A Finite Difference Method for Numerical Solution of Goursat Problem of Partial Differential Equation

Pramod Kumar Pandey [1,2]

[1]Department of Mathematics, Dyal Singh College (Univ. of Delhi), New Delhi, India
[2]Present Address: Department of Information Technology, College of Applied Sciences, Ministry of Higher Education, Salalah, Sultanate of Oman
Email: pramod_10p@hotmail.com

Received 30 May 2014; revised 2 July 2014; accepted 17 August 2014

Abstract

In this article, we report the finite difference method for numerically solving the Goursat Problem, using uniform Cartesian grids on the square region. We have considered both linear and nonlinear Goursat problems of partial differential equations for the numerical solution, to ensure the accuracy of the developed method. The results obtained for these numerical examples validate the efficiency, expected order and accuracy of the method.

Keywords

Numerical Method, Finite Difference Method, Goursat Problem, Maximum Absolute Error, Relative Error, Root Mean Square Error, Nonlinear Problem

Subject Areas: **Numerical Mathematics, Ordinary Differential Equation**

1. Introduction

In the present article, we wish to develop a finite difference method for numerically solving Goursat problem [1],

$$\frac{\partial^2 u}{\partial x \partial y} = f\left(x, y, u, u_x, u_y\right)$$

Subject to boundary conditions

$$u(x,0) = \sigma(x) \text{ and } u(0,y) = \gamma(y), \sigma(0) = \gamma(0). \tag{1}$$

How to cite this paper: Pandey, P.K. (2014) A Finite Difference Method for Numerical Solution of Goursat Problem of Partial Differential Equation. *Open Access Library Journal*, **1**: e537. http://dx.doi.org/10.4236/oalib.1100537

There are several methods in the literature for numerical solution of this problem such as Runge-Kutta [1], linear and non-linear trapezoidal [2]-[4] and finite difference method [5] and references there in.

The existence and uniqueness of the solution of the problem (1) is assumed. We have not considered any specific assumption on the source function *f* to ensure existence and uniqueness of the solution [6].

In this article, we develop an algorithm to solve numerically problem (1). The development of the present method is based on an idea discussed and developed [7] for source function $f(x, y, u)$ which is given as,

$$u_{i+1,j+1}-u_{i+1,j-1}-u_{i-1,j+1}+u_{i-1,j-1}=4h^2 f_{i,j}\mathrm{e}^{\left(\frac{h^2\left(f_{xxi,j}+f_{yyi,j}\right)}{6f_{i,j}}\right)}. \tag{2}$$

A novel exponential finite difference method and precisely satisfies the initial conditions. Also this method is an explicit method and can be solved directly at each mesh point.

The present work is organized as follows. In Section 2, we present novel exponential finite difference approximation for the Goursat problem. A novel finite difference method is presented so that the resulting difference equation need satisfies the initial conditions exactly. A derivation of the present method discussed in Section 3 and finally, the application of the developed method presented together with illustrative numerical results has been produced to show the efficiency of the method in Section 4. A discussion and conclusion on the performance of the method are presented in Section 5.

2. The Finite Difference Method

In this section we present proposed finite difference method to numerically solve problem (1). We superimpose on the region of interest a mesh by lines $x_m = mh, y_m = mh, m = 0,1,2,\cdots,N,$ with mesh size $h = 1/N$ in *x* and *y* directions respectively. For convenience of notation the following symbolism is used. We denote the nodal point (x_i, y_j) as (i, j) and value of the source function *f* evaluated at the mesh point (x_i, y_j) by $f_{i,j}$ and similarly we can define other notations in this article. Suppose we have to determine a number $u_{i+1,j+1}$, which is a numerical approximation of the theoretical value of $u(x_i + h, y_j + h)$, a solution of the problem (1) at the mesh point $(x_i + h, y_j + h)$. We propose our difference method to numerically solve problem (1) as,

$$u_{i+1,j+1}-u_{i+1,j-1}-u_{i-1,j+1}+u_{i-1,j-1}=4h^2 \bar{\bar{f}}_{i,j}\mathrm{e}^{\left(\frac{h^2\left(\bar{\bar{f}}_{xxi,j}+\bar{\bar{f}}_{yyi,j}\right)}{6\bar{\bar{f}}_{i,j}}\right)}. \tag{3}$$

Following the notations in [8], we will define the terms in (3) as,

$$\bar{\bar{f}}_{i,j} = f\left(x_i, y_j, u_{i,j}, \bar{\bar{u}}_{xi,j}, \bar{\bar{u}}_{yi,j}\right) \tag{4}$$

Thus using (4), we can define term $\bar{\bar{f}}_{xxi,j} = \left(\frac{\partial^2 \bar{\bar{f}}}{\partial x^2}\right)_{i,j}$ etc. The value of the partial derivative of the solution *u* is not known to us. Thus an approximation to these may be obtained using following difference approximations.

$$\begin{aligned}
\bar{\bar{u}}_{xi,j} &= \frac{u_{i+1,j}-u_{i-1,j}}{2h}, \\
\bar{\bar{u}}_{xi\pm1,j} &= \frac{\pm3u_{i\pm1,j}\mp4u_{i,j}\pm u_{i\mp1,j}}{2h}, \\
\bar{\bar{u}}_{xi,j\pm1} &= \frac{u_{i+1,j\pm1}-u_{i-1,j\pm1}}{2h}, \\
\bar{\bar{u}}_{yi,j} &= \frac{u_{i,j+1}-u_{i,j-1}}{2h}, \\
\bar{\bar{u}}_{yi,j\pm1} &= \frac{\pm3u_{i,j\pm1}\mp4u_{i,j}\pm u_{i,j\mp1}}{2h}, \\
\bar{\bar{u}}_{yi\pm1,j} &= \frac{u_{i\pm1,j+1}-u_{i\pm1,j-1}}{2h}.
\end{aligned} \tag{5}$$

Further we set

$$\bar{\bar{f}}_{xxi,j} = \frac{\bar{\bar{f}}_{i+1,j} - 2\bar{\bar{f}}_{i,j} + \bar{\bar{f}}_{i-1,j}}{h^2},\ \bar{\bar{f}}_{yyi,j} = \frac{\bar{\bar{f}}_{i,j+1} - 2\bar{\bar{f}}_{i,j} + \bar{\bar{f}}_{i,j-1}}{h^2}. \tag{6}$$

Substituting (6) in (3), the difference equation can be written as,

$$u_{i+1,j+1} - u_{i+1,j-1} - u_{i-1,j+1} + u_{i-1,j-1} = 4h^2 \bar{\bar{f}}_{i,j} \mathrm{e}^{\left(\frac{\bar{\bar{f}}_{i+1,j} + \bar{\bar{f}}_{i-1,j} + \bar{\bar{f}}_{i,j+1} + \bar{\bar{f}}_{i,j-1} - 4\bar{\bar{f}}_{i,j}}{6\bar{\bar{f}}_{i,j}}\right)}. \tag{7}$$

In computation of (7) we need the initial values. To compute these initial values we shall define an algorithm similar to that reported in [5] as

$$u_{i,j} = u_{i,j-1} + u_{i-1,j} - u_{i-1,j-1} + \frac{h^2}{4}\left(\bar{f}_{i,j} + \bar{f}_{i,j-1} + \bar{f}_{i-1,j} + \bar{f}_{i-1,j-1}\right). \tag{8}$$

where we have set

$$\bar{f}_{i,j} = f\left(x_i, y_j, u_{i,j}, \bar{u}_{xi,j}, \bar{u}_{yi,j}\right) \tag{9}$$

Similarly we can define $\bar{f}_{i+1,j+1}, \bar{f}_{i+1,j}, \bar{f}_{i,j+1}$ etc. in (9). Also the values of u_x and u_y not known, so we approximate these values using following difference formulas.

$$\begin{aligned}
\bar{u}_{xi,j} &= \frac{u_{i+1,j} - u_{i,j}}{h}, \\
\bar{u}_{xi+1,j} &= \frac{u_{i+1,j} - u_{i,j}}{h}, \\
\bar{u}_{xi,j+1} &= \frac{u_{i+1,j+1} - u_{i,j+1}}{h}, \\
\bar{u}_{xi+1,j+1} &= \frac{u_{i+1,j+1} - u_{i,j+1}}{h}, \\
\bar{u}_{yi,j} &= \frac{u_{i,j+1} - u_{i,j}}{h}, \\
\bar{u}_{yi,j+1} &= \frac{u_{i,j+1} - u_{i,j}}{h}, \\
\bar{u}_{yi+1,j} &= \frac{u_{i+1,j+1} - u_{i+1,j}}{h}, \\
\bar{u}_{yi+1,j+1} &= \frac{u_{i+1,j+1} - u_{i+1,j}}{h}.
\end{aligned} \tag{10}$$

The difference method (7) requires the function value at all the initial mesh points. Thus evaluation started at the initial mesh points using (8). In method (8), we evaluate four functions while in computation of (7), we evaluate five functions. Thus in computation of $u_{i+1,j+1}$ we re-evaluate three functions and evaluate two new functions. Thus this method requires the storage of the three function values at all the mesh points of the square domain. In numerical experiments we can say that the method (7) gives satisfactory and competitive results for the examples considered in this article.

3. Derivation of the Method

In the following expressions, we define $H = \frac{\partial f}{\partial u_x}$ and $G = \frac{\partial f}{\partial u_y}$. From approximation (10)

$$\bar{u}_{xi,j} = \frac{u_{i+1,j} - u_{i,j}}{h} = u_{xi,j} + \frac{h}{2}u_{xxi,j} + O\left(h^2\right).$$

and

$$\overline{u}_{yi,j}=\frac{u_{i,j+1}-u_{i,j}}{h}=u_{yi,j}+\frac{h}{2}u_{yyi,j}+O\left(h^2\right).$$

Thus from definition (9), we have

$$\begin{aligned}\overline{f}_{i,j}&=f\left(x_i,y_j,u_{i,j},u_{xi,j}+\frac{h}{2}u_{xxi,j}+O\left(h^2\right),u_{yi,j}+\frac{h}{2}u_{yyi,j}+O\left(h^2\right)\right)\\&=f\left(x_i,y_j,u_{i,j},u_{xi,j},u_{yi,j}\right)+\left(\frac{h}{2}u_{xxi,j}+O\left(h^2\right)\right)H_{i,j}+\left(\frac{h}{2}u_{yyi,j}+O\left(h^2\right)\right)G_{i,j}\\&=f_{i,j}+\frac{h}{2}\left(u_{xxi,j}H_{i,j}+u_{yi,j}G_{i,j}\right)+O\left(h^2\right).\end{aligned}\tag{11}$$

Thus $\overline{f}_{i,j}$ provides at least an $O(h)$ approximation for the $f_{i,j}$. Similarly we can prove that

$$\overline{f}_{i+1,j}=f_{i+1,j}-\frac{h}{2}\left(u_{xxi,j}H_{i,j}+u_{yi,j}G_{i,j}\right)+O\left(h^2\right).\tag{12}$$

$$\overline{f}_{i,j+1}=f_{i,j+1}+\frac{h}{2}\left(u_{xxi,j}H_{i,j}+u_{yi,j}G_{i,j}\right)+O\left(h^2\right).\tag{13}$$

$$\overline{f}_{i+1,j+1}=f_{i+1,j+1}-\frac{h}{2}\left(u_{xxi,j}H_{i,j}+u_{yi,j}G_{i,j}\right)+O\left(h^2\right).\tag{14}$$

Thus from (11) - (14), we have

$$\overline{f}_{i+1,j+1}+\overline{f}_{i+1,j}+\overline{f}_{i,j+1}+\overline{f}_{i,j}=f_{i+1,j+1}+f_{i+1,j}+f_{i,j+1}+f_{i,j}+O\left(h^2\right).\tag{15}$$

Thus from (15), we define the discretization (8) for the initial values to solve numerically problem (1). Similarly from approximation (5), we have

$$\overline{\overline{f}}_{i,j}=f_{i,j}+O\left(h^2\right).\tag{16}$$

$$\overline{\overline{f}}_{i\pm1,j}=f_{i\pm1,j}+O\left(h^2\right).\tag{17}$$

$$\overline{\overline{f}}_{i,j\pm1}=f_{i,j\pm1}+O\left(h^2\right).\tag{18}$$

Thus from (16) - (18), we have

$$\overline{\overline{f}}_{i+1,j}+\overline{\overline{f}}_{i,j+1}+\overline{\overline{f}}_{i-1,j}+\overline{\overline{f}}_{i,j-1}-4\overline{\overline{f}}_{i,j}=f_{i+1,j}+f_{i,j+1}+f_{i-1,j}+f_{i,j-1}-4f_{i,j}+O\left(h^2\right).\tag{19}$$

Thus from (19), we define the discretization (7) for the problem (1). Thus method (7) is at least $O\left(h^2\right)$ accurate. Though methods (7) and (8) are explicit methods which compute approximate value of $u\left(x_i+h,u_j+h\right)$ but if problem is nonlinear then it may be implicit method. Thus if differential Equation (1) is nonlinear, (7) and (8) can be solved by the Newton-Raphson iterative method.

4. Numerical Experiment

To illustrate our method and demonstrate its computational efficiency, we will consider the examples discussed in [1] [2], in which the errors taken to be the root mean square *RMU*, maximum absolute error *MAU* and maximum relative error *RAU i.e.*

$$RMU=\sqrt{\frac{\sum_{i=2}^{N}\left(u\left(x_i,y_j\right)-u_{i,j}\right)^2}{\left(N-1\right)^2}},\quad j=2,4,\cdots,N.$$

$$MAU=\max_{2\le i,j\le N}\left|u\left(x_i,y_j\right)-u_{i,j}\right|$$

and

$$RAU = \max_{2\le i,j\le N}\left(\frac{u\left(x_i,y_j\right)-u_{i,j}}{u\left(x_i,y_j\right)}\right)$$

We have used the Newton-Raphson iteration method to compute the values (7) and (8) in all above examples. All computations in the experiment were performed on MS Window 2007 professional operating system in the GNU FORTRAN environment version-99 compiler (2.95 of gcc) running on Intel Duo core 2.20 GHz. PC. The solutions are computed on $(N-1)^2$ nodes, in computation of initial values and value of solution at advance mesh points, iterations continued until either maximum difference between two iterates is less than 10^{-9} or number of iterations reached 10^3.

Problem 1.

Consider a nonlinear problem discussed in [1] which, when solving consists of

$$u_{xy} = \frac{u_x u_y}{u}$$

in the region $[1,2]\times[1,2]$ with the boundary conditions $u(x,1)=\sin(1)\mathrm{e}^{(x+1)}$, $u(1,y)=\sin(y)\mathrm{e}^{(y+1)}$, for which the analytical solution is found to be $u(x,y)=\sin(y)\mathrm{e}^{(y+x)}$. We have computed the solution and presented *MAU*, *RAU* and *RMU* for the developed method for different values of *N* in **Table 1**.

Problem 2.

Consider a non linear problem discussed in [2] which, when solving consists of

$$u_{xy} = \frac{u_x u_y}{u} + u$$

in the region $[0,1]\times[0,1]$ with the boundary conditions $u(x,0)=\mathrm{e}^{x^2}$, $u(0,y)=\cos(y)$, for which the analytical solution is found to be $u(x,y)=\mathrm{e}^{x(x+y)}\cos(y)$. We have computed the solution and presented *MAU*, *RAU* and *RMU* for the developed method for different values of *N* in **Table 2**.

Problem 3.

Consider a linear problem discussed in [1] which, when solving consists of

$$u_{xy} = \frac{u_x + u_y + u}{3.0}$$

in the region $[0,1]\times[0,1]$ with the boundary conditions $u(x,0)=\mathrm{e}^{x}$, $u(0,y)=\mathrm{e}^{y}$, for which the analytical solution is found to be $u(x,y)=\mathrm{e}^{x+y}$. We have computed the solution and presented *MAU*, *RAU* and *RMU* for the developed method for different values of *N* in **Table 3**.

Table 1. Maximum absolute and relative error, root mean square error in $u(x,y)=\sin(y)\mathrm{e}^{(y+x)}$.

N	Error		
	MAU	*RAU*	*RMU*
4	0.84312596(−5)	0.23413723(−6)	0.47533363(−5)
8	0.70326987(−5)	0.91173128(−7)	0.39888882(−5)

Table 2. Maximum absolute and relative error, root mean square error in $u(x,y)=\mathrm{e}^{x(x+y)}\cos(y)$.

Error	*N*				
	4	8	16	32	64
MAU	0.57418332(−1)	0.25210236(−2)	0.19413770(−3)	0.56662439(−5)	0.54690790(−5)
RAU	0.10430584(−3)	0.47750591(−3)	0.15391412(−4)	0.15785952(−5)	0.96294229(−6)
RMU	0.20805337(−1)	0.48529863(−3)	0.18880772(−4)	0.81907029(−6)	0.96616520(−6)

Table 3. Maximum absolute and relative error, root mean square error in $u(x,y)=\mathrm{e}^{x+y}$.

Error	N				
	4	8	16	32	64
MAU	0.56023598(−2)	0.86784363(−3)	0.12207031(−3)	0.21457672(−4)	0.57220459(−5)
RAU	exact	exact	exact	exact	0.10807225(−5)
RMU	0.35539242(−2)	0.36674476(−3)	0.41264524(−4)	0.52586661(−5)	0.17699658(−5)

5. Conclusions

In general, each numerical method has its own merit and demerit in its application. The present method is therefore good for use under the initial conditions. The demerit of this method is in computation of initial values which highly affect the all subsequent computational results. The present method which is at least second order accurate seems competitive with other finite difference methods. Method is computationally efficient which can be observed in numerical results obtained in our experiments. It is observed from the results that method has higher accuracy *i.e*. small discretization error.

In the present article a different approach, we have presented at least second order finite accurate difference method for the numerical solution of the nonlinear Goursat problem. The development of this method will lead to a possibility to further raise the order and accuracy of the method. Work in this direction is in progress and soon it will appear.

References

[1] Day, J.T. (1966) Runge-Kutta Method for the Numerical Solution of the Goursat Problem in Hyperbolic Partial Differential Equations. *Computer Journal*, **9**, 81-83. http://dx.doi.org/10.1093/comjnl/9.1.81

[2] Jain, M.K. and Sharma, K.D. (1968) Cubature Method for the Numerical Solution of the Characteristic Initial Value Problems, $u_{xy}=f(x,y,u,u_x,u_y)$. *Journal of the Australian Mathematical Society*, **8**, 355-368. http://dx.doi.org/10.1017/S1446788700005425

[3] Evans, D.J. and Sanugi, B.B. (1988) Numerical Solution of the Goursat Problem by a Nonlinear Trapezoidal Formula. *Applied Mathematics Letters*, **1**, 221-223. http://dx.doi.org/10.1016/0893-9659(88)90080-8

[4] Stetter, H.J. and Torning, W. (1963) General Multistep Finite Difference Method for the Solution of $u_{xy}=f(x,y,u,u_x,u_y)$. *Rendiconti del Circolo Matematico di Palermo*, **12**, 281-298. http://dx.doi.org/10.1007/BF02851264

[5] Nasir, M.A.S. and Ismail, A.I.M. (2005) A New Finite Difference Scheme Based on Heronian Mean Averaging for the Goursat Problem. *ISAME Transactions*, **2**, 98-103.

[6] Jeffrey, A. and Taniuti T. (1964) Nonlinear Wave Propagation. Academic Press, New York.

[7] Pandey, P.K. (2014) A Fourth Order Finite Difference Method for the Goursat Problem. *Acta Technica Jaurinensis*, **7**, 319-327.

[8] Chawla, M.M. (1979) A Sixth Order Tridiagonal Finite Difference Method for General Nonlinear Two Point Boundary Value Problems. *Journal of the Institute of Mathematics and its Applications*, **24**, 35-42. http://dx.doi.org/10.1093/imamat/24.1.35