Taylor & Francis
Taylor & Francis Group

Journal of Discrete Mathematical Sciences and Cryptography

ISSN: 0972-0529 (Print) 2169-0065 (Online) Journal homepage: https://www.tandfonline.com/loi/tdmc20

Partial differential equation calculation and visualization

Li Li

To cite this article: Li Li (2017) Partial differential equation calculation and visualization, Journal of Discrete Mathematical Sciences and Cryptography, 20:1, 217-229, DOI: 10.1080/09720529.2016.1178918

To link to this article: https://doi.org/10.1080/09720529.2016.1178918

Published online: 27 Dec 2016.

Journal of Discrete Mathematical Sciences & Cryptography
Vol. 20 (2017), No. 1, pp. 217–229
DOI : 10.1080/09720529.2016.1178918

Partial differential equation calculation and visualization

Li Li *
College of Mathematics and Computer Science
Yan'an University
Yan'an 716000
China

Abstract

The calculation of partial differential equations has wide applied background. Many of the problems in scientific research and engineering can be the mathematical model of partial differential equations. Differential equation containing multiple independent variables is called as PDE (partial differential equation). As for PDE problem, it is very difficult to solve. Apart from a few special cases, the vast majority of cases are difficult to obtain exact solutions. For the majority of applications workers, from the point of partial differential equations, using the finite element method or finite difference method has to spend a lot of effort to get numerical solution. By MATLAB PDEToolbox, the article introduced the accurate solution process of two-dimensional problem with high-speed. In order to see the relationship between graph, function value and independent variable from the mathematical expression of partial differential equations by MATLAB programming, numerical solution, and the results can be visualized. PDE Toolbox provides a powerful and flexible as well as practical environment for research and problem solving about partial differential equations in two-dimensional space.

Keywords: *Partial differential equations, Difference method, Spectral method, MATLAB*

1. Introduction

PDE mainly means the general variation law based on unknown function and its derivatives to describe the physical quantities of objective world. In theory, the solution of partial differential equations for research has a long history. The initial research focused on physics, mechanics, geometry and other aspects of specific issues; the classic representation

**E-mail:* nwlily2008@163.com

is wave equation, heat conduction equation and potential equation (harmonic equation). Through the study of these issues, an effective method which is still used is formed, for example, separation of variables, fourier transform method. Early research of partial differential equations focused on theory, but in practice, research methods and findings are difficult to be widely used. The main method to solve is: finite difference method [5], the finite element method [3], spectral methods. However, for the majority of applications workers, from the point of partial differential equations, using the finite element method or finite difference method has to spend a lot of effort to get numerical solution, in order to get numerical solution. Now, MATLAB PDEToolbox has achieved high-speed and accurate solution process of the spatial two-dimensional problem [2]. With the physical sciences studied phenomenon extended in both breadth and depth, application scope of partial differential equations is more extensive. From a mathematical point of view, to solve partial differential equations can promote mathematical theory, variational method, series expansions, various aspects of ordinary differential equations, algebra, differential geometry, etc. development. From this perspective, partial differential equations became the center of mathematics. Many of the problems in scientific and engineering can be a mathematical model of partial differential equations. Equations which contain multiple independent variables are called PDE (partial differential equation). PDE problem is very difficult to solve. Apart from a few special cases, the vast majority of cases are difficult to obtain exact solutions.

2. Calculation of Partial Differential Equation

2.1 *Theory of Partial Differential Equation*

If the unknown function which appears in a differential equation contain only one argument, the equation is called ordinary differential equations, also referred to as the differential equation; if a differential equation appears the partial derivative function of a pluralistic differential equations, or if the unknown function and several variables are related, and the unknown function of equation appear the derivative of several variables, then this differential equation is a partial differential equation.

A partial differential equation is usually expressed as follows:

$$A\Phi_{xx} + B\Phi_{xy} + C\Phi_{yy} = f(x, y, \Phi, \Phi_x, \Phi_y)$$

In the formula, *A, B, C*are the constant，called quasilinear number. Typically, there are three kinds of quasi linear equations:

Hyperbolic equation: $B^2 - 4AC > 0$;

Parabolic equation: $B^2 - 4AC = 0$;

Elliptic equation: $B^2 - 4AC < 0$.

2.2 *Partial differential equations solving by difference method*

Difference method, also known as finite difference method or mesh method, is one of the most widely used method in seeking the numerical solution of partial differential equations about definite solution problems.

The basic idea is: first is mesh generation to solve the region, then replace the finite discrete points (grid points) in continuously variable region of independent variables, and replace the function of continuous variable appears in problems with the discrete variables function in the grid points; by replacing derivative with the difference coefficient of function in grid point, then transform partial differential equation solving problem with continuous variables into algebraic equations containing only a finite number of unknowns (referred to as differential format). If differential format has solution, and the grid point becomes smaller and the solution is converges to the solution of the original differential equation problem of definite solution, therefore the solution of difference format is the approximation (numerical solution) of the original problem.

Therefore, using difference method for definite solution problem of the partial differential equations generally needs to address the following issues:

(1) Select the grid;
(2) Choose difference approximation for differential equations and boundary conditions, list difference scheme;
(3) Solve the differential form;
(4) Discuss convergence and error estimate of difference scheme solutions for differential equations.

Here we use a simple example to illustrate the basic concepts of a differential method and general process of differential methods for solving partial differential equations.

Set a first order hyperbolic equation initial value problem.

$$\begin{cases} \dfrac{\partial u}{\partial t} + a\dfrac{\partial u}{\partial x} = 0 \qquad t > 0, \quad -\infty < x < +\infty \\ u(x,0) = \varphi(x) \end{cases} \tag{1}$$

Select Grid:

The difference schematic diagram is shown as Figure 1.

First, for a given area $D = \{(x,t) \big| -\infty < x < +\infty, t \geq 0\}$, proceed meshing solution, the easiest used grid is equidistant linear axis parallel to x axis and t axis respectively.

$$x = x_k = kh,\ t = t_j = j\tau\ (k = 0, \pm 1, \pm 2 \cdots, j = 0, 1, 2, \cdots)$$

Divide D into many small rectangular areas. These straight lines are called grid lines, whose intersection are called mesh points, also referred to as nodes, h and τ are referred to the step length of direction x and t. This grid is called a rectangular grid.

(1) Choose difference approximation for differential equations and definite conditions, list difference scheme. If we use the forward difference quotient represents the first-order partial derivatives, namely:

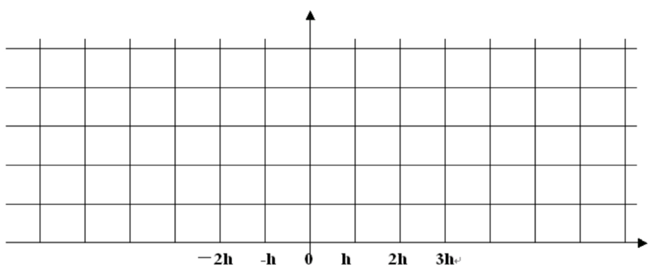

Figure 1

Difference schematic diagram

$$\left.\frac{\partial u}{\partial x}\right|_{(x_k,t_j)} = \frac{u(x_{k+1},t_j)-u(x_k,t_j)}{h} - \frac{h}{2}u''_{x^2}(x_k+\theta_1 h,\ t_j)$$
$$\left.\frac{\partial u}{\partial t}\right|_{(x_k,t_j)} = \frac{u(x_k,t_{j+1})-u(x_k,t_j)}{\tau} - \frac{\tau}{2}u''_{t^2}(x_k,t_j+\theta_2\tau) \quad (2)$$

In which $0<\theta_1,\theta_2<1$.
Equation :

$$\frac{\partial u}{\partial t}+a\frac{\partial u}{\partial x}=0$$

The node (x_k, t_j) can be expressed as:

$$\frac{u(x_k,t_{j+1})-u(x_k,t_j)}{\tau}+a\frac{u(x_{k+1},t_j)-u(x_k,t_j)}{h}$$
$$=\frac{\tau}{2}u''_{t^2}(x_k,t_j+\theta_2\tau)+\frac{ah}{2}u''_{x^2}(x_k+\theta_1 h,t_j)$$
$$=R(x_k,t_j) \qquad (k=0,\pm1,\pm2,\cdots,j=0,1,2,\cdots)$$

Among them: $u(x_k,0)=\varphi(x_k)\,(k=0,\ \pm1,\ \pm2,\cdots)$. Because when h, τ is small enough, the formula will omit $R(x_k, t_j)$, you get a difference equation approximating equation:

$$\frac{u_{k,j+1}-u_{k,j}}{\tau}+a\frac{u_{k+1,j}-u_{k,j}}{h}=0$$

here, $u_{k,j}$ can be seen as an approximation at node (x_k, t_j) when solving problem. With initial conditions:

$$u_{k,0}=\varphi(x_k) \qquad (k=0,\pm1,\pm2,\cdots)$$

Combined, you get a differential format on numerical solution of the problem.

equation: $R(x_k,t_j)=\frac{\tau}{2}u''_{t^2}(x_k,t_j+\theta_2\tau)+\frac{ah}{2}u''_{x^2}(x_k+\theta_1 h,t_j)=O(\tau+h)$

are called truncation error of differential equations. If the truncation error for a differential equation is $R = O(\tau^q + h^p)$, it is called that a differential equation is the order of accuracy q to t, and the order of accuracy p is for x. Obviously, the larger order of truncation error is, the better approximation of differential equations is on differential equations.

If the grid spacing tends to be 0, truncation error of differential equations tends to 0 , then differential equations and corresponding differential equations are compatible. This is a necessary condition for using the finite difference method to solve the problem of partial differential equations.

If the grid spacing tends to be 0, the solution of difference schemes converges to the solution for corresponding problem of definite solution by differential equations, and then such a difference scheme is convergent.

2.3 *Spectral methods for solving partial differential equations*

Spectral method is originated in Ritz-Galerkin method, which is based on orthogonal polynomial (trigonometric polynomial, Chebyshev polynomials, Legendre polynomials, etc.) as the basis functions' Galerkin method, Tau method or configuration method; they were called spectral method, Tau method or the spectral method proposed (collocation method) respectively, commonly known as spectral method. Spectral method is a calculation method based on orthogonal function or intrinsic function as approximated function. From the perspective of function approximation, spectral can be divided into Fourier spectrum method, Chebyshev method or Legendre method. The former applies to a cyclical problem; the latter two methods are applicable to non-cyclical issues. And the basis of these methods is to create space for group functions. Spectral method is a calculation method based on orthogonal function or intrinsic function as approximated function. Approximate spectrum can be divided into two kinds of approximate methods, function approximation and equation approximation. From the perspective of function approximation, spectral spectrum can be divided into Fourier method, Chebyshev method or Legendre method. The former applies to a cyclical problem; the latter two methods are applicable to non-cyclical issues. From the perspective of the approximate equation, spectral methods can be divided into Collocation method in solving discrete physical space and Galerkin method in solving a discrete spectrum space and Pseudo-spectral method in solving the first discrete quadrature in physical space, and then transformed into the spectrum space. Collocation method is suitable for nonlinear problems.

Galerkin method applies to linear problems, and Pseudo-spectral method is for handling nonlinear terms of expanded equations. The characteristics of spectral method are the spectrum precision to smooth exponential function approximation; with fewer grid points to get high accuracy; no phase error; for multiscale volatility issues; calculation accuracy better than other methods. The changes of Fast Fourier proposed greatly promoted the development of spectral method; so far, various spectral methods of calculating format have been presented and were used to calculate astronomy, electromagnetism, geography, and other issues.

When using the spectral method to solving partial differential equations, you often need to use differential truncate function or interpolation function and the following article will discuss this issue.

Assuming $S_u = \sum_{k=-\infty}^{+\infty} ik\hat{u}_k \varphi_k(x)$ is the u's Fourier series, therefore we can obtain

$$P_N u' = (P_N u)',$$

Namely the cut-off and derivation are interchangeable, if $u' \in L^2$, then S_u' also converge to u' under the meaning of L^2.

3. PDE visualization

3.1 *PDE toolbox features*

PDE Toolbox provides a powerful and flexible and practical environment to research and solve the problem of partial differential equations in two-dimensional space. PDE Toolbox features include:

(1) Set the PDE (PDE) definite solution of problem, which set a two-dimensional area of a given solution, boundary conditions and forms and coefficient of equations;
(2) Use finite element method (FEM) to solve the numerical solution of PDE;
(3) Visualization of solutions.

Beginners and advanced researchers will all feel very convenient when using PDE Too1box. As long as the reference PDE problem of definite solution is correct, then after starting MATLAB, typing pdetool in the command line of MATLAB workspace, the system of partial differential equations will generate graphical user interface (Graphical User Interface, Jane immediately referred to as GUI), namely PDE solution graphical environment from Toolbox (PDE Toolbox) and then you can draw on it definite solution area, set the equations and boundary conditions, doing meshing, solving, mapping, etc.

3.2 *PDE toolbox solving equations type*

The basic equations for solving PDE Toolbox have elliptic equations, parabolic equations, hyperbolic equations, eigen value equation, elliptic equations and nonlinear elliptic equations.

Elliptic equation: $-\nabla \cdot (c\nabla u) + au = f, in\,\Omega$

The Ω is bounded plane region; c, a, f and unknown number u are real (or complex) function defined on Ω.

Parabolic Equation: $d\frac{\partial u}{\partial t} - \nabla \cdot (c\nabla u) + au = f, in\,\Omega$

Hyperbolic equation: $d\frac{\partial^2 u}{\partial t^2} - \nabla \cdot (c\nabla u) + au = f, in\,\Omega$

Eigen value equation: $-\nabla (c\nabla u) + au = \lambda du, in\,\Omega$

There d is a complex function defined on Ω, λ is the unknown eigen values. In parabolic equations and hyperbolic equations, the coefficients *c, a, f* and *d* may be dependent on the time *t*.

We can solve nonlinear elliptic equations:

$$-\nabla \cdot (c(u)\nabla u) + a(u) = f(u), in\,\Omega$$

In which *c, a, f* may be a function of the unknown function *u*.

PDE can also solve the following equations;

$$\begin{cases} -\nabla \cdot (c_{11}(u)\nabla u_1) - \nabla \cdot (c_{12}(u)\nabla u_2) + a_{11}u_1 + a_{12}u_2 = f_1 \\ -\nabla \cdot (c_{21}(u)\nabla u_1) - \nabla \cdot (c_{22}(u)\nabla u_2) + a_{21}u_1 + a_{22}u_2 = f_1 \end{cases}$$

You can use the command line to solve high-order equations. For elliptic equations, you can use adaptive mesh algorithm and it also can be combined and used with the nonlinear solution together.

3.3 *Boundary conditions*

(1) Dirichlet conditions: $hu = r$

(2) Neumann conditions: $\vec{n} \cdot (c\nabla u) + qu = g$

There $\vec{n}$ is a unit outside the normal vector on the boundary $\partial\Omega$ of Ω, g, q, h and r is function defined in $\partial\Omega$. For eigen value problems are only limited to second condition: $g = 0$ and $r = 0$. For nonlinear case, factor g, q, h and r depend on u; for parabolic equations and hyperbolic equations, coefficients may depend on the time t.

For the case of equations and boundary conditions are:

(1) Dirichlet conditions:

$$h_{11}u_1 + h_{12}u_2 = r_1$$

$$h_{21}u_1 + h_{22}u_2 = r_2$$

(2) Neumann conditions:

$$\vec{n} \cdot (c_{11}\nabla u_1) + \vec{n} \cdot (c_{12}\nabla u_2) + q_{11}u_1 + q_{12}u_2 = g_1$$

$$\vec{n} \cdot (c_{21}\nabla u_1) + \vec{n} \cdot (c_{22}\nabla u_2) + q_{21}u_1 + q_{22}u_2 = g_2$$

(3) Mixed boundary conditions: $h_{11}u_1 + h_{12}u_2 = r_1$

$$\vec{n} \cdot (c_{11}\nabla u_1) + \vec{n} \cdot (c_{12}\nabla u_2) + q_{11}u_1 + q_{12}u_2 = g_1 + h_{11}\mu$$

$$\vec{n} \cdot (c_{21}\nabla u_1) + \vec{n} \cdot (c_{22}\nabla u_2) + q_{21}u_1 + q_{22}u_2 = g_2 + h_{12}\mu$$

In which calculation of μ should make the Dirichlet condition met. In the finite element method, Dirichlet condition is also called essential boundary conditions; Neumann condition is called natural boundary conditions.

4. Application

Solving Poisson equation on boundary problems on the unit circle:

$$\begin{cases} -\Delta u = 1, \Omega = \left\{ (x, y) \middle| x^2 + y^2 < 1 \right\} \\ u \big| \partial\Omega = 0 \end{cases}$$

Exact solution to this problem is:

$$u(x, y) = \frac{(1 - x^2 - y^2)}{4}$$

If you use a graphical user interface (Graphical User Interface, abbreviated as GUI), the first step of work is typing pdetool in the MATLAB window, press the Enter key OK, and then PDE Toolbox window appears. If you need to coordinate grid, click the Grid option to Options menu. Proceed as follows:

Set boundary conditions (Figure 2); set equation (Figure 3), a solution can be obtained after the decomposition equation with cross-sectional grid (Figure 4).

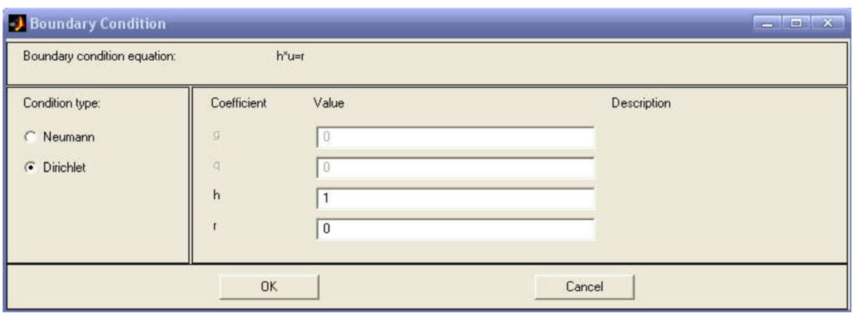

Figure 2

The Condition of Set boundary conditions

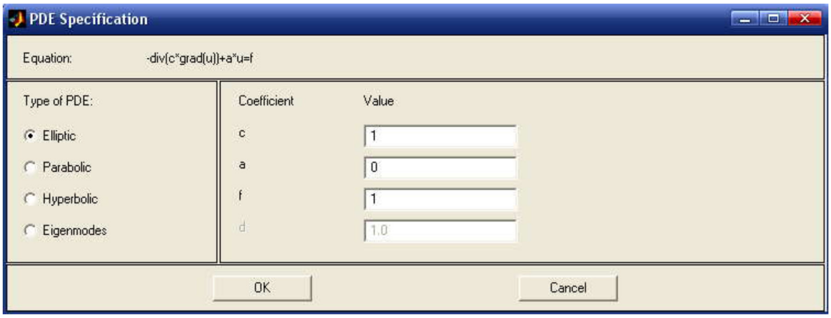

Figure 3

The specification of set equation

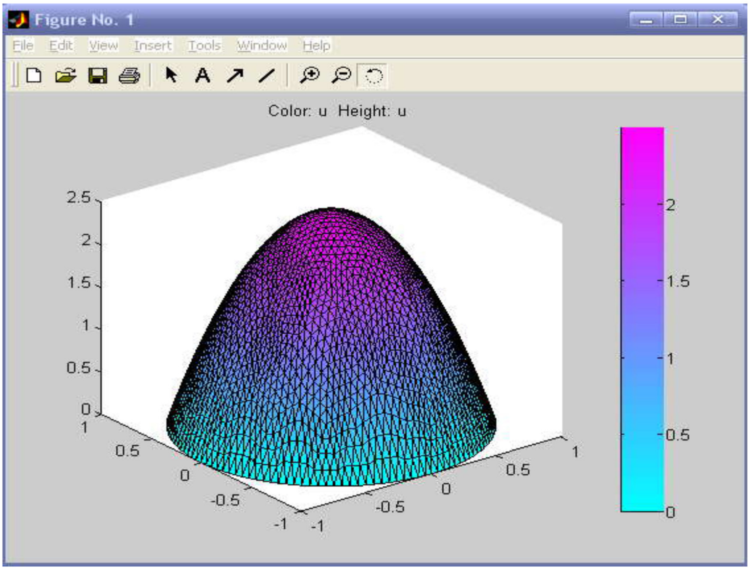

Figure 4

A solution can be obtained after the decomposition equation with cross-sectional grid

Compare with accurate solution. Click the Plot Parameters ... menu option, open Plot Selection dialog, choose user entry in Height (3-D plot) line Property drop-down box, and type u-(1-x.^2-y.^2)/4 in the line of User entry input box, click the plot button to see the absolute error graphics solution, as shown in Figure 5. We can see that at the boundary the error is zero.

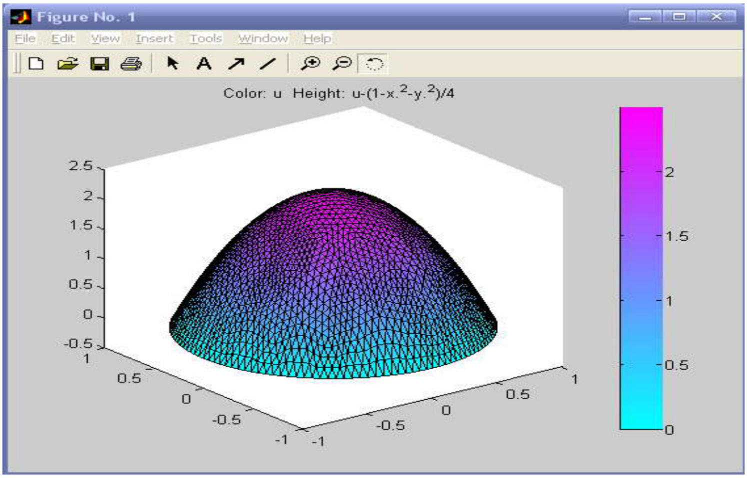

Figure 5

The absolute error graphics solution after click the plot button

5. Conclusion

Solution of partial differential equations has become the core content of scientific and engineering computing, including some large-scale computing and many of them have become routine calculations. As for a high-level technical computing language and interactive environment of algorithm development, data visualization, data analysis and numerical calculation, the MATLAB can solve mathematical problems faster. The PDE is a very practical subject. For the issue of partial differential equations in theoretical research and practical application, due to its complexity and other reasons of proposed boundary and boundary conditions, to seek analytical solution is very difficult or even impossible. It is essential to take advantage of computer research corresponding numerical solution of the problem. Programming for solving the whole process requires a good theoretical basis and programming techniques from numerical partial differential equations, and partial differential equations Toolbox provides a powerful and flexible and practical environment to research and solve partial differential equations in two-dimensional space. By means of this tool, we can learn from cumbersome and common steps for solutions freed to focus on core issues namely description of the problem, definitions and simplifying the determination of the boundary conditions, solving and precision control method selection, greatly improving the computational efficiency.

References

[1] Li Ming. MATLAB Solution on partial differential equations[J]. *Hunan Agricultural Machinery*, 2010, 37(3):89-91.

[2] Feng Guilian. Numerical Solution and Visualization by MATLAB of Partial Differential Equations [J]. *Compter Technology and Development*, 2013, 23(13):120-123.

[3] Feng Guilian. Reality and Visualization of Finite Element Numerical Method in MATLAB [J]. *Software Engineer*, 2015, 18(1):61-64.

[4] Jiang Jun, Zhu Shaoming, Shu Shi, A Software Offers Solution for Partial Differential Equations-UG[J]. *Computer Engineering and Applications*. 2010(3):63-66.

[5] Pan Shulong, Sun Weifu.Multistep Finite Difference Methods for Parabolic Partial Differential Equations[J]. *Journal of Yantai Vocational College*, 2011, 17(1):76-80.

[6] Wang Ziyu, Ren Xingmin.Solution of Partial Differential Equations in Friction Damping Computation[J]. *Mechanical Science and Technology*.2010, 29(12):1669-1672.

[7] Shang Yueqiang. FreeFem + + and finite element computations of partial differential equations [J]. *Journal of Guizhou Normal College*.2011, 27(12):1-5.

[8] Li Wuming, Characteristic method and its some applications in solcing partial differential equations [J].*Journal of Henan Polytechnic University*, 2012, 31(2):235-238.

[9] YueqiangShang, YinnianHe.Parallel iterative finite element algorithms based on full domain partition for the stationary Navier-Stokes equations[J].*Applied Numerical Mathematics*, 2010, 60(7):719-737.

[10] YueqiangShang.New stabilized finite element method for time-dependent incompressible fl-owproblems[J].*International Journal for Numerical Methods in Fluids*, 2010, 62(2):166-187.

Received October, 2015

Revised January, 2016