

Integrating Feynman-Kac Equations Using Hermite Quantic Finite Elements

A White Paper by Rogue Wave Software.

November 2008



Rogue Wave Software
5500 Flatiron Parkway,
Suite 200
Boulder, CO 80301, USA
www.rougewave.com

Integrating Feynman-Kac Equations Using Hermite Quintic Finite Elements

by Rogue Wave Software and Richard Hanson¹

© 2012 by Rogue Wave Software. All Rights Reserved

Printed in the United States of America

Publishing History:

April 2012 - Update

November 2008 – First Publication

Trademark Information

The Rogue Wave Software name and logo, SourcePro, Stingray, HostAccess, IMSL and PV-WAVE are registered trademarks of Rogue Wave Software, Inc. or its subsidiaries in the US and other countries. JMSL, JWAVE, TS-WAVE, PyIMSL and Knowledge in Motion are trademarks of Rogue Wave Software, Inc. or its subsidiaries. All other company, product or brand names are the property of their respective owners.

IMPORTANT NOTICE: The information contained in this document is subject to change without notice. Rogue Wave Software, Inc. makes no warranty of any kind with regards to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Rogue Wave Software, Inc. shall not be liable for errors contained herein or for incidental, consequential, or other indirect damages in connection with the furnishing, performance, or use of this material.

¹ The C version of the VNI software [12] was converted from Fortran 95 by Martin Preiss, Rogue Wave Software, Germany. Preiss re-edited this white paper based on an early draft that used a Fortran 95 implementation for development of the algorithm.

TABLE OF CONTENTS

Abstract.....	4
Introduction	4
Section 1 – Finite Elements and Galerkin Methods.....	5
The Finite Elements	5
The Galerkin Process and Matrix Assembly.....	8
The Boundary Conditions	9
Section 2 – Some Black-Scholes Examples.....	11
Section 3 – A Non-Linear Forcing Term for American Options.....	16
About the Author	20
References	20

Abstract

Methods are given for numerically solving a generalized version of the Feynman-Kac partial differential equation. The solution is expressed as a linear combination of piece-wise Hermite quintic polynomials. This twice-differentiable representation has the attributes of being a high-order method that allows easy evaluation of the solution and certain of its partial derivatives. The time-dependent solution coefficients are computed by finite element Galerkin procedures. Boundary values and initial conditions are required.

Many significant problems in financial modeling can be expressed as particular choices of coefficients, initial conditions, and boundary values. That fact allows this Feynman-Kac solver to be used in many financial modeling applications. These include the Black-Scholes models with European type or American type exercise opportunities on Calls or Puts.

Introduction

This report² describes mathematical and numerical software methods for solving the generalized [Feynman-Kac](#)³ partial differential equation

$$f_t + \mu(x,t)f_x + \frac{\sigma^2(x,t)}{2}f_{xx} - \kappa(x,t)f = \phi(f,x,t), \quad f(x,T) = p(x), \{f_t = \frac{\partial f}{\partial t}, \text{ etc.}\}$$

Equation 1. Feynman-Kac (F-K) Equation with Additional Terms κf and ϕ

The equation is generally solved by integrating backwards, starting at $t = T$. Thus the function $p(x)$ is a terminal condition. Numerical values are computed on a compact interval $[x_{\min}, x_{\max}]$. Boundary conditions are required:

$$a(x,t)f + b(x,t)f_x + c(x,t)f_{xx} = d(x,t), \quad x = x_{\min} \ \& \ x_{\max}$$

Equation 2. F-K Vector Boundary Conditions

The boundary conditions can be vectors of dimension 1, 2 or 3. Details on their form and smoothness requirements are given below in the section on boundary conditions.

This F-K equation includes a number of interesting models as special cases. One of our primary interests is the Black-Scholes equation for the value $f(S,t)$ of an option [1]:

² This work benefited by discussions with Fred Krogh, Gerald Hanweck, John Brophy and Martin Preiss.

³ http://en.wikipedia.org/wiki/Feynman-Kac_formula

$$\frac{\partial f}{\partial t} + (r - D)S \frac{\partial f}{\partial S} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 f}{\partial S^2} - rf = 0, \quad S_{\min} \leq S \leq S_{\max}, \quad t_{\text{invest}} \leq t \leq t_{\text{expiry}}$$

Equation 3. Black-Scholes Model, a Special Case of F-K

Here r is the instantaneous risk-free interest rate, D is the continuous dividend of the instrument, and σ is the continuous positive volatility⁴ of that instrument. There are also boundary conditions, as in Equation 2, which will be defined and used as required. A part of this report gives numerical results for particular choices occurring in the Black-Scholes model. The CNL functions `imsl_f_feynman_kac` and `imsl_f_feynman_kac_evaluate` (documented in [12]) are used to compute these models.

Section 1 – Finite Elements and Galerkin Methods

The mathematical methods we use for solving F-K equations are well-known and tested (See [2], p.89). We give a detailed description of the finite elements and the construction of the matrices that arise when solving the Differential-Algebraic equations (DAE) for the solution.

The Finite Elements

This development begins with a grid of points or a partition of the interval $[x_{\min}, x_{\max}]$. Write this grid as $(x_{\min} =) x_1 < x_2 < \dots < x_m (= x_{\max})$. We see that it is a decomposition of the interval into $N_{\text{int}} = m - 1$ abutting subintervals. For each fixed t , on each subinterval, the unknown solution is given by a Hermite quintic polynomial. An issue that we do not discuss is placement of these grid points. The development given here is a more general method than that found in [11].

The polynomial is represented by six basis polynomials. Thus in the interval

$$x \in [x_i, x_{i+1}], \quad h_i = x_{i+1} - x_i, \quad z = (x - x_i) / h_i, \quad z \in [0, 1],$$

the function and its first three derivatives are given by

$$\begin{aligned} f(x, t) &= f_i b_0(z) + f_{i+1} b_0(1-z) + h_i f_i' b_1(z) - h_i f_{i+1}' b_1(1-z) + h_i^2 f_i'' b_2(z) + h_i^2 f_{i+1}'' b_2(1-z) \\ f_x(x, t) &= h_i^{-1} (f_i b_0'(z) - f_{i+1} b_0'(1-z) + h_i f_i' b_1'(z) + h_i f_{i+1}' b_1'(1-z) + h_i^2 f_i'' b_2'(z) - h_i^2 f_{i+1}'' b_2'(1-z)) \\ f_{xx}(x, t) &= h_i^{-2} (f_i b_0''(z) + f_{i+1} b_0''(1-z) + h_i f_i' b_1''(z) - h_i f_{i+1}' b_1''(1-z) + h_i^2 f_i'' b_2''(z) + h_i^2 f_{i+1}'' b_2''(1-z)) \\ f_{xxx}(x, t) &= h_i^{-3} (f_i b_0^{(3)}(z) - f_{i+1} b_0^{(3)}(1-z) + h_i f_i' b_1^{(3)}(z) + h_i f_{i+1}' b_1^{(3)}(1-z) + h_i^2 f_i'' b_2^{(3)}(z) - h_i^2 f_{i+1}'' b_2^{(3)}(1-z)) \end{aligned}$$

Equation 4. Partial Derivatives of the Piecewise Solution

⁴ In the F-K equation identify the general term $\sigma(x, t)$ with the Black-Scholes volatility factor multiplied by the price—i.e. σS .

For our purpose the values $f_i, f'_i, f''_i, f_{i+1}, f'_{i+1}, f''_{i+1}$ are time-dependent coefficients associated with that interval. The local error term is $O(h_i^6)$. The basis functions and their partial derivatives are given by the formulas

$$\begin{aligned}
 b_0(z) &= -6z^5 + 15z^4 - 10z^3 + 1 = (1-z)^3(6z^2 + 3z + 1), \\
 b_1(z) &= -3z^5 + 8z^4 - 6z^3 + z = (1-z)^3 z(3z + 1), \\
 b_2(z) &= \frac{1}{2}(-z^5 + 3z^4 - 3z^3 + z^2) = \frac{1}{2}(1-z)^3 z^2 \\
 b'_0(z) &= -30(1-z)^2 z^2; \quad (b''_0) b_0^{(2)}(z) = -60z(2z^2 - 3z + 1); \quad b_0^{(3)}(z) = -60(6z^2 - 6z + 1) \\
 b'_1(z) &= -(1-z)^2(15z^2 - 2z - 1); \quad b_1^{(2)}(z) = 12z(1-z)(5z - 3); \quad b_1^{(3)}(z) = -12(15z^2 - 16z + 3) \\
 b'_2(z) &= -\frac{1}{2}(1-z)^2 z(5z - 2); \quad b_2^{(2)}(z) = (1-z)(10z^2 - 8z + 1); \quad b_2^{(3)}(z) = -3(10z^2 - 12z + 3)
 \end{aligned}$$

Equation 5. Hermite Quintic Basis Functions and Three Derivatives

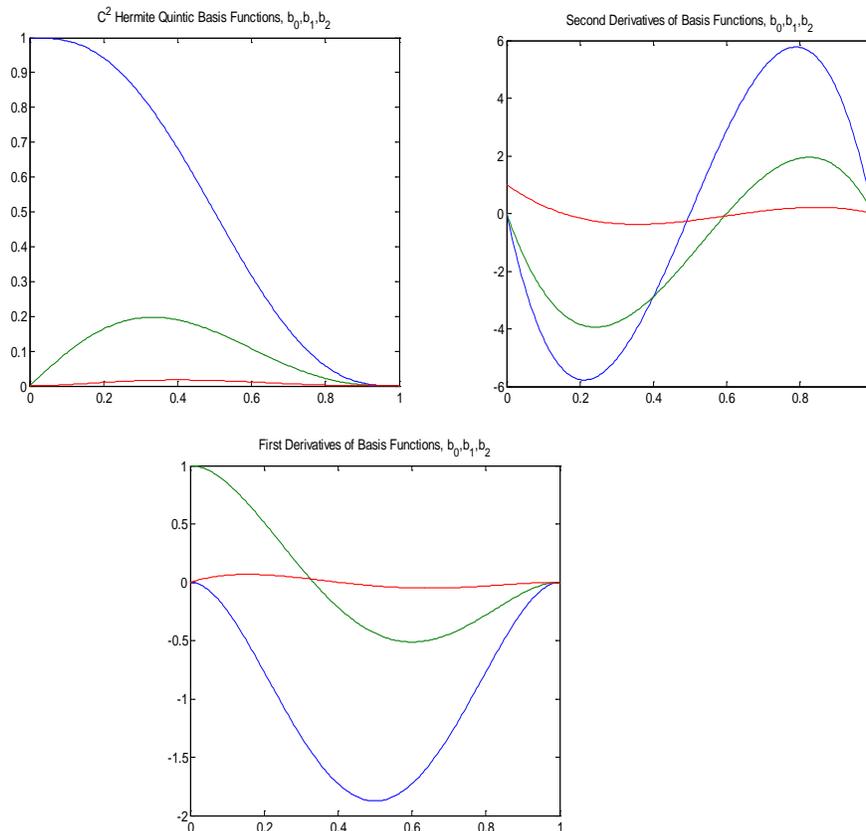


Figure 1. Graphs of Hermite Basis Functions and Two Derivatives

During progression from one interval to the next, there is overlap with three coefficients from the adjacent interval. This observation and the formulas for the derivatives show that for each fixed t , the function $f(x, t)$ is continuous at the interior grid points and has two continuous derivatives with respect to x . Formulas for the piece-wise quadratic third derivatives are included here, even though that function is no longer generally continuous at the interior points. This third derivative function may be of interest to users. The total number of parameters on this grid is $3(N_{\text{int}} + 1)$. The rationale for choosing this set of elements is based on these observations:

- Since the local error term is of order $O(h_i^6)$, this translates into fewer unknowns than a finite difference method. Typically there is high accuracy when using even a modest-sized grid. This should result in efficient computational algorithms since the size of the system of differential-algebraic equations and thus the associated linear algebra will be of a smaller dimension. There is also special banded structure for the matrices of the linear algebra. This allows for the use of tailored banded matrix linear system solvers and matrix-vector products.
- The functions and intrinsic continuous partial derivatives $\{f, f_x, f_{xx}, f_t, f_{tx}, f_{ttx}\}$ are available by direct evaluation of the solution and its derivative functions. These additional partial derivative values are often required by users and they can be obtained at essentially no extra cost in computing time. In the case of the Black-Scholes model these functions include many of [The Greeks](#)⁵. The CNL function **imsl_f_feynman_kac_evaluate** can compute these derivatives, which we call the *Intrinsic Greeks*. Evaluation can occur for any set of values of x at a given t . There is no resulting sub-problem such as interpolation. The evaluator does that for the user.
- For some applications it is necessary to interpolate in the orthogonal direction for the solution as a function of t with x fixed. A way to do this is assemble values $\{f, f_t\}$ at a grid of t values at a fixed x . Then perform Hermite cubic interpolation at any set of values within this grid of t values. This C^1 cubic has a local error of $O(\Delta t_i^4)$ and is given by the formulas

$$\begin{aligned}\hat{b}_0(z) &= (1+2z)(1-z)^2, & \hat{b}_1(z) &= z(1-z)^2 \\ \hat{b}'_0(z) &= -6(1-z)z, & \hat{b}'_1(z) &= 3z^2 - 4z + 1\end{aligned}$$

To interpolate at any $t \in [t_i, t_{i+1}]$ compute

$$\begin{aligned}f(x, t) &= f_i \hat{b}_0(z) + f_{i+1} \hat{b}_1(z) + \Delta t_i f'_i \hat{b}'_0(z) - \Delta t_i f'_{i+1} \hat{b}'_1(z), & \Delta t_i &= t_{i+1} - t_i, \\ z &= (t - t_i) / \Delta t_i, & f'_i &= \frac{\partial f}{\partial t}(x, t_i), \text{ etc.}\end{aligned}$$

⁵ http://en.wikipedia.org/wiki/The_Greeks

- The complexity of these representations and the quintic basis functions are a concern. Verifying the formulas was aided by the use of algebraic manipulation software. The Hermite quintic formulas have been coded in `imsf_feynman_kac_evaluate`. Testing with the Black-Scholes equation has yielded results consistent with analytic solutions.

The Galerkin Process and Matrix Assembly

Using a development similar to that of [2], p. 116, write the F-K equation in divergence form

$$\Lambda(f) - \phi \equiv \left(\frac{\partial}{\partial t} + (\mu(x,t) - \sigma\sigma') \frac{\partial}{\partial x} + \frac{\partial}{\partial x} \left(\frac{\sigma^2(x,t)}{2} \frac{\partial}{\partial x} \right) - \kappa(x,t) \right) f - \phi = 0$$

$$R(x,t) \equiv f_t + (\mu(x,t) - \sigma\sigma') f_x + \left(\frac{\sigma^2(x,t)}{2} f_x \right)_x - \kappa(x,t) f - \phi = 0, \{ \sigma' = \frac{\partial \sigma}{\partial x} \}$$

By adding the piece-wise definitions we may arrange the unknown solution function as the

series $f(x,t) = \sum_{i=1}^{3(N_{\text{int}}+1)} y_i(t) \beta_i(x)$. The coefficient functions are defined by re-labeling:

$y_1(t) = f(x_1, t)$, $y_2(t) = f_x(x_1, t)$, $y_3(t) = f_{xx}(x_1, t)$, etc. Then impose the Galerkin

orthogonality conditions $\int_{x_{\min}}^{x_{\max}} R(x,t) \beta_j(x) dx = 0$, $j = 1, \dots, 3(N_{\text{int}}+1)$. Expand and integrate by parts the third term in each expression to yield

$$\int_{x_{\min}}^{x_{\max}} \left(\frac{\sigma^2(x,t)}{2} f_x \right)_x \beta_j(x) dx = \left. \frac{\sigma^2(x,t)}{2} f_x \beta_j(x) \right|_{x_{\min}}^{x_{\max}} - \int_{x_{\min}}^{x_{\max}} \left(\frac{\sigma^2(x,t)}{2} f_x \right) \beta_j'(x) dx.$$

Collect all terms after performing this integration to get five matrices contributing to the DAE

$$M\tilde{y}' + N\tilde{y} - R\tilde{y} - K\tilde{y} + Q\tilde{y} - \tilde{p} = M\tilde{y}' + (N - R - K + Q)\tilde{y} - \tilde{p},$$

where

$$\tilde{y} := \tilde{y}(t) = \left[y_1(t), \dots, y_{3(N_{\text{int}}+1)}(t) \right]^T, \quad \tilde{y}' = \frac{d\tilde{y}}{dt},$$

$$\tilde{\beta} := \tilde{\beta}(x) = \left[\beta_1(x), \dots, \beta_{3(N_{\text{int}}+1)}(x) \right]^T, \quad \tilde{\beta}' = \frac{d\tilde{\beta}}{dx},$$

$$M = \int \tilde{\beta} \tilde{\beta}^T, \quad N = \int (\mu - \sigma\sigma') \tilde{\beta} \tilde{\beta}^T, \quad R = \int \frac{\sigma^2}{2} \tilde{\beta}' \tilde{\beta}'^T,$$

$$K = \int \kappa \tilde{\beta} \tilde{\beta}^T, \quad Q = \frac{\sigma^2(x_{\max}, t)}{2} \tilde{\beta}(x_{\max}) \tilde{\beta}'(x_{\max})^T - \frac{\sigma^2(x_{\min}, t)}{2} \tilde{\beta}(x_{\min}) \tilde{\beta}'(x_{\min})^T,$$

$$\tilde{p} = \int \phi \tilde{\beta}.$$

The integration is approximated using Gauss-Legendre (G-L) quadrature normalized to the interval $[0,1]$, see [3], p. 390. The leading matrix term M will be the integral of products of resulting degree 10. It will not be dependent on t nor will it depend



on any coefficient in the F-K equation. This means it will only have to be computed once. To exactly integrate this symmetric matrix an order 6 G-L formula is used on each subinterval. This is because the error term for G-L quadrature has a factor involving the 12-th derivative. This will be identically zero on each subinterval. However using just an order 5 formula gives an error term involving the 10-th derivative which is a non-zero constant. [See [3], (8.5.6), p. 391.] The user can reset the order of the G-L formula to a higher or lower value to deal with particular behavior in the remaining coefficients. Note that the exact form of the polynomial representation for $\beta_j(x)$ is not an issue. Only the 6 evaluations are required to integrate the leading term M exactly, apart from rounding errors.

With the particular basis of quintic splines, the only non-zero value of $\beta_j(x_{\max})$ is at the right end, or $\beta_{3N_{\text{int}}+3}(x_{\max}) = 1$, and for $\beta'_{3N_{\text{int}}+2}(x_{\max}) = h_{N_{\text{int}}}$. Similarly, the only non-zero value of $\beta_j(x_{\min})$ is at the left end, or $\beta_1(x_{\min}) = 1$, and for $\beta'_2(x_{\min}) = h_1$. Thus we have the last component of $Q\tilde{y}$ with the value $h_{N_{\text{int}}} \frac{\sigma^2(x_{\max}, t)}{2} y_{3N_{\text{int}}+2}(t)$ and the first component with the value $-h_1 \frac{\sigma^2(x_{\min}, t)}{2} y_2(t)$. Notice that the boundary conditions have not yet been used in formulating the DAE. We then consider the intermediate problem DAE

$$M\tilde{y}' + N\tilde{y} - R\tilde{y} - K\tilde{y} + Q\tilde{y} - \tilde{p} \equiv M\tilde{y}' + P\tilde{y} - \tilde{p} = 0, \{ \tilde{y}' = \frac{d\tilde{y}}{dt}, \tilde{p} = \int \phi \tilde{\beta} \}$$

Equation 6. DAE of Coefficients -- No Boundary Conditions Provided

For computational efficiency it is important to note when the matrix $P = N - R - K + Q$ has dependence on t . If there is no dependence this matrix can be computed once, at the outset of integration. It is thereafter not necessary to re-compute it at every time step. If any coefficient in the set $\{\mu, \kappa, \sigma, \sigma'\}$ has time dependence then some terms of P must be re-computed at each time step. The Black-Scholes Equation 3, with constant values of D, r, σ is a case where there is no time dependence in the accumulated element matrices. Also note that in this problem an order 6 G-L formula exactly integrates each term of the matrix P for the Black-Scholes model with constant $\{r, \sigma\}$. This follows from the fact that the polynomial degree in each integrand is 10.

The Boundary Conditions

Following the derivation of the DAE system that follows from applying the Galerkin method, the boundary value conditions of Equation 2 are applied. These end conditions then take precedence over those in Equation 5. Thus we replace the first and last equations by the corresponding boundary value equations. After this replacement the DAE system can be written as $U\tilde{y}' + V\tilde{y} - w = 0$. The first and last rows of U are zero. The remaining rows are those of M . The first and last rows of V are replaced by the values of the boundary value coefficients from Equation 2. The remaining rows are



those of P . The vector w is \tilde{p} except in the first and last components that correspond to the values $d(x, t)$ at the boundaries. A technical point here is that the boundary conditions imply that the first and last rows and V and w may depend on t even though the remaining rows or components do not.

During the integration of the DAE there is a requirement to solve a linear algebraic system with a coefficient matrix $A = \gamma U + V$. The value of γ depends on details of the integration, including step-size and order of the backward differentiation formulas, [4]. The boundary values must be specified so that for some range of values for γ the banded matrix A is non-singular. This property defines the DAE to be an *index 1 problem*. Existing DAE software such as **DDASLX**, [5], translated into CNL, is used for its solution. According to the [LAPACK definition and data structure](#)⁶ for a [banded matrix](#)⁷, the upper and lower bandwidth of A is 5 and the matrix bandwidth is 11. The fact that A is banded implies that efficient algorithms for solving banded matrix linear system problems can be applied. There is also a requirement to compute the residual function of the DAE $F(\tilde{y}', \tilde{y}, t) = U\tilde{y}' + V\tilde{y} - w = 0$. Since A , U and V have the same banded properties, [Level-2 BLAS](#)⁸, namely the CLAPACK version of **SGBMV**, are used for computing these matrix-vector products in the right-hand side of $F(\tilde{y}', \tilde{y}, t)$.

For starting the integration it is necessary to have initial values of the coefficient set $\tilde{y} = \tilde{y}_0$. For this we need to represent the initial data in terms of the Hermite piecewise quintic coefficients. To that end we compute the local contribution $u := \int p(x)\beta(x) dx$. With this available we compute the least-squares solution for the initial coefficients by minimizing $\int (p(x) - \beta(x)^T \tilde{y}_0)^2 dx$. The particular choice that minimizes this least-squares condition is given by solving $M \tilde{y}_0 = u$ after accumulation over local contributions over all subintervals. This solution is obtained by using the left and right boundary conditions as equality constraints and the remaining rows as least squares equations. This solution for \tilde{y}_0 is used as the starting values for the coefficients of the integration.

For the Black-Scholes equation the vector u is typically not integrated exactly. It costs very little to increase the degree of the Gauss-Legendre formula, so the degree used in building the matrices is a parameter of the software. This process does not provide the time derivatives \tilde{y}'_0 . The values for \tilde{y}'_0 are computed using the regularization method given in Appendix II of [10]. There are cases where the initial values \tilde{y}_0 should be first provided from basic principles without relying on the least-squares estimation process outlined above. This information can be communicated using function **imsl_f_feynman_kac** together with optional argument **IMSL_FCN_INIT** (see [12]).

⁶ <http://www.netlib.org/lapack/lug/node124.html>

⁷ http://en.wikipedia.org/wiki/Band_matrix

⁸ <http://www.netlib.org/lapack/lug/node145.html>

Section 2 – Some Black-Scholes Examples

As mentioned, the reasons for using the Hermite quintic basis elements are their small truncation error and the smoothness of the intrinsic partial derivatives at an evaluation time. There is no significant additional expense to obtain the partials. Their shapes are illustrated with a pair of Vanilla Options using the Black-Scholes model of Equation 3. The intrinsic third partial derivative or Greek (*speed*) is not a smooth function. One could re-do the finite element model using Hermite-seven degree splines. These will have three continuous intrinsic partial derivatives, including *speed*. We believe it is not likely that accurate values of *speed* are required by financial analysts.

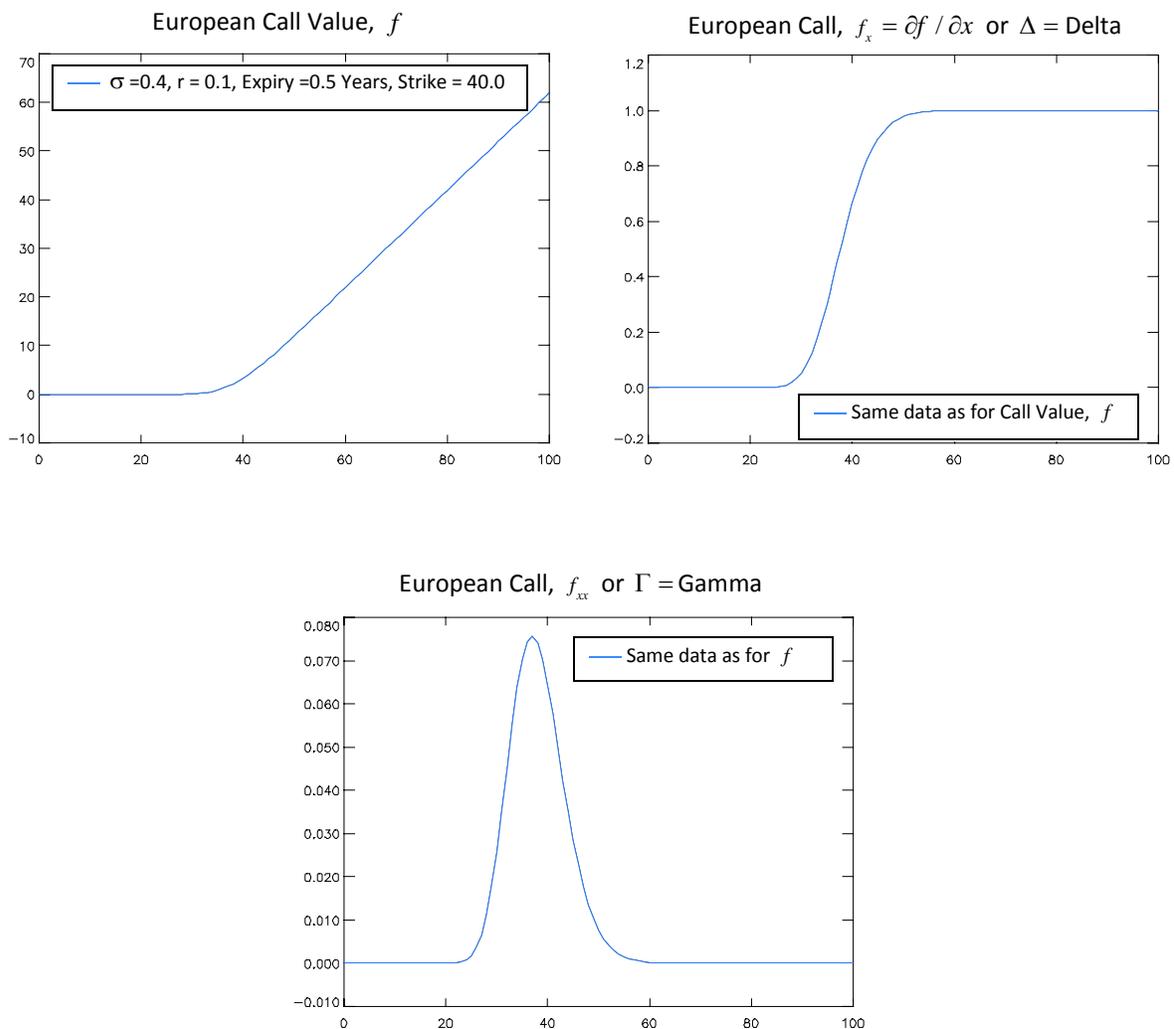


Figure 2. Numerical results with intrinsic greeks for European Call

Values of Black-Scholes PDE Coefficients	$\sigma = 0.2, r = 0.1, T = 0.5, E = 40, D = 0$
Requested Integration Accuracy	$\text{atol} = \text{rtol} = 10^{-6}$
Number of Intervals	100
Number of Evaluations ⁹ and Factorizations	281, 24

Table 1. Black-Scholes Equation Solution, European Call. See [7], p. 116

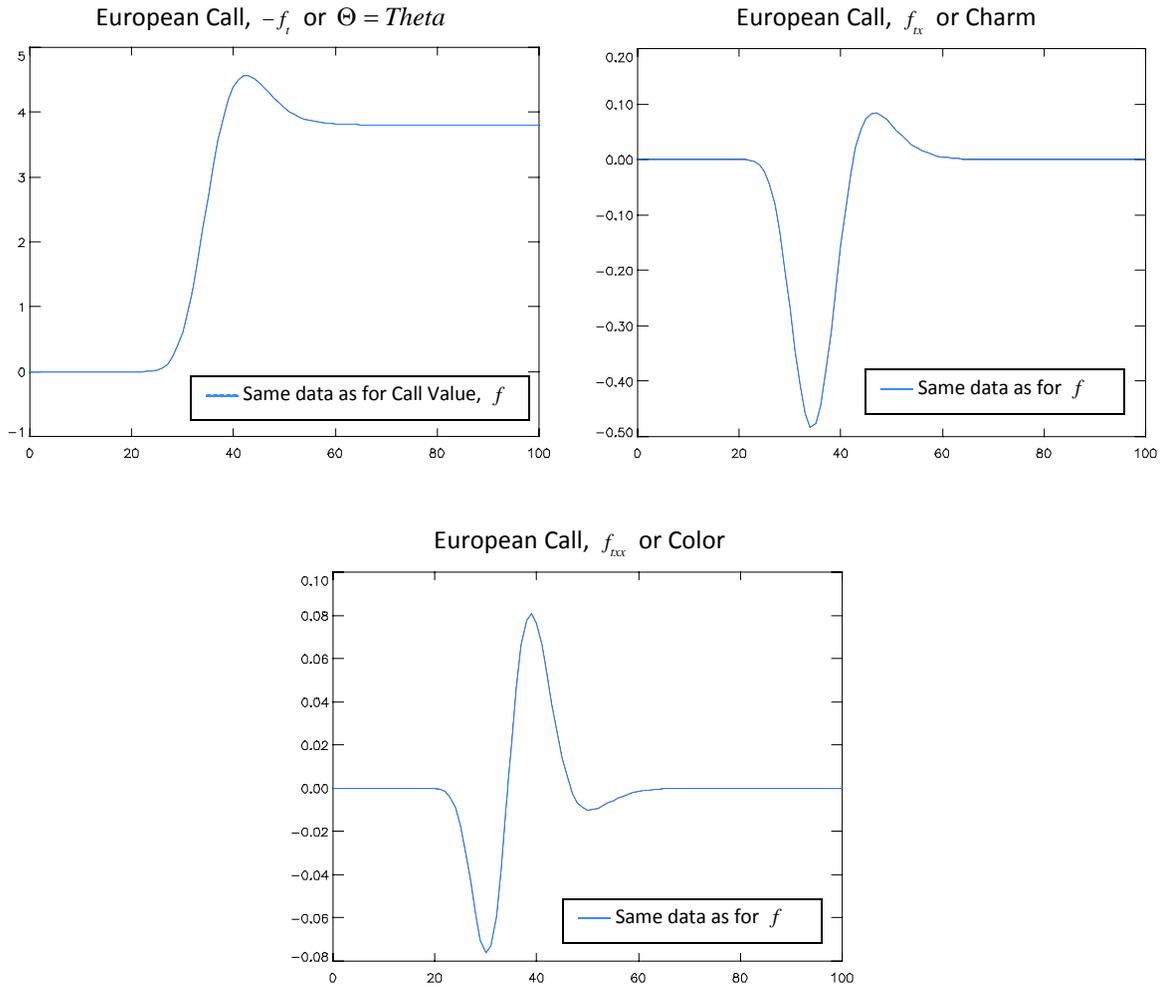


Figure 3. More Numerical Results of Intrinsic Greeks with European Call

⁹ This is the number of evaluations of the residual DAE function $F(\tilde{y}', \tilde{y}, t) = U\tilde{y}' + V\tilde{y} - w$ and the number of solve steps after a banded matrix factorization. The boundary conditions are given by a 3×3 identity matrix at both ends and respective values $d = [0, 0, 0]^T$ at the left and $d = [x_{\max} - Ee^{-r(T-t)}, 1, 0]^T$ at the right end.

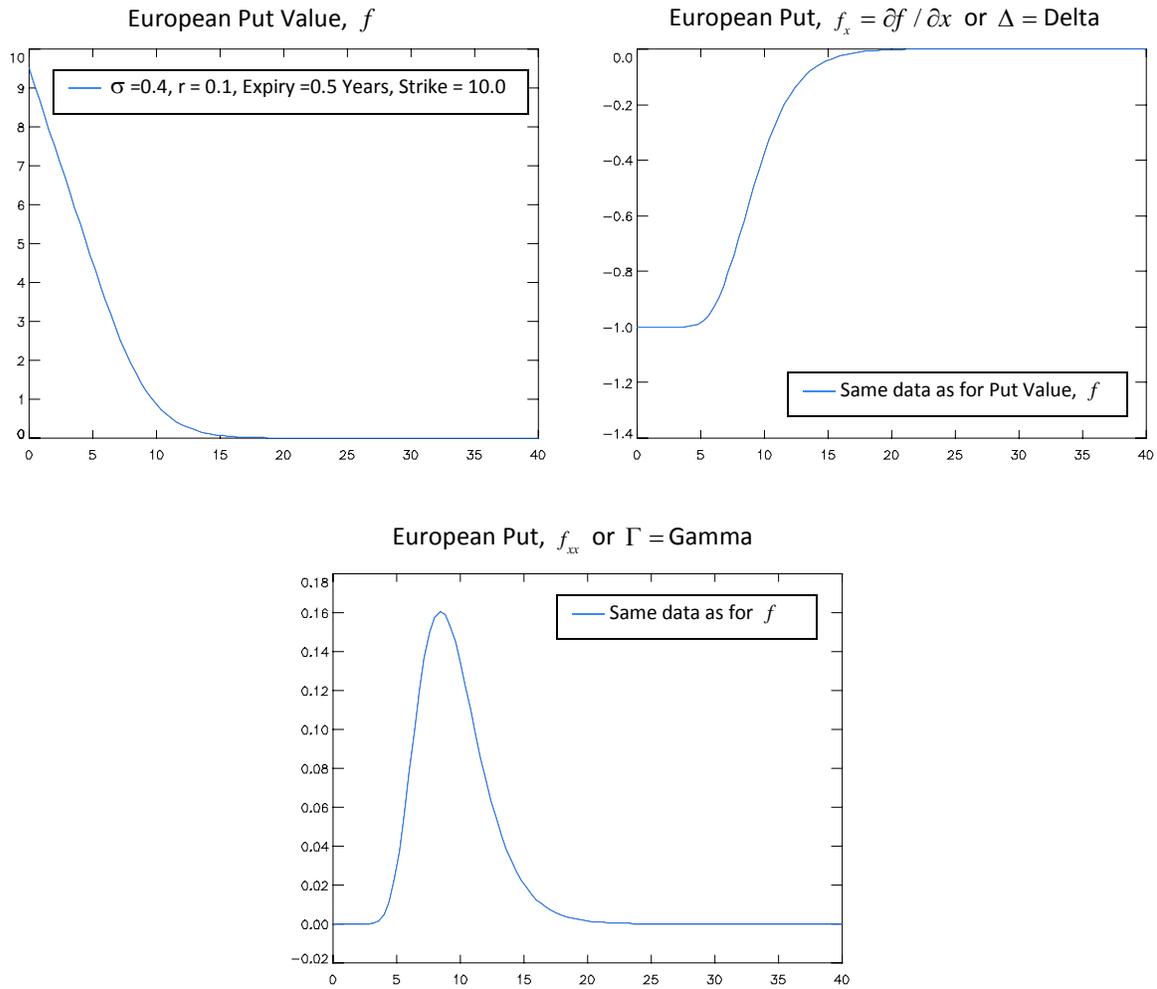


Figure 4. Numerical Results with Intrinsic Greeks for European Put

Values of Black-Scholes PDE Coefficients	$\sigma = 0.4, r = 0.1, T = 0.5, E = 10, D = 0$
Requested Integration Accuracy	$\text{atol}=\text{rtol}=10^{-6}$
Number of Intervals	100
Number of Evaluations ¹⁰ and Factorizations	316, 27

Table 2. Black-Scholes Equation Solution, European Put. See [1], p. 176

¹⁰ This is the number of evaluations of the residual DAE function $F(\tilde{y}', \tilde{y}, t) = U\tilde{y}' + V\tilde{y} - w$ and the number of solve steps after a banded matrix factorization. The boundary conditions are given by respective values $d = [-1, 0]^T$ at the left end and $d = [0, 0, 0]^T$ at the right end.

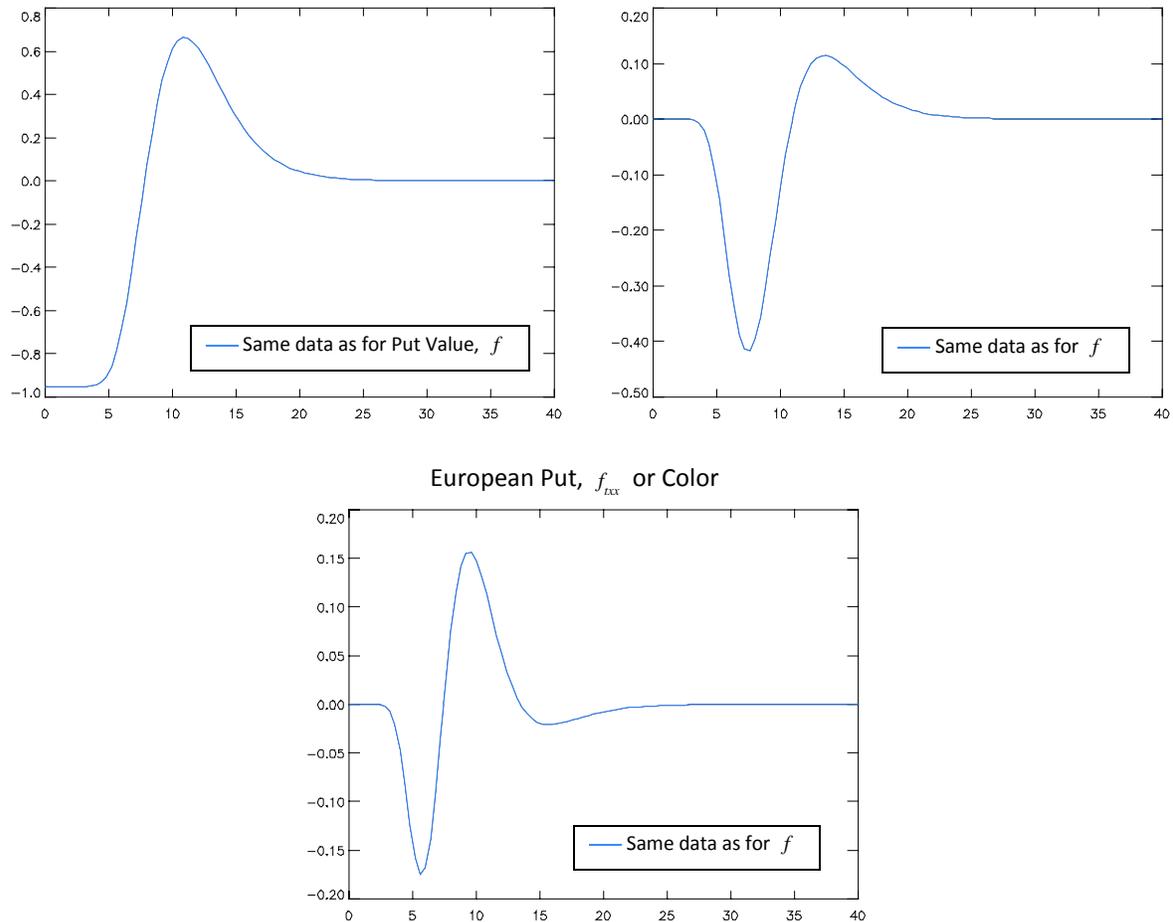


Figure 5. More Numerical Results of Intrinsic Greeks with European Put

The evaluation of the value or intrinsic Greeks does not give all the function information commonly used by financial analysts. For example there are variational equations in the Black-Scholes model that may be required. Two common ones are $\frac{\partial f}{\partial r}$, $\frac{\partial f}{\partial \sigma}$. These are known as *Rho* and *Vega*. They can be integrated as additional variational equations, e. g. [7], pp. 126-132. For constant interest rates and volatility factors an elementary approach is to use divided differences but with care to choose the optimal value of the difference.

The following example shows some numerical results for a plain Vanilla European Put ([Source code](#)). The data for this put in the Black-Scholes model are listed in Table 3.

Parameter	Value
Volatility σ	0.4
Interest rate r	0.1
Time to Maturity	3 months, 6 months
Strike Price E	10.0
Dividend D	0

Table 3. Parameters for Plain Vanilla European Put

For plain Vanilla European Puts, the Black-Scholes Equation has an analytical solution. Setting

$$a := \frac{\log(S/E) + (r + \sigma^2/2) \Delta T}{\sigma \sqrt{\Delta T}},$$

where $\Delta T := T - t$ denotes the time to expiry, the value of the put at stock price S and time point t is given by

$$f(S, t) = -S \Phi(-a) + E e^{-r \Delta T} \Phi(-a + \sigma \sqrt{\Delta T}).$$

Here, $\Phi(\cdot)$ is the standard normal distribution function.

The numerical values are compared with these analytical solutions.

European Option Premium for Vanilla Put, 3 and 6 Months Prior to Expiry						
Number of equally spaced spline knots: 101						
Number of unknowns: 303						
Strike= 10.00, sigma= 0.40, Interest Rate= 0.10						
Environment: SPARC(64-bit), Solaris 10						
Compiler: C-compiler from Sun Studio 12						
Integration time (in CPU secs): 0.850879						
		3 months		6 months		
Stock price	Formula	PDE	Formula	PDE		
0.0	9.75310	9.75310	9.51229	9.51229		
2.0	7.75310	7.75310	7.51229	7.51229		
4.0	5.75310	5.75310	5.51282	5.51282		
6.0	3.75689	3.75689	3.55829	3.55829		
8.0	1.90243	1.90243	1.91810	1.91810		
10.0	0.66939	0.66938	0.87033	0.87033		
12.0	0.16751	0.16751	0.34769	0.34769		
14.0	0.03262	0.03262	0.12792	0.12792		
16.0	0.00539	0.00539	0.04484	0.04484		
	Greeks at 3 months					
Stock price	Put value	Delta	Gamma	Theta	Charm	Color
0.0	9.7531	-1.0000	0.0000	-0.9753	-0.0000	0.0000
2.0	7.7531	-1.0000	-0.0000	-0.9753	-0.0000	-0.0000
4.0	5.7531	-1.0000	0.0000	-0.9753	-0.0003	-0.0015
6.0	3.7569	-0.9901	0.0221	-0.9062	-0.1472	-0.2415
8.0	1.9024	-0.8135	0.1677	0.0176	-0.7194	-0.0651
10.0	0.6694	-0.4110	0.1945	1.0780	-0.1750	0.4087



12.0	0.1675	-0.1278	0.0871	0.8336	0.2872	0.0383
14.0	0.0326	-0.0282	0.0231	0.3195	0.1886	-0.0823
16.0	0.0054	-0.0050	0.0045	0.0842	0.0616	-0.0405
Greeks at 6 months						
Stock price	Put value	Delta	Gamma	Theta	Charm	Color
0.0	9.5123	-1.0000	0.0000	-0.9512	-0.0000	0.0000
2.0	7.5123	-1.0000	0.0000	-0.9512	-0.0000	-0.0000
4.0	5.5128	-0.9983	0.0049	-0.9443	-0.0199	-0.0464
6.0	3.5583	-0.9316	0.0777	-0.6910	-0.2802	-0.1679
8.0	1.9181	-0.6811	0.1578	0.0713	-0.3954	0.0755
10.0	0.8703	-0.3752	0.1341	0.6105	-0.1207	0.1477
12.0	0.3477	-0.1678	0.0739	0.6157	0.0819	0.0507
14.0	0.1279	-0.0658	0.0323	0.4020	0.1115	-0.0101
16.0	0.0448	-0.0239	0.0124	0.2116	0.0755	-0.0206

Table 4. Numerical results for Plain Vanilla European Put

Section 3 – A Non-Linear Forcing Term for American Options

An approximate solution of the American Option (exercise at any time) is obtained by a *forcing or source term*, [7], (2.40)-(2.41):

$$f_t + \mu(x,t)f_x + \frac{\sigma^2(x,t)}{2}f_{xx} - \kappa(x,t)f = \phi(f, x, t), \quad f(x, T) = p(x), \{f_t = \frac{\partial f}{\partial t}, \text{ etc.}\}$$

$$\phi = 0, \quad \forall f > \max(K - x, 0)$$

$$\phi < 0, \quad \forall f = \max(K - x, 0)$$

We use the particular forcing function $\phi = -rK(\varepsilon/(f + \varepsilon - (K - x)))^\mu$, $\mu \geq 1$, $0 < \varepsilon \ll 1$ for evaluating the American Put option, as suggested in [8]. Our experiments lead to the values $\mu = 2$ and $\varepsilon = 10^{-3}$ as workable parameters. The value $\mu = 1$ was used in [8].

It is necessary to evaluate the contribution of partial derivatives of the Jacobian matrix of the integral term $\tilde{p} = \int \phi \tilde{\beta}$. Focus on a typical interval, say the first one $[x_1, x_2]$. Using the Gauss-Legendre formulas yields a local contribution

$$h_1 \sum_k w_k \beta_j(\tilde{x}_k) \phi(f, \tilde{x}_k, t) =$$

$$-h_1 r K \sum_k w_k \beta_j(\tilde{x}_k) (\varepsilon / (f(\tilde{x}_k, t) + \varepsilon - (K - \tilde{x}_k)))^\mu$$

The values w_k, \tilde{x}_k are the weights and translated abscissas for the G-L formula. For this interval

$$f(x, t) = y_1 b_0(z) + y_4 b_0(1-z) + h y_2 b_1(z) - h y_5 b_1(1-z) + h^2 y_3 b_2(z) + h^2 y_6 b_2(1-z)$$



From this we differentiate with respect to the y_i in this single interval. The contributions from all the intervals are accumulated as an outer step.

$$\frac{\partial}{\partial y_i} h_1 \sum_k w_k \beta_j(\tilde{x}_k) \phi(f, \tilde{x}_k, t) = -h_1 \mu \sum_k w_k \beta_j(\tilde{x}_k) \beta_i(\tilde{x}_k) \phi(f, \tilde{x}_k, t) / (f(\tilde{x}_k, t) + \varepsilon - (K - \tilde{x}_k))$$

This non-linear forcing term is useful for obtaining modest accuracy of the value of the American put function and related Greeks or partials of the function. It has the primary attribute of leading to a DAE that is solved with reasonable efficiency. Alternative methods appear to be relatively expensive or do not yield the Greeks without resorting to divided differences. On the other hand the amount of accuracy that can be obtained with this forcing term may be limited. This is difficult to measure without an alternative source or tables for comparison. There is 2-4 place agreement with tables found in [1], p. 176, and [9], p. 127. It is possible that (more expensive) adaptive quadrature for $\tilde{p} = \int \phi \tilde{\beta}$ would be helpful to get accurate results in the solution. That was not investigated.

These graphics show the option value and two of the Greeks for the American Put example presented by [9], Table 1, p. 127.

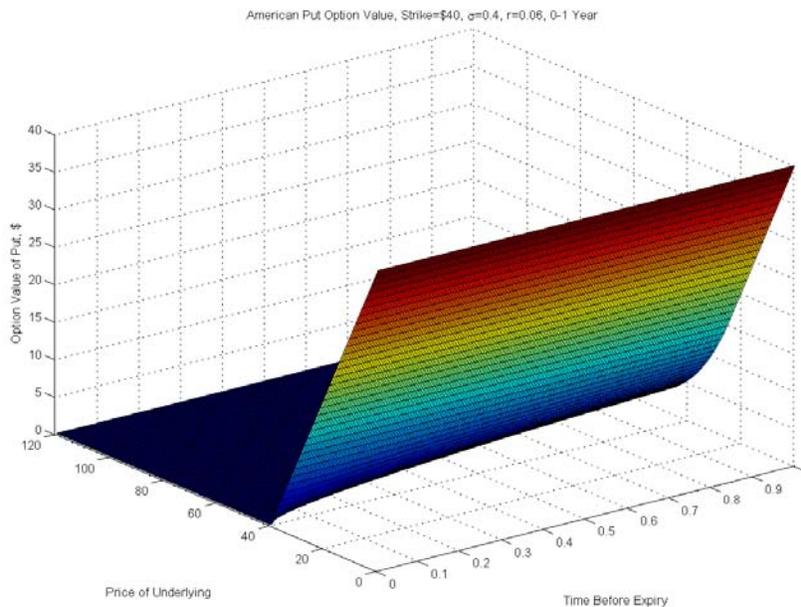


Figure 6. Value of American Put Option Over One Year

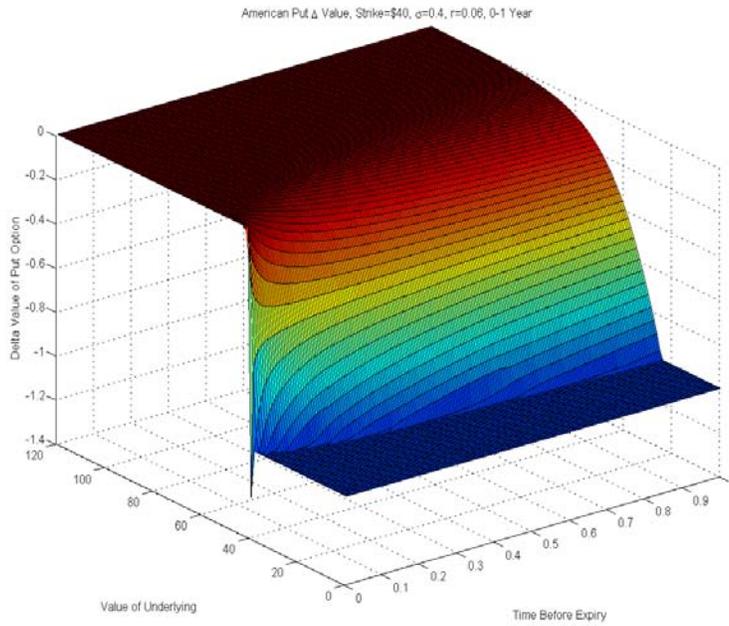


Figure 7. The Value of Delta on the American Put Over One Year

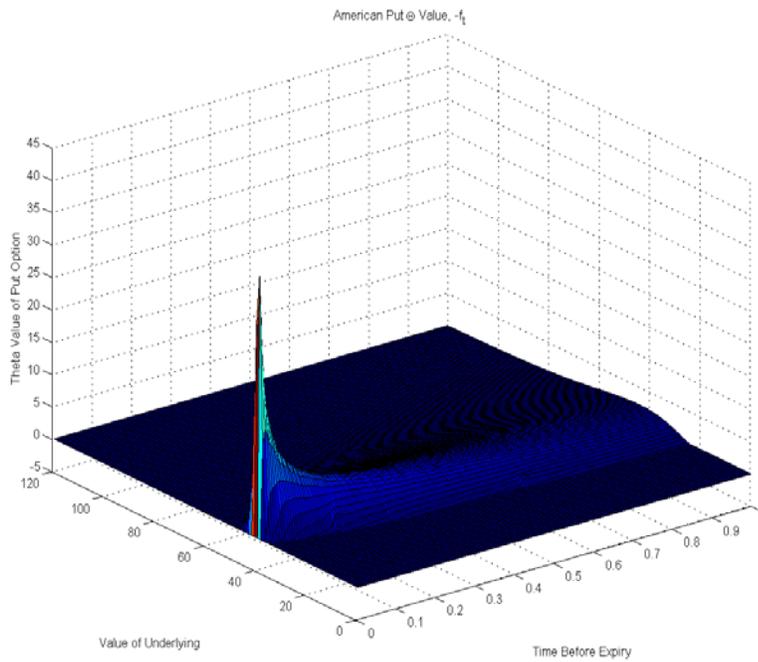


Figure 8. The Value of Theta on the American Put

The following table lists some numerical values of the put and its greeks shown graphically in Figures 6 – 8 ([Source code](#)).

American Option Premium for Vanilla Put, 3 and 6 Months Prior to Expiry				
Number of equally spaced spline knots: 101				
Number of unknowns: 303				
Strike= 40.00, sigma= 0.40, Interest Rate= 0.06				
Environment: SPARC(64-bit), Solaris 10				
Compiler: C-compiler from Sun Studio 12				
Integration time (in CPU secs): 51.041613				
Stock price	Put Value			
	3 months	6 months		
10.0	30.00000	30.00000		
20.0	20.00000	20.00000		
30.0	10.02310	10.25323		
40.0	2.92329	3.97848		
50.0	0.51123	1.28028		
60.0	0.06272	0.36811		
70.0	0.00625	0.09991		
80.0	0.00056	0.02644		
90.0	0.00005	0.00657		
Greeks at 3 months				
Stock price	Put value	Delta	Gamma	Theta
10.0	30.0000	-1.0000	0.0000	0.0000
20.0	20.0000	-1.0000	-0.0000	-0.0000
30.0	10.0231	-0.9586	0.0402	0.5501
40.0	2.9233	-0.4410	0.0511	5.3133
50.0	0.5112	-0.0997	0.0176	3.1959
60.0	0.0627	-0.0139	0.0030	0.8014
70.0	0.0063	-0.0015	0.0003	0.1288
80.0	0.0006	-0.0001	0.0000	0.0163
90.0	0.0000	-0.0000	0.0000	0.0018
Greeks at 6 months				
Stock price	Put value	Delta	Gamma	Theta
10.0	30.0000	-1.0000	0.0000	-0.0000
20.0	20.0000	-1.0000	0.0000	0.0000
30.0	10.2532	-0.8505	0.0451	1.1019
40.0	3.9785	-0.4191	0.0367	3.4490
50.0	1.2803	-0.1538	0.0171	2.8736
60.0	0.3681	-0.0472	0.0058	1.4925
70.0	0.0999	-0.0132	0.0017	0.6123
80.0	0.0264	-0.0035	0.0005	0.2201
90.0	0.0066	-0.0010	0.0001	0.0678

Table 5. Numerical results for American option with forcing term



About the Author

Dr. Richard Hanson, Ph.D. is a Senior Scientist at Rogue Wave Software. His major project responsibilities include technical supervision, project management, and involvement in the full cycle of software development. Early IMSL Library projects included notable improvements in the IMSL real, symmetric, dense eigenvalue-eigenvector code, from which (given the right environment) an order of magnitude speed-up is obtained compared with LAPACK.

For the past twenty years, Dr. Hanson has also served on the IFIP WG 2.5 Committee on Mathematical Software. He was appointed for significant contributions to the publication of mathematics software and continues to participate in projects, which monitor and improve mathematical software tools, languages and working environments.

Some of his most recent significant work has been to design and implement MPI-enhanced subroutines and functions for scientific applications on distributed network computers. The IMSL Distributed Network FORTRAN Library (DNFL) was developed on arrays of workstations, and is currently available for the IBM SP-2, Hitachi SR2201, and Cray T3E.

References

1. Wilmott, Paul, Howison, Sam, DeWynne, Jeff. *The Mathematics of Financial Derivatives*, Cambridge Univ. Press, (1995).
2. Strang, G., Fix, G. J. *An Analysis of the Finite Element Method*, Prentice-Hall, Inc., Englewood Cliffs, NJ, (1973). Also see The Second Edition, Wellesley-Cambridge Press, Wellesley, MA, (2008).
3. Hildebrand, F. B., *Introduction to Numerical Analysis*, Second Ed., Dover, Publications, Inc. New York, NY, (1987).
4. Brenan, K. E., Campbell, S. L., Petzold, L. R., *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, SIAM Publications, Classics in Applied Mathematics, Philadelphia, PA, (1996).
5. http://mathalacarte.com/c/math77_head.html Software Library, Math77, Chapter 14.3.
6. Hull, John C. *Options, Futures and Other Derivatives*, Prentice-Hall, India, Ed. 5.
7. Topper, Jürgen, *Financial Engineering with Finite Elements*, John Wiley & Son, (2005).
8. Nielson, B. F., Skavhaug, O., Tveito, "A. Penalty and Front-Fixing Methods for the Numerical Solution of American Option Problems", *J. Comp. Finance*, **5**(4), (2002).
9. Longstaff, Francis A., Schwartz, Eduardo S., "Valuing American Options by Simulation: A Simple Least-Squares Approach", UCLA Working Paper <http://repositories.cdlib.org/anderson/fin/1-01>
10. Hanson, R. J., Krogh, F. T., "Solving Constrained Differential-Algebraic Systems Using Projections", Rogue Wave White Paper (2008), [Solving Constrained Differential-Algebraic Systems Using Projections](#).
11. M. J. Tomas III, Kishore K. Yalamanchili, *The Journal of Futures Markets*, **21**, No. 1, 19–42, (2001).
12. [IMSL C Numerical Library Version 7.0](#) (see Chapter 5: Differential Equations).

