Progress report for river-routing development in C-CAM

J.M. Wilms

March 29, 2016

1 Introduction

River water inflow to the World Ocean is the most important component of the global hydrological cycle, providing a vital exchange between the ocean water and water of the continents.

Fresh water in rivers greatly differs from saline seawater in its physical and chemical properties. Mixed with seawater, the fresh water affects the general scheme of oceanic motion of water masses. Even a relatively small volume of fresh water causes a degree of desalination in the top layer; this spreads over the ocean surface over long distances and can greatly affect physical, chemical, and dynamic processes in seas and oceans (Shiklomanov [2009]).

The aim of our section of the VRESM project is to understand the underlying physics of how water will move from the continents to the ocean and to use this understanding to model the route that the water takes and thus predict not only the amount of freshwater reaching oceans, but also the locations at which such outflow will occur.

As a starting point our group has focused on a river-routing model, based on the work of J.R. Miller [1993]. This model has been built into the Conformal-Cubic Atmospheric Model (C-CAM), developed at CSIRO, Australia. During the past few months we have tried to understand not only how the river-routing works but also how it fits into the greater C-CAM model.

C-CAM differs from other climate models, which are based on a regular latitude-longitude configuration: It uses a grid derived from a gnomic projection of the cube onto the sphere. In the following section the geometry of the C-CAM grid and essential grid transformations are briefly described. This description is based on the work of McGregor [2005].

A discussion of the river-routing scheme is given. This is based on email correspondence with Thatcher [2016] as well as inspection and experimentation with the C-CAM source code. Results obtained from a spin-up simulation for the year 2000 are shown and discussed. The report is concluded with some general remarks on the spin-up procedure and possible different modelling methods which may yield more accurate routing results.

2 The geometry of the C-CAM grid

C-CAM is formulated on a quasi-uniform grid, derived by projecting the panels of a cube onto the surface of the Earth. With the exception of the eight singular vertices (i.e. the corners of the projected cube) the conformal cubic grid is isotropic.

Let the racius of the Earth be denoted by R and let (X, Y, Z) be the physical 3D Cartesian coordinates on the sphere with Z directed upwards through panel 1, normal to Earth's surface as shown in Fig. 1. Equations of motion are written in terms of 2D panel coordinates (x, y, p) where p is the panel number. Note that (x, y) still refer to the sphere. Since a cube has six faces, it follows that $0 \leq p \leq 5$ and since four panels will encompass the sphere the x and y values on each of the six panels $0 \leq x$ or $y \leq \pi R/2$. It follows that the average model grid resolution is given by

$$\Delta x \equiv \Delta y = 2\pi R/(4N) \tag{2.0.1}$$

From Eq. 2.0.1 it follows that a C48 grid has a quasi uniform resolution of 208km, since it contains 48x48 grid points on each panel.

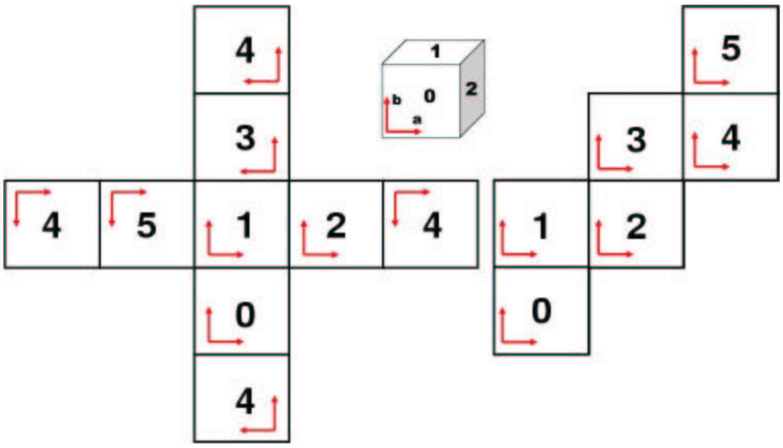


Figure 1: Layout of panels on cube.

On each panel the x-values are given by $x_1 = \pi R/4N$, $x_2 = 3\pi R/4N$, ..., $x_N = (2N-1)\pi R/(4N)$ with corresponding values for y. Each of the aforementioned x (and y) points has a map factor m associated with it. The physical spacing between grid points is $\Delta x/m$. These map factors vary from point to point.

C-CAM can be stretched if a higher resolution is desired. However, If we use an unstretched grid, the map factors on any panel are equal to those in the corresponding position of any other panel. The unstretched physical grid values on the sphere are related to the stretched factors (X', Y', Z') by means of the Schmidt transformation.

The Schmidt transformation converts any orthogonal grid system on the sphere to a stretched version of the same grid. The Schmidt magnification factor is denoted by S. For $S = 1$, the grid is unstretched, whereas higher values of S will result in a higher resolution over panel 1.

The (X, Y, Z) coordinate system is centred on panel 1. At the centre of the panel, the Z-coordinate is directed upwards while X and Y are directed south and east respectively. The (x, y, p) values are unaffected by the Schmidt transformation, but the associated map factors are altered according to

$$m = \frac{1 + \alpha Z'/R}{1 - \alpha} \frac{m'}{S}. \tag{2.0.2}$$

At the centre of panel 1, Z=Z'=R, Eq.2.0.2 reduces to $m = Sm'$, verifying that the grid resolution on the (X, Y, Z)-coordinates is magnified there by a factor S compared to the (X', Y', Z') coordinates.

The transformations from the physical (X, Y, Z) coordinates to the quasi-regular stretched (X', Y', Z') are given by

$$X' = \frac{1 - \alpha}{1 - \alpha Z/R} SX \tag{2.0.3}$$

$$Y' = \frac{1 - \alpha}{1 - \alpha Z/R} SY \tag{2.0.4}$$

$$Z' = \frac{Z - \alpha R}{1 - \alpha Z/R}. \tag{2.0.5}$$

(2.0.6)

where

$$\alpha = \frac{S^2 - 1}{S^2 + 1} \tag{2.0.7}$$

However, in the current applications, the grid remains unstretched and it follows that $S = 1$ and $(X', Y', Z') = (X, Y, Z)$.

Gnomonic projections (radially towards the centre of the Earth) are performed between the surface of the Earth and the panels of the cube. Let (A, B, C) denote 3D Cartesian coordinates on the cube with A, B, and C pointing outwards through panels 0,2, and 1 respectively and let (a, b, p) be 2D Cartesian on the panels of the cube as shown in Fig. 1. Each of these a and b coordinate values can range between $-R$ and R, whereas $0 < p < 5$. Each panel edge has N segments and each panel contains $N \times N$ four-sided elements. The (x, y) coordinates, mentioned earlier, are coordinates on the sphere and are orthogonal, whereas the projected (a, b) coordinate lines on the faces of the cube are not orthogonal.

It follows that each C-CAM grid point has an associated (X, Y, Z) value and an associated (X', Y', Z') value, and that projection onto the cube has an associated (A, B, C) value which corresponds to (a, b, p) values.

2.1 Grid transformations

2.1.1 (X, Y, Z) on sphere to (A, B, C) on cube

$$(A, B, C) = \frac{(X, Y, Z)}{max(|X|, |Y|, |Z|)} R. \tag{2.1.1}$$

$$(X, Y, Z) = \frac{(A, B, C)}{(A^2 + B^2 + C^2)^{1/2}} R. \tag{2.1.2}$$

2.1.2 (A, B, C) on cube to (a, b, p) on cube

$$\begin{aligned}
\text{if } A = R \quad & \text{then} \quad p = 0, a = B, b = C \\
\text{if } A = -R \quad & \text{then} \quad p = 3, a = -C, b = -B \\
\text{if } B = R \quad & \text{then} \quad p = 2, a = -C, b = -A \\
\text{if } B = -R \quad & \text{then} \quad p = 5, a = A, b = C \\
\text{if } C = R \quad & \text{then} \quad p = 1, a = B, b = -A \\
\text{if } C = -R \quad & \text{then} \quad p = 4, a = A, b = -B.
\end{aligned} \tag{2.1.3}$$

It can be verified by inspection that the following alternative equations produce equivalent results, but avoid the branching computations. The panel index, p, is then given by,

$$p = max\left[A_{int}(3A_{int-3}), B_{int}(7B_{int} - 3), C_{int}(5C_{int} - 3)\right]/2, \tag{2.1.4}$$

where

$$\begin{aligned}
A_{int} &= int\left(\frac{A}{R}\right) \\
B_{int} &= int\left(\frac{B}{R}\right) \\
C_{int} &= int\left(\frac{C}{R}\right)
\end{aligned} \tag{2.1.5}$$

and $int(I)$ is the truncated integer value. The Cartesian coordinate position (a, b, p) on a panel of the cube is then

$$a = E_{p1}A + E_{p2}B + E_{p3}C \tag{2.1.6}$$

$$b = F_{p1}A + F_{p2}B + F_{p3}C \tag{2.1.7}$$

where the nonzero members of matrices E and F are given by

$$\begin{aligned}
E_{02} = E_{12} = -E_{23} = -E_{33} = E_{41} = E_{51} \\
= F_{03} = -F_{11} = -F_{21} = -F_{32} = -F_{42} = F_{53} = 1.
\end{aligned} \tag{2.1.8}$$

2.1.3 (a, b, p) on cube to (A, B, C) on cube

$A = R$	$B = a$	$C = b$	for $p = 0$
$A = -b$	$B = a$	$C = R$	for $p = 1$
$A = -b$	$B = R$	$C = -a$	for $p = 2$
$A = R$	$B = -b$	$C = -a$	for $p = 3$
$A = a$	$B = -b$	$C = -R$	for $p = 4$
$A = a$	$B = -R$	$C = b$	for $p = 5$

2.1.4 X, Y, Z on sphere to/from longitude-latitude

Denote longitude by γ and latitude by ρ in polar coordinates. Denote unrotated Cartesian coordinates by $(\tilde{X}, \tilde{Y}, \tilde{Z})$ with $\tilde{Z}$ passing through the Norh pole and $\tilde{X}$ passing through the Equator at the Greenwich meridian. These can be converted to and from polar coordinates by

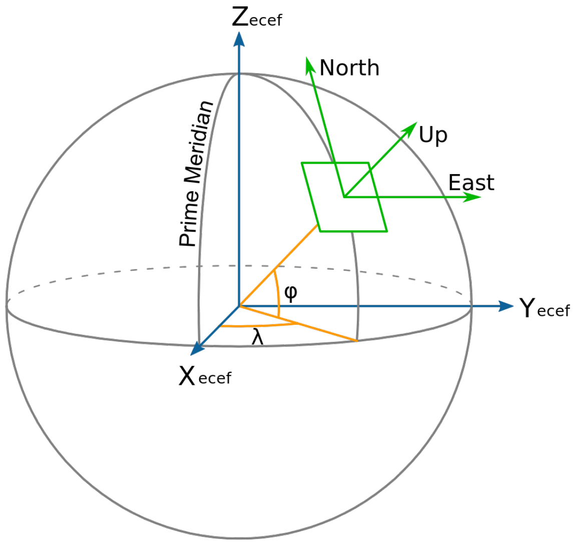


Figure 2: Conversion between lat-lon and Cartesian coordinates.

$$\tilde{X} = R\cos\phi\cos\gamma \tag{2.1.9}$$
$$\tilde{Y} = R\cos\phi\sin\gamma \tag{2.1.10}$$
$$\tilde{Z} = R\sin\phi \tag{2.1.11}$$
$$\tag{2.1.12}$$

It also follows that

$$\gamma = \tan^{-1}\frac{\tilde{Y}}{\tilde{X}} \tag{2.1.13}$$
$$\phi = \sin^{-1}\frac{\tilde{Z}}{R} \tag{2.1.14}$$

The CCAM grid is, however, set up so that the Cartesian grid is located at (γ_0, ϕ_0) for $(X, Y, Z) = (0, 0, R)$. The X-axis lies in a negative direction along $\gamma = \gamma_0$ and the Y-axis lies along $\phi = \phi_0$. The coordinate rotation corresponds to rotating $(\tilde{X}$,$\tilde{Y}$,$\tilde{Z})$ by γ_0 and then by $-\phi_0$. The rotated and non-rotated coordinates are related by

$$\tilde{X} = \cos\gamma_0\sin\phi_0 X - \sin\gamma_0 Y + \cos\gamma_0\cos\phi_0 Z, \tag{2.1.15}$$
$$\tilde{Y} = \sin\gamma_0\sin\phi_0 X + \cos\gamma_0 Y + \sin\gamma_0\cos\phi_0 Z, \tag{2.1.16}$$
$$\tilde{Z} = -\cos\phi_0 X + sin\phi_0 Z \tag{2.1.17}$$

The Grid setup and rotations are calculated in C-CAM's $utilities.f90$, $setxyz.f90$, and $latloij.f90$.

3 Grid decomposition for parallel processing

Face decomposition is used to decompose the grid. This methods keeps the cube faces together where possible; for example with 6 processors each gets a complete face. The computational domain is divided into a number of subdomains equal to the number or processors.

Each processor performs calculations for a subdomain. Although the subdomains do not overlap, there is a buffer of halo points added to their boundaries, including the virtual boundaries, i.e. boundaries between neighbouring processors. The bounds subroutine updates the grid halo information with adjacent MPI processes/CPUs.

4 River-routing module in C-CAM

All the river routing is in C-CAM's $river.f90$. The main prognostic array is $watbdy(:)$, which is the amount of water (kg/m^2) in a grid box. The scheme calculates the slope based on the surface geopotential, zs, where

$$zs = z * g, \tag{4.0.18}$$

and z is the orography, whereas gravity is denoted by g. The surface height output of C-CAM is shown in Fig. 3. The slope, i, calculation is adjusted using Z_{adj}, based on the water-body height, so that water attempts to equilibrate with the water level in adjacent grid boxes:

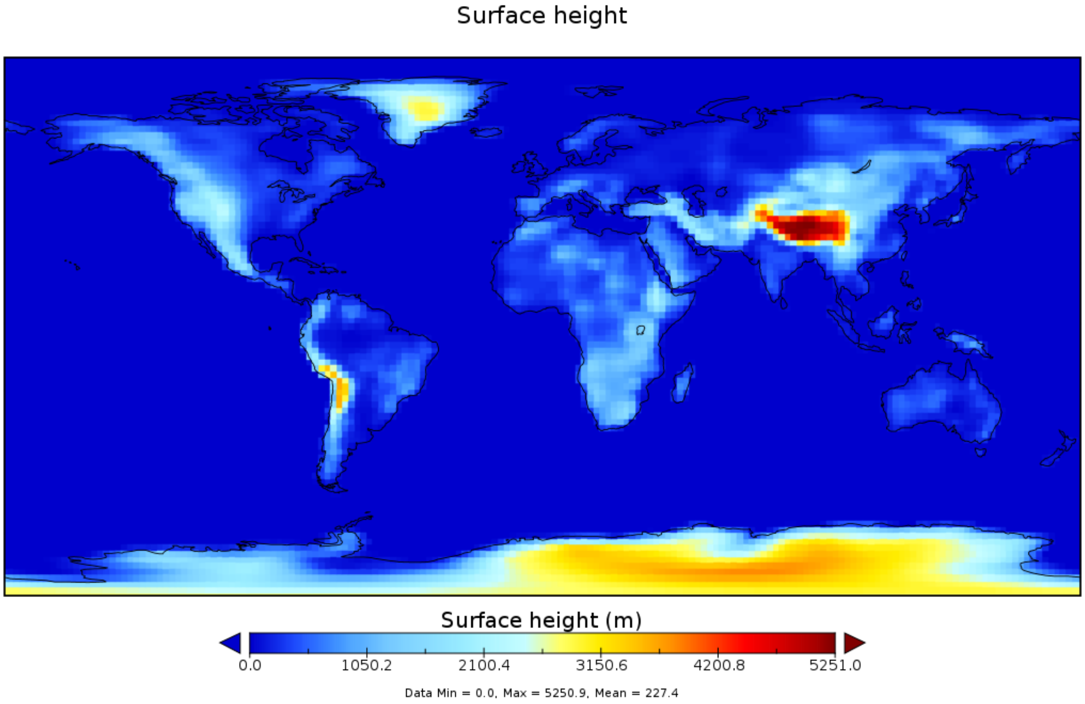


Figure 3: Surface height,z

$$i = \frac{Z_C + Z_{C,adj} - (Z_N + Z_{N,adj})}{r_{CN}}. \tag{4.0.19}$$

In Eq. 4.0.19, r_{CN} denotes the shortest distance between two points on the sphere and is given by Todd [2016] as

$$r_{CN} = cos^{-1}(\boldsymbol{C} \cdot \boldsymbol{N}), \tag{4.0.20}$$

where $\boldsymbol{C}$ and $\boldsymbol{N}$ are the (X, Y, Z) coordinates, normalised by Earth's radius, R. However, the result is that water cannot build-up in a single grid point because eventually it flows back out of the grid point.

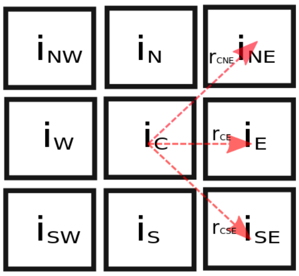


Figure 4: Centre Cell and its neighbours.

In this way, the C-CAM code is more of a flood model than a river model. Here C refers to the cell from which the slope is calculated. This slope calculation is repeated for each of cell C's eight neighbours, shown in Fig. 4.

Once the slope is calculated, the downhill velocity (outflow) and uphill velocity (inflow) is calculated, using an expression given by J.R. Miller [1993]

$$\boldsymbol{u} \quad = \quad 0.35\left(\frac{i}{i_0}\right), \tag{4.0.21}$$

where $i_0 = 0.00005$ is the reference topography gradient.

The downhill velocity and uphill velocities are used to estimate a flux for incoming and outgoing water and the new water level, *newbdy*, is updated.

To prevent overshoot and oscillations, Z_{adj} is recalculated, using *newbdy*:

$$Zadj = \frac{1}{\rho}\frac{1}{2}(newwat + watbdy), \tag{4.0.22}$$

where $newwat$ is the estimate of the water depth at $time = t+1$, whereas the $watbdy$ is the water depth at time=t. The factor of a half, on the right of Eq. 4.0.22, serves to average the water depth between time=t and time=$t+1$ and ρ is the density of water.

The flux calculation is then repeated. This forms a predictor/corrector loop where an estimate of the water flow is used for the initial calculation and the process is repeated with improved fluxes. This predictor/correction loop is only required because Z_{adj} depends on the water level.

The last stage is to handle basins where water flows in to a land grid box from all sides. This is problematic since the water needs to be removed. Some groups spread the 'trapped' water over the oceans, but that requires communications will all CPUs which would limit the speed that C-CAM can run at on high CPU counts (e.g., 10,000+). So, instead, in C-CAM, the trapped water is removed by adding the water back to the land-surface model as soil moisture. This option is activated by setting the switch $basinmd = 0$ in the run script. The land surface model hydrology cannot really handle this because the soil column is only 4 m deep, but at least the water can evaporate. Because of the Z_{adj}, the trapped water will eventually fill the basin and start being trapped in adjacent grid boxes, which increases the area that water can evaporate from. There are other basinmd options for simply storing the water in the trapped basin forever ($basinmd = 1$) or allowing water to 'leak' into the soil at all grid boxes instead of just basins ($basinmd = 2$).

5 Model spin-up

The water-body variable, $watbdy$ is zero for the very first month. It could be initialised with something else if data is available. Currently, the model spins up the rivers as it runs. However, C-CAM will save $watbdy$ in the Restart file at the end of each month, so the data is not lost. The water body value, $watbdy$ is loaded from the restart file in $onthefly.f90$ and saved to the restart file in $outcdf.f90$. It follows that, if the initial month is modified to contain some $watbdy$ data, the model could be initialised with non-zero values.

Model spin-up is an adjustment process where its internal stores move from an initial state of unusual conditions to one of equilibrium. Model outputs during this spin-up process are often unrealistic and misleading. It is therefore crucial that the model be initialised with an adequate spin-up time simulation.

The *watbdy* output is given by C-CAM as *surface water* in the variable *swater*. For a spin-up run the year 2000 was chosen. The results for January 2000 to December 2000 are shown in Fig. 5. Runoff results are shown in Fig. 6 and the average and maximum levels for the first day of each month are shown in Fig. 7 and 8

6 Conclusion

From Fig.'s 7 and 8, it appears as though the spin-up time needs to be increased for a stable input scenario to be obtained. Also, according to Shiklomanov [2009], the total annual water inflow from rivers to the World Ocean was 39 530 $km^3/year$ on average for the period 1921 to 1985. More than half of this amount (20 190 km^3) was discharged to the Atlantic ocean; the least water inflow occurs to the Arctic and Indian oceans (4280 km and 4530 km, respectively). About 40% of global river runoff to the ocean occurs within the equatorial zone, between 10°N and 10°S. Freshwater runoff from the territory of Antarctica as icebergs and melt water equals 2310 km^3/year.

From Fig. 8, it appears that the simulation results fall well below Shiklomanov [2009]'s measured values.

The river-routing scheme differs from traditional river routing schemes in that there may exist multiple outflows from a grid box. Sub-grid orography has not been accounted for because the scheme was intended for finer grid spacing. That information could be estimated in a pre-processing step and fed as a new input into the model at a later stage.

Meandering terms that slow down the water movement over a grid box have not been factored in. The water could also be localised within a grid box as the river usually occupies a small area fraction of the grid box. Finally, evaporation of the river could be estimated if water temperature was tracked, although that is not really done by the current generation of river routing schemes in GCMs.

References

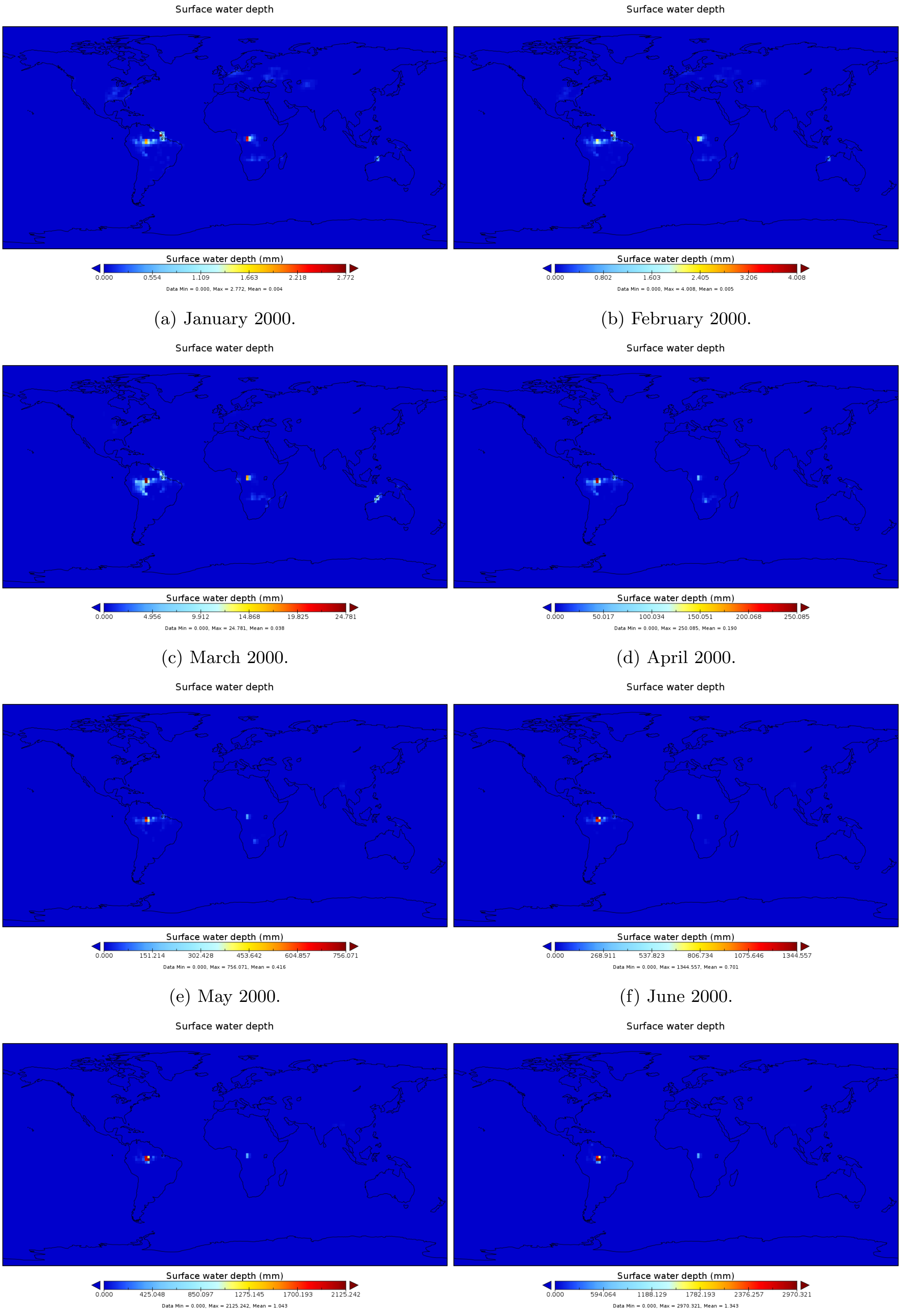
G. Caliri J.R. Miller, G.L. Russell. Continental-scale river flow in climate models. *Journal of Climate*, 7:914–926, 1993.

J.L. McGregor. C-cam: Geometric aspects and dynamical formulation. Technical report, CSIRO, 2005.

I.A. Shiklomanov. *Hydrological Cycle*, volume 2. Eolss Publishers Co Ltd, 2009.

M. Thatcher. private communication, 2016.

R. Todd. Spherical distance. `http://mathworld.wolfram.com/SphericalDistance.html`, 2016. Accessed: 2016-03-27.



(a) January 2000.

(b) February 2000.

(c) March 2000.

(d) April 2000.

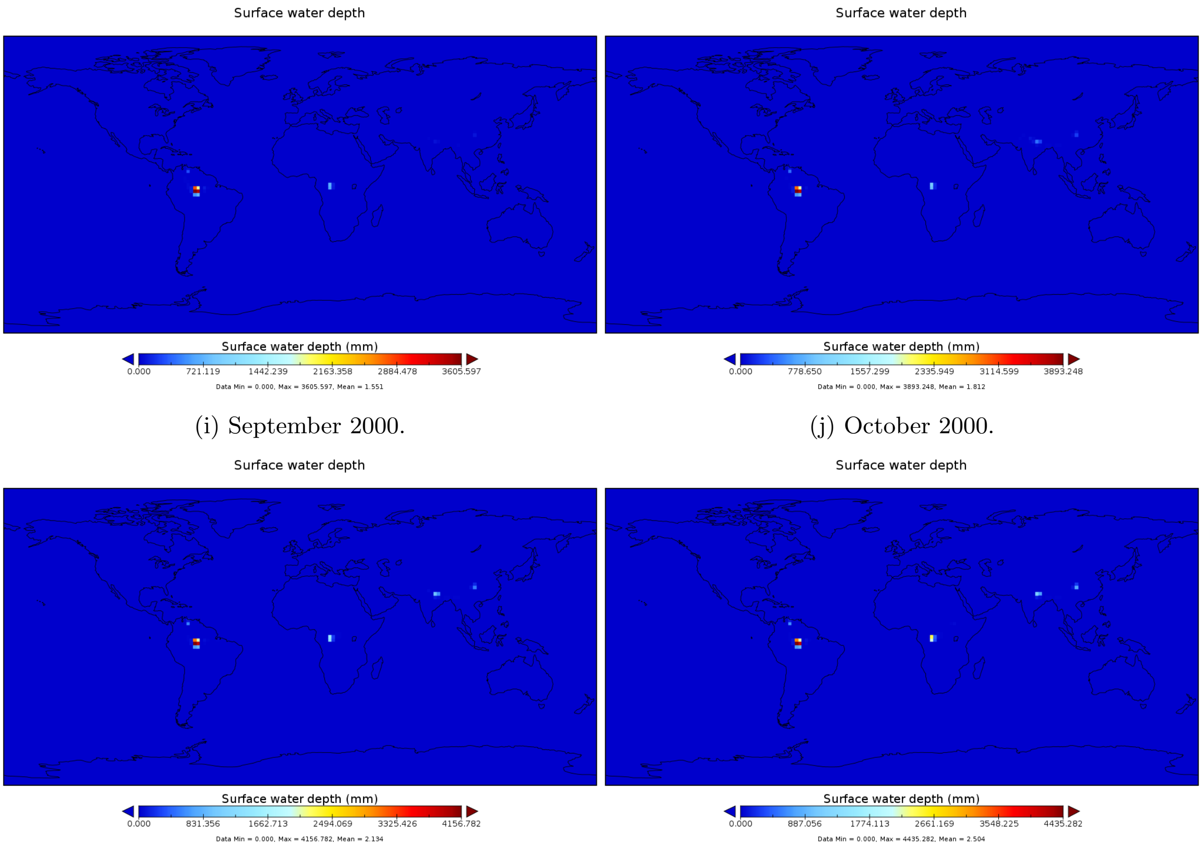
(e) May 2000.

(f) June 2000.

(g) July 2000.

(h) August 2000.

Figure 5: Surface water height in mm for spin-up run during 2000.



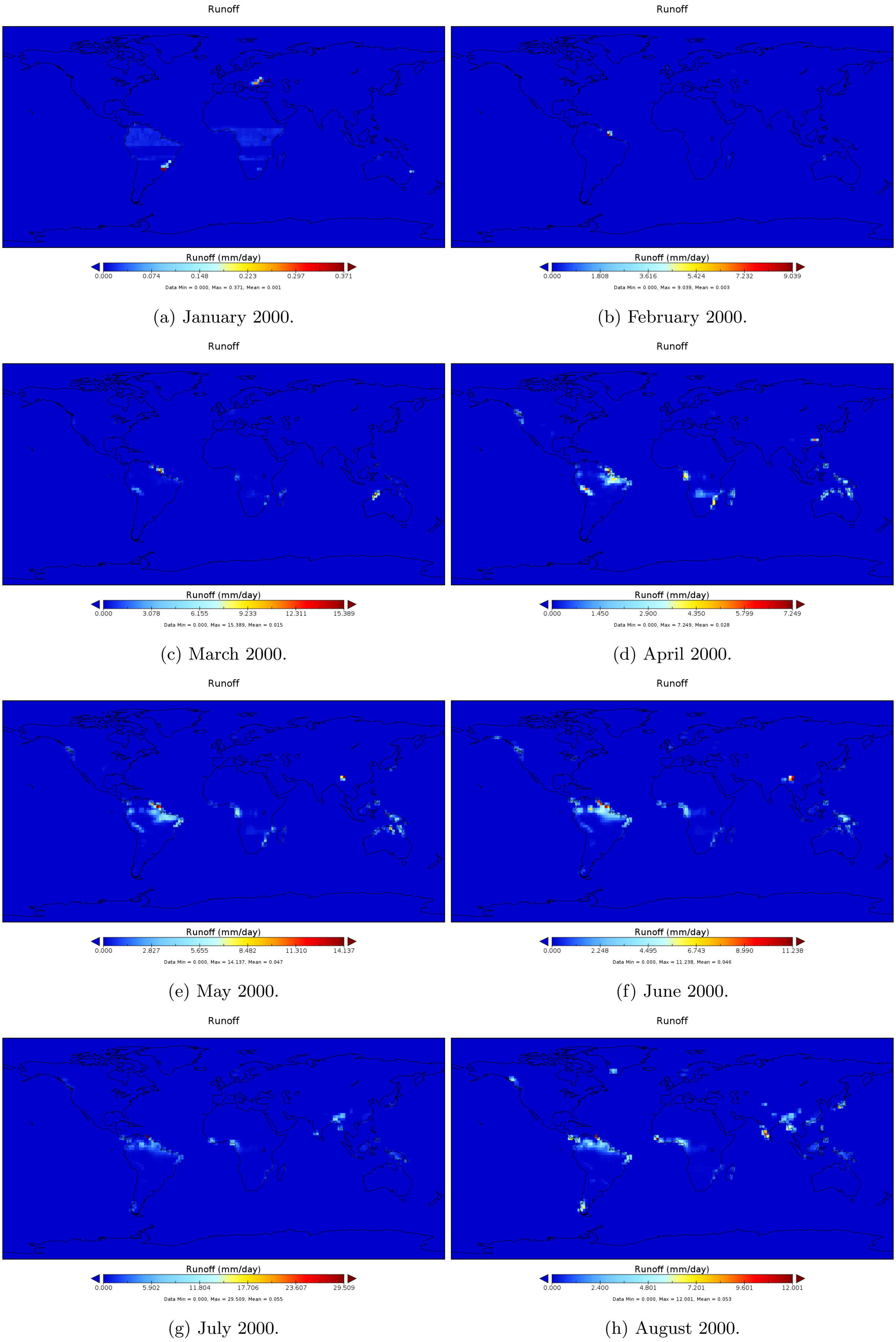
(i) September 2000.

(j) October 2000.

(k) November 2000.

(l) December 2000.

Figure 5: Surface water height in mm for spin-up run during 2000.



(a) January 2000.

(b) February 2000.

(c) March 2000.

(d) April 2000.

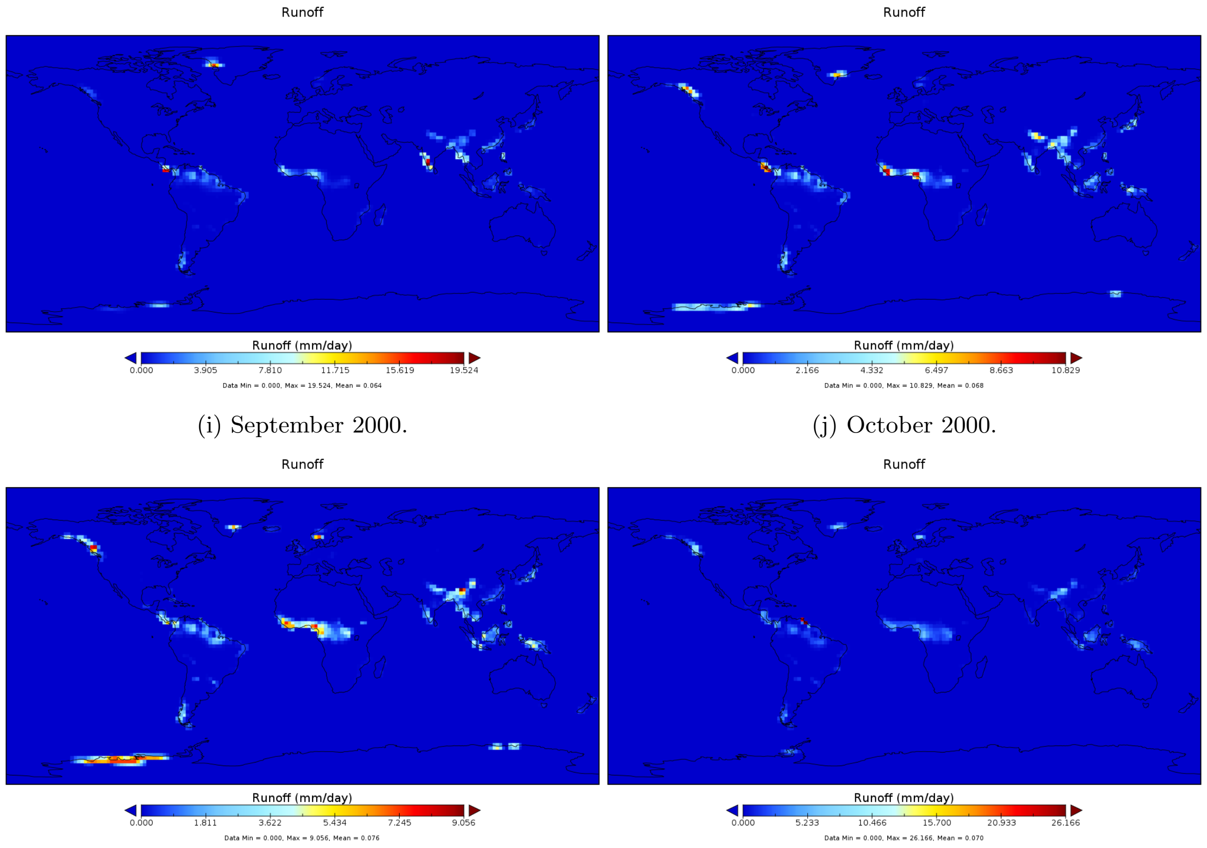
(e) May 2000.

(f) June 2000.

(g) July 2000.

(h) August 2000.

Figure 6: Surface water height in *mm* for spin-up run during 2000.



(i) September 2000.

(j) October 2000.

(k) November 2000.

(l) December 2000.

Figure 6: Runoff velocity in mm/day for spin-up run during 2000.

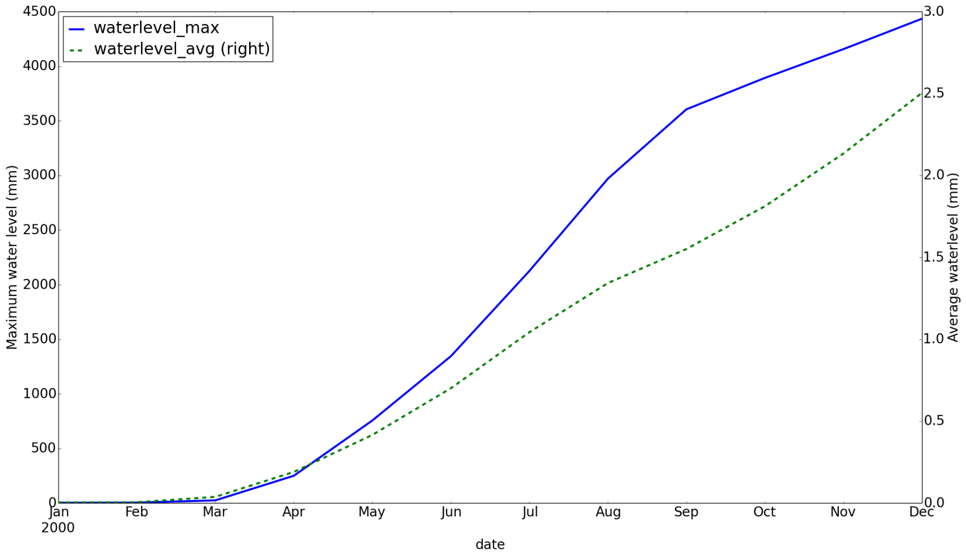


Figure 7: Surface water height average– and maximum values.

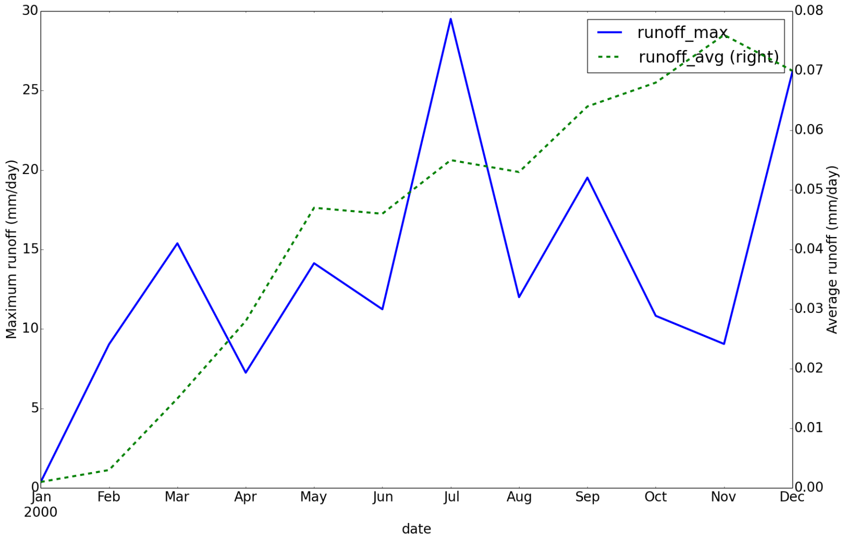


Figure 8: runoff speed average– and maximum values.