A Laplace Transform Approach for Pricing European Options

Edgard Ngounda and Kailash C. Patidar

Abstract In this paper we investigate two efficient numerical methods for solving the Black–Scholes equation for pricing European options. We use spectral methods to discretize the associated partial differential equation with respect to space (asset direction) and generate a system of ordinary differential equations in time. This system is then solved by applying the numerical inversion of the Laplace transform which is based on the Talbot's method [A. Talbot, The accurate numerical inversion of Laplace transforms, IMA J. Appl. Math. **23(1)**, 97–120 (1979)]. This involves an application of trapezoidal rule to approximate a Bromwich integral. Using Cauchy's integral theorem, we deform the Bromwich line into a contour which starts and ends in the left half plane. Comparative numerical results obtained by this and other three methods (Exponential Time Differencing Runge–Kutta Methods of order 4, MATLAB solver ode15s and Crank-Nicholson's method) are presented.

Keywords Option Pricing · Contour Integrals · Spectral Methods · Exponential Time Differencing Runge-Kutta Methods

1 Introduction

Since its development in the 1970s by F. Black and M. Scholes, the Black–Scholes equation has become a fundamental model for pricing financial derivatives [1]. A derivative security is a financial instrument whose value depends on the values of some other underlying variables, e.g. stocks, foreign currency. Among the most popular derivatives, options are actively traded on different financial markets over the world. An option gives its holder the right without any obligation to buy (call

E. Ngounda · K.C. Patidar (✉)
Department of Mathematics and Applied Mathematics, University of the Western Cape, Bellville 7535, South Africa
e-mail: kpatidar@uwc.ac.za

E. Ngounda
e-mail: edgardngounda@gmail.com

P.N. Agrawal et al. (eds.), *Mathematical Analysis and its Applications*,
Springer Proceedings in Mathematics & Statistics 143,
DOI 10.1007/978-81-322-2485-3_37

option) or to sell (put option) the underlying asset by a certain date (maturity date) for a certain price (strike price). The European options can only be exercised at maturity.

The Black–Scholes partial differential equation can be used to model different types of options. However, a closed form solution cannot always be found and we must therefore resort to numerical methods to solve such a PDE. Some of the most popular methods used in the past to tackle these type of problems are those based on Monte Carlo simulations [2], binomial trees [3] and finite difference methods [7].

Finite difference methods are classical methods for solving PDEs and have been used extensively to price options since the advent of the financial mathematics. The authors in [9] used a grid stretching in combination with backward difference method of fourth order in time to solve the European options. In [11], Tangman et al. used a method based on the grid stretching to generate a high-order compact scheme to improve on the well-known second-order Crank–Nicolson method for solving these problems. In spite of the popularity of these time marching methods, one of their critical drawback is that they usually require as many time steps as spatial meshes to maintain their stability.

In this paper, we consider the application of Laplace transform which has recently been investigated by some researchers and is considered to be a valuable alternative method to finite differences methods for solving parabolic PDEs [4, 10, 15]. This has led to great applications in the financial world.

The rest of the paper is organized as follows. In Sect. 2 we give a full description of the Black–Scholes equation which is used to model the European put and call options. In Sect. 3, we introduce the spectral discretization method. Application of the Laplace integration method is discussed in Sect. 4. Finally, in Sect. 2, we present comparative numerical results.

2 Problem Description

We consider the following Black–Scholes (BS) equation to price European options

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0, \qquad S \in (0, \infty), \quad t \in (0, T). \tag{1}$$

Final and boundary conditions are given by

$$V(S, T) = \begin{cases} \max(S - K, 0) \text{ for call} \\ \\ \max(K - S, 0) \text{ for put} \end{cases} \tag{2}$$

and

$$\left.\begin{aligned} &V(0,t) = 0, \quad V(S,t) \to S - Ke^{-r(T-t)} \quad \text{as} \quad S \to \infty \quad \text{for call,} \\ &V(0,t) = Ke^{-rt}, \quad V(S,t) \to 0 \quad \text{as} \quad S \to \infty \quad \text{for put.} \end{aligned}\right\} \tag{3}$$

In the above, $V(S,t)$ is the price of a call/put option for the underlying asset whose price is S at time t up to the expiry date T, r is the interest rate, σ is the volatility of the underlying asset, and K is the strike price.

We set $\tau = T - t$ to transform the backward formulation (1)–(3) to the following forward equation:

$$\frac{\partial V}{\partial \tau} - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rS\frac{\partial V}{\partial S} + rV = 0, \tag{4}$$

The initial condition is given by the terminal payoff

$$V(S,0) = \begin{cases} \max(S-K,0) \text{ for call} \\ \\ \max(K-S,0) \text{ for put} \end{cases} \tag{5}$$

and the boundary conditions are given by

$$\left.\begin{aligned} &V(0,\tau) = 0, \quad V(S,\tau) \to S - Ke^{-r\tau} \quad \text{as} \quad S \to \infty \quad \text{for call,} \\ &V(0,\tau) = Ke^{-r\tau}, \quad V(S,\tau) \to 0 \quad \text{as} \quad S \to \infty \qquad \text{for put.} \end{aligned}\right\} \tag{6}$$

3 Spectral Discretization

To semi-discretize the PDE (1), we consider a spectral method. The basic idea behind the spectral methods is as follows. For a given set of points, we interpolate the unknown solution and differentiate the interpolating polynomial at these grid points. This discretization process leads to a system of equations which can then be solved using any state-of-the-art solvers.

The discretization using spectral method (in this paper) is based on the Chebyshev polynomial interpolation [13]. Methods such as finite elements or finite differences divide the domain into subdomains and use local polynomials of low degree. By contrast, spectral methods use global representations of high degree over the entire domain.

The implementation of spectral methods can be divided into three categories, namely, the Galerkin, tau, and the collocation (or pseudospectral) methods. The first two of these methods use the expansion coefficients of the global approximation and the latter can be viewed as a method of finding numerical approximations to derivatives at collocation points. In a manner similar to finite difference or finite

element methods, the equation to be solved is satisfied in space at the collocation points. In this paper, we use the third one, i.e., the spectral collocation method.

The spectral process involves seeking the solution to a differential equation by polynomial interpolation. In order to review the concept of polynomial interpolation, we consider interpolating an arbitrary function $f(x)$ at $N+1$ distinct nodes $\{x_k\}_{k=0}^{N}$ in $[-1, 1]$.

Given a set of grid points $\{x_j\}_{j=0}^{N}$, an interpolating approximation to a function $f(x)$ is a polynomial $f_N(x)$ of degree N, determined by the requirement that the interpolant agrees with $f(x)$ at the set of interpolation points $\{x_j\}_{j=0}^{N}$, i.e.,

$$f_N(x_i) = f(x_i), \qquad i = 0, 1, ..., N. \tag{7}$$

We define by $L_k(x)$, the Lagrange polynomial of degree N,

$$L_k(x) = \prod_{\substack{j=0 \\ j\neq k}}^{N} \frac{x - x_j}{x_k - x_j}, \qquad k = 0, 1, ..., N.$$

Note that $L_k(x)$ satisfies $L_j(x_k) = \delta_{jk}$, where δ_{jk} is the Kronecker delta function. The interpolation polynomial $f_N(x)$ is then given by

$$f_N(x) = \sum_{k=0}^{N} f(x_k) L_k(x). \tag{8}$$

In this paper, we use the Chebyshev points as the grid points. These are given by

$$\text{Chebyshev zeros: } x_j = \cos\left(\frac{2j+1}{2(N+1)}\pi\right), \qquad j = 0, ..., N,$$

and

$$\text{Chebyshev extrema: } x_j = \cos\left(\frac{j\pi}{N}\right), \qquad j = 0, ..., N.$$

The Chebyshev points are often defined as the projection onto the interval $[-1, 1]$ of the roots of unity along the unit circle $|z| = 1$ in the complex plane [13]. For European options, since the payoff is nonsmooth, a direct application of the Chebyshev points for discretization leads to low-order approximation. To regain a high-order accuracy an alternative approach was proposed by Tangman [12]. The basic idea is to modify the Chebyshev points as

$$x = [x_k, x_\ell]^T, \tag{9}$$

where

$$x_k = S_{\min} + \left(\frac{K - S_{\min}}{2}\right)\left(1 - \cos\left(\frac{2\pi k}{N}\right)\right), \quad k = 0, 1, ..., \frac{N}{2}, \tag{10}$$

$$x_\ell = K + \left(\frac{S_{\max} - K}{2}\right)\left(1 - \cos\left(\frac{2\pi l}{N}\right)\right), \quad \ell = 1, 2, ..., \frac{N}{2}. \tag{11}$$

for N even. This discretization clusters grid nodes at the boundaries located at $S_{\min}$ and $S_{\max}$ as well as at the strike price K where the discontinuity of the payoff occurs. As we show in Sect. 5, it follows that local grid refinement improve accuracy of the spectral method at the payoff. Another advantage of this strategy is that it applies directly to the Eq. (4) without the need for transforming into the interval $[-1, 1]$.

Differentiation Matrices

The concept of collocation derivatives is associated with the interpolation polynomial $f_N(x)$ as described above. These are the derivatives of $f_N(x)$ at the collocation points $\{x_k\}_{k=0}^N$. Using (8), we can see that the mth order collocation derivative of $f_N(x)$ is given by

$$\frac{d^m f_N(x)}{dx^m} = \sum_{k=0}^{N} f(x_k) \frac{d^m L_k(x)}{dx^m}. \tag{12}$$

Nodal representation yields

$$\frac{d^m f_N(x_j)}{dx^m} = \sum_{k=0}^{N} f(x_k) \frac{d^m L_k(x_j)}{dx^m}, \qquad j = 0, ..., N, \tag{13}$$

which can be expressed by the matrix formula

$$\mathbf{f^{(m)}}_N = D_N^{(m)} \mathbf{f}_N, \tag{14}$$

where

$$\mathbf{f}_N = \begin{bmatrix} f_N(x_0) \\ \vdots \\ f_N(x_N) \end{bmatrix}, \quad \mathbf{f^{(m)}}_N = \begin{bmatrix} f_N^{(m)}(x_0) \\ \vdots \\ f_N^{(m)}(x_N) \end{bmatrix},$$

and $D_N^{(m)}$ is the $(N + 1) \times (N + 1)$ differentiation matrix of order m with entries

$$\left(D_N^{(m)}\right)_{j,k} = L_k^{(m)}(x_j), \qquad j, k = 0, ..., N. \tag{15}$$

The computation of these differentiation matrices for an arbitrary order m has been considered in [6, 13]. Following the approach in [16], Weideman and Reddy [14] developed a MATLAB algorithm (DMSUITE package) that computes the Chebyshev grid points as well as the differentiation matrices of an arbitrary order. The suite contains a function `chebdif` that computes the extreme points of the Chebyshev

polynomial $T_N(x)$ and the differentiation matrix $D_N^{(m)}$. The code takes as input the size of the differentiation matrix N and the highest derivative order m and produces matrices $D_N^{(\ell)}$ of order $\ell = 1, 2, ..., m$.

Formulas for the computation of the entries of $D_N^{(1)}$, $N \geq 1$, let $i, j = 0, 1, ...N$, are (as given in [13]):

$$\left(D_N^{(1)}\right)_{00} = \frac{2N^2+1}{6}, \qquad \left(D_N^{(1)}\right)_{NN} = \frac{2N^2+1}{6}, \tag{16}$$

$$\left(D_N^{(1)}\right)_{jj} = \frac{-x_j}{2(1-x_j^2)}, \qquad j = 1, ..., N-1, \tag{17}$$

$$\left(D_N^{(1)}\right)_{ij} = \frac{c_i}{c_j}\frac{(-1)^{i+j}}{(x_i - x_j)}, \qquad i \neq j, \qquad i, j = 0, ..., N, \tag{18}$$

where

$$c_i = \begin{cases} 2, \; i = 0 \;\text{ or }\; N \\ \\ 1, \quad \text{otherwise.} \end{cases}$$

Higher order derivatives are evaluated by recursions at a cost of $\mathcal{O}(N^2)$ operations [14, 16]. This turns out to be cost-effective as compared to $\mathcal{O}(N^3)$ if higher derivatives are obtained by taking powers of the first derivative [14].

Using the differentiation matrices as described above, we can rewrite (4) in matrix form as

$$\frac{\partial \mathbf{V}}{\partial \tau} - \frac{1}{2}\sigma^2 P D^{(2)}\mathbf{V} - rQD^{(1)}\mathbf{V} + r\mathbf{V} = 0, \tag{19}$$

where P and Q are the diagonal matrices with entries on the main diagonals as $(x_k+1)^2$ and (x_k+1), respectively, for $k = 0, ..., N$.

We will solve Eq. (19) using several time integration methods as indicated in the next two sections.

4 Application of the Laplace Transform to Price the European Call and Put Options

Applying the Laplace transform to Eq. (4), we obtain

$$z\bar{V} - \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 \bar{V}}{\partial S^2} - rS\frac{\partial \bar{V}}{\partial S} + r\bar{V} = V_0. \tag{20}$$

The boundary conditions are given by

$$\left.\begin{array}{l} \bar{V}(0,z)=0, \quad \bar{V}(S,z)=\frac{S_{\max}}{z}-\frac{K}{(z+r)} \quad \text{for call,} \\ \bar{V}(0,z)=\frac{K}{(z+r)}, \quad \bar{V}(S,z)=0 \quad \text{for put.} \end{array}\right\} \tag{21}$$

The Eq. (19), therefore, becomes

$$z\bar{\mathbf{V}}-\frac{1}{2}\sigma^2 P D_N^{(2)}\bar{\mathbf{V}}+rQD_N^{(1)}\bar{\mathbf{V}}-r\bar{\mathbf{V}}=\mathbf{V}_0,$$

$$\left(z_k\mathbf{I}-\frac{1}{2}\sigma^2 P D_N^{(2)}+rQD_N^{(1)}-r\mathbf{I}\right)\bar{\mathbf{V}}_k=\mathbf{V}_0 \qquad k=0,...,N-1. \tag{22}$$

A straight forward application of the Laplace inversion formula [8] yields

$$\mathbf{V}(t)=\frac{h}{2\pi i}\int_{-\infty}^{\infty} e^{z(\ell)t}\bar{\mathbf{V}}z'(\ell)d\ell. \tag{23}$$

Using the symmetry, the trapezoidal approximation yields

$$\mathbf{V}_M(t)=\frac{h}{\pi}\sum_{k=0}^{M-1} e^{z_k t}\bar{\mathbf{V}}_{\mathbf{k}}z'_k, \tag{24}$$

where

$$\bar{\mathbf{V}}_k=(z_k\mathbf{I}-A)^{-1}\mathbf{V}_0, \quad k=0,1,...,N-1, \tag{25}$$

and

$$A=\frac{1}{2}\sigma^2 P D_N^{(2)}-rQD_N^{(1)}+r\mathbf{I}. \tag{26}$$

Now since the differentiation matrices $D_N^{(1)}$ and $D_N^{(2)}$ are not sparse, the Eq. (25) indicates the bulk of the computation in the trapezoidal rules (24). To speed up this computation, an Hessenberg decomposition can be computed once at the beginning as follows:

$$A=MHM^T, \tag{27}$$

where $H=(h_{ij})$ is an upper Hessenberg matrix, i.e., $h_{ij}=0,\ i>j+1$, and M an orthogonal matrix. Then for each z_k, $k=0,1,...,M-1$, the Eq. (25) becomes

$$(z_k I-MHM^T)\mathbf{V}_k=\mathbf{V}_0. \quad k=0,1,...,N-1. \tag{28}$$

From this we have

$$(z_k I-H)\mathbf{U}_k=M^T\mathbf{V}_0 \quad k=0,1,...,N-1, \tag{29}$$

Table 1 Parameters used in the contour over an interval $[t_0, \Lambda t_0]$

Λ	α	$A(\alpha)$	$\widetilde{\mu}\Lambda t_0/M$	$B(\alpha)$
1	1.1721	1.0818	4.4921	2.3157
5	1.0791	2.4578	1.5013	1.2570
10	1.0236	3.3744	0.8871	1.0888
50	0.9381	5.5582	0.3452	0.7152

The right column $B(\alpha)$ shows the convergence rate over the contour for each set parameters

where $\mathbf{U}_k = M^T \mathbf{V}_k$, so that

$$\mathbf{V}_k = M\mathbf{U}_k, \quad k = 0, 1..., N-1. \tag{30}$$

The solution $\mathbf{V}_k$ for each z_k, is obtained by the computation of an almost triangular system (29) and combining the result in (30) at only $O(N^2)$ operations [5]. During this process, the Hessenberg reduction (27) is only computed once beforehand.

For numerical implementation, we considered the following contour parameters defined over an interval $[t_0, \Lambda t_0]$ (as defined in [15])

$$z = \widetilde{\mu}(1 + \sin(iw - \alpha)), \tag{31}$$

where

$$A(\alpha) = \cosh^{-1}\left(\frac{(\pi - 2\alpha)\,\Lambda - \pi + 4\alpha}{(4\alpha - \pi)\sin\alpha}\right)$$

and

$$h = \frac{A(\alpha)}{M}, \ \widetilde{\mu} = \frac{4\alpha\pi - \pi^2}{A(\alpha)}\left(\frac{M}{\Lambda t_0}\right),$$

with $\Lambda \in N$ and $\widetilde{M}$ is the number of points in the trapezoidal rule. The convergence rate of the Laplace method on these contour is given by $\mathscr{O}\left(e^{-B(\alpha)M}\right)$ where

$$B(\alpha) = \frac{\pi^2 - 2\pi\alpha}{\cosh^{-1}\left(\frac{(\pi-2\alpha)\Lambda + 4\alpha - \pi}{(4\alpha-\pi)\sin(\alpha)}\right)}.$$

Values of the above parameters are given in Table 1.

5 Numerical Results and Discussion

We compare the results obtained by using our Laplace transform method with those obtained by simulations that we perform using ETDRK-4 (Exponential Time Differencing Runge–Kutta Method of order 4) as well as the more conventional

Table 2 Comparison of the errors defined by (32), for the Crank–Nicolson's method, ETDRK4 and the Laplace inversion approach applied for a European call option

	ode15s		Crank–Nicolson		ETDRK4		Laplace inversion method	
N	Time (s)	Error	Time (s)	Error	Time (s)	Error	Time (s)	Error
20	12.5E-2	8.2E-2	6.0E-2	7.1E-3	5.2E-2	7.4E-3	1.1E-2	7.4E-3
30	14.8E-2	6.97E-4	4.47E-2	1.3E-3	5.4E-2	1.00E-3	5.1E-3	1.0E-3
40	19.3E-2	6.59E-5	10.9E-2	1.93E-4	7.5E-2	1.22E-4	7.0E-3	1.18E-4
50	22.3E-2	7.86E-6	13.3E-2	9.67E-5	9.5E-2	4.85E-5	9.1E-3	1.07E-5
60	24.2E-2	4.63E-6	1.63E-2	9.73E-5	12.6E-2	4.86E-5	1.2E-4	3.52E-6
80	31.0E-2	1.86E-5	26.1E-2	9.80E-5	21.8E-2	4.89E-5	2.0E-4	5.80E-7

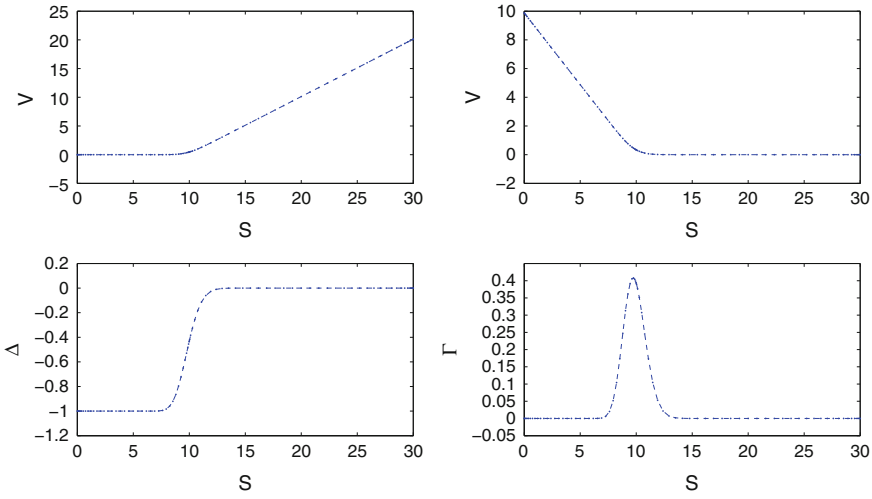

Fig. 1 *Top figures* Europeans call option (*left*), put (*right*). *Bottom figures* Δ (*left*) and Γ (*right*) for European put option. $K = 10$, $r = 0.05$, $\sigma = 0.2$, $S_{\max} = 3K$, $T = 0.25$ $N = 80$

time-marching methods such as Crank-Nicholson's method (with stepsize $2.5e-3$) and the well-known MATLAB solver *ode15s*. These results are presented in Table 2. For the numerical simulations, we fix spatial variable S at $S_{\max} = 3K$ to reduce the domain truncation error. Other parameters are chosen as follows $K = 15$, $\sigma = 0.2$, $r = 0.05$, $T = 0.25$. Maximum absolute errors are calculated using the formula

$$\text{error} = \max_{t\in[0,T]} |\mathbf{V}(t) - \mathbf{V}_M(t)|, \tag{32}$$

where $\mathbf{V}(t)$ is the analytical solution obtained by using the Black–Scholes formula and $\mathbf{V}_M(t)$ is the numerical solution obtained by any of the three methods as indicated in Table 2.

In Fig. 1, we plot values for Europeans call (and put) options as well as the Greeks Δ and Γ. We notice that both Greeks are free of oscillations.

It is worth mentioning here that even though in practice, the use of spectral methods for boundary value problems may be troublesome because the presence of boundaries often introduces stability conditions that are both highly restrictive and often difficult to analyze, one should note that for smooth solutions the results using spectral methods are of a degree of accuracy that local approximation methods cannot produce. For such solutions spectral methods can often achieve an exponential convergence rate as compared to the algebraic convergence rate of finite difference or finite element methods.

One may also think that the matrices in spectral methods are neither sparse nor symmetric, in contrast to the situation in finite differences or finite elements where the sparsity structure of the matrices simplifies the computation. However, the number of discretization points required to achieve the expected accuracy using the spectral method is much less than those required in finite difference or finite element methods, and therefore the spectral method is still very efficient as compared to these other two methods.

Acknowledgments E. Ngounda acknowledges the Agence National des Bourses du Gabon for the financial support. Patidar's research was supported by the South African National Research Foundation.

References

1. Black, F., Scholes, M.: The pricing of options and corporate liabilities. J. Pol. Eco. **81**, 637–659 (1973)
2. Boyle, P.: Options: a Monte-Carlo approach. J. Fin. Eco. **4**, 323–338 (1973)
3. Cox, J.C., Rubinstein, M.: Options Markets. Prentice-Hall, Englewood Cliffs (1985)
4. Duffy, D.G.: Transform Methods for Solving Partial Differential Equations, 2nd edn. Chapman & Hall/CRC, Boca Raton (2004)
5. Golub, G.H., van Loan, C.F.: Matrix Computations, 3rd edn. Johns Hopkins University Press, Baltimore (1996)
6. Huang, W.Z., Sloan, M.D.: The pseudospectral method for solving differential eigenvalue problems. J. Comput. Phys. **111**, 399–409 (1994)
7. Hull, J.C.: Options Futures and Others Derivatives, 4th edn. Prentice-Hall International, New Jersey (2000)
8. Ngounda, E., Patidar, K.C., Pindza, E.: Contour integral method for European options with jumps. Commun. Nonlinear Sci. Numer. Sim. **18**, 478–492 (2013)
9. Oosterlee, C.W., Leentvaar, C.C.W., Huang, X.: Accurate american option pricing by grid stretching and high-order finite differences. Delft University of Technology, The Netherlands, Technical Report (2005)
10. Talbot, A.: The accurate numerical inversion of Laplace transforms. IMA J. Appl. Math. **23**(1), 97–120 (1979)
11. Tangman, D.Y., Gopaul, A., Bhuruth, M.: Numerical pricing of option using high-order compact finite difference schemes. J. Comput. Appl. Math. **218**(2), 270–280 (2008)
12. Tangman, D.Y., Gopaul, A., Bhuruth, M.: Exponential time integration and Chebyshev discretization schemes for fast pricing of options. Appl. Numer. Math. **58**(9), 1309–1319 (2008)

13. Trefeten, L.N.: Spectral Method in MATLAB. SIAM, Philadelphia (2000)
14. Weideman, J.A.C., Reddy, S.C.: A MATLAB differentiation matrix suit. ACM Trans. Math. Soft. **26**, 465–519 (2000)
15. Weideman, J.A.C., Trefethen, L.N.: Parabolic and hyperbolic contours for computing the Bromwich integral. Math. Comput. **76**, 1341–1356 (2007)
16. Welfert, D.B.: Generation of pseudospectral differentiation matrices. I. SIAM J. Numer. Anal. **34**, 1640–1657 (1997)