

Romberg Algorithm to Solve a System of Nonlinear Fredholm Integral Equations of Second Kind

Abbas Y. Al-Bayati

College of Computer Sciences and
Mathematics
University of Mosul

Rostam K. Saeed

College of Sciences
University of Salahaddin

Kawa M. Aziz

Received on: 01/10/2007

Accepted on: 17/12/2007

المخلص

في هذا البحث استخدمنا خوارزمية رومبيرك المعروفة لحل منظومة من معادلات فريدهولم التكاملية اللاخطية من النوع الثاني وقد تم الحصول على نتائج جيدة باستخدام الحل المطلوب وتمت المقارنة باستخدام أربعة مسائل عدديّة وبالاعتماد على الزمن المضبوط المستغرق للحاسبة (CPU time) مع استخدام خاصية مربع أقل خطأ (Least-square error). تم كتابة برنامج حاسوبي باستخدام الـ Matlab 7.0 وللأبعاد 2×2 و 3×3 وعلى حاسبة شخصية P4.

ABSTRACT

In this paper, we use Romberg algorithm, to find an approximation solution for a system of nonlinear Fredholm integral equations of the second kind. A comparison between this approximations and exact solutions for four numerical examples are given, depending on the least-square error and running computer time. Our numerical results are obtained by MATLAB 7.0 program and for a system of rank 2×2 , 3×3 on Personal Computer P4.

Keywords: System of Nonlinear Integral Equations, Romberg Algorithm

1. Introduction:

Integral equations are used as mathematical models for many physical situations (Partial Differential Equations) and integral equations also occur as reformulations of other mathematical problems. Recently, many of the authors and co-workers are working in the system of nonlinear Fredholm integral equations of the second kind such as [1], [2] and [6].

Let us consider the following model for a system of nonlinear Fredholm integral equations of second kind

$$u_i(x) = f_i(x) + \int_a^b k_i(x, t, u_1(t), u_2(t), \dots, u_m(t)) dt, \text{ for } i=1, 2, \dots, m, \quad \dots(1)$$

where $u_i(x)$, $i=1, 2, \dots, m$ are the unknown functions which is to be determined in the interval $[a, b]$; The functions $f_i(x)$ and the kernels $k_i(x, t, u_1(t), u_2(t), \dots, u_m(t))$ are also given [8].

In this work, Romberg integration algorithm [4] has been used for solving the system of nonlinear Fredholm integral equations of the second kind defined in eq. 1.

2. Romberg Integration [3, 4]:

Let $T(f, h)$, $S(f, h)$ and $B(f, h)$ denote the trapezoidal rule, Simpson's rule and Boole's rule, respectively, with step size h . The order of their error terms is

$$\int_a^b f(x)dx = T(f, h) - O(h^2)$$

$$\int_a^b f(x)dx = S(f, h) - O(h^4) \quad \dots(2)$$

$$\int_a^b f(x)dx = B(f, h) - O(h^6)$$

where the error terms are:

$$O(h^2) = \frac{-(b-a)h^2 f^{(2)}(c)}{12}$$

$$O(h^4) = \frac{-(b-a)h^4 f^{(4)}(c)}{180} \quad \dots(3)$$

$$O(h^6) = \frac{-2(b-a)h^6 f^{(6)}(c)}{945}$$

The pattern in (2) can be extended in the following sense. If an approximation rule is used with step sizes h and $2h$, then an algebraic manipulation of the two answers can be used to produce an improved answer. Each successive level of improvement increases the order of the error term from $O(h^{2N})$ to $O(h^{2N+2})$, this process called the Romberg integration [5].

The development of Romberg integration relies on the theoretical assumption that $f(x)$ is smooth enough so that the error in the trapezoidal rule can be expanded in a series involving only even power of h , that is,

$$\int_a^b f(x)dx = T(f, h) + E(f, h) \quad \dots(4)$$

where

$$E(f, h) = a_1 h^2 + a_2 h^4 + a_3 h^6 + \dots, \quad \dots(5)$$

a_1, a_2, a_3, \dots are the coefficients of the integral

The *Romberg algorithm* produces a triangular array of numbers, all of which are numerical estimates of the definite integral $\int_a^b f(x) dx$. The array is denoted here by:

$$\begin{array}{ccccccc} R(1,1) & & & & & & \\ R(2,1) & R(2,2) & & & & & \\ R(3,1) & R(3,2) & R(3,3) & & & & \\ R(4,1) & R(4,2) & R(4,3) & R(4,4) & & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & & \\ R(m,1) & R(m,2) & R(m,3) & R(m,4) & \dots & R(m,m) & \end{array} \quad \dots(6)$$

The first column of this table contains estimates of the integral obtained by recursive trapezoid formula with decreasing values of the stepsize. Explicitly, that is $R(m,1)$ is the result of applying the trapezoid rule with 2^m equal subintervals. The first of them $R(1,1)$ is obtained with just one trapezoid:

$$R(1,1) = \frac{1}{2}(b-a)[f(a) + f(b)] \quad \dots(7)$$

Similarly, $R(2,1)$ is obtained with two trapezoids:

$$\begin{aligned} R(2,1) &= \frac{1}{4}(b-a) \left[f(a) + f\left(\frac{a+b}{2}\right) \right] + \frac{1}{4}(b-a) \left[f\left(\frac{a+b}{2}\right) + f(b) \right] \\ &= \frac{1}{2}R(1,1) + \frac{1}{2}(b-a) f\left(\frac{a+b}{2}\right) \end{aligned} \quad \dots(8)$$

Thus, $R(m,1)$ is obtained easily from $R(m-1,1)$; that is

$$R(m,1) = \frac{1}{2}R(m-1,1) + h \sum_{k=1}^{2^{m-2}} f(a + (k-0.5)*h) \quad \dots(9)$$

where $h = (b-a) / 2^m$ and $m \geq 2$.

The second and successive column in the Romberg array are generated by extrapolation formula

$$R(i, j) = R(i, j-1) + \frac{1}{4^{j-1} - 1} [R(i, j-1) - R(i-1, j-1)] \quad \dots(10)$$

where $i = 2, 3, \dots, m$ and $j = 2, \dots, i$.

3. Outlines of Romberg Algorithm to solve (eq. 1) [4]

First we find integrals of (eq.1) by Romberg algorithm and summarized by the following steps:

1) Input m , put $h_1 = b - a$.

2) Put $dd = \sum_{i=2}^m 2^{i-2} + 2$, $dd_1 = 2$.

3) Find $R_i(1,1) = \frac{h_1}{2} [k_i(x, a, r_{i1}, r_{21}, \dots, r_{n1}) + k_i(x, b, r_{1,dd}, r_{2,dd}, \dots, r_{n,dd})]$,

and

$$R_i(j,1) = \frac{1}{2} \left[R_i(j-1,1) + h_{j-1} \sum_{jj=1}^{2^{j-2}} k_i(x, a + (jj-0.5)h_{j-1}, r_{1,dd_{jj}}, r_{2,dd_{jj}}, \dots, r_{n,dd_{jj}}) \right],$$

where $r_{i1} = u_i(a)$; $r_{i,dd} = u_i(b)$; $r_{i,dd_{ss}} = u_i(a + (ss-0.5)h_{j-1})$,

$dd_{ss+1} = dd_{ss} + 1$ for $ss = 1, 2, \dots, dd-1$; $i = 1, 2, \dots, n$,

and $h_j = \frac{h_{j-1}}{2}$ for $j = 2, \dots, m$.

4) Find $R_i(j, jj) = R_i(j, jj-1) + \frac{1}{4^{jj-1} - 1} [R_i(j, jj-1) - R_i(j-1, jj-1)]$

for $i = 1, 2, \dots, n$.

Second reduce (eq.1) to:

$$u_i(x) = f_i(x) + R_i(m, m)$$

(11)

where $R_i(m, m)$ are functions of unknowns $x, r_{i1}, r_{i2}, \dots, r_{i,dd}$ for $i = 1, 2, \dots, n$.

Now, put $x_1 = a$, $x_j = a + (j-0.5)h$, for $j = 2, 3, \dots, dd-1$ and $x_{dd} = b$. In (eq. 11) replace x by x_j ; $j = 1, 2, 3, \dots, dd$ we get $L = i.dd$ nonlinear equations in the $L = i.dd$ unknowns $r_{i1}, r_{i2}, \dots, r_{i,dd}$ for $i = 1, 2, \dots, n$. And we can solve this nonlinear system by modified Newton-Raphson method [7] to find the values of unknowns

$r_{i1}, r_{i2}, \dots, r_{i,dd}$ for $i = 1, 2, \dots, n$ and substituted in the (eq. 11) we get the approximate solution of (eq. 1).

4. Numerical Results:

In this part we present four numerical examples. The first and second are 2x2 system with integrable integrals, if we solve the system by another method such as weighted residual methods, spline function methods, or any approximation method, while the third example is 2x2 system the integrals are not integrable if we solve by another method and the fourth example is 3x3 system where $m =$ number of columns in Romberg tables.

$It =$ number of iterations in Modified Newton-Raphson (Quadratically Convergent) method for solving nonlinear system or the number of rows in Romberg table.

L.S.E=Least square error.

R.T. =running time (CPU time).

Example 4.1: [2]

Consider the following system of nonlinear Fredholm integral equations of the second kind; with the exact solutions $u_1(x) = x$ and $u_2(x) = x^2$

$$u_1(x) = x - \frac{5}{18} + \int_0^1 \frac{1}{3}(u_1(t) + u_2(t)) dt$$

$$u_2(x) = x^2 - \frac{2}{9} + \int_0^1 \frac{1}{3}(u_1^2(t) + u_2(t)) dt .$$

We can find an approximation solution by applying the newly written program of this algorithm as follow:

```
>>> RombergforsolveSNFIESK2x2(0,1,2,10,'-u1+x-(5/18)', '(1/3)*(ut1+ut2)', '-u2+x^2-(2/9)', '(1/3)*(ut1^2+ut2)')
```

We get:

$$u_1(x) = x - .2265548246e - 4$$

$$u_2(x) = x^2 - .2097283615e - 4$$

Also if we choose $m = 3$ and $It = 15$; approximation to the solution is:

$$u_1(x) \approx x - .7247519682e - 6 \text{ and}$$

$$u_2(x) \approx x^2 - .6812422154e - 6$$

and if we choose $m = 4$ and $It = 30$; approximation to the solution is:

$$u_1(x) \approx x - .7590210880e - 11 \text{ and}$$

$$u_2(x) \approx x^2 - .7228766277e - 11$$

Some numerical values of this solution are presented in the following table:

Table (4.1)

x	Exact value		Approximate value of $u_1(x)$		Approximate value of $u_2(x)$	
	$u_1(x)$	$u_2(x)$	$m = 3; It = 15$	$m = 4; It = 30$	$m = 3; It = 15$	$m = 4; It = 30$
0	0	0	-7.24751e-7	-7.59030e-12	-6.81242e-7	-7.22878e-12
0.1	0.1	0.01	0.0999992752	0.0999999999	0.0099993187	0.0099999999
0.2	0.2	0.04	0.1999992752	0.1999999999	0.0399993187	0.0399999999
0.3	0.3	0.09	0.2999992752	0.2999999999	0.0899993187	0.0899999999
0.4	0.4	0.16	0.3999992752	0.3999999999	0.1599993187	0.1599999999

0.5	0.5	0.25	0.4999992752	0.4999999999	0.2499993187	0.2499999999
0.6	0.6	0.36	0.5999992752	0.5999999999	0.3599993187	0.3599999999
0.7	0.7	0.49	0.6999992752	0.6999999999	0.4899993187	0.4899999999
0.8	0.8	0.64	0.7999992752	0.7999999999	0.6399993187	0.6399999999
0.9	0.9	0.81	0.8999992752	0.8999999999	0.8099993187	0.8099999999
1	1	1	0.9999992752	0.9999999999	0.9999993187	0.9999999999
L.S.E			5.77791e-12	6.33720e-22	5.10500e-12	5.74804e-22
R.T.			00:00:02	00:00:04	00:00:02	00:00:04

Example 4.2 [2]:

Consider the following system

$$u_1(x) = \exp(-2) \left(0.5x - \frac{3}{4}\right) - x + 0.25 + \int_0^1 (x-t) u_2^2(t) dt$$

$$u_2(x) = \exp(-x) - 1 + \int_0^1 t \exp(-2u_1(t)) dt$$

With the exact solution $u_1(x) = -\frac{1}{2}x$ and $u_2(x) = \exp(-x)$.

Solution: If we choose:

I. $m = 3$ and $It = 30$; approximation to the solution is:

$$u_1(x) \approx -.4999391630 * x + .3381669305e - 4$$

$$u_2(x) \approx \exp(-1.* x) - .1324241319e - 3$$

II. $m = 4$ and $It = 50$; approximation to the solution is:

$$u_1(x) \approx -.5000000318 * x + .9932666759e - 7$$

$$u_2(x) \approx \exp(-1.* x) - .1356290455e - 6$$

Some numerical values of this solution are presented in the following table:

Table (4.2)

x	Exact value of		Approximate value of $u_1(x)$		Approximate value of $u_2(x)$	
	$u_1(x)$	$u_2(x)$	$m = 3; It = 30$	$m = 4; It = 50$	$m = 3; It = 30$	$m = 4; It = 50$
0	0	1	3.38166e-5	9.93266e-8	0.999867575	0.999999864
0.1	-0.05	0.904837418	-	-	0.904704993	0.904837282
0.2	-0.1	0.818730753	-	-	0.818598328	0.818730617
0.3	-0.15	0.740818220	-	-	0.740685796	0.740818085
0.4	-0.20	0.670320046	-	-	0.670187621	0.670319910
0.5	-0.25	0.606530659	-	-	0.606398235	0.606530524
0.6	-0.30	0.548811636	-	-	0.548679211	0.548811500
0.7	-0.35	0.496585303	-	-	0.496452879	0.496585168
0.8	-0.40	0.449328964	-	-	0.449196539	0.449328828
0.9	-0.45	0.406569659	-	-	0.406437235	0.406569524
1	-0.5	0.367879441	-	-	0.367747017	0.367879305
L.S.			4.94589e-8	7.77076e-14	1.92897e-7	2.02347e-13
R.T.			00:00:04	00:00:07	00:00:04	00:00:07

Example 4.3 [3]:

Consider the following system

$$u_1(x) = \sin(x) - 0.5 \exp(1) + \frac{7}{8} + \int_0^1 0.5 \left(\frac{t^2}{t + \exp(t)} \right) u_2^2(t) dt$$

$$u_2(x) = x + \exp(x) + \cos(1) - 1 + \int_0^1 u_1(t) dt.$$

With the exact solution $u_1(x) = \sin(x)$ and $u_2(x) = x + \exp(x)$.

Solution: If we choose:

I. $m = 2$ and $It = 10$; approximation to the solution is:

$$u_1(x) \approx \sin(x) + .7239700457e - 2$$

$$u_2(x) \approx x + \exp(x) + .7384419184e - 2$$

II. $m = 5$ and $It = 25$; approximation to the solution is:

$$u_1(x) \approx \sin(x) - .9465486392e - 12$$

$$u_2(x) \approx x + \exp(x) - .4611766377e - 11$$

Some numerical values of this solution are presented in the following table:

Table (4.3)

x	Exact value of		Approximate value		Approximate value	
	$u_1(x)$	$u_2(x)$	$m = 2; It = 10$	$m = 5; It = 25$	$m = 2; It = 10$	$m = 5; It = 25$
0	0	1	0.00723970	-9.46576e-	1.00738441	0.99999999
0.1	0.09983341	1.20517091	0.10707311	0.09983341	1.21255533	1.20517091
0.2	0.19866933	1.42140275	0.20590903	0.19866933	1.42878717	1.42140275
0.3	0.29552020	1.64985880	0.30275990	0.29552020	1.65724322	1.64985880
0.4	0.38941834	1.89182469	0.39665804	0.38941834	1.89920911	1.89182469
0.5	0.47942553	2.14872127	0.48666523	0.47942553	2.15610568	2.14872127
0.6	0.56464247	2.42211880	0.57188217	0.56464247	2.42950321	2.42211880
0.7	0.64421768	2.71375270	0.65145738	0.64421768	2.72113712	2.71375270
0.8	0.71735609	3.02554092	0.72459579	0.71735609	3.03292534	3.02554092
0.9	0.78332690	3.35960311	0.79056661	0.78332690	3.36698753	3.35960311
1	0.84147098	3.71828182	0.84871068	0.84147098	3.72566624	3.71828182
L.S			5.76545e-4	9.85530e-	5.99826e-4	2.33935e-
R.			00:00:01	00:00:09	00:00:01	00:00:09

Example 4.4 [3]:

Consider the following system

$$u_1(x) = \frac{x}{2} - \frac{1}{4} + \int_0^1 [x u_1(t) + t u_2(t)] dt$$

$$u_2(x) = x^2 - \frac{2}{9} + \int_0^1 \frac{1}{3} (u_3^2(t) + u_2(t)) dt$$

$$u_3(x) = x - \frac{2}{15} + \int_0^1 \frac{1}{5} (u_1(t) u_3(t) + u_2(t)) dt$$

With the exact solution $u_1(x) = x$, $u_2(x) = x^2$ and $u_3(x) = x$.

Solution: If we choose $m = 4$ and $It = 10$, the approximation to the solution is

$$u_1(x) \approx 1.003093978 * x + .1679859133e - 2$$

$$u_2(x) \approx .3401851590e - 2 + x^2$$

$$u_3(x) \approx .6850325302e - 2 + x.$$

By running the newly written program which is (in appendix) as:

```
>>RombergforsolveSNFIESK3x3(0,1,5,20,'-u1+(x/2)-(1/4)', 'x*ut1+t*ut2', '-u2+x^2-(2/9)', '(1/3)*(ut3^2+ut2)', '-u3+x-(2/15)', '(1/5)*(ut1*ut3+ut2)')
```

Some numerical values of this solution are presented in the following table:

Table (4.4)

x	Exact value of			Approximate value of		
	$u_1(x)$	$u_2(x)$	$u_3(x)$	$u_1(x)$	$u_2(x)$	$u_3(x)$
0	0	0	0	0.0016798591	0.0034018515	0.0068503253
0.1	0.1	0.01	0.1	0.1019892569	0.0134018515	0.1068503253
0.2	0.2	0.04	0.2	0.2022986547	0.0434018515	0.2068503253
0.3	0.3	0.09	0.3	0.3026080526	0.0934018515	0.3068503253
0.4	0.4	0.16	0.4	0.4029174504	0.1634018515	0.4068503253
0.5	0.5	0.25	0.5	0.5032268482	0.2534018515	0.5068503253
0.6	0.6	0.36	0.6	0.6035362461	0.3634018515	0.6068503253
0.7	0.7	0.49	0.7	0.7038456439	0.4934018515	0.7068503253
0.8	0.8	0.64	0.8	0.8041550417	0.6434018515	0.8068503253
0.9	0.9	0.81	0.9	0.9044644396	0.8134018515	0.9068503253
1	1	1	1	1.0047738374	1.0034018515	1.0068503253
L.S.				1.25068e-4	1.27298e-4	5.16196e-4
R.T				00:00:07		

5. Discussions and Conclusions

Romberg integration algorithm has been known to be powerful device for evaluating integrals for many times. Here we use this algorithm for solving a system of nonlinear integral equations of the second kind because it's construction involved integrations. The real hard part of this algorithm is to construct a nonlinear system with respect to some unknowns. Furthermore, a new general program for implementing this algorithm and by increasing the number of rows in Romberg table or the number of iterations in the modified Newton Raphson method (if it is used instead of Romberg algorithm) can reach the desired accuracy. The present algorithm has successfully been applied to the system of nonlinear Fredholm integral equations of the second kind. Extensions of this algorithm for solving another system of nonlinear integro-differential equations offers an excellent opportunity for future research.

From our numerical results, tables (4.1, 4.2, 4.3 and 4.4) indicate that the approximation solution of $u_1(x)$ at $m=5$, $it=25$ is very successful even it requires more running time than the case of $m=2$, $it=10$. Also the approximation solution of $u_2(x)$ at $m=4$, $it=30$ is very successful even it requires an approximately the same running time than $m=3$, $it=15$.

Finally the appendix of this paper contains the main steps of the MATLAB program 7.0 to solve a system of rank 2×2 and 3×3 .

REFERENCES

- [1] Aziz, K. M., (2007), "An Approximation Solution for Solving the System of Nonlinear Fredholm Integral Equations of the Second Kind", M.Sc. thesis University of Salahaddin-Erbil/College of Science.
- [2] Babolian, E., Biazar, J. and Vahidi, A. R., (2004) "The decomposition method applied to systems of Fredholm integral equations of the second kind", Applied Mathematics and Computation, 148, pp.443-452.
- [3] Burden, R. L. and Faires, J. D. , (2001) "Numerical Analysis Seventh Edition", Australia, Canada.
- [4] Cheney, W. and Kincaid, D., (2004) "Numerical Mathematics and Computing, Fifth Edition", Brooks /Cole Publishing company.
- [5] Delves, L. M. and Mohamed, J. L., (1985) "Computational Methods for Integral Equations ", Cambridge university Press.
- [6] Ivaz, K. , Mostahkam, B. S., (2006) "Newton-Tau Numerical Solution of a system of Nonlinear Fredholm Integral Equations of Second Kind ", Appl. Comput. Math. 5, pp.201-208.
- [7] Stark, P. A., (1970) "Introduction to Numerical Methods", Macmillan Publishing Co., Inc. New York.
- [8] Waz Waz, A., (2005), "The Modified Decomposition Method for Analytic Treatment of Non-linear Integral Equations and Systems of Non-linear Integral Equations", International Journal of Computer Mathematics, 82, pp. 1107-1115.

7. Appendix:

7.1 General Program for Solving 2×2 Systems as Follows:

```

function
ff2x2=RombergforsolveSNFIESK2x2(a,b,m,It,eq11,eq12,eq21,eq2
2)
syms u1 u2 ut1 ut2 x t
format long
h=b-a;dd=0;
tic
for i=2:m
    dd=dd+2^(i-2);
end
dd=dd+2;
y(1)=a;
y(dd)=b;
R1(1,1)=(h/2)*(subs(eq12,{ut1,ut2,t},{['r1'
int2str(1)],['r2'
int2str(1)],y(1)})+subs(eq12,{ut1,ut2,t},{['r1'
int2str(dd)], ['r2' int2str(dd)],y(dd)}));
R2(1,1)=(h/2)*(subs(eq22,{ut1,ut2,t},{['r1'
int2str(1)],['r2'
int2str(1)],y(1)})+subs(eq22,{ut1,ut2,t},{['r1'
int2str(dd)], ['r2' int2str(dd)],y(dd)}));
dd1=2;
for i=2:m
    ss1=0;
    ss2=0;
    for j=1:2^(i-2)
        y(dd1)=a+(j-0.5)*h;
        ss1=ss1+subs(eq12,{ut1,ut2,t},{['r1'
int2str(dd1)], ['r2' int2str(dd1)],y(dd1)});
        ss2=ss2+subs(eq22,{ut1,ut2,t},{['r1'
int2str(dd1)], ['r2' int2str(dd1)],y(dd1)});
        dd1=dd1+1;
    end
    ss1=ss1*h;
    ss2=ss2*h;
    R1(i,1)=0.5*(R1(i-1,1)+ss1);
    R2(i,1)=0.5*(R2(i-1,1)+ss2);
    for j=2:i
        R1(i,j)=R1(i,j-1)+((R1(i,j-1)-R1(i-1,j-
1))/(4^(j-1))-1));
        R2(i,j)=R2(i,j-1)+((R2(i,j-1)-R2(i-1,j-
1))/(4^(j-1))-1));
    end
    h=h/2;
end
for i=1:dd

```

```

    Res1(i)=subs(eq11,{u1,x},{['r1'
int2str(i)],y(i)})+subs(R1(m,m),x,y(i));
    Res2(i)=subs(eq21,{u2,x},{['r2'
int2str(i)],y(i)})+subs(R2(m,m),x,y(i));
end
for j = 1:dd
    eval(['r1',int2str(j),'=0']);
    eval(['r2',int2str(j),'=0']);
end
for jj=1:It
    for i=1:dd
        l=eval(['r1' int2str(i)]);
        ddiff1=eval(diff(Res1(i),['r1' int2str(i)]));
        eval(['r1',int2str(i),'=1-
(eval(Res1(i))/ddiff1)']);
        l=eval(['r2' int2str(i)]);
        ddiff2=eval(diff(Res2(i),['r2' int2str(i)]));
        eval(['r2',int2str(i),'=1-(eval(Res2(i))/ddiff2)']);
    end
end
    u1=0;u2=0;
    vpa(eval(eq11+R1(m,m)),10)
    vpa(eval(eq21+R2(m,m)),10)
toc

```

7.2 General Program for Solving 3×3 Systems as Follows:

```

function
ff2x2=RombergforsolveSNFIESK3x3(a,b,m,It,eq11,eq12,eq21,eq2
2,eq31,eq32)
syms u1 u2 ut1 ut2 x t u3 ut3
format long
h=b-a;dd=0;
tic
for i=2:m
    dd=dd+2^(i-2);
end
dd=dd+2;
y(1)=a;
y(dd)=b;
R1(1,1)=(h/2)*(subs(eq12,{ut1,ut2,ut3,t},{['r1'
int2str(1)],['r2' int2str(1)],['r3'
int2str(1)],y(1)})+subs(eq12,{ut1,ut2,ut3,t},{['r1'
int2str(dd)], ['r2' int2str(dd)],['r3'
int2str(dd)],y(dd)}));
R2(1,1)=(h/2)*(subs(eq22,{ut1,ut2,ut3,t},{['r1'
int2str(1)],['r2' int2str(1)],['r3'
int2str(1)],y(1)})+subs(eq22,{ut1,ut2,ut3,t},{['r1'
int2str(dd)], ['r2' int2str(dd)],['r3'
int2str(dd)],y(dd)}));

```

```

R3(1,1)=(h/2)*(subs(eq32,{ut1,ut2,ut3,t},{['r1'
int2str(1)],['r2' int2str(1)],['r3'
int2str(1)],y(1)}+subs(eq22,{ut1,ut2,ut3,t},{['r1'
int2str(dd)], ['r2' int2str(dd)],['r3'
int2str(dd)],y(dd)}));
dd1=2;
for i=2:m
    ss1=0;ss2=0;ss3=0;
    for j=1:2^(i-2)
        y(dd1)=a+(j-0.5)*h;
        ss1=ss1+subs(eq12, {ut1,ut2,ut3,t}, {['r1'
int2str(dd1)], ['r2' int2str(dd1)],['r3'
int2str(dd1)],y(dd1)});
        ss2=ss2+subs(eq22, {ut1,ut2,ut3,t}, {['r1'
int2str(dd1)], ['r2' int2str(dd1)],['r3'
int2str(dd1)],y(dd1)});
        ss3=ss3+subs(eq32, {ut1,ut2,ut3,t}, {['r1'
int2str(dd1)], ['r2' int2str(dd1)],['r3'
int2str(dd1)],y(dd1)});
        dd1=dd1+1;
    end
    ss1=ss1*h;
    ss2=ss2*h;
    ss3=ss3*h;
    R1(i,1)=0.5*(R1(i-1,1)+ss1);
    R2(i,1)=0.5*(R2(i-1,1)+ss2);
    R3(i,1)=0.5*(R3(i-1,1)+ss3);
    for j=2:i
        R1(i,j)=R1(i,j-1)+((R1(i,j-1)-R1(i-1,j-
1)))/((4^(j-1))-1));
        R2(i,j)=R2(i,j-1)+((R2(i,j-1)-R2(i-1,j-
1)))/((4^(j-1))-1));
        R3(i,j)=R3(i,j-1)+((R3(i,j-1)-R3(i-1,j-
1)))/((4^(j-1))-1));
    end
    h=h/2;
end
for i=1:dd
    Res1(i)=subs(eq11,{u1,x},{['r1'
int2str(i)],y(i)}+subs(R1(m,m),x,y(i)));
    Res2(i)=subs(eq21,{u2,x},{['r2'
int2str(i)],y(i)}+subs(R2(m,m),x,y(i)));
    Res3(i)=subs(eq31,{u3,x},{['r3'
int2str(i)],y(i)}+subs(R3(m,m),x,y(i)));
end
for j = 1:dd
    eval(['r1',int2str(j),'=0']);
    eval(['r2',int2str(j),'=0']);
    eval(['r3',int2str(j),'=0']);
end

```

```
for jj=1:It
    for i=1:dd
        l=eval(['r1' int2str(i)]);
        ddiff1=eval(diff(Res1(i),['r1' int2str(i)]));
        eval(['r1',int2str(i),'=1-
(eval(Res1(i))/ddiff1)']);
        l=eval(['r2' int2str(i)]);
        ddiff2=eval(diff(Res2(i),['r2' int2str(i)]));
        eval(['r2',int2str(i),'=1-(eval(Res2(i))/ddiff2)']);
        l=eval(['r3' int2str(i)]);
        ddiff3=eval(diff(Res3(i),['r3' int2str(i)]));
        eval(['r3',int2str(i),'=1-
(eval(Res3(i))/ddiff3)']);
    end
end
    u1=0;u2=0;err1=0;
    vpa(eval(eq11+R1(m,m)),10)
    vpa(eval(eq21+R2(m,m)),10)
    vpa(eval(eq31+R3(m,m)),10)
toc
```