International Journal of Mathematics and Mathematical Sciences
ISSN: 2454-7271, Copyright@universalprint
Volume: 01 No.: 02, Pages: 95- 109, Published:Dec.,2015
Web: www.universalprint.org , Publication: Universal Print
Title Key: Modified Iterative Mehod for the Solution of Fredholm.…

Modified Iterative Method For the Solution of Fredholm Integral Equations of the Second Kind via Matrices

Shoukralla, E. S [1], Saber. Nermein. A [2] and EL-Serafi, S. A.[3]
[1]st Author, Prof. Dr, faculty of engineering and technology, Future University, Egypt
Email:shoukrala@hotmail.com
[2]st Author, Post Graduate student, Faculty of Electronic University, Egypt
Email:engnermeinn#hotmail.com
[3]st Author, Prof. Dr, Faculty of Electronic University, Egypt
Email:said-elserafi@yahoo.com

Abstract
The Traditional Iterative technique is modified by expanding both the kernel and the known functions into Maclaurin polynomials of the same degree. The given modified method gives a very simple form for the iterated kernels via the well - known Hilbert matrix. Thus, the iterative solutions of an integral equation of the second kind can be reduced to the solution of a matrix equation, whereas only one coefficient matrix is required to be computed. Therefore, computational complexity can be considerably reduced and much computational time can be saved. The convergence of the given iterated solution is studied and the three conditions are given concluded results , figures, and Tables of calculations are observed during the solution of some numerical examples using MatLab.

Keywords: Integral Equations, Iterative Methods, Approximate Solutions. Matrix Treatment

1. Introduction

Integral equation is encountered in a variety of applications in potential theory, geophysics, electricity and magnetism, radiation, and control systems [1]. Many methods of solving Fredholm integral equation of the second kind have been developed in recent years [1,3,6,7,13], such as quadrature method, collocation method and Galerkin method, expansion method, product-integration method, deferred correction method, graded mesh method, and Petrov–Galerkin method. In addition, the iterated kernel method is a traditional method for solving the second kind. However, this method also requires a huge size of calculations. The objective of this paper is to establish a promising iterative algorithm that can be easily programmed on Matlab, and give numerical examples with tables and figures of the obtained results. Computational modifications are performed on the iterative algorithm presented in [9], where the procedure beginning by replacing the kernel of an integral equation approximately by a degenerate kernel [4,5,8] in a matrix form using Maclaurin polynomial of degree $\boldsymbol{n}$, whereas the data function is approximated by Maclaurin polynomial of the same degree $\boldsymbol{n}$ [2]. Owing to the simplicity of some operational matrix of integration, the iterated kernels is represented in a very simple form via Hilbert matrix. This simplifies the present iterative algorithm and reduces the problem of computing iterative solutions to the computation of only one matrix.

Despite of the advantages of methods [4,5] there was apparent higher cost comparing with the present method, that minimizes the computational effort and smooth the round-off errors out. The convergence of the given modified iterative solutions is studied and three conditions for the existence of the solutions are given.

Due to the simple form of the obtained iterated kernels, which is straightforward and convenient for computation, The present method may be generalized to solve both second kind and well-posed singular integral equations of the first kind [10,11].

2. Modified Iterative Method

Consider the Fredholm integral equation of the second kind

$$\phi(x)=\int_{\alpha}^{\beta} K(x,y)\phi(y)dy+f(x)\ \forall\ q\geq 1 \tag{1}$$

where the function $f(x)$ and the kernel $K(x,y)$ are given. The kernel $K(x,y)$ is defined in the square $\Omega=\{\alpha\leq x\leq\beta,\alpha\leq y\leq\beta\}$ in the $\boldsymbol{xy}-$ plane. The function $\phi(x)$ is the unknown required solution.

The given iterated Algorithm starts with an initial approximation $\phi^{(0)}(x)$ to the solution $\phi(x)$ of integral equation (1) and then generates a sequence of solutions $\left\{\phi^{(q)}(x)\right\}_{q=0}^{\infty}$ that converges to $\boldsymbol{\varphi(x)}$ such that $\left\|\frac{f^{(q)}(x)-f^{(q-1)}(x)}{f^{(q)}(x)}\right\|<\delta;\delta>0$

After the initial solution $\mathrm{f}^{(0)}(\mathrm{x})$ is chosen, the sequences of approximate solutions can be generated by computing

$$\mathrm{f}^{(\mathrm{q})}(\mathrm{x})=\mathrm{f}(\mathrm{x})+\int_{\alpha}^{\beta}\mathrm{K}(\mathrm{x},\mathrm{y})\mathrm{f}^{(\mathrm{q}\text{-}1)}(\mathrm{y})\mathrm{dy}\ \forall\ q\geq 1 \qquad (2)$$

If we begin by $\phi^{(0)}(x)=0$, then we get

$$\phi^{(q)}(x)=f(x)+\sum_{s=1}^{q}\int_{\alpha}^{\beta}K^{(S)}(x,y)\,f(y)\,dy \quad \forall\ q\geq 1 \qquad (3)$$

where the iterated kernels $\mathrm{K}^{(\mathrm{s})}(\mathrm{x},\mathrm{y})$ can be found by the recurrence form

$$K^{(S)}(x,y)=\int_{\alpha}^{\beta}K(x,z)K^{(s-1)}(z,y)dz \ ; \ s\geq 2 \quad \forall\ q\geq 1 \qquad (4)$$

where

$$K^{(1)}(x,y)=K(x,y)$$

Approximate the kernel $\boldsymbol{K(x,y)}$ using Maclaurin polynomial for functions of two variables of degree $\boldsymbol{n}$, then we can put it in the matrix form

$$K(x,y)=\mathrm{P}^{\mathrm{t}}(y)\mathrm{L}(\mathrm{n})\mathrm{K}\mathrm{L}(\mathrm{n})\mathrm{P}(x) \qquad (5)$$

where $\mathbf{K}=(\boldsymbol{k_{ij}})$ 1s an $(\mathrm{n}+1)\times(\mathrm{n}+1)$ coefficients matrix whose entries are defined by

$$\boldsymbol{k_{ij}}=\begin{cases}\left.\dfrac{\partial^{i}\partial^{j}K(x,y)}{\partial x^{i}\partial y^{j}}\right|_{(x,y)=(0,0)} & ; i+j\leq n \\ 0 & ; i+j>n\end{cases} \quad \vee\ i,j=\overline{o,n} \qquad (6)$$

and the matrix $\mathbf{L(n)}$ 1s an $(n+1)\times(n+1)$ matrix defined by

$$\mathrm{L}(\mathrm{n})=\mathrm{diag}\begin{pmatrix}\dfrac{1}{0!} & \dfrac{1}{1!} & \dots & \dfrac{1}{\mathrm{n}!}\end{pmatrix} \qquad (7)$$

The matrix $\mathbf{P(x)}$ of order $(n+1)\times(n+1)$ is defined to be

$$P^t(x)=\left[p_o(x) \quad p_1(x) \quad \dots \quad p_n(x)\right] \tag{8}$$

where

$$p_i(x) = x^i \quad \text{for } i=\overline{0;n}$$

Now, we define the Maclaurin operational integration matrix ,H, to be

$$H=\int_\alpha^\beta P(y)P^t(y)dy \tag{9}$$

$$=\beta B(\beta)H B(\beta)-\alpha B(\alpha)H B(\alpha)$$

Where

$$B(\alpha)=\text{diag}\left(1 \quad \alpha \quad \alpha^2 \quad \dots \quad \alpha^n\right);B(\beta)=\text{diag}\left(1 \quad \beta \quad \beta^2 \quad \dots \quad \beta^n\right) \tag{10}$$

Here **H** is the well - known Hilbert matrix of order $(n+1)\times(n+1)$ with elements $H_{ij}=(i+j-1)^{-1}$. Substituting (5) into (4) and by virtue of (9) the iterated kernels $K^{(S)}(X,Y)$ become

$$K^{(s)}(x,y)=P^t(x)\left[\left(L(n)K L(n)\right)^t H\right]^{(s-1)}\left(L(n)K L(n)\right)^t P(y) \tag{11}$$

Also, approximating the data function $f(x)$ in Maclourin polynomial of degree $\boldsymbol{n}$ yields

$$f(x)=P^t(x)F \quad ; \quad F^t=\left[f_o \quad f_1 \quad \dots \quad f_n\right] \tag{12}$$

where

$$f_i=\frac{1}{i!}\left\{\frac{d^i f(x)}{dx^i}\right\}_{x=0} \quad ; \; i=\overline{0,n}$$

Now, Substituting (11) and (12) into (3) we find that

$$f^{(q)}(x)=P^t(x)F + \sum_{s=1}^{q}\int_\alpha^\beta P^t(x)\left[\tilde{K}^t H\right]^{s-1}\tilde{K}^t P(y)\left[P^t(y)F\right]dy \tag{13}$$

Again using the operational matrix given by (9), we get

$$f^{(q)}(x)=P^{t}(x)\left[I_{n+1}+\sum_{s=1}^{q}\left[\tilde{K}^{t}H\right]^{s}\right]F\ \forall\, \boldsymbol{q}\geq \mathbf{1} \tag{14}$$

where

$$\tilde{K}=L(n)\,K\,L(n)$$

Let $Q^{(q)}$ is the $(n+1)\times 1$ iterative coefficients vector defined by

$$Q^{(q)}\left(a_{ij}\right)=\left[I_{n+1}+\sum_{s=1}^{q}\left[\tilde{K}^{t}H\right]^{s}\right] \tag{15}$$

Then we have

$$f^{(q)}(x)=P^{t}(x)\,Q^{(q)}\,F$$

If $C=\tilde{K}^{t}H$,then the approximate solution $f^{(q)}(x)$ converges to the exact $\boldsymbol{\phi}(\boldsymbol{x})$ if one of the following three conditions is satisfied

$$\|C\|_{\text{¥}}=\max_{1\text{£}i\text{£}n}\sum_{j=1}^{n}\left|C_{ij}\right|<1;$$

$$\|C\|_{\text{¥}}=\max_{1\text{£}j\text{£}n}\sum_{i=1}^{n}\left|C_{ij}\right|<1;$$

$$\|C\|_{2}=\left(\rho\left(CC^{t}\right)\right)^{2}<1$$

Where $\rho\left(CC^{t}\right)$ is the spectral radius of $\boldsymbol{CC^{t}}$.

That is, to find the iterative solutions $\phi^{(q)}(x)$ of integral equation (1) it is required only to compute the matrix K given by (6).

3. Computational Results

In this part we try to apply the old iterated kernel technique and the modified technique via matrices to approximate the solution of Fredholm integral equation of the second kind.

We will use suitable algorithms and MATLAB software, then we will compare the exact solution with the approximate one using suitable number of n points and finally graph the results.

The computation is made on a personal computer using MATLAB 2012.

Example(1)

$$\phi(x)=1+\int_{0}^{1} x\,\phi(t)\,\mathrm{dt}$$

Whose exact solution is given by $\phi(x)=1+2x$

Table (1)

the approximate solution using the old iterated technique

N	$\phi(x)$
1	x + 1.0
2	1.5 x + 1.0
3	1.75 x + 1.0
4	1.875 x + 1.0
5	1.9375 x + 1.0
6	1.96875 x + 1.0
7	1.984375 x + 1.0
8	1.9921875 x + 1.0
9	1.99609375 x + 1.0
10	1.998046875 x + 1.0
11	1.9990234375 x + 1.0
12	1.99951171875 x + 1.0
13	1.999755859375 x + 1.0
14	1.9998779296875 x + 1.0
15	1.99993896484375 x + 1.0
16	1.999969482421875 x + 1.0
17	1.9999847412109375 x + 1.0
18	1.99999237060546875 x + 1.0
19	1.999996185302734375 x + 1.0
20	1.9999980926513671875 x + 1.0
21	1.99999904632568359375 x + 1.0
22	1.999999523162841796875 x + 1.0
23	1.9999997615814208984375 x + 1.0
24	1.99999988079071044921875 x + 1.0
25	1.999999940395355224609375 x + 1.0
26	1.9999999701976776123046875 x + 1.0

27	1.99999998509883880615234375 x + 1.0
28	1.999999992549419403076171875 x + 1.0
29	1.9999999962747097015380859375 x + 1.0
30	1.99999999813735485076904296875 x + 1.0
31	1.999999999068677425384521484375 x + 1.0
32	1.9999999995343387126922607421875 x + 1.0
33	1.9999999997671693563461303710938 x + 1.0
34	1.9999999998835846781730651855469 x + 1.0
35	1.9999999999417923390865325927734 x + 1.0
36	1.9999999999708961695432662963867 x + 1.0
37	1.9999999999854480847716331481934 x + 1.0
38	1.9999999999927240423858165740967 x + 1.0
39	1.9999999999963620211929082870483 x + 1.0
40	1.9999999999981810105964541435242 x + 1.0
41	1.9999999999990905052982270717621 x + 1.0
42	1.999999999999545252649113535881 x + 1.0
43	.9999999999997726263245567679405 x + 1.0
44	2.0 x + 1.0

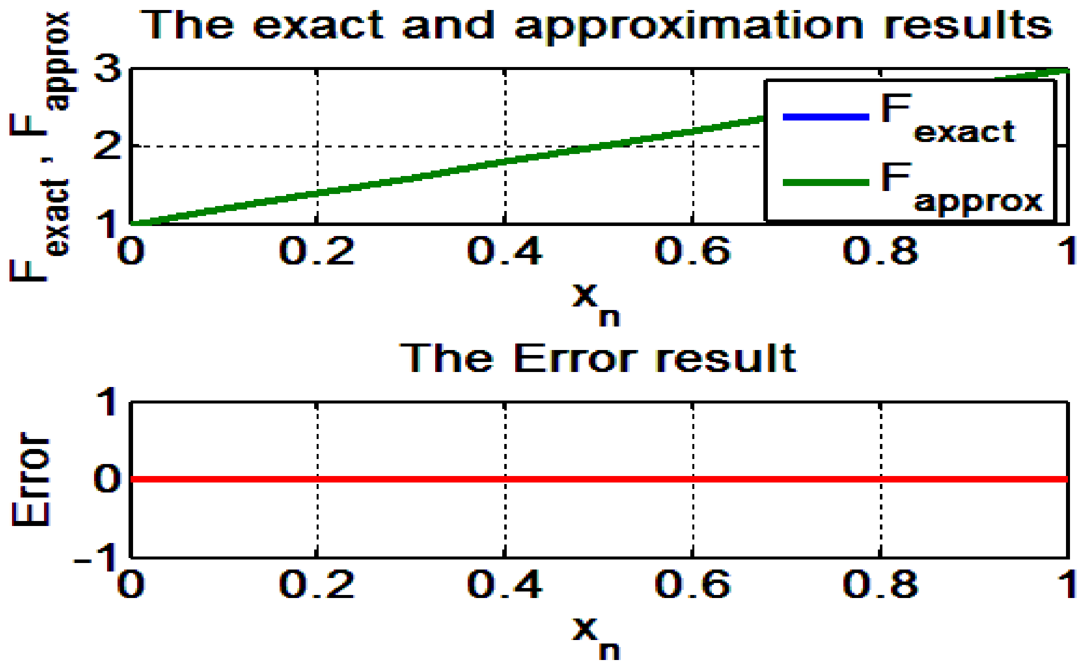
The CPU time is equal to 4.227627 Seconds.

As the above table show, we reached the exact solution after 44 iterations.

Table (2)

The exact and approximate solution of the old iterated technique

N	x_n	The exact sol	The approximate sol	The error
0	0	1	1	0
1	0.1	1.2	1.2	0
2	0.2	1.4	1.4	0
3	0.3	1.6	1.6	0
4	0.4	1.8	1.8	0
5	0.5	2	2	0
6	0.6	2.2	2.2	0
7	0.7	2.4	2.4	0
8	0.8	2.6	2.6	0
9	0.9	2.8	2.8	0
10	1	3	3	0



Figure(1)

Table (3)

the approximate solution using the modified technique via matrices

N	$\phi(x)$	N	$\phi(x)$
1	1.0*x + 1.0	12	1.99951*x + 1.0
2	1.5*x + 1.0	13	1.99975*x + 1.0
3	1.75*x + 1.0	14	1.99987*x + 1.0
4	1.875*x + 1.0	15	1.99993*x + 1.0
5	1.9375*x + 1.0	16	1.99996*x + 1.0
6	1.96875*x + 1.0	17	1.99998*x + 1.0
7	1.98437*x + 1.0	18	1.99999*x + 1.0
8	1.99218*x + 1.0		
9	1.99609*x + 1.0		
10	1.99804*x + 1.0		

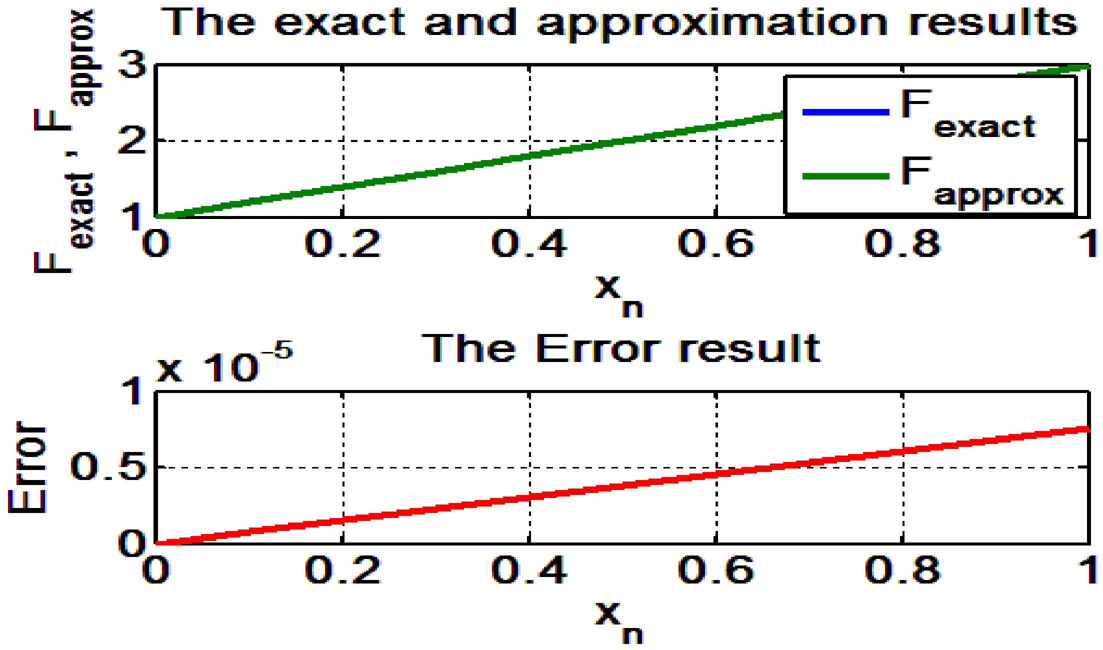
The CPU time is equal to 1.6224 Seconds

As the above table show, we reached the exact solution after 18 iterations.

Table (4)

The exact and approximate solution of the modified technique via matrices

N	x_n	The exact sol	The approximate sol	The error
0	0	1	1	0
1	0.1	1.2	1.19999923706055	7.62939453169409e-07
2	0.2	1.4	1.39999847412109	1.52587890611677e-06
3	0.3	1.6	1.59999771118164	2.28881835928618e-06
4	0.4	1.8	1.79999694824219	3.05175781245559e-06
5	0.5	2	1.99999618530273	3.81469726562500e-06
6	0.6	2.2	2.19999542236328	4.57763671901645e-06
7	0.7	2.4	2.39999465942383	5.34057617196382e-06
8	0.8	2.6	2.59999389648438	6.10351562491118e-06
9	0.9	2.8	2.79999313354492	6.86645507785855e-06
10	1	3	2.99999237060547	7.62939453125000e-06



Figure(2)

Example(2)

$$\phi(x) = e^{x} - 1 + \int_{0}^{1} t\,\phi(t)\,\mathrm{dt}$$

Whose exact solution is given by $\phi(x) = e^{x}$

Table (5)

the approximate solution using the old iterated technique

N	$\phi(x)$
1	exp(x)-0.42957045711477803706657141447067
2	exp(x) - 0.21478522855750270537100732326508
3	xp(x) - 0.10739261427852397901006042957306
4	exp(x) - 0.053696307139034615829586982727051
5	exp(x) - 0.026848153569972055265679955482483
6	exp(x) - 0.013424076784758653957396745681763
7	exp(x) - 0.0067120383923793269786983728408813
8	exp(x) - 0.0033560191959622898139059543609619
9	exp(x) - 0.0016780095984358922578394412994385
10	exp(x) - 0.00083900479876319877803325653076172
11	exp(x) - 0.00041950239983634673990309238433 8
12	exp(x) - 0.00020975119969079969450831413269043
13	exp(x) - 0.00010487559984539984725415706634521
14	exp(x) - 0.000052437799695326248183846473693848
15	exp(x) - 0.000026218900075036799535155296325684
16	exp(x) - 0.000013109449810144724324345588684082
17	exp(x) - 0.0000065547246776986867189407348632812
18	exp(x) - 0.0000032773627935966942459344863891602
19	exp(x) - 0.0000016386811694246716797351837158203
20	exp(x) - 0.00000081934058471233583986759185791016
21	exp(x) - 0.00000040967006498249247670173645019531
22	exp(x) - 0.0000002048354872385971248149871826171
23	exp(x) - 0.00000010241728887194767594337463378906
24	exp(x) - 0.000000051209099183324724435806274414062
25	exp(x) - 0.0000000256040948443114757537841796875
26	exp(x) - 0.000000012802502169506624341011047363281
27	exp(x) - 0.0000000064010237110778689384460449921875
28	exp(x) - 0.0000000032005118555389344692230224609375

29	exp(x) - 0.0000000016002559277694672346115112304688
30	exp(x) - 0.00000000079990059020929038524627685546875
31	exp(x) - 0.0000000004001776687800884246826171875
32	exp(x) - 0.00000000020008883439004421234130859375
33	exp(x) - 0.000000000100044417195022106170654296875
34	exp(x) - 0.00000000005002220859751105308532714843
35	exp(x) - 0.00000000002501110429875552654266357421875
36	exp(x) - 0.00000000001227817847393453121185302734375
37	exp(x) - 0.0000000000059117155615240335464477539062
38	exp(x) - 0.0000000000031832314562052488327026367187
39	exp(x) - 0.00000000000136424205265939235687255859375
40	exp(x) - 0.00000000000045474735088646411895751953125
41	exp(x) - 0.00000000000045474735088646411895751953125

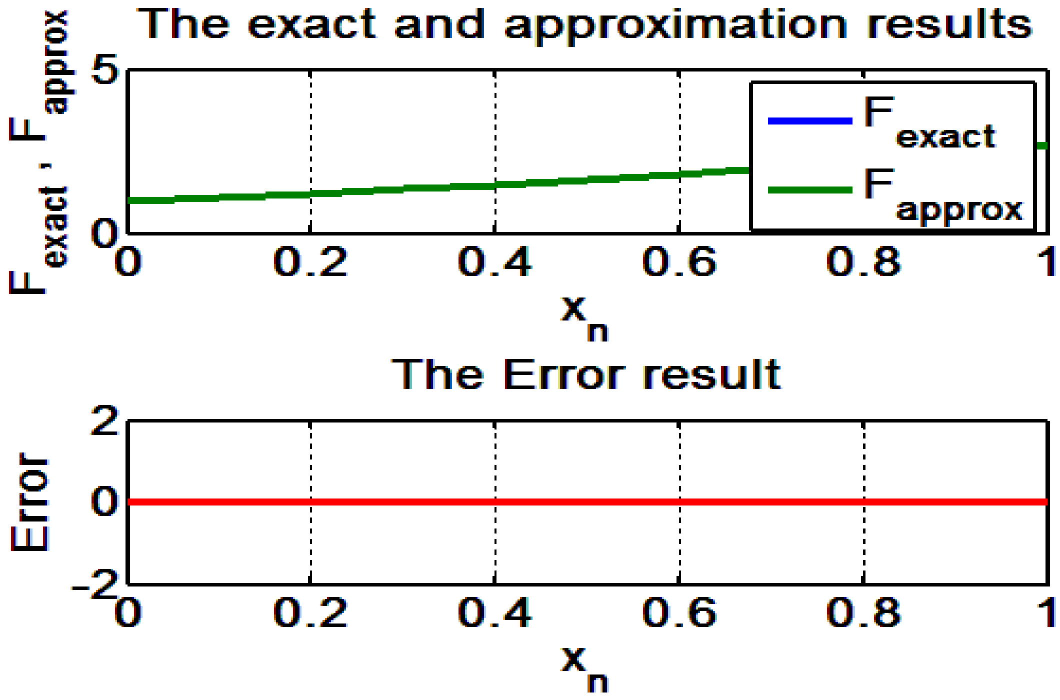
The CPU time is equal to 5.928038 Seconds.

As the above table show, we reached the exact solution after 41iterations.

Table (6)

The exact and approximate solution of the old iterated technique

N	x_n	The exact sol	The approximate sol	The error
0	0	1	0.999999999999545	4.54747350886464e-13
1	0.1	1.10517091807565	1.10517091807519	4.54747350886464e-13
2	0.2	1.22140275816017	1.22140275815972	4.54747350886464e-13
3	0.3	1.34985880757600	1.34985880757555	4.54747350886464e-13
4	0.4	1.49182469764127	1.49182469764082	4.54747350886464e-13
5	0.5	1.64872127070013	1.64872127069967	4.54747350886464e-13
6	0.6	1.82211880039051	1.82211880039005	4.54747350886464e-13
7	0.7	2.01375270747048	2.01375270747002	4.54747350886464e-13
8	0.8	2.22554092849247	2.22554092849201	4.54747350886464e-13
9	0.9	2.45960311115695	2.45960311115650	4.54747350886464e-13
10	1	2.71828182845905	2.71828182845859	4.54747350886464e-13



Figure(3)

Table (7)

the approximate solution using the modified technique via matrices

N	$\phi(x)$
1	0.00833*x^4 + 0.04166*x^3 + 0.16666*x^2 + 0.5*x + 1.5
2	0.00833*x^4 + 0.04166*x^3 + 0.16666*x^2 + 0.5*x +1.82940
3	0.00833*x^4 + 0.04166*x^3 + 0.16666*x^2 + 0.5*x +1.99411
4	0.00833*x^4 + 0.04166*x^3 + 0.16666*x^2 + 0.5*x +2.07646
5	0.00833*x^4 + 0.04166*x^3 + 0.16666*x^2 + 0.5*x +2.11764
6	0.00833*x^4 + 0.04166*x^3 + 0.16666*x^2 + 0.5*x +2.13823
7	0.00833*x^4 + 0.04166*x^3 + 0.16666*x^2 + 0.5*x +2.14852
8	0.00833*x^4 + 0.04166*x^3 + 0.16666*x^2 + 0.5*x +2.15367
9	0.00833*x^4 + 0.04166*x^3 + 0.16666*x^2 + 0.5*x +2.15624
10	0.00833*x^4 + 0.04166*x^3 + 0.16666*x^2 + 0.5*x +2.15753
11	0.00833*x^4 + 0.04166*x^3 + 0.16666*x^2 + 0.5*x +2.15817
12	0.00833*x^4 + 0.04166*x^3 + 0.16666*x^2 + 0.5*x +2.15849

13	0.00833*x^4 + 0.04166*x^3 + 0.16666*x^2 + 0.5*x +2.15865
14	0.00833*x^4 + 0.04166*x^3 + 0.16666*x^2 + 0.5*x +2.15873
15	0.00833*x^4 + 0.04166*x^3 + 0.16666*x^2 + 0.5*x +2.15877
16	0.00833*x^4 + 0.04166*x^3 + 0.16666*x^2 + 0.5*x +2.15879
17	0.00833*x^4 + 0.04166*x^3 + 0.16666*x^2 + 0.5*x +2.15880
18	0.00833*x^4 + 0.04166*x^3 + 0.16666*x^2 + 0.5*x +2.15881
19	0.00833*x^4 + 0.04166*x^3 + 0.16666*x^2 + 0.5*x +2.15881
20	0.00833*x^4 + 0.04166*x^3 + 0.16666*x^2 + 0.5*x +2.15881

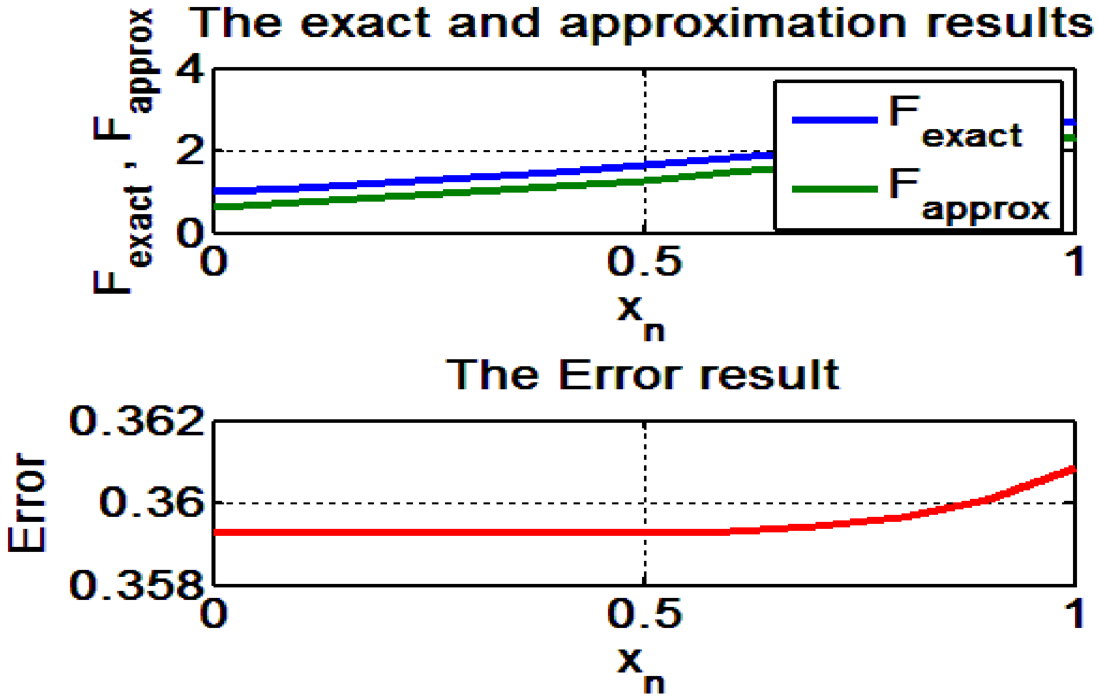
The CPU time is equal to 2.0904 Seconds

As the above table show, we reached the exact solution after 20 iterations.

Table (8)

The exact and approximate solution of the modified technique via matrices

N	x_n	The exact sol	The approximate sol	The error
0	0	1	0.640745949318671	0.359254050681329
1	0.1	1.10517091807565	0.745916865985337	0.359254052090310
2	0.2	1.22140275816017	0.862148615985337	0.359254142174833
3	0.3	1.34985880757600	0.990603699318670	0.359255108257333
4	0.4	1.49182469764127	1.13256461598534	0.359260081655934
5	0.5	1.64872127070013	1.28944386598534	0.359277404714792
6	0.6	1.82211880039051	1.46279394931867	0.359324851071841
7	0.7	2.01375270747048	1.65431736598533	0.359435341485143
8	0.8	2.22554092849247	1.86587661598533	0.359664312507136
9	**0.9**	2.45960311115695	2.09950419931866	0.360098911838288
10	**1**	2.71828182845905	2.35741261598533	0.360869212473721



Figure(4)

4. Conclusion

A simple Iterative Algorithm for the solution of Fredholm Integral Equations of the second kind has been presented. The given method gives a very simple form for the iterated kernels via the well - known Hilbert matrix. Thus, the iterative solutions of an integral equation of the second kind can be reduced to the solution of a matrix equation, whereas only one coefficient matrix is required to be computed. Therefore, computational complexity can be considerably reduced and much computational time can be saved. The new proposed approach needs a small number of iterations to provide an exact result, that proofs the power of the presented Algorithm, and stimulates to find out the relation between the integral equations and Hilbert Matrix. Briefly words the Advantages of the given modified method Maye be concluded as

- Decrease the CPU time In a spectacular way.
- less number of iterations to reach the iterative solution
- Easy to compute because only the kernel matrix is required to find the iterative solution
- It is more powerful if the kernel function is complicated as in our new method we differentiate the kernel to get the solution but in the old one we integrate

References

[1] Andrei D. Polyanin & Alexander V. Manzhirov, "*Handbook of Integral Equations* " ,CRC Press 2008.

[2] Burden Richard L. & Faires J. *Douglas "Numerical Analysis"*, PWS Publishing Company, Boston,2010.

[3] C.T.H. Baker, *The Numerical Treatment of Integral Equations, Clarendom Press,* Oxford , 4th Edition 1977.

[4] Guangqing L. , Gnaneshwar N., *Iteration methods for Fredholm integral equations of the second kind* ,Computers & Mathematics with Applications, Volume 53, Issue 6, March 2007, Pages 886–894.

[5] Graham I.G., S. Joe, I.H. Sloan, *Iterated Galerkin versus iterated collocation for integral equations of second kind,* IMA J. Numer. Anal., 5 (1985), pp. 355–369.

[6] k.E. Atkinson, *The numerical Solution of Integral Equations of the second kind,* Cambridge University pree, Cambridge, 1977.

[7] M .C .De Bonis ,C .Laurita,"*Numerical Treatment of Second Kind Fredholm Integral Equations Systems on Bounded Intervals,* Journal of Computational and Applied Mathematics, 2008, Pages 64-87.

[8] H. Kaneko, Y. Xu, *Super convergence of the iterated Galerkin methods for Hammerstein equations,* SIAM J. Numer. Anal., 33 (1996), pp. 1048–1064.

[9] Shoukralla, E. S., EL-Serafi, S. A., Saber, Nermein A,"*A matrix Iterative technique for the solution of Fredholm Integral Equations of the second kind*" "*Electronic Journal of Mathematical Analysis and Applications*" Vol. 4(1) Jan. 2016, pp. 192-196.ISSN: 2090-729.

[10] Shoukralla, E. S., "*A Algorithm For The Solution Of a Certain Singular Integral Equation Of The First Kind* ", Intern. J. Computer Math., Vol. 69, pp. 165-173, Gordon and Breach Science, England, June 1998.

[11] Shoukralla, E. S., "*Approximate Solution to weakly singular integral equations*", J. Appl. Math. Modeling, Elsevier, England, Vol. 20, pp. 800 -803, November 1996.

[12] S.S. Bellale and S.B. Birajdar, *On Existence Theorem for Nonlinear Functional Two Point Boundary Value Problems in Banach Algebras,* Int. Jr. Universal Mathematics and Mathematical Sciences 1(1) (2015), 47-58.

[13] S.S. Bellale, S.B. Birajdar and D.S. Palimkar, *Existence Theorem for Abstract Measure Delay Integro-Differential Equations,* Applied and Computational Mathematics 4(4) (2015), 225-231, doi:10.11648/j.acm.20150404.11.

[14] Verlane A. F., Cazakov V. C.,"*Integral Equations*", Nauka Domka, Kiev, USSR, 1986.

[15] Z. Chen, C.A. Micchelli, Y. Xu, *Fast collocation methods for second kind integral equations,* SIAM J. Numer. Anal., 40 (2002), pp. 344–375.

[16] Sajane B.A., *New Extensions of Fintehankel Type Transformation on Generalized Function,* International Journal of Mathematics and Mathematical Sciences, 1(1) (2015) 19-27.