SICOPOLIS V5-dev
– Quick Start Manual –

Ralf Greve

Institute of Low Temperature Science, Hokkaido University,
Kita-19, Nishi-8, Kita-ku, Sapporo 060-0819, Japan

November 8, 2022

1 Requirements

- Unix-like system (e.g., Linux).
- Fortran compiler.
  So far, the GNU GCC (gfortran) and Intel Fortran (ifort) compilers are supported. If you wish to use a different compiler, please contact <help@sicopolis.net>.
- SICOPOLIS writes output in NetCDF format (plus some ASCII). An installation of NetCDF version 3.6.x or newer (https://www.unidata.ucar.edu/software/netcdf/) is therefore required. For installation support, see Appendix A.
- For the shallow-shelf/shelfy-stream solver, the Library of Iterative Solvers for Linear Systems (Lis, version 1.4.43 or newer) is required (https://www.ssisc.org/lis/). For installation support, see Appendix B.

2 Download

- **Using Git**

  The Git repository of SICOPOLIS is kindly hosted by the GitLab system of the Alfred Wegener Institute for Polar and Marine Research (AWI) in Bremerhaven, Germany. Front page: https://gitlab.awi.de/sicopolis/sicopolis/ .

  Cloning the latest develop revision:

  ```
  git clone --branch develop \
      https://gitlab.awi.de/sicopolis/sicopolis.git
  ```

  (Cloning with SSH instead of HTTPS is also available. See the above GitLab front page link for details.)

  You should then have a new directory "sicopolis" that contains the entire program package.

3 Initial configuration

1. Go to the new directory "sicopolis" and execute the following bash scripts:

   ```
   ./copy_templates.sh
   ./get_input_files.sh
   ```

The latter can be configured if you want to download only selected input files (default is downloading everything). To do so, open it with a text editor and change the flag variables before execution.

2. Locate the file sico_configs.sh in the directory sicopolis/runs, and open it with a text editor.

3. Set the flags
   LIS_FLAG, OPENMP_FLAG and LARGE_DATA_FLAG
   according to your needs.

   Default is "true" / "true" / "false", which works for all test simulations included in the SICOPOLIS package. LIS_FLAG and OPENMP_FLAG can be set to "false" for simulations with pure shallow-ice dynamics. However, "true" is required for simulations with shallow-shelf dynamics (for floating ice) or hybrid shallow-ice–shelfy-stream dynamics (for grounded ice). For high-resolution simulations (e.g., Greenland/5 km or Antarctica/8 km), LARGE_DATA_FLAG must be set to "true".

4. Set NETCDFHOME to the correct path of your NetCDF installation.

   If LIS_FLAG = "true", set LISHOME to the correct path of your Lis installation.

5. Depending on your system, some additional settings might have to be added in sico_configs.sh (`module load` commands for dynamic loading etc.).

6. Locate the file sico_environment.sh in the directory sicopolis/runs, open it with a text editor, and replace the "Default" entry for SICO_INSTITUTION by the name of your institution (max. 256 characters).

4 Files and directories in "sicopolis"

- **runs**:

  Configuration file sico_configs.sh.

  Shell script (bash) sico.sh for running a single simulation.

  Shell scripts (bash) multi_sico_1.sh and multi_sico_2.sh for running multiple simulations by repeated calls of sico.sh.

  Subdirectory **headers**: specification files sico_specs_*run_name*.h (*run_name*: name of run) for a number of computationally rather inexpensive test runs.

– Run v5_vialov3d25
  ⟶ 3-d version of the 2-d "Vialov profile" (Vialov 1958),
  SIA, resolution 25 km, $t = 0 \ldots 100$ ka.
  Similar to the EISMINT Phase 1 fixed-margin experiment
  (Huybrechts et al. 1996), but without thermodynamics. Instead,
  isothermal conditions with $T = -10°C$ everywhere are assumed.

– Run v5_emtp2sge25_expA
  ⟶ EISMINT Phase 2 Simplified Geometry Experiment A,
  SIA, resolution 25 km, $t = 0 \ldots 200$ ka (Payne et al. 2000).
  The thermodynamics solver for this run is the one-layer melting-CTS
  enthalpy scheme (ENTM), while all other runs employ the
  polythermal two-layer scheme (POLY) (Greve and Blatter 2016).

– Run v5_grl16_bm5_ss25ka
  ⟶ Greenland ice sheet, SIA, resolution 16 km,
  short steady-state run ($t = 0 \ldots 25$ ka) for modern climate conditions
  (unpublished).

– Run v5_ant40_b2_ss25ka
  ⟶ Antarctic ice sheet without ice shelves, SIA, resolution 40 km,
  short steady-state run ($t = 0 \ldots 25$ ka) for modern climate conditions
  (unpublished).

– Run v5_grl20_b2_paleo21
  ⟶ Greenland ice sheet, SIA, resolution 20 km,
  $t = -140 \ldots 0$ ka, basal sliding ramped up during the first 5 ka
  [modified, low-resolution version of the spin-up for ISMIP6 InitMIP;
  Greve et al. (2017)].

– Runs v5_grl10_b2_paleo21 and
  v5_grl10_b2_future21_ctrl/..._asmb
  ⟶ Greenland ice sheet, SIA, resolution 10 km,
  $t = -9 \ldots 0$ ka for the paleo run, $t = 0 \ldots 100$ a for the future runs
  [10-km version of the spin-up and the schematic future climate runs for
  ISMIP6 InitMIP; Greve et al. (2017)].

– Runs v5_ant64_b2_spinup09_init100a,
  v5_ant64_b2_spinup09_fixtopo, v5_ant64_b2_spinup09 and
  v5_ant64_b2_future09_ctrl

$\longrightarrow$ Antarctic ice sheet with hybrid shallow-ice–shelfy-stream dynamics (Bernales et al. 2017) and ice shelves (SSA), resolution 64 km,
$t = -140.1 \ldots -140.0$ ka for the init run without basal sliding (..._init100a),
$t = -140 \ldots 0$ ka for the run with almost fixed topography (..._fixtopo), basal sliding ramped up during the first 5 ka,
$t = -0.5 \ldots 0$ ka for the final, freely-evolving-topography part of the spin-up (..._spinup09),
$t = 0 \ldots 100$ a for the constant-climate control run (..._future09_ctrl)
[64-km version of the spin-up and the constant-climate control run for ISMIP6 InitMIP; Greve and Galton-Fenzi (pers. comm. 2017)].

- Runs v5_asf2_steady and v5_asf2_surge
  $\longrightarrow$ Austfonna, SIA, resolution 2 km, $t = 0 \ldots 10$ ka
  [similar to Dunse et al.'s (2011) Exp. 2 (steady fast flow) and Exp. 5 (surging-type flow), respectively].

- Runs v5_nmars10_steady and v5_smars10_steady
  $\longrightarrow$ North-/south-polar cap of Mars, SIA, resolution 10 km, $t = -10$ Ma $\ldots 0$
  [steady-state runs by Greve (2007)].

- Run v5_nhem80_nt012_new
  $\longrightarrow$ northern hemisphere, SIA, resolution 80 km, $t = -250 \ldots 0$ ka
  [similar to run nt012 by Greve et al. (1999)].

- Run v5_heino50_st
  $\longrightarrow$ ISMIP HEINO standard run ST,
  SIA, resolution 50 km, $t = 0 \ldots 200$ ka (Calov et al. 2010).

- **src**:

  Directory that contains the main program file sicopolis.F90.

  - Subdirectory **subroutines/general**: general subroutines, for any modelled domain.

  - Subdirectory **subroutines/ant**: subroutines specific for the Antarctic ice sheet.

  - Subdirectory **subroutines/emtp2sge**: subroutines specific for the EISMINT Phase 2 Simplified Geometry Experiments.

  - Subdirectory **subroutines/grl**: subroutines specific for the Greenland ice sheet.

  - Accordingly subdirectories subroutines/asf, nhem, scand, tibet, nmars and smars

for Austfonna, the northern hemisphere, Scandinavia, Tibet and the north and south polar caps of Mars, respectively.

  - Subdirectory **subroutines/xyz**: see Appendix C.

- **sico_in**:

  Directory that contains input data files for SICOPOLIS.

  - Subdirectory **general**: general input files, for any modelled domain.
  - Subdirectory **ant**: input files specific for the Antarctic ice sheet.
  - Subdirectory **emtp2sge**: input files specific for the EISMINT Phase 2 Simplified Geometry Experiments.
  - Subdirectory **grl**: input files specific for the Greenland ice sheet.
  - Accordingly subdirectories asf, nhem, scand, tibet, nmars and smars for Austfonna, the northern hemisphere, Scandinavia, Tibet and the north and south polar caps of Mars, respectively.
  - Subdirectory **xyz**: see Appendix C.

- **sico_out**:

  Empty directory into which output files of SICOPOLIS simulations are written.

- **docu**:

  Directory that contains some documentation.

  - Subdirectory **quick_start**:
    LaTeX source for this manual (PDF must be built with make).
  - Subdirectory **doxygen**: documentation to be created by Doxygen
    (optional, see doxygen-config/README.md).
    * html/index.html $\longrightarrow$ Source code browser.
    * latex/refman.pdf $\longrightarrow$ Reference manual.

- **tools**:

  See Sect. 8.

- **license**:

  Directory that contains a copy of the GNU General Public License (version 3).

5 How to run a simulation

1. For example, to run simulation v5_grl16_bm5_ss25ka, use the script sico.sh:

   `(./sico.sh -m v5_grl16_bm5_ss25ka) >out_001.dat 2>&1 &`

   (from directory sicopolis/runs, bash required). Accordingly for the other simulations.
   For further options, try `./sico.sh -h`.
   \!/ Do not use out_*run_name*.dat for the redirected output of sico.sh.
   This name is reserved for the runtime output of SICOPOLIS itself.
   (Both are very useful in case of compilation or runtime errors!)

2. Alternatively, if you prefer to run all simulations consecutively, execute the script multi_sico_1.sh:

   `(./multi_sico_1.sh) >out_multi_100.dat 2>&1 &`

   For further options, try `./multi_sico_1.sh -h`.

Computing times

The approximate computing times for the simulations are listed in Table 1 (Appendix D).

6 Output files

Output files are written by default to the directory sicopolis/sico_out/*run_name* (this can be changed with the `-d` option). Four types are produced:

- **run_name.log**:

  ASCII file that lists the main specifications of simulation *run_name*.

- **run_name.ser, run_name_ser.nc**:

  Time-series files (ASCII, NetCDF) that contain scalar variables:

  - Time, t
  - Surface temperature anomaly, D_Ts, or glacial index, glac_ind (forcing)
  - Sea level, z_sl (forcing)
  - Total ice volume, V
  - Volume of grounded ice, V_g
  - Volume of floating ice, V_f
  - Total ice area, A

  - Area of grounded ice, A_g
  - Area of floating ice, A_f
  - Ice volume above flotation in sea level equivalent, V_sle
  - Volume of temperate ice, V_t
  - Area of temperate-based grounded ice, A_t
  - Maximum ice thickness, H_max
  - Maximum thickness of temperate ice, H_t_max
  - Maximum surface elevation, zs_max
  - Maximum surface speed, vs_max
  - Maximum basal temperature (relative to pmp), Tbh_max
  - (Some more in the NetCDF file, try `ncdump -h run_name_ser.nc`)

- **run_name.core, run_name_core.nc**:

  Time-series files (ASCII, NetCDF) that contain for selected locations xxx:

  - Time, t
  - Surface temperature anomaly, D_Ts, or glacial index, glac_ind (forcing)
  - Sea level, z_sl (forcing)
  - Thickness, H_xxx
  - Surface velocity, v_xxx
  - Basal temperature, T_xxx
  - (Some more in the NetCDF file, try `ncdump -h run_name_core.nc`)

  For the Greenland ice sheet, these data are written for seven locations:
  GRIP (xxx=GR), GISP2 (xxx=G2), Dye 3 (xxx=D3), Camp Century (xxx=CC), NorthGRIP (xxx=NG), NEEM (xxx=NE), EastGRIP (xxx=EG).

  For the Antarctic ice sheet, these data are written for six locations:
  Vostok (xxx=Vo), Dome A (xxx=DA), Dome C (xxx=DC), Dome F (xxx=DF), Kohnen (xxx=Ko), Byrd (xxx=By).

- **run_name0001.nc**, **run_name0002.nc**, ...:

Complete set of fields (topography, velocity, temperature etc., written in NetCDF (*.nc) format) for selected time slices defined in specifications file.

For example, simulation v5_grl16_bm5_ss25ka produces three files v5_grl16_bm5_ss25ka0001.nc, v5_grl16_bm5_ss25ka0002.nc and v5_grl16_bm5_ss25ka0003.nc, which correspond to $t = 0$, 10 ka and 25 ka, respectively.

7 Plotting

The output described in Sect. 6 can be visualized with any plotting tool at the user's preference. Ncview (http://meteora.ucsd.edu/∼pierce/ncview_home_page.html) is a very nice browser for NetCDF files to get a quick and easy look. For more sophisticated plots, one possibility is to use MATLAB, which has an extensive library for NetCDF files (https://www.mathworks.com/help/matlab/network-common-data-form.html). For instance, the following script plots the final surface topography of the Greenland simulation v5_grl16_bm5_ss25ka (credit: Mathieu Morlighem, University of California Irvine).

```
filename = 'v5_grl16_bm5_ss25ka0003.nc';
x = ncread(filename,'x');
y = ncread(filename,'y');
surf = ncread(filename,'zs');
% Display surface elevation
%    (transposition needed because MATLAB is column-oriented)
imagesc(x*1e-3,y*1e-3,surf'); axis xy equal; caxis([0 3200]); colorbar
```

8 Some useful tools

The directory sicopolis/tools contains some useful tools.

8.1 Program make_ismip_output

Generating ISMIP output (see http://tinyurl.com/clic-ismip6) from the NetCDF time-slice files produced by SICOPOLIS (see Sect. 6). For simulation *run_name*, to be executed by

```
./tools.sh -p make_ismip_output -m run_name
```

For further options, try `./tools.sh -h`.

8.2 Program resolution_doubler

Doubling the horizontal resolution of a NetCDF time-slice output file produced by SICOPOLIS (see Sect. 6). For simulation *run_name*, to be executed by

```
./tools.sh -p resolution_doubler -m run_name
```

For further options, try `./tools.sh -h`.

For example, run v5_grl10_b2_paleo21 (10 km resolution) requires the resolution-doubled output of run v5_grl20_b2_paleo21 (20 km resolution) for $t = -9$ ka as initial condition. In order to create it, execute the resolution doubler for run v5_grl20_b2_paleo21 (i.e., with the option `-m v5_grl20_b2_paleo21`) and enter

```
Number of time-slice file (with leading zeros, 4 digits) > 0004
```

This will convert the original time-slice file v5_grl20_b2_paleo210004.nc to the resolution-doubled file v5_grl20_b2_paleo21_dbl_0004.nc that serves as initial conditions for run v5_grl10_b2_paleo21.

A Installation of NetCDF

NetCDF (Network Common Data Form) is a common format for scientific data (https://www.unidata.ucar.edu/software/netcdf/) that is also used by SICOPOLIS. The NetCDF C and Fortran libraries are required.

For **GCC**, installation from a package manager is recommended. Under openSUSE Leap 15.2, install netcdf, netcdf-devel, netcdf-devel-static, netcdf-fortran, netcdf-fortran-devel, netcdf-fortran-static, ncview. This requires the repositories "Software for Scientists and Engineers" and "sebschub's Home Project". Details (especially the required repositories) will differ for other systems.

For the **Intel compiler**, manual installation is required. The C and Fortran libraries are available for download on the NetCDF website as zip or tar archives. Unzip them into temporary source directories.

- Prior to version 4.2, a single archive contained both the C and Fortran libraries. A minimal installation for version 4.1.3 (without NetCDF-4 support) can be done by changing to the source directory, then:

```
export NCDIR=/opt/netcdf
export CC=icc
export FC=ifort
export CFLAGS="-O2"
export CPPFLAGS=
export FCFLAGS="-O2"
export FFLAGS=${FCFLAGS}
./configure --prefix=${NCDIR} --libdir=${NCDIR}/lib \
            --disable-netcdf-4
make install
```

- Since version 4.2, the C and Fortran libraries must be installed separately. If NetCDF-4 support is dispensable, the following installation should work (tested under openSUSE Leap 15.2 and icc/ifort 19.1 with versions netcdf-c-4.8.0 and netcdf-fortran-4.5.3 as of January 25, 2021).

  Step 1: Change to the source directory of the C library, then:

```
export NCDIR=/opt/netcdf
export CC=icc
export FC=ifort
```

```
export CFLAGS="-O2"
export CPPFLAGS=
export FCFLAGS="-O2"
export FFLAGS=${FCFLAGS}
./configure --prefix=${NCDIR} --libdir=${NCDIR}/lib \
            --disable-netcdf-4 --enable-logging
make install
```

Step 2: Change to the source directory of the Fortran library, then:

```
export NFDIR=/opt/netcdf
export LD_LIBRARY_PATH=${NCDIR}/lib:${LD_LIBRARY_PATH}
export CPPFLAGS=-I${NCDIR}/include
export LDFLAGS=-L${NCDIR}/lib
./configure --prefix=${NFDIR} --libdir=${NFDIR}/lib \
            --disable-netcdf-4 --enable-logging
make install
```

- For a complete build with NetCDF-4 support, additional libraries are required. See the NetCDF website for further instructions.

Installation under /opt usually requires admin rights. The same holds for the common alternative /usr/local. For a local installation, replace it by '/home/<my_user_name>/local'.

B Installation of Lis

Lis (Library of Iterative Solvers for linear systems) is a software library for solving discretized linear equations (Nishida 2010).

Download the latest version of Lis as a zip archive from https://www.ssisc.org/lis/ (as of January 23, 2021: lis-2.0.30.zip). Unzip the archive into a temporary directory.

For **GCC**, install lis by executing:

```
export LISDIR=/opt/lis
./configure --prefix=${LISDIR} --libdir=${LISDIR}/lib \
            --enable-fortran --enable-f90 \
            --enable-omp --enable-saamg --enable-fma \
            CC=gcc FC=gfortran F77=gfortran \
            CFLAGS="-mcmodel=medium" CPPFLAGS="-mcmodel=medium" \
            FCFLAGS="-mcmodel=medium" FFLAGS="-mcmodel=medium"
```

```
make install
```

This has been tested under openSUSE Leap 15.2 and Linux Mint 20.1 (some modifications might be needed under different systems).

For the **Intel compiler**, replace 'gcc' and 'gfortran' by 'icc' and 'ifort', respectively.

Installation under /opt usually requires admin rights. The same holds for the common alternative /usr/local. For a local installation, replace it by '/home/<my_user_name>/local'.

C Domain XYZ

This framework allows creating new domains (Laurentide ice sheet, simple testing geometry etc.). The directory sicopolis/src/subroutines/xyz, which hosts the domain-specific subroutines, is by default empty. If you want to create a new domain, copy the subroutines from the most similar existing domain (northern hemisphere, EISMINT etc.), e.g.:

```
cp sicopolis/src/subroutines/nhem/*.F90 \
   sicopolis/src/subroutines/xyz/
```

Then modify the routines according to your needs. Input files (topography etc.) must be placed in sicopolis/sico_in/xyz and specified in the run-specification header file *.h as usual. The domain must be defined by the domain code '#define XYZ' in the header file. For flexible testing, it is recommended to set the parameter CHECK_RES_IMAX_JMAX (compatibility check between horizontal resolution and number of grid points) to 0. If the new domain requires new global variables, they can be defined in the module sicopolis/src/subroutines/xyz/sico_vars.F90.

The subroutines for ISMIP HEINO are available in sicopolis/src/subroutines/xyz/heino, and the input files are in sicopolis/sico_in/xyz. If you copy the subroutines from sicopolis/src/subroutines/xyz/heino to sicopolis/src/subroutines/xyz, you can run ISMIP HEINO experiments (e.g., the run v5_heino50_st for which a header file is available).

D Table: Simulations and computing times

Run	Model time	Time step†	CPU time‡
v5_vialov3d25	100 ka	20 a	1.0 min
v5_emtp2sge25_expA	200 ka	20 a	3.9 min
v5_grl16_bm5_ss25ka	25 ka	5 a	9.7 min
v5_ant40_b2_ss25ka	25 ka	10 a	5.0 min
v5_grl20_b2_paleo21	140 ka	5 a	0.8 hrs
v5_grl10_b2_paleo21*	9 ka	1 a	1.0 hrs
v5_grl10_b2_future21_ctrl	100 a	1 a	0.9 min
v5_grl10_b2_future21_asmb	100 a	1 a	0.9 min
v5_ant64_b2_spinup09_init100a	100 a	2 / 10 a†	4.1 sec
v5_ant64_b2_spinup09_fixtopo	140 ka	5 / 10 a†	0.7 hrs
v5_ant64_b2_spinup09	500 a	2 / 10 a†	0.5 min
v5_ant64_b2_future09_ctrl	100 a	2 / 10 a†	6.1 sec

Table 1: Model times, time steps and computing (CPU) times for the EISMINT, Greenland and Antarctica simulations contained in the script multi_sico_1.sh, run with SICOPOLIS V5-dev (revision develop_239_rv5.2-122-g9c909c3) and the Intel Fortran Compiler 19.1 for Linux (optimization options -xHOST -O3 -no-prec-div) on a 12-Core Intel Xeon Gold 6256 (3.6 GHz) PC under openSUSE Leap 15.4.

†: If one value is given, this is the common dynamic (velocity, ice thickness) and thermodynamic (temperature, water content, age) time step. If two values are given (marked by the dagger (†) symbol), the first one is the dynamic, the second one the thermodynamic time step.

‡: All runs were done on a single core only. The v5_ant64_b2_xxx runs that include ice shelves can be done on multiple cores using OpenMP for the SSA solver. However, at the employed, low resolution of 64 km the solver does not scale well, and the gain in wall clock time by using multiple cores is very small.

*: For this run, see the remark in Sect. 8.2 on the tool resolution_doubler.

References

Bernales, J., I. Rogozhina, R. Greve and M. Thomas. 2017. Comparison of hybrid schemes for the combination of shallow approximations in numerical simulations of the Antarctic Ice Sheet. *The Cryosphere*, **11** (1), 247–265. doi:10.5194/tc-11-247-2017.

Calov, R., R. Greve, A. Abe-Ouchi, E. Bueler, P. Huybrechts, J. V. Johnson, F. Pattyn, D. Pollard, C. Ritz, F. Saito and L. Tarasov. 2010. Results from the Ice-Sheet Model Intercomparison Project – Heinrich Event INtercOmparison (ISMIP HEINO). *Journal of Glaciology*, **56** (197), 371–383. doi:10.3189/002214310792447789.

Dunse, T., R. Greve, T. V. Schuler and J. O. Hagen. 2011. Permanent fast flow versus cyclic surge behaviour: numerical simulations of the Austfonna ice cap, Svalbard. *Journal of Glaciology*, **57** (202), 247–259. doi:10.3189/002214311796405979.

Greve, R. 2007. The polar ice caps of Mars. *Low Temperature Science*, **66**, 139–148. URL `http://hdl.handle.net/2115/34722`.

Greve, R. and H. Blatter. 2016. Comparison of thermodynamics solvers in the polythermal ice sheet model SICOPOLIS. *Polar Science*, **10** (1), 11–23. doi:10.1016/j.polar.2015.12.004.

Greve, R., R. Calov and U. C. Herzfeld. 2017. Projecting the response of the Greenland ice sheet to future climate change with the ice sheet model SICOPOLIS. *Low Temperature Science*, **75**, 117–129. doi:10.14943/lowtemsci.75.117.

Greve, R., K.-H. Wyrwoll and A. Eisenhauer. 1999. Deglaciation of the Northern Hemisphere at the onset of the Eemian and Holocene. *Annals of Glaciology*, **28**, 1–8. doi:10.3189/172756499781821643.

Huybrechts, P., A. J. Payne and EISMINT Intercomparison Group. 1996. The EISMINT benchmarks for testing ice-sheet models. *Annals of Glaciology*, **23**, 1–12. doi:10.3189/S0260305500013197.

Nishida, A. 2010. Experience in developing an open source scalable software infrastructure in Japan. In: D. Taniar, O. Gervasi, B. Murgante, E. Pardede and B. O. Apduhan (Eds.), *Computational Science and Its Applications – ICCSA 2010*, Lecture Notes in Computer Science 6017, pp. 448–462. Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-12165-4_36.

Payne, A. J., P. Huybrechts, A. Abe-Ouchi, R. Calov, J. L. Fastook, R. Greve, S. J. Marshall, I. Marsiat, C. Ritz, L. Tarasov and M. P. A. Thomassen. 2000. Results from

the EISMINT model intercomparison: the effects of thermomechanical coupling. *Journal of Glaciology*, **46** (153), 227–238. doi:10.3189/172756500781832891.

Vialov, S. S. 1958. Regularities of glacial shields movement and the theory of plastic viscous flow. In: *Physics of the Motion of Ice*, IAHS Publication No. 47, pp. 266–275. IAHS Press, Wallingford, UK.