

## RECENT CHANGES TO COMPACK

### Legend:

- (N): New feature.
- (C): Change to existing feature.
- (F): Bug fix.

### Version 1.1 BETA.

- (C) Added functions `getAtTime` and `getFinal` to the `Solution` class. The base functions `getAtTime` and `getFinal` are deprecated.
- (N) Implemented higher-order reconstruction on 2D rectangular meshes.
- (N) Implementation of source terms. See `Examples.SWBotTop` and `Source.SWBotTop` for an example with the shallow water equations with bottom topography.
- (N) Non-uniform, rectangular meshes have been implemented in the `Mesh.Rect` class. An example can be found in `Example.linAdv_rect`.
- (C) Flux parameters `U1`, `Ur` are now the reconstructed input, while additional parameters `U1A`, `UrA` (A for Average) are added.
- (C) Flux has parameter `dir = {1, 2}` that specifies X- or Y-direction. (`G()` is deprecated).
- (C) `Mesh.cartesianMesh()` is deprecated. Instead, use `Mesh.Cartesian(xLim, nx)`. Boundary condition objects for Cartesian and rectangular grids can now be found in `Mesh.BC`.
- (C) `getVariable()` has been deprecated; use `model.getVariable()` instead.

### Version 1.04 (29/4/2011).

- Changed syntax for Cartesian mesh generation: instead of running `Mesh.cartesianMesh(config, [0,1], 100)` now run `Mesh.Cartesian.CartesianMesh([0,1], 100)`. The `config` object does not need to be initialized when the mesh is generated. `Mesh.cartesianMesh()` is deprecated and will be removed in a later version.
- Implemented figure drawing methods for Chapter 4 and 5.
- Several small fixes.

### Version 1.033 (31/3/2011).

- Fixed incorrect implementation of the Godunov scheme for Burgers' equation.
- Implemented figure drawing methods for Chapter 2.

### Version 1.032 (28/3/2011).

- Bug fixes.

**Version 1.03 (16/2/2011).**

- Moved time update functions from `TimeUpdate` to `NumFlux` (and as a result, `TimeUpdate` has been deleted). Now, schemes that are not in flux form can be implemented by overloading the `timestep` member function of `NumFlux`.
- Implemented more intelligent selection of timesteps. As a result, the program can run much bigger simulations before running out of memory. (The parameter `config.maxNumWrite` and the constant `solution.maxMemUsage` may have to be lowered in special cases.)

**Version 1.02 (17/5/2010).**

- Facilitated second-order reconstruction characteristic variables through the `SlopeLimiterEigen` class. To use this, the model class must implement an `eigenvectors()` function. See the source code for `/+Model/Wave` for an example.
- Implemented the linearized Euler equations and the corresponding Roe flux (also called the Godunov flux).
- The functions that have been used to plot figures in Siddhartha Mishra's lecture notes will now appear in a subdirectory of the `/+Notes/` folder. For now, only the figures of Chapter 7 have been included.
- Added the `tInclude` parameter to `Configuration` objects to guarantee that given time steps will be computed. See the `Configuration` documentation for more details.
- Added the utility function `/+Utility/periodic` for use with periodic boundary conditions.
- Fixed bug where the upwind scheme for scalar linear advection only works for advection speed  $a = 1$ .

**Version 1.01 (8/5/2010).**

- For backwards compatibility, changed the use of the tilde character to suppress function input or output (this is a new feature in MATLAB 2009b).
- Implemented the Euler equations and the Roe flux for the Euler equations.