Adaptive Finite Element Method for Fredholm Integro-Differential Equation

E. Adel¹, A. Elsaid¹*, M. El-Agamy¹

¹Mathematics & Engineering Physics Department, Faculty of Engineering, Mansoura University, PO 35516, Mansoura, Egypt.

Abstract
In this work, we apply an adaptive finite element method for solving Fredholm integro-differential equations. The adaptive refinement is based on a recovery-type error estimator. The results of some numerical examples are shown to illustrate that prescribed accuracy is achieved with minimum degrees of freedom.

Keywords: Adaptive finite element, Solution recovery, Fredholm integro-differential equation.

1 Introduction

An integro-differential equation (IDE) is an equation in which differential and integral operators appear at the same time. The mathematical formulations in many applied areas of sciences and engineering contain integro-differential equations. These include the study of fluid, physics, chemistry, biology, mechanics, astronomy, potential theory, electrostatics and chemical kinetics [1]-[3]. Also some partial differential equations are converted into IDEs as in [4], where a steady state diffusion equation which contains two unknowns is replaced by a single integro-differential equation containing only one unknown.

IDEs with constant limits of integration are called Fredholm IDEs. There are varieties of numerical methods that have been developed to solve Fredholm IDEs. Some of these methods are direct method based on Fourier and block-pulse functions [5], compact finite difference method [6], Taylor and Chebyshev matrix methods [7]-[10], an extrapolation method [11], method of regularization [12], Adomian decomposition method [13], and series solution method [14]. One of the most powerful methods utilized to solve IDEs is the finite element method (FEM).

FEMs are numerical methods that successfully obtain good approximate solutions to different types of equations. FEMs have many subclasses where the CO-Galerkin FEM is considered the simplest of them. One of the main features that differ FEMs from other numerical methods is the combination of variational methods and piece-wise polynomial approximation. Also, FEMs deal with complicated domains with stability and flexibility. FEMs, as they are known today, were introduced in 1960 [15]. Different forms of IDEs were solved using FEMs (see for example [16]-[23]).

Adaptive procedures for the FEM approximate solutions are important techniques that enhance the accuracy and the efficiency of finite element solutions. They perform refinement only in parts of the domain where the error in the solution is large by using error estimators. A priori and a posteriori error estimates for the optimal control problems

*Corresponding author. Email address: a.elsaid@mans.edu.eg
governed by integral or IDEs of Fredholm type were established in [23].

In this article, an adaptive FEM procedure is proposed to solve Fredholm IDE of the second kind of the following form

\[ \lambda u_{xx}(x) - \int_{a}^{b} K(x,t)u(t)dt = f(x), \quad x \in [a,b], \lambda \neq 0, \]  

(1.1)

subject to boundary conditions: \( u(a) = d_1 \) and \( u(b) = d_2 \). In this adaptive procedure, a recovery is constructed for the FEM solution using superconvergent patch recovery (SPR) technique. The mesh is refined to obtain higher accuracy solution till the error calculated between the FEM solution and the recovered solution reaches a prescribed tolerance.

2 Finite element solution

At first, the weighted residual is formed as

\[ \int Rw \, dx = 0, \]  

(2.2)

where \( R \) denotes the residual which is expressed as

\[ R(x) = \lambda u_{xx}(x) - \int_{a}^{b} K(x,t)u(t)dt - f(x), \]  

(2.3)

where \( u \) is the approximate solution and \( w \) is a weighting function to be chosen. Substituting (2.3) in (2.2) we get

\[ \int_{a}^{b} \lambda u_{xx}(x)wdx - \int_{a}^{b} \int_{a}^{b} K(x,t)u(t)wdtdx = \int_{a}^{b} f(x)wdx. \]  

(2.4)

In this step, the derivative inside the integral is reduced from second order to first order by integrating by parts. Thus (2.4) becomes

\[ -\int_{a}^{b} \lambda u_{x}(x)w_{x}dx - \int_{a}^{b} \int_{a}^{b} K(x,t)u(t)wdtdx = \int_{a}^{b} f(x)wdx - \lambda wu_{x}|_{a}^{b}. \]  

(2.5)

Now, we divide the domain into finite number of elements and let the FEM approximate solution \( u_{h} \) be given in the form

\[ u_{h}(x) = \sum_{j=1}^{m} c_{j} \varphi_{j}(x), \quad x \in [a,b], \]  

(2.6)

where \( \varphi_{j}(x) \) are the basis functions, \( c_{j} \) are the coefficients to be determined and \( h \) denotes the maximum subinterval length.

Choosing \( w_{i}(x) = \varphi_{i}(x) \) (The Galerkin assumption), so that the residual is kept orthogonal to the space of functions used in the approximation of \( u \). By substituting (2.6) in (2.5) and rearranging, we get

\[ -\sum_{j=1}^{m} c_{j} \left[ \lambda \int_{a}^{b} \varphi'_{j}(x)\varphi'_{j}(x)dx + \int_{a}^{b} \int_{a}^{b} K(x,t)\varphi_{j}(t)\varphi_{j}(x)dtdx \right] = \int_{a}^{b} f(x)\varphi_{i}(x)dx - \left[ \lambda \varphi_{i} \sum_{j=1}^{m} c_{j} \varphi'_{j}(x) \right]_{a}^{b}. \]  

(2.7)

We obtain a system of equations which has the following matrix form

\[ (E + K)C = F, \]  

(2.8)
where
\[
E = \begin{bmatrix}
E_{11} & \cdots & E_{1m} \\
E_{21} & \cdots & E_{2m} \\
\vdots & \cdots & \vdots \\
E_{m1} & \cdots & E_{mm}
\end{bmatrix}, \quad E_{ij} = -\lambda \int_a^b \varphi_j'(t) \varphi_i(x) \, dt, \quad (2.9)
\]
\[
K = \begin{bmatrix}
K_{11} & \cdots & K_{1m} \\
K_{21} & \cdots & K_{2m} \\
\vdots & \cdots & \vdots \\
K_{m1} & \cdots & K_{mm}
\end{bmatrix}, \quad K_{ij} = -\int_a^a K(x,t) \varphi_j(t) \varphi_i(x) \, dt, \quad (2.10)
\]
\[
F = \begin{bmatrix}
F_1 \\
F_2 \\
\vdots \\
F_m
\end{bmatrix}^T, \quad (2.11)
\]
\[
F_i = \int_a^b f(x) \varphi_i(x) \, dx + \left[ \lambda \sum_{j=1}^m c_j \varphi_j'(x) \right]_{x=a}, \quad (2.12)
\]
\[
F_m = \int_a^b f(x) \varphi_m(x) \, dx - \left[ \lambda \sum_{j=1}^m c_j \varphi_j'(x) \right]_{x=b}, \quad (2.13)
\]
\[
F_i = \int_a^b f(x) \varphi_i(x) \, dx, \quad i = 2,3,\ldots,m-1, \quad (2.14)
\]
\[
C = \begin{bmatrix}
c_1 \\
c_2 \\
\vdots \\
c_m
\end{bmatrix}^T. \quad (2.15)
\]

Applying boundary conditions gives directly the first and last nodal values: i.e., \(u_1 = d_1\) and \(u_m = d_2\). These Dirichlet boundary conditions replace the first and last rows in the global system of equations (2.8). Solving these equations for \(C\), the approximate solution \(u_h(x)\) is obtained.

### 3 Solution recovery

We are concerned with the matter of enhancing the accuracy of the approximate solution by a posteriori treatment of the data. Such process is referred to as recovery technique and the error estimator is classified as a recovery-based error estimator. Here, the SPR technique is utilized to produce superconvergent displacements. We use the recovery procedure presented in [24]. This procedure differs from the general gradient recovery technique proposed in [25] and [26] in two things. Firstly, recovery is made for the whole element instead of each node and an element patch is used instead of a node patch. Secondly, the local error is calculated directly over the element from the recovered solution for this element.

First, the element patch is constructed which consists of the current element \(e\) and the adjacent elements sharing common nodes with it. For each element, the recovered solution \(R_h u_h\) is expressed as
\[
R_h u_h = Q(x)a, \quad (3.16)
\]
this is a single and continuous polynomial representation of the function describing the solution over the element where \(a\) denotes the unknown parameters and \(Q(x)\) is the basis of the assumed polynomial for \(R_h u_h\). Let the superconvergent finite element solutions in this patch be \(S_h\) and the number of \(S_h\) be \(n_s\). To determine the parameters \(a\), the expansion of \(R_h u_h\) is made to fit the superconvergent points on the element patch in a least squares manner. Thus we have to minimize the functional
\[
F(a) = \sum_{i=1}^{n_s} (S_h(x) - Q(x)a)^2. \quad (3.17)
\]
This leads to
\[ a = A^{-1}b, \] (3.18)
where
\[ A = \sum_{i=1}^{ns} Q^T(x)Q(x), \quad b = \sum_{i=1}^{ns} Q^T(x)S_h(x). \] (3.19)

After calculating the parameters \( a \), the local error (over the element \( e \)) is calculated directly from the recovered solution.

One of the major applications of the recovery methods is its use in the calculation of \textit{a posteriori} error estimators. Having the recovered solutions, the error can now be evaluated simply by replacing the exact value of \( u \), which is in general unknown, by the recovered value \( R_hu_h \) which is better than the direct finite element solution. Here we choose the \( L_2 \) norm so the error estimator \( \varepsilon_h \) can be written as follows
\[ \|\varepsilon_h\|_{L_2} = \|R_hu_h - u_h\|_{L_2} \] (3.20)

To ensure the accuracy and the quality of the error estimator, the effectivity index \( \theta \) is calculated. The effectivity index is the ration between the recovery error estimator and the exact error \( \varepsilon_h \), i.e.
\[ \theta = \frac{\|\varepsilon_h\|}{\|\varepsilon_h\|} \] (3.21)

The error estimator is said to be asymptotically exact if \( \theta \) approaches unity as the mesh partition size approaches zero.

4 Adaptive procedure

In adaptive techniques, iterative procedures are performed starting with an initial mesh until a condition that involves a predefined tolerance \( \varepsilon \) is satisfied. In this work, the SPR technique is applied at every iteration and the local error \( \varepsilon_{h,e} \) between the recovered solution and the finite element solution is calculated for every element. The maximum of all local errors is checked to see if it is less than the predefined tolerance \( \varepsilon \). If this is the case, the adaptive procedure is put to an end. Otherwise, we mark some elements to be refined, and repeat the adaptive procedure. The marking strategy also depends on the local error \( \varepsilon_h \) at each element as follows. A ratio \( \tau \) is to chosen such that \( 0 < \tau < 1 \). If \( (\varepsilon_{h,e_k} > \tau \times (\max_i(\varepsilon_{h,e})) \), the \( k \)th element is marked for refinement. The algorithm used is summarized in the following steps:

1. Define the initial mesh and the desired tolerance \( \varepsilon \).
2. Calculate the finite element solution \( u_h \).
3. Apply solution recovery using SPR and determine the local error \( \varepsilon_{h,e} \) for each element.
4. If \( (\max_i(\varepsilon_{h,e_i}) \leq \varepsilon) \) then stop
   else
   If \( (\varepsilon_{h,e_k} > \tau \times (\max_i(\varepsilon_{h,e}) \)) \) then mark element \( k \), refine marked elements and go back to step 2.

5 Numerical examples

The following examples are solved by FEM using quadratic element approximation then we use fourth order polynomial for the recovered solution. The tolerance is chosen as \( \varepsilon = 10^{-6} \).

Example 5.1. Consider the following IDE
\[ u_{xx}(x) = \int_0^1 \tanh(10x)\tanh(10t)u(t)dt + f(x), \] (5.22)
where \( f(x) \) is chosen such that the exact solution is given by

\[
    u(x) = \operatorname{sech}^2(10x).
\]  
(5.23)

Figure 1: Exact solution versus recovered solution for Example 5.1.

Figure 2: Final mesh of Example 5.1 at Dofs = 67.
Figure 3: Effectivity index for Example 5.1.

Figure 1 shows the good agreement between the exact solution and the recovered solution. It also indicates that this accuracy is achieved with minimum degrees of freedom (DOFs) as the refinement is carried out only at the parts of the domain where the solution exhibits fast changes. This is confirmed by the graph of final mesh in Figure 2. Figure 3 shows that the effectivity index approaches unity after several iterations.

Example 5.2. Consider the following IDE

\[ u_{xx}(x) = \int_0^1 e^{xt} u(t) dt + f(x), \quad 0 < x < 1, \]

where \( f(x) \) is chosen such that the exact solution is given by

\[ u(x) = e^{-7x} \cos(15x). \]

Figure 4: Exact solution versus recovered solution for Example 5.2.
Figure 5: Final mesh for Example 5.2 at Dofs = 143.

Figure 6: Effectivity index for Example 5.2.

Figure 4 and Figure 5 confirm that good agreement between the exact solution and the recovered solution is obtained with minimum DOFs and that the refinement is performed only at parts of the domain where the solution exhibits fast changes. Figure 6 shows that the effectivity index approaches unity after several iterations.

6 Conclusion

An adaptive finite element scheme is presented to solve Fredholm IDEs. The scheme is based on using a recovery based error estimator to choose the parts of the domain to be refined. The recovery technique utilized is a function recovery type which is based on SPR technique. The numerical examples illustrate the advantage of this scheme as the it is successful in marking the elements where the solution exhibits fast changes and refine them only, hence achieving high accuracy with minimum DOFs.
References


