FloPy Release Notes

Mark Bakker Vincent Post Christian D. Langevin Joseph D. Hughes
Jeremy T. White Andrew T. Leaf Scott R. Paulinski Joshua D. Larsen
Michael W. Toews Eric D. Morway Jason C. Bellino Jeffrey Starn
Michael N. Fienen

Introduction

FloPy includes support for MODFLOW 6, MODFLOW-2005, MODFLOW-NWT, MODFLOW-USG, and MODFLOW-2000. Other supported MODFLOW-based models include MODPATH (version 6 and 7), MT3DMS, MT3D-USGS, and SEAWAT.

For general modeling issues, please consult a modeling forum, such as the MODFLOW Users Group. Other MODFLOW resources are listed in the MODFLOW Resources section.

Contributing

Bug reports, code contributions, or improvements to the documentation are welcome from the community. Prior to contributing, please read up on our guidelines for contributing and then check out one of our issues in the hotlist: community-help.

Documentation

FloPy code documentation is available at http://modflowpy.github.io/flopydoc/

How to Cite

Citation for FloPy: Bakker, M., Post, V., Langevin, C. D., Hughes, J. D., White, J. T., Starn, J. J. and Fienen, M. N., 2016, Scripting MODFLOW Model Development Using Python and FloPy: Groundwater, v. 54, p. 733–739, doi:10.1111/gwat.12413.

Software/Code citation for FloPy: Bakker, M., Post, V., Langevin, C.D., Hughes, J.D., White, J.T., Starn, J.J., and Fienen, M.N., 2019, FloPy v3.2.13: U.S. Geological Survey Software Release, 17 November 2019, http://dx.doi.org/10.5066/F7BK19FH

Disclaimer

This software has been approved for release by the U.S. Geological Survey (USGS). Although the software has been subjected to rigorous review, the USGS reserves the right to update the software as needed pursuant to further analysis and review. No warranty, expressed or implied, is made by the USGS or the U.S. Government as to the functionality of the software and related material nor shall the fact of release constitute any such warranty. Furthermore, the software is released on condition that neither the USGS nor the U.S. Government shall be held liable for any damages resulting from its authorized or unauthorized use.

Installation

To install FloPy version 3.2.13 from the USGS FloPy website:

```
pip install https://water.usgs.gov/ogw/flopy/flopy-3.2.13.zip
```

To update to FloPy version 3.2.13 from the USGS FloPy website:

```
pip install https://water.usgs.gov/ogw/flopy/flopy-3.2.13.zip --upgrade
```

FloPy Supported Packages

MODFLOW-2000, MODFLOW-2005, and MODFLOW-NWT

Package	Creation and Write	Load Available	Template Creation
Basic (BAS6)	Supported	Supported	Not supported
Block Centered Flow (BCF)	Supported	Supported	Not supported
Time-Variant Specified-Head (CHD)	Supported	Supported	Not supported
Direct Solver (DE4)	Supported	Supported	Not supported
Discretization (DIS)	Supported	Supported	Not supported
Drain (DRN)	Supported	Supported	Not supported
Drain Return (DRT)	Supported	Supported	Not supported
Evapotranspiration (EVT)	Supported	Supported	Not supported
Evapotranspiration Segments (ETS)	Not supported	Not supported	Not supported
Flow and Head Boundary (FHB)	Supported	Supported	Not supported
General Head Boundary (GHB)	Supported	Supported	Not supported
Geometric Multi-Grid (GMG)	Supported	Supported	Not supported
Horizontal Flow Barrier (HFB)	Supported	Supported	Not supported
Hydrogeologic-Unit Flow (HUF)	Not supported	Not supported	Not supported
Interbed-Storage (IBS)	Not supported	Not supported	Not supported
Lake (LAK)	Supported	Supported	Not supported
Layer Property Flow (LPF)	Supported	Supported	Supported
Link-AMG (LMG)	Not supported	Not supported	Not supported
MODFLOW Link-MT3DMS (LMT)	Supported	Supported	Not supported
Multiplier (MULT)	Not supported	Supported	Not supported
Multi-Node Well 1 (MNW1)	Supported	Supported	Not supported
Multi-Node Well 2 (MNW2)	Supported	Supported	Not supported
Multi-Node Well Information (MNWI)	Supported	Supported	Not supported
Newton (NWT)	Supported	Supported	Not supported
Output Control (OC)	Supported	Supported	Not supported
Periodic Boundary Condition (PBC)	Supported	Not supported	Not supported
Preconditioned Conjugate Gradient (PCG)	Supported	Supported	Not supported
Preconditioned Conjugate Gradient Nonlinear (PCGN)	Supported	Supported	Not supported
Parameter Value (PVAL)	Not supported	Supported	Not supported
Recharge (RCH)	Supported	Supported	Not supported
River (RIV)	Supported	Supported	Not supported
Streamflow Routing (SFR2)	Supported	Supported	Not supported
Strongly Implicit Procedure (SIP)	Supported	Not supported	Not supported
Slice-successive Overrelaxation (SOR)	Supported	Not supported	Not supported
Stream (STR)	Supported	Supported	Not supported
Seawater Intrusion (SWI2)	Supported	Supported	Not supported
Surface-Water Routing (SWR)	Not supported	Not supported	Not supported
Subsidence (SUB)	Supported	Supported	Not supported
Subsidence and Aquifer-System Compaction (SWT)	Supported	Supported	Not supported
Upstream Weighted (UPW)	Supported	Supported	Not supported
Unsaturated Zone Flow (UZF)	Supported	Supported	Not supported
Well (WEL)	Supported	Supported	Not supported
Zone (ZONE)	Not supported	Supported	Not supported

MODFLOW-USG

Package	Creation and Write	Load Available	Template Creation
Unstructured Discretization (DISU)	Supported	Supported	Not supported
Sparse Matrix Solver (SMS)	Supported	Supported	Not supported

MODPATH 6

Package	Creation and Write	Load Available	Template Creation
MODPATH Basic (MPBAS)	Supported	Not supported	Not supported
MODPATH Simulation (MPSIM)	Supported	Not supported	Not supported

MODPATH 7

Package	Creation and Write	Load Available	Template Creation
MODPATH Basic (MPBAS)	Supported	Not supported	Not supported
MODPATH Simulation (MPSIM)	Supported	Not supported	Not supported
Starting Location Data - Input Style 1	Supported	Not supported	Not supported
Starting Location Data - Input Style 2	Supported	Not supported	Not supported
Starting Location Data - Input Style 3	Supported	Not supported	Not supported
Starting Location Data - Input Style 4	Not supported	Not supported	Not supported

MT3DMS, MT3D-USGS

Package	Creation and Write	Load Available	Template Creation
Advection (ADV)	Supported	Supported	Not supported
Basic Transport (BTN)	Supported	Supported	Not supported
Dispersion (DSP)	Supported	Supported	Not supported
Generalized Conjugate Gradient (GCG)	Supported	Supported	Not supported
Lake (LKT)	Supported	Supported	Not supported
PHT3D-PHREEQC Interface (PHC)	Supported	Not supported	Not supported
Streamflow (SFT)	Supported	Supported	Not supported
Reaction (RCT)	Supported	Supported	Not supported
Sink and Source Mixing (SSM)	Supported	Supported	Not supported
Transport Observation (TOB)	Supported	Not supported	Not supported
Unsaturated-zone (UZT)	Supported	Supported	Not supported

SEAWAT

Package	Creation and Write	Load Available	Template Creation
Variable Density Flow (VDF)	Supported	Supported	Not supported
Viscosity (VSC)	Supported	Supported	Not supported

MODFLOW-2000, MODFLOW-2005, and MODFLOW-NWT Observations

Package	Creation and Write	Load Available	Template Creation
Drain Observation (DROB)	Not supported	Not supported	Not supported
HYDMOD (HYD)	Supported	Supported	Not supported
Gage (GAGE)	Supported	Supported	Not supported
General Head Boundary Observation (GBOB)	Supported	Not supported	Not supported

Package	Creation and Write	Load Available	Template Creation
Head Observation (HOB)	Supported	Supported	Not supported
River Observation (RVOB)	Supported	Not supported	Not supported
Stream Observation (STOB)	Supported	Not supported	Not supported
Specified-Head Flow Observation (CHOB)	Supported	Not supported	Not supported

MODFLOW 6

Package	Creation and Write	Load Available
Temporal Discretization (TDIS6)	Supported	Supported
Structured Discretization (DIS6)	Supported	Supported
Discretization with Vertices (DISV6)	Supported	Supported
Unstructured Discretization (DISU6)	Supported	Supported
Initial Conditions (IC6)	Supported	Supported
Output Control (OC6)	Supported	Supported
Groundwater Flow Observations (OBS6)	Supported	Supported
Node Property Flow (NPF6)	Supported	Supported
Horizontal Flow Barrier (HFB6)	Supported	Supported
Storage (STO6)	Supported	Supported
Constant-Head (CHD6)	Supported	Supported
Constant-Head Observations (OBS6)	Supported	Supported
Well (WEL6)	Supported	Supported
Well Observations (OBS6)	Supported	Supported
Drain (DRN6)	Supported	Supported
Drain Observations (OBS6)	Supported	Supported
River (RIV6)	Supported	Supported
River Observations (OBS6)	Supported	Supported
General-Head-Boundary (GHB6)	Supported	Supported
General-Head-Boundary Observations (OBS6)	Supported	Supported
Recharge (RCH6) - List-Based	Supported	Supported
Recharge (RCH6) - Array-Based	Supported	Supported
Recharge Observations (OBS6)	Supported	Supported
Evapotranspiration (EVT6) - List-Based	Supported	Supported
Evapotranspiration (EVT6) - Array-Based	Supported	Supported
Evapotranspiration Observations (OBS6)	Supported	Supported
Multi-Aquifer Well (MAW6)	Supported	Supported
Multi-Aquifer Well Observations (OBS6)	Supported	Supported
Streamflow Routing (SFR6)	Supported	Supported
Streamflow Routing Observations (OBS6)	Supported	Supported
Lake Package (LAK6)	Supported	Supported
Lake Table Input	Supported	Supported
Lake Observations (OBS6)	Supported	Supported
Unsaturated Zone Flow (UZF6)	Supported	Supported
Unsaturated Zone Flow Observations (OBS6)	Supported	Supported
Water Mover (MVR6)	Supported	Supported
Ghost-Node Correction (GNC6)	Supported	Supported
Groundwater Flow Exchange (GWF-GWF)	Supported	Supported
Iterative Model Solution (IMS6)	Supported	Supported
Timeseries File (TS6)	Supported	Supported

FloPy Model Checks

List of available FloPy model checks

Package	Check	Implemented	Type
NAM	unit number conflicts	Supported	Error
NAM	compatible solver package	Supported	Error
NAM	minimum packages needed to run the model	Not supported	Error
all BC packages	overlapping boundary conditions	Not supported	Error
all BC packages	NaN values in stress_period_data	Supported	Error
all BC packages	valid indices for stress_period_data	Supported	Error
LPF/UPW	hk or vka <=0	Supported	Error
LPF/UPW	hani < 0	Supported	Error
LPF/UPW	vkcb (quasi-3D kv values) <=0	Supported	Error
LPF/UPW	unusually high or low values in hk and vka arrays	Supported	Warning
LPF/UPW	unusually high or low values in vkcb (quasi-3D kv values)	Supported	Warning
LPF/UPW	storage values <=0 (transient only)	Supported	Error
LPF/UPW	unusual values of storage (transient only)	Supported	Error
RIV/SFR/STR	check for surface water BCs in confined layers	Not supported	Warning
BAS	isolated cells	Supported	Warning
BAS	NaN values	Supported	Error
DIS	cell thicknesses <= 0	Supported	Error
DIS	cell thicknesses < thin_cell_threshold (default 1.0)	Supported	Warning
DIS	NaN values in top and bottom arrays	Supported	Error
DIS	discretization that violates the 1.5 rule	Not supported	Warning
DIS	large changes in elevation	Not supported	Warning
DISU	large changes in elevation	Not supported	Warning
DISU	cell thicknesses <= 0	Not supported	Error
DISU	cell thicknesses < thin_cell_threshold (default 1.0)	Not supported	Warning
DISU	NaN values in top and bottom arrays	Not supported	Error
DISU	discretization that violates the 1.5 rule	Not supported	Warning
DISU	large changes in elevation	Not supported	Warning
MNW2	ITMP >= 0 for first stress period	Supported	Error
MNW2	ITMP > MNWMAX	Supported	Error

Package	Check	Implemented	Type
MNWI	MNWI present without MNW2 package	Supported	Warning
RCH	unusually high or low R/T ratios	Supported	Warning
RCH	NRCHOP not specified as 3	Supported	Warning
SFR	continuity in segment and reach numbering	Supported	Error
SFR	segment number decreases in downstream direction	Supported	Warning
SFR	circular routing	Supported	Error
SFR	multiple non-zero conductances in a model cell	Supported	Warning
SFR	elevation increases in the downstream direction	Supported	Error
SFR	streambed elevations above model top	Supported	Warning
SFR	streambed elevations below cell bottom	Supported	Error
SFR	negative stream depth when icalc=0	Not supported	Error
SFR	slopes above or below specified threshold	Supported	Warning
SFR	unusual values for manning’s roughness and unit constant	Not supported	Warning
SFR	gaps in segment and reach routing	Not supported	Warning
SFR	outlets in interior of model domain	Not supported	Warning
WEL	PHIRAMP is < 1 and should be close to recommended value of 0.001	Not supported	Warning
MPSIM	invalid stop times	Supported	

Visualizations

Package	Check	Implemented	Type
All	Shapefile with detected errors	Not supported	Information
All	Shapefile with detected warnings	Not supported	Information
SFR/STR	Segment Connectivity	Not supported	Information
SFR/STR	Identification of diversions	Not supported	Information
SFR/STR	Identification of outlet tributaries	Not supported	Information

Additional model checks and visualizations

Please submit additional proposed model checks as issues on the FloPy development branch on github.

FloPy Changes

Version 3.2.13

- ModflowFlwob: Variable `irefsp` is now a zero-based integer (#596)
- ModflowFlwob: Added a load method and increased precision of `toffset` when writing to file (#598)
- New feature GridIntersect (#610): The GridIntersect object allows the user to intersect shapes (Points, LineStrings and Polygons) with a MODFLOW grid. These intersections return a numpy.recarray containing the intersection information, i.e. cell IDs, lengths or areas, and a shapely representation of the intersection results. Grids can be structured or vertex grids. Two intersections methods are implemented: `"structured"` and `"strtree"`: the former accelerates intersections with structured grids. The latter is more flexible and also works for vertex grids. The GridIntersect class is available through `flopy.utils.gridintersect`. The functionality requires the Shapely module. See the example notebook for an overview of this new feature.
- New feature `Raster` (#634): The Raster object allows the user to load raster files (Geotiff, arc ascii, .img) and sample points, sample polygons, create cross sections, crop, and resample raster data to a Grid. Cropping has been implemented using a modified version of the ray casting algorithm for speed purposes. Resampling a raster can be performed with structured, vertex, and unstructured Grids. Rasters will return a numpy array of resampled data in the same shape as the Grid. The Raster class is available by calling `flopy.utils.Raster`. The functionality requires rasterio, affine, and scipy. See the (example notebook) for an overview of this feature.
- Modify NAM and MFList output files to remove excessive whitespace (#622, #722)
- Deprecate `crs` class from `flopy.utils.reference` in favor of `CRS` class from `flopy.export.shapefile_utils` (#608)
- New feature in `PlotCrossSection` (#660). Added a `geographic_coords` parameter flag to `PlotCrossSection` which allows the user to plot cross sections with geographic coordinates on the x-axis. The default behavior is to plot distance along cross sectional line on the x-axis. See the (example notebook) for an overview of this feature.
- New feature with binaryfile readers, including `HeadFile` and `CellBudgetFile` (#669): `with` statement is supported to open files for reading in a context manager, which automatically close when done or if an exception is raised.
- Improved the flopy list reader for MODFLOW stress packages (for mf2005, mfnwt, etc.), which may use SFAC to scale certain columns depending on the package. The list reading now supports reading from external files, in addition to open/close. The (binary) option is also supported for both open/close and external. This new list reader is used for reading standard stress package lists and also lists used to create parameters. The new list reader should be consistent with MODFLOW behavior.
- SfrFile detects additional columns (#708)
- Add a `default_float_format` property to mfsfr2, which is string formatted by NumPy versions > 1.14.0, or `{:.8g}` for older NumPy versions (#710)
- Support for pyshp 1.2.1 dropped, pyshp 2.1.0 support maintained
- Improved VTK export capabilities. Added export for VTK at array level, package level, and model level. Added binary head file export and cell by cell file export. Added the ability to export point scalars in addition to cell scalars, and added smooth surface generation. VTK export now supports writing transient data as well as exporting to binary .vtu files.
- Support for copying model and package instances with `copy.deepcopy()`
- Added link to Binder on README and notebooks_examples markdown documents. Binder provides an environment that runs and interactively serves the FloPy Jupyter notebooks.
- Bug fixes:
  - When using the default `iuzfbnd=None` in the `__init__` routine of mtuzt.py, instantiation of IUZBND was generating a 3D array instead of a 2D array. Now generates a 2D array

  - ModflowSfr2 `__init__` was being slowed considerably by the `ModflowSfr2.all_segments` property method. Modified the `ModflowSfr2.graph` property method that describes routing connections between segments to handle cases where segments aren't listed in stress period 0.
  - Ensure disordered fields in `reach_data` (Dataset 2) can be supported in `ModflowSfr2` and written to MODFLOW SFR input files.
  - When loading a MF model with UZF active, item 8 ("`[IUZROW] [IUZCOL] IFTUNIT [IUZOPT]`") wasn't processed correctly when a user comment appeared at the end of the line
  - MODFLOW-6 DISU JA arrays are now treated as zero-based cell IDs. JA, IHC, CL12 are outputted as jagged arrays.
  - Models with multiple MODFLOW-6 WEL packages now load and save correctly.
  - Exporting individual array and list data to a shapefile was producing an invalid attribute error. Attribute reference has been fixed.
  - Fix UnboundLocalError and typo with `flopy.export.shapefile_utils.CRS` class (#608)
  - Fix Python 2.7 issue with `flopy.export.utils.export_contourf` (#625)
  - When loading a MT3D-USGS model, keyword options (if used) were ignored (#649)
  - When loading a modflow model, spatial reference information was not being passed to the SpatialReference class (#659)
  - Fix specifysurfk option in UZF, ModflowUZF1 read and write surfk variable
  - Fix minor errors in flopy gridgen wrapper
  - Close opened files after loading, to reduce `ResourceWarning` messages (#673)
  - Fix bugs related to flake8's F821 "undefined name 'name'", which includes issues related to Mt3dPhc, ModflowSfr2, ModflowDe4, ListBudget, and ModflowSms (#686)
  - Fix bugs related to flake8's F811 "redefinition of unused 'name'" (#688)
  - Fix bugs related to flake8's W605 "invalid escape sequence '\s'" (or similar) (#700)
  - Fix EpsgReference class behavior with JSON user files (#702)
  - Fix ModflowSfr2 read write logic for all combinations of isfropt and icalc
  - IRCH array of the Recharge Package is now a zero-based variable, which means an IRCH value of 0 corresponds to the top model layer (#715)
  - MODFLOW lists were not always read correctly if they used the SFAC or binary options or were used to define parameters (#683)
  - Changed VDF Package density limiter defaults to zero (#646)

Version 3.2.12

- Added a check method for OC package (#558)
- Change default map projection from EPSG:4326 to None (#535)
- Refactor warning message visibility and categories (#554, #575)
- Support for MODFLOW 6 external binary files added. Flopy can read/write binary files containing list and array data (#470, #553).
- Added silent option for MODFLOW 6 write_simulation (#552)
- Refactored MODFLOW-6 data classes. File writing operations moved from mfdata*.py to new classes created in mffileaccess.py. Data storage classes moved from mfdata.py to mfdatastorage.py. MFArray, MFList, and MFScalar interface classes simplified with most of the data processing code moved to mfdatastorage.py and mffileaccess.py.
- Added MODFLOW 6 quickstart example to front page.
- Added lgrutil test as autotest/t063_test_lgrutil.py and implemented a get_replicated_parent_array() method to the Lgr class so that the user can pass in a parent array and get back an array that is the size of the child model.
- Refactored much of the flopy code style to conform with Python conventions and those checked by Codacy. Added an automated Codacy check as part of the pull request and commit checks.
- Bug fixes:
  - Fixed bug in Mt3dms.load to show correct error message when loading non-existent NAM file (#545)

  - Removed errant SFT parameter contained in Mt3dUzt.___init___ routine (#572)
  - Fixed DISV shapefile export bug that applied layer 1 parameter values to all model layers during export (#508)
  - Updated ModflowSfr2.load to store channel_geometry and channel_flow_data (6d, 6e) by nseg instead of itmp position (#546)
  - Fixed bug in ModflowMnw2.make_node_data to be able to set multiple wells with different numbers of nodes (#556)
  - Fixed bug reading MODFLOW 6 comma separated files (#509)
  - Fixed bug constructing a grid class with MODFLOW-USG (#513)
  - Optimized performance of grid class by minimizing redundant operations through use of data result caching (#520)
  - Fixed bug passing multiple auxiliary variables for MODFLOW 6 array data (#533)
  - Fixed bug in Mt3dUzt.___init___; the variable ioutobs doesn't exist in the UZT package and was removed.
  - Fixed MODFLOW-LGR bug in which ascii files were not able to be created for some output. Added better testing of the MODFLOW-LGR capabilities to t035_test.py.
  - Fixed multiple issues in mfdis that resulted in incorrect row column determination when using the method get_rc_from_node_coordinates (#560). Added better testing of this to t007_test.py.
  - Fixed the export_array_contours function as contours would not export in some cases (#577). Added tests of export_array_contours and export_array to t007_test.py as these methods were not tested at all.

Version 3.2.11

- Added support for the drain return package.
- Added support for pyshp version 2.x, which contains a different call signature for the writer than earlier versions.
- Added a new flopy3_MT3DMS_examples notebook, which uses Flopy to reproduce the example problems described in the MT3DMS documentation report by Zheng and Wang (1999).
- Pylint is now used on Travis for the Python 3.5 distribution to check for coding errors.
- Added testing with Python 3.7 on Travis, dropped testing Python 3.4.
- Added a new htop argument to the vtk writer, which allows cell tops to be defined by the simulated head.
- Generalized exporting and plotting to also work with MODFLOW 6. Added a new grid class and deprecated SpatialReference class. Added new plotting interfaces, `PlotMapView` and `PlotCrossSection`. Began deprecation of `ModelMap` and `ModelCrossSection` classes.
- Spatial reference system cache moved to epsgref.json in the user's data directory.
- Attempts to read empty files from flopy.utils raise a IOError exception.
- Changed interface for creating and accessing MODFLOW 6 observation, time series, and time array series packages. These packages can now be created and accessed directly from the package that references them. These changes are not backward compatible, and will require existing scripts to be modified. See the flopy3_mf6_obs_ts_tas.ipynb notebook for instructions.
- Changed the MODFLOW 6 fname argument to be filename. This change is not backward compatible, and will require existing scripts to be modified if the fname argument was used in the package constructor.
- Added modflow-nwt options support for `ModflowWel`, `ModflowSfr2`, and `ModflowUzf1` via the `OptionBlock` class.
- Bug fixes:
  - Removed variable MXUZCON from `mtuzt.py` that was present during the development of MT3D-USGS, but was not included in the release version of MT3D-USGS.
  - Now account for UZT -> UZT2 changes with the release of MT3D-USGS 1.0.1. Use of UZT is no longer supported.
  - Fixed bug in `mfuzf1.py` when reading and writing `surfk` when `specifysurfk = True`.
  - Fixed bug in `ModflowStr.load()`, utility would fail to load when comments were present.
  - Fixed bug in MNW2 in which nodes were not sorted correctly.
  - Ensure that external 1-D free arrays are written on one line.

  - Typos corrected for various functions, keyword arguments, property names, input file options, and documentation.
  - Fixed bug in `Mt3dUzt.__init__` that originated when copying code from mtsft.py to get started on mtuzt.py class. The variable ioutobs doesn't exist in the UZT package and should never have appeared in the package to begin with.

Version 3.2.10

- Added parameter_load variable to `mbase` that is set to true if parameter data are applied in the model (only used in models that support parameters). If this is set to `True` `free_format_input` is set to `True` (if currently `False`) when the `write_input()` method is called. This change preserves the precision of parameter data (which is free format data).
- MODFLOW 6 model and simulation packages can not be retrieved as a `MFSimulation` attribute
- Added support for multicomponent load in `mfsft.py`
- Added functionality to read esri-style epsg codes from spatialreference.org.
- Added functionality to MODFLOW 6 that will automatically replace the existing package with the one being added if it has the same name as the existing package.
- Added separate MODFLOW 6 model classes for each model type. Model classes contain name file options.
- Added standard `run_model()` method arguments to mf6 `run_simulation()` method.
- some performance improvements to checking
- `SpatialReference.export_array()` now writes 3-D numpy arrays to multiband GeoTiffs
- Add load support to for MNW1; ModflowMnw1 now uses a `stress_period_data` `Mflist` to store MNW information, similar to other BC packages.
- Added a Triangle class that is a light wrapper for the Triangle program for generating triangular meshes. Added a notebook called flopy3_triangle.ipynb that demonstrates how to use it and build a MODFLOW 6 model with a triangular mesh. The current version of this Triangle class should be considered beta functionality as it is likely to change.
- Added support for MODPATH 7 (beta).
- Added support for MODPATH 3 and 5 pathline and endpoint output files.
- Added support for MODPATH timeseries output files (`flopy.utils.TimeseriesFile()`).
- Added support for plotting MODPATH timeseries output data (`plot_timeseries()`) with ModelMap.
- Bug fixes:
  - Fixed issue in HOB when the same layer is specified in the `MLAY` data (dataset 4). If the layer exists the previous fraction value is added to the current value.
  - Fixed bug in segment renumbering
  - Changed default value for `ioutobs **kwargs` in `mtsft.py` from None to 0 to prevent failure.
  - Fixed bug when passing extra components info from load to constructor in `mtsft.py` and `mtrct.py`.
  - Fixed bug in `mt3ddsp` load - if `multidiffusion` is not found, should only read one 3d array.
  - Fixed bug in `zonbud` utility that wasn't accumulating flow from constant heads.
  - Fixed minor bug that precluded the passing of mass-balance record names (`TOTAL_IN`, `IN-OUT`, etc.).
  - Fixed bug when writing shapefile projection (`.prj`) files using relative paths.
  - Fixed bugs in `sfr.load()` – `weight` and `flwtol` should be cast as floats, not integers.
  - Fixed bug when `SpatialReference` supplied with geographic CRS.
  - Fixed bug in `mfsfr.py` when writing kinematic data (`irtflg >0`).
  - Fixed issue from change in MODFLOW 6 `inspect.getargspec()` method (for getting method arguments).
  - Fixed MODFLOW 6 BINARY keyword for reading binary data from a file using `OPEN/CLOSE` (needs parentheses around it).
  - Fixed bug in `mtlkt.py` when initiating, loading, and/or writing lkt input file related to multi-species problems.

Version 3.2.9

- Modified MODFLOW 5 OC stress_period_data=None default behaviour. If MODFLOW 5 OC stress_period_data is not provided then binary head output is saved for the last time step of each stress period.
- added multiple component support to `mt3dusgs SFT` module
- Optimized loading and saving of MODFLOW 6 files
- MODFLOW 6 identifiers are now zero based
- Added remove_package method in MFSimulation and MFModel that removes MODFLOW 6 packages from the existing simulation/model
- Changed some of the input argument names for MODFLOW 6 classes. Note that this will break some existing user scripts. For example, the stress period information was passed to the boundary package classes using the periodrecarray argument. The argument is now called stress_period_data in order to be consistent with other Flopy functionality.
- Flopy code for MODFLOW 6 generalized to support different model types
- Flopy code for some MODFLOW 6 arguments now have default values in order to be consistent with other Flopy functionality
- Added `ModflowSfr2.export_transient_variable` method to export shapefiles of segment data variables, with stress period data as attributes
- Added support for UZF package gages
- Bug fixes:
  - Fixed issue with default settings for MODFLOW 5 SUB package `dp` dataset.
  - Fixed issue if an external BC list file has only one entry
  - Some patching for recarray issues with latest `numpy` release (there are more of these lurking...)
  - Fixed setting model relative path for MODFLOW 6 simulations
  - Python 2.7 compatibility issues fixed for MODFLOW 6 simulations
  - IMS file name conflicts now automatically resolved
  - Fixed issue with passing in numpy ndarrays arrays as layered data
  - Doc string formatting for MODFLOW 6 packages fixed to make doc strings easier to read
  - UZF package: fixed issues with handling of finf, pet, extdp and extwc arrays.
  - SFR package: fixed issue with reading stress period data where not all segments are listed for periods > 0.
  - `SpatialReference.write_gridSpec` was not converting the model origin coordinates to model length units.
  - shorted integer field lengths written to shapefiles to 18 characters; some readers may misinterpret longer field lengths as float dtypes.

Version 3.2.8

- Added `has_package(name)` method to see if a package exists. This feature goes nicely with `get_package(name)` method.
- Added `set_model_units()` method to change model units for all files created by a model. This method can be useful when creating MODFLOW-LGR models from scratch.
- Added SFR2 package functionality
  - `export_inlets` method to write shapefile showing locations where external flows are entering the stream network.
- Bug fixes:
  - Installation: Added dfn files required by MODFLOW 6 functionality to MANIFEST.in so that they are included in the distribution.
  - SFR2 package: Fixed issue reading transient data when `ISFOPT` is 4 or 5 for the first stress period.

Version 3.2.7

- Added beta support for MODFLOW 6 See here for more information.
- Added support for retrieving time series from binary cell-by-cell files. Cell-by-cell time series are accessed in the same way they are accessed for heads and concentrations but a text string is required.
- Added support for FORTRAN free format array data using n*value where n is the number of times value is repeated.
- Added support for comma separators in 1D data in LPF and UPF files
- Added support for comma separators on non array data lines in DIS, BCF, LPF, UPW, HFB, and RCH Packages.
- Added `.reset_budgetunit()` method to OC package to facilitate saving cell-by-cell binary output to a single file for all packages that can save cell-by-cell output.
- Added a `.get_residual()` method to the `CellBudgetFile` class.
- Added support for binary stress period files (`OPEN/CLOSE filename (BINARY)`) in `wel` stress packages on load and instantiation. Will extend to other list-based MODFLOW stress packages.
- Added a new `flopy.utils.HeadUFile` Class (located in binaryfile.py) for reading unstructured head files from MODFLOW-USG. The `.get_data()` method for this class returns a list of one-dimensional head arrays for each layer.
- Added metadata.acdd class to fetch model metadata from ScienceBase.gov and manage CF/ACDD-complaint metadata for NetCDF export
- Added sparse export option for boundary condition stress period data, where only cells for that B.C. are exported (for example, `package.stress_period_data.export('stuff.shp', sparse=True)`)
- Added additional SFR2 package functionality:
  - `.export_linkages()` and `.export_outlets()` methods to export routing linkages and outlets
  - sparse shapefile export, where only cells with SFR reaches are included
  - `.plot_path()` method to plot streambed elevation profile along sequence of segments
  - `.assign_layers()` method
  - additional error checks and bug fixes
- Added `SpatialReference` / GIS export functionality:
  - GeoTiff export option to `SpatialReference.export_array`
  - `SpatialReference.export_array_contours`: contours an array and then exports contours to shapefile
  - inverse option added to `SpatialReference.transform`
  - automatic reading of spatial reference info from .nam or usgs.model.reference files
- Modified node numbers in SFR package and `ModflowDis.get_node()` from one- to zero-based.
- Modified HYDMOD package `klay` variable from one- to zero-based.
- Added `.get_layer()` method to DIS package.
- Added `.get_saturated_thickness()` and `.get_gradients()` methods
- Bug fixes:
  - OC package: Fixed bug when printing and saving data for select stress periods and timesteps. In previous versions, OC data was repeated until respecified.
  - SUB package: Fixed bug if data set 15 is passed to preserved unit numbers (i.e., use unit numbers passed on load).
  - SUB and SUB-WT packages: Fixed bugs `.load()` to pop original unit number.
  - BTN package: Fixed bug in obs.
  - LPF package: Fixed bug regarding when HANI is read and written.
  - UZF package: added support for MODFLOW NWT options block; fixed issue with loading files with thti/thtr options
  - SFR package: fixed bug with segment renumbering, issues with reading transient text file output,
  - Fixed issues with dynamic setting of `SpatialReference` parameters
  - NWT package: forgive missing value for MXITERXMD
  - MNW2 package: fix bug where ztop and zbotm were written incorrectly in `get_allnode_data()`. This was not affecting writing of these variables, only their values in this summary array.
  - PCGN package: fixed bug writing package.
  - Fixed issue in `Util2d` when non-integer `cnstnt` passed.

Version 3.2.6

- Added functionality to read binary grd file for unstructured grids.
- Additions to SpatialReference class:
  - xll, yll input option
  - transform method to convert model coordinates to real-world coordinates
  - epsg and length_multiplier arguments
- Export:
  - Added writing of prj files to shapefile export; prj information can be passed through spatial reference class, or given as an EPSG code or existing prj file path
  - Added NetCDF export to MNW2
- Added MODFLOW support for:
  - FHB Package - no support for flow or head auxiliary variables (datasets 2, 3, 6, and 8)
  - HOB Package
- New utilities:
  - `flopy.utils.get_transmissivities()` Computes transmissivity in each model layer at specified locations and open intervals. A saturated thickness is determined for each row, column or x, y location supplied, based on the well open interval (sctop, scbot), if supplied, otherwise the layer tops and bottoms and the water table are used.
- Added MODFLOW-LGR support - no support for model name files in different directories than the directory with the lgr control file.
- Additions to MODPATH:
  - shapefile export of MODPATH Pathline and Endpoint data
  - Modpath.create_mpsim() supports MNW2
  - creation of MODPATH StartingLocations files
  - Easy subsetting of endpoint and pathline results to destination cells of interest

- New ZoneBudget class provides ZONEBUDGET functionality:
  - reads a CellBudgetFile and accumulates flows by zone
  - pass `kstpkper` or `totim` keyword arguments to retrieve a subset of available times in the CellBudgetFile
  - includes a method to write the budget recarrays to a .csv file
  - ZoneBudget objects support numerical operators to facilitate conversion of units
  - utilities are included which read/write ZONEBUDGET-style zone files to and from numpy arrays
  - pass a dictionary of {zone: “alias”} to rename fields to more descriptive names (e.g. {1: ‘New York’, 2: ‘Delmarva’}
- Added new precision=‘auto’ option to flopy.utils.binaryfile for HeadFile and UcnFile readers. This will automatically try and determine the float precision for head files created by single and double precision versions of MODFLOW. ‘auto’ is now the default. Not implemented yet for cell by cell flow file.
- Modified MT3D-related packages to also support MT3D-USGS
  - BTN will support the use of keywords (e.g., ‘MODFLOWStyleArrays’, etc.) on the first line
  - DSP will support the use of keyword NOCROSS
  - Keyword FREE now added to MT3D name file when the flow-transport link (FTL) file is formatted. Previously defaulted to unformatted only.
- Added 3 new packages:
  - SFT: Streamflow Transport, companion transport package for use with the SFR2 package in MODFLOW
  - LKT: Lake Transport, companion transport package for use with the LAK3 package in MODFLOW
  - UZT: Unsaturated-zone Transport, companion transport package for use with the UZF1 package in MODFLOW
- Modified LMT
  - load() functionality will now support optional PACKAGE_FLOWS line (last line of LMT input)
  - write_file() will will now insert PACKAGE_FLOWS line based on user input
- Bug fixes:
  - Fixed bug in parsenamefile when file path in namefile is surrounded with quotes.
  - Fixed bug in check routine when THICKSTRT is specified as an option in the LPF and UPW packages.
  - Fixed bug in BinaryHeader.set_values method that prevented setting of entries based on passed kwargs.
  - Fixed bugs in reading and writing SEAWAT Viscosity package.
  - The DENSE and VISC arrays are now Transient3d objects, so they may change by stress period.

  - MNW2: fixed bug with k, i, j node input option and issues with loading at model level
  - Fixed bug in ModflowDis.get_cell_volumes().

Version 3.2.5

- Added support for LAK and GAGE packages - full load and write functionality supported.
- Added support for MNW2 package. Load and write of .mnw2 package files supported. Support for .mnwi, or the results files (.qsu, .byn) not yet implemented.
- Improved support for changing the output format of arrays and variables written to MODFLOW input files.
- Restructured SEAWAT support so that packages can be added directly to the SEAWAT model, in addition to the approach of adding a modflow model and a mt3d model. Can now load a SEAWAT model.
- Added load support for MT3DMS Reactions package
- Added multi-species support for MT3DMS Reactions package
- Added static method to Mt3dms().load_mas that reads an MT3D mass file and returns a recarray
- Added static method to Mt3dms().load_obs that reads an MT3D mass file and returns a recarray
- Added method to flopy.modpath.Modpath to create modpath simulation file from modflow model instance boundary conditions. Also added examples of creating modpath files and post-processing modpath pathline and endpoint files to the flopy3_MapExample notebook.
- Bug fixes:
  - Fixed issue with VK parameters for LPF and UPW packages.
  - Fixed issue with MT3D ADV load in cases where empty fields were present in the first line of the file.
  - Fixed cross-section array plotting issues.
  - BTN observation locations must now be entered in zero-based indices (a 1 is now added to the index values written to btn file)
  - Uploaded supporting files for SFR example notebook; fixed issue with segment_data submitted as array (instead of dict) and as 0d array(s).
  - Fixed CHD Package so that it now supports options, and therefore, auxiliary variables can be specified.
  - Fixed loading BTN save times when numbers are touching.

Version 3.2.4

- Added basic model checking functionality (`.check()`).
- Added support for reading SWR Process observation, stage, budget, flow, reach-aquifer exchanges, and structure flows.
- `flopy.utils.HydmodObs` returns a numpy recarray. Previously numpy arrays were returned except when the `slurp()` method was used. The slurp method has been deprecated but the same functionality is available using the `get_data()` method. The recarray returned from the `get_data()` method includes the `totim` value and one or all of the observations (`HYDLBL`).
- Added support for MODFLOW-USG DISU package for unstructured grids.
- Added class (`Gridgen`) for creating layered quadtree grids using GRIDGEN (`flopy.utils.gridgen`). See the flopy3_gridgen notebook for an example of how to use the `Gridgen` class.
- Added user-specified control on use of `OPEN/CLOSE` array options (see flopy3_external_file_handling notebook).
- Added user-specified control for array output formats (see flopy3_array_outputformat_options IPython notebook).
- Added shapefile as optional output format to `.export()` method and deprecated `.to_shapefile()` method.
- Bug fixes:
  - Fixed issue with right justified format statement for array control record for MT3DMS.
  - Fixed bug writing PHIRAMP for MODFLOW-NWT well files.

  - Fixed bugs in NETCDF export methods.
  - Fixed bugs in LMT and BTN classes.

Version 3.2.3

- Added template creation support for several packages for used with PEST (and UCODE).
- Added support for the SEAWAT viscosity (VSC) package.
- Added support for the MODFLOW Stream (STR), Streamflow-Routing (SFR2), Subsidence (SUB), and Subsidence and Aquifer-System Compaction Package for Water-Table Aquifers (SWT) Packages.
- Mt3d model was redesigned based on recent changes to the Modflow model. Mt3d packages rewritten to support multi-species. Primary packages can be loaded (btn, adv, dsp, ssm, gcg). Array utilities modified to read some MT3D RARRAY formats.
- Fixed array loading functionality for case when the CNSTNT value is zero. If CNSTNT is zero and is used as an array multiplier, it is changed to 1 (as done in MODFLOW).
- Added support for the MODFLOW HYDMOD (HYD) Package and reading binary files created by the HYDMOD Package (`HydmodObs` Class) in the `flopy.utils` submodule.
- `flopy.utils.CellBudgetFile` returns a numpy recarray for list based budget data. Previously a dictionary with the `node` number and `q` were returned. The recarray will return the `node` number, `q`, and the `aux` variables for list based budget data.
- Added travis-ci automated testing.

Version 3.2.2

- FloPy now supports some simple plotting capabilities for two- and three-dimensional model input data array classes and transient two-dimensional stress period input data using the `.plot()` methods associated with the data array classes (`util_2d`, `util_3d`, and `transient_2d`). The model results reader classes (`HeadFile`, `UcnFile`, and `CellBudgetFile`) have also been extended to include a `.plot()` method that can be used to create simple plots of model output data. See the notebook flopy3_PlotArrayExample.
- Added `.to_shapefile()` method to two- and three-dimensional model input data array classes (`util_2d` and `util_3d`), transient two-dimensional stress period input data classes (`transient_2d`), and model output data classes (`HeadFile`, `UcnFile`, and `CellBudgetFile`) that allows model data to be exported as polygon shapefiles with separate attribute columns for each model layer.
- Added support for ASCII model results files.
- Added support for reading MODPATH version 6 pathline and endpoint output files and plotting MODPATH results using mapping capabilities in `flopy.plot` submodule.
- Added `load()` method for MODFLOW GMG solver.
- Bug fixes:
  - Multiplier in array control record was not being applied to arrays
  - vani parameter was not supported

Version 3.2.1

- FloPy can now be used with **Python 3.x**
- Revised setters for package class variables stored using the `util_2d` or `util_3d` classes.
- Added option to load a subset of MODFLOW packages in a MODFLOW model name file using `load_only=` keyword.

Version 3.1

- FloPy now supports some simple mapping and cross-section capabilities through the `flopy.plot` submodule. See the notebook flopy3_MapExample.
- Full support for all Output Control (OC) options including DDREFERENCE, SAVE IBOUND, and layer lists. All Output Control Input is specified using words. Output Control Input using numeric codes is still available in the ModflowOc88 class. The ModflowOc88 class is currently deprecated and no longer actively maintained.
- Added support for standard MULT package FUNCTION and EXPRESSION functionality are supported. MODFLOW parameters are not supported in `write()` methods.

Version 3.0

FloPy is significantly different from earlier versions of FloPy (previously hosted on googlecode). The main changes are:

- FloPy is fully zero-based. This means that layers, rows and columns start counting at *zero*. The reason for this is consistency. Arrays are zero-based by default in Python, so it was confusing to have a mix.
- Input for packages that take *layer, row, column, data* input (like the wel or ghb package) has changed and is much more flexible now. See the notebook flopy3boundaries
- Input for the MT3DMS Source/Sink Mixing (SSM) Package has been modified to be consistent with the new MODFLOW boundary package input and is more flexible than previous versions of FloPy. See the notebook flopy3ssm
- Support for use of EXTERNAL and OPEN/CLOSE array specifiers has been improved.
- *load()* methods have been developed for all of the standard MODFLOW packages and a few less used packages (*e.g.* SWI2).
- MODFLOW parameter support has been added to the `load()` methods. MULT, PVAL, and ZONE packages are now supported and parameter data are converted to arrays in the `load()` methods. MODFLOW parameters are not supported in `write()` methods.