PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

UNIVERSITY MOHAMED BOUDIAF- M'SILA

Faculty of Mathematics and computer science

Department of Mathematics

Master thesis

Domaine: Mathematics and informatics.
Field : Mathematics.
Option : mathematical and numerical analysis

Theme

Radial Basis Function approximations for partial differential equations.

Presented by :
BRIK OSEMA

Publicly supported: 11 June 2024

Board of Examiners:

GASMI Abdelkader	**Prof,**	**Université of M'sila**	**President**
GAGUI Bachir	**M.C.A,**	**Université of M'sila**	**Supervisor**
BLIZAK Tahar	**M.C.B,**	**Université of M'sila**	**Examinator**

Academic year : 2023/2024.

Acknowledgement

In the Name of Allah, the Most Gracious, the Most Merciful.Praise and gratitude be to Allah, the Lord of the Worlds, for granting me success in completing my master's thesis and facilitating the difficulties for me. Peace and blessings be upon the most noble of Prophets and Messengers, our master Muhammad, and upon his family and companions.I extend my sincere gratitude and appreciation to my supervisor, Professor Bachir Gagui, for his patience with me and for accompanying me throughout the completion of this work, for the academic and administrative support he provided during the preparation of this thesis, for the great effort and valuable advice he offered, and for the scientific guidance that enriched this work. I am truly grateful for his patience and follow-up.I also express my heartfelt thanks to the distinguished defense committee: Professor.GASMI Abdelkader and Professor BLIZAK Tahar, for accepting to discuss and examine this thesis.Furthermore, I extend my gratitude to all my professors in the Department of Mathematics at the Faculty of Mathematics and Computer Science for the knowledge and guidance they provided throughout my academic journey.Finally, I express my sincere thanks and deepest appreciation to my parents and dear family for their moral support, patience, and endurance during the preparation of this thesis.In conclusion, I ask Allah, the Almighty, to make this work beneficial and to make it sincerely for His Noble Face, for He is the Best Supporter and the Best Helper.

Dedications

I dedicate my thesis work and feelings of gratitude, love and respect to my dear parents and all my family members who have been my source of inspiration, encouragement and strength throughout the preparation of this work. I will always appreciate everything they have done.I also dedicate this work and express my special thanks to my beloved wife, the apple of my eye, and my daughters Heba, Hidaya and Shahad, who have been my inspiration, support and motivation during the completion of this thesis. I will forever be grateful for their presence in my life

ABSTRACT

In this work, we introduce the radial basis functions and their types, and we applied this technique to solve the Laplace, Poisson and telegraph equations, these solutions are approximates solutions.

RESUME

Dans ce travail, on introduit la méthode de fonctions à base radiale et à leurs types, pour déterminer la solution approchée des équations différentielles aux dérivées partielles de Laplace, Poisson et de télégraphe.

الملخص

في هذا العمل، تناولنا مقدمة عن الوظائف ذات الأساس الشعاعي وأنواعها، لكي نقوم بحل بعض المعادلات التفاضلية ذات المشتقات الجزئية مثل معادلة لابلاس وبواسون و التلغراف، علما ان هذه الحلول هي حلول تقريبية لهاته المعادلات التفاضلية.

Introduction

The main subject of this thesis concerns the applications of radial basis functions for solving differential equations (Laplace and Poisson equations as models). The proposed algorithm is based on the pseudo-spectral method using compactly supported and generalized multiquadric radial basis functions.Scattered data approximation is a rapidly growing research area that deals with the problem of reconstructing an unknown function from given scattered data. This field has many applications such as fluid-structure interaction, terrain modeling, computer science, and various fields like applied mathematics, biology, geology, and others.Polynomial interpolation is a powerful tool for approximating given data sites in the univariate setting, as a set of distinct points can be interpolated using a unique polynomial. In high-dimensional problems, it is not always possible to obtain a unique polynomial interpolation for multivariate data sites, which is justified by the Mairhuber-Curtis theorem.Traditional numerical methods, such as finite difference, finite element, or finite volume methods, were primarily motivated by early one- or two-dimensional simulations of engineering problems via partial differential equations. The discretization of these methods requires some form of underlying computational mesh, which becomes a rather difficult task in high dimensions. To overcome this problem, we must establish a basis that depends on the data locations selected arbitrarily on certain domains. Therefore, we can approximate a function without mesh generation on the domain using these basis functions called radial basis functions (RBFs). Radial basis function methods belong to a category of methods called meshless (meshfree) methods, which do not require connectivity of grid/mesh points. This is achieved by composing a univariate basic function with a norm, usually the Euclidean norm, that makes the problem insensitive to the dimension and virtually one-dimensional. Meshless methods have received much attention not only from applied mathematics but also in different fields of science.Radial basis functions are effective techniques for interpolating an unknown function on a scattered set of points, which have been used in the past few decades. These functions involve a single independent variable regardless of the dimension of the problem, so applying them in higher dimensions does not increase the difficulties. It should also be noted that the RBF approach does not require any domain elements, so it does not depend on the geometry of a domain

Chapter 1

Introductory of Radial basis function methods

Radial basis function (RBF) methods have shown the potential to be a universal grid free method for the numerical solution of partial difierential equations. Both global and compactly supported basis functions may be used in the methods to achieve a higher order of accuracy. In this chapter, we take advantage of the grid free property of the methods and use an adaptive algorithm to choose the location of the collocation points. The RBF methods produce results similar to the more well known and analyzed spectral methods, but while allowing greater flexibility in the choice of grid point locations. The adaptive RBF methods are most successful when the basis functions are chosen so that the PDE solution can be approximated well with a small number of the basis functions.

adaptive radial basis function in one and higher dimensions Ultimately, we are interested in adaptive radial basis function (RBF) PDE algorithms in two and three spatial dimensions. In this chapter, we gain insight in one dimension before proceeding to higher dimensions. The implementation and complexity of RBF methods in higher dimensions are essentially the same as in one dimension. Only the adaptive algorithm will need to be different

RBF methods RBF methods for time dependent PDEs enjoy large advantages in accuracy over other flexible, but low order methods, such finite differences, finite volumes, and finite elements. However, RBF methods share the ease of implementation and flexibility of these lower order methods. Moving grid RBF methods are easily implemented, potentially even in complex computational domains in several space dimensions. Other highly accurate spatial discretization schemes such as pseudo spectral methods do not have the inherent flexibility of the RBF methods and adaptation and complex geometries are more difficult to deal with. We have applied a modification of a simple moving grid algorithm, which was developed for use with low order finite difference methods, to RBF methods for time dependent PDEs. The adaptive RBF algorithm produces excellent results.

1.1 Radial Basis Functions

The choice of basis function is another of the flexible features of RBF methods. We will review some properties of the RBFs that we use in the numerical examples. RBFs can be globally supported, infinitely differentiable, and contain a free parameter, ε, called the shape parameter. Representatives of this type of RBF are listed in table 1.1. The global nature of RBFs of this type leads to a dense interpolation matrix. Global, infinitely differentiable RBFs typically interpolate smooth data with spectral accuracy. Details can be found in the references .

The shape parameter affects both the accuracy of the approximation and the conditioning of the interpolation matrix. In general, for a fixed number of centers N, smaller shape parameters produce the more accurate approximations, but also are associated with a poorly conditioned H. The condition number also grows with N for fixed values of the shape parameter ε. In practice, the shape parameter must be adjusted with the number of centers in order to produce a interpolation matrix which is well conditioned enough to be inverted in finite precision arithmetic. Many researchers .have attempted to develop algorithms for selecting optimal values of the shape parameter. By optimal, we mean the value of the shape parameter that produces the most accurate interpolant. However, results in this area have been limited by the realities of floating point arithmetic. The optimal choice of the shape parameter

is still an open question. In practice it is most often selected by brute force. Recently, Fornberg et. al. developed a Contour-Padé algorithm which is capable of stably computing the RBF approximation for all $\varepsilon \geqslant 0$. The results of using the Contour-Padé algorithm have shown that the optimal value of the shape parameter may not be reachable in standard floating point precision when applying traditional algorithms such as Gaussian elimination to solve a system. Several difierent strategies have been somewhat successful in reducing the ill-conditioning problem when using RBF methods in PDE problems. The strategies include:variable shape parameters, domain decomposition, preconditioning the interpolation matrix, and optimizing the center locations. Often, more than one of these strategies are used together.

In our numerical examples, we have used the multiquadric (MQ) RBF which is defined in table 1.1. Alternatively, the MQ may be deflned as $\phi(r, c) = \sqrt{r^2 + c^2}$. This is seen to be equivalent to our definition with $\varepsilon = 1/c$. It seems more natural to define the MQ in this way rather than in the traditional way, as the shape parameter now behaves in the same way it does in other infinitely smooth RBFs. The behavior as $\varepsilon \to 0$ is that the interpolant becomes more accurate, the condition number of the interpolation matrix gets larger, and the shape of the RBF becomes flatter.

For approximation with the MQ RBF we consider the generalized interpolation problem with $M = 1$. The interpolation problem with $M = 1$ takes the form

$$s(x) = \sum_{i=0}^{N} \lambda_i \phi_i(x) + b \tag{1.1}$$

where b is a constant and the auxilliary equation is

$$\sum_{i=0}^{N} \lambda_i = 0 \tag{1.2}$$

The resulting interpolation matrix will be of the form

$$H = \begin{bmatrix} \phi\left(\|x_0 - x_0\|_2\right) \dots \phi\left(\|x_0 - x_N\|_2\right) 1 \\ \vdots \quad \vdots \quad \vdots \quad \vdots \\ \phi\left(\|x_N - x_0\|_2\right) \dots \phi\left(\|x_N - x_N\|_2\right) 1 \\ 1 \quad \dots \quad 1 \quad 0 \end{bmatrix}$$

For the MQ the interpolation matrix constructed from the generalized interpolations problem with $M = 1$ is guaranteed to be invertible for distinct centers

Name of RBF	Definition
Multiquadric (MQ)	$\phi(r,\varepsilon) = \sqrt{1+(\varepsilon r)^2}$
Inverse Quadratics (IQ)	$\phi(r,\varepsilon) = 1/\left(1+(\varepsilon r)^2\right)$
Inverse Multiquadric (IMQ)	$\phi(r,\varepsilon) = 1/\sqrt{1+(\varepsilon r)^2}$
Gaussian (GA)	$\phi_i(r,\varepsilon) = e^{-(\varepsilon r)^2}$

Table 1.1: Deferente RBF types

Global, inflnitely smooth RBFs

For a discussion of the merits of using the MQ RBF with an appended constant see references .

An alternative to the global, infinitely smooth RBFs are compactly supported RBFs (CSRBFs). Wendland's CSRBFs are representative of this class.

The Wendland functions, $\phi_{t,K}$ are strictly positive definite in $\mathbb{R}^d$ for all $d \leqslant d_0$ and can be constructed to have any desired amount of smoothness $2K$, i.e., $\phi \in C^{2k}$. The parameter ℓ is $\ell = \left[\frac{d}{2}\right] + K + 1$. For $K = 0, 1, 2, 3$, the functions can be computed by an explicit formulae [10]. The Wendland functions are defined to have compact support on the interval $[0; 1]$ but may be scaled to have compact support on $[0, \delta]$ by replacing r with r/δ for $\delta > 0$. The scaling factor δ can be constant or it can be variable at different centers. A way to specify the optimal value of δ is currently not known. In our numerical experiments we have used the Wendland CSRBF

$$\phi_{4,2} = (1-r)_+^6 \left(3 + 18r + 35r^2\right) \tag{1.3}$$

Where

$$(1-r)_+^6 = \begin{cases} (1-r)_+^6 & ; 0 \leqslant r < 1 \\ 0 & ; r \geqslant 1 \end{cases} \tag{1.4}$$

which are in C^4 and are positive definite in up to three space dimensions. Since $\phi_{4,2}$ (W42) is positive definite we need only consider the standard interpolation problem and the matrix H will be nonsingular for a distinct set of centers. If the support of the basis functions are small compared to the size of the computational domain of the PDE, banded matrix algorithms can be used to invert the interpolation matrix. Error estimates for approximations of

$f \in H^s(\mathbb{R}^d)$ by Wendland's CSRBFs are of the form

$$\|f - S_f\|_{I_\infty(\Omega)} \leqslant C h^{k+\frac{1}{2}} |f|_{H(\mathbb{R}^d)} \tag{1.5}$$

where h denotes the "meshsize", i.e., the separation distance of the centers, $h = \sup_{x\in\Omega} \min \|x - x_j\|$ for $x_j \in \mathbb{R}^d$. $H^s(\mathbb{R}^d)$ is the usual Sobolev s of functions with s derivatives bounded in L^2 and $s = \frac{d}{2} + k + \frac{1}{2}$ gives the regularity of the data.

1.2 The numerical solution of PDEs by RBF

The numerical solution of PDEs by RBF methods is based on a scattered data interpolation problem which we review in this section. let $x_0, x_1, \ldots, x_N \in \Omega \subset \mathbb{R}^n$ be a given set of centers. A radial basis function is a function $\phi_i(x) = \phi(\|x - x_i\|_2)$, which depends only on the distance between $x \in \mathbb{R}^d$ and a fixed point $x_j \in \mathbb{R}^d$. Each function ϕ_j is radially symmetric about the center x_j. The radial basis function interpolation problem may be described as, given data $f_j = f(x_j), i = 0, 1, \ldots, N$, the interpolating RBF approximation is: $s(x) = \sum_{i=0}^N \lambda_i \phi_i(x)$.

where the expansion coefficients λ_j, are chosen so that $s(x_i) = f_i$. That is, they are obtained by solving the linear system

$$H\lambda = f$$

Where the elements of the interpolation matrix are $H_{ij} = \phi(\|x_i - x_j\|_2), \lambda = [f_0, \ldots, f_N]^T$. For the RBFs that we have considered in this work, the interpolation matrix can be shown to be invertible for distinct interpolation points.

A generalized interpolation problem also may be considered. The generalized interpolation problem is

$$s(x) = \sum_{i=0}^{N} \lambda_i \phi(x) + \sum_{k=1}^{M} b_k p_k(x)$$

in which a finite number of d-variate polynomials of at most order M are added to the RBF basis. The polynomials $p_k(x)$ are the polynomials spanning π_M, that is they are the polynomials of degree at most M. The extra equation(s) needed to complete the generalized interpolation problem are chosen to be

$$\sum_{j=0}^{N} \lambda_j p_k(x_j) = 0$$

For $k = 1, \ldots, M$. Interpolation problem must be considered when using RBFs, such as the cubics $\phi(r) = r^3$, as the basic interpolation problem does not lead to a guaranteed invertible interpolation matrix . Also, the generalized interpolation problem may lead to an approximation with some desirable properties that an approximation from the standard interpolation problem may lack, such as a degree of polynomial accuracy. This is the case with the multiquadric RBF .

Despite the fact that H can be shown to be invertible for all ϕ of the interest, the linear system may often be very ill-conditioned and it may be impossible to solve accurately using standard floating point arithmetic. The conditioning of H is measured by the condition number defined as

$$k(H) = \|H\|\|H^{-1}\| = \sigma_{\max}/\sigma_{\min} \tag{1.6}$$

where are the singular values of H. The condition number of H is influenced by the number of centers, the minimum separation distance of the centers, as well as values of parameters, defined below, such as the shape parameter and the support.

1.3 RBF methods for time dependent PDEs

Derivatives of the interpolant (1) or (3) may be calculated in a straightforward manner. For instance, using the interpolant in , the derivatives at the centers x_j can be calculated as

$$S^{(n)}(x_j) = \sum_{i=0}^{N} \lambda_i \phi_i^{(n)}(x_j). \tag{1.7}$$

for $n = 1, 2, \ldots$. The spatial derivatives can be written compactly in matrix form as

$$S^{(n)} = H^{(n)}\lambda. \tag{1.8}$$

where the elements of $H^{(n)}$ are $\phi^{(n)}(\|x_i - x_j\|_2)$.

In the context of a time dependent PDE method, where derivatives may need to be evaluated thousands of times, it is often more efficient to form the derivative matrix

$$D^{(n)} = H^{(n)} H^{-1}. \tag{1.9}$$

Then spatial derivatives can be approximated by a single matrix by vector multiplication

$$s^{(n)} = D^{(n)} s. \tag{1.10}$$

To describe how to implement a RBF method for solving a time dependent PDE on a fixed grid, we use Burgers' equation as an example. A fixed time step has been used, but variable time stepping is possible. The PDE is discretized in space with radial basis functions to get the semi-discrete system

$$s_t = F(s). \tag{1.11}$$

The system of ODEs is then advanced in time with any ODE method. In the numerical examples, we have used an explicit fourth-order Runge-Kutta method. At time $t = 0$ the derivative matrixes, $D^{(1)}$ and $D^{(2)}$, are constructed. At each internal Runge-Kutta stage, we calculate $s^{(1)} = D^{(1)} s, s^{(2)} = D^{(2)} s$, and then for $i = 0, \ldots, N$; $F_1 = v s^{(2)}(x_i) - s(x_i) s^{(1)}(x_i)$. The implementation of the method is extremely simple.

1.4 Adaptive Grids

It has generally been accepted, at least for problems in one space dimension, that adaptive grid methods are capable of resolving PDE solutions that contain regions of rapid variation with acceptable accuracy and without using an excessive number of grid points. Adaptive grid methods and applications in one space dimension, have been extensively studied. Many one-dimensional adaptive grid algorithms for time-dependent PDEs in the context of finite difference, finite element, and pseudo spectral methods have been described. Details and further references may be found in [1,2,4]. The adaptive grid algorithm that we have used is a slightly modification of the equidistribution of arclength algorithm for one dimensional systems of PDEs described in [1]. We have modified the interpolation step, which used cubic polynomials at interior nodes and quadratic polynomials at the nodes next to the boundary, to instead use the same RBFs used in the PDE solution at all nodes. Thus, the method does not require any modifications near the boundaries.

This allows the adaptive RBF methods to maintain an overall high order of accuracy. When we apply the adaptive algorithm with second-order finite differences, we have retained the cubic interpolation step. The algorithm is simple and computationally inexpensive in that it is not necessary to transform the original PDE into a new coordinate system, or is it necessary to solve an additional companion PDE to choose the coordinate system and node distribution. Other algorithms may result in different, and possibly "better" grids being used, but for our purposes, the features of the RBF methods we wish to examine will remain very similar, regardless of particular adaptive algorithm used.

In the adaptive algorithm, we start at time t^0 to with a uniform grid x_j^0. To advance the PDE in time with the adaptive grid algorithm, we start by assuming that at time level t^n we have computed approximate solutions s_j^n, by a radial basis function method, to the true solution $u\left(x_j^n, t^n\right)$ on a grid x_j^n , where $j = 0, \ldots, N$. Then, the RBF method is used on the grid x_j^n to obtain approximations s_j^{-n+1} to $u\left(x_j^n, t^{n+1}\right)$. Next, the points $\left(x_j^n, s_j^{-n+1}\right)$ are joined by straight lines and the length θ^{n+1} of the resulting polygon is computed. Then the points P_j^{n+1} on the polygon are found which divide its total length into N equal parts. The new nodes x_j^{n+1} are found as the projection of P_j^{n+1} onto the x-axis. Finally, s_j^{n+1} ,

the approximation to $u\left(x_j^{n+1}, t^{n+1}\right)$, is computed by using RBFs to interpolate the values $\left(x_j^n, s_j^{-n+1}\right)$ to $\left(x_j^{n+1}, s_j^{n+1}\right)$. The adaptive algorithm contains two parameters that control the adaptation. The parameter μ causes the adaptation to be performed every μ time steps. The parameter β controls the relative size of the largest and smallest grid spacings by ensuring that

$$\max_i h_i \leqslant \sqrt{1+\beta} \min_i h_i \tag{1.12}$$

where $h_i = x_i - x_{i-1}$. When using the adaptive algorithm with CSRBFs, the bandwidth, b , of the interpolation matrix H will be

$$b = \left\lceil \frac{\delta}{\min_i h_i} \right\rceil \tag{1.13}$$

The computational cost of choosing a new grid is relatively small. However, setting up the RBF method on the new grid involves constructing derivative matrices for the new grid. Thus, the setup costs of the adaptive method may be prohibitive, unless the PDE solution can be approximated with a relatively small number of centers or unless CSRBFs are used which lead to a narrowly banded interpolation matrix that can be efficiently inverted.

To obtain an accurate numerical approximation with RBFs and the adaptive grid algorithm, we have found that a good strategy is to monitor the condition number of the interpolation matrix H that is used to form new derivative matrices and to interpolate to the new set of centers each time the solution is re-gridded. The condition number of H should not exceed 5×10^{10} during any stage of the time evolution of the solution. If the condition number does become too large, the stage should be rejected and recalculated with a smaller value of β, which decreases the minimum separation distance of the centers. Another option to reduce the condition number of H for a rejected stage is to decrease the number of centers used. Both the values of β and N may be adapted when a re-gridding takes place. If it is not possible to achieve small enough condition numbers in a single domain, domain decomposition will be necessary.

Chapter 2

Preliminaries and notions about partial differential equations

2.1 Fundamental Notions on PDEs

Definition 2.1.1. Let $u : \mathbb{R}^n \to \mathbb{R}$ be a function. A partial differential equation (PDE) for the function u is a relation between $u = u(x_1, x_2, \ldots, x_n)$ and a finite number of partial derivatives. A PDE is of the form: $F(x_1, x_2, \ldots, x_n, u, \frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_1}, \ldots, \frac{\partial u}{\partial x_1}, \ldots, \frac{\partial^n u}{\partial x_1^k \partial x_2^k \ldots \partial x_n^k}) = 0$ Where $k_1, k_2, \ldots, k_n$ are positive integers such that $k_1 + k_2 + \cdots + k_n = m$. The equation is considered in a domain Ω of $\mathbb{R}^n$ .We say that u is a solution of the partial differential equation in $\Omega \subset \mathbb{R}^n$ if, after substituting u and its partial derivatives, F vanishes for all $x_1, x_2, \ldots, x_n \in \Omega$ The order m of a partial differential equation is the order of the highest partial derivative appearing in it.A system of partial differential equations is formed by several PDEs involving one or more unknown functions u_1. A partial differential equation is linear if F is linear with respect to u and its partial derivatives, and non-linear otherwise.The solution of a PDE of order m generally depends on m arbitrary functions of $n-1$ variables.The general solution of a PDE is the one that allows finding all the solutions of the equation (except for singular solutions) by assigning particular values to the arbitrary functions

Example 2.1.1. The equation $\frac{\partial^2 u}{\partial t^2} = 2\frac{\partial u}{\partial x \partial t} + u$ is a second PDE.

The equation $\frac{\partial u}{\partial x^3} + 2\left(\frac{\partial u}{\partial t}^2\right) = \frac{\partial^2 u}{\partial x^2}$ is a non-linear PDE for the function $u(x,t)$

The equation $\frac{\partial^2 u}{\partial x^2} = x^2 \frac{\partial u}{\partial x^2} + e^{-x^2}$ is a linear PDE. In general, a second-order linear PDE in $\mathbb{R}^2$ takes the form: $A(x,y)\frac{\partial^2 u}{\partial x^2} + 2B(x,y)\frac{\partial^2 u}{\partial x \partial y} + C(x,y)\frac{\partial u}{\partial x} + b(x,y)\frac{\partial u}{\partial y} + c(x,y)u = f(x,y)$ where $A(x,y), B(x,y), \ldots, C(x,y), f(x,y)$ are functions of x, y definded in a domain D of the plane. If $f(x,y) = 0$ in D, the equation is said to be homogeneous or without a right-hand side, otherwise non-homogeneous or with a right-hand side.

Initial and Boundary Conditions To find particular solutions from the general solution, we impose the respective conditions on the set of solutions. The most common constraints are:

1. Initial condition:
   If u is a function of $u(x,t) \in \mathbb{R}^n \times \mathbb{R}$, we give $u(x,t_0) = \phi_0(x)$ or $D_2^n u(x,t_0) = \phi_n(x)$, which are also called Cauchy conditions.

2. Boundary condition:
   If u is a function of $x \in \Omega \subset \mathbb{R}^n$, we have two types of constraints:

   - Dirichlet condition where u is fixed on the boundary of Ω. $u/_{\partial\Omega} = g$
     Neumann condition where the normal derivative of u is fixed: $\frac{du}{dk}/_{\partial\omega} = g$.

Well-Posed Problems

Consider a partial differential equation on a domain Ω with possible auxiliary conditions on the solution.

Definition 2.1.2. A problem is said to be well-posed if we have:

- Existence of a solution to the problem.
- Uniqueness of this solution.
- Stability with respect to the problem data.

Example 2.1.2. The question of uniqueness is not a superior one. Indeed, take the Poisson problem

$$-\Delta u(x, y) = f(x, y) \ \forall (x, y) \in \partial\Omega$$

which can, for example, represent the problem of finding the temperature in a domain Ω knowing the heat flux on the boundary $\partial\Omega$ of the domain. It is easy to verify that if there is a solution, there are infinitely many solutions up to an additive constant. This problem is ill-posed. Similarly, in structural analysis, it is necessary to fix a certain number of constraints to obtain a unique equilibrium solution. Otherwise, we can also obtain a solution up to additive constants, which mechanics express as a solution up to a rigid displacement. Additional conditions are therefore needed to have a single solution. These appropriate conditions depend on the problem studied. In practice, we distinguish the time variable from the spatial variables due to the irreversible nature of certain physical phenomena. We then distinguish the boundary conditions of the spatial domain and the initial conditions

2.2 Classical Partial Differential Equations

The classical second-order partial differential equations that appear quite often in applications and dominate the theory of partial differential equations are

2.2.1 The Heat Equation

The heat equation is given by the following partial differential equation:

$$\frac{\partial u}{\partial t} = \alpha^2 \frac{\partial^2 u}{\partial x^2} \tag{2.1}$$

Equation (2.1) is known as the heat equation and appears in the study of heat conduction and other diffusion processes.

2.2.2 The Wave Equation

The wave equation is given by the following partial differential equation:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} \tag{2.2}$$

Equation (3.1) is the wave equation, which appears in the study of acoustic waves, water waves, and electromagnetic waves

2.2.3 Laplace's Equation

Laplace's equation is given by the following partial differential equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \tag{2.3}$$

Equation (2.3) is known as Laplace's equation, the most famous partial differential equation. It arises in the study of various applications such as steady flows, vibrating membranes, and electric potentials. For this reason, Laplace's equation is often called the potential equation.The partial differential equations (2.1), (2.2), and (2.3) impose initial and boundary conditions on the function u. These conditions will be given by the physical and biological problems themselves and will be chosen to ensure a unique solution to our equation.

Chapter 3

Application RBF for solving a PDEs

Recently, it is found that telegraph equation is more suitable than ordinary diffusion equation in modelling reaction diffusion for such branches of sciences. In this article, we propose a numerical scheme to solve the one-dimensional hyperbolic telegraph equation using collocation points and approximating the solution using thin radial basis function. The scheme works in a similar fashion as finite difference methods. The results of numerical experiments are presented, and are compared with analytical solutions to confirm the good accuracy of the presented scheme.

3.1 Introduction

In the present work we are dealing with the numerical approximation of the following second order hyperbolic problem:

$$\frac{\partial^2 u}{\partial t^2} + \alpha \frac{\partial u}{\partial t} + \beta u = \frac{\partial^2 u}{\partial x^2} + f(x,t), \tag{3.1}$$

where α and β are known constant coefficients. Equation (3.1), referred to as second-order telegraph equation with constant coefficients, models mixture between diffusion and wave prop- agation by introducing a term that accounts for effects of finite velocity to standard heat or mass transport equation [1]. However, Eq.(3.1) is commonly used in signal analusis for transmission and propagation of electrical signals [2] and also has applications in other fields (see [3] and the references therein).

In recent years, much attention has been given in the literature to the development, analysis, and implementation of stable methods for the numerical solution of second-order hyperbolic equations,. three-level implicit unconditionally stable alternating direction implicit schemes for the two and three-space-dimensional linear hyperbolic equations. These schemes are second-order accurate both in space and time. Of concern are suspension flows. These combine directed and random motion and are tradition- ally modeled by parabolic partial differential equations. Sometimes they can be better modeled (in terms of fitting the data generated by certain blood flow experiments) by hyperbolic equations such as the telegraph equation, which has parabolic (or analytic) asymptotic. In particular, the experimental results described seem to be better modeled by the telegraph equation than by the heat equation. Also the interested reader for an application of the model. The existence of time-bounded solutions of nonlinear bounded perturbations of the telegraph equation with Neumann boundary conditions has recently been considered. The approach is based upon a Galerkin method combined with the use of some Lyapunov functionals. Finite difference methods are known as the first techniques for solving partial differential equa- tions. Even

though these methods are very effective for solving various kinds of partial differential equations, conditional stability of explicit finite difference procedures and the need to use large amount of CPU time in implicit finite difference schemes limit the applicability of these methods. Furthermore, these methods provide the solution of the problem on mesh points only and accuracy of these well known techniques is reduced in non-smooth and nonregular domains. To avoid the mesh generation, in recent years meshless techniques have attracted attention of researchers. In a meshless (mesh free) method a set of scattered nodes are used instead of meshing the domain of the problem. Some meshless schemes are the element free Galerkin method, the reproducing kernel particle, the local point interpolation and etc. For more descriptions In the last 20 years, the radial basis functions method is known as a powerful tool for scattered data interpolation problem. The use of radial basis functions as a meshless procedure for numer- ical solution of partial differential equations is based on the collocation scheme. Because of the collocation technique, this method does not need to evaluate any integral. The main advantage of numerical procedures which use radial basis functions over traditional techniques is meshless property of these methods. Radial basis functions (RBFs) are used actively for solving partial differential equations. In the last decade, the development of the RBFs as a truly meshless method for approximating the solutions of PDEs has drawn the attention of many researchers in science and engineering. One of the domain-type meshless methods, the so-called Kansa's method developed by Kansa in 1990 [16,18], is obtained by directly collocating the RBFs, particularly the multiquadric (MQ), for the numerical approximation of the solution. Kansa's method was recently extended to solve various ordinary and partial differential equa- tions including 1D nonlinear Burgers' equation with shock wave, shallow water equations for tide and currents simulation, heat transfer problems, KdV equation, and free boundary problems]. Fasshauer later modified Kansa's method to a Hermite type collocation method for the solvability of the resultant collocation matrix. The traditional RBFs are globally defined functions which result in a full resultant coefficient matrix. This hinders the application of the RBFs to solve large scale problems due to severe ill-conditioning of the coefficient matrix. To tackle this ill-conditioning problem, a new class of compactly supported RBFs were constructed. For the theoretical developments of the RBFs in scattered data interpolation, Madych and Nelson showed that the RBF-MQ interpolant employs exponential convergence with minimal semi-norm errors. Recently, Franke and Schaback provided a theoretical proof in using the RBFs for the numerical solutions of PDEs. More recently, Hon and Wu] gave a theoretical justification in combining the RBFs with those advanced techniques of domain decomposition, multilevel/multigrid, Schwarz iterative schemes, and preconditioning in the FEM discipline. This chapter presents a new numerical scheme to solve the second-order hyperbolic telegraph equation using the collocation method and approximating directly the solution using thin plate splines radial basis function (Kansa's method). The scheme is similar to finite-difference methods

3.2 RADIAL BASIS FUNCTION APPROXIMATION

The approximation of a distribution $u(x)$, using radial basis functions, may be written as a linear combination of N radial functions; usually it takes the following form:

$$u(x) \simeq \sum_{j=1}^{N} \lambda_j \varphi(x, x_j) + \psi(x) \quad \text{for} x \in \Omega \subset \mathbb{R}^d, \tag{3.2}$$

where N is the number of data points, $x = (x_1, x_2, \ldots, x_d)$, d is the dimension of the problem, λ' s are coefficients to be determined and φ is the radial basis function. Equation (3.2) can

be written without the additional polynomial ψ. In that case φ must be unconditionally positive definite to guarantee the solvability of the resulting system (e.g., Gaussian or inverse multiquadrics). However, ψ is usually required when φ is conditionally positive definite, i.e., when φ has a polynomial growth towards infinity. For two examples we can mention thin plate splines and multiquadrics. We will use thin plate splines for the numerical scheme. The generalized thin plate splines (TPS) defined as:

$$\varphi(x, x_j) = \varphi(r_j) = r_j^{2m} \log(r_j), \quad m = 1, 2, 3, \ldots, \tag{3.3}$$

where $r_j = \|x - x_j\|$ is the Euclidean norm.

Since φ in (3.2) is C^{2m-1} continuous, so higher order thin plate splines must be used for higher order partial differential operators. For the one-dimensional hyperbolic telegraph equation, $m = 2$ is used for thin plate splines (i.e., second order thin plate splines). If $\mathcal{P}_q^d$ denotes the space of d-variate polynomials of order not exceeding q, and letting the polynomials $P_1, \ldots, P_m$ be the basis of $\mathcal{P}_q^d$ in $\mathbb{R}^d$, then the polynomial $\psi(x)$, in Eq. (3.2), is usually written in the following form:

$$\Psi(x) = \sum_{i=1}^{m} \xi_i P_i(x), \tag{3.4}$$

where $m = (q - 1 + d)!/(d!(q - 1)!)$.

To determine the coefficients $(\lambda_1, \ldots, \lambda_N)$ and $(\xi_1, \ldots, \xi_m)$, the collocation method is used. However, in addition to the N equations resulting from collocating (3.2) at the N points, extra m equations are required. This is insured by the m conditions for (3.2),

$$\sum_{j=1}^{N} \lambda_j P_i(x_j) = 0, \quad i = 1, \ldots, m \tag{3.5}$$

In a similar representation as (3.2), for any linear partial differential operator $\mathcal{L}$, $\mathcal{L}u$ can be approximated by

$$\mathcal{L}u(x) \simeq \sum_{j=1}^{N} \lambda_j \mathcal{L}_\varphi(x, x_j) + \mathcal{L}\psi(x). \tag{3.6}$$

3.3 THE HYPERBOLIC TELEGRAPH EQUATION

Let us consider the following one-dimensional hyperbolic telegraph equation:

$$\frac{\partial^2 u}{\partial t^2} + \alpha \frac{\partial u}{\partial t} + \beta u = \frac{\partial^2 u}{\partial x^2} + f(x, t), \quad x \in \Omega = [a, b] \subset \mathbb{R},\ 0 < t \leqslant T, \tag{3.7}$$

with the initial conditions

$$\begin{cases} u(x, 0) = g_1(x), & x \in \Omega \\ u_t(x, 0) = g_2(x), & x \in \Omega' \end{cases} \tag{3.8}$$

and Dirichlet boundary condition

$$u(x, t) = h(x, t), \quad x \in \partial\Omega, 0 < t \leqslant T, \tag{3.9}$$

where α and β are known constant coefficients, f , g_1, g_2, and h are known functions, and the function u is unknown.

First, let us discretize (3.2) according to the following θ-weighted scheme

$$\frac{u(x,t+\delta t)-2u(x,t)+u(x,t-\delta t)}{(\delta t)^2}+\alpha\frac{u(x,t+\delta t)-u(x,t-\delta t)}{2\delta t}$$
$$=\theta[\nabla^2 u(x,t+\delta t)-\beta u(x,t+\delta t)]+(1-\theta)[\nabla^2 u(x,t)-\beta u(x,t)]+f(x,t+\delta t),$$

where ∇ is the gradient differential operator,$0 \leqslant \theta \leqslant 1$, and δt is the time step size. Rearranging (3.4), using the notation $u^n = u(x,t^n)$ where $t^n = t^{n-1}+\delta t$, we obtain

$$\left(1+\frac{\alpha\delta t}{2}+\beta\theta(\delta t)^2\right)u^{n+1}-\theta(\delta t)^2\nabla^2 u^{n+1}$$
$$=(2-\beta(1-\theta)(\delta t)^2)u^n+(1-\theta)(\delta t)^2\nabla^2 u^n+\left(\frac{\alpha\delta t}{2}-1\right)u^{n-1}+(\delta t)^2 f^{n+1}.$$

Assuming that there are a total of $(N-2)$ interpolation points, $u(x,t^n)$ can be approximated by

$$u^n(x)\simeq\sum_{j=1}^{N-2}\lambda_j^n\varphi(r_j)+\lambda_{N-1}^n x+\lambda_N^n. \tag{3.10}$$

To determine the interpolation coefficients $(\lambda_1,\lambda_2,\ldots,\lambda_{N-1},\lambda_N)$, the collocation method is used by applying (3.6) at every point x_i, $i=1,2,\ldots,N-2$. Thus we have

$$u^n(x_i)\simeq\sum_{j=1}^{N-2}\lambda_j^n\varphi(r_{ij})+\lambda_{N-1}^n x_i+\lambda_N^n, \tag{3.11}$$

where $r_{ij}=\sqrt{(x_i-x_j)^2}$. The additional conditions due to (2,4) are written as

$$\sum_{j=1}^{N-2}\lambda_j^n=\sum_{j=1}^{N-2}\lambda_j^n x_j=0.$$

Writing (3.7) together with (3.8) in a matrix form we have

$$[u]^n=A[\lambda]^n.$$

where $[u]^n=\left[u_1^n u_2^n.....u_{N-2}^n 00\right]^T$, $[\lambda]^n=[\lambda_1^n\lambda_2^n\ldots\lambda_N^n]^T$ and $A=[a_{ij},1\leqslant i,j\leqslant N]$ is given by

$$\begin{bmatrix}\Phi_{11} & \ldots & \Phi_{1(N-2)} & x_1 & 1\\ \vdots & \ddots & \vdots & \vdots & \vdots\\ \Phi_{(N-2)} & \ldots & \Phi_{(N-2)(N-2)} & x_{N-2} & 1\\ x_1 & \ldots & x_{N-2} & 0 & 0\\ 1 & \ldots & 1 & 0 & 0\end{bmatrix} \tag{3.12}$$

Assuming that there are $p<(N-2)$ internal points and $(N-2-p)$ boundary points, then the $(N\times N)$ matrix A can be splitted into: $A=A_b+A_b+A_e$, where

$$\begin{aligned}A_d &= [a_{ij} \text{ for } (1\leq i\leq p, 1\leq j\leq N) \text{ and 0 elsewhere }],\\ A_b &= [a_{ij} \text{ for } (p+1\leq i\leq N-2, 1\leq j\leq N) \text{ and 0 elsewhere }],\\ A_e &= [a_{ij} \text{ for } (N-1\leq i\leq N, 1\leq j\leq N) \text{ and 0 elsewhere }].\end{aligned} \tag{3.13}$$

Using the notation $\mathcal{L}A$ A to designate the matrix of the same dimension as A and containing the elements a_{ij} , where a_{ij}= $\mathcal{L}A$ a_{ij} 1 ≤i, j ≤N, then Eq. (3.5) together with (3.3) can be written, in the matrix form, as

$$B[\lambda]^{n+1} = C[\lambda]^n + (\frac{\alpha\delta t}{2} - 1)[u_d]^{n-1} + (\delta t)^2[f]^{n+1} + [H]^{n+1}, \tag{3.14}$$

where

$$B = \left(1 + \frac{\alpha\delta t}{2} + \beta\theta(\delta t)^2\right) A_d - \theta(\delta t)^2\nabla^2 A_d + A_b + A_e,$$
$$C = (2 - \beta(1-\theta)(\delta t)^2)A_d + (1-\theta)(\delta t)^2\nabla^2 A_d,$$
$$[u_d]^{n-1} = [u_1^{n-1} \dots u_p^{n-1} 0 \dots 0]^T, [f]^{n+1} = [f_1^{n+1} \dots f_p^{n+1} 0 \dots 0]^T$$

and $[H]^{n+1} = [0 \dots 0 h_{p+1}^{n+1} \dots h_{N-2}^{n+1} 0\ 0]^T$. Equation (3.12) is obtained by combining (3.5), which applies to the domain points, while (3.3) applies to the boundary points., At $n = 0$ the Eq. (3.12) has the following form:

$$B[\lambda]^1 = C[\lambda]^0 + \left(\frac{\alpha\delta t}{2} - 1\right)[u_d]^{-1} + (\delta t)^2[f]^1 + [H]^1 \tag{3.15}$$

To approximate u^{-1} the second in itial condition can be used. For this purpose we discretize the second initial condition as

$$\frac{u^1(x) - u^{-1}(x)}{2\delta t} = g_2(x), x \in \Omega. \tag{3.16}$$

Writing (3.13) together with (3.14) we have

$$\left(B + \left(1 - \frac{\alpha\delta t}{2}\right) A_d\right)[\lambda]^1 = C[\lambda]^0 + (2 - \alpha\delta t)\delta t[G] + (\delta t)^2[f]^1 + [H]^1 \tag{3.17}$$

where $[G] = [(g_2)_1 \dots (g_2)_p 0 \dots 0]^T$.

Since the coefficient matrix is unchanged in time steps, we use the LU factorization to the coefficient matrix only once and use this factorization in our algorithm.

3.4 NUMERICAL RESULTS

In this section we present some numerical results to test the efficiency of the new scheme for solving the hyperbolic telegraph equation.

Example 3.4.1. In this example, we consider the hyperbolic telegraph Eq. (3.1) with $\alpha = 1$ and $\beta = 1$ in the interval $0 \leqslant x \leqslant 4$. The initial conditions are given by

$$\left\{ \begin{array}{c} \mu(x,o) = g_1(x) = e^x . 0 \leq x \leq 4 \\ \mu_t(x,o) = g_2(x) = -e^x . 0 \leq x \leq 4 \end{array} \right\}$$

and the analytical solution is given

$$\mu(x,t) = \exp(x - t) \tag{3.18}$$

In this case $f(x,t) = 0$.We extract the boundary function $h(x,t)$ from the exact solution. The L_∞ and L2 errors and Root-Mean-Square (RMS) of errors are obtained in Table I for $t = 1, 2, 3, 4$,and 5. The graph of analytical and estimated functions for t = 5 is given in Fig. 1. We also draw the space-time graph of the estimated solution in Fig. 2.

Table 3.1: Computational domain is $[0, 4]$

t	$L_\infty -$ error	$L_2 -$ error	RMS	Time(s)
1	2.2931×10^{-5}	1.7163×10^{-4}	8.5711×10^{-6}	7
2	1.3305×10^{-5}	1.0283×10^{-4}	5.1353×10^{-6}	14
3	6.9011×10^{-6}	4.3509×10^{-5}	2.1727×10^{-6}	21
4	3.2736×10^{-6}	1.8865×10^{-5}	9.4211×10^{-7}	28
5	3.0766×10^{-6}	2.0235×10^{-5}	1.0105×10^{-6}	35

L_∞, L_2, and RMS errors, with $dt = 0.001$,$dx = 0.01$.

Figure 3.1: Analytical and estimated function in $t = 5s$, with $dt = 0.001$ and $dx = 0.01$, for Example 1.

Figure 3.2: Space-time graph of the solution up to $t = 5s$, with $dt = 0.001$ and $dx = 0.01$, for Example 1.

Table 3.2: Computational domain is $[0, 4]$

t	$L_\infty -$ error	$L_2 -$ error	RMS	Time(s)
0.5	8.3721×10^{-6}	7.9491×10^{-4}	6.3239×10^{-6}	5
1.0	1.5680×10^{-5}	1.4554×10^{-4}	1.1579×10^{-5}	12
1.5	1.7412×10^{-5}	1.5895×10^{-4}	1.2645×10^{-5}	19
2.0	1.5813×10^{-5}	1.4185×10^{-4}	1.1285×10^{-5}	28

L_∞, L_2, and RMS errors, with $dt = 0.0001$,$dx = 0.02$.

Example 3.4.2. Consider the hyperbolic telegraph Eq. (3.1) in the interval $0 \leqslant x \leqslant \pi$. The initial conditions are given by

$$\begin{cases} u(x,0) = g_1(x) = \sin(x), & 0 \leqslant x \leqslant \pi \\ u_1(x,0) = g_2(x) - \sin(x), & 0 \leqslant x \leqslant \pi \end{cases} \tag{3.19}$$

The analytical solution is given in [4] as

$$u(x,t) = \exp(-t)\sin(x). \tag{3.20}$$

For this example, we solved the equation in two cases $\alpha = 4$, $\beta = 2$ and $\alpha = 6, \beta = 2$. For each of these two cases we have

$$f(x,t) = (2 - \alpha + \beta)\exp(-t)\sin(x).$$

We extract the boundary function $h(x,t)$ from the exact solution. The L_∞ and L_2 errors and RMS of errors for these two cases are obtained in Tables II and III for $t = 0.5, 1, 1.5$, and 2. The graph of analytical and estimated functions for the second case at $t = 2$ is given in Fig. 3. We also draw the space-time graph of estimated solution for the second case in Fig. 4.

Example 3.4.3. In this example, we consider the hyperbolic telegraph Eq. (3.1) with $\alpha = 1$ and $\beta = 1$ in the interval $0 \leqslant x \leqslant 1$. The initial conditions are given by

$$\begin{cases} u(x,0) = g_1(x) = 0, & 0 \leqslant x \leqslant 1 \\ u_t(x,0) = g_2(x) = 0, & 0 \leqslant x \leqslant 1 \end{cases} \tag{3.21}$$

and the exact solution by [1] is

$$u(x,t) = (x - x^2)t^2 \exp(-t). \tag{3.22}$$

Table 3.3: Computational domain is $[0, 4]$

t	$L_\infty -$ error	$L_2 -$ error	RMS	Time(s)
0.5	8.3721×10^{-6}	7.9491×10^{-4}	6.3239×10^{-6}	5
1.0	1.5680×10^{-5}	1.4554×10^{-4}	1.1579×10^{-5}	12
1.5	1.7412×10^{-5}	1.5895×10^{-4}	1.2645×10^{-5}	19
2.0	1.5813×10^{-5}	1.4185×10^{-4}	1.1285×10^{-5}	28

L_∞, L_2, and RMS errors, with $dt = 0.0001$,$dx = 0.02$.

Figure 3.3: Analytical and estimated function in $t = 2s$, with $dt = 0.0001$ and $dx = 0.02$, for Example 2

The right hand side function is

$$f(x,t) = (2 - 2t + t^2)(x - x^2)\exp(-t) + 2t^2 \exp(-t). \tag{3.23}$$

The L_∞ and $L2$ errors and RMS of errors are obtained in Table IV for $t = 1, 2, 3, 4$, and 5. The graph of analytical and estimated functions for $t = 5$ and space-time graph of estimated solution are given in Figs. 5 and 6.

Figure 3.4: Space-time graph of the solution up to $t = 2s$, with $dt = 0.0001$ and $dx = 0.02$, for Example 2

Table 3.4: Computational domain is $[0, 4]$

t	$L_\infty -$ error	$L_2 -$ error	RMS	Time(s)
1	1.8479×10^{-5}	1.4386×10^{-4}	1.4315×10^{-5}	0
2	1.0713×10^{-5}	8.0879×10^{-5}	8.0478×10^{-6}	0
3	1.8161×10^{-5}	1.2944×10^{-4}	1.2880×10^{-5}	1
4	1.6489×10^{-5}	1.1845×10^{-4}	1.1786×10^{-5}	1
5	1.0455×10^{-5}	7.5545×10^{-5}	7.5170×10^{-6}	2

L_∞, L_2, and RMS errors, with $dt = 0.001$,$dx = 0.01$.

Figure 3.5: Analytical and estimated function in $t = 5s$, with $dt = 0.001$ and $dx = 0.01$, for Example 3.

Table 3.5: Computational domain is $[0, 1]$

t	$L_\infty -$ error	$L_2 -$ error	RMS	Time(s)
1	8.5573×10^{-5}	6.1544×10^{-4}	6.1239×10^{-5}	0
3	6.4755×10^{-5}	4.6574×10^{-4}	4.6343×10^{-5}	1
5	5.7041×10^{-5}	4.1033×10^{-4}	4.0829×10^{-5}	2
7	5.4338×10^{-5}	3.9092×10^{-4}	3.8898×10^{-5}	3
10	5.2901×10^{-5}	3.8057×10^{-4}	3.7868×10^{-5}	5

L_∞, L_2, and RMS errors, with $dt = 0.001$,$dx = 0.01$.

Example 3.4.4. Similar to previous examples, we consider the hyperbolic telegraph Eq. (3.1) with $\alpha = 1$, $\beta = 1$, and $f(x, t) = x^2 + t - 1$ in the interval $0 \leqslant x \leqslant 1$. The initial conditions are given by

$$\begin{cases} u(x, 0) = g_1(x) = x^2, & 0 \leqslant x \leqslant 1 \\ u_t(x, 0) = g_2(x) = 1, & 0 \leqslant x \leqslant 1 \end{cases} \tag{3.24}$$

and the exact solution is

$$u(x, t) = x^2 + t. \tag{3.25}$$

We extract the boundary function $h(x, t)$ from the exact solution. The L_∞ and L_2, errors and RMS of errors are obtained in Table V for $t = 1, 3, 5, 7$, and 10. The graph of analytical and estimated functions for $t = 10$ is given in Figure 7. We also draw the space-time graph of estimated solution in Figure 8.

Figure 3.6: Analytical and estimated function in $t = 10s$, with $dt = 0.001$ and $dx = 0.01$, for Example 4

Figure 3.7: Space-time graph of the solution up to $t = 10s$, with $dt = 0.001$ and $dx = 0.01$, for Example 4.

ABSTRACT

In this work, we introduce the radial basis functions and their types, and we applied this technique to solve the Laplace, Poisson and telegraph equations, these solutions are approximates solutions.

RESUME

Dans ce travail, on introduit la méthode de fonctions à base radiale et à leurs types, pour déterminer la solution approchée des équations différentielles aux dérivées partielles de Laplace, Poisson et de télégraphe.

الملخص

في هذا العمل، تناولنا مقدمة عن الوظائف ذات الأساس الشعاعي وأنواعها، لكي نقوم بحل بعض المعادلات التفاضلية ذات المشتقات الجزئية مثل معادلة لابلاس وبواسون و التلغراف، علما ان هذه الحلول هي حلول تقريبية لهاته المعادلات التفاضلية.

References

1. E. J. Kansa, Multiquadrics—a scattered data approximation scheme with applications to computational fluid dynamics-I, Comput Math Appl 19 (1990),.
2. M. Dehghan and A. Shokri, A numerical method for KdV equation using collocation and radial basis functions, Nonlinear Dyn
3. G. E. Fasshauer, Solving partial differential equations by collocation with radial basis functions, A. Le Méhauté, C. Rabut, and L. L. Schumaker, editors, Surface fitting and multiresolution methods, Vanderbilt University Press, Nashville TN, 1997
4. C. Franke and R. Schaback, Convergence orders of meshless collocation methods using radial basis functions, Adv Comput Math 8 (1997)
5. F. Bernal. Trust-region methods for nonlinear elliptic equations with radialbasis functions. *Comput. Math. Appl.*, 72(7):1743–1763, 2016
6. P. P. Chinchapatnam, K. Djidjeli, and P. B. Nair. Radial basis function meshless method for the steady incompressible Navier-Stokes equations. *Int.J. Comput. Math.*, 84(10):1509–1526, 2007
7. F. Gao and C. Chi, Unconditionally stable difference schemes for a one-space-dimensional linear hyperbolic equation, Appl Math Comput 187 (2007), 1272–1276.
8. R. K. Mohanty and M. K. Jain, An unconditionally stable alternating direction implicit scheme for the two space dimensional linear hyperbolic equation, Numer Meth Partial Differ Equat 17 (2001), 684–688.
9. R. K. Mohanty, M. K. Jain, and U. Arora, An unconditionally stable ADI method for the linear hyperbolic equation in three space dimensions, Int J Comput Math 79 (2002), 133–142.
10. R. C. Cascaval, E. C. Eckstein, C. L. Frota, and J. A. Goldstein, Fractional telegraph equations, J Math Anal Appl 276 (2002), 145–159.