User's Manual for CitcomSVE v2.1

Shijie Zhong
Department of Physics
University of Colorado at Boulder
Boulder, Colorado 80309
USA

szhong@colorado.edu

June 28, 2022

Table of Content

1. Introduction

1.1. What CitcomSVE is used for

CitcomSVE is a finite element package for computing Earth's (or planetary) mantle deformation and stress in response to surface loads and tidal loads. The computational domain is the Earth's mantle that is considered as an either incompressible or compressible viscoelastic medium. The surface loads that drive the deformation can be time-dependent growth and melting of ice sheets (or building-up of a shield volcano). The outputs of CitcomSVE include displacements, stress and strain rate in the mantle, and surface displacements and gravity anomalies. CitcomSVE can be used for a global scale problem in a spherical shell geometry or for a 3-D regional scale problem in either Cartesian or spherical geometry. CitcomSVE utilizes MPI for parallel computing and works on massively parallel computers with 1000's CPU cores.

1.2. History

CitcomSVE is developed from a finite element package CitcomS for mantle convection [Zhong et al., 2000; 2008]. The key difference between CitcomSVE and CitcomS is the mantle deformational property. CitcomS considers the mantle as a purely viscous flow medium, while CitcomSVE treats the mantle as a viscoelastic medium. To incorporate elastic deformation, CitcomSVE implements a Lagrangian, deformable grid, while a Eulerian, fixed grid is used in CitcomS. However, CitcomSVE inherits the same finite element formulation, full multi-grid solver of the linear system of equation, and parallel computing strategies as CitcomS.

The first version of CitcomSVE was developed and used in Zhong et al. [2003] for modeling glacial isostatic adjustment process with laterally varying lithospheric thickness. Subsequently, Paulson et al. [2005] added solutions of sea level equation, polar wander effect and apparent motion of center-of-mass to CitcomSVE, together with 3-D mantle viscosity structure derived from seismic structure. A et al. [2013] further expanded the formulation to include mantle compressibility in CitcomSVE. Zhong et al., [2012] and Qin et al. [2014] implemented tidal-rotational potential forcing to the code. This line of CitcomSVE was developed out of the original version of CitcomS [Zhong et al., 2000] that permitted domain decomposition of the mantle into 12 blocks in horizontal directions for parallel computing purpose, although arbitrary domain divisions were permissible in the radial direction. As a result, CitcomSVE was only used for 12 CPUs in computations (Zhong et al. [2003], and A et al. [2013]). As CitcomS became one of publicly available community codes for mantle convection at CIG and as parallel computers became more powerful, the domain decomposition for CitcomS was improved such that more than 12 CPUs can be used in horizontal directions and that 1000's of CPUs can be used in CitcomS (e.g., Zhong et al. [2008]).

A recent effort has been to re-develop and update CitcomSVE using the newer version of CitcomS (version 3.1) such that CitcomSVE can also use 1000's of CPUs [Zhong et al., 2022]. This document describes this updated CitcomSVE (i.e., version 2) that you have downloaded from github. As to the spring of 2022, CitcomSVE version 2 only works for an incompressible

mantle on a global scale, and future versions will include models for compressible mantle and also on a regional scale.

1.3. Governing equations, numerical method and grids

In CitcomSVE, the mantle is considered as a viscoelastic medium with Maxwell rheology (i.e., a spring and dashpot connected in series). The mantle can be treated as either incompressible or compressible, which will lead to a slightly different governing equations [e.g., Zhong et al., 2003; A et al., 2013]. The governing equations include the conservation law of the momentum (i.e., the equation of the motion), a Poisson's equation for gravitational potential perturbation and a rheological equation between stress and displacement [e.g., A et al., 2013]. For an incompressible medium, an additional equation is the conservation of the mass [e.g., Zhong et al., 2003; 2022].

For details of the governing equations, their finite element analysis and solutions, and calculations for sea-level equations, polar wander, and apparent motion of center-of-mass, readers can find them in Zhong et al. [2022] that provides a comprehensive summary of the developmental work of CitcomSVE in Zhong et al. [2003], Paulson et al. [2005] and A et al. [2013]. For formulation of a compressible mantle, see A et al., [2013]. For 3-D models of a regional scale, see Zhong and Watts [2013] and Bellas and Zhong [2021].

It is helpful to briefly describe the computational domain, domain decomposition and finite grids used in CitcomSVE or CitcomS for the global scale model. The mantle of a spherical shell is divided into 12 spherical blocks or caps, and each cap is further divided into finite elements or grids of similar sizes (Fig. 1, Zhong et al., [2022, 2000]). Each of the 12 caps can be further decomposed into smaller blocks in horizontal and radial directions for parallel computing purpose with each block for a CPU core. Message passing interface (MPI) routine calls were implemented in CitcomS/CitcomSVE for communications between different blocks or CPU cores for parallel computing [Zhong et al., 1998; 2000]. CitcomS/CitcomSVE uses 8-node tri-linear elements, inherited from the original Cartesian Citcom code [Moresi and Gurnis, 1996], which leads to solutions of displacement (or velocity) field to be second-order accurate, as confirmed by error convergence analyses for CitcomS [Zhong et al., 2008] and CitcomSVE [Zhong et al., 2022].

1.4. Acknowledgement, Citations and User's support.

Some of the important development of CitcomSVE has been done by former graduate students in Shijie Zhong's group: Archie Paulson and Geruo A. CitcomSVE's development had also been benefitted from the guidance and inspiration of Late Professor John Wahr. NSF, NASA and Packard Foundation have provided financial support for the development of CitcomSVE. The development of CitcomS from the original Citcom code that works for a Cartesian box on a single CPU computer [Moresi and Gurnis, 1996] was benefited from help by Louis Moresi, Michael Gurnis, and Eh Tan [Zhong et al., 2000; 2008].

Users of CitcomSVE may cite Zhong et al., [2003, 2022] as a general reference. To use CitcomSVE for a compressible mantle and sea-level calculations, users may cite A et al. [2013]

and Paulson et al. [2005], respectively. If you have questions on CitcomSVE or would like to report a bug, you may contact Shijie Zhong at szhong@colorado.edu.

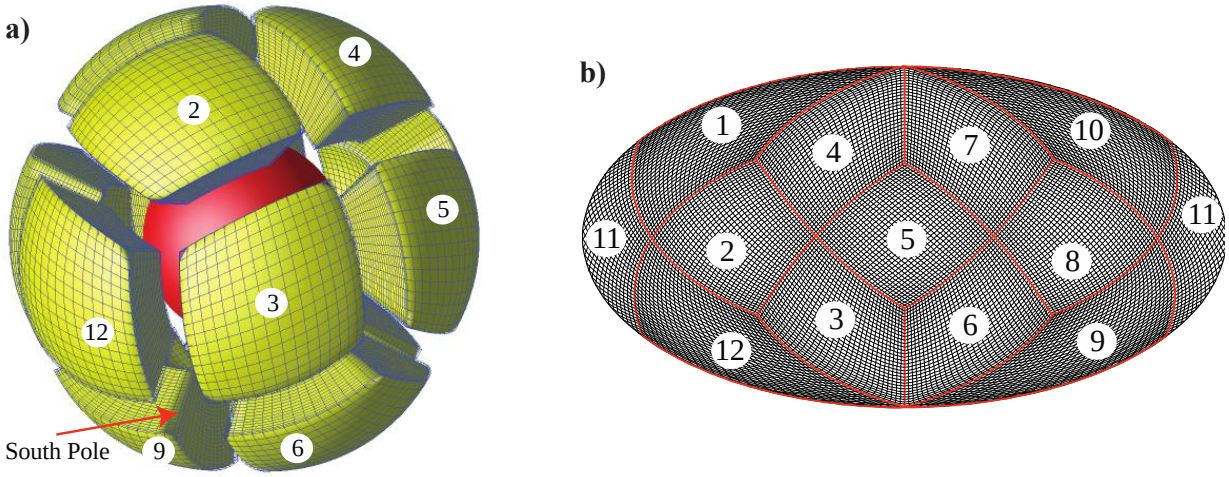

Figure 1. A 3-D view (a) and mapview (b) of the finite element grid with 12 spherical caps covering a spherical shell used in CitcomSVE. Each of the 12 caps in Figure 1a and 1b includes a certain number of finite elements in the radial direction and two azimuthal directions, and each cap can also be further divided into smaller blocks for a parallel computing purpose. The thick red lines in Figure 1b mark the shared boundaries of the 12 caps. In Figure 1a, the caps are separated for a better view. The numbers in Figure 1a and 1b show the ordering of the 12 caps. Figure 1a and 1b are modified from Zhong et al., [2022, 2000].

2. Installation of CitcomSVE

CitcomS/CitcomSVE has been tested and used successfully on many different Unix-like computer platforms. CitcomSVE is developed out of CitcomS 3.1, and installation process is the same as that for CitcomS. Here, a brief description is given on the installation procedures, and it should be sufficient for most users. In case you would like to know more about the installation procedures, you can read User's manual on CitcomS.

To use CitcomSVE effectively, you may want to work on a parallel computer, either a PC-Cluster or a supercomputer. The only system requirements for your computer are: 1) a C compiler and 2) a MPI library. Most Unix-like computers have them installed already. If not sure, you can ask your computer system administrator.

Here are the three steps to install CitcomSVE on your computer.

1) Download CitcomSVE package from https://github.com/shjzhong/CitcomSVE. You will need to download ***three compressed files***: CitcomSVE-2.1.tar.gz, ICE6G_1x1.tar.gz, and Compare_Cookbook.tar.gz. Make a new project directory, move CitcomSVE-2.1.tar.gz to the new directory, and unpack it first. To unpack, use the tar command:

```
$ tar xzf CitcomSVE-2.1.tar.gz
```

The source codes and all the other files in CitcomSVE package are in directory *CitcomSVE-2.1*.

2) Go to directory *CitcomSVE-2.1* with CitcomSVE source code, configure the compiler, and generate executables.

```
$ cd CitcomSVE-2.1
$ ./configure -- without-pyre
$ make
```

An executable CitcomSFull is generated in *CitcomSVE-2.1/bin*, and it will be used for modeling calculations. Notice that three directories under *CitcomSVE-2.1* are quite useful: *lib, bin* and *DATA*. Most of the source codes (i.e., the C subroutines and head files) are in *CitcomSVE-2.1/lib*, while several others are in *CitcomSVE-2.1/bin*, together with executable CitcomSFull. *CitcomSVE-2.1/DATA* stores some model input files.

3) Move the other two compressed files ICE6G_1x1.tar.gz and Compare_Cookbook.tar.gz to *CitcomSVE-2.1/DATA* and unpack them both. In directory *CitcomSVE-2.1/DATA*, execute these two commands:

```
$ tar xzf ICE6G_1x1.tar.gz
$ tar xzf Compare_Cookbook.tar.gz
```

Two new directories are now generated: *CitcomSVE-2.1/DATA/ICE6G_1x1 and CitcomSVE-2.1/DATA/Compare_Cookbook.* The first directory stores relevant files for ice history model ICE6G and the second one contains files for cookbook example cases. The files from both the directories will be discussed later. Note that three compressed files are given to reduce the file size to fit the limit of 25 Mbytes set by github.

3. Running CitcomSVE

Running CitcomSVE would require input files for model calculations, which can vary depending on the models. How to run CitcomSVE also depends on the computer systems you use. The description here focuses on running CitcomSVE on a PC-cluster using mpirun command (A supercomputer center may only allow a user to run jobs in a batch mode, and ask a system administrator to help with the batch mode). The example model calculation is for a glacial isostatic adjustment (GIA) modeling with an ice history model (i.e., ICE 6G) and 1-D viscosity structure (i.e., VM5a). It is case GIA_R2 in Zhong et al., [2022] (see Table 4 of this paper).

Here is an example command line for running executable CitcomSFull on a PC cluster.

Go to directory *CitcomSVE-2.1/DATA* with input files including *inputfileVM5a1* and *machinefile0*.

```
$ mpirun -np 96 -nolocal -machinefile machinefile0 ../bin/CitcomSFull inputfileVM5a1
```

Note that "-np 96" indicates that for this run, the number of processors or cores (i.e., np) to be used is 96, and "-machinefile machinefile0" indicates that the run will use compute nodes of the PC cluster that is listed in file *machinefile0*. File *machinefile0* lists 24 compute nodes of the PC-Cluster and contains the follow 24 lines:

node01
node02
…
node24

This example run uses 96 processors in total on 24 compute nodes. Therefore, 4 processors from each of the 24 compute nodes (from node01 to node24) as listed in file *machinefile0* is used for computations. Note that each compute node of this PC-cluster has 4 processors or cores. The number of processors used for each model run is specified in input file *inputfileVM5a1*, for example,

Input file *inputfileVM5a1* contains all the important information about this model including grid, mantle parameters, ice history models, …, all of which will be discussed in detail in next section.

This mpirun command, once issued, will start the CitcomSVE run. Output files (e.g., for surface vertical displacements and gravitational potential) are to be generated and stored during the model run in directories that you specify in *inputfileVM5a1*.

4. Input files

Two types of input files are needed for computing GIA models using CitcomSVE: 1) a general input file (e.g., inputfile1 in mpirun command line in the last section) and 2) files for ice history models (e.g., ICE6G). This section describes these two types of input files. Note that CitcomSVE package that you have downloaded from gitbub contains sample input files like *inputfileVM5a1* and also ICE6G.

4.1. The input file in the mpirun command line

The input file in mpirun command line (e.g., *inputfileVM5a1*) provides the most essential information for a CitcomSVE model run. This sub-section describes an example of such input file *inputfileVM5a1*. The lines (with italic font) from *inputfileVM5a1* are listed first, and explanation to these lines is provided using regular font.

datadir="BM_ICE6G"
datafile="case1"

BM_ICE6G is the directory in which output files are stored, and it is under directory *DATA*. Parameter *datafile* (i.e., *case1*) specifies the prefix of output file names. For example, output log

file will be *DATA/BM_ICE6G/case1.log*. Directory *BM_ICE6G* should be created in directory *DATA* before the model run starts.

```
apply_SLE=1
polar_wander=1
polar_wander_kf=1.11664062
```

Parameter *apply_SLE* determines if the sea-level equation is solved (1 means yes, and 0 is no). Parameter *polar_wander* determines if polar wander effect is included (1 means yes, and 0 is no). Parameter *polar_wander_kf* specifies degree-2 fluid Love number used for calculations. For GIA problems, both *apply_SLE* and *polar_wander* should be set as 1.

```
# only for Heaviside=1 or step function loading
Heaviside=2    # 1: Heaviside   2: ice model e.g., ice6g
```

Parameter *Heaviside* determines loading time history. *Heaviside=1* specifies a step-function in time (i.e., the Heaviside function), and *Heaviside=2* is for more complicate loading history, e.g., ICE6G, for which more information is needed as dicussed below

```
apply_potential=0      #0: apply surface loads, or 1:apply tidal potential
```

Parameter *apply_potential=0* is for surface loading such as ICE6G loading, and *apply_potential=1* specifies tidal potential loading, which is often used together with *Heaviside=1*.

```
# Loading harmonics and amplitude for parameter Heaviside=1
perturbmag=0.0        # e.g., 1e-6
perturbl=2
perturbm=0
```

These three lines set the loading harmonic degree *l* in *perturbl* and order *m* in *perturbm* with loading amplitude in *perturbmag*, for single harmonic loading with *Heaviside=1*. Note that the amplitude is normalized. For surface loading, the amplitude is the normalized height of the load with the density of the mantle (normalized by the radius of the planet). For tidal potential loading, the amplitude is the normalized potential (see Zhong et al., [2022] on how the potential is normalized).

```
# only useful for Heaviside=2 of some ice model
load_stages_time_file="ice6g_26ka_stages_timeA.dat"
ice_datafile="ICE6G_1x1/ice6g"
ocean_datafile="ICE6G_1x1/ocn6g"
nlong=361
nlati=181
```

When *Heaviside=2*, i.e., for ice loading, *load_stages_time_file* specifies the file (i.e., *ice6g_26ka_stages_timeA.dat*) that determines the loading history (see section 4.3 for

description of this file). *ice_datafile and ocean_datafile* specify the prefixes of files for ice history and ocean function, respectively (see section 4.2 for description). Parameters *nlong* and *nlati* determine the number of grid points of the input ice history and ocean function files in longitudinal and latitudinal directions, respectively. For example, the input ice history and ocean functions are given on 1° by 1° grids, so *nlong=361* and *nlati=181*. Note that these grids are different from CitcomSVE's finite element grids (Fig. 1), and CitcomSVE code reads the ice history and ocean functions on these regular grids and interpolate them to finite element grids. Also note that it is important to unpack compressed *ICE6G_1x1.tar.gz* in directory *CitcomSVE-2.1/DATA*, as discussed in section 2.

storage_spacing=10 # write data every ...

Parameter *storage_spacing=10* means that the results of surface displacements and gravity anomalies will be output every 10 time-steps. Note that viscosity and stress fields which are in 3-D and take more disk space will be output less frequently and which can be customized. Also, note that the final time-step results are always output.

ll_max=32
output_ll_max=32

Parameter *ll_max=32* sets the highest number of spherical harmonic degree and order (i.e., 32 here in this example) with which the gravitational potential anomalies are computed. Parameter *output_ll_max=32* is that for the outputs. In most cases, these two parameters are the same.

 # for viscosity and rheology

3d_visc_from_file=0 #0 for 1d layered viscosity; 1 for 3d visc from files
3d_visc_datafile="ViscA/ViscA2x2.dat"

Parameter *3d_visc_from_file* determines the viscosity structure used in the model calculation. *3d_visc_from_file=0* is for 1-D layered viscosity, while *3d_visc_from_file=1* is for 3-D viscosity structure. For the latter case, parameter *3d_visc_datafile="ViscA/ViscA2x2.dat"* specifies the file in which 3-D viscosity structure is given. See section 4.4 for description of the viscosity file.

 # number of the material groups with different viscosity or elastic parameter
num_mat=6
shearModulus=1.0,1.0,1.0,1.0,1.0,1.0

Parameter *num_mat* determines the number of material groups for the model. For example, *num_mat=6* means that the model has 6 different material groups. The definitions of each material group start with its shear modulus (the normalized value). For example, *shearModulus=1.0,1.0,1.0,1.0,1.0,1.0* specifies shear modulus for groups 1, 2, … and 6, respectively (all is 1 in this case or uniform shear modulus).

 # for 1d layered viscosity, i.e., only applicable when 3d_visc_from_file=0
visc0=3.0951e0,1.5048e0,4.8531e-1,4.8531e-1,1.0e1,1.0e5

Parameter *visc0* specifies normalized viscosity for each of the group from group 1 to 2, … and 6, respectively. Note that groups 3 and 4 here have the same viscosity and shear modulus. Note that this would be ignored if *3d_visc_from_file=1,* i.e., when 3-D viscosity is used.

VMIN=on
visc_min=1.0e-3
VMAX=on
visc_max=1.0e05

Parameter *VMIN* and *visc_min* specify the minimum viscosity cutoff (i.e., 10^{-3} in this example), and *VMIN* and *visc_min* give the maximum viscosity cutoff (i.e., 10^{5} here), both for normalized viscosity).

sdepv_misfit=3.e-3 # relative change cutoff for self-grav iteration

Parameter *sdepv_misfit* sets iteration tolerance or accuracy for stress-dependent viscosity iteration (when the stress-dependent viscosity is applied) and for self-gravitation iteration. *sdepv_misfit=3.e-3* means that the iteration ends when relative change of displacement field between two consecutive iterations is less than 0.3%.

SDEPV=off   # on for stress-dependent viscosity
sdepv_expt=1.0,1.0,1.0,1.0,1.0,1.0
sdepv_trns=1.0e6,1.0e6,1.0e6,1.0e6,1.0e6,1.0e6
sdepv_bg=1.0e-6,1.0e-6,1.0e-6,1.0e-6,1.0e-6,1.0e-6
sdepv_relax_alpha=0.75

The above segment is for stress-dependent viscosity. Parameter *SDEPV* sets on and off stress-dependent viscosity. When *SDEPV=off*, the subsequent 4 parameters will be ignored. Parameter specify exponents *n* of stress-dependent viscosity, transition stress and background stress for each material group. Note that these two stresses are in SI units or Pa. Note that n=1 (i.e., Newtonian viscosity) can be assigned to one or more groups, when n=3.5 is assigned to other groups. Parameter *sdepv_relax_alpha* is used as a relaxation parameter for stress-dependent viscosity iteration and it should be between 0 and 1 with 1 meaning no relaxation. In this example, *sdepv_relax_alpha=0.75*.

Note that so far we have not defined where each material group is in the model. We will do just that after the coordinates/geometry/grid of the models are defined.

 # related to parallel processors, grids, and mesh
nproc_surf=12
nprocx=2
nprocy=2
nprocz=2
These four lines set the parallel processors or cores to be used for this case. *nproc_surf=12* indicates 12 spherical caps in the global version CitcomSVE on which the finite element grid is built (Fig. 1). Parameters *nprocx, nprocy, and nprocz* indicate the numbers of processors or cores

that each of the cap is further divided and assigned in each of two horizontal directions and radial direction, respectively. In this example case, they are all equal to 2. Therefore, the total number of processors or cores used for this example case run is the product of these four numbers, *nproc_surf*nprocx*nprocy*nprocz* or 96, which should be the same as -np in the mpirun command line that we discussed before.

```
nodex=81      # each cap, no of nodes in one horizontal dirc
nodey=81      # each cap, no of nodes in the other horizontal dirc
nodez=49      # each cap, no of nodes in the radial dirc
```

Parameters *nodex, nodey and nodez* specify the number of grid points in each of two horizontal directions and radial direction, respectively for each of the spherical caps. Note that in the way that the grid is designed in CitcomSVE (Fig. 1), *nodex* is always equal to *nodey*, and they are the same for all the 12 caps.

```
mgunitx=5     # multi-grid base level, no of element
mgunity=5
mgunitz=3
levels=4      # multiplying by 2^(levels-1)
```

These four parameters set up a nested grid for the multi-grid solver of the matrix equation resulting from the finite element discretization of the governing equations. These parameters determine the total number of grid points for the computational domain of each processor. Parameters *mgunitx, mgunity and mgunitz* specify the number of elements in each of two horizontal directions and radial direction, respectively, at the base (i.e., the coarsest grid) level of the nested grid for each domain (or processor). Parameter *levels* indicates the number of levels of the nested grid. Therefore, the number of elements in each direction for each domain (or processor) is, taken x direction as an example, $mgunitx*2^{(levels-1)}$, or 40 in this example case. Therefore, the grid for each domain in this example case is 40*40*24, and this is consistent with the number of processors (e.g., *nprocx=2*) and the number of nodes (e.g., *nodex=81*) in each direction for a spherical cap (note that the number of nodes in each direction is the number of elements plus 1, given that an 8-node brick element is used in CitcomSVE).

```
# model domain definition
radius_inner=0.55
radius_outer=1.0
```

Parameters *radius_inner* and *radius_outer* are dimensionless radii of the core-mantle boundary and the surface, respectively, both normalized by the radius of the Earth.

```
# grid radial layers
r_grid_layers=6
rr=0.55,0.81632653,0.894819466,0.9356357928,0.984301413,1.0
nr=1,20,30,36,44,49
```

These three lines set up radial coordinates of each grid points. Parameter *r_grid_layers* specifies the number of radial grid layers (i.e., *r_grid_layers-1* or 5 in this example case) with each grid layer that can have a different resolution (for the purpose of grid refinement). Parameters *rr* and *nr* determine, starting from the bottom (or inner radius) and up to the surface, radial coordinate and radial nodal index for each grid layer.

```
# material group in layers
num_mat_layer=6  # number of the layers with diff. visc. or elastic parameter
n_mat_layer=1,20,30,36,44,46
```

Parameter *num_mat_layer* specifies the number of radial layers of material groups, and each material group can have different material properties (e.g., viscosity and shear modulus). Note that *num_mat_layer* should be equal to or less than *num_mat* which was discussed earlier. Parameter *n_mat_layer* gives, starting from the bottom, radial nodal index of each layer's starting node (e.g., for the surface lithosphere layer or 6^{th} layer in this example case, the starting node is 46^{th} node). This list should be consistent with the definitions of grid layers (i.e., *rr* and *nr*) and material properties (i.e., *visc0, shearModulus, …*).

```
# Dimensional Information
radius=6370e3
density=4604.4
shearmodulus=1.4305e11
density_above=0.0
density_below=10005.4
gravacc=9.8
refvisc=1e21
```

This set of parameters are the reference values all in SI unit that are used to normalize the model parameters. Note that this example case is for incompressible mantle with uniform density.

```
    # Solver iteration parameters
accuracy=3.0e-3       # dv/v and dp/p error tolerance
tole_compressibility=1e-8      # div(v)/v error tolerance
```

These two parameters specify solver iteration accuracy or tolerance level (i.e., *dv/v* and *dp/p*, where *v* and *p* are displacement and pressure fields, respectively. Pressure *p* is only relevant for an incompressible medium). Parameter *tole_compressibility* set the tolerance level for incompressibility iteration.

The rest of the parameters in the input file rarely need any changes.

4.2. The input files for loading

For any loading calculation, CitcomSVE requires at least one input file that specify loading history. For loading with complex temporal and spatial distributions of loads (e.g., glaciation/deglaciation, ICE-6G), additional input files for the spatial distributions of loads at different times are also needed. Here we describe these files' formats so that you can build your own loading history files for your own loading models.

4.2.1. Loading history file

Here we first discuss the loading history file, using the same ICE6G/VM5a calculation we discussed in section 4.1 as an example. For this example model run, its mpirun command line input file *inputfileVM5a1* specifies the loading history file (see page 6) as *ice6g_26ka_stages_timeA.dat* by *load_stages_time_file="ice6g_26ka_stages_timeA.dat"*.

This model run covers the Earth's last 122 kyrs of glaciation and deglaciation history that is divided into 49 stages. Here are the first few lines and the final two lines in *ice6g_26ka_stages_timeA.dat* that specify the loading history.

```
49 1
-122.0 180
-104.0 780
-26.0  10
-25.0  10
…
-0.5  5
0.0  0
```

In line 1, the first number indicates the number of loading stages (i.e., *49*), and the second number indicates the unit of the time in subsequent lines. If the second number is "*1*" (as in this file), it means that the time is given in terms of 1000 years; if the number is "*0*", it means that the time is given in terms of Maxwell times (i.e., dimensionless).

In line 2, the first number (i.e., *-122.0*) represents the beginning time (i.e., 122 kyrs ago, and the negative sign means the past) of the 1st stage, and the second number (i.e., *180*) specifies the number of time steps to be used to cover this stage.

In line 3, the first number (i.e., *-104.0*) represents the beginning time of the 2nd stage (i.e., 104 kyrs ago) and also the ending time of the 1st stage. This means that the 1st stage lasts for 18 kyrs. And again the second number (i.e., *780*) specifies the number of time steps for the 2nd stage.

This pattern continues to the final stage or 49th stage of this example case that is described by the final two lines in this file. The final stage starts at 0.5 kyrs ago and ends at the present-day (i.e., time 0) with 5 time steps.You may have noticed that for this example case, time increment is 100 years for the entire loading history, and the total number of time steps of the model run is 1220.

As a quick reference, here is a sample loading history file for single harmonic loading calculation with a Heaviside time function (i.e., 1 stage), *heaviside1_stages_time.dat.*

```
1 0
```

-40 200
0.0 0

Based on our description above, this file specifies the model calculation with 1 stage loading that starts at 40 Maxwell time ago and ends at time 0 with a total of 200 time steps.

4.2.2. Load distribution files

Load distribution files are only needed for loading calculations with complex load distributions (i.e., non-single harmonic) in space with *Heaviside=2* as in this example calculation with ICE6G of 49 loading stage. Basically, for each stage, one load distribution file is needed. In directory *ICE6G_1x1* which was generated from one of the downloaded compressed file (see section 2), one may find 49 such files that describe ice distribution on 1^o by 1^o uniform Gauss-Legendre grid: *ice6g180x360.1*, *ice6g180x360.2*, …, and *ice6g180x360.49*, with the final number in the file name indicating the stage number. CitcomSVE will read these ice distribution files and interpolate the load to the finite element grid. For every stage with an ice distribution file, there is also an accompanying ocean function file, e.g., *ocn6g180x360.1*, that describes where the oceans are.

The following 4 lines in input file *inputfileVM5a1* specify the locations of the ice distribution and ocean function files and grid sizes of these files (i.e., 1^o by 1^o).
ice_datafile="ICE6G_1x1/ice6g"
ocean_datafile="ICE6G_1x1/ocn6g"
nlong=361
nlati=181

These ice distribution files have certain format. For example, here are the beginning and ending lines, and also lines where the latitude changes in *ice6g180x360.1:*

5.0000e-01 8.9500e+01 0.00000e+00
1.5000e+00 8.9500e+01 0.00000e+00
2.5000e+00 8.9500e+01 0.00000e+00
…
3.5850e+02 8.9500e+01 0.00000e+00
3.5950e+02 8.9500e+01 0.00000e+00
5.0000e-01 8.8500e+01 0.00000e+00
1.5000e+00 8.8500e+01 0.00000e+00
…
3.5750e+02 -8.9500e+01 2.85275e+03
3.5850e+02 -8.9500e+01 2.87000e+03
3.5950e+02 -8.9500e+01 2.87000e+03

Each line is for a point on the grid and the 1st, 2nd and 3rd columns are for longitude, latitude and ice height in meters, respectively. It starts from the North Pole region with increasing longitude from 0.5^o to 359.5^o for a fixed latitude of 89.5^o, and then it goes to next latitude of 88.5^o. It continues until reaching to the South Pole region with latitude of -89.5^o.

The ocean function files have the same format as the ice files. The 3rd column is 1 for oceanic region and 0 for land. Admittedly, given that the ice files and ocean function files are on the same 1° by 1° grid, it could have been more economical to only store the 3rd column data in each file.

4.3. The input file for 3-D viscosity

For 1-D or uniform viscosity model calculations, the viscosity structure can be specified in *mpirun* command line input file (e.g., *inputfileVM5a1*). For models with 3-D viscosity, users will need to set parameter *3d_visc_from_file* to be 1 and put 3-D viscosity in a file for CitcomSVE to read from. For example, set in the command line input file as the following:

```
3d_visc_from_file=1     #0  for 1d layered viscosity; 1 for 3d visc from files
3d_visc_datafile="ViscA/ViscA2x2.dat"
```

where *ViscA/ViscA2x2.dat* is the file for 3-D viscosity. In this example 3-D viscosity file, 3-D viscosity is given on a 2° by 2° grid at different depths. Note that the grid for 3-D viscosity structure is completely independent from the 3-D finite element grid in CitcomSVE. CitcomSVE reads in the 3-D viscosity and interpolates it to the finite element grid. Although the file is almost self-explanatory, here are some descriptions of the file.

```
#The 3D viscosity model has 48 radial layers and is on 2 degree by 2 degree grid.
48 180 90
```

The first line informs general structure of the viscosity file: 48 radial layers and 2° by 2° grid. The three numbers in the second line (i.e., 48, 180, 90) specify the numbers of radial layers, longitude cells, and co-latitude cells, respectively. In this case, the latter two numbers correspond to 2° by 2° grid.

```
#The following 48 lines give depth (in km) of viscosity layers.
2.81160e+03
2.70181e+03
2.59202e+03
…
5.45454e+01
3.27272e+01
1.09090e+01
```

The first line of this segment describes the content of this segment (i.e., depth of each layer). The subsequent 48 lines give the depth of each viscosity layer from the bottom most layer to the topmost layer. The depth interval can be arbitrary and does not need to uniform.

```
#The following 180 lines give longitudes of viscosity grid.
1.00000e+00
3.00000e+00
5.00000e+00
```

…
3.57000e+02
3.59000e+02

The first line of this segment describes the content of this segment (i.e., longitudes of the viscosity grid). The subsequent 180 lines define the longitude grid of the viscosity grid, from 1^{o} to 359^{o} with increment of 2^{o}.

#The following 90 lines give latitudes of viscosity grid.
8.90000e+01
8.70000e+01
8.50000e+01
…
-8.70000e+01
-8.90000e+01

The first line of this segment describes the content of this segment (i.e., co-latitudes of the viscosity grid). The subsequent 90 lines define the co-latitude grid of the viscosity grid, from 89^{o} (near the North Pole) to -89^{o} (near the South Pole) with increment of 2^{o}.

#For layer 1 at depth 2.8116e+03 km depth.
8.82480e+19
8.78593e+19
8.77908e+19
…
#For layer 2 at depth 2.7018e+03 km depth.
6.48673e+20
6.45041e+20
6.44448e+20
…
#For layer 48 at depth 1.0909e+01 km depth.
9.42349e+24
9.34908e+24
9.34426e+24
…

Each segment specifies viscosity of a layer on a 2^{o} by 2^{o} grid, starting from the bottom most layer (i.e., layer 1) to the topmost layer (i.e., layer 48 in this example case). The viscosity is given in unit of Pas. For each layer, there should be 180x90=16,200 grid points with assigned viscosity. The ordering of the grid points is identical to that of ice model that was discussed in section 4.2.2, i.e., (1^{o}, 89^{o}N), (3^{o}, 89^{o}N), …, (359^{o}, 89^{o}N), (1^{o}, 87^{o}N), (3^{o}, 87^{o}N), …, (359^{o}, 87^{o}S), (1^{o}, 89^{o}S), (3^{o}, 89^{o}S), …, (359^{o}, 89^{o}S).

5. Output files

Model outputs from CitcomSVE of 3-component displacements, rates of displacements, gravitational potential anomalies and their rates, all on the surface, 2nd-invariant of stress and

viscosity throughout the mantle and their horizontal averages are stored in a number of different files. For the example model run with input file *inputfileVM5a1*, these files are in directory *BM_ICE6G*, and are *case1.tps_sharm.xx*, *case1.vtps_sharm.x*, *case1.pttl_sharm.x*, *case1.pttldot_sharm.x*, *case1.topo_s.y.x*, *case1.stress_visc.y.x*, and *case1.horiz_ave.z.x*, where *x* is the index for time step (i.e., every 10 time steps as in *inputfileVM5a1*), *y* is the index for the CPU core (i.e., from 0 to 95, as the model run uses 96 CPU cores), and *z* is the index for the CPU core in vertical direction (i.e., from 0 to 1, as there are two CPU cores in vertical direction with *nprocz=2*). Coordinates of co-latitudes and longitudes for the finite element grid are stored in *case1.coord_s.y*, and radial coordinates can be found in *case1.horiz_ave.z.x*. Note that the displacements on the finite element nodes are relatively small, so we could use the initial coordinates of the nodes as that for all the time during the loading process.

File *case1.log* stores information on convergence, iterations, and CPU times.

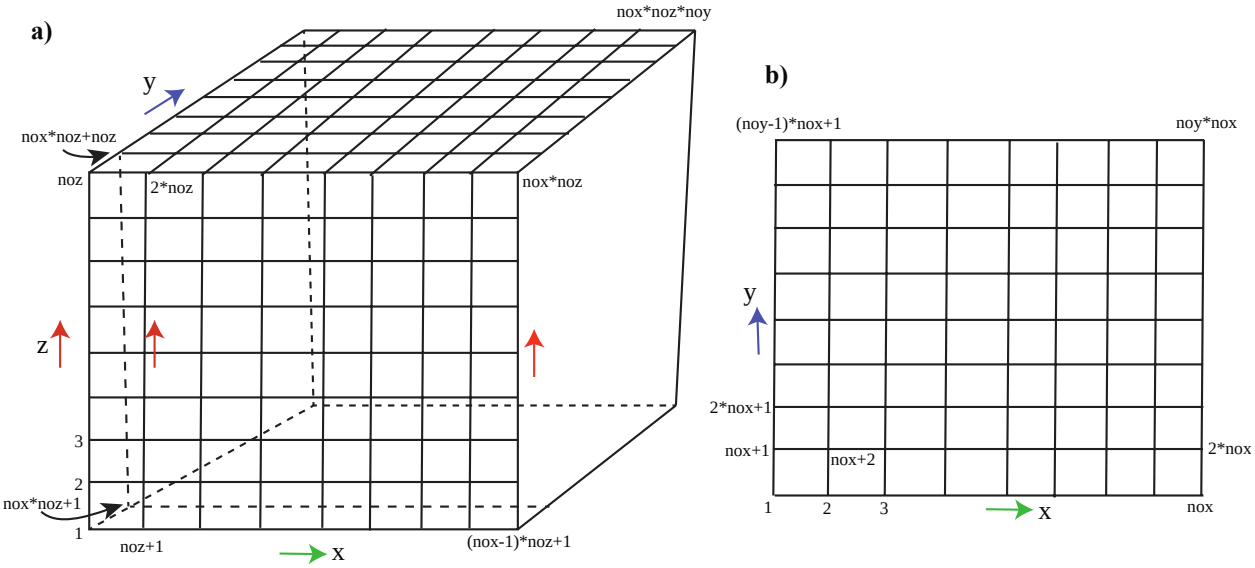

Figure 2. Node numbering for 3-D grid in a computational domain (an arbitrary geometry) (a) and node numbering for its surface grid (b) in CitcomSVE. For the 3-D grid, the numbers of nodes in x, y and z directions are *nox*, *noy* and *noz*, respectively. Therefore, the total number of nodes for this 3-D domain is *nox*noy*noz*, while the number of nodes for the surface is *nox*noy*. In the 3-D domain, the node numbering increases the fastest in *z* or radial direction, the second fastest in *x* or one of the horizontal directions, and the slowest in *y* or the other horizontal direction. For the surface nodes, the node numbering increases the first in *x* direction and then in *y* direction.

5.1. Displacements, gravity anomalies, stress and viscosity on finite element grid

Displacements and gravitational potential anomalies are only output for the surface in files *case1.topo_s.y.x* at a frequency as specified by *storage_spacing=10* (i.e., every 10 time steps in this case) in *inputfileVM5a1*, while 2nd invariant of stress and viscosity are output in files *case1.stress_visc.y.x* at a frequency of 5 times of *storage_spacing* (i.e., every 50 time steps) as set

in the code. These outputs are produced on nodes of the finite element grid of CitcomSVE for each computational domain (or each CPU core) (Fig. 1 and 2).

Files *case1.topo_s.y.x* are only produced for CPU cores or sub-domains that contain the surface. Therefore, for the example model run with a total of 96 CPU cores with 2 in radial direction (i.e., *nprocz=2*), there are 48 files *case1.topo_s.y.x* with CPU core index *y* of 1, 3, …, and 95.

Here are the first several lines of *case1.topo_s.3.50,* one of the 48 files for time step 50.

00050 5.00000e+03 1.00000e+02 -1.09164e-05 2.47745e-05 -2.33696e-07 5.54058e-07
4.0833e-07 9.5711e-09 1.0206e-07 1.6080e-09 8.5591e-09 2.6311e-09 -1.7756e-07 -3.9690e-09
3.5307e-07 7.8751e-09 9.9309e-08 1.5804e-09 -5.8336e-09 2.3410e-09 -1.9663e-07 -4.4199e-09
…

where line 1 is mostly for information, and each of the subsequent lines is for a node on the surface and there are 41x41=1681 such lines or nodes on the surface, given that *nodex=81, nodey=81*, *nprocx=2 and nprocy=2* in *inputfileVM5a1* for this example case. In line 1, the first three numbers are for time step (i.e., 50), elapsed time in years (i.e., 5,000 years) and incremental time Δt in years (i.e., 100 years), respectively. The subsequent four numbers are for cumulative and incremental polar motions. In each of the subsequent lines, columns 1 to 8 are cumulative (i.e., topography) and incremental displacement in radial direction, total and incremental gravitational potential, cumulative and incremental displacements in co-latitude direction (the South is positive), and cumulative incremental displacements in longitude direction (the East is positive), respectively. Note that all the output quantities are dimensionless (see section 2.5 of Zhong et al., [2022] for normalizations of these variables. To determine the rate of displacement (i.e., velocity), one can simply divide incremental displacement by time increment or Δt. The incremental gravitational potential anomalies can be used to compute rate of gravity change (e.g., for comparing with GRACE data) by dividing time increment or Δt.

Files *case1.stress_visc.y.x* stores 2nd invariant of mantle stress and viscosity for domain or CPU core index *y* at time step *x*, i.e., these files are produced for each of the CPU cores or domains. For the example model run with a total of 96 CPU cores, there are 96 such files for a given time step *x* with *y* of 0, 1, 2, …, 94, and 95.

Here are the first several lines of *case1.stress_visc.3.50,* one of the 96 files for time step 50.

00050 42025 5.00000e+03 1.00000e+02
1.50480e+00 5.46710e-07
1.50480e+00 5.57376e-07
1.50480e+00 5.65310e-07
…

where in line 1, the first three numbers are for time step (i.e., 50), elapsed time in years (i.e., 5,000 years) and incremental time Δt in years (i.e., 100 years), respectively; each of the subsequent lines is for mantle viscosity (1st number of the line) and 2nd invariant of stress (2nd number of the line) of a node. For the example case with 41 nodes in each of the two horizontal

directions and 25 nodes in radial direction for each domain, there are 41x41x25=42025 nodes in each domain.

The above-mentioned output files require coordinate information for the finite element nodes. The coordinates for surface nodes can be found in files *case1.coord_s.y* (i.e., for the example case *y* can be 1, 3, …, and 95). The ordering of the nodes in these files is identical to that in files *case1.topo_s.y.x*. Note that these coordinate files are produced only once for the first time step to save the disk space.

Here are the first several lines of *case1.coord_s.3,* one of the 48 files (i.e., for the 4th domain).

00001 1.00000e+02
4.8487e-01 1.4788e-10
4.9789e-01 1.5290e-11
5.1090e-01 6.2832e+00
…

where in line 1, the two numbers represent time step (i.e., 1) and elapsed time in years (i.e., 100 years), respectively; each of the subsequent lines is for co-latitude and longitude (both in radium) of a node, respectively. Again, there are 41x41=1681 such nodes or such lines for co-latitude and longitude coordinates in this file.

The radial coordinates of the finite element grid can be found in files *case1.horiz_ave.z.x* for horizontal averages of viscosity and stress, where *z* is the domain or CPU index in radial direction that can be *0, 1, …*, *nprocz-1*, and for our example case, *z* can be *0,* or *1*. These files are generated for 1st, 50th, …, time steps, for the example case.

Here are the first several lines of *case1.horiz_ave.0.1* (i.e., the bottom domain)

00001 25 1.00000e+02 1.00000e+02
5.50000e-01 3.09510e+00 3.98731e-08
5.64017e-01 3.09509e+00 3.85919e-08
5.78034e-01 3.09509e+00 3.66130e-08
…
8.55573e-01 1.50480e+00 2.75407e-08

where the four numbers in line 1 are for time step (i.e., 1), the number of nodes in radial direction for this domain (i.e., 25), elapsed time in years (i.e., 100 years), and incremental time Δt in years (i.e., 100 years), respectively. Each of the subsequent lines is for a node in radial direction, starting at the core-mantle boundary or radius of *0.55* and ending at the 25th node for radius of *8.55573e-01*, i.e., in the middle of the mantle. The three numbers in each line are the radius, horizontally averaged viscosity and 2nd invariant of stress at the radius, respectively.

File *case1.horiz_ave.1.1* is for the top domain, and has the same format as *case1.horiz_ave.0.1* that we just discussed. Here are the first several lines of *case1.horiz_ave.1.1*.

00001 25 1.00000e+02 1.00000e+02
8.55573e-01 1.50480e+00 2.75407e-08
8.63422e-01 1.50480e+00 2.72184e-08
8.71272e-01 1.50480e+00 2.68796e-08
…
9.96860e-01 1.00000e+05 1.54502e-08
1.00000e+00 1.00000e+05 1.49591e-08

where the 1st line after the information line is for the overlapping node with the bottom domain (i.e., at radius of *8.55573e-01*), and the 25th line is for the surface where radius is 1.

The radial coordinates from *case1.horiz_ave.0.1* and *case1.horiz_ave.1.1,* together with *case1.coord_s.y* for surface co-latitude and longitude coordinates, can be used to build 3-D finite element grid coordinates that are needed to analyze or graph stress and viscosity with files *case1.stress_visc.y.x*. The nodal ordering in files *case1.stress_visc.y.x* is to increase node numbering along the radial direction from the bottom of the domain to the surface (i.e., 25 nodes in this example run), then shifts horizontally for one node in the same order as in *case1.coord_s.y*, and repeats the increase in node numbering from the bottom of the domain (Figure 2). A given column of nodes in radial direction (i.e., 25 nodes for the domain or 49 nodes in total) have the same co-latitude and longitude coordinates as the corresponding surface node in *case1.coord_s.y*. Likewise, a given layer of nodes in horizontal directions have the same radius as the corresponding node in *case1.horiz_ave.0.1* or *case1.horiz_ave.1.1*.

5.2. Fields with spherical harmonic expansion

Files *case1.tps_sharm.x* and *case1.vtps_sharm.x* are spherical harmonic expansions of cumulative and incremental radial (or vertical) displacements, respectively, at the surface for time step *x*. Files *case1.pttl_sharm.x*, and *case1.pttldot_sharm.x* are for the total and incremental gravitational potential anomalies, respectively, at the surface for time step *x*. Again, to determine the rate of displacement (or gravitational potential change), one can simply divide incremental displacement (or incremental gravitational potential) by time increment or Δt. These spherical harmonic expansions are computed from *case1.topo_s.y.x* that we discuss in last section. For definition of the spherical harmonic expansion in CitcomSVE, see equation 30 of Zhong et al., [2022].

Here are the first several lines of *case1.tps_sharm.50.*

50 2.25716e+01 tps
ll mm cos sin
0 0 2.706563e-13 0.000000e+00
1 0 1.810005e-06 0.000000e+00
1 1 3.577230e-07 4.120553e-07
2 0 -2.536003e-06 0.000000e+00
2 1 -4.378330e-08 1.098494e-07
2 2 1.267364e-07 7.286699e-08
…

32 31 -1.016968e-09 -6.657468e-09
32 32 -3.395314e-10 -3.685291e-09

Lines 1 and 2 are for information, and the 1st and 2nd numbers of line 1 are for the time step and time (in terms of Maxwell time), respectively. The subsequent lines are the spherical harmonic expansion from degree 0 and order 0 to degree 32 and order 32 (as set by *output_ll_max=32* in the input file). Starting from line 3, 1st and 2nd numbers of each line are spherical harmonic degree *l* and order *m*, respectively, and 3rd and 4th numbers of the line are the cosine and sine coefficients of the expansion.

The other three harmonic expansion files *case1.vtps_sharm.x*, *case1.pttl_sharm.x*, and *case1.pttldot_sharm.x* have the same format as that above.

6. Cookbook example cases

Directory *DATA* includes file *README_sample_cases* that describes 11 sample cases. This directory also stores 11 input files for these samples cases including input file *inputfileVM5a1* that was discussed extensively in previous sections. Here we will discuss five different sample cases as cookbook example cases.

6.1. Two example cases on GIA: 1-D and 3-D viscosity

6.1.1. A case with VM5a (1-D) viscosity

Input file *inputfileVM5a1* was discussed extensively in previous section and it is for case GIA_R2 in Zhong et al., [2022] that uses ice model ICE6G and viscosity model VM5a. From file *README_sample_cases*, this case uses CitcomSVE grid with 12x48x80x80 (i.e., 3,686,400 elements) spatial resolution and incremental time of 100 years per time step and is computed on 96 CPU cores. This case should compute for a total of 1,220 time steps for a total of 122,000 years of glaciation and deglaciation history until the present day. Read sections 3, 4 and 5 for details on running this case, input and output files.

Some of key input parameters in input file *inputfileVM5a1:*

```
datadir="BM_ICE6G"
datafile="case1"
apply_SLE=1
polar_wander=1
Heaviside=2    # 1: Heaviside   2: ice model e.g., ice6g
load_stages_time_file="ice6g_26ka_stages_timeA.dat"
ice_datafile="ICE6G_1x1/ice6g"
ocean_datafile="ICE6G_1x1/ocn6g"
storage_spacing=10
3d_visc_from_file=0     #0 for 1d layered viscosity; 1 for 3d visc from files
```

For details in *inputfileVM5a1*, read section 4.1 on input file.

To run this case, go to directory *CitcomSVE-2.1/DATA* with input files including *inputfileVM5a1* and *machinefile0*, and issue the following command to run this case on a PC cluster.

```
$ mpirun -np 96 -nolocal -machinefile machinefile0 ../bin/CitcomSFull inputfileVM5a1
```

Depending on your computer's speed, it may take ~2 hours to complete the model run. Output files are case1.* in subdirectory *DATA/BM_ICE6G*. Surface output files (e.g., cumulative and incremental surface displacement and gravity) are generated every 10 time steps, and the volume output files (e.g., stress, viscosity and their horizontal averages) are generated every 10*5=50 time steps.

Note that directory *CitcomSVE-2.1/DATA/Compare_Cookbook* was generated from one of the downloaded compressed file *Compare_Cookbbook.tar.gz*. This directory includes two subdirectories *BM_ICE6G* and *BM_LoveNumbers* that stores some selective output files. In subdirectory *BM_ICE6G*, you may find output files case1.* for the first 50 time steps. Note that to reduce file size for *Compare_Cookbbook.tar.gz*, some output files were exclude (e.g., all of those files for stress and viscosity on the 3-D finite element grid and some files for surface displacement and gravity on the finite element grid). Nevertheless, you can compare your output files of results for this example case with those files, if they exist, in *CitcomSVE-2.1/DATA/Compare_Cookbook/BM_ICE6G/case1.** for the first 50 time steps, to verify your calculations.

6.1.2. A case with 3-D viscosity

The second GIA example case uses 3-D mantle viscosity but is otherwise identical to the first example case (e.g., ICE6G). The input file is *inputfile3DVisc* which differs from inputfile *inputfileVM5a1* only in two places: datafile="case4" and 3d_visc_from_file=1 in file *inputfile3DVisc* which has datafile="case1" and 3d_visc_from_file=0. The 3-D viscosity is specified in file *ViscA/ViscA2x2.dat* as defined by 3d_visc_datafile="ViscA/ViscA2x2.dat" in *inputfile3DVisc.* See section 4.1 on input file and section 4.3 for details of this 3-D viscosity file. This 3-D viscosity is from A et al., [2013].

Issue the following command in directory *CitcomSVE-2.1/DATA* to run this case on a PC cluster.

```
$ mpirun -np 96 -nolocal -machinefile machinefile0 ../bin/CitcomSFull inputfile3DVisc
```

Output files are case4.* in subdirectory *DATA/BM_ICE6G* (i.e., the same place where case1.* files were stored). Files *CitcomSVE-2.1/DATA/Compare_Cookbook/BM_ICE6G/case4.** are the results for the first 50 time steps for this sample case, and you can compare your results with these files for the first 50 time steps. Again, not all the results for this cookbook case were included.

File *README_sample_cases* lists two more GIA sample cases with their input files *inputfileVM5a2* and *inputfileVM5a3* in *CitcomSVE-2.1/DATA.* These two cases using VM5a

viscosity have different spatial and temporal resolutions than case1, and comparing their input files with input file *inputfileVM5a1* for case1 will help see their setups.

6.2. Three example cases on Love number calculations

File *README_sample_cases* lists six sample cases for computing load Love numbers and one sample case for computing tidal Love numbers with their input files like *inputfileLl2m1A* or *inputfileTl2m0* in *CitcomSVE-2.1/DATA,* where in the file name, letters "*L*" and "*T*" indicate for loading and tidal Love number calculations, respectively, and letters "*A*" and "*B*", if present, represent spatial resolutions 12x32x32x32 and 12x64x64x64, respectively. As indicated in file *README_sample_cases*, these seven sample cases can be found in Zhong et al., [2022].

Here only three sample cases are described and their input files are *inputfileLl3m1A*, *inputfileLl2m1B*, and *inputfileTl2m0*, respectively.

6.2.1. A surface-loading case with input file inputfileLl3m1A

This case is for computing load Love numbers for single harmonic loading for spherical harmonic degree *1=3* and order *m=1*, with Heaviside function loading history and resolution of 12x32x32x32 using 12 CPU cores. The mantle is homogeneous but with 100 km thick high viscosity lithosphere. Input file *inputfileLl3m1A* contains all the relevant information for this case and see section 4.1 to understand the details of this input file.

Some key input parameters in *inputfileLl3m1A* are:

```
datadir="BM_LoveNumbers"
datafile="case2"
apply_SLE=0
polar_wander=0
polar_wander_kf=1.11664062
Heaviside=1    # 1: Heaviside   2: ice model e.g., ice6g
apply_potential=0      #0: apply surface loads, or 1:apply tidal potential
perturbmag=1e-6        # e.g., 1e-6
perturbl=3
perturbm=1
load_stages_time_file="heaviside1_stages_time.dat"
storage_spacing=10
```

Loading history file *heaviside1_stages_time.dat,* as specified above in *inputfileLl3m1A,* indicates that this case computes for 40 Maxwell times in 200 time steps.

Issue the following command in directory *CitcomSVE-2.1/DATA* to run this case on a PC cluster.

```
$ mpirun -np 12 -nolocal -machinefile machinefile1 ../bin/CitcomSFull inputfileLl3m1A
```

where file machinefile1 lists 6 compute nodes from node01 to node06. Output files are case2.* in subdirectory *DATA/BM_LoveNumbers*. As specified in input file inputfileLl3m1A, output results are stored every 10 time steps for surface displacements and gravity and every 50 time steps for mantle stress and viscosity. See section 5 for description of these output files. Love numbers, *h*, *k*, and *l*, as a function of time or time step, are stored in output file *case2.Love_numbers*, and the first several lines in this file are given as following:

0 0.0000e+00 -6.146858e-01 4.382950e-04 -2.343796e-01 8.503954e-04 6.123423e-02
10 2.0000e+00 -1.297651e+00 9.640255e-04 -4.864053e-01 5.733022e-04 1.492942e-01
20 4.0000e+00 -1.707063e+00 1.268237e-03 -6.351737e-01 4.108417e-04 2.138978e-01
30 6.0000e+00 -1.963441e+00 1.462319e-03 -7.266482e-01 3.098439e-04 2.612704e-01
…

Each line is for a time step (i.e., every 10 time steps). The 1st and 2nd columns are for time step and time (i.e., in terms of Maxwell time), respectively. The 3rd – 7th columns are for *h*, *h*'s dispersion error, *k*, *k*'s dispersion error, and |*l*|, respectively. See Zhong et al., [2022] for how the Love numbers are computed and the results for these cases (i.e., in Table S1 of this paper).

Files *CitcomSVE-2.1/DATA/Compare_Cookbook/BM_LoveNumbers/case2.** are the results for the first 50 time steps for this sample case, and you can compare your results with these files for the first 50 time steps.

6.2.2. A surface loading case with input file inputfileLl2m1B

This case is for computing load Love numbers for single harmonic loading for spherical harmonic degree *1=2* and order *m=1*, with Heaviside function loading history and resolution of 12x64x64x64 using 96 CPU cores. The mantle is homogeneous but with 100 km thick high viscosity lithosphere. Input file *inputfileLl2m1B* contains all the relevant information for this case, and see section 4.1 for the details of this input file.

Some key input parameters in *inputfileLl2m1B* are:

```
datadir="BM_LoveNumbers"
datafile="case6"
apply_SLE=0
polar_wander=1
polar_wander_kf=1.11664062
Heaviside=1    # 1: Heaviside   2: ice model e.g., ice6g
apply_potential=0      #0: apply surface loads, or 1:apply tidal potential
perturbmag=1e-6        # e.g., 1e-6
perturbl=2
perturbm=1
load_stages_time_file="heaviside1_stages_time.dat"
storage_spacing=5
```

Using the same loading history file as for the previous case, this case computes for 40 Maxwell times in 200 time steps. Note that given the loading harmonic degree *1=2* and order *m=1*, polar wander is generated and *polar_wander=1*.

Issue the following command in directory *CitcomSVE-2.1/DATA* to run this case on a PC cluster.

```
$ mpirun -np 96 -nolocal -machinefile machinefile0 ../bin/CitcomSFull inputfileLl2m1B
```

Output files are case6.* in subdirectory *DATA/BM_LoveNumbers*. As specified in input file inputfileLl2m1B, output results are stored every 5 time steps for surface displacements and gravity and every 25 time steps for mantle stress and viscosity. See section 5 for description of these output files. Love numbers, *h*, *k*, and *l*, as a function of time or time step, are stored in output file *case6.Love_numbers*.

Files *CitcomSVE-2.1/DATA/Compare_Cookbook/BM_LoveNumbers/case6.** are the results for the first 25 time steps for this sample case, and you can compare your results with these files for the first 25 time steps.

6.2.3. A tidal loading case with input file inputfileTl2m0

This case is for computing tidal Love numbers for single harmonic loading for spherical harmonic degree *1=2* and order *m=0*, with Heaviside function loading history and resolution of 12x64x64x64 using 96 CPU cores. The mantle is homogeneous but with 100 km thick high viscosity lithosphere. Input file *inputfileTl2m0* contains all the relevant information for this case, and see section 4.1 for the details of this input file.

Some key input parameters in *inputfileTl2m0* are:

```
datadir="BM_LoveNumbers"
datafile="case7"
apply_SLE=0
polar_wander=0
Heaviside=1    # 1: Heaviside   2: ice model e.g., ice6g
apply_potential=1      #0: apply surface loads, or 1:apply tidal potential
perturbmag=1e-6        # e.g., 1e-6
perturbl=2
perturbm=0
load_stages_time_file="heaviside2_stages_time.dat"
storage_spacing=10
```

Differing from previous cases with surface loading, now *apply_potential=1* means that tidal potential is applied. With loading history file *heaviside2_stages_time.dat*, this case computes for 400 Maxwell times in 800 time steps.

Issue the following command in directory *CitcomSVE-2.1/DATA* to run this case on a PC cluster.

```
$ mpirun -np 96 -nolocal -machinefile machinefile0 ../bin/CitcomSFull inputfileTl2m0
```

Output files are case7.* in subdirectory *DATA/BM_LoveNumbers*. As specified in input file inputfileTl2ml0, output results are stored every 10 time steps for surface displacements and gravity and every 50 time steps for mantle stress and viscosity. See section 5 for description of these output files. Love numbers, *h'*, *k'*, and *l'*, as a function of time or time step, are stored in output file *case7.Love_numbers*.

Files *CitcomSVE-2.1/DATA/Compare_Cookbook/BM_LoveNumbers/case7.** are the results for the first 30 time steps for this sample case, and you can compare your results with these files for the first 30 time steps.

7. References

A, G.R., J. Wahr, and S.J. Zhong (2013). Computations of the viscoelastic response of a 3-D compressible Earth to surface loading: an application to glacial isostatic adjustment in Antartica and Canada, Geophys. J. Int., 192, 557-572.

Bellas, A., and S.J. Zhong (2021). Seismic Strain Rate and Flexure at the Hawaiian Islands Constrain the Fictional Coefficient, Geochemistry, Geophysics, Geosystems, 22, e2020GC009547.https://doi.org/10.1029/2020GC009547

Kang, K.S., S.J. Zhong, G.R., A, W. Mao (2022), The effects of non-Newtonian rheology in the upper mantle on relative sea level change and geodetic observations induced by glacial isostatic adjusted process, Geophys. J. Int., 228, 1887–1906, https://doi.org/10.1093/gji/ggab428.

Moresi, L.N., and V.S. Solomatov (1995), Numerical investigation of 2D convection with extremely large viscosity variations, Phys. Fluids, 7, 2154-2162.

Paulson, A., S.J. Zhong, and J. Wahr (2005), Modeling post-glacial rebound with lateral viscosity variations, Geophys. J. Int., 163, 357-371.

Qin, C., S.J. Zhong, and J. Wahr (2014). A perturbation method and its application: elastic tidal response of a laterally heterogeneous planet, Geophys. J. Int., 199, 631-647.

Zhong, S.J., K.X. Kang, G.R. A, and C. Qin (2022). CitcomSVE: A Three-Dimensional Finite Element Software Package for Modeling Planetary Mantle's Viscoelastic Deformation in Response to Surface and Tidal Loads, Geochem. Geophys. Geosyst, 23, e2022GC010359, https://doi.org/10.1029/2022GC010359.

Zhong, S.J., and A.B. Watts (2013). Lithospheric deformation induced by loading of the Hawaiian Islands and its implications for mantle rheology, J. Geophys. Res., 118, 6025-6048, doi:10.1002/2013JB010408.

Zhong, S.J., C. Qin, G.R. A, and J. Wahr (2012). Can tidal tomography be used to unravel the long-wavelength structure of the lunar interior? Geophys. Res. Lett., 39, L15201, doi:10.1029/2012GL052362.

Zhong, S.J., A.K. McNamara, E. Tan, L. Moresi, and M. Gurnis (2008). A benchmark study on mantle convection in a 3-D spherical shell using CitcomS, Geochem. Geophys. Geosyst, 9, Q10017, doi:10.1029/2008GC002048.

Zhong, S.J., A. Paulson, and J. Wahr (2003). Three-dimensional Finite Element Modeling of Earth's Viscoelastic Deformation: Effects of Lateral Variations in Lithospheric Thickness, Geophys. J. Int., 155, 679-695.

Zhong, S.J., Zuber, M.T., Moresi, L., Gurnis, M. (2000). Role of temperature-dependent viscosity and surface plates in spherical shell models of mantle convection. J. Geophys. Res.-Solid Earth 105, 11063–11082. doi:10.1029/2000JB900003.