King Fahd University of Petroleum & Minerals

DEPARTMENT OF MATHEMATICAL SCIENCES

Technical Report Series

TR 182

August 1995

Approximate Solution of Integral Equations and Convolution Integrals Using Legendre Polynomials

Bokhari, Chaudhry An., A. Qadir

DHAHRAN 31261 ● SAUDI ARABIA ● www.kfupm.edu.sa/math/ ● E-mail: mathdept@kfupm.edu.sa

APPROXIMATE SOLUTION OF INTEGRAL EQUATIONS AND CONVOLUTION INTEGRALS USING LEGENDRE POLYNOMIALS

M.A. BOKHARI, M. ANWAR CHAUDHRY* and ASGHAR QADIR +

Department of Mathematical Sciences
King Fahd University of Petroleum and Minerals
Dhahran, Saudi Arabia

Abstract

It has been argued that Chebyshev polynomials are ideal to use as approximating functions to obtain solutions of integral equatioins and convolution integrals on account of their fast convergence. Using the standard deviation as a measure of the accuracy of the approximation and the CPU time as a measure of the speed, we find that for reasonable accuracy Legendre polynomials are more efficient.

*On leave from the Centre for Advanced Studies in Pure and Applied Mathematics, Bahauddin Zakariya University, Multan, Pakistan.

+On leave from the Department of Mathematics, Quaid-i-Azam University, Islamabad, Pakistan.

1 Introduction

With readier access to fast electronic computers many problems previously regarded as intractible can now be solved. In particular, integral equations with fairly complicated kernels can now be solved numerically using approximation methods. To increase the number of problems that can be solved there is a perpetual need for more efficient schemes to solve such problems.

There has been much work done on the numerical solution of integral equations and convolution integrals [1-7]. Of particular interest is the use of orthogonal functions and specially polynomials [8-13]. Due to the product formula for Chebyshev polynomials

$$2T_m(x)T_n(x) = T_{|m-n|}(x) + T_{m+n}(x), \tag{1}$$

there is economisation of coefficients as the higher order polynomial can be neglected without introducing very large truncation errors [14-18].

Though Chebyshev polynomials attain rapid convergence due to the aforementioned economisation, they do not have convenient orthogonality properties, due to the presence of a wieghting factor of $(1 - x^2)^{-1/2}$. On the other hand Legendre polynomials do have convenient orthogonality properties. The schemes developed by Elliott [14] and Chou and Horng [18] are built around Eq.(1). We modify their scheme so that any orthogonal polynomial set can be used for the approximation procedure and use it for Legendre as well as Chebyshev polynomials. Using the standard deviation as a measure of the error of approximation and the CPU time as a measure of the speed, we find that for reasonable accuracy the Legendre polynomials are more efficient than Chebyshev polynomials and our modified scheme is preferable for some problems.

2 Preliminaries

We will be using shifted Chebyshev and Legendre polynomials which we will denote by $T_n(x)$ and $P_n(x)$ respectively, so that $0 \leq x \leq 1$. we shall use the **Einstein summation convention** whereby repeated indices and suffices will be implicitly summed over. The vector with an index will represent a column vector and with a suffix a row vector. Thus any separable kernel $h(x, y)$ can be written as

$$h(x,y) = \sum_{\alpha=1}^{p} X_\alpha(x) Y^\alpha(y) \doteqdot X_\alpha(x) Y^\alpha(y) \equiv Y^\alpha(y) X_\alpha(x). \qquad (2)$$

This notation is convenient as we do not need to use the position of a vector to specify the order of multiplication while at the same time obviating the need to insert the summation sign.

The approximation of some function, $f(x)$, in terms of Chebyshev polynomials, for example, will be written as

$$f(x) \approx f_i T^i(x) \qquad (i = 0, 1, \ldots, m) \qquad (3)$$

A non-separable kernel can be approximated by a double Chebyshev series

$$h(x,y) \approx h_{ia} T^i(x) T^a(y) \qquad (a = 0, 1, \ldots, n), \qquad (4)$$

where the Chebyshev coefficients are given by

$$h_{ia} = \frac{c}{\pi^2} \int_0^1 \int_0^1 \frac{h(x,y)T_i(x)T_a(y)}{\sqrt{x-x^2}\sqrt{y-y^2}}\, dx\, dy, \tag{5}$$

where $c = 1$ when $i = a = 0$, $c = 2$ when either i or $a = 0$ (but not both) and $c = 4$ when $i \neq 0 \neq a$. Transposes of vector are denoted by lowering or raising the index. This may be obtained formally by, for example,

$$T_i(x) = \delta_{ij}T^j(x)\, ,\, T_a(y) = \delta_{ab}T^b(y), \tag{6}$$

where δ_{ij}, δ_{ab} are the Kronecker deltas, being the $(m+1)\times(m+1)$ and $(n+1)\times(n+1)$ identity matrices. Thus they are +1 if $i = j$ (or $a = b$) and 0 if $i \neq j$ (or $a \neq b$). Their inverses are denoted by δ^{ij} , δ^{ab} respectively.

3 Solution of the Fredholm Equation

The analytic solution of the Fredholm equation (e.g. [19])

$$f(x) = g(x) + \lambda \int_0^1 h(x,y)f(y)\, dy, \tag{7}$$

where f is the unknown function, g a given function, h a separable kernel expressible by Eq.(2) and λ an eigenvalue, is obtained by noting that f must be of the form

$$f(x) = g(x) + \lambda F^{\alpha} X_{\alpha}(x) = g(x) + \lambda F_{\alpha} X^{\alpha}(x), \tag{8}$$

where the vector $\mathbf{F}$ is given by

$$F_{\alpha} = \int_0^1 Y_{\alpha}(y)f(y)\, dy. \tag{9}$$

Inserting Eq.(9) back into Eq.(8) and solving gives

$$F_\beta = B^\alpha_\beta G_\alpha\,, \qquad B^\alpha_\beta = (\delta^\beta_\alpha - \lambda A^\beta_\alpha)^{-1}, \tag{10}$$

where

$$G_\alpha = \int_0^1 Y_\alpha(y) g(y) dy, \; A^\beta_\alpha = \int_0^1 X_\alpha(y) Y^\beta(y) dy, \tag{11}$$

provided λ^{-1} is not an eigenvalue of the matrix **A**. Further, even if λ^{-1} is not an eigenvalue of **A** but is numerically close to it, the numerical solution becomes unstable.

In the presence of the analytic solution there seems to be little justification for obtaining numerical solutions. However, for a non- separable kernel the numerical scheme may become necessary. While Elliott [14] dealt mainly with numerical solutions of equations with separable kernels, he pointed out that he could obtain them for non-separable kernels point by point at specific values of x. This procedure is far from satisfactory as it is very tedious and does not provide the solution, even approximately, in the form of a function. Chou and Horng [18] considered the approximation of the kernel given by Eq.(4). Though they only considered examples with separable kernels, their procedure obviously applies to non-separable kernels as well, providing an approximate solution in terms of Chebyshev polynomials.

The procedure described by Chou and Horng is somewhat convoluted and involved. We have straightened their procedure out and avoided using Eq.(1). We will not reproduce their procedure here but will only give our modification of it. Using the approximations given by Eqs. (3) and (4) along with Eq.(2) in Eq.(7) we get

$$f_i T^i(x) = g_i T^i(x) + \lambda f_j C^{jk} X_{\alpha k} Y_i^{\alpha} T^i(x), \tag{12}$$

where

$$C^{jk} = \int_0^1 T^j(y) T^k(y)\, dy. \tag{13}$$

Comparing the coefficients of the Chebyshev polynomials and simplifying, we obtain

$$f_j = E_j^i g_i \ , \ E_j^i = \left(\delta_j^i - C^{ik} X_{\alpha k} Y_j^{\alpha}\right)^{-1}. \tag{14}$$

Clearly the above procedure could be used with Legendre polynomials, $P_n(x)$, instead of $T_n(x)$. Instead of the reduction of Eq.(13) by the product formula we obtain a simple diagonal matrix on account of the orthogonality of Legendre polynomials. The benefit here is greater than the loss, so Legendre polynomials give significantly better efficiency than the Chebyshev polynomials.

Chou and Horng [18] found an exact fit of their approximation scheme solution in the example they chose, with the kernel $h(x, y) = x + y$. This exact fit arose because $X_\alpha(x)$ and $Y^\alpha(y)$ are exactly given by first order Chebyshev polynomials and their product by second order polynomials gives $\mathbf{C}$ exactly. Since they used fourth order polynomials, there was no approximation involved. (In our procedure second order polynomials would have sufficed but due to their more involved procedure they actually needed fourth order polynomials for the exact fit.)

To compare efficiency of different schemes we need non-trivial examples. Even with such examples the schemes are not really distinguishable when more polynomials are used unless more components are taken for the separable kernel (i.e. p is increased). Thus we have limited the analysis to polynomials of order 2, 3 and 4 (and in some cases 5) so that the errors involved should not become totally negligible. We used the analytic solution to obtain the "correct" result, with Simpson's rule integration over 200 points providing errors much less than 10^{-8}, well beyond the errors we are comparing with. We use a sample of 200 points to work out the standard deviation as a measure of the error of the numerical scheme being investigated. The speed was measured by the CPU time. We considered the integral equation with $g(x) = \lambda = 1$ and the kernel

$$h(x,y) = \frac{1}{\sqrt{1+4\cos^2 x}} + \frac{1}{\sqrt{1+4\sin^2 y}}. \tag{15}$$

The comparison of the scheme of Chou and Horng [18], our modified version with Chebyshev polynomials and with Legendre polynomials is given in Table 1. Clearly, the Legendre polynomials are most efficient, as expected.

4 Solution of the Volterra Equation

For the Volterra equation

$$f(x) = g(x) + \lambda \int_0^x h(x,y) f(y)\, dy, \tag{16}$$

there is no general analytic solution and so numerical methods become essential. The approximation procedure is unchanged except that the range of integration to obtain **C** in Eq.(13) now becomes 0 to x instead of 0 to 1. Thus **C** becomes a function of x. (Notice that here we must again use an orthogonal polynomial approximation). The speed of our procedure reduces on account of this fact, without the corresponding benefit of a diagonal C for the Legendre polynomials. However, the procedure of Chou and Horng [18] introduces greater errors as it must repeatedly use the economisation of the Chebyshev polynomials, allowing errors to build up more.

For the Volterra equation even the simple example considered by Chou and Horng can introduce errors as repeated products of the linear functions in the kernel do not continue to be linear. As such we have studied their example and one more. Their example takes

$$g(x) = x - 2\cos x + 2,\ \lambda = -1,\ h(x,y) = x - y \tag{17}$$

and has the solution

$$f(x) = (1 + x)\sin x. \tag{18}$$

The comparison is given in Table 2a. The other example has

$$g(x) = e^x,\ \lambda = 2,\ h(x,y) = \cos(x - y), \tag{19}$$

whose solution is

$$f(x) = (1 + x)^2 e^x. \tag{20}$$

The comparison is given in Table 2b. We have chosen examples where the analytic solution is known so that the comparison can be made easily. It is apparent that the modified method with Chebyshev polynomials is slower but more accurate. The use of Legendre polynomials is clearly more efficient for reasonably accurate solutions but for very precise results it is not clear that it will remain so.

5 Convolution Integrals

To obtain the convolution integral

$$f(x) = \int_0^x h(x,y)g(y)\,dy, \tag{21}$$

we use the approximations used before to obtain

$$f(x) = g_i Y_j^\alpha C^{ij}(x) X_\alpha(x), \tag{22}$$

where $\mathbf{C}(x)$ is defined here as for the Volterra equation.

Since this procedure, unlike the solution of an integral equation, involves a simple integration, differences between the various schemes do not show up as easily. Further, there is no matrix inversion involved. The only difference between our procedure and that of Chou and Horng is that they use a double Chebyshev series while we use a single series. Consequently the example used to study the behaviour of errors needs to be more complicated. We take

$$\left.\begin{aligned} h(x,y) &= e^{3(y-x)} + \cos\pi(x-y), \\ g(y) &= e^{-2y}, \end{aligned}\right\} \tag{23}$$

which has the solution

$$f(x) = \frac{1}{4+\pi^2}\left(2\cos\pi x + \pi\sin\pi x - 2e^{-2x}\right). \tag{24}$$

The comparison is given in Table 3. Here our procedure with Chebyshev polynomials comes out to be the most efficient.

6 Discussion and Conclusion

We have shown that, depending on the nature of the problem being discussed and the accuracy required, the Legendre polynomials may be preferable to use than the Chebyshev polynomials. This is most clearly so when we are dealing with problems with fixed limits of integration. The orthogonality property outweighs the benefit of economisation on account of the fact that we get a diagonal matrix to invert in the former case and an arbitrary one in the latter case. In the case of the Volterra equation, for lower accuracy requirements the Legendre polynomials are definitely more efficient. However, for more accurate results the time expenditure may start becoming greater than the enhancement of accuracy is worth. For the convolution integral the Legendre polynomials start taking much more time than our scheme with Chebyshev polynomials. However, our schemes are more efficient than that of Chou and Horng in any case for higher m. This advantage would be lost for non-separable kernels. It is worth noting that Legendre polynomials continue to give more accurate results for each order of the polynomial used. (Incidentally, there is an accidentally accurate result of the Chou and Horng scheme for third order polynomials and a corresponding worsening of our scheme result with Chebyshev polynomials which causes a reversal between their accuracy).

It is also necessary to consider propagation of errors in the scheme of Chou and Horng, as compared with ours, when there are more components in the kernel. Denoting the number by p, the matrix calculations would cause the truncation error to be multiplied by p^4 in our scheme and $4p^4$ in their scheme (on account of repeated numerical matrix calculations by them). Since they also have a higher truncation error, their error build-up is likely to be greater.

To conclude, we would like to stress the importance of using criteria like those used in this paper. With easy access to fast electronic computing one can easily experiment with different schemes to determine their efficiency as measured by such parameters.

Acknowledgements

The authors would like to thank the King Fahd University of Petroleum and Minerals for providing the facilities for this research, and Dr. M. Iqbal for helpful suggestions.

Table Captions and Tables

Table 1

Errors and CPU time for numerical solutions of Fredholm equations. The Chou and Horng scheme is labelled CHC, ours with Chebyshev polynomials BCQC and with Legendre polynomials BCQL. The order of the polynomials denoted by m, the standard deviation estimated over 200 points by σ and CPU time by t.

$$h(x,y) = (1 + 4\cos^2 x)^{-1/2} + (1 + 4\sin^2 y)^{-1/2}$$

	CHC		BCQC		BCQL	
m	σ	t	σ	t	σ	t
2	1.15×10^{-2}	0.118	1.16×10^{-2}	0.115	0.86×10^{-2}	0.104
3	8.79×10^{-3}	0.137	9.82×10^{-3}	0.134	1.12×10^{-3}	0.121
4	2.45×10^{-4}	0.160	2.93×10^{-4}	0.156	0.98×10^{-4}	0.138

Table 2

Errors and CPU time for numerical solutions of Volterra equations. Notation used as before, in Table 1.

(a) $h(x,y) = x - y$

	CHC		BCQC		BCQL	
m	σ	t	σ	t	σ	t
2	1.04×10^{-2}	0.097	1.01×10^{-2}	0.102	0.87×10^{-2}	0.075
3	7.45×10^{-4}	0.112	6.30×10^{-4}	0.131	5.56×10^{-4}	0.093
4	4.61×10^{-5}	0.130	4.23×10^{-5}	0.183	3.76×10^{-5}	0.139

(b) $h(x,y) = \cos(x - y)$

	CHC		BCQC		BCQL	
m	σ	t	σ	t	σ	t
2	1.25×10^{-1}	0.123	1.28×10^{-1}	0.130	0.96×10^{-1}	0.072
3	1.62×10^{-2}	0.153	1.12×10^{-2}	0.169	0.90×10^{-2}	0.092
4	1.22×10^{-3}	0.182	0.73×10^{-3}	0.231	0.64×10^{-3}	0.139
5	4.91×10^{-5}	0.219	4.11×10^{-5}	0.330	3.65×10^{-5}	0.242

Table 3

Errors and CPU time for calculation of convolution integral.Notation used as before, in Tables 1 and 2.

	CHC		BCQC		BCQL	
m	σ	t	σ	t	σ	t
2	2.37×10^{-2}	0.227	1.21×10^{-2}	0.254	1.60×10^{-2}	0.231
3	2.51×10^{-3}	0.390	4.21×10^{-3}	0.392	2.18×10^{-3}	0.402
4	6.08×10^{-4}	0.696	2.26×10^{-4}	0.602	1.30×10^{-4}	0.713
5	9.03×10^{-5}	1.242	2.55×10^{-5}	0.900	1.67×10^{-5}	1.250

References

[1] Crout, P.D., Jour. Math. and Phys 29 (1940) 34.

[2] Fox, L. and Goodwin, E.T., Phil. Roy. Soc. A 245 (1953) 50.

[3] Clenshaw C.W., Proc. Camb. Phil. Soc. 53 (1957) 134.

[4] Wallman, H., J. Franklin Inst. 250 (1960) 45.

[5] Tomovic, R. and Parezanovic, N., IRE Trans. Electron. Comput. 9 (1960) 503.

[6] Stroud, A.H. and Secrest, D., Gaussian Quadrature Formulas (Prentice Hall 1966).

[7] Froberg, C.E., Introduction to Numerical Analysis (Svenkha Bokforlagar 1969).

[8] Chen, C.F. and Hsiao, C.H., IEEE Trans. Autom. Control 20 (1975) 596.

[9] Corrington, M.S., IEEE Trans. Circuit Theory 20 (1973) 470.

[10] Shih, Y.P. and Chia, W.K., J. Chinese Inst. Engr. 1 (1978) 135.

[11] Kung, F.C. and Chen, S.Y., J. Franklin Inst. 306 (1978) 283.

[12] Hwang, C. and Shih, Y.P., Comput. Elect. Engng. 9 (1982) 1223; Chem. Engng. J. 27 (1983) 81.

[13] Wang, M.L. and Chang., R.Y., ASME Trans. Dynam. Syst. Measurement Control, 105 (1983) 222.

[14] Elliott, D., J. Austr. Math. Soc. 1 (1960) 344.

[15] Paraskevopoulos, P.N., J. Franklin Inst. 316 (1983) 135.

[16] Chou, J.H. and Horng, I.R., Int. J. Control 41 (1985) 135.

[17] Horng, I.R. and Chou, J.H., Int. J. Control 41 (1985) 549.

[18] Chou, J.H. and Horng, I.R., Int. J. Control 42 (1985) 225.

[19] Moisewitch, B.L., Integral Equations (Longman Mathematical Texts 1977).