Solution of the Developed Homogenous &Non-Homogenous Wave Equation Using Mathematica

IOP Conf. Series: Materials Science and Engineering **571** (2019) 012001 doi:10.1088/1757-899X/571/1/012001

Solution of the developed homogenous &non-homogenous Wave equation using MATHEMATICA

Ahmed Hadi Hussain[a*], Mohamad Abd Al-daem [b] , Saba Nazar Al-Khfaji [c] , [d] Sameer Annon Abbas
[a,b] University of Babylon/College of Engineering Al-Musayab/Department of Energy
[c] University of Kufa/ College of Science /Department of Mathematics
[d] Iraqi Ministry of Education
ahmedhadi99@yahoo.com (corresponding author)[a*]

Abstract: The purpose of this research is to develop the wave equation by replacing the constant with a quadratic function based on the two variables (x, t). Some elementary and boundary conditions using MATHEMATICA.

1. Introduction

A highly accurate numerical method is drove for the solution of one-dimensional wave equation with Neumann boundary conditions. A multi-dimensional fractional wave equation that describes propagation of damped waves is introduced and analyzed this feature is a decisive factor for inheriting some crucial characteristics of wave equation, such as a constant phase velocity of the damped waves which is now described by the fractional wave equation in [1] and [2]. In [3] , the wave equation with variable wave speed on nonconforming domains with fourth order accuracy in both space and time. This accomplished using an implicit finite difference scheme for the wave equation and solving an elliptic (modified Helmholtz) equation at each time step with fourth order spatial accuracy by the method of difference potentials. In [4] , The present a numerical algorithm for the linear one-dimensional heat and wave equation .In this method ,a finite difference approach had been used to discrete the time derivative while quantic spline is applied as an interpolation function in the space dimension. The numerical methods are of great importance for approximating the solutions of nonlinear ordinary or partial differential equations, especially when the nonlinear differential equation under consideration faces difficulties in obtaining its exact solution. In this latter case, we usually resort to one of the efficient numerical methods In [5]. An initial-boundary value problem for fractional in time diffusion –wave equation is considered. A priori estimates in Sobolev spaces are derived in [6] . The fractional diffusion equation is solved for different boundary value problems, these being absorbing and reflecting boundaries in half-space and in a box. Thereby , the method of images and Fourier –Laplace transformation technique are employed in [7]. The illustration with numerical experiments the behavior of certain algorithms based on exact regularization. First ,we consider an elliptic PDE with a nonlinear discontinuity in [8].The first-order (or linear) Rytov or Born approximation is the

foundation for formulation of wave – equation tomography and waveform inversion, so the validity of the Rytov/ Born approximation can substantially affect the applicability of these theories in [9]. In [10] The two-dimensional nonlinear wave equations are considered .Solution to the problem is approximated by using optimal homotopy method .We can develop this work through the study of linear partial differential equations by using the periodic conditions and we can find the uniqueness solution. Also we can graph all the problems by using the Mathematica program.

2-Setting the problem

We consider the following for the system partial differential equations:

$$\frac{\partial^2 u}{\partial t^2} - k^2(x,t)\frac{\partial^2 u}{\partial x^2} = f(x,t),\ 0 < x < a, t > 0$$
$$u(x,0) = f(x),$$
$$\left.\frac{\partial u(x,t)}{\partial t}\right|_{t=0} = g(x),$$
$$u(0,t) = u(a,t) = 0. \qquad (2.1)$$

Where $u(x,t)$ is the unknown continuous function and $k^2(x,t), f(x,t)$ are an arbitrary continuous function dependent with respect variables x and t. The initial boundary conditions are held in equation (2.1).

NDSolve is able to solve some partial differential equations directly when you specify more independent variables. NDSolve currently uses the numerical method of lines to compute solutions to partial differential equations.

The method is restricted to problems that can be posed with an initial condition in at least one independent variable. For example, the method cannot solve elliptic partial differential equations such as Laplace's equation. NDSolve uses the method of lines and is typically suitable for solving problems of a hyperbolic equation in partial differential equations, NDSolve computes the solution for the partial differential equation the results is a two-dimension Interpolating Function. Now to solve the system (2.1) numerically, we use typical commands:

sol= u[x, t]/. NDsolve[eqns, u[x, t], {x, a, b}, {t, c, d}}[1]solve the Problem

Plot3D[sol, {x, a, b}, {t, c, d}]Plot the solution

Next, we show several examples of using NDSolve.

3 .Study some application examples for equation (2.1)

3.1. Let consider the problem (2.1), with

$\frac{\partial^2 u}{\partial t^2} - t^2 \frac{\partial^2 u}{\partial x^2} = \sqrt{x}$, *when* t^2 *is a quadratic function*

$u(x,0) = 10x^2(1-x)^2,$

$\left.\frac{\partial u(x,t)}{\partial t}\right|_{t=0} = 0,$

$u(0,t) = u(1,t) = 0.$

Solution: using Mathematica code we obtain:

In Put In Mathematica: $eqns = \{\partial_{t,t} u[x,t] - t^2 \partial_{x,x} u[x,t] == \sqrt{x}, u[x,0] == 10x^2(1-x)^2,$

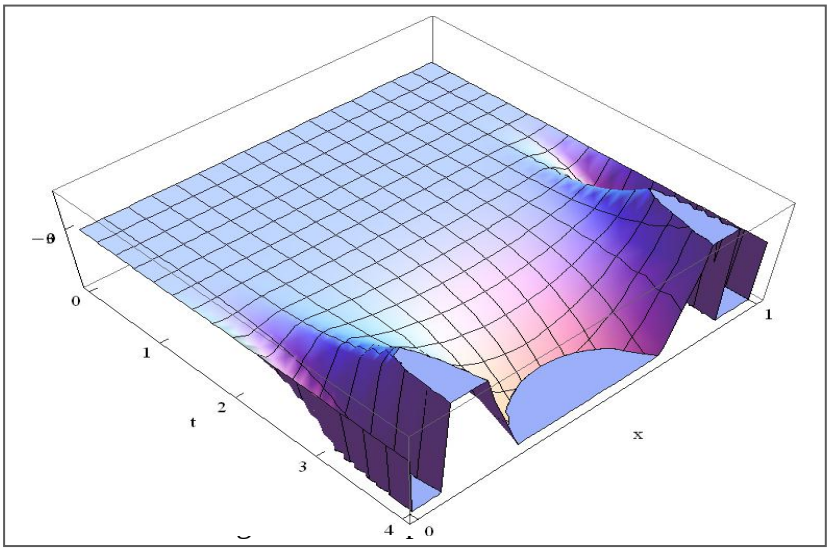
$Derivative[0,1][u][x,0] == 0, u[0,t] == 0, u[1,t] == 0\};$

$sol = u[x,t] /. NDSolve[eqns, u[x,t], \{x,0,0.6\}, \{t,0,0.4\}, PrecisionGoal \to 1][1]$

Results after implementation of the program are:

Out Put Mathematica: $InterpolatingFunction[\{\{0.,1.\},\{0.,0.4\}\},<>][x,t]$

Solution is generated by this command as it is shown in figure (1)



3.2. Let's consider the problem (2.1), with

$\frac{\partial^2 u}{\partial t^2} - \left(\cos^2(t) + \sin^2(x)\right)\frac{\partial^2 u}{\partial x^2} = \sin(t) + x,$

When $\cos^2(t) + \sin^2(x)$ *is a trigonometric function dependent on two* var *iables* x *and* t

$u(x,0) = 2x^2(1-x)^4,$

$\left.\frac{\partial u(x,t)}{\partial t}\right|_{t=0} = 0,$

$u(0,t) = u(1,t) = 0.$

Solution: the numerical solution of this problem is:

$$In\ Put\ In\ Mathematica: eqns = \{\partial_{t,t}u[x,t] - (\cos[t]^2 + \sin[x]^2)\partial_{x,x}u[x,t] == \sin[t] + x, u[x,0] == 2x^2(1-x)^4,$$

$$Derivative[0,1][u][x,0] == 0, u[0,t] == 0, u[1,t] == 0\};$$

$$sol = u[x,t]/.NDSolve[eqns, u[x,t], \{x,0,1\}, \{t,0,5\}, PrecisionGoal \to 1][1]$$

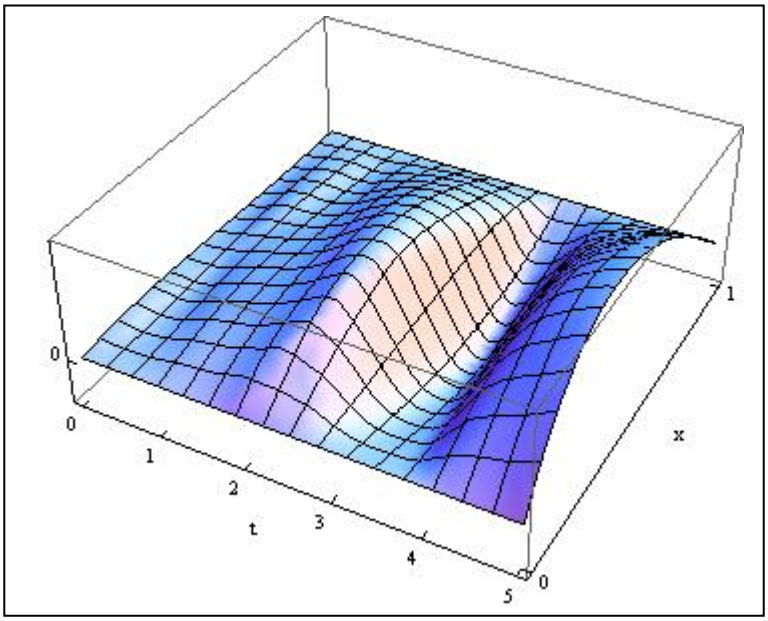
Results after implementation of the program are:

$$Out\ Put\ Mathematica: InterpolatingFunction[\{\{0.,1.\},\{0.,5.\}\},<>][x,t]$$

Then, we can plot the numerical solution of the problem (4.3.3) by using Plot3DCommand,weget

$$Plot3D[sol, \{t,0,5\}, \{x,0,1\}, AxesLabel \to \{"t","x",""\}, Ticks \to \{\{0,1,2,3,4,5\},\{0,1\},\{-2,0\}\}]$$

The solution is generated by this command as it is shown in figure (2)



4. Numerical Solution of Linear Hyperbolic Equation

Let's consider the predefined functions in both systems with the aid of which we can obtain the approximate numerical solutions solving various linear partial differential equations problem. In Mathematica, various initial boundary value problems can be solved numerically with the aid function NDSolve (with various options). This can be done by the Mathematica it is possible to solve numerically only evolution equations using NDSolve. Additionally, it is possible to specify explicit the method of lines for solving partial differential equations.

We study the linear second order partial differential equations define for the following system:

$$\frac{\partial^2 u}{\partial t^2} - k^2(x,t)\frac{\partial^2 u}{\partial x^2} = f(x,t), \qquad (4.1)$$

$$\frac{\partial^2 u}{\partial t^2} - k^2(x,t)\frac{\partial^2 u}{\partial x^2} = g_i(x,t) \quad . \tag{4.2}$$

Where $f(x,y)$ and $g_i(x,t)$, $i = 1,2$, are arbitrary functions, then equations (4.1) and (4.2) have the same initial and boundary conditions

$$u(x,0) = 0,$$

$$\left.\frac{\partial u(x,t)}{\partial t}\right|_{t=0} = f(x\pi),$$

There is no need for 8 boundary conditions because the equations are programmatically solved. In Mathematica there are constant steps that cannot be added.

$$u(0,t) = u(L,t) = 0.$$

It can solve the problems (4.1) and (4.2) numerically by using Mathematica. In reality, the system (4.1) and (4.2) are quite easy to solve with the method of lines.

5. Study some application examples for equation (4.1) and (4.2)

-Let's consider the problem (4.1) and (4.2) with

$$\frac{\partial^2 u}{\partial t^2} - x^2\frac{\partial^2 u}{\partial x^2} = x, \tag{5.1}$$

$$\frac{\partial^2 u}{\partial t^2} - x^2\frac{\partial^2 u}{\partial x^2} = x + \cos(x), \text{When } x^2 \text{a quadratic function} \tag{5.2}$$

$$u(x,0) = 0,$$

$$\left.\frac{\partial u(x,t)}{\partial t}\right|_{t=0} = \sin(4\pi x),$$

$$u(0,t) = u(L,t) = 0.$$

Solution: The problem (5.1) can be solved numerically by using Mathematica where L=0.5 and T=1.5

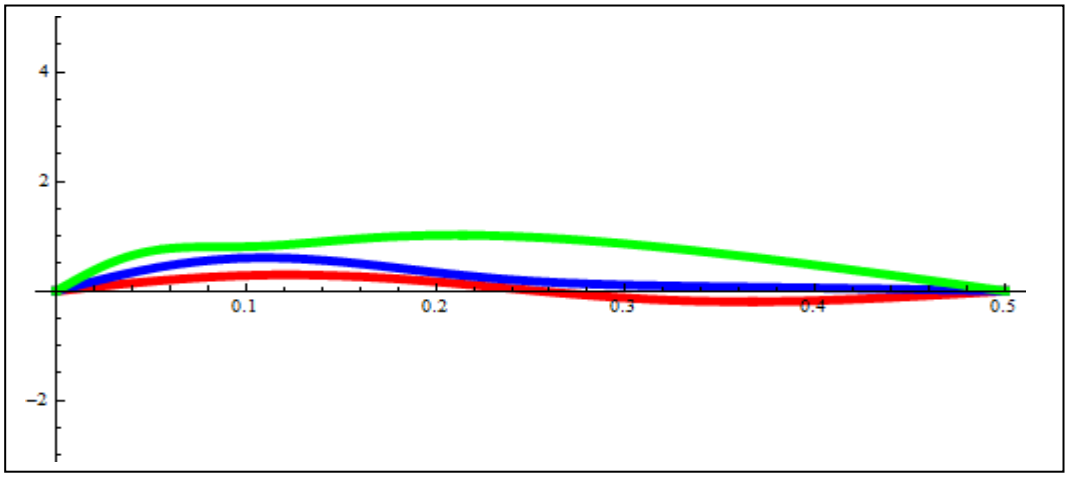
```
In Put Mathematica :<< DifferentialEquations 'InterpolationAnatomy '
SetOptions[Plot, ImageSize → 500, PlotRange → {All, {−5,5}}, PlotPo ints → nP * 2,
PlotStyle → {Blue, Thickness[0.01]}};
SetOptions[Plot 3D, Im ageSize → 500, PlotRange → All ];
{1 = 0.5, tF = 1.5, s = 1/100, nP = 100, nN = 3,11 = {Red, Blue, Green},12 = {0.3, 0.7,1.5}}
pde[1] = D[u[x, t], {t, 2}] − x^2 * D[u[x, t], {x, 2}] == x
pde[2] = D[u[x, t], {t, 2}] − x^2 * D[u[x, t], {x, 2}] − Cos[x]^2 == x
bc = {u[0, t] == 0, u[1, t] == 0}
Do[sol[ j ] = NDSolve[Flatten[{pde[ j ], ic, bc}], u, {x, 0,1}, {t, 0, tF},
MaxStepSize → s, Pr ecisionGoal → 2];
f [ j ] = u / .First[sol[ j ]], { j ,1, 2}];
Map[Length, InterpolatingFunctionCoordinates[f [2]]]
Do[g[ j , i ]Plot[Evalute[u[x ,12[[i ]]] / .sol[ j ]], {x , 0,1},
PlotStyle → {11[[i ]], Thicness[0.01]}], { j ,1, 2}, {i ,1, nN }];
GraphicsRow [{Show [Table[g[1, i ], {i ,1, nN }]], Show [Table[g[2, i ], {i ,1, nN }]]}]
ic = {u[x , 0] == 0, (D[u[x , t ], t ] / t → 0) == Sin[4 * Pi * x ]}
```

Out put Mathematica :

$$\{0.5,1.5,1/100,100,3,\{RGBColor[1,00],RGBColor[0,0,1],RGBColor[0,1,0]\},\{0.3,0.7,1.5\}\}$$

$$\{51,176\}$$

Then it can plot numerical solution of the initial boundary value problems(5.1) and (5.1) as shown in figure (3)



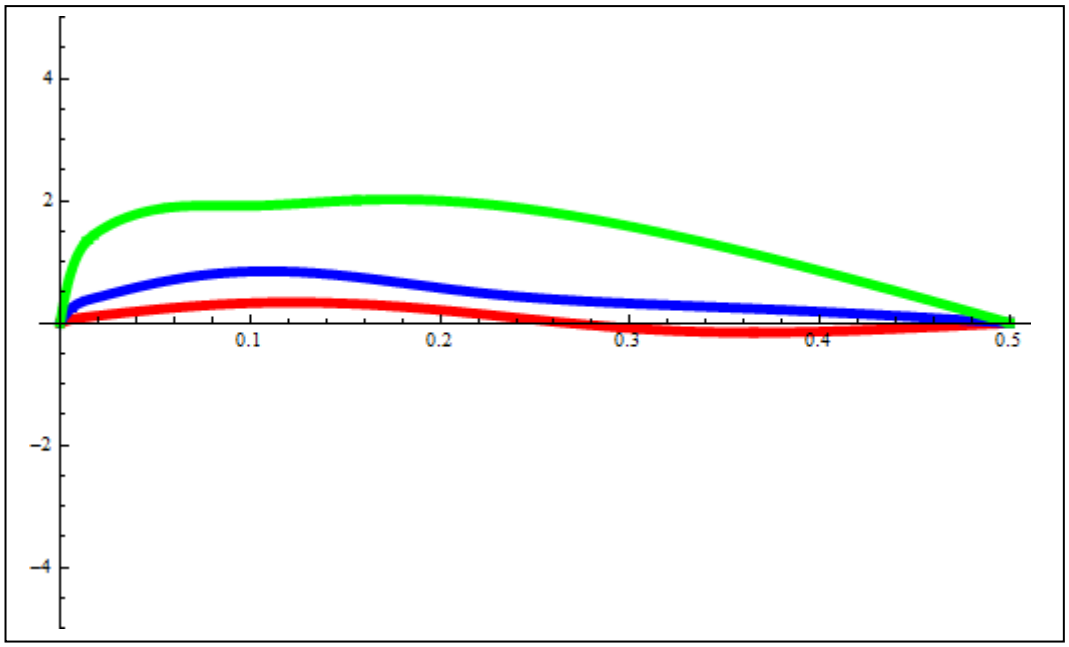


Figure 4: Plots of numerical solutions of the initial boundary value problems for equation (5.1) (upper plot) and equation (5.1) (lower plot) at different times t_k= 0.3 (Red), 0.7 (Blue), 1.5 (Green)

A 3D graphical solution of the problem (5.1) is Plotted using Mathematica Programm as follows:

In Put Mathematica :
$Do[numVals[i] = Evalute[u[0.1, 1/2]/.sol[i]]; \Pr int[numVals[i]];$
$g3D[i] = Plot3D[Evalute[u[x, t]/.sol[i]], \{x, 0, 1\}, \{t, 0, tF\},$
$ColorFunction \to Function[\{x, y\}, Hue[x]],$
$BoxRatios \to 1, ViewPo\,int \to \{-1, 2, 1\}, Im\,ageSize \to 500], \{i, 1, 2\}];$
$GraphicsRow[\{g3D[1], g3D[2]\}]$

Out Put Mathematica :
{0.456055}
{0.579758}

A surface plot of the solution of equations (5.1) and (5.1) is shown in figure(4.5)

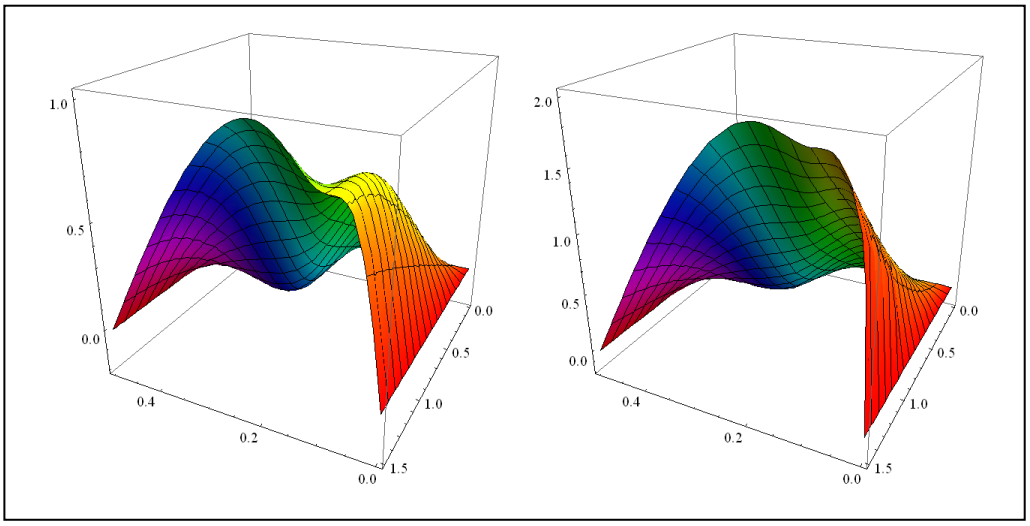


Figure 5: 3D Plots of numerical solutions of the initial boundary value problems for the equation(5.1) on the left and equation(5.1) on the right

6. Discussion of result

At the beginning it can consider the construction of numerical and graphical solutions of various initial boundary value problems. That can be by using predefined functions and embedded methods in system (Mathematica), for the second-order partial differential equations. In some cases, the numerical solutions are compared with the corresponding analytical solutions and obtain the corresponding error function. In addition to that, it can explain how to find numerical and graphical solutions, by specifying a particular numerical method and numerical boundary conditions. In Mathematica various initial boundary value problems can be solved numerically with the aid of the function NDSolve (with various option). While in Maple with the aid of predefined function pdsolve (with option numeric). It can solve numerically initial boundary value problems for a single partial differential equation (of higher order). As well as the partial differential equation systems can solve by using the embedded methods or by specifying a particular method for solving a single partial differential equation. It is possible to impose Dirichlet, Neumann, Robin , or periodic boundary conditions.

7-Conclusions

It can prove the numerical solution method for linear partial differential equations in the Mathematica function NDSolve is presented. Mathematica's NDSolve command includes a general solver for partial differential equations based on the method of lines. In Mathematica with the aid of the predefined function NDSolve, it is possible to obtain approximate numerical solutions of various linear and nonlinear partial differential equations problems (initial boundary value problems). This can be done by the Mathematica system by applying the method of lines. It is noted that in Mathematica (Ver $\leq$ 8), it is possible to solve numerically only evolution equations NDSolve. Additionally, it is possible to specify explicitly the method of lines for solving partial differential equations and the proper suboptions for the method of lines.

Mathematica:

```
NDSolve[{pde,ic,bc}, depVars, indVars, ops]
NDSolve[{pde,ic,bc},  u, {x,x1,x2}, {t,t1,t2},...]
NDSolve[{pde,ic,bc}, {u1,...,un}, {x,x1,x2}, {t,t1,t2},...]
NDSolve[{pde,ic,bc},u,{x,x1,x2},{t,t1,t2},Method->{m,subOps}]
Options[NDSolve`MethodOfLines]
```

NDSolve finding numerical solutions to partial differential equations problems (initial boundary value problems), where depVars and indVars are the dependent and independent variables respectively.

NDSolve, Method, finding numerical solutions to partial differential equations problems by method of lines with some specific sub options (sub Ops).

The option Method and the most important sub options as follows:

```
Method->{"MethodOfLines","SpatialDiscretization"->{
 "TensorProductGrid","MinPoints"->val,"MaxPoints"->val,
 "MaxStepSize"->val,"PrecisionGoal"->val,"DifferenceOrder"->val}}
```

References

[1] A.H. Choudhury, R.K.Deka,(2011)"Wavelet Method for Numerical Solution of Wave Equation With Neumann Boundary Conditions"

[2] Yuri Luchko, "Multi-dimensional fractional wave equation and some properties of its fundamental solution"

[3] S. Britt , S. Tsynkov , E. Turkel, (2018)" Numerical solution of the wave equation with variable wave speed on nonconforming domains by high-order difference potentials"

[4] J. Rashidinia, M. Mohsenyzadeh,(2015)"Numerical Solution of One-Dimensional Heat and Wave Equation by Non-Polynomial Quintic Spline"

[5] N. Y. A. Elazem , A. Ebaid ,(2013)"Numerical study of some nonlinear wave equations via Chebyshev collocation method"

[6] Aleksandra Delic, (2016)"Fractional in Time Diffusion-Wave Equation and its Numerical Approximation"

[7] Ralf Metzler, Joseph Klafter,(2000)"Boundary value problems for fractional diffusion equations"

[8] R. Echevarria ,(1995)"The Numerical Solution of Some Elliptic Problems with Nonlinear Discontinuities Using Exact Regularization"

[9] WenjunXu, Xiao-Bi Xie and JianhuaGeng,(2015)"Validity of the Rytov Approximation in the Form of Finite-Frequency Sensitivity Kernals"

[10] H. Ullah , S.Islam, L. C. C. Dennis, T. N. Abdelhameed, I. Khan and M.Fiza (2015)"Approximate Solution of Two-Dimensional Nonlinear Wave Equation by Optimal Homotopy Asymptotic Method"