Simulation Results from the Regional Glaciation Model for Western Canada

Version 1.10
Date 23 March 2015

Data link `http://www.unbc.ca/research/supplementary-data-unbc-publications`

Introduction

The projection results are stored as binary MATLAB ("MAT") files version 7.3 and generated using MATLAB version 2014a. If your efforts to load and read the MATLAB results files are not sucessful this is probably because you are using an older version of MATLAB that does not support the version 7.3 MAT files.

Directory structure and examples

Once you have entered the University of Northern British Columbia website (see data link above) select "Supplementary material" in your web browser to navigate to the "Index of /data screen". At this point you are in the directory `/data`. Select the subdirectory `RGM_archive`.

The RGM_archive subdirectory has the following directory structure:
`/GCM_name/ Scenario_name`

where `GCM_name`: `Scenario_name`:

GCM_name	Scenario_name
CRU	HIST
CSIRO-320km	RCP26, RCP45, RCP60, RCP85
CanESM-320km	RCP26, RCP45, RCP85
GFDL-320km	RCP26, RCP45, RCP60, RCP85
HadGEM-320km	RCP26, RCP45, RCP60, RCP85
MIROC-320km	RCP26, RCP45, RCP60, RCP85
MPI-320km	RCP26, RCP45, RCP85
NARR	[none]

As an example, the run output files for the RCP26 scenario and the CSIRO-320km GCM are located in the directory `/CSIRO-320km/RCP26`

Each of the run output directories contains 10 MAT files, one for each of the 10 subregions of this study (see Fig. 1 of the Clarke et al. 2015 *Nature Geoscience* paper for a map showing the locations of the subregions).

The 10 MAT files in the directory /CSIRO-320km/RCP26 have the following names

```
CSIRO-320km_RCP26_R01_2009-2100.mat
CSIRO-320km_RCP26_R02_2009-2100.mat
CSIRO-320km_RCP26_R03_2009-2100.mat
CSIRO-320km_RCP26_R04_2009-2100.mat
CSIRO-320km_RCP26_R05_2009-2100.mat
CSIRO-320km_RCP26_R06_2009-2100.mat
```

```
CSIRO-320km_RCP26_R07_2009-2100.mat
CSIRO-320km_RCP26_R08_2009-2100.mat
CSIRO-320km_RCP26_R09_2009-2100.mat
CSIRO-320km_RCP26_R10_2009-2100.mat
```

Here and for each of the output file directories the file naming convention is

```
[GCM_name]_[Scenario_name]_[Region_ID]_[Date_limits].mat
```

Extract the files that you require from the archive and place them in your own workspace. Use a current version of MATLAB (or the supplied Python script) to extract simulation results from one of MAT files that you have downloaded in your workspace. As as example, to extract data from the file `CSIRO-320km_RCP26_R01_2009-2100.mat` launch MATLAB and enter the following on the MATLAB command line:

```
load('CSIRO-320km_RCP26_R01_2009-2100.mat')
```

When the above command is executed the following data will be placed in the MATLAB workspace:

Variable name	Description	Size
year	List of years in this data file	nt × 1
B	Bed surface elevation array (km)	ny × nx
S	Ice surface arrays	nt × ny × nx
R_mask	Region mask	ny × nx
map_eastern_edge	*X* coordinate of eastern map edge (m)	1 × 1
map_western_edge	*X* coordinate of western map edge (m)	1 × 1
map_northern_edge	*Y* coordinate of northern map edge	1 × 1
map_southern_edge	*Y* coordinate of southern map edge	1 × 1
x_centre	*X* coordinate of map centre (m)	1 × 1
y_centre	*Y* coordinate of map centre (m)	1 × 1
dx	Grid spacing in *X* direction (m)	1 × 1
dy	Grid spacing in *Y* direction (m)	1 × 1

All projection run output files have this same structure. The array size variables `nx`, `ny` and `nt` can be obtained from the data using the following MATLAB command

```
[nt,ny,nx] = size(S);
```

Using MATLAB, cell-centred grid can be constructed from the above data as follows:

```
x_grid = map_western_edge+0.5*dx :dx:map_eastern_edge-0.5*dx
y_grid = map_northern_edge-0.5*dy:-dy:map_southern_edge+0.5*dy
```

where `x_grid` and `y_grid` are row vectors.

Maps of the bed surface *B*, topographic surface *S*, ice thickness *H*, ice mask *I*, region mask *R* and the region-masked ice thickness H_{masked} be readily plotted in MATLAB as follows:

```
figure(1)
imagesc(x_grid, y_grid, B), axis equal
colorbar
xlabel('Easting distance (m)')
ylabel('Northing distance (m)')
title('Bed surface topography (m)')

% The year 2060 corresponds to the array element year(51).
% To extract the surface topography at 2060 (year(51))

S_yr_map = squeeze(S(51,:,:));

figure(2)
imagesc(x_grid, y_grid, S_yr_map), axis equal
colorbar
xlabel('Easting distance (m)')
ylabel('Northing distance (m)')
title('Surface topography (m)')

H = S_yr_map-B;

figure(3)
imagesc(x_grid, y_grid, H), axis equal
colorbar
xlabel('Easting distance (m)')
ylabel('Northing distance (m)')
title('Ice thickness (m)')

I = zeros(size(H));
I(H>0) = 1;

figure(4)
imagesc(x_grid, y_grid, I), axis equal
colorbar
xlabel('Easting distance (m)')
ylabel('Northing distance (m)')
title('Ice mask')

figure(5)
imagesc(x_grid, y_grid, R_mask), axis equal
colorbar
xlabel('Easting distance (m)')
ylabel('Northing distance (m)')
title('Region mask')

% Apply region mask to ice thickness data and plot masked result

H_masked = zeros(size(H));
H_masked(R_mask==1) = H(R_mask==1);

figure(6)
imagesc(x_grid, y_grid, H_masked), axis equal
colorbar
xlabel('Easting distance (m)')
ylabel('Northing distance (m)')
title('Region-masked ice thickness (m)')
```

The above MATLAB script is reproduced in the file `map_plotter.m` and an equivalent version in the Python file `map_plotter.py`.

Notes on map projections

The simulation results are presented in a rectangular Cartesian map grid with (*x*,*y*) coordinates in this grid

The bounding box for the map of the entire region is

```
x_E = -438000;
x_W = -1835000;
y_S = -167000;
y_N = +1588000;
```

Cell-centred coordinates of points in this grid are

```
x_grid = x_W-0.5*dx:dx:x_E+0.5*dx;
y_grid = y_N+0.5*dy:-dy:y_S-0.5*dy;
```

The above arrays are one-dimensional row vectors. Two-dimensional arrays carrying the same information can be generated as follows;

```
[X,Y] = meshgrid(x_grid,y_grid);
```

where (X(21,31),Y(21,31)) are the (x,y) coordinates of the array element (21,31). Thus, for example, the bed elevation B(21,31) applies to the point having Cartesian coordinates [X(21,31),Y(21,31)] .

The Cartesian grid used in our calculations has been generated by converting data from a Lambert Conformal Conic projection having the following projection parameters

```
Map parallels      [50 50]
Origin             [50 -107]
Geoid              [6371200   0]
```

These (*x*,*y*) values can be reprojected back to a geographical (latitude, longitude) grid using the inverse projection.

To simplify the task of matching (*x*,*y*) grid coordinates to geographical coordinates we have included the following additional MAT files:

```
grids_ALL.mat
grids_R01.mat
grids_R02.mat
grids_R03.mat
grids_R04.mat
grids_R05.mat
grids_R06.mat
grids_R07.mat
```

```
grids_R08.mat
grids_R09.mat
grids_R10.mat
```

These files can be found in the subdirectory `RGM_archive`. The individual files contain projection results for the entire region (ALL) and for each of the subregion (R01, R02, ... R10) and each file has the same structure.

As an example, suppose one is interested in subregion 1. Loading the file `grids_R01.mat` using the MATLAB command

```
load('grids_R01.mat')
```

places the following arrays in the workspace:

Array name	Description	Size
`X`	X cartesian coordinates	`ny` × `nx`
`Y`	Y cartesian coordinates	`ny` × `nx`
`latitude`	Latitude coordinates	`ny` × `nx`
`longitude`	Longitude coordinates	`ny` × `nx`

For the cell with row-column indices (21, 31) the arrays contain

`X(21,31)`	*X* cartesian coordinate (m) of centre of cell (21,31)
`Y(21,31)`	*Y* cartesian coordinate (m) of centre of cell (21,31)
`longitude(21,31)`	Decimal longitude (deg) of centre of cell (21,31)
`latitude(21,31)`	Decimal latitude (deg) of centre of cell (21,31)