# Mapping Toolbox Release Notes

The "Mapping Toolbox 2.1 Release Notes" on page 1-1 describe the changes introduced in the latest version of the Mapping Toolbox. The following topics are discussed in these Release Notes:

- "New Features" on page 1-2
- "Major Bug Fixes" on page 1-4
- "Upgrading from an Earlier Release" on page 1-5
- "Known Software and Documentation Problems" on page 1-15

The Mapping Toolbox Release Notes listed below provide information about previous versions of the product.

- "Mapping Toolbox 2.0.3 Release Notes" on page 2-1
- "Mapping Toolbox 2.0.2 Release Notes" on page 3-1
- "Mapping Toolbox 2.0.1 Release Notes" on page 4-1
- "Mapping Toolbox 2.0 Release Notes" on page 5-1
- "Mapping Toolbox 1.3.1 Release Notes" on page 6-1
- "Mapping Toolbox 1.3 Release Notes" on page 7-1
- "Mapping Toolbox 1.2 Release Notes" on page 8-1

---

**Note** Mapping Toolbox versions 2.0 and 2.0.1 were released in Web download form after Release 13 with Service Pack 1 was shipped. Mapping Toolbox 2.0.2 was released with Release 14 of MATLAB. Mapping Toolbox 2.0.3 was released with Release 14 with Service Pack 1.

---

### Printing the Release Notes

If you would like to print the Release Notes, you can link to a PDF version.

# Mapping Toolbox 2.1 Release Notes

**1**

# Mapping Toolbox 2.0.3 Release Notes

**2**

**3**

## Mapping Toolbox 2.0.2 Release Notes

**4**

## Mapping Toolbox 2.0.1 Release Notes

**5**

## Mapping Toolbox 2.0 Release Notes

## **6** Mapping Toolbox 1.3.1 Release Notes

## **7** Mapping Toolbox 1.3 Release Notes

# Mapping Toolbox 1.2 Release Notes

**8**

# Mapping Toolbox 2.1 Release Notes

# New Features

This section summarizes the new features and enhancements introduced in the Mapping Toolbox 2.1.

If you are upgrading from a release earlier than Version 2.0, then you should see "New Features" on page 5-2 in the Version 2.0 Release Notes.

### New Function Reads Both 2-Minute and 5-Minute ETOPO Data

The new function `etopo` reads from either the 5-minute (ETOPO5) or the 2-minute (ETOPO2) global terrain data set. This function supersedes function `etopo5` and fixes several significant bugs.

### Function gshhs Now Returns a Version 2 Geostruct

Function `gshhs`, which reads the Global Self-consistent Hierarchical High-resolution Shoreline data set, has been upgraded. It now returns a Version 2 geographic data structure (geostruct2) array instead of a Version 1 geostruct. Polygons returned from `gshhs` now follow the shapefile vertex-ordering convention (supported by functions `polybool`, `shaperead`, and `shapewrite`, for example). Under this convention the coordinates of outer rings (e.g., continent outlines) are given in clockwise order, while counterclockwise ordering is used for inner rings (e.g., lakes and inland seas within a continent). Note that function `gshhs` does not yet support Version 1.3 of the data set, released on Sept. 27, 2004.

### Geodata Can Now Be Exported in Shapefile Format

This new function `shapewrite` writes a geographic data structure to a shapefile. It exports a Version 3 geographic data structure array (geostruct2), creating `.shp`, `.shx`, and `.dbf` files. Like `shaperead`, the function supports the Point, MultiPoint, PolyLine, and Polygon shape types. The contents of string-valued attribute fields and scalar numerical attribute fields are written to the dBase (`.dbf`) file.

## Accessing Geodata Resources on the Internet

Links and URLs to documentation and data files for various Internet sources of digital map data are now collected in the following technical note on the MathWorks Web site:

`http://www.mathworks.com/support/tech-notes/2100/2101.html`

This technical note replaces many individual links formerly scattered across the User's Guide, reference pages, and M-file help. Collecting this information on a Web page rather than on product CDs or printed documentation should substantially mitigate recurrent problems with stale links. You are encouraged to report any stale links that you might find in the technical note to MathWorks Technical Support, so that it can be updated promptly.

# Major Bug Fixes

Mapping Toolbox 2.1 includes several bug fixes made since Version 2.0.3. You can see a list of the particularly important Version 2.1 bug fixes.

If you are viewing these release notes in PDF form, you can refer to the HTML form of the release notes, using either the Help browser or the MathWorks Web site and use the link provided.

If you are upgrading from a release earlier than Version 2.0, then you should see "Major Bug Fixes" on page 5-13 in the Mapping Toolbox 2.0 Release Notes, the Mapping Toolbox 2.0.3 Release Notes, and the Mapping Toolbox 2.0.1 Release Notes.

# Upgrading from an Earlier Release

This section describes issues involved in upgrading from the Mapping Toolbox Version 2.0, 2.0.1, 2.0.2 or 2.0.3 to Version 2.1:

If you are upgrading from a version earlier than 2.0, then you should see "Upgrading from an Earlier Release" on page 5-14 in the Mapping Toolbox 2.0 Release Notes.

## Atlas Data Changes: Overview

Through Version 2.0.3, the Mapping Toolbox included a set of *atlas data* with global geopolitical data embedded as MATLAB arrays in four MAT-files: `worldlo`, `worldhi`, `worldmtx`, and `worldmtxmed`. However, geopolitical data is difficult to keep current, and is subject to inaccuracies and interpretations that can cause contention. Therefore, starting with Version 2.1, the Mapping Toolbox now excludes geopolitical data that would specify national sovereignty over specific regions of the Earth. The only exceptions are the boundaries of the 50 U.S. states and the District of Columbia.

This change means that the `worldlo`, `worldhi`, `worldmtx`, and `worldmtxmed` MAT-files are no longer part of the Mapping Toolbox. However, the nonpolitical data on global coastlines, major lakes and inland seas, major rivers, and major cities and populated places that was in `worldlo.mat` has been retained in the toolbox and transformed into shapefile format. This includes the addition of name attributes for many previously unnamed features. There are four new shapefiles in this category: `landareas.shp`, `worldlakes.shp`, `worldrivers.shp`, and `worldcities.shp`.

For consistency, the atlas data for the United States that was originally stored in the `usalo` and `usahi` MAT-files has also been transformed, although none has been removed. These data sets now reside in the following shapefiles and MAT-files: `usastatelo.shp`, `usastatehi.shp`, `conus.mat`, and `greatlakes.mat`.

The Mapping Toolbox originally included four functions dedicated to extracting data from the atlas data MAT-files: functions `worldlo`, `worldhi`, `usalo`, and `usahi`. With the data removal/transformation described above, these functions are no longer needed and have been removed from the toolbox in Version 2.1. You can easily access the new shapefiles using the `shaperead` function, which includes powerful and flexible options for selecting features and even controlling which attributes are read. In addition, function `country2mtx`,

whose sole purpose was to rasterize the country boundary polygons in `worldlo.mat`, has been removed.

Related changes extend to the `worldmap` function, which formerly combined two purposes:

- Select an appropriate map projection and parameters with which to display a given latitude-longitude area, and
- Automatically display atlas data for that area

In Version 2.1, `worldmap` supports only the first of these actions. A call to `worldmap` constructs a map axes object and can easily be followed with a variety of Mapping Toolbox commands to display the map data of your choice. Because the `usamap` function is so similar to `worldmap`, corresponding changes have been made there as well.

To help those who have relied heavily on `worldmap` and `usamap` to plot base maps with automatically-selected vector map data, examples throughout the User's Guide, reference pages, and M-file Help entries have been updated to illustrate the new behavior of `worldmap` and `usamap`, and to show how to create maps including vector shapefile data layers. These examples cover a wide variety of ways to read and subset data with `shaperead` and display data with `geoshow` and other Mapping Toolbox display functions. A good place to start is with the examples for the `worldmap` and `usamap` functions.

To help you update commands, scripts, and data for constructing and maintaining base maps, a recently-published technical note on The MathWorks Web site provides links to data and documentation for many sources of vector and raster digital map data that you can access over the Internet:

`http://www.mathworks.com/support/tech-notes/2100/2101.html`

## Changes to WORLDMAP and USAMAP

`worldmap` and `usamap` have been simplified to construct appropriate map axes for a given area without displaying any map data.

In all cases, map frames, latitude-longitude grid lines, meridian labels, and parallel labels are turned on. You can use the following command sequence to remove them:

```
framem off; gridm off; mlabel off; plabel off
```

Other changes include the following:

- `usamap` now accepts two-letter U.S. Postal Service abbreviations for state names (e.g., AL, AK, AR ...).
- The following input options are now obsolete (if used, a warning is issued):
  - A first argument equal to `'lo'` or `'hi'`
  - The `regiononly` and `stateonly` syntax: a state or country name with the string `'only'` appended
  - All `type` options: `'none'`, `'line'`, `'lineonly'`, `'patch'`, `'patchonly'`, `'mesh'`, `'meshonly'`, `'dem'`, `'demonly'`, `'dem3d'`, `'dem3donly'`, `'lmesh3d'`, `'lmesh3donly'`, `'ldem3d'`, and `'ldem3donly'` (the new behavior matches the `'none'` option)

## Data Files Added in this Release

The following files were added to the `mapdemos` directory, for use in toolbox demos and examples:

- `landareas` — Polygon shapefile: global coastlines, both exterior and interior, including names for larger land masses
- `worldlakes` — Polygon shapefile: coastlines and names of major lakes and inland seas worldwide
- `worldrivers` — PolyLine shapefile: major world rivers and their names
- `worldcities` — Point shapefile: locations and names of major cities and populated places worldwide
- `usastatelo` — Polygon shapefile: low-resolution outlines and names of the 50 U.S. states plus D.C.
- `usastatehi` — Polygon shapefile: moderate-resolution outlines and names of the 50 U.S. states plus D.C.
- `conus` — MAT-file: Low-resolution latitudes and longitudes, in degrees, for the perimeter of the conterminous United States (CONUS), the Great Lakes, and interstate borders
- `greatlakes` — MAT-file: A Version 1 geographic data structure (geostruct1) with outlines and names for the Great Lakes of North America

## Atlas Data MAT-files Removed in this Release

MAT-files containing Atlas Data have been removed in Version 2.1. Some of the data has been retained in a different form. The disposition of these data sets and variables is described below:

### World MAT-File Data

- `worldlo.mat`, which contained the following variables:
  - `DNline` — Data moved to `worldrivers.shp`
  - `DNpatch` — Data moved to `worldlakes.shp`
  - `POline` — Data removed from toolbox
  - `POtext` — Data removed from toolbox
  - `PPpoint` — Data moved to `worldcities.shp`
  - `PPtext` — Data moved to `worldcities.shp`
  - `gazette` — Data removed from toolbox
- `worldhi.mat` — Data removed from toolbox
- `worldmtx.mat` — Data removed from toolbox
- `worldmtxmed.mat` — Data removed from toolbox

### United States MAT-File Data

- `usalo.mat`, which contained the following variables (all retained):
  - `conus` — Data moved to `conus.mat`
  - `greatlakes` — Data moved to `greatlakes.mat`
  - `gtlakelat` — Data moved to `conus.mat`
  - `gtlakelon` — Data moved to `conus.mat`
  - `state` — Data moved to `usastatelo.shp`
  - `stateborder` — Data moved to `conus.mat`
  - `statelat` — Data moved to `conus.mat`
  - `statelon` — Data moved to `conus.mat`
  - `uslat` — Data moved to `conus.mat`
  - `uslon` — Data moved to `conus.mat`
- `usahi.mat` — Data moved to `usastatehi.shp`

## Functions Deleted in This Release

The following functions, which performed specific operations on Atlas Data sets, have been removed in Version 2.1:

- `country2mtx` — Create a raster map grid of a country from `worldlo` atlas data
- `usahi` — Return high-resolution vector atlas data for the United States
- `usalo` — Return vector atlas data for the United States
- `worldhi` — Return high-resolution vector atlas data for the world
- `worldlo` — Return vector atlas data for the world or oceans

## Functions Made Obsolete in This Release

- `etopo5` — Replaced by `etopo`. Use `etopo` instead.
- `tigerp` — Download U.S. Census cartographic boundary files in shapefile format and use `shaperead` instead.
- `tigermif` — Download U.S. Census cartographic boundary files in shapefile format and use `shaperead` instead.

## worldmap and usamap Display Types Superseded

As of this release, the `worldmap` and `usamap` functions no longer supports the `type` input argument. This argument provided an easy way to control display behavior.

The `type` option in `worldmap` was a single argument that could be one of the following strings: `'none'`, `'line'`, `'lineonly'`, `'patch'`, `'patchonly'`, `'mesh'`, `'meshonly'`, `'dem'`, `'demonly'`, `'dem3d'`, `'dem3donly'`, `'lmesh3d'`, `'lmesh3donly'`, `'ldem3d'`, and `'ldem3donly'`. In `usamap`, `type` was a subset of the above names (the 3-D options were not supported).

In the current release, the various `type` display options can be simulated by following a call to `worldmap` or `usamap` with an appropriate set of Mapping Toolbox commands. The following table specifies how you can achieve the effects of the old `worldmap` `type` argument using such auxiliary methods:

| Mapping 1.x - 2.0.x Usage | Mapping 2.1 Usage |
|---|---|
| ```
load topo
worldmap(topo, ...
  topolegend, 'dem')
``` | ```
load topo
worldmap(topo, topolegend)
meshm(topo, topolegend)
demcmap(topo)
land =
shaperead('landareas.shp',...
  'UseGeoCoords', true);
geoshow([land.Lat], [land.Lon])
``` |
| ```
load topo
worlmdap(topo, ...
  topolegend, 'demonly')
``` | ```
load topo
worldmap(topo, topolegend)
meshm(topo, topolegend)
demcmap(topo)
``` |
| ```
load topo
worldmap(topo, ...
  topolegend, 'dem3d')
``` | ```
load topo
worldmap(topo, topolegend)
meshm(topo,
topolegend,size(topo),topo)
da = daspect;
pba = pbaspect;
da(3) = 7.5*pba(3)/da(3);
daspect(da);
demcmap(topo)
land =
shaperead('landareas.shp',...
  'UseGeoCoords', true);
geoshow([land.Lat], [land.Lon])
``` |
| ```
load topo
worldmap(topo, ...
  topolegend, 'dem3donly')
``` | ```
load topo
worldmap(topo, topolegend)
meshm(topo,
topolegend,size(topo),topo)
demcmap(topo)
``` |

| Mapping 1.x - 2.0.x Usage | Mapping 2.1 Usage |
|---|---|
| ```
load korea
worldmap (map, refvec,...
  'mesh')
``` | ```
load korea
worldmap(map, refvec)
meshm(map, refvec)
land =
shaperead('landareas.shp',...
  'UseGeoCoords', true);
geoshow([land.Lat], [land.Lon])
``` (Text North Korea and South Korea will be missing) |
| ```
load korea
worldmap(map, refvec,...
'meshonly')
``` | ```
load korea
worldmap(map, refvec)
meshm(map,refvec)
``` |
| ```
load topo
worldmap(topo,...
topolegend, 'mesh3d')
``` | ```
load topo
worldmap(topo, topolegend)
meshm(topo,
topolegend,size(topo),topo)
da = daspect;
pba = pbaspect;
da(3) = 7.5*pba(3)/da(3);
daspect(da);
``` |
| ```
load topo
worldmap...
(topo,topolegend,...
'mesh3donly')
``` | ```
load topo
worldmap(topo, topolegend)
meshm(topo,
topolegend,size(topo),top
``` |

| Mapping 1.x - 2.0.x Usage | Mapping 2.1 Usage |
|---|---|
| ```
load topo
worldmap(topo,...
  topolegend, 'ldem3d')
``` | ```
load topo
worldmap(topo, topolegend)
meshm(topo,
topolegend,size(topo),topo)
da = daspect;
pba = pbaspect;
da(3) = 7.5*pba(3)/da(3);
daspect(da);
demcmap(topo)
camlight(90,5);
camlight(0,5);
lighting phong
material([0.25 0.8 0])
``` |
| ```
load topo
worldmap(topo, ...
  topolegend,'ldem3donly')
``` | ```
load topo
worldmap(topo, topolegend)
meshm(topo,
topolegend,size(topo),topo)
da = daspect;
pba = pbaspect;
da(3) = 7.5*pba(3)/da(3);
daspect(da);
demcmap(topo)
``` |
| ```
load topo
worldmap(topo,...
  topolegend, 'lmesh3d'
``` | ```
load topo
worldmap(topo, topolegend)
meshm(topo,
topolegend,size(topo),topo)
da = daspect;
pba = pbaspect;
da(3) = 2*pba(3)/da(3);
daspect(da);
camlight(90,5);
camlight(0,5);
lighting phong
material([0.25 0.8 0])
``` |

| Mapping 1.x - 2.0.x Usage | Mapping 2.1 Usage |
|---|---|
| ```
load topo
worldmap(topo,...
  topolegend,...
  'lmesh3donly')
``` | ```
load topo
worldmap(topo, topolegend)
meshm(topo,
topolegend,size(topo),topo)
da = daspect;
pba = pbaspect;
da(3) = 2*pba(3)/da(3);
daspect(da);
``` |
| ```
load korea
worldmap(map, refvec,...
  'line')
``` | ```
load korea
worldmap(map, refvec)
land =...
  shaperead('landareas.shp',...
  'UseGeoCoords', true);
geoshow([land.Lat], [land.Lon])
```
(Text North Korea and South Korea will be missing. Land area boundaries resolution is lower.) |
| ```
load korea
worldmap(map, refvec,...
  'lineonly')
``` | ```
load korea
worldmap(map, refvec)
land = ...
  shaperead('landareas.shp',...
  'UseGeoCoords', true);
geoshow([land.Lat], [land.Lon])
``` |
| ```
load korea
worldmap(map, refvec,...
  'none')
``` | ```
load korea
worldmap(map, refvec)
``` |

| **Mapping 1.x - 2.0.x Usage** | **Mapping 2.1 Usage** |
|---|---|
| ```load korea``` <br> ```worldmap(map, refvec,...``` <br> ``` 'patch')``` | ```load korea``` <br> ```worldmap(map, refvec)``` <br> ```land = ...``` <br> ```  shaperead('landareas.shp',...``` <br> ```  'UseGeoCoords', true);``` <br> ```faceColors = ...``` <br> ```  makesymbolspec('Polygon',...``` <br> ```  {'INDEX', [1 numel(land)],...``` <br> ```  'FaceColor', ...``` <br> ```  polcmap(numel(land))});``` <br> ```geoshow(land,'SymbolSpec',...``` <br> ```  faceColors)``` <br> (Text North Korea and South Korea will be missing. Country coloring will be missing.) |
| ```load korea``` <br> ```worldmap(map, refvec,...``` <br> ``` 'patchonly')``` | ```load korea``` <br> ```worldmap(map, refvec)``` <br> ```land = ...``` <br> ```  shaperead('landareas.shp',...``` <br> ```  'UseGeoCoords', true);``` <br> ```faceColors =...``` <br> ```  makesymbolspec('Polygon',...``` <br> ```  {'INDEX', [1 numel(land)],...``` <br> ```  'FaceColor', ...``` <br> ```  polcmap(numel(land))});``` <br> ```geoshow(land,'SymbolSpec',...``` <br> ```  faceColors)``` <br> (Country coloring will be missing.) |

# Known Software and Documentation Problems

The MathWorks Web site includes a list of known software and documentation problems in Version 2.1.

If you are viewing these release notes in PDF form on the MathWorks Web site, you can refer to the HTML form of the release notes on the MathWorks Web site and use the link provided.

**2**

# Mapping Toolbox 2.0.3 Release Notes

# New Features

This section summarizes the new features and enhancements introduced in the Mapping Toolbox 2.0.3.

If you are upgrading from a release earlier than Version 2.0, then you should see "New Features" on page 5-2.

## Polygon Vertex Ordering Functions

Four new functions have been added to Mapping 2.0.3. These are called by or can be used in conjunction with the `polybool` function (see "Polybool Revised, Has New Helper Functions" on page 2-5.) Three of the functions check or change the ordering of vertices that define a polygon, and the fourth one splits polygons with holes in a consistent fashion.

These functions are

- `ispolycw` — `True` if vertices of polygonal contour are clockwise ordered
- `poly2cw` — Convert polygonal contour to clockwise ordering
- `poly2ccw` — Convert polygonal contour to counterclockwise ordering
- `poly2fv` — Convert polygonal region to face-vertex form for use with `patch` in order to properly render polygons containing holes.

# Major Bug Fixes

The Mapping Toolbox 2.1 includes important bug fixes made since Version 2.0.3. You can see a list of major Version 2.1 bug fixes on the MathWorks Web site.

If you are viewing these release notes in PDF form on the MathWorks Web site, please refer to the HTML form of the release notes on the MathWorks Web site and use the link provided.

Mapping Toolbox 2.0.3 included several bug fixes made since Version 2.0.2. You can see a list of the particularly important Version 2.0.3 bug fixes.

If you are viewing these Release Notes in PDF form, please refer to the HTML form of the Release Notes, using either the Help browser or the MathWorks Web site and use the link provided.

If you are upgrading from a release earlier than Version 2.0.1 or 2.0.2, then you should see "Major Bug Fixes" on page 5-13 in the Mapping Toolbox 2.0 Release Notes.

# Upgrading from an Earlier Release

This section describes these issues involved in upgrading from the Mapping Toolbox 2.0, 2.0.1, or 2.0.2 to Version 2.0.3:

If you are upgrading from a version earlier than 2.0, then you should see "Upgrading from an Earlier Release" on page 5-14. of the Mapping Toolbox 2.0 release notes.

## Axesm Does Not Reorder Longitude Limits

Function axesm now requires that 'MapLonLimit' values have the form [west east], but east can be less than west numerically (e.g., [120 -60]). The form [east west] is no longer supported.

## Contourm Returns Contourgroup Handle

Function contourm now returns a handle to a contourgroup object instead of an array of handles to line objects.

## Elevation Has New Azimuth Output

Function elevation can now return a third ouput, the line-of-sight azimuth corresponding to the elevation angle (output 1) and slant range (output 2). This eliminates the need to use a geodesic or rhumb line azimuth, computed with functions azimuth or distance, as an approximate substitute.

## Gtopo30 Has New Options

The functionality of gtopo30 has been extended so that:

- Data can be read from a more general directory structure.
- Empty may be supplied in place of any input argument.
- Latitude and longitude limits are no longer restricted to lie within a tile. Instead, areas not covered by available data are automatically filled with NaNs.

## Inputm Has New Button Output

Function inputm now an optional third output argument, 'button', which functions like the third output of ginput.

## Polybool Revised, Has New Helper Functions

Mapping 2.0.3 fixes major problems with the polybool function, which computes set operations on polygonal regions. The fixes include incompatible data structure and syntax changes.

- The 'cutvector' option has been removed from the syntax. This option facilitated plotting of polygonal regions containing holes, but the cutvector computation was not robust, nor was it an effective way to represent polygonal regions containing multiply nested holes and/or discontiguous regions. To replace the cutvector functionality, a new function, poly2fv, has been added. poly2fv converts a polygonal region, possibly including holes and discontiguous regions, into a faces and a vertices matrix that can be used with the patch function to display the region. The M-file help for for polybool and poly2fv contain examples illustrating how to use poly2fv.

- The 'cell' and 'vector' options have been removed from the syntax. polybool now returns the output polygonal region using the same format as the input polygonal region. Use polysplit and polyjoin to convert from one format to the other.

- polybool now uses the "clockwise" rule to distinguish between external polygonal contours and internal polygonal contours. This is the same rule used by ESRI shapefiles. polybool assumes that a polygonal contour whose vertices are arranged in clockwise order is an external contour, and that a polygonal contour whose vertices are arranged in counterclockwise order is an internal contour. Use ispolycw to determine whether a polygonal contour is clockwise ordered, and use poly2cw and poly2ccw to convert polygonal contours to be clockwise or counterclockwise ordered, respectively. If a polygonal region input has no external contours according to this rule, polybool issues a warning message.

An input polygonal region can either take the form of a pair of NaN-separated vectors, or it can take the form of a pair of cell arrays. In the cell array format, each cell must contain the vertices of a single polygonal contour. Representing a polygonal region as a cell array whose cells can contain NaN-separated vectors is no longer supported, and an error message will be issued.

Summary of new polybool-related helper functions

- ispolycw — True if vertices of polygonal contour are clockwise ordered
- poly2cw — Convert polygonal contour to clockwise ordering

- `poly2ccw` — Convert polygonal contour to counterclockwise ordering
- `poly2fv` — Convert polygonal region to face-vertex form for use with `patch`

Summary of known problems addressed

- `polybool` no longer errors out or produces incorrect results for proper inputs.
- `polybool` now produces consistent output when computing intersections, with results no longer depending on the order of its inputs.
- `polybool` now operates correctly on polygons with collinear edges.

## Polymerge Ouput Vertex Order Changes

Improvements in the algorithm that `polymerge` uses to chain together adjacent polygon or line segments, required to address a bug and improve efficiency, in some cases also change the ordering of the output vertices.

# Known Software and Documentation Problems

You can see a list of known software and documentation problems in Version 2.0.3. If you are viewing these Release Notes in PDF form, please refer to the HTML form of the Release Notes, using either the Help browser or the MathWorks Web site and use the link provided.

# Mapping Toolbox 2.0.2 Release Notes

# New Features

Version 2.0.2 of the Mapping Toolbox adapts Version 2.0.1 to the MATLAB 7 (Release 14) environment. Versions 2.0.1 and 2.0 were Release 13SP1+ web releases. The resulting changes are transparent to users, except as noted below.

If you are upgrading from a version earlier than Version 2.0.1, then you should also see "New Features" on page 5-2 in the Mapping Toolbox 2.0 Release Notes.

The following enhancements are included in Mapping Toolbox 2.0.2.

## Mapping Toolbox Compilability Enhanced

Almost all Mapping Toolbox command line functions can be compiled using the new MATLAB Compiler. However, large Atlas Data MAT-files need to be added explicitly to use them in applications. For details, see "Using the Compiler with the Mapping Toolbox" in the MATLAB Compiler Release Notes.

## SDTS Functions Now Available on HPUX

The input functions `sdtsinfo` and `sdtsdemread` are now available on the HPUX platform.

# Major Bug Fixes

Mapping Toolbox Version 2.0.2 contains no significant bug fixes. However, Version 2.0.1 did address a number of bugs. See "Major Bug Fixes" on page 4-3 for details.

Mapping Toolbox 2.0 included a number of significant bug fixes made since Version 1.3.1. For a description of those Mapping Toolbox bug fixes, see "Major Bug Fixes" on page 5-13 in the Version2.0 release notes.

If you are upgrading from a version earlier than 2.0, then see should also see "Bug Fixes and Enhancements" on page 6-2 in the Mapping Toolbox 1.3.1 Release Notes.

**4**

# Mapping Toolbox 2.0.1 Release Notes

# New Feature

The following enhancement has been added to Mapping Toolbox 2.0.1.

If you are upgrading from a version earlier than Version 2.0, then you should also see "Mapping Toolbox 2.0 Release Notes" on page 2-1 in the Mapping Toolbox 2.0 Release Notes.

## mapexreg Demo (Georeferencing an Image to an Orthotile Base Layer) Generalized and Clarified

Calls to `imtransform` and `makerefmat` are no longer hard-coded for an output pixel size of one meter. The choice of output pixel size now is explained in detail.

# Major Bug Fixes

Mapping Toolbox Version 2.0.1 fixes several significant bugs that existed in Version 1.x which were not addressed in Version 2.0.

Mapping Toolbox 2.0 included a number of significant bug fixes made since Version 1.3.1. For a description of those Mapping Toolbox bug fixes, see "Major Bug Fixes" on page 2-13 in the Version 2.0 release notes.

If you are upgrading from a version earlier than 2.0, then see should also see "Bug Fixes and Enhancements" on page 1-2 in the Mapping Toolbox 1.3.1 Release Notes.

Version 2.0.1 contains the following bug fixes:

### almanac

Function almanac now gives a precise inverse flattening (1/f) value for Bessel 1841 ellipsoid. Previously, almanac used a truncated value (299.1528) instead of the full precision value (299.1528128).

### axesm

Function axesm now accepts the 'meridianlabel' parameter. Previously, axesm would throw an eror if the 'meridianlabel' parameter was specified.

### distance

Function distance now computes correct results for rhumbline distances along a parallel (including the equator) on a (non-spherical) ellipsoid.

### distance

Function distance now uses the robust "haversine formula" to compute great circle distances on a sphere. For certain inputs roundoff sensitivities in the previous implementation resulted in a small, but non-zero, distance between identical points.

### distance

Function distance now correctly obtains correct results for rhumb lines that follow the equator, or any other parallel, and cross the 180-degree meridian; it gives the distance along the short arc of the parallel. Previously the length of the long arc could be returned instead.

### distance, azimuth

Functions `distance` and `azimuth` now produce accurate results even for long geodesics on an ellipsoid, up to near-antipodal distances.

### projfwd/projinv

Functions `projfwd` and `projinv` now use `NaN` separators on all platforms. Previously, under certain circumstances, `projfwd` and `projinv` returned coordinate arrays with `Inf` separators rather than `NaN`s.

### reckon

Function `reckon` now supports geodesics on an ellipsoid, in addition to rhumblines on an ellipsoid and both geodesics and rhumblines on a sphere

**5**

# Mapping Toolbox 2.0 Release Notes

# New Features

This section summarizes the new features and enhancements added in the Mapping Toolbox 2.0 since Version 1.3.1.

If you are upgrading from a Version earlier than 1.3.1, then you should also see "Bug Fixes and Enhancements" on page 6-2 of the Mapping Toolbox 1.3.1 Release Notes.

## Release Summary

Mapping Toolbox 2 provides a comprehensive set of functions and graphical user interfaces for building map displays and performing geospatial data analysis in MATLAB. In this major release of the toolbox, new functions and capabilities include:

- Support for standard GIS and geospatial file formats:
  - ESRI shapefile; GeoTIFF; Arc ASCII Grid; SDTS Raster Profile (read only)
  - "Worldfiles" for spatially-registered images (read and write)
- New display functions for vector features, georeferenced imagery, and data grids in *x-y* map coordinates (as well as unprojected latitude-longitude)
- New functions for vector symbolization based on feature attributes
- Transverse Mercator projection and PROJ.4 projection library support
- A new interactive Map Viewer with the following features:
  - Support for multiple raster and vector layers
  - Data import from file or workspace
  - Zooming/panning/map navigation tools
  - Cursor coordinate and map-scale readout
  - Data tip and info tools
  - Graphical overlays
  - Map exporting/printing
- A major update of the Mapping Toolbox User's Guide, focused on clarifying concepts and terminology, explaining features and functions, adding new and enhancing existing examples, and organizing and cross-referencing material to make the documentation easier to access

These and other new features are described on the following pages.

## New Demos and Sample Data

The Mapping Toolbox now has six new demos, available from the **Demos** tab of the Help Browser. You can also access this demos page by typing

```
mapdemos
```

The demos consist of the following:

- `mapexenhance` — Enhancing Multispectral GeoTIFF Images
- `mapexfindcity` — Interactive Global City Finder
- `mapexgeo` — Creating Maps using `geoshow` (for latitude, longitude data)
- `mapexmap` — Creating Maps using `mapshow` (for x, y data)
- `mapexrefmat` — Creating and Using Referencing Matrices
- `mapexreg` — Georeferencing an Image to an Orthotile Base Layer
- `viewmaps` — GUI Demonstrating Map Projections (not new in 2.0)

Note that the above commands run the demo scripts to produce figures, whereas `mapdemos` describes and illustrates the demos in the Help Browser.

In addition, a number of new sample data sets containing vector features, digital elevation models, and georeferenced images have been added for use in demos and examples. The new data includes satellite images of the Boston area, topographic grids of the White Mountains in New Hampshire, and vector data for roads and hydrographic features in the Boston area. The data sets are provided in standard geospatial and GIS formats: GeoTIFF, TIFF images with worldfiles, SDTS Raster Profile, Arc ASCII Grid, and ESRI shapefiles. They are accompanied by ASCII text files (with suffix `.txt`), containing descriptive metadata.

The new data files are in addition to existing sample data sets stored in MAT-files, such as `geoid`, `russia`, and `korea`. In addition, metadata indicating the source and describing the contents of most sample data and atlas data has been added to their respective MAT-files.

To read the metadata for a Mapping Toolbox MAT-file data set, `load` it and inspect the `source` and `description` workspace variables.

To read brief descriptions of the demos and sample data sets, type

```
help mapdemos
```

## New Map Viewer

The toolbox includes a new interactive tool for displaying and examining map data, called the Map Viewer. The Map Viewer helps you work with data that is already in a projected map coordinate system, which is the case for many high-resolution satellite and aerial imagery products, as well as many vector map data sets that cover small areas of the earth in substantial detail. The Map Viewer gives you a view (or views) of an *x-y* map coordinate plane where, for example, *x* and *y* may correspond to the easting and northing coordinates of a given UTM or State Plane zone. If some of your data is referenced to geographic (latitude-longitude) coordinates, you first need to use an appropriate projection to transform it to map coordinates.
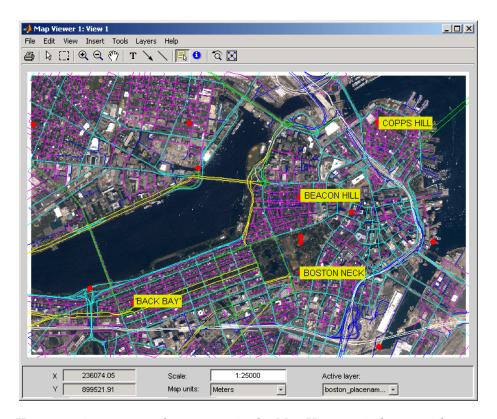
Key features of the Map Viewer include:

- A two-dimensional view of the *x-y* map plane with interactive tools for navigating in that plane: magnification, de-magnification, panning, setting the map scale, and zooming the view to the extent of a given data set, or all data sets

- Vector and image data import from files or from the MATLAB workspace

- Management of each data set in a separate layer. You can control layer ordering and visibility to bring what you need into view.

- Multiple, simultaneous views (differing in scale or view extent) of the same data layers.

- A data tip tool to identify vector shapes or query a feature attribute that you specify

- An info tool to display, in a separate window, all the attributes of a selected feature

- Tools to annotate the map with lines, arrows, and text

- Ability to export a raster snapshot of the map view

To start the Map Viewer, use the `mapview` function Typing

```
mapview
```

will open a Map Viewer window. You can then import standard format files or load workspace data into the Map Viewer. The following figure shows the Map Viewer's main window after loading image, line, and point data sets, also illustrating user-specified symbology for streets and landmarks, and Data Tip labels for selected point features:

You can print maps as they appear in the Map Viewer window, to either a printer or a file. Either click the **Print** button on the left end of the toolbar or select **Print** in the **File** menu. A standard figure print dialog appears. You should consider selecting a driver appropriate to your printer; this may be necessary if you attempt to print in color.

## Spatially Referenced Image Formats

The toolbox adds support for several industry-standard formats for spatially referenced images and data grids: Arc ASCII Grid files, GeoTIFF files, and worldfiles. You do this by using the following new functions: `arcgridread`, `geotiffinfo`, `geotiffread`, `getworldfilename`, `worldfileread`, and `worldfilewrite`.

The grid and image file-reading functions construct referencing matrices, enabling a general and consistent approach to managing the relationship between map coordinates and pixel columns and rows (see below).

## Working with Images Referenced to Map Coordinates

The toolbox includes a family of functions supporting images and gridded data sets that are referenced to two-dimensional projected map coordinates. These functions represent the relationship between a point at position (row,col) in an image and point in map coordinates (x,y) with a *referencing matrix*. A referencing matrix is a 3-by-2 matrix, R, that represents a general affine transformation such that

```
[x y] = [row col 1] * R
```

Most often the image (or data grid) consists of square pixels and has edges that align with the map coordinate axes, but you are not limited to this situation.

The new function makerefmat provides several options for constructing referencing matrices from scratch. See its M-file help for further information. However, you can most often obtain a referencing matrix from a world file, via worldfileread, or from a GeoTIFF file via geotiffread. If you have created a new referencing matrix for an image in a generic image format such as TIFF or JPEG, you can save this information in a world file using worldfilewrite.

New functions pix2map and map2pix implement the basic transformation defined above, along with its inverse. With functions mapbbox and mapoutline you can calculate the image bounds or perimeter in map coordinates. You can use the function pixcenters to calculate the coordinates of the center of each image pixel. This is helpful if you have a data grid that you want to display using surface.

The new function mapshow accepts a referencing matrix in order to correctly position an image in a standard Handle Graphics axes (i.e., not a map axes).

Because the affine transformation is extremely general, it can equally well represent the registration of an image to geographic coordinates, with longitude taking the place of map x and latitude replacing map y:

```
[lon lat] = [row col 1] * R
```

A referencing matrix is a more general version of the 1-by-3 map legend vector, perhaps better characterized as a *referencing vector*, that many toolbox functions already use. To convert a referencing vector to a referencing matrix, you can use function `refvec2mat`. The inverse transformation exists only for referencing matrices having a special form, but if it does you can compute it with `refmat2vec`. Functions `latlon2pix` and `pix2latlon` support transformations for referencing matrices in geographic coordinates, allowing for longitude-wrapping differences.

Use the new function `geoshow` along with map axes to display data grids referenced to latitude and longitude via either a referencing vector or a referencing matrix. Or, you can use the new function `grid2image` to quickly display an unprojected grid with latitude as the ordinate (*x*-axis) and longitude as the abscissa (*y*-coordinate).

## SDTS Terrain Data Access

The Mapping Toolbox can now import terrain grids stored in United States Spatial Data Transfer Standard (SDTS) raster profile format, giving you access to a wide selection of public domain terrain data sets, principally those prepared by the U.S. Geological Survey (USGS). USGS DEM data is no longer distributed from government Internet sites. Instead, these and other USGS files are now available from a commercial server at no charge. For further information, see `http://edc.usgs.gov/products/elevation/dem.html` and `http://edc.usgs.gov/geodata/`.

Two new Mapping Toolbox functions support SDTS data handling:

- `sdtsinfo` returns a structure containing metadata for an SDTS data set.
- `sdtsdemread` returns elevation data and a referencing matrix from an SDTS DEM data set.

For details, type `help sdtsinfo` or `help sdtsdemread` at the MATLAB prompt.

## Shapefiles and Vector Features

The toolbox now reads the shapefile format defined by Environmental Systems Research Institute (ESRI) and widely used as a data exchange format for nontopological GIS data. You can query a shapefile for its shapetype, attribute names and types, number of records, etc., using `shapeinfo`, and read 2-D shapes using `shaperead`. The `shaperead` function constructs a *geostruct2*

geographic data structure array (see"Geographic Data Structure Enhancements" on page 5-8 ) ,tailored to the contents read in from shapefiles.

The shape data is represented in MATLAB as a 1-D structure array, with one element per feature. Attribute values are stored in structure fields, or in a separate, parallel structure array. Because attributes are defined according to the needs of the shapefile author and might be unknown until the time a given shapefile is read, the new geographic data structures (described below) are necessarily more general than the geographic data structures already in use in the Mapping Toolbox.

## Geographic Data Structure Enhancements

Certain functions introduced in Version 2 of the Mapping Toolbox use an enhanced geographic data structure layout (called a *geostruct2*) to store and manipulate vector geodata. This layout has the flexibility to store any kind and number of attributes, and handles either geographic (latitude and longitude) or plane (*x* and *y*) coordinates. In contrast, a Version 1 geographic data structure (*geostruct1*) — which is still supported — is limited to a fixed set of fields and can contain geographic coordinates only.

One way to create a Version 2 geographic data structure is to input vector geodata to the workspace from a shapefile. The function shaperead returns a geostruct2 that encapsulates some or all of the data stored in a shapefile and its supporting index and dBASE files. To determine what kinds of data a shapefiles contains, you can use the shapeinfo function to query it.

The new functions mapshow, geoshow, and mapview each display vector data in geostruct2 form. Use geoshow when Lat and Lon coordinate fields exist in the geostruct, and use mapshow or mapview when X and Y fields are present.

You can transform a geostruct1 into a geostruct2 (but not the reverse). Use the function updategeostruct for this purpose. See the section "Mapping Toolbox Geographic Data Structures" in the Mapping Toolbox documentation for descriptions of geostruct formats and uses.

Another new function, extractfield, lets you conveniently combine all the values of a given geostruct field into a single array. For example, given a shapefile with a 'Name' attribute, extractfield can output a cell array of feature names from the geostruct returned by shaperead.

## Attribute-Driven Vector Data Symbolization

When you display shape features as points, lines, or polygons using `mapview`, `mapshow`, and `geoshow`, you can make feature attributes control how they are symbolized. This lets the data control graphic attributes such as color, line width, marker symbol, and visibility on a per-feature basis. In prior versions of the toolbox, attribute-specific symbology was difficult to accomplish, but now a general mechanism exists for this purpose. The new function `makesymbolspec` generates a structure called a *symbolspec* that maps specific attribute values (or ranges) to symbology parameters, and lets you specify default parameters for unspecified values. For example, the following symbolspec colors all roads of `class` 1 red, all roads of `class` 2 green, all roads of `class` 3 blue, and defaults other classes to black (where `class` is an attribute of the road layer):

```
roadColors = makesymbolspec('Line',{'CLASS',1,'Color','r'},...
                                    {'CLASS',2,'Color','g'},...
                                    {'CLASS',3,'Color','b'},...
                                    {'Default','Color','k'});
```

To see an example of what you can do with symbolspecs, look at the illustration for "New Map Viewer" on page 5-4. For details, type `help shaperead`, `help mapshow`, `help geoshow`, `help mapview`, or `help makesymbolspec` at the MATLAB prompt.

## Access to PROJ.4 Map Projections

The Mapping Toolbox now incorporates the PROJ.4 cartographic projections library, originally written by Gerald Evenden of the USGS. Currently the toolbox uses this library primarily to support the GeoTIFF format. You can access selected PROJ.4 projections and their inverses using new functions `projfwd` and `projinv`. Use the `projlist` function to learn about which PROJ.4 projections are available. In addition to accepting GeoTIFF `info` structures, `projfwd` and `projinv` also work with a standard map projection structure (mstruct) used to define projections within map axes.

## Minor Enhancements

In addition to the major new features described above, this release includes the following enhancements and new functions, among others.

### almanac

Now accepts `'ellipsoid'` as a synonym for `'geoid'` as a parameter or refbody value.

### convertlat

A new function that unifies the latitude conversions previously dispersed among functions `geod2aut`, `aut2geod`, etc. For a list of the files it replaces, see "Upgrading from an Earlier Release" on page 5-14.

### geoloc2grid

New function for converting a geolocated data grid (general matrix map) to a regular gridded data set (matrix map). Easier to use for this purpose than `imbedm`, and produces a smoother output with 2-D resampling.

### grid2image

New function to display a regular data grid, with a referencing matrix or referencing vector, as an image. The grid can contain `double`, `uint8`, or `uint16` values. `grid2image` replaces the obsoleted function `imagem`.

### tranmerc

A new map projection, Transverse Mercator, has been added. This is a general Transverse Mercator, as opposed to the more specific Universal Transverse Mercator supported by the `utm` function.

### unitsratio

Provides conversion factors among units of length, and between radians and degrees. Supports a wider range of length units than `distdim`.

For information on using any of these functions, type `help` *function* at the MATLAB prompt.

## Summary of Functions Added in Version 2.0

Version 2.0 features many new functions, all of which are itemized below. Several existing functions have been deprecated or deleted. These are described in "Upgrading from an Earlier Release" on page 5-14.

## Geospatial Data Import and Access

- `arcgridread` — Read a gridded data set in Arc ASCII Grid Format.
- `geotiffinfo` — Information about a GeoTIFF file
- `geotiffread` — Read a georeferenced image from GeoTIFF file.
- `getworldfilename` — Derive a worldfile name from an image file name.
- `sdtsdemread` — Read data from an SDTS raster/DEM data set.
- `sdtsinfo` — Information about an SDTS data set
- `shapeinfo` — Information about a shapefile
- `shaperead` — Read vector feature coordinates and attributes from a shapefile.
- `worldfileread` — Read a worldfile and return a referencing matrix.
- `worldfilewrite` — Construct a worldfile from a referencing matrix.

## Vector Map Data and Geographic Data Structures

- `extractfield` — Extract the field values from a structure.
- `updategeostruct` — Update a geographic data structure.

## Spatial Referencing of Georeferenced Images and Data Grids

- `latlon2pix` — Convert latitude-longitude coordinates to pixel coordinates.
- `makerefmat` — Construct an affine spatial-referencing matrix.
- `map2pix` — Convert map coordinates to pixel coordinates.
- `mapbbox` — Compute bounding box of a georeferenced image or data grid.
- `mapoutline` — Compute outline of a georeferenced image or data grid.
- `pix2latlon` — Convert pixel coordinates to latitude-longitude coordinates.
- `pix2map` — Convert pixel coordinates to map coordinates.
- `pixcenters` — Compute pixel centers for georeferenced image or data grid.
- `refmat2vec` — Convert a referencing matrix to a referencing vector.
- `refvec2mat` — Convert a referencing vector to a referencing matrix.
- `geoloc2grid` — Convert a geolocated data array to a regular data grid.

### Map Projection Properties and Transformations

- `geotiff2mstruct` — Convert GeoTIFF info to a map projection structure.
- `projlist` — List map projections supported by `projfwd` and `projinv`.
- `projfwd` — Forward map projection using the PROJ.4 library
- `projinv` — Inverse map projection using the PROJ.4 library
- `tranmerc` — Transverse Mercator Projection

### Map Display and Interaction

- `geoshow` — Display map latitude and longitude data.
- `grid2image` — Display a regular data grid as an image.
- `mapshow` — Display map data.
- `mapview` — Interactive map viewer
- `makesymbolspec` — Construct a vector symbolization specification.

### Geographic Calculations

- `convertlat` — Convert between geodetic and auxiliary latitudes.

### Utilities

- `ind2rgb8` — Convert an indexed image to a `uint8` RGB image.
- `unitsratio` — Unit conversion factors

# Major Bug Fixes

This section describes major Mapping Toolbox bug fixes for Version 2.0. Mapping Toolbox 1.3.1 included several dozen bug fixes made since Version 1.3. Most were minor or cosmetic in nature. However, there were several significant Version 1.3 bug fixes.

Mapping Toolbox 2.0 includes the following bug fixes:

### sizem

`sizem` now respects the order of its longitude limits. Now, for example, `[r,c] = sizem([-5 5],[170 -170],1)` sizes a 10-by-20 grid that crosses the 180-degree meridian, rather than a 10-by-340 grid that extending from -170 all the way to 170. `maptriml`, `maptrimp`, and `maptrims` now respect the order of their longitude limits as well.

### Fixed Error in Inverse UTM Projection

This release corrects an error in the inverse UTM projection. This error caused mislocations that ranged from negligible near the central meridian to several meters at the east-west zone boundaries.

This release also removes the rounding that decreased precision of the latitude-longitude outputs.

### Improved Numerical Behavior of Geodetic-to-Conformal Latitude Conversion

A new formulation, using the new `convertlat` function, fixes subtle inaccuracies near the poles that could cause unexpected, hard-to-explain behavior in the stereographic projection.

### Line-of-Sight Computation Corrected for Observer at Zero Elevation

A problem that caused `los2` and `viewshed` to report some visible points as invisible has been fixed.

# Upgrading from an Earlier Release

If you are upgrading from Version 1.2.1, see "Upgrading from an Earlier Release" on page 7-5 in the Mapping Toolbox 1.3 Release Notes.

## Functions Obsolete in Version 2.0

The following functions are still available but should no longer be used.

- `imagem` — which displayed a regular matrix map as an image — has been replaced by `grid2image`.
- The following 12 latitude conversion functions have been replaced by a single utility function, `convertlat`:
  - `aut2geod`
  - `cen2geod`
  - `cnf2geod`
  - `iso2geod`
  - `par2geod`
  - `rec2geod`
  - `geod2aut`
  - `geod2cen`
  - `geod2cnf`
  - `geod2iso`
  - `geod2pa`
  - `geod2rec`

  This improves the stability of numerical results in certain projections.

## Functions Deleted in Version 2.0

The following functions are no longer available in the Mapping Toolbox.

- `coast` — This function simply loaded the `coast` MAT-file containing world coastlines. Instead of calling it, type

  ```
  load coast
  ```

- `loadmoonalb` — This function simply loaded the `moonalb` MAT-file of the Moon's albedo. Instead of calling it, type

  ```
  load moonalb
  ```

- `maskm` — This function, which reassigned a scalar value to an array based on a Boolean mask, has been removed. Ordinary MATLAB command syntax does the same thing.

- `movescale` — This undocumented function now is a subfunction of `scaleruler`.

- `surfdist` — This function was in fact obsolete prior to Version 1.3, and has now been removed. Use `trackg` instead.

## Changes in Nomenclature

To achieve greater consistency with the literature on geospatial data handling, documentation for Version 2.0 of the Mapping Toolbox has changed the usage of certain terms and names of variables provided as sample data. The primary changes are described below.

**Geoid** — Where it is used to describe the geometric shape of the earth (an equipotential surface), this term has been retained. Where it was used as a synonym for ellipsoid, occurances of *geoid* have been changed to *ellipsoid*. This includes changing references to *geoid vector* to *ellipsoid vector*. You also now specify ellipsoid models for `almanac` data using the keyword `'ellipsoid'` (however, `'geoid'` still works). Note that a similar update has *not* been made for `axesm`, `getm`, or `setm`, which also `'geoid'` as a keyword to identify the userdata structure (mstruct) field `geoid`.

**Map** — Where this term referred to data sets (either raster or vector), occurances of *map* have been changed to *data grid*, *data set*, or some other appropriate term. Where it refers to a cartographic presentation of geodata, *map* has been retained.

**Map legend** — This term, which used to refer to a three-element vector that georeferenced a data grid, has been replaced with *referencing vector*. In some contexts (when more degrees of freedom are involved), the term *referencing matrix* replaces it.

Most sample data sets that contain `map` and `maplegend` as variables (data grids and referencing vectors, respectively) have been updated. For example, the `geoid` MAT-file now has variables `geoid` (the data grid), and `geoidrefvec` and `geoidlegend` (referencing vectors; the second is a copy of the first, provided to to maintain compatibility). This file, along with most other sample data except for `topo` and `coast`, also now contains metadata in the form of `source` and `description` strings.

# Mapping Toolbox 1.3.1 Release Notes

# Bug Fixes and Enhancements

The Mapping Toolbox 1.3.1 includes several bug fixes and enhancements made since Version 1.3.

If you are upgrading from Version 1.2, you should also see the list of Version 1.3 bug fixes.

### Bug Fixes for Functions

| Function | Description of Fix/Enhancement |
| --- | --- |
| almanac | Minor typographical errors in defining the parameters for the Clarke 1866 and Clarke 1880 ellipsoids have been fixed. |
| areaint | Every simple polygon divides the earth's surface into two regions. Now areaint consistently returns the area of the smaller region. |
| avhrrgoode | No longer fails due to an illegal assignment of NaN to class logical (mitigated side effect of new logical class in Release 13) |
| axesmui | Now updates frame limits when the projection is changed |
| demdataui | No longer disables zoom mode |

| Function | Description of Fix/Enhancement |
| --- | --- |
| dted | The calculation of directory names is now correct, even for longitudes between -10 and +10. |
| | The calculation of file names for tiles close to the equator no longer includes an extra period. |
| | dted now handles the case of missing tiles that result in nonretangular tile configurations. |
| | Transparently handles a kind of error found in some data tiles just north of the equator (affecting some tiles with file names ending in n00.dt0). In these files, the NW and NE corners are incorrectly designated with south instead of north latitudes. |
| | Directory and filename calculations are more efficient. |
| km2sm, nm2sm, sm2km, sm2nm | These functions now use precise factors for conversion to/from statute miles. |
| | These changes also correct distdim results for both statute miles and feet (assuming U.S. Survey Foot). |
| tigerp | Terminates file read properly when an empty data point is encountered (signaling end of data) |
| worldmap | No longer generates errors when there is no data to display within the specified latitude/longitude limits; it now displays complete mesh, even for data grids (matrix maps) that span all longitudes |

**7**

# Mapping Toolbox 1.3
# Release Notes

# New Features

This section summarizes the new features and enhancements added in the Mapping Toolbox 1.3.

If you are upgrading from a release earlier than Release 12.0, then you should also see the Mapping Toolbox 1.2.1 Release Notes (for enhancements introduced between Release 11.0 and Release 12.0).

## New and Enhanced Map Display Functions

### New northarrow Annotation Function

The new `northarrow` function lets you add a north arrow symbol to a map, pointing to geographic north. You can reposition a north arrow by clicking and dragging it, or adjust other properties such as position, color, and size via alternate-clicking.

### New mlabelzero22pi Axes Label Command

The new `mlabelzero22pi` command converts meridian labels to use the range 0 to 360 degrees, instead of the default -180 to 180 degrees.

### Enhanced clegendm Contour Annotation Function

The `clegendm` funciton has been enhanced with a new, optional syntax that accepts a string indicating the contour line units. Alternatively, text label strings can be supplied for each and every contour level via a cell array.

### New Interactive Interface

The new `lightmui` function provides an interactive, graphical user interface to control the position of lights on a globe or 3-D map.

## Mapping Data Enhancement

### Updated Atlas Data

Political boundaries and country names have been updated in both the `worldlo` and `worldhi` atlas files.

## Other Enhancements

### Ellipses Now Drawn More Smoothly

The `ellipse1` function has been modified to create smoother ellipses. A
weighted distribution of azimuth points is used instead of the uniform
distribution between the starting and ending points. More points are added at
locations near the semi-major and semi-minor axes and fewer points at the
other intermediate locations.

# Major Bug Fixes

The Mapping Toolbox 1.3 includes several dozen bug fixes made since Version 1.2.1 Most are minor or cosmetic in nature. However, there are several significant Version 1.3 bug fixes.

If you are viewing these Release Notes in PDF form, please refer to the HTML form of the Release Notes, using either the Help browser or the MathWorks Web site and use the link provided.

# Upgrading from an Earlier Release

This section describes an upgrade issue involved in moving from the Mapping Toolbox 1.2.1 to Version 1.3.

## Obsolete Functions

Functions `contorm` and `contor3m` are now obsolete. You should now use `contourm` and `contour3m` instead, which are functionally identical. The older functions still exist, but have been removed from the Mapping Toolbox documentation.

**8**

# Mapping Toolbox 1.2 Release Notes

# New Features

This section introduces the new features and enhancements added in the Mapping Toolbox 1.2 since the Mapping Toolbox 1.1 (Release 11.0).

## Higher Resolution Atlas Data

There are now high-resolution country outlines and more city locations available through the `worldhi` database. The `worldmap` command automatically chooses this high-resolution data if the region's area is small enough.

The `worldlo` atlas file has been updated to make it coincide more closely with high-resolution coastlines and boundaries.

The `worldmtxmed` MAT-file provides a medium-resolution political world matrix map.

## External Data Interface Improved

Importing high-resolution atlas data is now much easier with these two visual interfaces:

- Digital Elevation Map Data user interface (invoked with the `demdataui` function)
- Vector Map Level 0 user interface (invoked with the `vmap0ui` function)

Many of the matrix map data interface functions now automatically concatenate data across separate files.

The external interface now supports the GLOBE digital elevation map data, a product similar to GTPO30. Use the `globedem` function for working with that data.

## New Interactive Interfaces

You can now adjust interactively, on a map display, geographic lines such as great circle tracks, small circles and sectors of small circles. Use the `trackg`, `scircleg`, and `sectorg` functions to make these interactive adjustments. While in an edit mode, you can drag the lines around on the map, modify the lines in a control panel, or read measurements.

There is a new visual interface to create colormaps. Use the `cmapui` function to invoke this new interface.

## New Analysis Functions for Geographic Data

You can use the new `elevation` function to find the elevation angle of a geographic point.

The new `gradientm` function performs matrix map data calculations, including gradient, slope, and aspect.

You can use the new `los2` and `viewshed` functions with terrain data to check the line of sight visibility between points or the visibility of entire regions.

Several new functions have been added to support polygon operations.

| Function | Description |
|----------|-------------|
| `bufferm` | Compute polygon buffer regions |
| `polybool` | Perform polygon Boolean operations |
| `polyjoin` | Combine polygon segments into a NaN-clipped polygon |
| `polymerge` | Merge polygon segments with abutting ends |
| `polysplit` | Separate NaN-clipped polygon segments into cell arrays |
| `polyxpoly` | Polygon intersections |

## Other New Functions

Several new functions have been added to support polygon operations.

| Function | Description |
| --- | --- |
| contourcmap | Contour-like color jumps on surface objects |
| driftcorr | Compute correction for drift |
| driftvel | Compute drift speed and direction |
| flatearthpoly | Cut polygons at dateline |
| lcolorbar | Labeled colorbar |
| mapprofile | Matrix map values along a path |
| str2angle | String conversion to angle |