Solution of Second Order Ordinary Differential Equation Associated with Toeplitz and Stiffness Matrices

1,2Samis H. Altoum

1AL-Qunfudhah University College, Umm Al-Qura University, KSA
2Department of Mathematics, Academy of Engineering Sciences, Sudan

Abstract: In this work we develop a technique solution of second order ordinary differential equation (integrating by parts) to reach to Toeplitz matrices and Stiffness matrix to solve O.D.Es. We investigate that solution numerically using finite difference method and we find the error between exact and numerical solution computed using finite element.

Keywords: Differential Equation, Toeplitz Matrix, Trapezoidal Method, Stiffness Matrix

Introduction

Ordinary differential equations frequently occur as mathematical models in many branches of science, physics, chemistry, biology, engineering and economy. In the finite element method for the numerical solution of elliptic partial differential equations, the stiffness matrix represents the system of linear equations. That must be solved in order to ascertain an approximate solution to the differential equation (Chniti et al., 2016; Gavin, 2014; Ye and Lim, 2015). The stiffness matrix for other PDE follows essentially the same procedure, but it can be complicated by the choice of boundary conditions. In order to implement the finite element method on a computer, one must first choose a set of basis functions and then compute the integrals defining the stiffness matrix. Usually, the domain Ω is discretized by some form of mesh generation, where it is divided into non-overlapping triangles or quadrilaterals, which are generally referred to as elements. The basis functions are then chosen to be polynomials of some order with in each element and continuous across element boundaries. The simplest choices are piecewise linear for triangular elements and piecewise bilinear for rectangular elements. In linear algebra, a Toeplitz matrix or diagonal-constant matrix, named after Otto Toeplitz, is a matrix in which each descending diagonal from left to right is constant. A matrix equation of the form \( Ax = b \) is called a Toeplitz system if \( A \) is a Toeplitz matrix. If \( A \) is an \( n \times n \) Toeplitz matrix, then the system has only 2n-1 degrees of freedom, rather than \( n^2 \). We might therefore expect that the solution of a Toeplitz system would be easier and indeed that is the case. Toeplitz systems can be solved by the Levinson algorithm in \( O(n^2) \) time. Variants of this algorithm have been shown to be weakly stable (i.e., they exhibit numerical stability for well-conditioned linear systems). The algorithm can also be used to find the determinant of a Toeplitz matrix in \( O(n^2) \) (mathematical notation that describes the limiting behaviour of a function) time (Saoudi et al., 2018; Agarwal and El-Sayed, 2018; Zhou and Agarwal, 2017; Ruzhansky et al., 2017). A Toeplitz matrix can also be decomposed (i.e., factored) in \( O(n^2) \) time. The Bareiss algorithm for an LU decomposition is stable. An LU decomposition (where 'LU' stands for 'lower upper' and also called LU factorization) gives a quick method for solving a Toeplitz system and also for computing the determinant. Algorithms that are asymptotically faster than those of Bareiss and Levinson have been described in the literature, but their accuracy cannot be relied upon (Cheb-Terrab et al., 1988; Altoum et al., 2017; Mukherjee and Maiti, 1988; Altoum, 2018a). In this study we introduce solution of second order ordinary differential equations. The fundamental idea based on Toeplitz matrices with absolutely summable elements and stiffness matrix. By limiting the generality of the matrices considered, the essential ideas and results can be conveyed in a more intuitive manner without the mathematical machinery required for the most general cases. As an application the results are applied to the study of the covariance matrices and their factors of linear models of discrete time random processes (Wazwaz, 2002; Altoum, 2018c; Gupta et al., 1995; Altoum, 2018b). In this study we consider the following equation:

\[
-\frac{d}{dx}\left(k(x)\frac{dT(x)}{dx}\right) = f(x), \quad -L < x < L
\]

(1)

\(k\): kernel conductivity of the material and \(f(x)\) is the heat source for simplicity we work with \(k(x) = k = \text{constant}\).
Weak Formulation

From Equation (1) we get:

\[ \int_{L}^{a} k \frac{dT(x)}{dx} \frac{dv(x)}{dx} \, dx = \int_{L}^{a} f(x) v(x) \, dx \]

(2)

where, \( v \) is a test function. Let \( v \) such that \( v(-L) = v(L) = 0 \); then integrating by parts, the left hand side of Equation (2) becomes:

\[ \int_{L}^{a} \frac{d}{dx} \left( k \frac{dT(x)}{dx} v(x) \right) \, dx = \left[ -k \frac{dT(x)}{dx} v(x) \right]_{L}^{a} - \int_{L}^{a} k \frac{dT(x)}{dx} \frac{dv(x)}{dx} \, dx \]

\[ \int_{L}^{a} -k \frac{dT(x)}{dx} \frac{dv(x)}{dx} \, dx = \int_{L}^{a} f(x) v(x) \, dx \]

Now the weak formulation is:

\[ \int_{L}^{a} k \frac{dT(x)}{dx} \frac{dv(x)}{dx} \, dx = \int_{L}^{a} f(x) v(x) \, dx \]

we look now for \( T \) and \( V_{h} = \text{Span}\{\phi_{j}\} \):

\[ \phi_{j}(x) = \begin{cases} \frac{x - x_{j-1}}{\Delta x_{j}}, & x_{j-1} \leq x \leq x_{j} \\ \frac{x_{j+1} - x}{\Delta x_{j}}, & x_{j} \leq x \leq x_{j+1} \\ 0, & \text{otherwise} \end{cases} \]

\( \Delta x_{j} = h \): uniform spacing:

\[ \phi_{j}(x) = \begin{cases} \frac{1}{h}, & x_{j-1} \leq x \leq x_{j} \\ 0, & \text{otherwise} \end{cases} \]

and it is derivative is given by:

\[ \phi'_{j}(x) = \begin{cases} 1, & x_{j-1} \leq x \leq x_{j} \\ -1, & x_{j} \leq x \leq x_{j+1} \\ 0, & \text{otherwise} \end{cases} \]

Let \( v = \phi_{j}(x) \) then we have:

\[ \int_{L}^{a} k \frac{dT(x)}{dx} \frac{d\phi_{j}(x)}{dx} \, dx = \int_{L}^{a} f(x) \phi_{j}(x) \, dx \]

\[ \bar{T} = T \]

(3)

where, \( \bar{T} \) is the approximate solution. Then Equation (3) becomes:

\[ \int_{L}^{a} k \frac{d\bar{T}(x)}{dx} \frac{d\phi_{j}(x)}{dx} \, dx + \int_{L}^{a} k \frac{d\bar{T}(x)}{dx} \frac{d\phi_{j}(x)}{dx} \, dx \]

\[ = \int_{L}^{a} f(x) \phi_{j}(x) \, dx \]

\[ \frac{1}{h} \int_{L}^{a} k \frac{d\bar{T}(x)}{dx} \, dx = \frac{1}{h} \int_{L}^{a} k \frac{d\bar{T}(x)}{dx} \, dx - \frac{1}{h} \int_{L}^{a} k \frac{d\bar{T}(x)}{dx} \, dx = \int_{L}^{a} f(x) \phi_{j}(x) \, dx \]

\( N \): linear elements and \( N + 1 \) degree of freedom and therefore \( N + 1 \) basis functions:

\[ \bar{T}(x) = \sum_{j=1}^{N+1} a_{j} \phi_{j}(x) \]

(4)

we have:

\[ \int_{L}^{a} k \frac{d\bar{T}(x)}{dx} \frac{d\phi_{j}(x)}{dx} \, dx = \frac{1}{h} \int_{L}^{a} k \frac{d\bar{T}(x)}{dx} \, dx - \frac{1}{h} \int_{L}^{a} k \frac{d\bar{T}(x)}{dx} \, dx \]

using:

\[ \frac{d\bar{T}(x)}{dx} = \sum_{j=1}^{N+1} a_{j} \frac{d\phi_{j}(x)}{dx} \]

we only use:

\[ \sum_{j=1}^{N+1} a_{j} \frac{d\phi_{j}(x)}{dx} = a_{j-1} \frac{d\phi_{j}(x)}{dx} + a_{j} \frac{d\phi_{j}(x)}{dx} \]

and:

\[ \sum_{j=1}^{N+1} a_{j} \frac{d\phi_{j}(x)}{dx} = a_{j-1} \frac{d\phi_{j}(x)}{dx} + a_{j} \frac{d\phi_{j}(x)}{dx} \]

\[ \int_{L}^{a} k \frac{d\bar{T}(x)}{dx} \frac{d\phi_{j}(x)}{dx} \, dx = \frac{k}{h} \int_{L}^{a} \frac{d\bar{T}(x)}{dx} \, dx + a_{j-1} \frac{d\phi_{j}(x)}{dx} \, dx \]

\[ = \frac{k}{h} \int_{L}^{a} \frac{d\bar{T}(x)}{dx} \, dx + a_{j-1} \frac{d\phi_{j}(x)}{dx} \, dx \]

\[ = \frac{k}{h} \int_{L}^{a} \frac{d\bar{T}(x)}{dx} \, dx + a_{j+1} \frac{d\phi_{j}(x)}{dx} \, dx \]

\[ = \frac{k}{h} \int_{L}^{a} \frac{d\bar{T}(x)}{dx} \, dx + a_{j+1} \frac{d\phi_{j}(x)}{dx} \, dx \]

\[ = \frac{k}{h} \int_{L}^{a} \frac{d\bar{T}(x)}{dx} \, dx + a_{j-1} \frac{d\phi_{j}(x)}{dx} \, dx \]

\[ = \frac{k}{h} \int_{L}^{a} \frac{d\bar{T}(x)}{dx} \, dx + a_{j-1} \frac{d\phi_{j}(x)}{dx} \, dx \]

we conclude the following:
\[
\frac{k}{h} \left[2a_j - a_{j-1} - a_{j+1}\right] = \int_{x_{j-1}}^{x_j} f(x) \delta_j(x) dx \tag{6}
\]

for given function \(f\) we compute this part of our problem. Now let \(j = 1, \ldots, N\) our unknowns: \(a_1, a_2, \ldots, a_{N+1}\).

\[
T(x) = \sum_{j=1}^{N+1} a_j \delta_j(x)
\]

\[
\begin{bmatrix}
2 & 1 & 0 & 0 & 0 & \cdots & 0 & a_1 \\
1 & 2 & 1 & 0 & 0 & \cdots & 0 & a_2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & -1 & 2 & 1 & 0 & a_{N+1}
\end{bmatrix}
\begin{bmatrix}
L(\phi_1) \\
L(\phi_2) \\
\vdots \\
L(\phi_{N+1})
\end{bmatrix}
\]

where: the left matrix is Stiffness matrix and \(L(\phi_j) = \int_{x_{j-1}}^{x_j} f(x) \phi_j(x) dx\).

**Example 1**

Let \(L = 1\), \(x \in [0,1]\) and \(f(x) = 50e^x\), \(T(-1) = 100\), \(T(1) = 100\) boundary condition \(\tilde{T} = 100\), we will compute the value of \(T\) at each nodes: \(x(1) = -1, \ldots, x(N+1) = 1\):

\[
\tilde{T} = \begin{bmatrix} 1 & 2 & \cdots & N & N+1 \end{bmatrix}
\]

where, \(\tilde{T}\): vector \(T(1)\) and \(T(N+1)\) are knowns, so we will determine only \(\tilde{T}(2), \ldots, \tilde{T}(N)\) corresponding to the value of \(T\) at nodes \(x(2), \ldots, x(N)\), using \(j = 2, \ldots , N\):

\[
\frac{k}{h} \left[2a_j - a_{j-1} - a_{j+1}\right] = L(\phi_j)
\]

\[
\begin{bmatrix}
2 & 1 & 0 & 0 & 0 & \cdots & 0 & a_1 \\
1 & 2 & 1 & 0 & 0 & \cdots & 0 & a_2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \cdots & -1 & 2 & 1 & 0 & a_{N+1}
\end{bmatrix}
\begin{bmatrix}
L(\phi_1) \\
L(\phi_2) \\
\vdots \\
L(\phi_{N+1})
\end{bmatrix}
\]

\[
\frac{k}{h} A Y = b
\]

where:

\[
\begin{bmatrix}
\frac{k}{h} a_1 \\
0 \\
\vdots \\
0 \\
\frac{k}{h} a_{N+1}
\end{bmatrix} +
\begin{bmatrix}
L(\phi_1) \\
L(\phi_2) \\
\vdots \\
L(\phi_{N+1})
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_2 \\
\vdots \\
a_{N+1}
\end{bmatrix}
\]

In matlab, we can calculate \(A\):

\[
A = \text{Toeplitz}([2,-1,0,\ldots,0],[2,-1,0,\ldots,0])
\]

size of \(A = N\) because we have \(a_2, \ldots, a_N\) unknowns, so how many zeros Toeplitz matrix we have \((N-1)-(2) = N-3\), because we have two known coefficients 2 and 1:

\[
A = \text{Toeplitz}([2,-1,\text{zeros}(1,N-3)],[2,-1,\text{zeros}(1,N-3)])
\]

Boundary condition:

\[
\begin{bmatrix}
a_1 \\
0 \\
\vdots \\
0 \\
a_{N+1}
\end{bmatrix}
\begin{bmatrix}
T(1,1) \\
T(1,2) \\
\vdots \\
T(N+1,1)
\end{bmatrix}
\]

nodes = \([T(1,1); \text{zero}(N-3,1); T(N+1,1)]; \text{L}(a)\) must computed using integration by parts or Trapezoidal method.

First method: integration by parts:

\[
L(\phi_j) = \int_{x_{j-1}}^{x_j} f(x) \delta_j(x) dx
\]

\[
= \int_{x_{j-1}}^{x_j} f(x) \phi_j(x) dx + \int_{x_{j-1}}^{x_j} f(x) \phi_j(x) dx
\]

\[
= \int_{x_{j-1}}^{x_j} 50e^{x-x_{j-1}} + \int_{x_{j-1}}^{x_j} 50e^{x_{j+1}-x} \frac{dx}{\Delta x}
\]

\[
= \int_{x_{j-1}}^{x_j} e^{x-x_{j-1}} - 2e^{x-x_{j-1}} + e^{x_{j+1}-x} + e^{x_{j+1}-x}
\]

\[
= \int_{x_{j-1}}^{x_j} 2fx_J + \frac{fx_{j+1}}{h}
\]

Second method: We can use direct approximation of the integral without integral by parts (Trapezoidal):

\[
\int f(x) = \frac{b-a}{2} (f(a) + f(b))
\]

\[
\int f(x) \frac{x-x_{j-1}}{\Delta x} = \frac{f(x_j)\Delta x + f(x_{j+1})\Delta x}{2} (x-x_{j-1})
\]

and:

\[
\frac{\Delta x}{2} f(x_j)
\]
\[
\int_{x_{0,1}}^{x_{N,1}} f(x) \frac{x_{j+1} - x_j}{\Delta x} dx = \frac{\Delta x}{2} \left[ f(x_{j+1}) \frac{x_{j+1} - x_{j+1}}{\Delta x} + f(x_j) \frac{x_{j+1} - x_j}{\Delta x} \right] \\
= \frac{\Delta x}{2} \left[ 0 + \frac{\Delta x}{\Delta x} f(x) \right] \\
= \frac{\Delta x}{2} f(x_j).
\]

Then:
\[
\int_{x_{0,1}}^{x_{N,1}} f(x) \phi_j(x) dx = \frac{\Delta x}{2} f(x_j) \frac{\Delta x}{2} f(x_j).
\]

Conclusion second approximation of the integral:
\[
L(\phi_j) = \left\{ \frac{1}{h} \left[ f(x_{j+1}) - 2f(x_j) + f(x_{j-1}) \right], \text{ integrating by parts} \right\}
\]
\[
\frac{h}{2} f(x_j), \text{ Trapezoidal rule}
\]

Finally:
\[
T = \left[ T(1,1), T(2,1), T(3,1), \ldots, T(N,1), T(N+1,1) \right]
\]

where, the values of \( N \) at \( T(1,1) \) and \( T(N+1,1) \) are knows the value of \( N \) at \( T(1,1) = -L = x_1 \) and \( T(N+1) = L = x_{N+1} \) respectively. We need only calculate \( T(2,1), \ldots, T(N,1) \) where:
\[
T(2:N,1) = \left( \frac{h}{2} \right)^4 b.
\]

**Numerical Results**

To certify the proposed method, we consider the case where the data \( f(x) \) is the following:
\[
f(x) = -50e^x + 50x \sinh(1) + 50 \cosh(1) + 100
\]

Figure 1 presents the behavior the exact and numerical solution. Whereas Fig. 2 present the error between exact and numerical solution computed using finite element. The error is defined as: \( \text{error} = |u_{\text{analytical}} - u_{\text{numerical}}| \), for a fixed value of \( N \), we get an error about \( 10^{-3} \). This confirm that the proposed method converge well with error about \( 10^{-3} \). Her, we limit ourselves to numerical convergence, Fig. 1-4 present the behavior of error for different value of \( N \), its noticed that the error is sharp decreasing versus \( N \).

![Fig. 1: Analytical solution and numerical solution using finite element](image-url)
Fig. 2: \( \text{Error} = |u_{\text{analytical}} - u_{\text{numerical}}| \)

Fig. 3: Behavior of error for different value of \( N \)

\( \times 10^{-3} \)
Discussion and Conclusion

The strategy proposed using Stiffness and Toeplitz matrices is more flexible and more faster computed to other methods, i.e., Finite element. The structure of the system obtained (Toeplitz matrix) permit to use faster matrix vector product. This method will be extended and applied for other equation, i.e. Diffusion problem in two dimensional case. In practice, we will obtain a very large system with Toeplitz structure. This will be solved using the main properties of the Toeplitz matrix. Finally, this paper is one of a list of papers that will be published to show the importance of Toeplitz matrices to get the system.

Acknowledgement

We would like to thanks anonymous referees for their careful reading of our manuscript and helpful suggestions.

Ethics

The author declare that there is no conict interests regarding the publication of this manuscript. This article is original and contains unpublished material.

References


Fig. 4: $\log_{10}(\text{error})$, for different value of $\mathbf{N}$


