Thai Journal of Mathematics
Volume 18 Number 2 (2020)
Pages 751–763

http://thaijmath.in.cmu.ac.th

ISSN 1686-0209

Multistage Integral Mean Value Method for the Fredholm Integral Equations of the Second Kind

Ahmad Molabahrami[1,*] and Sakineh Niasadegh[2]

[1] *Department of Mathematics, Ilam University, PO Box 69315516, Ilam, Iran*
e-mail : a.molabahrami@ilam.ac.ir

[2] *Islamic Azad University of Hamedan Branch, PO Box 65138-734, Hamedan, Iran*
e-mail : s.niyasadegh@yahoo.com

Abstract In this paper, a modified multistage integral mean value method, for handling the Fredholm integral equations of the second kind, to improve the accuracy of the solutions, is applied. The application of the proposed algorithm is based on the applying the multistage schema to the modified integral mean value method. Also, the equivalency of integral mean value method and degenerate kernel method (DKM) is established. The efficiency of the approach will be shown by applying the procedure on some prototype examples. The MATHEMATICA programs based on the procedures in this paper are designed.

MSC: 45Fxx; 45G10; 35C10

Keywords: second kind Fredholm integral equations; integral mean value method; modified multistage integral mean value method; degenerate kernel method; non-degenerate kernels; approximate degenerate kernel

Submission date: 28.09.2017 / Acceptance date: 16.08.2019

1. Introduction

The solutions of integral equations have a major role in the fields of science and engineering. A physical event can be modeled by the differential equation, an integral equation (IE) or an integro-differential equation (IDE) or a system of these. Several numerical and analytical methods were used such as the successive approximation method that some of them were mentioned in [1]. Fredholm integral equations of the second kind (FIE2s) are of the form [1, 2]

$$u(x) = f(x) + \lambda \int_a^b K(x,t)F(u(t))dt, x \in [c,d], \tag{1.1}$$

where $F(u(x))$ is a function of $u(x)$, λ is a parameter, $f(x)$ is the data function, $K(x,t)$ is the kernel of the integral equation, and $u(x)$ is the unknown function that will be determined. For the linear case, it is assumed that $F(u(x)) = u(x)$.

*Corresponding author.

Published by The Mathematical Association of Thailand.

Recently, a new method, namely integral mean value method (IMVM), for handling the FIE2s, proposed by Loghmani et al [3, 4]. This method is based on the mean value theorem for integrals. In this study, we propose a modification of IMVM and a multistage schema of modifed IMVM, namely modified multistage integral mean value method (MIMVM), to improve the accuracy of the solutions. To achieve about mentioned modification, we give an approach of IMVM to reduce the order of the corresponding algebraic system and we show that IMVM and DKM are equivalent. In the frame of the DKM, IMVM and MIMVM, for a non-degenerate kernel, we use a degenerate approximation of the kernel by using Taylor series and Lagrange interpolation.

The remainder of present paper is organized as follows. In Section 2, DKM, traditional IMVM/MIMVM and modified IMVM/MIMVM are described. Also, in this section, we show the equivalency of modified IMVM and DKM. Section 3 gives some prototype examples. Section 4 provides some comparisons of the obtained results with some selected problems available in the literature. Finally, Section 5 appears our conclusions.

2. Analysis of the Methods

In this section, we first give a brief review of the DKM, then we show equivalency of DKM and IMVM and finally we propose a multistage schema of the integral mean value method. Assume that the kernel of Eq. (1.1) is of the form

$$K(x,t) = \sum_{j=1}^{m} g_j(x)h_j(t). \tag{2.1}$$

For a non-degenerate kernel, a degenerate approximation is used as (2.1).

2.1. A Brief Review of DKM

The degenerate kernel method (DKM) is a well-known classical method for solving Fredholm integral equations of the second kind, and it is one of the easiest numerical methods to define and analyze [5, page 23]. This method for a given degenerate kernel is called direct computation method (DCM) [6] and [7, page 141].

DKM transforms an integral equation of the second kind, with a degenerate kernel, to a system of algebraic equations. To handle the Eq. (1.1), by using the DKM, we can express the procedure as follows:

1. Substituting (2.1) into (1.1) gives

$$u(x;\alpha) = f(x) + \lambda \sum_{j=1}^{m} \alpha_j g_j(x), \tag{2.2}$$

where

$$\alpha_k = \int_a^b h_k(t)F(u(t))dt, k = 1, 2, ..., m. \tag{2.3}$$

2. Replacing Eq. (2.2) into (2.3) leads to the following algebraic equation/system

$$\alpha_k = \int_a^b h_k(t)F\left(f(t) + \lambda \sum_{j=1}^{m} \alpha_j g_j(t)\right) dt, k = 1, 2, ..., m. \tag{2.4}$$

3. Solving the Eq. (2.4) provides the values of α_k, $k = 1, 2, ...m$, for substituting them into the Eq. (2.2) to obtain the solution of Eq. (1.1).

2.2. A Brief Review of IMVM

In this subsection, considering the discussion of Section 3 of [8], we first give a brief review of traditional IMVM, and then we propose a modification of IMVM to show the equivalency of this method with DKM. From Eqs. (1.1) and (2.1) we find that

$$u(x) = f(x) + \lambda \sum_{i=1}^{m} g_i(x) \int_a^b h_i(t) F(u(t)) \mathrm{d}t. \tag{2.5}$$

Applying the one dimensional mean value theorem for Eq. (2.5) yields

$$u(x) = f(x) + \lambda \sum_{i=1}^{m} g_i(x)(b-a) h_i(c_i) F(u(c_i)). \tag{2.6}$$

Using the traditional IMVM, we find the following algebraic system

$$\begin{cases} u(c_j) = f(c_j) + \lambda \sum\limits_{i=1}^{m} g_i(c_j)(b-a) h_i(c_i) F(u(c_i)), \\ u(c_j) = f(c_j) + \lambda \sum\limits_{i=1}^{m} g_i(c_j) \int_a^b h_i(t) F\left(f(t) + \lambda \sum\limits_{r=1}^{m} g_r(t)(b-a) h_r(c_r) F(u(c_r)) \right) \mathrm{d}t. \end{cases} \tag{2.7}$$

Solving algebraic system (2.7) provides the values of c_i and $u(c_i)$, $i = 1, 2, ..., m$, for substituting them into the Eq. (2.6) to obtain the solution/solutions of Eq. (2.5). Now, we propose a modification of IMVM. From (2.7) we find that

$$\begin{cases} u(c_j) = f(c_j) + \lambda \sum\limits_{i=1}^{m} g_i(c_j)(b-a) h_i(c_i) F(u(c_i)), \\ (b-a) h_i(c_i) F(u(c_i)) = \int_a^b h_i(t) F\left(f(t) + \lambda \sum\limits_{r=1}^{m} g_r(t)(b-a) h_r(c_r) F(u(c_r)) \right) \mathrm{d}t, \end{cases} \tag{2.8}$$

where $j = 1, 2, ..., m$. Now, by assuming $\alpha_j = (b-a) h_j(c_j) F(u(c_j))$, $j = 1, 2, ..., m$, Eqs. (2.6) and (2.8) become as follows:

$$u(x; \alpha) = f(x) + \lambda \sum_{i=1}^{m} g_i(x) \alpha_i, \tag{2.9}$$

and

$$\begin{cases} u(c_j) = f(c_j) + \lambda \sum\limits_{i=1}^{m} g_i(c_j)(b-a) h_i(c_i) F(u(c_i)), j = 1, 2, ..., m, \\ \alpha_i = \int_a^b h_i(t) F\left(f(t) + \lambda \sum\limits_{r=1}^{m} g_r(t) \alpha_r \right) \mathrm{d}t, i = 1, 2, ..., m, \end{cases} \tag{2.10}$$

respectively. According to the (2.9), there is no need to calculate $(c_i, u(c_i))$, $i = 1, 2, ..., m$, then, the first mth equations of Eq. (2.10) are extra and therefore the $2m \times 2m$ algebraic system given by Eq. (2.10) reduces to the following $m \times m$ algebraic system

$$\alpha_i = \int_a^b h_i(t) F\left(f(t) + \lambda \sum_{r=1}^{m} g_r(t) \alpha_r \right) \mathrm{d}t, i = 1, 2, ..., m. \tag{2.11}$$

It is clear that (2.9) and (2.11) are exactly the same results of applying the DKM for Eq. (2.5) ((2.9) and (2.11) are exactly (2.2) and (2.4) respectively), then DKM and MIVM are equivalent.

2.3. Multistage Integral Mean Value Method

In multistage schema,we often can make a more accurate approximation by breaking up the integral region, interval $[a, b]$, into some number n of subintervals, then applying the mean value theorem for each sub-integral. Bearing this in mind and using the results of pervious subsection, for handling the Eq. (1.1), the modified MIMVM, structured on modified IMVM can be applied as follows.

1. Breaking up the integral region, the Eq. (2.5) yields

$$u(x) = f(x) + \lambda \sum_{j=1}^{m} g_j(x) \sum_{i=1}^{n} \int_{a_{i-1}}^{a_i} h_j(t)F(u(t))dt, x \in [c, d], \tag{2.12}$$

   where $a_0 = a$, $a_i = a_0 + ih$ and $h = \frac{b-a}{n}$.
2. Applying the mean value theorem for each integral in Eq. (2.12), gives

$$u(x) = f(x) + \lambda \sum_{j=1}^{m} g_j(x)h \sum_{i=1}^{n} h_j(c_i)F(u(c_i)), \tag{2.13}$$

   where $c_i \in [a_{i-1}, a_i]$. Assume $\alpha_{ij} = hh_j(c_i)F(u(c_i))$, $i = 1, 2, ..., n$, $j = 1, 2, ..., m$, (2.13) leads to

$$u(x; \alpha) = f(x) + \lambda \sum_{j=1}^{m} \sum_{i=1}^{n} \alpha_{ij} g_j(x). \tag{2.14}$$

3. From (2.12) and (2.14) we obtain the following algebraic system

$$\alpha_{ij} = \int_{a_{i-1}}^{a_i} h_j(t)F\left(f(t) + \lambda \sum_{r=1}^{m} \sum_{k=1}^{n} \alpha_{kr} g_r(t)\right) dt. \tag{2.15}$$

4. Solving the Eq. (2.15) provides the values of α_{ij}, $i = 1, 2, ..., n$, $j = 1, 2, ..., m$, for substituting them into the Eq. (2.14) to obtain the solution/solutions of Eq. (2.5). For this respect, we first try to find the exact solution/solutions of the algebraic system Eqs. (2.15), by using a solver tool of Mathematica, namely Solve/Reduce. However, the Newton's method is used to obtain the numerical solution/solutions of the above mentioned algebraic system.

Remark 2.1. For the traditional MIMVM (2.14) and (2.15) become as follows:

$$u(x) = f(x) + \lambda h \sum_{j=1}^{m} \sum_{i=1}^{n} g_j(x)(b - a)h_j(c_{i,j})F(u(c_{i,j})). \tag{2.16}$$

Using the traditional IMVM proses, we find the following algebraic system

$$\begin{cases} u(c_{r,k}) = f(c_{r,k}) + \lambda h \sum\limits_{j=1}^{m} \sum\limits_{i=1}^{n} g_j(c_{r,k})(b - a)h_j(c_{i,j})F(u(c_{i,j})), \\ u(c_{r,k}) = f(c_{r,k}) + \lambda \sum\limits_{i=1}^{m} g_i(c_{r,k}) \\ \int_a^b h_i(t)F(f(t) + \lambda h \sum\limits_{l=1}^{m} \sum\limits_{p=1}^{n} g_l(t)(b - a)h_l(c_{p,l})F(u(c_{p,l})))\mathrm{d}t, \end{cases} \tag{2.17}$$

respectively.

Remark 2.2. It is clear that for value of $n = 1$ the multistage integral mean value method reduces to the integral mean value method.

Remark 2.3. The main step of the modified and traditional MIMVM/IMVM, in handling a Fredholm integral equation of the second kind, say Eq. (1.1), is to solve/handel the corresponding algebraic system. Therefore, the methods break down when the mentioned step comes to a deadlock.

Remark 2.4. To handel the Eq. (1.1) with a non-degenerate kernel, by means of the DKM, IMVM and MIMVM, it is necessary to approximate its kernel by a degenerate kernel of the form (2.1). In this study, we use an m-order degenerate approximation of the kernel by using Taylor series and Lagrange interpolation as follows

$$K(x,t) \cong \sum_{j=0}^{m} \left\{ \frac{1}{j!} \left((x-x_0)\frac{\partial}{\partial x} + (t-t_0)\frac{\partial}{\partial t} \right)^j K(x,t) \right\}_{x=x_0, t=t_0}, \tag{2.18}$$

and

$$K(x,t) \cong \sum_{j=1}^{m} L_j(x) K(x_j, t), \tag{2.19}$$

respectively. In (2.19), $L_j(x)$ and x_j, $k = 1, ..., m$, are Lagrange polynomials and collocation nodes respectively. For Taylor series approximation, denoted by (2.18), we set $x_0 = t_0 = 0$. It is well known that an arbitrary $\mathcal{L}_2$-kernel can be approximated in norm by a degenerate kernel [9].

3. Test Examples

To show the efficiency of the present algorithm, modified MIMVM described in the previous part, we present some prototype examples. The computations will be performed using the program ***MMIMVMforOneDimenFIE2*** reported in the Appendix. This program was designed in a general manner.

We use approximate solution given by MIMVM, for $m = l$, the order of degenerate kernel, and $n = k$ by $u_{l,k}(x)$ corresponding to the exact solution $u(x)$. For comparison the solution given by MIMVM with the exact solution, we report the absolute error which is defined by

$$Error(l,k) = |u(x) - u_{l,k}(x)|, \tag{3.1}$$

and the maximum error which is defined by

$$\|E_{l,k}\| = \max_{x \in [a,b]} Error(l,k). \tag{3.2}$$

Example 3.1. Consider the following linear Fredholm integral equation of the second kind with a non-degenerate kernel [9]

$$u(x) = e^x - x - \int_0^1 x(e^{xt} - 1) u(t) dt, x \in [0,1]. \tag{3.3}$$

The exact solution is $u(x) = 1$.

The results of applying the modified MIMVM for Eq. (3.3) are shown in Table 1 and Fig. 1.

Table 1. The maximum errors for Example 3.1, by using modified MIMVM corresponding to the Lagrange interpolation approximation, with equally-spaced/Chebychev collocation nodes, and Taylor series approximation to the kernel.

	Equally-spaced nodes	Chebychev nodes	Taylor approximation
l	$\|E_{l,1}\|$	$\|E_{l,1}\|$	$\|E_{l,1}\|$
5	4.59546×10^{-05}	2.80684×10^{-05}	1.37865×10^{-03}
7	1.06648×10^{-07}	4.21565×10^{-08}	2.45259×10^{-05}
9	1.60152×10^{-10}	3.52286×10^{-11}	2.72328×10^{-07}
11	1.66618×10^{-13}	2.09832×10^{-14}	2.06350×10^{-09}
13	3.44169×10^{-15}	7.77156×10^{-16}	1.13437×10^{-11}

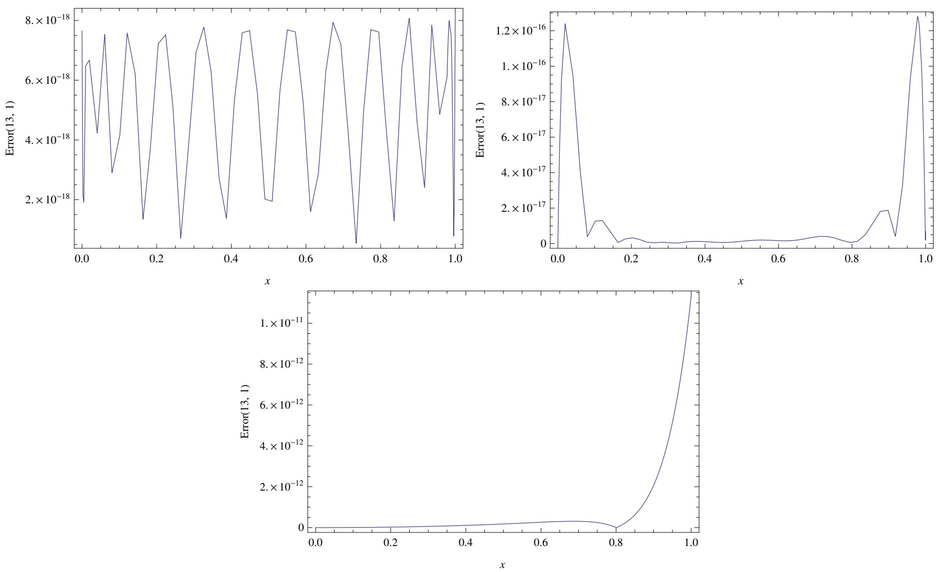

Figure 1. The absolute error of the approximate solutions given by modified MIMVM corresponding to the 13-order degenerate approximation of the kernel of Eq. (3.3), Example 3.1. Left: Lagrange interpolation approximation with equally-spaced collocation nodes, right: Lagrange interpolation approximation with Chebychev collocation nodes, and center: Taylor series approximation.

Example 3.2. Consider the following linear Fredholm integral equation of the first kind with a degenerate kernel [10]

$$u(x) = \cos(2\pi x) + \frac{1}{2}\sin(4\pi x) - \int_0^1 \sin(4\pi x + 2\pi t)u(t)dt, x \in [0,1]. \tag{3.4}$$

The exact solution is $u(x) = \cos(2\pi x)$.

By applying the modified MIMVM, for $n = 1$, the solution of the corresponding algebraic system is

$$\begin{cases} \alpha_{1,1} = -\frac{1}{2} \\ \alpha_{1,2} = 0. \end{cases} \tag{3.5}$$

And also

$$u(x;\alpha) = \alpha_{1,1}\sin(4\pi x) + \alpha_{1,2}\cos(4\pi x) + \frac{1}{2}\sin(4\pi x) + \cos(2\pi x). \tag{3.6}$$

Substituting (3.5) into the (3.6) gives

$$u(x) = \cos(2\pi x). \tag{3.7}$$

Therefore, modified MIMVM by $n = 1$ (modified IMVM) gives the exact solution of Eq. (3.4).

Example 3.3. Consider the following nonlinear Fredholm integral equation of the second kind [11]

$$u(x) = e^{x+1} - \int_0^1 e^{x-2t}u^3(t)dt, 0 \le x < 1. \tag{3.8}$$

The reported exact solution in [11] is $u(x) = e^x$.

Applying the modified MIMVM, for $n = 1$, the solutions of the corresponding algebraic system are

$$\alpha_{1,1} = e - 1, \frac{1}{2}\left(1 - e \pm i\sqrt{-1 - 2e + 3e^2},\right). \tag{3.9}$$

where $i^2 = -1$. And also

$$u(x;\alpha) = e^x\alpha_{1,1} + e^{x+1}. \tag{3.10}$$

Substituting (3.9) into the (3.10) gives

$$u(x) = e^x, \frac{e^x\left(1 - e \pm i\sqrt{-1 - 2e + 3e^2}\right)}{2(-1 + e)}. \tag{3.11}$$

Therefore, modified MIMVM by $n = 1$ (modified IMVM) gives the exact solutions of Eq. (3.8).

Example 3.4. Consider the following nonlinear Fredholm integral equation of the second kind [11]

$$u(x) = e^x - \frac{1}{9}\left(1 + 2e^3\right)x + \int_0^1 xtu^3(t)dt, 0 \le x < 1. \tag{3.12}$$

The reported exact solution in [11] is $u(x) = e^x$.

Applying the modified MIMVM, for $n = 1$, the solutions of the corresponding algebraic system are

$$\alpha_{1,1} = \frac{1}{9}\left(1 + 2e^3\right), \frac{1}{648}\left(-29088 + 9720e + 144e^3 \pm i\beta\right), \tag{3.13}$$

where $i^2 = -1$ and $\beta = \sqrt{-853979760 + 566870400e - 92903760e^2}$. And also

$$u(x;\alpha) = x\alpha_{1,1} - \frac{1}{9}\left(1 + 2e^3\right)x + e^x. \tag{3.14}$$

Substituting (3.13) into the (3.14) gives

$$u(x) = e^x, 15ex - 45x + e^x \pm \frac{1}{2} ix\sqrt{15(360 - 59e)e - 8135}. \tag{3.15}$$

Therefore, modified MIMVM by $n = 1$ (modified IMVM) gives all exact solutions of Eq. (3.12).

4. Comparison and Discussion

In this part, a comparison between the results given by modified MIMVM and traditional MIMVM, to check the accuracy of the modified MIMVM, is given. For this respect, in this section, we discuss about applying the traditional MIMVM to the all examples of the previous section. The computations will be performed using the program ***TMIMVMforOneDimenFIE2*** reported in the Appendix. Applying the traditional MIMVM, for $n = 1$ to Examples 3.3 and 3.4 gives the real valued exact solutions. Fig. 2 shows the results of applying traditional MIMVM to Example 3.2. These results show that, traditional MIMVM for $n = 3$ gives the exact solution. For Eq. (3.3), traditional MIMVM does not give suitable numerical solution (see Fig. 3). Unfortunately, by increasing the order of degenerate kernel and value of n, the volume of the computations increases considerably.

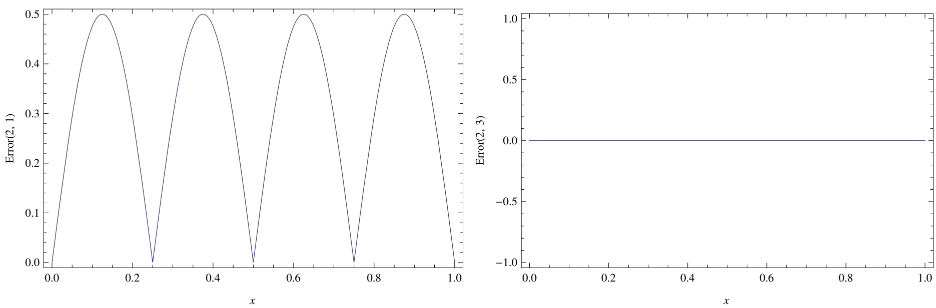

Figure 2. The absolute error of the approximate solutions given by traditional MIMVM for Eq. (3.4), Example 3.2.

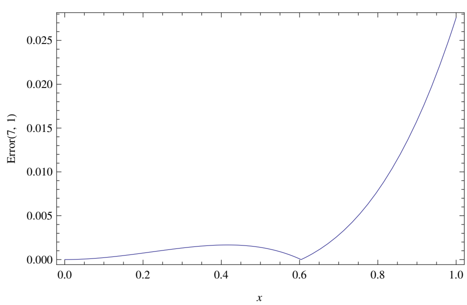

Figure 3. The absolute error of the approximate solutions given by traditional MIMVM for Eq. (3.3), Example 3.1.

It is well known that accuracy of the interpolation can be obtained by the selection of the proper collocation nodes to be related to the selection of the interpolation functions [12]. In the case of Lagrange interpolation, a Chebychev collocation is needed. It is important to notice that, IMVM/MIMVM for handling the FIE2s with non-degenerate kernels or degenerate kernel with more terms may need significant more computation time and computer hardware requirements.

We finish this section by promoting the modified IMVM/MIMVM to the Lagrange-collocation method. In [13, 14] it was shown that DKM, on the condition that the source function is approximated by the same way of producing degenerate kernel, becomes as a projection method. Therefore, by approximating the source function by the same way of producing degenerate kernel by using the Lagrange interpolation method, IMVM is promoted to the Lagrange-collocation method. We prefer to use this promotion of modified IMVM/MIMVM for Eq. (3.3). By choosing three Chebychev/equally collocation nodes, to make a degenerate approximation of the kernel as well as an approximation of same order to the source function, we find that approximate solution obtained by modified IMVM (MIMVM) gives the exact solution of Eq. (3.3).

5. Concluding Remarks

In this paper, a multistage schema of the integral mean value method, namely modified multistage integral mean value method, was proposed as a reliable treatment of the one-dimensional Fredholm integral equations of the second kind. The proposed algorithm showed reliability in handling these problems. For a non-degenerate kernel, the choice of Lagrange polynomials is related to the fact that the making the approximate degenerate kernel is based on approximation generated by interpolation. The alternative of using Bernstein polynomials and sinc functions are also possible. Finally, extensions of the methods to higher order and dimensional can be accommodated.

Acknowledgements

The authors would like to thank the anonymous referees, especially the second referee, for his/her constructive comments and suggestions. Many thanks are due to the Ilam University of Iran for financial support.

Appendix (Mathematica Programs)

All programs provided in Mathematica 9.

Program. 1: A Sample Mathematica Program of Modified MIMVM

MMIMVMforOneDimenFIE2 is a sample Mathematica program for solving FIE2s by using MIMVM. The general command is

MMIMVMforOneDimenFIE2[functionF,functionf,Type,n2]

in which the user must define the function $F(u)$ by **functionF**, source function, $f(x)$, by **functionf**, the integral region, **IntRegion={a,b}**, and a degenerate kernel as **kernelg=**$\{\mathbf{g_j(x)}\}_{\mathbf{j=1}}^{\mathbf{m}}$ and **kernelh=**$\{\mathbf{h_j(t)}\}_{\mathbf{j=1}}^{\mathbf{m}}$. For a non-degenerate kernel, denoted by **GivenKernel**, user must define Taylor series or interpolation approximations by using the programs **TylorApproximation** and **LagrangeInterpolation** respectively. Also, user must define the number of subintervals by $n2$, and the type of handling the corresponding algebraic system by **Type** (for exact solution take **Type=exact**, for numerical solutions without initial guesses take **Type=nexact** and for numerical solution by Newton method take **Type=numerical**). The program monitors given/approximate equation, number of the subintervals, unknowns, algebraic system, $u(x; Unknowns)$ and solution(s) of the given equation. For the **numerical** case, the program requests to input the exact solution for monitoring the plot and table of the absolute error as well as maximum error.

(∗ − − − − − − − − − − **The Main Block of Modified MIMVM** − − − − − − − − − − ∗)
ClearAll[Global`*]
MMIMVMforOneDimenFIE2[functionF_, functionf_, Type_, n2_]:=Module[{}, a = IntRegion[[1]];
b = IntRegion[[2]];
$h = \frac{b-a}{\text{n2}}$;
$\text{a1}_{\text{i\_}}:=a+hi$;
$F(\text{u\_}):=\text{functionF}$;
n1 = First[Dimensions[kernelg]];
$\text{u1}(x)=\sum_{j=1}^{\text{n1}}\sum_{i=1}^{\text{n2}}\alpha_{i,j}\text{kernelg}[[j]]+\text{functionf}$;
$\text{Eq}_{\text{i\_,j\_}}:=\alpha_{i,j}-\int_{\text{a1}_{i-1}}^{\text{a1}_i}\text{kernelh}[[j]](F(u)/.\{u\to(\text{u1}(x)/.x\to t)\})\,dt$;
$\text{EqsSys}=\text{Flatten}\left[\text{Table}\left[\left\{\text{Eq}_{i,j}\right\},\{i,1,\text{n2}\},\{j,1,\text{n1}\}\right]\right]$;
$\text{Unknowns}=\text{Flatten}\left[\text{Join}\left[\text{Table}\left[\alpha_{i,j},\{i,1,\text{n2}\},\{j,1,\text{n1}\}\right]\right]\right]$;
nn1 = First[Dimensions[Unknowns]];
Which[Type===numerical $\wedge\, a\neq-\infty\wedge b\neq\infty$, {NewtonInitialGuess1 = b;
NewtonInitialGuess2 = Table[NewtonInitialGuess1, {k, 1, nn1}];
NewtonInitialGuess3 = Riffle[Unknowns, NewtonInitialGuess2];
NewtonInitialGuess = Partition[Flatten[NewtonInitialGuess3], 2]}];
Which[Type===numerical, Module[{s = 0, e = 0},
{FindRoot[EqsSys, NewtonInitialGuess, StepMonitor :→ s++, EvaluationMonitor :→ e++,
WorkingPrecision → 60]}], Type===exact, Solve[EqsSys = 0, Unknowns],
Type===nexact, NSolve[EqsSys = 0, Unknowns]]]
(∗ − − − − − − − − **The End of Main Block of Modified MIMVM** − − − − − − ∗)

(∗ − − − − − − − − **The Main Block of Lagrange Interpolation** − − − − − − − − ∗)
LagrangeInterpolation[function_, a_, b_, order_, type_:=Module[{}, $\text{h1}:=\frac{b-a}{\text{order}-1}$;
$\text{x2}_{\text{j\_}}:=a+\text{h1}(j-1)$;
$\text{x1}_{\text{j\_}}:=N\left[\frac{1}{2}\left((b-a)\cos\left(\frac{\pi(2j-1)}{2\text{order}}\right)+a+b\right),60\right]$;
$\text{roots22}=N[\text{Solve}[P_{\text{order}}(x)=0,x,\mathbb{R}],60]$;
u23 = Table[x, {i, 1, 1}];
$\text{x21}_{\text{i\_}}:=\text{Sort}[\text{Flatten}[\text{u23}//.\text{roots22}],\text{Less}][[i]]$;
$\text{x22}_{\text{i\_}}:=\frac{1}{2}\left((b-a)\text{x21}_i+a+b\right)$;
$\text{Which}\left[\text{type===Equal1},\text{x3}_{\text{j\_}}:=\text{x2}_j,\text{type===Cheby},\text{x3}_{\text{j\_}}:=\text{x1}_j,\text{type===Legendre},\text{x3}_{\text{j\_}}:=\text{x22}_j\right]$;
$\text{Lagr1}(\text{p\_},\text{i\_},\text{x\_}):=\text{If}\left[p\neq i,\frac{x-\text{x3}_i}{\text{x3}_p-\text{x3}_i},1\right]$;
$\text{LagrangeBase}(\text{p\_},\text{n\_},\text{x\_}):=\prod_{i=1}^{n}\text{Lagr1}(p,i,x)$;
kernelg = Flatten[Table[LagrangeBase(p, order, x), {p, 1, order}]];
$\text{kernelh}=\text{Flatten}\left[\text{Table}\left[\text{function}/.\,x\to\text{x3}_i,\{i,1,\text{order}\}\right]\right]$;
ApproxFunc = Inner[Times, kernelg, kernelh, Plus]];
(∗ − − − − − − − − **The End of Main Block of Lagrange Interpolation** − − − − − − ∗)

(∗ − − − − − − − − **The Main Block of Tylor Approximation** − − − − − − − − ∗)
TylorApproximation[function_, order_]:=Module[{},
TaylorSeries = Series[GivenKernel, {t, 0, order}, {x, 0, order}];

```
kernelg = Table [x^n, {n, 0, order}] ;
kernelh = CoefficientList[TaylorSeries, x];
ApproxFunc = Inner[Times, kernelg, kernelh, Plus]];
(* − − − − − − − −The End of Main Block of Tylor Approximation − − − − − −*)


(* − − − − − − − − − − − − − − − − Commands to our Examples − − − − − − − − − − −*)
(* − − − − − − − − − − − − − − −Example 1 − − − − − − − − − − − − − − − − − *)
n2 = 1;
order = 5; IntRegion = {0, 1};
functionf = Exp[x] − x;
GivenKernel = −x(Exp[tx] − 1);
LagrangeInterpolation[GivenKernel, 0, 1, order, Cheby];
{functionF, Type} = {u, numerical};
(* − − − − − − − − − − − − − − −Example 2 − − − − − − − − − − − − − − − − − *)
n2 = 1;
kernelg = {Sin[4πx], Cos[4πx]};
kernelh = {−Cos[2πt], −Sin[2πt]};
IntRegion = {0, 1};
{functionF, functionf, Type} = {u, 1/2 Sin[4πx] + Cos[2πx], exact} ;
(* − − − − − − − − − − − − − − −Example 3 − − − − − − − − − − − − − − − − − *)
n2 = 1;
kernelg = {Exp[x]};
kernelh = {−Exp[−2t]};
IntRegion = {0, 1};
{functionF, functionf, Type} = {u^3, Exp[x + 1], exact} ;
(* − − − − − − − − − − − − − − −Example 4 − − − − − − − − − − − − − − − − − *)
n2 = 1;
kernelg = {x};
kernelh = {t};
IntRegion = {0, 1};
{functionF, functionf, Type} = {u^3, Exp[x] − 1/9 (2Exp[3] + 1)x, exact} ;

(* − − − −Common Commands for Monitoring the Results − − − − − −*)
Style["Equation: One Dimensional FIE2", "Title", 20]
Style["Method: Modified MIMVM", "Title", 20]
Print[" ↓ − − − − − − − − − − − ", Style["Given Equation", Red, Bold], " − − − − − − − − − − − ↓ "];
GivenKernel1 = Inner[Times, kernelh, kernelg, Plus];
Print["u(x)=", TraditionalForm[functionf], "+",
DisplayForm[SubsuperscriptBox[" ∫ ", IntRegion[[1]], IntRegion[[2]]]],
TraditionalForm[GivenKernel1] ∗ (functionF/.u → "u(t)"), "dt"]
Print [" ↓ − − − − − − − − − − − − − −", Style ["Unknowns", Blue, Bold] , " − − − − − − − − − ↓ "]
Sol = MMIMVMforOneDimenFIE2[functionF, functionf, Type, n2];
Sol//TraditionalForm
Print[" ↓ − − − − − − − − − − ", Style["Algebraic system/equation", Blue, Bold]," − − − − − − − − ↓ "]
For[i = 1, i < nn1 + 1, i++, Print["Equation", i, ":", TraditionalForm[Simplify[EqsSys[[i]]]], "=0"]]
Print[" ↓ − − − − − − − − − − − − − − ", Style["u(x;Unknowns)", Blue, Bold]," − − − − − − − − − ↓ "]
Print["u(x;Unknowns)=", TraditionalForm[u1[x]]]
Print[" ↓ − − − − − − − − − − − − − − "Style["Solution(s)", Blue, Bold]," − − − − − − − − − − − − ↓ "]
Sol1 = Simplify[u1[x]//.Sol];
Print["u(x)=", TraditionalForm[Simplify[Sol1]]]
```

Which[Type===numerical, Module[{},
ExactSolution =Input["Please input the exact solution of the equation for monitoring the plot & table of the error?"]/.{Global\`$x \to x$};
Framed[Print[Style["For the Following Plot and Table the Given Exact Solution is: u(x)=",
Red, Bold], Style[ExactSolution//TraditionalForm, Red, Bold]]];
Print[" ↓ − − − − − − − − − − − ", Style["The Plot of the Error", Blue, Bold]," − − − − − − − − − ↓ "];
Print[Plot[Abs[Sol1 − ExactSolution], {x, IntRegion[[1]], IntRegion[[2]]}, Frame → True, Axes → False,
FrameLabel → {x, Error[*order*, *n*2]}]];
Print[" ↓ − − − − − − − − − − −", Style["The Table of the Error", Blue, Bold]," − − − − − − − − − ↓ "];
TableForm[Table[{x, First[Abs[Sol1 − ExactSolution]]//ScientificForm},
{x,IntRegion[[1]],IntRegion[[2]],0.1}],
TableHeadings → {None, {"x", DisplayForm[Error[*order*, *n*2]]//TraditionalForm,
DisplayForm["Exact Solution="]DisplayForm[ExactSolution//TraditionalForm]}}]]]
Print[Style[The maximum value of the error is:, Red, Bold]];
NMaximize[{|First[Sol1] − ExactSolution| , IntRegion[[1]] ≤ x ≤ IntRegion[[2]]}, x]]//ScientificForm
Print[" ↑ − − − − − − − − − ", Style["The end of the Results", Red, Bold], " − − − − − − − − − − ↑ "]

Program. 2: A Sample Mathematica Program of Traditional MIMVM

TMIMVMforOneDimenFIE2 is a sample MATHEMATICA program for solving FIE2s by using DCM. The general command is

TMIMVMforOneDimenFIE2[functionF,functionf,Type,n2]

The descriptions of needed definitions are similar to **Problem. 1**. Here, we set $c_j = u(\xi_j)$. For Example 3.2, user must set Type=numerical.

ClearAll[Global\`*]
TMIMVMforOneDimenFIE2(functionF_, functionf_, Type_, n2_):=Module[{}, a = IntRegion[[1]];
b = IntRegion[[2]];
n1 = First[Dimensions[kernelg]];
$h = \frac{b-a}{\mathrm{n2}}$;
$\mathrm{a1}_{\mathrm{i\_}}$:=$a + hi$;
C1 = Join [Table [$c_{i,j}$, {i, 1, n2}, {j, 1, n1}]] ;
ξ1 = Join [Table [$\xi_{i,j}$, {i, 1, n2}, {j, 1, n1}]] ;
F(u_):=functionF; u1(x) = $h \sum_{j=1}^{\mathrm{n1}} \sum_{i=1}^{\mathrm{n2}}$ kernelg[[j]] ($F(u)$/. $u \to c_{i,j}$) (kernelh[[j]]/. $t \to \xi_{i,j}$)+functionf;
u11 = u1(x)/. $x \to t$;
$\mathrm{Eq1}_{\mathrm{i\_,j\_}}$:= (u11/. $t \to \xi_{i,j}$) − $c_{i,j}$;
$\mathrm{Eq2}_{\mathrm{i\_,j\_}}$:= $\sum_{\mathrm{r1}=1}^{\mathrm{n1}}$ (kernelg[[r1]]/. $x \to \xi_{i,j}$) $\int_a^b$ kernelh[[r1]]($F(u)$/. u → u11) dt + (functionf/. $x \to \xi_{i,j}$) − $c_{i,j}$;
EqsSys = Flatten [Join [Table [{$\mathrm{Eq1}_{i,j}$, $\mathrm{Eq2}_{i,j}$} , {i, 1, n2}, {j, 1, n1}]]] ;
Unknowns = Flatten[Join[ξ1, C1]];
nn1 = First[Dimensions[Unknowns]];
Which[Type===numerical ∧ $a \neq -\infty \wedge b \neq \infty$, {NewtonInitialGuess(i_):=$\frac{1}{2}$ ($\mathrm{a1}_{i-1} + \mathrm{a1}_i$) ;
NewtonInitialGuess1(i_):=$\mathrm{a1}_i$;
NewtonInitialGuess2 = Table[NewtonInitialGuess1(i), {i, 1, nn1}];
NewtonInitialGuess3 = Riffle[Unknowns, NewtonInitialGuess2];
NewtonInitialGuess = Partition[Flatten[NewtonInitialGuess3], 2]}];
Which[Type===numerical, Module[{s = 0, e = 0}, {FindRoot[EqsSys,
NewtonInitialGuess, StepMonitor :→ s++, EvaluationMonitor :→ e++, WorkingPrecision →
MachinePrecision]}], Type===exact, Solve[EqsSys = 0, Unknowns, ℝ],
Type===nexact, NSolve[EqsSys = 0, Unknowns, ℝ]]

References

[1] A. Shidfar, A. Molabahrami, Solving a system of integral equations by an analytic method, Mathematical and Computer Modelling 54 (2011) 828–835.

[2] A. Molabahrami, An algorithm based on the regularization and integral mean value methods for the Fredholm integral equation of the first kind, Appl. Math. Modelling 37 (2013) 9634–9642.

[3] Z. Avazzadeh, M. Heydari, G.B. Loghmani, Numerical solution of Fredholm integral equations of the second kind by using integral mean value theorem, Appl. Math. Modelling 35 (2011) 2374–2383.

[4] M. Heydari, Z. Avazzadeh, H. Navabpour, G.B. Loghmani, Numerical solution of Fredholm integral equations of the second kind by using integral mean value theorem II. High dimensional problems, Appl. Math. Modelling 37 (2013) 432–442.

[5] K. Atkinson, The Numerical Solution of Integral Equations of the Second Kind, Cambridge University Press, UK, 1997.

[6] A. Molabahrami, Direct computation method for solving a general nonlinear Fredholm integro-differential equation under the mixed conditions: Degenerate and non-degenerate kernels, Journal of Computational and Applied Mathematics 282 (2015) 34–43.

[7] A.M. Wazwaz, Linear and Nonlinear Integral Equations: Methods and Applications, Higher Education Press, Beijin, 2011.

[8] X.C. Zhong, Note on the integral mean value method for Fredholm integral equations of the second kind, Applied Mathematical Modelling 37 (2013) 8645–8650.

[9] R.P. Kanwal, Linear Integral Equations, Theory and Technique, Academic Press, New York, 1971.

[10] X.Z. Liang, M.C. Liu, X.J. Che, Solving second kind integral equations by Galerkin methods with continuous orthogonal wavelets, Journal of Computational and Applied Mathematics 136 (2001) 149–161.

[11] E. Babolian, A. Shahsavaran, Numerical solution of nonlinear Fredholm integral equations of the second kind using Haar wavelets, Journal of Computational and Applied Mathematics 225 (2009) 87–95.

[12] N. Bellomo, L. Preziosi, Modelling, MATHEMATICAl Methods and Scientific Computation, CRC Press, Boca Raton, 1996.

[13] A. Molabahrami, The relationship of degenerate kernel and projection methods on Fredholm integral equations of the second kind, Commun. Numer. Anal. 1 (2017) 34–39.

[14] A. Molabahrami, A modified degenerate kernel method for the system of Fredholm integral equations of the second kind, Iranian Journal of Mathematical Sciences and Informatics 14 (2019) 43–53.