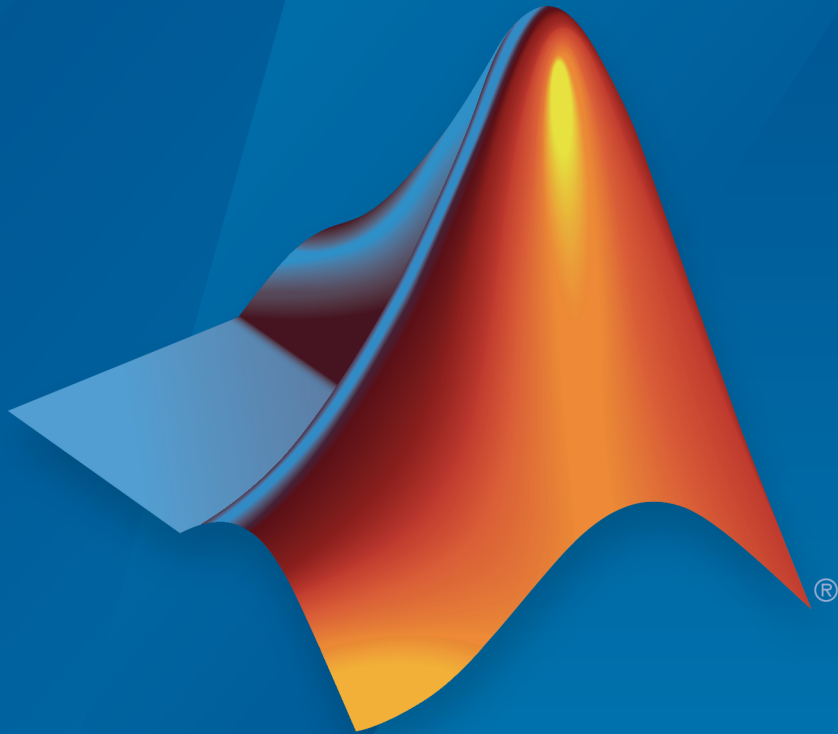


Partial Differential Equation Toolbox™ Release Notes



MATLAB®

How to Contact MathWorks



Latest news: www.mathworks.com
Sales and services: www.mathworks.com/sales_and_services
User community: www.mathworks.com/matlabcentral
Technical support: www.mathworks.com/support/contact_us



Phone: 508-647-7000



The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098

Partial Differential Equation Toolbox™ Release Notes

© COPYRIGHT 2012–2015 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

R2015b

3-D geometry creation from a finite element mesh using the <code>geometryFromMesh</code> function	1-2
Data structure to represent solutions using the <code>createPDEResults</code> function	1-2
Functionality being removed or changed	1-2

R2015a

3-D finite element analysis	2-2
Equation coefficients and boundary conditions for 3-D problems	2-2
Elliptic, parabolic, hyperbolic, nonlinear, eigenvalue solvers for 3-D problems	2-2
3-D geometry import from STL files	2-3
3-D unstructured meshing using tetrahedra	2-3
Plot function to inspect 3-D solutions	2-3
Featured examples with 3-D geometry	2-3
<code>pdebound</code> and <code>pdegeom</code> reference pages removed	2-3

R2014b

Functions for modular definition of boundary conditions . .	3-2
pdeInterpolant object for solution interpolation	3-2

R2014a

Damping option for hyperbolic solver	4-2
---	------------

R2013b

Display option in hyperbolic and parabolic solvers	5-2
Eigenvalue example	5-2

R2013a

Performance and robustness enhancements in meshing algorithm	6-2
New example	6-2

Coefficients of parabolic and hyperbolic PDEs that can be functions of the solution and its gradient	7-2
Graphics export from pdetool	7-2
pdegplot labels edges and subdomains	7-2
New examples	7-2
pdesmech shear strain calculation change	7-3

R2015b

Version: 2.1

New Features

Bug Fixes

Compatibility Considerations

3-D geometry creation from a finite element mesh using the `geometryFromMesh` function

The `geometryFromMesh` function creates 3-D geometry from a finite element mesh, or from a triangulated surface mesh. For details, see the function reference page or “Create and View 3-D Geometry”.

Data structure to represent solutions using the `createPDEResults` function

The `createPDEResults` function converts a PDE solution into a `PDEResults` object. The `PDEResults` object allows you to interpolate the solution using `interpolateSolution`. For details, see the reference pages.

Functionality being removed or changed

Functionality	What Happens When You Use This Functionality?	Use This Instead	Compatibility Considerations
<code>pdesmech</code>	Still runs	PDE app	Use the PDE app instead of <code>pdesmech</code>

R2015a

Version: 2.0

New Features

Bug Fixes

Compatibility Considerations

3-D finite element analysis

You can now solve partial differential equations with 3-D geometry. To do so, there is a new workflow that combines the geometry, mesh, and boundary conditions into a PDEModel object. You can also use this workflow for 2-D geometry. For details, see [Solve Problems Using PDEModel Objects](#).

Equation coefficients and boundary conditions for 3-D problems

To specify problem coefficients or boundary conditions in 3-D geometry, you can use strings with a syntax similar to that of a 2-D problem. There is a new way of writing functions for coefficients in 3-D geometries. For details, see [PDE Coefficients and Boundary Conditions](#).

Compatibility Considerations

To accommodate both 2-D and 3-D geometry, the format of boundary condition objects changed from that introduced in R2014b. The new object is `BoundaryCondition` Properties, and calling `pdeBoundaryConditions` now warns that it will be removed in a future release. If you saved a `pdeBoundaryConditions` object in an R2014b-format MAT file, then loading that file in R2015a can produce an error. Additionally, the syntax for specifying nonconstant boundary conditions has changed. Functions written in the previous syntax continue to work for now.

R2014b Syntax	R2015a Syntax
<code>function bcMatrix = myfun(problem,region,state)</code>	<code>function bcMatrix = myfun(region,state)</code>

For details, see [Changes to Boundary Conditions Object From R2014b](#).

Elliptic, parabolic, hyperbolic, nonlinear, eigenvalue solvers for 3-D problems

The main toolbox solvers now support problems with 3-D geometry. For a listing of functions that do or do not support 3-D geometry, see [Functions That Support 3-D Geometry](#). Solvers take a `model` argument instead of the previous `b`, `p`, `e`, `t` arguments. For details, see the function reference pages.

3-D geometry import from STL files

Import the geometry for a 3-D problem in the STL file format using the `importGeometry` function. For details, see [Create and View 3-D Geometry](#).

3-D unstructured meshing using tetrahedra

Create finite element meshes using the `generateMesh` function. For 3-D geometry, the meshes consist of tetrahedra. See [Mesh Data for \[p,e,t\] Triples: 3-D](#).

Plot function to inspect 3-D solutions

The `pdeplot3D` function plots solutions on the boundaries of 3-D geometry. For details, see [Plot 3-D Solutions](#).

Featured examples with 3-D geometry

There are two new featured examples related to linear elasticity that have 3-D geometry:

- [Deflection Analysis of a Bracket](#)
- [Vibration of a Square Plate](#)

There is also a new example of plotting slices through a 3-D solution: [Contour Slices Through a 3-D Solution](#).

To run the examples at the MATLAB[®] command line:

```
echodemo StrainedBracketExample  
echodemo Eigenvaluesofa3DPlateExample  
echodemo ContourSlices3DExample
```

`pdebound` and `pdegeom` reference pages removed

The `pdebound` and `pdegeom` reference pages have been replaced by the [Boundary Conditions](#) and [2-D Geometry](#) documentation categories.

R2014b

Version: 1.5

New Features

Bug Fixes

Functions for modular definition of boundary conditions

To specify PDE boundary conditions in a modular fashion, per edge or set of edges, use a `pdeBoundaryConditions` specification. For details, see [Steps to Specify a Boundary Conditions Object](#).

`pdeInterpolant` object for solution interpolation

Interpolate a PDE solution to a set of points using `evaluate` on an interpolant. Create the interpolant using `pdeInterpolant`.

R2014a

Version: 1.4

New Features

Bug Fixes

Damping option for hyperbolic solver

You can include damping in the hyperbolic solver in matrix form. There is a new example of dynamics of a damped cantilever beam that shows how to use this feature.

R2013b

Version: 1.3

New Features

Bug Fixes

Display option in hyperbolic and parabolic solvers

You can disable the display of internal ODE solution details that the hyperbolic and parabolic solvers report. To disable the display, set the `Stats` name-value pair to `'off'`.

Eigenvalue example

There is a new example of eigenvalues of a circular membrane. View the example [here](#). To run the example at the MATLAB command line:

```
echodemo eigsExample
```

R2013a

Version: 1.2

New Features

Bug Fixes

Performance and robustness enhancements in meshing algorithm

The meshing (geometry triangulation) functions in `initmesh` and `adaptmesh` provide an enhancement option for increased meshing speed and robustness. Choose the enhanced algorithm by setting the `MesherVersion` name-value pair to `'R2013a'`. The default `MesherVersion` value of `'preR2013a'` gives the same mesh as previous toolbox versions.

The enhancement is available in `pdetool` in the **Mesh > Parameters > Mesher version** menu.

New example

There is a new example of heat distribution in a radioactive rod. View the example [here](#). To run the example at the MATLAB command line:

```
echodemo radioactiveRod
```

R2012b

Version: 1.1

New Features

Compatibility Considerations

Coefficients of parabolic and hyperbolic PDEs that can be functions of the solution and its gradient

You can now solve parabolic and hyperbolic equations whose coefficients depend on the solution u or on the gradient of u . Use the parabolic or hyperbolic commands, or solve the equations using `pdeTool`. For details, see the function reference pages.

Graphics export from `pdeTool`

You can save the current `pdeTool` figure in a variety of image formats. Save the figure using the **File > Export Image** menu. See File Menu.

`pdegplot` labels edges and subdomains

`pdegplot` now optionally labels:

- The edges in the geometry
- The subdomains in the geometry

To obtain these labels, set the `edgeLabels` or `subdomainLabels` name-value pairs to `'on'`. For details, see the `pdegplot` reference page.

New examples

There is a new example of uniform pressure load on a thin plate. View the example here. To run the example at the MATLAB command line:

```
echodemo clampedSquarePlateExample
```

There is a new example of nonlinear heat transfer in a thin plate. View the example here. To run the example at the MATLAB command line:

```
echodemo heatTransferThinPlateExample
```

There is a new example of a system of coupled PDEs. View the example here. To run the example at the MATLAB command line:

```
echodemo deflectionPiezoelectricActuator
```

pdesmech shear strain calculation change

The pdesmech function now calculates shear strain according to the engineering shear strain definition. This has always been the documented behavior. However, the previous calculation was performed according to the tensor shear strain calculation, which gives half the value of the engineering shear strain.

Compatibility Considerations

pdesmech now returns shear strain values exactly twice as large as before.

