CHAPTER - 3

NUMERICAL METHODS FOR SOLVING SYSTEM OF GENERALISED ABEL INTEGRAL EQUATIONS BY OPERATIONAL MATRICES

Chapter 3

Numerical Methods for Solving System of Generalised Abel Integral Equations by Operational Matrices

3.1 Introduction

In this chapter we deal with study of linear as well as nonlinear system of generalized Abel integral equations. We consider the system of integral equations, namely

$$\left.\begin{aligned} a_{11}(x)\int_0^x \frac{F_1(u_1(t),u_2(t))dt}{(x-t)^b} + a_{12}(x)\int_x^1 \frac{F_2(u_1(t),u_2(t))dt}{(t-x)^b} = f_1(x) \\ a_{21}(x)\int_0^x \frac{F_3(u_1(t),u_2(t))dt}{(t-x)^b} + a_{22}(x)\int_x^1 \frac{F_4(u_1(t),u_2(t))dt}{(x-t)^b} = f_2(x) \end{aligned}\right\}, \tag{3.1.1}$$

where $x, b \in (0,1)$, $a_{ij}(x)$, $i = 1,2$ and $j = 1,2$ are continuous on $[0,1]$, f_1 and f_2 are the forcing terms. Assuming $F_1(u_1(t),u_2(t)) = F_3(u_1(t),u_2(t))$ and $F_2(u_1(t),u_2(t)) = F_4(u_1(t),u_2(t))$ to be $u_1(t)$ and $u_2(t)$ respectively, we obtain the well-known system of generalized Abel integral equations

$$\left.\begin{aligned} a_{11}(x)\int_0^x \frac{u_1(t)dt}{(x-t)^b} + a_{12}(x)\int_x^1 \frac{u_2(t)dt}{(t-x)^b} = f_1(x) \\ a_{21}(x)\int_0^x \frac{u_1(t)dt}{(t-x)^b} + a_{22}(x)\int_x^1 \frac{u_2(t)dt}{(x-t)^b} = f_2(x) \end{aligned}\right\}. \tag{3.1.2}$$

This system of generalized integral equations of Abel type was studied by Lowengrub (1976) and Walton (1979). As stated in Walton (1979), certain mixed boundary value problems arising in the classical theory of elasticity reduce to the problem of determining functions u_1 and u_2 satisfying the above Abel integral equations. Mandal *et al.* (1996) solved this problem analytically for the special case $b = 1/2, a_{ij}(x) = 1,$ using the idea of fractional calculus.

Recently Pandey and Mandal (2010) obtained a numerical solution of the above system Eq. (3.1.2) using Bernstein polynomials of order 8. Singh *et al.* (2010) constructed an operational matrix of integration using orthonormal Bernstein polynomials and used it to propose a stable algorithm to solve Eq. (3.1.2). They considered linear type Abel integral equations only but the method proposed by Pandey and Mandal (2010) is less accurate when the coefficients $a_{ij}(x)$ are polynomial functions compared to the case when the coefficients are constants. This along with the results obtained in Singh *et al.* (2010) motivated us for the present work. Yousefi (2006) has provided Legendre wavelet based method for solving Abel integral equations. In Pandey *et al.* (2009) they have discussed analytical methods like homotopy perturbation method (HPM), modified homotopy perturbation method (MHPM), adomian decomposition method (ADM) and modified adomian decomposition method (MADM) for solving Abel integral equations. Further, Huang *et al.* (2008) discussed an approximate method for solving Abel integral equation by approximating the unknown function using Taylor series. Singh *et al.* (2008) constructed an operational matrix of integration based on orthonormal Bernstein polynomials, and used it to propose an algorithm for solving the Abel's integral equation.

In chapter 2, we have already constructed the operational matrices for fractional integration and differentation (Theorems 2.3.1., 2.3.2. and 2.3.3.) using the properties of Legendre's scaling functions. In the continuation of that, the present chapter 3 deals with the situation of nonlinearity in the generalized form of Abel integral equations. Following the same basic facts and properties about the Legendre's scaling functions as discused in chapter 2, we have directly obtained the numerical algorithm for the solution of the problem. We also establish the convergence of the approximate solutions to the true solution under certain mild condition followed by numerical experiments, in sub sequent sections.

3.2 Numerical Algorithm

Type 1:

In type first we solve System of Generalized Abel's integral equation which is given by

$$\left.\begin{aligned} a_{11}(x)\int_0^x \frac{(u_1)^a(t)dt}{(x-t)^b} + a_{12}(x)\int_x^1 \frac{(u_2)^c(t)dt}{(t-x)^b} = f_1(x) \\ a_{21}(x)\int_0^x \frac{(u_1)^a(t)dt}{(t-x)^b} + a_{22}(x)\int_x^1 \frac{(u_2)^c(t)dt}{(x-t)^b} = f_2(x) \end{aligned}\right\}, \tag{3.2.1}$$

where $(u_1)^a(t) = (u_1(t))^a$ and $(u_2)^c(t) = (u_2(t))^c$.

Approximating various terms involved in the above integral by their Legendre's scaling function approximations as

$$(u_1)^a(t) = C_1^T\phi(t),\ (u_2)^c(t) = C_2^T\phi(t),\ f_1(t) = F_1^T\phi(t) \text{ and } f_2(t) = F_2^T\phi(t). \tag{3.2.2}$$

Substituting Eq. (3.2.2) in Eq. (3.2.1), we get

$$a_{11}(x)\int_0^x \frac{C_1^T\phi(t)dt}{(x-t)^b} + a_{12}(x)\int_x^1 \frac{C_2^T\phi(t)dt}{(t-x)^b} = F_1^T\phi(x), \tag{3.2.3}$$

$$a_{21}(x)\int_0^x \frac{C_1^T\phi(t)dt}{(t-x)^b} + a_{22}(x)\int_x^1 \frac{C_2^T\phi(t)dt}{(x-t)^b} = F_2^T\phi(x). \tag{3.2.4}$$

From the Eqs. (3.2.3) and (3.2.4), we have

$$\left.\begin{aligned} C_1^T &= \left[\frac{a_{22}F_1^TQ^T - a_{12}F_2^TP^T}{a_{11}a_{22}PQ^T - a_{12}a_{21}QP^T}\right] \\ C_2^T &= \left[\frac{a_{21}F_1^TQ^T - a_{11}F_2^TP^T}{a_{12}a_{21}P^TQ - a_{22}a_{11}Q^TP}\right] \end{aligned}\right\}, \tag{3.2.5}$$

where,

$$\int_0^x \frac{\phi(t)}{(x-t)^b}dt = P\phi(x), \int_x^1 \frac{\phi(t)}{(t-x)^b}dt = P^T\phi(x), \int_0^x \frac{\phi(t)}{(t-x)^b}dt = Q\phi(x), \int_x^1 \frac{\phi(t)}{(x-t)^b}dt = Q^T\phi(x).$$

Hence, the approximate solutions of $u_1(t)$ and $u_2(t)$ for generalized Abel integral Eq. (3.2.1) are obtained by putting the values of C_1^T and C_2^T from Eq. (3.2.5) in Eq. (3.2.2).

Type 2:

In type second we solve the system of Generalized Abel's integral equation which is given by

$$\left.\begin{aligned} a_{11}(x)\int_0^x \frac{u_1(t)u_2(t)dt}{(x-t)^b} + a_{12}(x)\int_x^1 \frac{(u_2)^\beta(t)dt}{(t-x)^b} = f_1(x) \\ a_{21}(x)\int_0^x \frac{u_1(t)u_2(t)dt}{(t-x)^b} + a_{22}(x)\int_x^1 \frac{(u_2)^\beta(t)dt}{(x-t)^b} = f_2(x) \end{aligned}\right\}, \tag{3.2.6}$$

if $u_1(x)u_2(x), (u_2)^\beta(x), f_1(x)$ and $f_2(x)$ are approximated as

$$u_1(x)u_2(x) = C_1^T\phi(x), (u_2)^\beta(x) = C_2^T\phi(x), f_1(x) = F_1^T\phi(x) \text{ and } f_2(x) = F_2^T\phi(x). \tag{3.2.7}$$

Substituting Eq. (3.2.7) in Eq. (3.2.6), we get

$$a_{11}(x)\int_0^x \frac{C_1^T\phi(t)dt}{(x-t)^b} + a_{12}(x)\int_x^1 \frac{C_2^T\phi(t)dt}{(t-x)^b} = F_1^T\phi(x), \tag{3.2.8}$$

$$a_{21}(x)\int_0^x \frac{C_1^T\phi(t)dt}{(t-x)^b} + a_{22}(x)\int_x^1 \frac{C_2^T\phi(t)dt}{(x-t)^b} = F_2^T\phi(x). \tag{3.2.9}$$

Grouping the Eqs. (3.2.8) and (3.2.9), we have

$$\left.\begin{aligned} C_1^T &= \left[\frac{a_{22}F_1^TQ^T - a_{12}F_2^TP^T}{a_{11}a_{22}PQ^T - a_{12}a_{21}QP^T}\right] \\ C_2^T &= \left[\frac{a_{21}F_1^TQ^T - a_{11}F_2^TP^T}{a_{12}a_{21}P^TQ - a_{22}a_{11}Q^TP}\right] \end{aligned}\right\}, \tag{3.2.10}$$

where,

$$\int_0^x \frac{\phi(t)}{(x-t)^b}dt = P\phi(x), \int_x^1 \frac{\phi(t)}{(t-x)^b}dt = P^T\phi(x), \int_0^x \frac{\phi(t)}{(t-x)^b}dt = Q\phi(x), \int_x^1 \frac{\phi(t)}{(x-t)^b}dt = Q^T\phi(x) \ .$$

Hence, the approximate solutions of $u_1(t)$ and $u_2(t)$ for generalized Abel integral Eq. (3.2.6) are obtained by putting the values of C_1^T and C_2^T from Eq. (3.2.10) in Eq. (3.2.7).

3.3 Convergence Analysis

Theorem 3.3.1. If the approximated sequences $u_1^n(t)$ and $u_2^n(t)$ converge uniformly into $u_1(t)$ and $u_2(t)$ respectively on [0, 1]. Then $u_1(t)$ and $u_2(t)$ forms a solutions for the system of equations

$$\left.\begin{array}{l} a_{11}\int_0^x \frac{F_1(u_1(t),u_2(t))dt}{(x-t)^b} + a_{12}\int_x^1 \frac{F_2(u_1(t),u_2(t))dt}{(t-x)^b} = f_1(x) \\ a_{21}\int_0^x \frac{F_3(u_1(t),u_2(t))dt}{(t-x)^b} + a_{22}\int_x^1 \frac{F_3(u_1(t),u_2(t))dt}{(x-t)^b} = f_2(x) \end{array}\right\}. \tag{3.3.1}$$

Proof.

From the construction of approximate sequences, it is evident that

$$\left(\int_0^1 \left\{ a_{11}\int_0^x \frac{F_1(u_1^n(t),u_2^n(t))dt}{(x-t)^b} + a_{12}\int_x^1 \frac{F_2(u_1^n(t),u_2^n(t))dt}{(t-x)^b} - f_1(x) \right\} dx \right)\phi_k(x) = 0, \quad \forall, k = 1,2,\dots,n. \tag{3.3.2}$$

Let $V_N = span(\phi_1, \phi_2, \dots, \phi_n)$.

Then,

$$\left(\int_0^1\left\{a_{11}\int_0^x\frac{F_1(u_1^n(t),u_2^n(t))dt}{(x-t)^b}+a_{12}\int_x^1\frac{F_2(u_1^n(t),u_2^n(t))dt}{(t-x)^b}-f_1(x)\right\}dx\right)\phi(x)=0. \qquad (3.3.3)$$

For $\phi\in V_N$. (By linearity property)

Let's assume,

$$L_1(u_1^n,u_2^n)=a_{11}\int_0^x\frac{F_1(u_1^n(t),u_2^n(t))dt}{(x-t)^b}+a_{12}\int_x^1\frac{F_2(u_1^n(t),u_2^n(t))dt}{(t-x)^b}-f_1(x).$$

Then clearly $L_1(u_1^n,u_2^n)$ converges to $L_1(u_1,u_2)$ in L^2-norm.

So, $\| L_1(u_1^n,u_2^n)\|_{L^2}\le K$.

Let ϕ be a L^2 -function. Then for each given $\varepsilon>0$, there exists $\phi_N\in V_N$ such that

$$\|\phi-\phi_N\|<\varepsilon .$$

So,

$$\begin{aligned}\int_0^1 L_1(u_1^n(x),u_2^n(x))\phi(x)dx &= \int_0^1 L_1(u_1^n(x),u_2^n(x))(\phi(x)-\phi_N(x))dx\\ &\quad+\int_0^1 L_1(u_1^n(x),u_2^n(x))(\phi_N(x))dx\\ &=\int_0^1 L_1(u_1^n(x),u_2^n(x))(\phi(x)-\phi_N(x))dx+0\\ &=\int_0^1 L_1(u_1^n(x),u_2^n(x))(\phi(x)-\phi_N(x))dx.\end{aligned} \qquad (3.3.4)$$

Now,

$$\begin{aligned}\left|\int_0^1 L_1(u_1^n(x),u_2^n(x))\phi(x)dx\right| &\le\| L_1(u_1^n(x),u_2^n(x))\|_{L^2}\ \phi(x)-\phi_N(x))\|_{L^2}\\ &=K\varepsilon\end{aligned}$$

$$\lim_{N\to\infty}\int_0^1 L_1(u_1^n(x),u_2^n(x))\phi(x)dx\le K\varepsilon \qquad \text{, for arbitrary } \varepsilon>0 .$$

So, $\lim_{N\to\infty}\int_0^1 L_1(u_1^n(x),u_2^n(x))\phi(x)dx = 0$.

Now, $L_1(u_1^n,u_2^n)$ uniformly converges to $L_1(u_1,u_2)$ in L^2-norm.

So,

$$\int_0^1 L_1(u_1,u_2)\ \phi\ dx = 0\,,\ \forall\ \phi\in L^2\,. \tag{3.3.5}$$

Then we take $\phi = L_1(u_1,u_2)$.

so,

$$\begin{aligned}&\int_0^1 L_1(u_1,u_2)L_1(u_1,u_2)dx = 0\\ &\Rightarrow L_1(u_1,u_2) = 0, \quad a.e.\end{aligned} \tag{3.3.6}$$

Similarly we can show that

$L_2(u_1^n,u_2^n) = a_{21}\int_0^x \frac{u_1^n(t)dt}{(t-x)^b} + a_{22}\int_x^1 \frac{u_2^n(t)dt}{(x-t)^b} - f_2(x)$ uniformly converges to $L_2(u_1,u_2)$

and $L_2(u_1,u_2) = 0$, $a.e.$

3.4 Numerical Results

In this section, we discuss the implementation of our proposed algorithm and investigate its accuracy and stability by applying it on test functions with known analytical Abel inverse. For, it is always desirable to test the behaviour of a numerical inversion method using simulated data, for which the exact results are known and thus making the comparison possible, between the results obtained through numerical inversion and the theoretical data . We have chosen three profiles having various shapes for this purpose.

The following examples are solved with and without noise terms to illustrate the efficiency and stability of our method. Note that in all examples to follow, the series given by Eq. (2.2.2, **Chapter-2**) is truncated at level $m = 6$, and hence the operational matrices in Eq. (2.3.1, **Chapter-2**) are of order 7×7. The accuracy of proposed algorithm is demonstrated by calculating the parameters of absolute error $\Delta u_j(t_i)$ and average deviation σ^j also known as root mean square error (RMS). They are calculated using the following equations:

$$E_j(t_i) = \left|u_j(t_i) - \tilde{u}_j(t_i)\right|, \qquad \sigma^j{}_N = \{\frac{1}{N}\sum_{i=1}^{N}[u_j(t_i) - \tilde{u}_j(t_i)]^2\}^{1/2} = \{\frac{1}{N}\sum_{i=1}^{N}\Delta u_j{}^2(t_i)\}^{1/2} = \left\|u_j\right\|_2, \qquad (3.4.1)$$

where $\tilde{u}_j(t_i)$ is the approximate solution calculated at point t_i and $u_j(t_i)$ is the exact solution at the corresponding point, $j = 1, 2$. Note that σ^j, henceforth, denoted by $\sigma^j{}_N$ (for computational convenience) is the discrete l^2-norm of the absolute error Δu_j denoted by $\left\|\Delta u_j\right\|_2$.

Note that calculation of $\sigma^j{}_N$ is performed by taking $N = 1000$ in Eq. (3.4.1). In all the examples, the exact and noisy profiles are denoted by $f_j(x)$ and $f_j^{\delta}(x)$, respectively, where $f_j^{\delta}(x)$, is obtained by adding a noise δ *to* $f_j(x)$, such that $f_j^{\delta}(x_i) = f_j(x_i) + \delta\theta_i$,

where $x_i = ih,\ i = 1,\ldots,N,\ Nh = 1$ and θ_i is uniform random variable with values in [-1, 1] such that $\underset{1\le i\le N}{Max}\left|f_{ji}^{\delta} - f_{ij}\right| \le \delta$.

The following examples are solved with and without noise to illustrate the efficiency and stability of our method by choosing two different values of the noises δ_k *as* $\delta_0 = 0,\ \delta_1 = \sigma_N^j$.

The absolute errors E_j^{**}, E_j^{*} *and* E_j, $j = 1, 2, 3$ are shown in all the three examples for $n = 4, 5$ and 6 respectively. Similarly in tables 4.1., 4.2. and 4.3., we show the maximum absolute errors without noise and root mean square errors with noise σ_j^1 , $j = 1, 2$. The accuracy and stability of the propsed method is establised by figures and tables.

Example 3.4.1. Consider the System of Generalized Abel's integral Eq. (3.2.1) take

$$a_{11}(x) = 1, a_{12}(x) = 3, a_{21}(x) = 2, a_{22}(x) = 1, b = \frac{1}{3}, a = 2, c = 3,$$

$$f_1(x) = \frac{729x^{\frac{14}{3}}}{1540} + \frac{9}{440}(1-x)^{\frac{2}{3}}(40 + 9x(5 + 6x + 9x^2))$$

$$and \quad f_2(x) = \frac{-729(-x)^{\frac{14}{3}}}{770} - \frac{3}{440}(-1+x)^{\frac{2}{3}}(40 + 9x(5 + 6x + 9x^2)).$$

This has the exact solution $u_1(t) = t^2$, $u_2(t) = t$.

Table 3.1.

(Maximum absolute errors (E1 and E2) and Root mean square errors $(\sigma_{1000}^1 \ and \ \sigma_{1000}^2)$ for example-3.4.1)

N	Maximum absolute error (E1) For $u_1(t) = t^2$	Maximum absolute error (E2) For $u_2(t) = t$	Root mean square error (σ_{1000}^1) For $u_1(t) = t^2$	Root mean square error (σ_{1000}^2) For $u_2(t) = t$
4	2.4379×10^{-3}	1.6525×10^{-2}	2.9640×10^{-4}	1.7141×10^{-3}
5	1.5531×10^{-3}	1.6304×10^{-2}	2.1723×10^{-4}	1.2611×10^{-3}
6	1.2575×10^{-3}	7.1092×10^{-3}	1.9642×10^{-4}	5.4436×10^{-4}

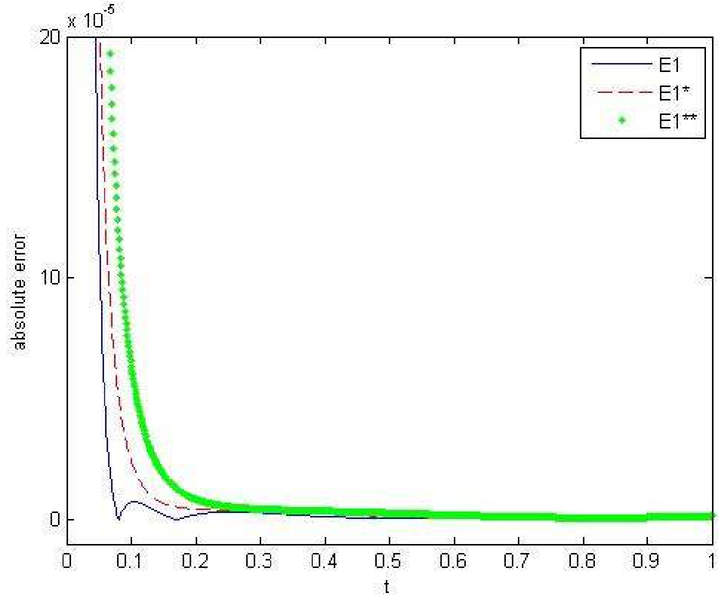

Figure 3.1.1 Comparison of absolute errors at n=4, 5 and 6 for the solution $u_1(t)$.

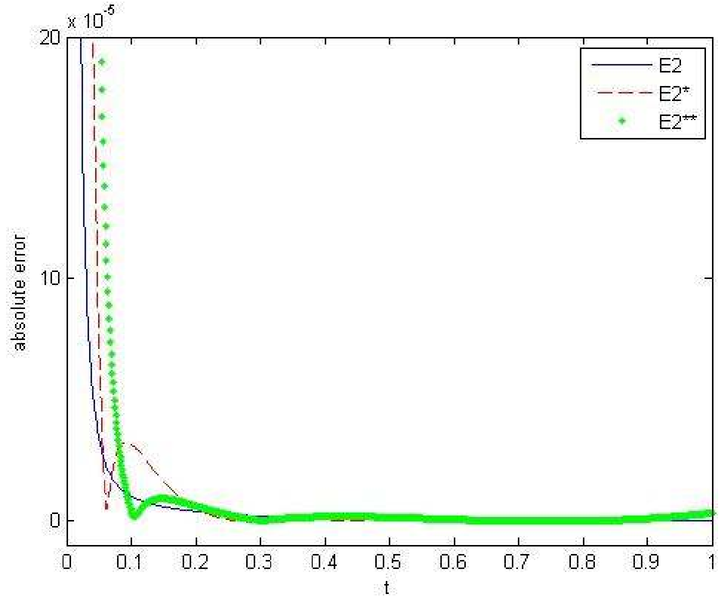

Figure 3.1.2 Comparison of absolute errors at n=4, 5 and 6 for the solution $u_2(t)$.

Example 3.4.2. Consider the System of Generalized Abel's integral Eq. (3.2.6) take $a_{11}(x)=\frac{1}{4}, a_{12}(x)=\frac{1}{3}, a_{21}(x)=\frac{1}{7}, a_{22}(x)=\frac{1}{2}, b=\frac{1}{7}, \beta=\frac{3}{2}$

$$f_1(x)=\frac{\Gamma\left[\frac{6}{7}\right]\Gamma\left[\frac{16}{5}\right]}{4\Gamma\left[\frac{142}{35}\right]}x^{\frac{107}{35}}+\frac{1}{3}\left(\frac{70}{81}(1-x)^{\frac{6}{7}}+\frac{\Gamma\left[\frac{-81}{70}\right]\Gamma\left[\frac{6}{7}\right]}{\Gamma\left[\frac{-3}{10}\right]}x^{\frac{81}{70}}\right.$$
$$\left.+\frac{490}{297}.x.Hypergeometric\,{}_2F_1\left[\frac{-11}{70},\frac{1}{7},\frac{59}{70},x\right]\right),$$

and
$$f_2(x)=\frac{1}{7}\left(\frac{\Gamma\left[\frac{6}{7}\right]\Gamma\left[\frac{16}{5}\right]}{4\Gamma\left[\frac{142}{35}\right]}x^{\frac{16}{5}}(-x)^{\frac{1}{7}}\right)+\frac{1}{2}\left(\frac{(-1)^{\frac{6}{7}}}{891}\left(-770(1-x)^{\frac{6}{7}}-891x^{\frac{81}{70}}\frac{\Gamma\left[\frac{-81}{70}\right]\Gamma\left[\frac{6}{7}\right]}{\Gamma\left[\frac{-3}{10}\right]}\right.\right.$$
$$\left.\left.-1470\bullet x\bullet Hypergeometric\,{}_2F_1\left[\frac{-11}{70},\frac{1}{7},\frac{59}{70},x\right]\right)\right).$$

This has the exact solution $u_1(t)=t^2$, $u_2(t)=t^{\frac{1}{7}}$.

Table 3.2.

(Maximum absolute errors (E1 and E2) and Root mean square errors $(\sigma^1_{1000}\ and\ \sigma^2_{1000})$ for example-3.4.2)

N	Maximum absolute error E1 for $u_1(t)=t^2$	Maximum absolute error E2 For $u_2(t)=t^{\frac{1}{7}}$	Root mean square error (σ^1_{1000}) for $u_1(t)=t^2$	Root mean square error (σ^2_{1000}) for For $u_2(t)=t^{\frac{1}{7}}$
4	1.7635×10^{-3}	4.6865×10^{-1}	4.3546×10^{-4}	2.2101×10^{-2}
5	1.3822×10^{-3}	4.3701×10^{-1}	2.7916×10^{-4}	1.8663×10^{-2}
6	1.0356×10^{-3}	4.1196×10^{-1}	1.8883×10^{-4}	1.6386×10^{-2}

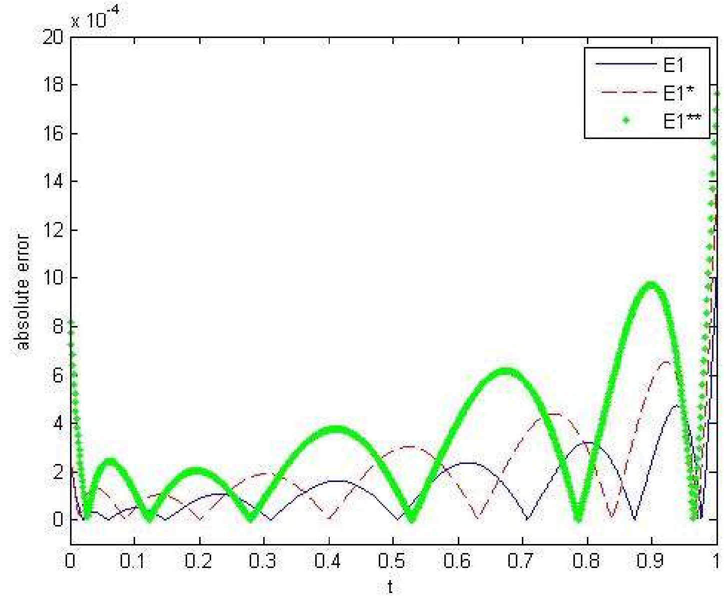

Figure 3.2.1 Comparison of absolute errors at n= 4, 5 and 6 for the solution $u_1(t)$.

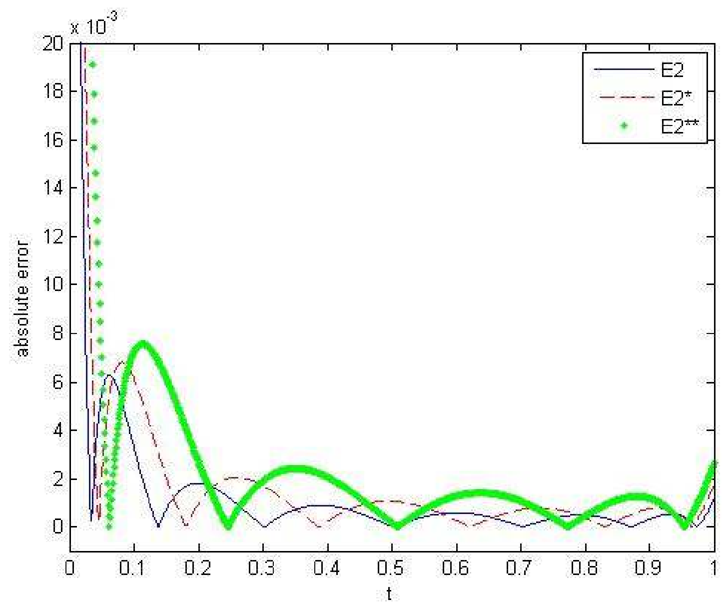

Figure 3.2.2 Comparison absolute errors at $n = 4, 5$ and 6 for the solution $u_2(t)$.

3.5 Conclusions

We have constructed the fractional order operational matrices of integration and differentiation for Legendre scaling functions and used them to propose algorithm for numerical solution of system of generalized Abel integral equations. The stability with respect to the data is stored and good accuracy is obtained, even for small intervals and high noise levels in the data. The choice of only seven orthonormal polynomial of degree 6 makes the method very simple and easy to use.