Planet Generating Program

Torben Ægidius Mogensen

July 22, 2024

Abstract

The program generates zoomable planet maps based on recursive spatial subdivision of a tetrahedron containing the globe. The output is a bitmap file, with a choice of different formats, different colour maps and different map projections.

1 Web usage

A web interface can be found at `https://topps.diku.dk/torbenm/maps.msp`. This limits the size of bitmaps and it does not support all of the options of the command-line version, see below, but it requires no installation and it is easier to use. The web interface is mostly self-explanatory, but for more details of the workings of the different options, see the explanations for the command-line options below.

2 Installation of command-line version

The program and a number of files (including this manual) are provided in a Zip file called `planet.zip`. Extract this using e.g. WinZip (on Windows) or unzip (linux/unix). Most browsers support extraction directly from the browser.

A makefile is provided for compiling the program, so if you have make installed on your computer you can just write "make all" to compile the program. If not, you can use the following simplified method:

How to compile the program:

On a unix platform, just write

```
gcc planet.c -o planet -lm -O2
```

On a Windows machine, a similar method can be used, but the executable should be names "planet.exe" instead of just plain "planet". The Zip file contains a Windows executable cross compiled from Linux using the command

```
x86_64-w64-mingw32-gcc -lm -O2 planet.c -o planet.exe
```

This has been tested using Wine (a Windows emulator for Linux). It gives a few warnings, but it seems to work anyway.

Note that the program assumes that floating-point numbers are IEEE format double precision. If you compile for a platform that uses a different format, the program should still work, but your maps will look different.

Enquiries and error reports can be sent to `torbenm@di.ku.dk`.

3 Command-line usage

The program is designed to be used from the command line, using a multitude of options.

Usage: `planet [options]` (Linux) or `planet.exe [options]` (Windows). For Windows use, I suggest installing PowerShell.

Do not start the program just by clicking the program icon! This will just fill your screen with seemingly random characters.

The command-line options are:

-s seed	Specifies seed as number between 0.0 and 1.0
-w width	Specifies width in pixels, default = 800
-h height	Specifies height in pixels, default = 600
-m magnification	Specifies magnification, default = 1.0
-o output-file	Specifies output file, default is standard output
-l longitude	Specifies longitude of centre in degrees, default = 0.0
-L latitude	Specifies latitude of centre in degrees, default = 0.0
-g gridsize	Specifies vertical gridsize in degrees, default = 0.0 (no grid)
-G gridsize	Specifies horisontal gridsize in degrees, default = 0.0 (no grid)
-i init-alt	Specifies initial altitude (default = -0.02)
-c	Colour depends on latitude (cumulative, default: only altitude)
-n	Apply non-linear scaling to altitude. This makes land flatter near sea level
-S	Make more "wrinkly" maps
-C file	Read colour definitions from file
-O	Produce a black and white outline map
-E	Trace the edges of land in black on colour map
-B	Use "bumpmap" shading
-b	Use "bumpmap" shading on land only
-d	Use "daylight" shading
-a angle	Angle of "light" in bumpmap shading or longitude of sun in daylight shading
-A latitude	Latitude of sun in daylight shading
-P	Use PPM file format (default is BMP)
-x	Use XPM file format (default is BMP)
-H	Output heightfield (default is BMP)
-M delta	Read map from standard input and match new points to map if edge length greater than delta (default = 0.1)
-V number	Distance contribution to variation (default = 0.035)
-v number	Altitude contribution to variation (default = -0.45)
-T lo la	Rotate map so what would otherwise be at latitude la and longitude lo is moved to (0,0). This is different from using -l and -L because this rotation is done before applying gridlines and latitude-based effects.
-z	Show biomes
-Z file	read biome colours from file
-R	Print version info
-pprojection	Specifies projection:
	m = Mercator (default)
	p : Peters
	q : Square
	s : Stereographic
	o : Orthographic
	g : Gnomonic
	a : Area preserving azimuthal
	c : Conical (conformal)
	M : Mollweide
	S : Sinusoidal
	h : Heightfield (obsolete. Use -H option instead)
	i : Icosahedral

The options will be explained in more detail below.

The order of the options is irrelevant, except that later options override earlier ones. The projection letter can follow immediately after "-p" (as in "-pM") or they can be separated by space (as in "-p M").

As an example, to generate a stereographic projection of a planet generated from seed 0.3 with magnification 3 centered around 30° north, 35° east with gridlines every 10 degrees on latitudes and every 20 degrees on longitudes, outputting the result as a 320x256 bitmap in the file p.bmp, use the command

```
planet -ps -s 0.3 -m 3 -L 30 -l 35 -G 10 -g 20 -w 320 -h 256 -o p.bmp
```

The initial altitude (-i) option specifies the initial altitude of the corners of the box, before subdivison. Increasing this will increase the amount of land on the planet and vice-versa. The default value is -0.015, which gives a slight preference for water. Changing this value by +/- 0.01 will change the water percentage by 5-15% (depending on the planet in question). If you, for example, want a planet with 70% water, you can make a world map using the

-pp option (which prints the water percentage, see below) and adjust the initial altitude using the -i option until the indicated water percentage is 70.

The -V and -v options control the contribution of distance and altitude difference to altitude variation. The higher these numbers are, the more ragged your landscape will be. The -V number controls how much altitude changes by distance and the -v number controls how much the steepness of terrain affects roughness. The idea is that level terrain tends to be less rugged than sloping terrain. You shouldn't let the values vary by more than a factor of 2 from the default values.

The -B option activates bumpmap shading. This shades the colour depending on the angle, creating more "3-dimensional" looking maps. The -b option activates bumpmap shading for land only (leaving sea smooth). The -a option can change the angle of "light" for bumpmap shading. The -d option activates daylight shading, which means that shading is done relative to a sun positioned at a longitude/latitude specified by the -a and -A options, creating a dark night side and a light dayside and visible bumps near the terminator. Note that using bumpmaps will make the map use full 24-bit colours, irrespective of the number of colours specified in the colour scheme (see below).

The -E option traces the outline at the land/sea border in black, but depends on the colour scheme), and -O shows only this outline (and contour lines, if these are specified), making both land and sea white. These maps are output as 1bpp bitmaps, making them compact, but you can't use bumpmaps. Use the white colour scheme (see below) for that. Both -E and -O can optionally draw additional contour lines at other altitudes. This is enabled by following the -E or -O (with no space in between) with an integer that specifies the number of contour lines. With the -E option, contour lines alternate between two colours. Contour lines are slightly thinner than the sea/land outline. Note that the colours of contour lines can be changed by the colour scheme. If a negative number of contour lines is specified, this indicates the number of contour lines at sea, each spaced at 5% of the difference between sea level and deep sea. This can to some extend emulate the tradition of drawing 1–3 thin lines outside coasts on hand-drawn maps, but whereas these are traditionally spaced equally by distance from the coast, the contour lines are spaced by water depth. By using -E or -O twice with positive and negative parameters, you can get contour lines both on land and sea.

The -C option specifies a file from which colour definitions are read. The format of a colour file is a sequence of lines each consisting of four integers:

colour-number red green blue

where $0 \leq$ colour-number ≤ 65535 and $0 \leq$ red, green, blue ≤ 255.

The colour numbers must be increasing. The first colours have special uses:

0 is usually black (0,0,0)
1 is usually white (255,255,255)
2 is the background colour
3 is used for latitude/longitude grid lines
4 and 5 are used for outlines and contour lines
6 upwards are used for altitudes

Halfway between 6 and the max colour is sea level. The shallowest sea is coloured with (max+6)/2, and land with colours above this. With 65536 colours, (max+6)/2 = 32770. With 256 colours, (max+6)/2 = 130. Colours between specified colour numbers are interpolated.

An example colour file (`default.col`) is shown below

```
0 0 0 0
1 255 255 255
2 255 255 255
3 0 0 0
4 0 0 0
5 255 0 0
6 0 0 255
130 0 128 255
131 0 255 0
200 64 192 16
250 128 128 32
255 255 255 255
```

This uses 256 colours. The background (2) is set to white and grid lines (3) to black. Outlines are also black, but if more contour lines are used, every other is red. Colour 6 is deep blue, for deep sea. This changes to a light blue at colour 130 (shallowest sea), then at colour 131, a light green is used, which changes to a darker, yellowish green

at colour 250, then to brown at colour 250 and white at colour 255. Note that the change between sea and land occurs at (6+255)/2 = 130.

Some predefined colour schemes are provided:

- Olsson.col is a colour map inspired by a map program made by Olsson. This is now the default colour map (replacing `default.col`).

- OlssonLight.col is a variant of Olsson.col using lighter colours.

- default.col (described above) is the previous default colour map.

- defaultB.col is like default.col, except that the background colour is black.

- burrows.col is an alternative colour scheme provided by Jim Burrows, supposed to be more like what is used in atlases.

- burrowsB.col is like burrows.col, except that the background colour is black.

- Bathymetric.col imitates a colourmap used in some old atlases.

- Lefebvre.col gives a pale, atlas-like colouring suitable for adding text and symbols on top.

- Lefebvre2.col is another contribution by Olivier Lefebvre.

- mars.col is supposed to give colouring similar to Mars. You can probably do better yourself.

- wood.col is supposed to give a wood-like texture. The effect is not convincing.

- white.col makes land and sea uniformly white. This may sound uninteresting, but if you add outlines and bumpmaps or contour lines you can get maps that work well in black-and-white print.

- greyscale.col uses shades of grey to represent altitudes.

- Blackbody.col is based on black-body radiation (heat maps).

- 2col.col uses blue for sea and green for land.

The "-H" option produces a heightfield instead of a bitmap file. Each point on the map is printed as an integer in the range (approximately) -1.2×10^6 to 1.2×10^6, representing the altitude at that point. Points outside the planet are given height 0. Options controlling colours and gridlines are ignored. This option is intended to produce input to programs that, e.g., show 3D views of small areas of the planet. The obsolete "-ph" option corresponds to "-po -H".

3.1 Projections

The available projections are briefly described below. See `http://www.quadibloc.com/maps/mapint.htm` for more details.

The Mercator, Peters and square projections project the globe onto a cylinder wrapped around the equator of the globe. The cylinder is then unfolded to give a flat map. Hence, lines of equal longitude or latitude map to vertical or horisontal lines respectively.

The Mercator projection preserves angles (it is a conformal mapping) and compass directions but distorts area quite heavily towards the poles. The map extends infinitely north/south.

The Peters (Gall-Peters) projection project a point horisontally out to the cylinder (which is afterwards strected by a factor of 2). It preserves relative area, but not angles or compass directions (expect for directly east, west, north or south). Areas near the equator will appear stretched up/down (by a factor of 2 at the equator) whereas areas near the poles are flattened. At latitude 45 degrees, shapes have approximately the right proportions. When the Peters projection is specified, the water percentage is printed on the stderr stream. This is mainly interesting if a full-world map is generated (magnification = 1).

The square projection projects latitudes equidistantly. This projection preserves neither area nor angles, but it preserves distances in the vertical direction.

The Mollweide projection maps the earth to an ellipse. It preserves area. It is mainly used for world-maps and is best viewed with the -g option, as longitudes are strongly distorted.

The sinusoidal projection maps the globe onto 12 sinusoidal "slices". The projection preserves area as well as distance from equator. A globe can be made from this projection by folding the map into a cylinder and bending the slice tops/bottoms inwards.

The icosahedral projection maps the globe onto the faces of an unfolded icosahedron (as used in maps for the Traveller game), using the gnonomic projection for each face. There are tiny discontinuities where the faces meet, so you shouldn't zoom too much at these edges.

Mercator, Peters, square, Mollweide, icosahedral and sinusoidal maps at magnification 1 are scaled to fit the Width. Except for the Mercator projection (which has infinite height), icosahedral and Peters, a full world map using these projections is twice as wide as it is high. Icosahedral maps are 2.117 times as wide as high. With the Peters projection, the width is pi/2 times the height, so a map 628 pixels wide needs 400 pixels height to fit the full map. If the -L option (center latitude), the Mollweide projection rotates the planet to center at this latitude, but the rest of these projections just shifts the map up or down, so the standard meridian is still the equator.

Assuming an Earth-sized planet of circumference 40000km we can relate this to scaling factors. To find the equatorial scaling factor, divide 4×10^9 by the width of the map (in centimeters) and then by the magnification factor. Hence, a map of width 20cm at magnification 2 has a scale of $1 : 1 \times 10^8$ at equator. For the Mercator projection, the scale at latitude L is 1/cos(L) times the equatorial scale. Hence, a map of width 20cm and magnification 2 has scale $1 : 1.416 \times 10^8$ at latitude $45°$ and scale $1 : 2 \times 10^8$ at latitude $60°$.

A number of azimuthal projections are available. An azimuthal projection projects the globe onto a plane that touches the globe at the specified longitude and latitude. Azimuthal projections are approximately accurate at the centre of the map, but distort areas away from this. The different azimuthal projections distort in different ways:

Stereographic projection projects a point on the surface by following a line that starts in the point on the globe opposite the point that touches the plane and goes through the point of interest, until this line hits the plane. It preserves angles (i.e. local shape) but neither area nor compass directions. The entire globe is mapped onto the infinite plane, so you can never get a full world map. The distortion will be very severe when more than a hemisphere is shown.

Orthographic projection projects at right angles to the plane. It preserves nothing of interest, but is what you would see if you view the planet from space (from infinite distance). It can never show more than a hemisphere (which is mapped to a circle disk).

Gnomonic projection projects along a line starting in the center of the globe. It has the property that great circles are mapped to straight lines, which makes it useful for navigating by sea or air. A hemisphere maps to the infinite plane, so it can never show all of a hemisphere and distortion gets severe when you get more than 60 degrees away from the centre.

Area preserving azimuthal projection preserves relative areas, but not shapes or angles etc. It maps the entire globe onto a circle of a radius twice that of the globe, but distortion is severe away from the centre.

The orthographic projections are scaled so the diameter of the planet at magnification 1 is the full Height of the picture. Stereographic, gnomonic and area preserving azimuthal projections use the same scale as orthographic in the center of the picture. To find the scaling factor (again assuming Earth-sized planets) at the centre of the picture, divide 1.273×10^9 by the height of the map (in centimeters) and the magnification factor. Hence, a map of height 20cm with magnification 2 will have a central scale of $1 : 3.18 \times 10^7$. This is independent of latitude. To obtain a scale of 1:S, multiply 1.273×10^9 by the height of the map (in cm) and divide by S to obtain the magnification factor to use with the program. To get, e.g., $1 : 10^7$ on a map of height 20cm, use -m 2546. As the size of the map in cm depends on how many pixels per cm the output media uses, the program can not calculate actual scales.

The conical projection is a conformal (angle preserving) projection from the sphere onto a cone that touches the specified latitude. The cone is cut opposite the specified longitude and laid flat, so it appears like a pie section. For latitudes close to 0 or $90°$, the conical projection approaches the Mercator and stereographic projections (respectively). If the extremes are specified, these will be used instead of the conical projection. The scale at the center of the map is the same as for ortographic and stereographic projections. This projection is good for mapping areas that have large east-west extent and are not close to the equator (such as Canada or Russia). Using conical projection, all of the planet is mapped onto a finite section of a cone, so you can make complete world maps. It is, however, not really suited for this, as latitudes far away from the specified latitude are very distorted.

Regardless of the projection used, maps that show only small sections of the planet will have little distortion and the scale will differ very little across the map.

3.2   Match feature

The -M delta option specifies a number delta (in the range 0.03 to 1.0, default 0.1). A map sketch must be given on standard input. This map is 12 lines of 24 characters. The characters are:

.	very deep water
,	deep water
:	water
;	shallow water
-	sea level
	low land
o	land
O	high land
@	very high land

An example of a map specification (roughly representing Earth) is shown here:

```
;:::----ooo;;*;;;;;-;;;;;
-*****---*-:-o**oooooo**
;;:-o***-::-**oO0ooo*--:
,,,,;Oo*::::*;;-*0@0o-;,,
,,,:;*-;::;ooo**-**-::,,
,,,,,;-*-::**oo-:;--;::,
:,,,,,-oo*,:;o*;:,;--*-;
;,:::::0*;,:;o;;::::-oo-;
;,,:::::o;,::,;:;;;;;;*:-
:::::::-;;:::,::;;:::::;
;::;;;-*:::;;;;-------;
***********************
```

It can also be found in the file `earth.map`.

The character positions correspond to grid points at 15-degree spacings, both horisontally and vertically, corresponding to a square projection of a full world map, no matter which projection or magnification is used for the generated map.

The match feature is implemented so the first few subdivisions (while the edge length is greater than delta) do not use random mid-point displacement, but instead chooses midpoint altitudes based on the nearest point on the map sketch. What this means is that the overall shape of the generated map will be similar to the map sketch, but you will not get an exact match, and the seed given to the program affects the finer details of the generated map. The larger delta is, the less like the sketch the map will be. A delta smaller than 0.03 will give a pixelated map, as more points are forced to equal the pixelated sketch.

4 Biomes

Option -z creates a biome map. On sea, it uses the same colours as in the specified color map, but on land, regardless of the specified colour map, the following colours are used to specify different biomes:

	Ice
	Tundra
	Grasslands
	Taiga / boreal forest
	Desert
	Savanna
	Temperate forest
	Temperate rainforest
	Xeric shrubland and dry forest
	Tropical dry forest
	Tropical rainforest

The colours shown above are read from the file `default.bio`, but you can use a different colour file for biomes by using the `-Z` option (see above). Each row specifies a colour for one biome as a letter specifying the biome followed by three numbers in the range 0 to 255, specifying the RGB values of the colour. The letters used are:

I	ice
T	tundra
G	grasslands
B	Taiga / boreal forest
D	desert
S	savanna
F	temperate forest
R	temperate rainforest
W	xeric shrubland and dry forest
E	tropical dry forest
O	tropical rainforest

The order in which the biome colours are specified does not matter. If a biome is not specified, the default colour is used for this.

Biome colours are only used for land areas, sea areas use the colours specified in the colour map (if specified, otherwise `Olsson.col`).

The biomes are generated from rough estimates of temperature and rainfall based on latitude, altitude, and approximated rain shadow. This is converted into biomes using a Whittaker diagram.

The -n option is ignored when calculating biomes, as I have found it gives bad results. The coastlines are the same, so you can use the -z option even if you normally use the -n option.

5 Various notes

It is assumed that pixels are square. I have included procedures that will print the maps as uncompressed bmp (default), ppm or xpm bitmaps on standard output or specified files. Use external programs to convert to JPEG, PNG or other compressed formats.

I have tried to avoid using machine specific features, so it should be easy to port the program to any machine. Beware, though that due to different floating point precision on different machines, the same seed numbers can yield very different planets. The IEEE 754 standard should in theory guarantee identical results across IEEE 754 compliant platforms, but this depends on whether the same precision and rounding modes are used.

The primitive user interface is primarily a result of portability concerns.

Jim Burrows has contributed some Mac-specific code to the program. Various people have contributed colour maps.

6 Changelog

In reverse chronological order:

- July 2024: Improved error handling of command line options.

- July 2024: Added option for reading biome colours from file.

- July 2024: Added `2col.col` colour map that uses one shade of blue for sea and one shade of green for land (when no bumpmap is used).

- July 2024: Fixed problem that didn't allow spaces in file names.

- September 2023: Fixed level of detail for some projections when zoom less than 1.

- September 2023: Fixed colour interpolation (thanks to Divus Iulios).

- June 2021: Fixed bug in daylight shading.

- June 2021: Added biome generation.

- June 2021: Allow contour lines at land and sea on same map.

- March 2021: Fixed longitude displacement in icosahedral projection.

- February 2020: Changed the way contour lines work, see description below.

- January 2018: Replaced search feature with a "match" option (-M). See details below.

- January 2018: Fixed bug in square projection that would sometimes leave blank top line where it should not be. Also fixed bug in sinusoidal projection that distorted zoomed maps.

- May 2016: When outputting a BMP file, a copy of the command line is appended to the end of the file. This should not affect bitmap viewers (I tried a few different ones), but it allows you to recreate the parameters used to create the map. On Linux, MacOS and other Unix-derived systems, the command "tail -1 <file name>" outputs the last line of a file. When outputting XPM or PPM files, the command line is added as a comment line near the start of the file.

- March 2016: Added a possibility to rotate the planet befoer applying grids and latitude-dependent colouring. This required a small change in the generation method which means that maps generated with the same seed give different results before and after the change. The old version (named "oldplanet.c") will no longer be updated, but can be used if you wish to recreate old maps.

- November 2014: Made heightfield an output option instead of a projection.

- March 2014: Made -c option cumulative, fixed bug in grid generation.

- March 2013: Added icosahedral projection.

- February 2013: A web interface was added.

  A thing to note if you are using both the web tool and the offline program is that seeds in the web tool are integers that get "0." prefixed to them to get the equivalent for seed for the offline version. So, for example, the seed 1111111 in the web tool corresponds to the option "-s 0.1111111" in the offline program.

- November 2011: A new, more generic, method is used to draw gridlines. This uses more memory but can be much faster on maps with high magnification.

- November 2011: Snowcaps (when using -c option) are made smaller, as they were excessively large in previous versions.

- November 2011: The default colour map is changed.

- November 2011: The Mollweide projection can now be shown centered at arbitrary latitudes.