occurs a soon as the melt fraction is larger than 0. +20 is added to the type rock type (not for 11/13/31/33 and 19/20/21). A rock type >20 signifies that the material is partially molten.

Freezing -20

If the melt fraction becomes 0, -20 is subtracted from the type (not for 11/13/31/33 and 19/20/21). A rock type <20 signifies that the material is fully solid.

Hydrated Peridotite 11 -> 34 (-> 14) (and 31 -> 14)

Hydrated wet mantle peridotite (type 11) can be molten (type 34) and resolidified again to quenched dry mantle peridotite (type 14).

Hydrated, wet mantle peridotite (11) -> Molten Peridotite (34) -> Resolidified dry quenched mantle peridotite (14)

(31 -> 14)

Mantle hydration [9,10,12,14] -> 11

If water is present, lithospheric (type 9) and asthenospheric (type 10) peridotite as well as dry peridotite from the shear zone (type 12) and the resolidified quenched (type 14) peridotite are hydrated (type 11).

Crust hydration 5/6 -> 17/18

If water is present, continental upper (type 5) and lower (type 6) crust are hydrated (type 17+18).

Layered Sedimentation Sequence of 3,4,3,4,3,4,....

Formation of new crust -> 16

Antigorite weakening 13 -> 11 (11 -> 13)

Serpentinization of hydrated peridotite depending on Antigorite pressure and temperature field following (**?**)

Eclogitization No type change.

Happens for type 7/8 (upper/lower oc. crust) and 27/28 (molten upper/lower oc. crust).

1.2 I2ELVIS

To model two-dimensional creeping flow under extended Boussinesq approximation, with both thermal and chemical buoyancy, the conservative finite-difference code I2ELVIS (**??**) is used, which operates on a staggered grid and uses the moving marker technique. Silicate material is assumed to have temperature-, pressure-, strain-rate and melt fraction-dependant visco-elasto-plastic rheology. Furthermore, impact heat, batch melting of silicates and phase changes have all been taken into account.

1.3 I3ELVIS

The 3D models have been carried out by the 3D numerical I3ELVIS (**?**) code which is also based on a conservative finite difference method with a marker-in-cell technique and multigrid solver (**??**). Additionally, the 3D code also features impact heat, batch melting of silicates

and phase changes as discussed in **?** for the 2D case. The initial thermal-chemical model setup (including initial conditions, boundary condition and fluid/melt transport mechanism) and numerical approach are kept as similar as possible to the 2D models. Furthermore, the 3D code also features computation of the primordial crust from silicate melt.

Parameter	Symbol	Value	Unit
Radius of planetary body	R_{Mars}	3389	km
Radius of impactor core	r_{ic}	$232 - 500$	km
Radius of final core rel. to R_{planet}	r_{core}	0.5	%
Temperature of impactor core	T_{ic}	$1300 - 2300$	K
Temperature of protocore	T_p	$1300 - 2500$	K
Temperature of diapirs	T_d	$1300 - 2300$	K
Mean temperature of final core	$\bar{T}_c$	–	K
Mean temperature of silicate mantle	$\bar{T}_m$	–	K
Mean temperature of planetary body	$\bar{T}_{tot}$	–	K
Mean density of final core	$\bar{\rho}_c$	–	$kg\,m^{-3}$
Mean density of silicate mantle	$\bar{\rho}_m$	–	$kg\,m^{-3}$
Mean density of planetary body	$\bar{\rho}_{tot}$	–	$kg\,m^{-3}$
Volume fraction of iron (3D)	$f_{Fe,vol}$	0.1	%
Mass fraction of iron (3D)	$f_{Fe,mass}$	0.2	%
Gravitational acceleration	g	3.73	$m\,s^{-2}$

Table 1: **TODO: complete table** List of parameters

2 Code Layout

2.1 Files layout

Short description of the code files and their functions.
The code is divided into eight .c files

- in3mg.c : This is the file that will read the parameters files to write the initial condition file.
- i3mg.c : Main code file.
- head3mg.c : In this file are the global variable and function are declared (internal counters are declared inside the function or loop that use them). Most variables roles are describe in this file.
- load3mg.c : In this file are the functions to read the input files and write their content to the code variables, and to save the results to the output files. The functions have a binary or text option but only the binary option should be used as the text option would slow the code significantly.
- move3mg.c : Contain the function to resolve continuity-momentum equations. From the function to calculate the stress and strain to the multi-grid resolution functions.
- mark3mg.c : Contain all the interpolation functions and the functions to advect the marker.
- gauss3mg.c : others Gauss-Scheidel solvers
- heat3mg.c : Solve heat conservation equation and using the thermodynamic database.

To those eight files we can add three configuration files. mode.t3c that contain all the configurations and parameters of the code that you will be running and the name of all the output files. init.t3c contain the parameters defining the initial geometry of model (support rectangular, Ellipsoidal and cylindrical geometry), the different rock types and there rheological parameters.files.t3c contain only the number of the next output file (or 0).
Finally there is a large number of thermodynamics database that are used for proper calculation of temperature controlled by phase changes and rock behaviour in P,T,t conditions.

2.2 Functions layout

3 t3c files

3.1 file.t3c

The file 'file.t3c' contains only one number, which gives the number of the NEXT file to write. The number given here is the actual number of the file, if all files would be numbered from the beginning (including the initial file), starting with 0. The file name though can be different (and is given in the 'mode.t3c'-file) and can contain any different number.

3.2 init.t3c

Null-point is in the frontal upper left corner of the grid.

3.2.1 Grid parameter description

The first part of 'init.t3c' describes some basic grid parameters:

Parameter	Description	Initial value	Unit
xnumx	Number of nodes in x	$16N + 5$ (4 multi-grid levels)	-
ynumy	Number of nodes in y	$16N + 5$ (4 multi-grid levels)	-
znumz	Number of nodes in z	$16N + 5$ (4 multi-grid levels)	-
mnumx	Number markers per cell in x	-	-
mnumy	Number markers per cell in y	-	-
mnumz	Number markers per cell in z	-	-
xsize	Dimension of model in x	-	$[m]$
ysize	Dimension of model in y	-	$[m]$
zsize	Dimension of model in z	-	$[m]$
pxinit	x-coordinate of initial pressure cell	$(xnumx - 1)/2$	-
pyinit	y-coordinate of initial pressure cell	$(ynumy - 1)/2$	-
pzinit	z-coordinate of initial pressure cell	-	-
pinit	Initial pressure in initial pressure cell	-	$[Pa]$
GXKOEFF	Gravitational acceleration in x	-	$[m/s^2]$
GYKOEFF	Gravitational acceleration in y	-	$[m/s^2]$
GZKOEFF	Gravitational acceleration in z	-	$[m/s^2]$
timesum	Starting time	-	$[years]$
nonstab	number of random number generated	-	
xnonstab	maximum random displacement of markers in x direction	-	
ynonstab	maximum random displacement of markers in y direction	-	
znonstab	maximum random displacement of markers in z direction	-	
markers types file name Y(Name) N(0)	Number of initial output file	-	-
data output file name	Name of initial output file	name_0.prn	-
TYPE	type of data output	b (= binary), no other supported	-

Table 2: Grid parameter description

3.2.2 Rock type description

In the next part a table lists all possible types of material compositions together with there rheological properties.

Parameter	Description	Unit
rocknum	Rock number	-
markn0	Individual lower viscosity limit	-
markn1	Individual upper viscosity limit	-
marks0	Connate water content at surface and surface temperature	$[wt\%]$
marks1	Individual upper stress limit	-
Nu or marknu	Newtonian viscosity	$[Pa^{MM} * s]$
DE or markdh	Activation energy E_a	$[J]$
DV or markdv	Activation volume V_a	$[J/bar]$
SS or markss	dislocation-diffusion transition stress σ_{crit}	$[Pa]$
MM or markmm	Stress exponent for creep law	$(Power)$
LL or markll	Pore fluid pressure factor $1 - \lambda$ (see eq. 1.11)	$(koef)$
a0 or marka0	Cohesion (see eq. 1.11)	$[Pa]$
a1 or marka1	unused	-
b0 or markb0	Sinus of effective internal friction angle in dry rock (see eq. 1.11)	[deg°]/[rad]
b1 or markb1	unused	-
e0 or marke0	Koef in ductile rheology weakening **(TODO: ???)**	**(TODO: ???)**
e1 or marke1	Koef in ductile rheology weakening **(TODO: ???)**	**(TODO: ???)**
RO or markro	Density	$[kg/M^3]$
bb or bRo or markbb	Thermal expansion α (see eq.**??**)	$[1/K]$
aa or aRo or markaa	Compressibility β (see eq.**??**)	$[1/kbar]$
CP or markcp	Heat capacity	$[J/kg]$
Kt or markkt	Thermal conductivity	$[Wt/(m * K)]$
Kf or markkf	Temperature dependency coefficient of conductivity	$[Wt/m]$
Kp or markkp	Pressure dependency coefficient of conductivity	$[1/bar]$
Ht	heat generation	$[Wt/kg]$

Table 3: Rock type parameters

Nr.	Description	Rheology
00	Air	-
01	Water	-
02	-	-
03	Sediments 2	WET QUARTZITE RANALLI 1995
04	Sediments 3	WET QUARTZITE RANALLI 1995
05	Upper Continental Crust	dry,felsic - WET QUARTZITE RANALLI 1995
06	Lower Continental Crust	dry,felsic - WET QUARTZITE RANALLI 1995
07	Upper Oceanic Crust	Basalts - WET QUARTZITE RANALLI 1995
08	Lower Oceanic Crust	Gabbro - An75 Ranalli1995
09	Lithospheric Mantle	dry peridotite - DRY OL Ranalli1995
10	Asthenospheric Mantle	dry peridotite - DRY OL Ranalli1995
11	Hydrated Mantle	wet peridotite - WET OL Ranalli1995
12	Shear Zone	dry mantle peridotite - WET OL Ranalli1995
13	Serpentinized Mantle	wet peridotite
14	Resolidified peridotite/ quenched mantle	dry peridotite
15	-	-
16	Newly formed crust	Basalt
17	Hydrated Upper Crust	felsic - Hydrated WET QUARTZITE RANALLI 1995
18	Hydrated Lower Crust	felsic - Hydrated WET QUARTZITE RANALLI 1995
19	-	
20	-	
21	-	
22	-	-
23	Partially molten Sediments 2	
24	Partially molten Sediments 3	
25	Partially molten Upper Continental Crust	felsic
26	Partially molten Lower Continental Crust	molten gabbro
27	Partially molten Upper Oceanic Crust	molten basalt
28	Partially molten Lower Oceanic Crust	molten gabbro
29	Partially molten Lithospheric Mantle	dry peridotite
30	Partially molten Asthenospheric Mantle	dry peridotite
31	-	-
32	Partially molten Shear Zone	dry peridotite
33	-	-
34	Partially molten Peridotite	
35	-	-
36	Partially molten Newly Formed Crust	molten basalt
37	Partially molten Hydrated Upper Crust	molten gabbro
38	Partially molten Hydrated Lower Crust	molten gabbro
+50	Water markers	
+100	External Composition	
NAN	Undefined composition	black

Table 4: Rock types

Compositions 0-19 are all solids (except air and water). 20-39 are the equivalent melts. 50-89 are fluid markers. 100-139 are external compositions of the equivalent type, waiting outside the boundary to come into the model. They have the same properties as their internal equivalents.

3.2.3 Boundary conditions

Boundary conditions are prescribed separately for each of the six faces of the cube. Boundary conditions on each face don't have to be homogeneous. Special 'open boundary conditions' can be prescribed, where e.g. the last marker gets 99% of the velocity of the second to last and so on.

Parameter	Description	Range of values
Val	BC type	P, Vx, Vy, Vz, T, X, Y, Z, M
m10	starting node position in x	
m11	ending node position in x	
m20	starting node position in y	
m21	ending node position in y	
m30	starting node position in z.	
m31	ending node position in z	
Const		
Koef		
nshiftx		
nshifty		
nshiftz		
Koef1		
nshiftx1	Ignored if Koef1 is 0	
nshifty1	Ignored if Koef1 is 0	
nshiftz1	Ignored if Koef1 is 0	
Koef2		
nshiftx2	Ignored if Koef2 is 0	
nshifty2	Ignored if Koef2 is 0	
nshiftz2	Ignored if Koef2 is 0	

Table 5: Boundary condition parameters

P, Vx, Vy, Vz, T boundary conditions For all nodes in the given range $[m10, m11] \times [m20, m21] \times [m30, m31]$ the following boundary condition is applied:

$$A(x, y, z) = Const + Koef \cdot A(x + nshiftx, y + nshifty, z + nshiftz) \tag{3.1}$$

E.g. the following part in the *init.t3c* file:

/Left_Bondary

/Val___m10_m11___m20_m21___m30_m31___Const___Koef_dm1_dm2_dm3
Vx____0___0_____0___y-1___0___z-1___3e-10___0____0___0___0

translates into the following boundary condition:

$$V_x(x,y,z) = 3 \times 10^{-10} \quad | \; x = 0, y \in [0, y-1], z \in [0, z-1] \tag{3.2}$$

X, Y, Z coordinates definition For all nodes in the given range $[m10, m11] \times [m20, m21] \times [m30, m31]$ the following equation is applied, shown at the example of the x-grid:

$$X(x,y,z) = X(x-1,y,z) + Const + (Koef - Const) \cdot \frac{(x - m10)}{(m11 - m10)} \quad | \; Koef1 = 0 \tag{3.3a}$$

$$X(x,y,z) = X(x-1,y,z) + exp\left(log(Const) + log\left(\frac{Koef1}{Const}\right)\right) \cdot \frac{(x - m10)}{(m11 - m10)} \tag{3.3b}$$

M marker grid set to cell For all nodes in the given range $[m10, m11] \times [m20, m21] \times [m30, m31]$ additional markers are added. All parameters ($Koef$, $Koef1$, $Koef2$, $nshiftx$, $nshiftx1$,..,$nshiftz1$,$nshiftz2$) need to be set. $nshiftx$ gives the shift in x, $nshifty1$ gives the shift in y, $nshiftz2$ gives the shift in z and all other shift parameters are ignored. If $Koef$ is set, random nonstability is set on the new marker field in X in the following way:

$$markx = x + \frac{rand()\,\%\,(\lfloor Const \rfloor * 2 + 1) - \lfloor Const \rfloor}{Const} \cdot \frac{X(x+1,y,z) - X(x,y,z)}{nshiftx} \cdot Koef \tag{3.4}$$

Random nonstability is set in Y for $Koef1 > 0$ and in Z for $Koef2 > 0$ in the same way.

3.2.4 Box description

Arbitrary shapes of cubes are prescribed by setting the (x,y,z) coordinate of each of the 8 corners.

Parameter	Description	Range of value
Type	Rock type	$0 - 140$
$[x_0, y_0, z_0]$	front upper left	
$[x_1, y_1, z_1]$	front lower left	
$[x_2, y_2, z_2]$	front upper right	
$[x_3, y_3, z_3]$	front lower right	
$[x_4, y_4, z_4]$	back upper left	
$[x_5, y_5, z_5]$	back lower left	
$[x_6, y_6, z_6]$	back upper right	
$[x_7, y_7, z_7]$	back lower right	

Table 6: Box parameters

Coordinates can be given either relative [0,1], or absolute [m0, mMAXSIZE], where MAXSIZE cannot be larger than the maximum domain size in this direction. With the leading 'm', the value is interpreted as an absolute value in meters.

e.g.

/Type=RockType
/Type__x0__y0__z0___x1__y1__z1_____x2__y2__z2_____x3__y3__z3
/_____x4__y4__z4___x5__y5__z5_____x6__y6__z6_____x7__y7__z7
/Asthenosphere
10____0_m15000_0___0__1.1__0_____1.1_m15000_0_____1.1_1.1__0
_____0_m15000_1___0__1.1__1_____1.1_m15000_1_____1.1_1.1__1

3.2.5 Temperature box description

Arbitrary shapes of cubes are prescribed by setting the (x,y,z) coordinate of each of the 8 corners.

Parameter	Description
Type	Box type: 0 - simple box; 1&4 - age box; 5&6 - transitional
$[x_0, y_0, z_0]$	front upper left
$[x_1, y_1, z_1]$	front lower left with $x_1 = x_0$, $z_1 = z_0$
$[x_2, y_2, z_2]$	front upper right with $z_2 = z_0$
$[x_3, y_3, z_3]$	front lower right with $x_3 = x_2$, $z_3 = z_0$
$[x_4, y_4, z_4]$	back upper left with $x_4 = x_5$, $z_4 = z_7$
$[x_5, y_5, z_5]$	back lower left with $z_4 = z_7$
$[x_6, y_6, z_6]$	back upper right with $x_6 = x_7$, $z_6 = z_7$
$[x_7, y_7, z_7]$	back lower right
t_0	simple box: temperature P_0; age box: surface temperature
t_1	simple box: temperature P_1; age box: initial temperature
t_2	simple box: temperature P_2; age box: thermal diffusivity κ in P_0
t_3	simple box: temperature P_3; age box: thermal diffusivity κ in P_2
t_4	simple box: temperature P_4; age box: thermal diffusivity κ in P_4
t_5	simple box: temperature P_5; age box: thermal diffusivity κ in P_6
t_6	simple box: temperature P_6; age box: characteristic diffusion time
t_7	simple box: temperature P_7; age box: overprinted linear geotherm $[°/m]$

Table 7: Temperature box parameters

Coordinates can be given either relative [0,1], or absolute [m0, mMAXSIZE], where MAXSIZE cannot be larger than the maximum domain size in this direction. With the leading 'm', the value is interpreted as an absolute value in meters.

Box type 0: simple box The given temperatures are set to the given coordinates and temperature is interpolated linearly within the box.

Box type 1&4: age box Within the given coordinates the temperature is given by conductive cooling in the following way:

$$T(y,t) = T_1 - (1 - erf(\frac{y}{2\sqrt{\kappa\tau}}))(T_1 - T_0), \tag{3.5}$$

where $\kappa = \dfrac{k}{\rho c_p}$ is the thermal diffusivity interpolated over the corner of the box and $\tau = \dfrac{l^2}{\kappa}$ is the characteristic diffusion time.

Box type 5&6: transitional box No new temperatures are set but existing temperatures at the coordinates of the box corners are read. Those are then interpolated linearly through out the box.

E.g. the continental crust and lithosphere is modelled in a simple box by a geothermal gradient of $\sim 17\,K/km$ spanning the whole width of the model and starting at $12\,km$ depth with a surface temperature of $273\,K$, going down to a depth of $90\,km$:

```
/T_BOXES_DESCRIPTION
/Typ__x0_______x2_______y0_______y1_______y2_______y3_______z0_______
          _x5_______x7_______y4_______y5_______y6_______y7_______z7_______
          t0___t1___t2___t3___t4___t5___t6___t7
/Oceanic_and_continental_geotherms
0      -0.000001  m350001   m12000      m90000     m12000      m90000  -0.000001
          -0.000001  m350001   m12000      m90000    m12000      m90000   1.000001
          273   1595  273   1595  273   1595  273   1595
```

3.3 mode.t3c

3.3.1 Timestepping description

The first/second line gives the filename and type of the initial file.

In the next very large block one timestep corresponds with one line and the data is only valid for one timestep. Seperate timestep parameters are given in table 8.

Parameter	Description
SAVEFILE	name of the file (if this name is non-unique, the file will be overwritten without warning)
TYPE	file type: b = binary (no other supported)
cyc0max	number of iterations
maxxystep	maximum step size in dimension **(TODO: ???)**
maxtkstep	maximum step size in temperature **(TODO: ???)**
maxtmstep	maximum step size in time **(TODO: ???)**
nubeg	global lower viscosity cut-off
nuend	global upper viscosity cut-off
p0koef	Pressure penalty factor 1, Relaxation parameter in Continuity equation
p1koef	Pressure update interpolation from coarser multigrid levels
p2koef	Pressure penalty factor 2 **(TODO: ???)**
unused	-
multinum	number of multigrid levels
stp1	number of iterations/ repeats for whole timestep
multinum2	number of multigrid levels 2 **(TODO: ???)**

Table 8: Timestep parameters

3.3.2 General parameters

The following parameters are mainly used to control the general behaviour of the program.

Parameter	Description	Default
loadmod	load from data file (1) or set initial conditions (0)	1
printmod	print information on the monitor, Yes (1)/ No (2)	1
crustmod	print information on crustthickness in files for each nth timestep; 0 = disable	-
dynamod	do dynamo calculations for each nth timestep; 0 = disabled	-
fl0num	number of otput file Names	-
movemod	do velocity-pressure iterations, solve continuity and Stokes equation	1
tempmod	do temperature iterations, solve heat transfer equation	1
markmod	move markers Y(1-simple,2-Runge-Kutta4)/N(0)	2
gridmod	recalculate density and viscosity	1
outgrid	marker move out of grid Y(0)/N(1) Orthogonal Only (2)	2
densimod	mode of density calculation: 0-constant, 1-PT-dependent, 2-TDbase 3-PT-dependent+WaterTDbase	3
stp100	**(TODO: ???)**	9000
CTreset	composition/temperature reset for water/air at 100 km above surface Y(1)/N(0)	1
smeltextract	extract melt when moving markers Y(1)/N(0)	1
sthdatabase	Use of Mars thermodynamic database Y(1) or standard database N(0)	1
p2vmod	convert each nth prn to vtr /N(0)	10
filestop	number of timesteps to execute before exiting	50

Table 9: General parameters

3.3.3 Erosion and Sedimentation parameters

Parameter	Description	Initial value
eroslev	Erosion level: markers above this depth which are neither sticky air nor water get converted to sticky air	8000
sedilev	Sedimentation level: sticky air or water below this depth gets converted to sediments	20000
waterlev	Water level: sticky air markers below water level are converted to water and water markers above water level are converted to sticky air	12000

Table 10: Erosion and Sedimentation parameters

3.3.4 Velocity- and Pressure-iterations parameters

The following parameters are mainly needed to solve the Continuity and Stokes equation with multigrid. Some these parameters already appeared in table 8 and their global value given here will be overwritten with the new value for each timestep.

Parameter	Description	Initial value
cyc1max	unused	3000
DIVVMIN	Continuity equation lower error bound	$3e{-}03$
STOKSMIN	Stokes equation lower error bound	$5e{+}01$
DIVVMAX	Continuity equation upper error bound	$0e{-}03$
STOKSMAX	Stokes equation upper error bound	$3e{-}03$
multinum	number of multigrid levels; overwritten by separate timestep value	4
multicyc	Number of whole multigrid V-cycle iterations	1
multinnn	V-cycle structure: number of GS-iterations for each multigrid level in upcycle and downcycle.	4 16 16 32 0; 4 16 16 32 64
p0koef	Global pressure penalty factor 1, Relaxation parameter in Continuity equation; overwritten by separate timestep value	$3.0e{-}01$
p1koef	Global pressure update interpolation from coarser multigrid level; overwritten by separate timestep value	$1.0e{-}00$
p2koef	Global pressure penalty factor 2 **(TODO: ???)**; overwritten by separate timestep value	$0.0e{-}00$
v0koef	Velocity penalty facto. Relaxation parameter in vx-,vy-,vz-Stokes equation	$1.0e{-}00$
v1koef	**(TODO: ???)**	$1.0e{-}00$
nubeg	global lower viscosity cut-off; overwritten by separate timestep value	$1e{+}18$
nuend	global upper viscosity cut-off; overwritten by separate timestep value	$1e{+}25$
nukoef	Average ν for pressure optimisation	0.0
viscmod	Effective viscosity mode. 0 = lin. interp; 1 = exp. interp; 2 = inverse interp;	0
viscoutermod	viscosity in space/air/water; 1-gradual increase in space, 2-gradual increase in water/air	2
spheryn	Spherical gravity. 0 = off; 1 = on;	0

Table 11: V and P iterations parameters

viscmod: Effective viscosity interpolation Type of interpolation done to obtain effective viscosity. viscmod = 0 uses linear interpolation (eq. 3.6a), viscmod = 1 uses exponential interpolation (eq. 3.6b) and viscmod = 2 uses inverse interpolation (eq. 3.6c).

$$\eta_{eff} = \frac{1}{8}\sum_i \eta_i \tag{3.6a}$$

$$\eta_{eff} = exp\left(\frac{1}{8}\sum_i \left(log\left(\eta_i\right)\right)\right) \tag{3.6b}$$

$$\eta_{eff} = \frac{1}{\frac{1}{8}\sum_i \frac{1}{\eta_i}} \tag{3.6c}$$

3.3.5 Temperature-iterations parameters

The following parameters are mainly needed to solve the temperature equation with multigrid.

Parameter	Description	Initial value
cyc2max	unused	2500
HEATMIN	Temperature equation lower error bound	$1e{-}4$
multinumt	Number of multigrid levels	0
multicyct	Number of whole multigrid V-cycle iterations	1
multittt	V-cycle structure: number of GS-iterations for each multigrid level in upcycle and downcycle	1; 0
t0koef	**(TODO: ???)**	$1.0e{-}00$
t1koef	**(TODO: ???)**	$1.0e{-}00$
heatdif	**(TODO: ???)**	1.0
frictyn	**(TODO: ???)**	1
adiabyn	**(TODO: ???)**	1

Table 12: T iterations parameters

3.3.6 Hydration and melting parameters

Parameter	Description	Initial value
tkpor	(TODO: ???)	97300000.0
zmpor	(TODO: ???)	75000
vyfluid	Initial fluid velocity	$-3e-09$
vymelt	Initial melt velocity	$-3e-09$
dmwamin	Minimum water release difference	$1e-1$
tdeep	(TODO: ???)	1880.0
dtdeep	(TODO: ???)	100.0
drdeep	(TODO: ???)	000.0
zdeep	(TODO: ???)	660000.0
vdeep	(TODO: ???)	670000.0
nudeep	(TODO: ???)	$1e+21$
dxwater	Fluid extension in x	$2e+3$
dywater	Fluid extension in y	$2e+3$
dzwater	Fluid extension in z	$2e+3$
maxwater	(TODO: ???)	$5e-1$
minmelt	(TODO: ???)	$1e-2$
maxmelt	(TODO: ???)	$1e-2$

Table 13: Hydration and melting parameters

3.3.7 Collision velocity parameters

Linearly change collision velocity. Use initial constant velocity set from boundary conditions. Between *timebeg* and *timeend* linearly change the collision velocity to the final velocity *velocitykf*.

Parameter	Description	Initial value
timebeg	Begin velocity change	$20e+6$
timeend	End velocity change	$25e+6$
velocitykf	Final collision velocity	0

Table 14: Collision velocity parameters

4 Raw output files: .prn

Valid for subduction/collision setup

4.1 Part I: General Information

Parameter	Description	Initial value	Unit
xnumx	Number of nodes in x	$16N + 5$ (4 multi-grid levels)	-
ynumy	Number of nodes in y	$16N + 5$ (4 multi-grid levels)	-
znumz	Number of nodes in z	$16N + 5$ (4 multi-grid levels)	-
mnumx	Number markers per cell in x	-	-
mnumy	Number markers per cell in y	-	-
mnumz	Number markers per cell in z	-	-
xsize	Dimension of model in x	-	$[m]$
ysize	Dimension of model in y	-	$[m]$
zsize	Dimension of model in z	-	$[m]$
pxinit	x-coordinate of initial pressure cell	$(xnumx - 1)/2$	-
pyinit	y-coordinate of initial pressure cell	$(ynumy - 1)/2$	-
pzinit	z-coordinate of initial pressure cell	-	-
pinit	Initial pressure in initial pressure cell	-	$[Pa]$
GXKOEFF	Gravitational acceleration in x	-	$[m/s^2]$
GYKOEFF	Gravitational acceleration in y	-	$[m/s^2]$
GZKOEFF	Gravitational acceleration in z	-	$[m/s^2]$
rocknum			
bondnum			
marknum			
n1			
timesum	Starting time	-	$[years]$
ival1			
gridcur			
gridtot			

Table 15: prn-file general information block

4.2 Part II: Rock type information

Parameter	Description	Unit
markn0	Individual lower viscosity limit	$[Pa * s]$
markn1	Individual upper viscosity limit	$[Pa * s]$
marks0	Individual lower stress limit	$[Pa]$
marks1	Individual upper stress limit	$[Pa]$
marknu	Newtonian viscosity	$[Pa^{MM} * s]$
markdh	Activation energy	$[J]$
markdv	Activation volume	$[J/bar]$
markss	Dislocation/diffusion creep stress threshold	$[Pa]$
markmm	Stress exponent	$(Power)$
markll	Pore fluid pressure factor	$(koef)$
marka0	Cohesion	$[Pa]$
marka1	Cohesion 2	$[Pa]$
markb0	Sine of dry friction angle	$[-]$
markb1	Sine of dry friction angle 2	$[-]$
marke0	Lower strain threshold for brittle/ductile transition	$[-]$
marke1	Upper strain threshold for brittle/ductile transition	$[-]$
markro	Density	$[kg/M^3]$
markbb	Density koef b	$[1/K]$
markaa	Density koef a	$[1/kbar]$
markcp	Heat capacity	$[J/kg]$
markkt	Thermal conductivity	$[W/(m * K)]$
markkf	Temperature dependency koef in conductivity	$[W/(m)]$
markkp	Pressure dependency koef in conductivity	$[Pa^{-1}]$
markht	heat generation	$[W/kg]$

Table 16: prn-file Rock type information block, see also tbl 3.

4.3 Part III: Nodes information

For each node the following parameters are given. The order is the following, where n is the number of nodes *nodenum*:

$$pr_0, vx_0, ...ht_0, pr_1, .., ht_1, .., pr_n, .., ht_n$$

Parameter	Description	Unit
pr	Pressure	$[Pa]$
vx	Velocity in x	$[m/s]$
vy	Velocity in y	$[m/s]$
vz	Velocity in z	$[m/s]$
ro	Density	$[kg/m^3]$
nu	Viscosity	$[Pa*s]$
nxx	normal viscosity	$[Pa*s]$
nxy	shear viscosity on z axis	$[Pa*s]$
nxz	shear stress on y axis	$[Pa*s]$
nyz	shear stress on x axis	$[Pa*s]$
tk	Temperature	$[K]$
cp	Heat capacity	$[J/kg]$
et	Thermal expansivity	$[1/K]$
kt	Thermal conductivity	$[Wt/m/K]$
ht	Heat sources	$[Wt/kg]$
rr	grain size	$[m]$

Table 17: prn-file Node information block

4.4 Part IV: Gridline positions

Position of gridlines in x direction $xnumx$ numbers are given

Position of gridlines in y direction $ynumy$ numbers are given

Position of gridlines in z direction $znumz$ numbers are given

4.5 Part V: Boundary Condition Equations

For each boundary condition there are 5 values needed. The total number of boundary conditions is *bondnum*. The boundary conditions are all of the following general form:

$$CURPAR = CONST + KOEF1 * PAR1$$

Parameter	Description
m2	Index in Indexmatrix bondm
m3	Index in BC-matrices, saved in bondm
bondv1[m3][0]	$CONST$ value in BC equ.
bondv1[m3][1]	$KOEF1$ value in BC equ.
bondn1[m3]	$PAR1+1$ value in BC equ., $PAR1 = 0$ means no boundary

Table 18: prn-file boundary condition information block

4.6 Part VI: Markers

4.6.1 Principal markers

The total number of markers is *marknum*. For each marker the following parameters is given:

name	type	size	description	note
markx	float	MAXMRK	x position	•
marky	float	MAXMRK	x position	•
markz	float	MAXMRK	x position	•
markw	float	MAXMRK	water percentage	•
markd	float	MAXMRK	density	•
marktm			•	•
markt	char	MAXMRK	rocktype in marker	•
markk	float	MAXMRK	temperature in K	
markv	float	MAXMRK	viscosity	•
marke	float	MAXMRK	strain at marker	!! ϵ not $\dot{\epsilon}$
markex	float	MAXMRK	melt fraction	•
markkt	double	MAXTMR	thermal conductivity	•
markkf	double	MAXTMR	thermal conductivity T coeff	•
markrr	double	MAXTMR	grain size	only in grain reduction version
markrn	double	MAXMRK	ductile viscosity ratio	only in grain reduction version
markkp	double	MAXTMR	thermal conductivity P coeff	•
markcp/markcp0	double	MAXTMR	thermal capacity	•
markaa	double	MAXTMR	first density parameter	
markbb/markbb0	double	MAXTMR	second density parameter	•

Table 19: prn-file Marker information block

4.6.2 Secondary markers

Those markers are not defined globally (i.e. head3mg.c) but are internal to functions in the code.

name	type	size	description	function
sigin	double	variable	stress rate invariant	viscalc()
epsin	double	variable	stress rate invariant	viscalc()
marksii1	double	variable	stress rate invariant	movemark()
markeii1	double	variable	strain rate invariant	movemark()
markrii1	double	variable	strain rate	movemark()
markdis1	double	variable	dissipation energy ($\sigma_{ii} * \epsilon_{ii}$)	viscalc()
mtk	double	variable	conversion in double of markk	movemark()
mpb	double	variable	conversion in bar of pressure at markers	movemark()
mnu	double	variable	saved viscosity	movemark()
mro/mro0	double	variable	density	ronurecalc()
mcp/mcp0	double	variable	thermal capacity	ronurecalc()
mkt	double	variable	thermal conductivity	ronurecalc()
mht	double	variable	radiactive energy source	ronurecalc()
xmelt/xmelt0	double	variable	melt fraction with and without counting previous melt	meltpart0()

Table 20: prn-file Marker information block

5 Index

5.1 Function Index

functions in load3mg.c :

`int loadconf()` : load configuration files mode.t3c to variables. Has a binary and text option but only the binary option is used. Return the number of the input file: content of files.t3c-1. If the value is incorrect it is set to 0.
`void loader()` : load information from datafile to variables.
`void gridcheck()` : verify that the size of grid and number of markers are compatible with the calculation matrices size. Then create the different grid levels for the multi-grid method (variables `mgg*[multinum][mg*]`).
`void saver(int,int)` : save the variables to the output file. The input are the index of the cycle between each output files and the number of the output file.
`void ffscanf()` : read text (from configuration files) while omitting text and comments (sign "/" for comments in configuration file) and suppressing the last character (always space in C).

functions in heat3mg.c :

`void titerate(int m0)` : resolve temperature equation.
`double heaterr` : gives the left sides, right side or error of temperature equation for resolution.

functions in mark3mg.c :

this part of the code principally control the movement of marker and associated recalculations of parameters. In particular the water effects are implemented in it.
`void antigor(double mtk, double mpb long int mm1)` : Look if the marker is in the P,T stability field of antigorite (from Schmidt and Poli 1998) the marker rock type is changed from serpentinized mantle(13) to hydrated mantle(11). The effect is that it weaken the mantle.
`long int m[1,2,3]serch(double x)` : find the closest lower node to the position x using bisection method. 1,2,3 respectively does it for x,y,z. it respectively return m10,m20,m30 for x,y,z.
`void movemark()` : Principal function of mark3mg.c : it implement all the sub-steps.

- INCOMPLETE!!!
- run the melt extraction and the topography correction
- compute velocity, stress, strain and stress/strain invariants on marker for non water markers(<50) and external markers inside the grid ($\geq$ 100). use them to compute viscosity on markers

`void tdbasecalc1(double mpb, double mtk,int mm2, long int mm1, eps1)` : Use thermodynamic database for ρ and C_p calculation and save to eps1 ρ, $wa\%$, C_p, β (isentropic thermal compressibility), α (compressibility factor), and activation enthalpy. The input are mpb,mtk, rocktype(mm2), marker index(mm1), and eps1 pointer.
`long int m[1,2,3]serch(double x)` : find the closest lower node to the position x using bisection method. 1,2,3 respectively does it for x,y,z. it respectively return m10,m20,m30 for x,y,z.
`double denscalc1(double mtk, double mpb, int mm2, double *eps1)` : Calculate density for markers and return the result of calculation depending of whivh densimod we are in. This function only P,T dependant effect. The effect of the thermodynamic database and water is calculated later in `ronurecalc`.
`void prcalc1(double x, double y, double z, double *eps1, long int *wn1)` : Interpolate pressure from nodes to marker and return the value in `eps1[10]`.
`void melting1(double mtk, double mpb, long int mm1, double *eps1)` : Check marker rocktype and call function `meltpart11` define the melting effects on rock type. the result is saved in `eps1[21]`.
`void meltpart11(double mtk, double mpb, long int mm1, double *eps1)` : Calculate the defined solidus and liquidus based one rock type and litterature defined equations. The melt fraction and latent enthalpy are calculated and saved in `eps1[21]` and `eps1[22]`, respectively.
`void meltpart0(double mtk, double mpb, long int mm1,int mm2, double *eps1)` : Calculate the melt density heat capacity and thermal conductivity and save it to `eps1[20][22][23][25][26]`.
`void meltextract()` : Control melt markers extraction and accretion.
`void epscalc1(double x, double y, double z, int yn, double *eps1, long int *wn1)` : Interpolate exx, eyy, ezz, sxx, syy, szz, exy, sxy, exz, sxz, eyz, syz and pr from nodes and save them to `eps1[]`.
`void tkrecalc()` : Replace boundary conditions inside T_K.
`double s??calc(long int m1, long int m2, long int m3, double ynval, int mgi, double *eps1, long int *un1, double *ui1)` : Calculate ν_{eff}, σ?? and ϵ??. Depending on `ynval` value it is either return as output or is written to `ui1`.
`void tkcalc1(double x, double y, double z, double *eps1, long int *wn1,`

`double wi1)` : Interpolate temperature from nodes to markers and save the value in `eps1[]`.
`void ronurestrict(int mgi)` : Interpolate ρ and ν from grid level `mgi-1` to `mgi`.

5.2 Variables index

5.2.1 Computation variables

`mgi` : Index used to define the level in the multi-grid. It is used for rheological parameters calculation.
`gx/gy/gz[MAXXLN]` : position of nodes for the initial grid. This grid is not staggered (possible in other version?).
`mg*[multinum]` : vector of the number of nodes for the different grids. there is for finer level: $mgx \longmapsto x = xnumx, mgy \longmapsto y = ynumy, mgz \longmapsto z = znumz, mgn \longmapsto nodenum, mgp \longmapsto P = 0, mgs \longmapsto s = 1, mgxy \longmapsto x * y = xnumx * ynumy$. And $mgp[i] = mgp[i-1] + mgn[i-1]$, $mgs[i] = 2 * mgs[i-1]$, $mg(x-y-z)[i] = (mg(x-y-z)[i-1] - 5)/2 + 5$, $mgn[i] = \Sigma mg(x-y-z)[i]$.
`mgg*[multinum][mg*]` : position of nodes for the different grids. See `mg*` for more.
`double *kf` : size of cell for a define multigrid level.
`min**1[100] and max**1[100]` : maximum and minimum value for the given variable for a given grid.
`min** and max**` : maximum and minimum value for the given variable for all grids.
`maxvxyz` : maximum velocity of the previous step.
`markx/marky/markz[MAXMRK]` : position of the markers. Their position is randomized from the nodes position when they are initialized. The randomization intervals are defined by xnonstab/ynonstab/znonstab.
`tk[MAXNOD], tk0-tk1-tk2[MAXNN]` : matrix of temperatures. `tk0` is the previous timestep temperature.
`bondv1[MAXBON][2]` : matrix of boundary conditions. `bondv1[:][0]` correspond to $CONST$ and `bondv1[:][1]` correspond to KOEF1.
`bondn1[MAXBON]` : matrix of bondary parameter $PAR1$.It control what type of boundary condition we have. If $PAR1 = 0$ it mean we do not impose a boundary condition.
`double mpb` : pressure in bar. Is used for water transport.
`float td[351][351][15][3]` : vecotr of vector of thermodynamic database it is filed by the function `loadconf`.The two first vector are temperature and pressure dimension, respectively. The others corresponds to different variables. In the last vector array one is density, 2 enthalpie

`double tkstp, pbstp, tkmin, pbmin, tknum, pbnum` : steps of temperature and pressure of the thermodynamic database. Minimum of temperature and pressure of the thermodynamic database. Number of steps of the thermodynamic database.
`double mycel` : steps of temperature and pressure of the thermodynamic database. Minimum of temperature and pressure of the thermodynamic database. Number of steps of the thermodynamic database.

5.2.2 counters

Only global counters are listed in this section.

`n0` : global counter of the timesteps between each output file. goes from from 0 to `cycmax-1`. It is printed in the terminal as "KRUG : ".
`n1` : global counter that save the thread ID. Is used to the parallel part of the code (OpenMP).
`f0` : global counter of the output file written during the run.
`m1` : usally local node counter.
`mm1` : usally marker counter.
`mm2` : usally rocktype counter.
`markfluid` : number of fluid markers.
`m2prn` : counter of the V-cycles of the multi-grid method. Is printed in the terminal as "Cycles : ".
`m1,m2,m3,m4` : Nodes counter used in the codes. m1 is for x, m2 is for y, m3 is for z. All nodes parameters are vector. thus, m4 , the index of unknown is use to go along them. it is defined by : $m4 = ynum * xnum * m3 + ynum * m1 + m2$.
`mcmax1` : node counter. Is used to relate dimension counters to position in the variables vectors (vx,vy,vz,pr...)
`mcmax0` : degrees of freedom counter. Used to relate dimension counters to position in the sol0 vectors.
`nodenumX` : Correspond to the number of node time X. Example: `nodenum8=nodenum*8`.
`meltnum` : Melt counter.

5.2.3 Dump variables

Many variables are used for temporary storage of data in some of the code functions. This index shows to which global variables they correspond to facilitate the understanding of the

code.

`eps1[100]` : Buffer to store markers properties during markers loop. Each array correspond to a specific property.

0	1	2	3	4	5	6	7	8	9
e_{II}/tk0	tk1	nueff/tk	tk2	mexx	meyy	mezz	mexy	mexz	meyz
10									
mpb									
20	21	22	23		25	26			
std adia terme	xmelt	H_{latent}	ρ		Cp melt	Kt			
	41	42	43	44	45	46			
	mro	mwa	mcp	mbb	maa	mdhh			
				54	55	56	57	58	59
				msxx	msyy	mszz	msxy	msxz	msyz
60									
$\epsilon_{ii} * \sigma_{ii}$									

`wn1[500]` : used to store nodes to marker positions while looping through markers. wn1[0]=x,wn1[1]=y,wn1[2]=z.

6 Numerical problems and solutions

- *Program stops because pressure gets too high.*

  Solution: This might indicate an awkward or inappropriate model setup. Try to lower the two pressure penalty factors from table 8 to $[0.5, 0.3, 0.1, 0.001]$. If that doesn't help, try another model setup.

- *Timestep doesn't converge within the time-limit of the Brutus queue.*

  Solution: Use a longer queue, e.g. 24h-queue instead of the 8h-queue. If the problem still persists try one of the following:

  a) Increase continuity equation error limit DIVMIN in tbl. 11 to $3e{-}3$

  b) Decrease multinum1 and multinum2 in tbl. 8 to 3.

  c) Increase both pressure penalty factors in tbl. 8. Choose something along $[0.05, 0.11, 0.31, 0.41]$.

  d) Decrease continuity equation error limit DIVMIN in tbl. 11 to $1e{-}3$.

  e) Try the following configurations for the pressure penalty factors [p0koef,p1koef,p2koef,p3koef] = $[0.31, 1.00, 0, 0]$, $[0.41, 1.00, 0, 0]$, $[0.11, 0.50, 0, 0]$

  f) Make 4 v-cycles series. In tbl. 11 set multicyc = 4.

- *Unstable model, model goes into very fast convection*

  Solution: Try one of the following:

  a) Decrease maximum timestep *maxtkstep* from 500 to 100.

  b) Decrease maximum timestep *maxtmstep* from $5e3$ to $1e3$.

  c) Increase lower viscosity limit from 10^{18} to 10^{19}.

7 Usage of paraview

7.1 Conversion from raw output to Paraview files

prn2vtr

7.2 Visualisation with Paraview

7.2.1 Visualize composition

Use the following pipeline:

- Threshold filter:
  set upper and lower threshold to the same composition number. Don't set the 'All Scalars' option.
- Contour filter:
  Contour by composition. Choose x.001 as value for the isosurface.
- GenerateSurfaceNormals filter:
  This generates a polygonal mesh, which has a smoothing effect.