

(Free) Software for General Partial Differential Equation Problems in Non-rectangular 2D and 3D Regions

Granville Sewell *

PDE2D is a general-purpose partial differential equation solver which solves very general systems of nonlinear, steady-state, time-dependent and eigenvalue PDEs in 1D intervals, general 2D regions (see Figure 1), and a wide range of simple 3D regions (see Figure 2), with general boundary conditions. It uses a collocation finite element method [2] for 3D problems, and either a collocation or Galerkin finite element method can be used for 1D and 2D problems. It has been sold commercially for 30 years, but recently a version has been made available, which can be downloaded at no cost from www.pde2d.com.

The free version limits the number of unknowns, but the limits are large enough that it can be used to accurately solve fairly large problems. For 1D problems there is a limit of 400 unknowns, for 2D problems the limit is 7500 unknowns, and for 3D problems no more than 25,000 unknowns are allowed. Since high-order finite elements are used, high accuracy can often be obtained for moderately difficult problems with these limits on the number of unknowns.

The two downloadable versions run on Windows with the Intel FORTRAN 11.0 compiler, and on Linux with gfortran 4.3 (a FORTRAN compiler is required with any version). Users who need to solve even larger problems will have to purchase the standard version. There is a video at www.pde2d.com which gives a good overview of the product.

The author began work on PDE2D in 1974 while he was a professor at Universidad Simon Bolivar, and development has continued uninterrupted for 37 years now. Here are some of the major features of PDE2D.

*Mathematics Dept., University of Texas El Paso, USA

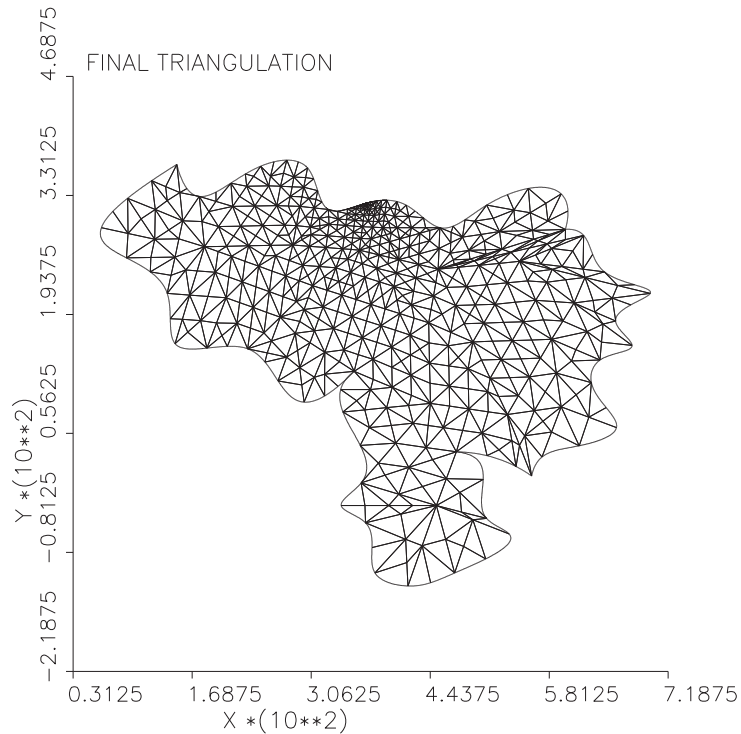


Figure 1. 2D Triangulation of Venezuela

1. There are two user interfaces, a Graphical User Interface (GUI) which can be used to access the collocation finite element methods, and an Interactive Driver, which can be used to access both the collocation and Galerkin finite element algorithms.
2. PDE2D produces its own graphics, but also will automatically produce a MATLAB program, which can be run to produce MATLAB graphics.
3. PDE2D has all the flexibility of FORTRAN: for example, you can write FORTRAN functions to define any PDE and boundary condition coefficients, or write your own FORTRAN post-processing routines.
4. For 2D problems, using the Galerkin option, a user-supplied initial triangulation can be refined adaptively, or graded according to user-supplied specifications (see Figure 1).
5. For 2D problems, using the Galerkin option, curved boundaries can be defined by parametric equations, or a cubic spline can be drawn through user-supplied boundary points.

6. The Galerkin finite element method options use up to 4th degree elements, thus up to $O(h^5)$ accuracy. The collocation options use cubic elements, and thus generally provide $O(h^4)$ accuracy.
7. Newton's method is used to solve the nonlinear algebraic equations, for nonlinear problems.
8. The shifted inverse power method is used to find a single eigenvalue, with associated eigenfunction, of an eigenvalue PDE. If all eigenvalues are desired (without eigenfunctions), a shifted QR iteration from EISPACK is used.
9. Adaptive time step control is available for time-dependent problems.
10. For 2D and 3D problems, there are several options to solve the large linear systems, including iterative methods, sparse direct solvers, and frontal methods. On multiprocessor computers with MPI message passing software, PDE2D also offers parallel linear system solver options.

The best detailed documentation on PDE2D is in Appendix A of reference [1]. Details of how PDE2D solves problems in nonrectangular 3D regions can be found in reference [2]. A list of over 200 journal articles in which PDE2D has been used to generate some or all of the numerical results is given at www.pde2d.com.

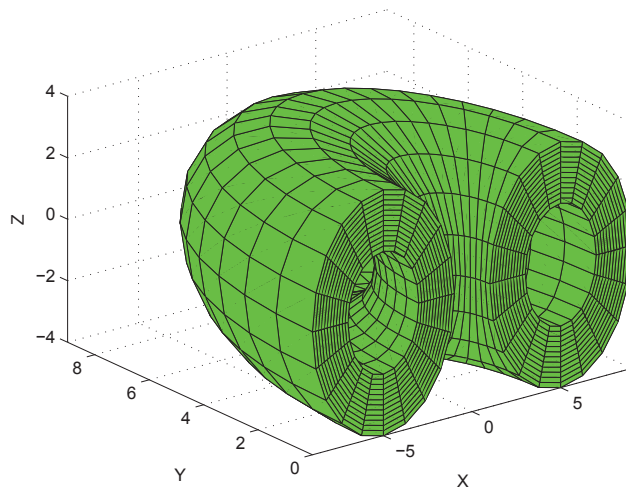


Figure 2. 3D Grid, Part of Torus

References

- [1] Sewel G. *The Numerical Solution of Ordinary and Partial Differential Equations*. John Wiley & Sons, 2005.
- [2] Sewel G. Solving pdes in non-rectangular 3d regions using a collocation finite element method. *Advances in Engineering Software*, 41(5):748–753, 2010.