Xi Chen, John Burkardt and Max Gunzburger]

Abstract

The Heston stochastic volatility model extends the Black-Scholes-Merton model by allowing the volatility to vary stochastically. Exact solutions are derivable for special cases but, in general, approximate solutions obtained through numerical simulations are needed. Previous approaches have concentrated on finite difference methods (FDMs). The current study demonstrates how the Heston model can be simulated using finite element methods (FEMs). This approach can more efficiently reproduce the results obtained using FDMs, but can also be applied to far more general cases in which the boundary conditions, domain shape, or limited regularity of the data limit or forbid the use of FDMs. FEMs for the Heston model are discussed and the results of computational experiments are provided that demonstrate their accuracy and efficiency. Then, FEMs are applied to case studies for European vanilla option pricing.

High Accuracy Finite Element Methods for Option Pricing under Heston's Stochastic Volatility Model

[

February 11, 2024

1 Introduction

The problem of option pricing is central to modern financial theory and practice. In 1993, Heston [?] produced an extension of the Black-Scholes-Merton model in which the volatility was modeled by a stochastic process rather than being treated as a constant. The Heston model provides a more accurate evaluation of financial derivatives. Also, for European vanilla options, it allows for the derivation of a closed-form exact solution whereas other stochastic volatility models can only be treated numerically.

Closed-form solutions are rarely obtainable, and accurate estimates from Monte Carlo simulations can often be expensive to compute. As a result, most option value calculations are done by a numerical approximation of the solution of the partial differential equation (PDE) system. The most commonly used solution approaches are finite difference methods (FDMs); these methods have some severe limitations such as requiring sufficiently smooth terminal and boundary conditions, a rectilinear domain, and logically rectangular (i.e., Cartesian) grids. Moreover, the solutions produced are available only at the grid points; at other points, one has to rely on interpolation of the grid values. By contrast, finite element methods (FEMs) can handle a wider variety of boundary and terminal conditions; curved, irregular, or hollow domains; and fairly arbitrary grids that conform to the geometry. Moreover, approximate solutions obtained are piecewise-smooth functions defined over all points in the solution domain.

A comprehensive overview and demonstration of how the finite element method may be used in solving option pricing problems can be found in [?], including specific information on applications to quantitative finance. To date, most studies of the Heston model have focused on FDMs; see [?, ?, ?, ?, ?]. One of the first papers employing finite element methods was Winkler et al. [?], which carried out a valuation of European vanilla options and showed that the problem was well-posed; however, the paper did not address convergence

behavior, in particular the expected convergence rates of the approximate solution. The thesis of Xiong [**?**], which mostly uses a single $150 \times 150 \times 150$ grid for temporal and spatial resolutions by linear FEM basis functions, similarly does not investigate the convergence behavior of the solution to the Heston model. In Schwab et al., [**?**], the convergence rate of Heston model is considered, but the computation of convergence rates is again not carried out in accordance with PDE theory.

The goal of this paper is to apply the finite element method to the Heston model and show that good approximate solutions can be determined efficiently, that the error in an initial solution estimate can be driven down rapidly, that the expected rate of error decay can easily be determined, and that comparison to the observed rate of error decay is an important guarantee that the method has been implemented correctly and that suitable spatial and temporal resolutions have been chosen. Several variations of the Heston model will be considered, and optimal convergence will be expected. It is hoped that this discussion will demonstrate some of the advantages that the finite element method enjoys as an alternative to finite difference methods. These advantages should recommend wider familiarity and use of the FEM for option pricing and similar problems in financial theory that have an underlying PDE formulation.

In Section **??**, the Heston stochastic volatility model is introduced and the corresponding PDE is derived along with appropriate terminal and boundary conditions. FDMs for the Heston model are then briefly discussed in Section **??**. Then, in Section **??**, the mathematical framework associated with FEMs is presented; the Heston model is recast in a form suitable for discretization by FEMs, and the notion of approximating spaces is discussed; this allows us to seek an approximate solution that is a continuous piecewise-linear or quadratic polynomial. Computational issues such as numerical integration, linear system solving, and convergence detection are also considered. In Section **??**, we use several numerical experiments to demonstrate the flexibility and effectiveness of FEMs for the Heston stochastic volatility model, focusing on obtaining accurate approximate solutions for European vanilla option pricing.

2   The Heston model

2.1   Motivation

In the forty years since its publication in 1973 [**?**], the Black-Scholes-Merton model has become the most widely used mathematical model for option pricing problems. Not only does it give an analytical solution for the price of vanilla options, i.e., the Black-Scholes formula, but it also serves as a robust foundation for more refined or extensional models. However, there are a number of ways in which the Black-Scholes model has been shown to disagree with observed reality. For instance, the most questionable assumption of the model is that continuously compounded stock returns are normally distributed with constant volatility. To the contrary, many empirical studies and economic arguments,

especially after the stock market crash of 1987, have shown that equity return distributions exhibit skewness and kurtosis and are always negatively correlated with implied volatility, facts which conflict with the normality assumption made in the Black-Scholes-Merton model.

In order to replace this normality assumption by a time-varying volatility, many researchers specify a stochastic process that drives the volatility, in what are known as stochastic volatility models; see [?, ?]. The Heston model falls into this class, and is widely regarded as a benchmark against which other such models are commonly judged. The Heston stochastic model is formally defined as the system of stochastic differential equations given by

$$
\begin{cases}
\mathrm{d}S(t) = S(t)\Big[(r-q)\mathrm{d}t + \sqrt{v(t)}\mathrm{d}W_1(t)\Big] \\
\mathrm{d}v(t) = \kappa(\theta - v(t))\mathrm{d}t + \xi\sqrt{v(t)}\mathrm{d}W_2(t) \\
\mathrm{d}W_1(t)\mathrm{d}W_2(t) = \rho\mathrm{d}t,
\end{cases}
\tag{1}
$$

where $S(t)$ denotes the spot process at time t, $v(t)$ the volatility, r the risk free interest rate, q the continuous dividend rate, κ the mean reversion speed for the variance, θ the mean reversion level for the variance, ξ the volatility of the variance, and $W_i(t)$, $i = 1, 2$, two Brownian motions with correlation ρ. The model for the volatility $v(t)$ is the same as the one used by Cox, et al. [?] for short-term interest rates.

The parameters ρ, ξ, and κ, which are included in the Heston model, provide the ability to capture observed features of the market and to produce a wide range of distributions. For instance, the parameter ρ, the correlation between the log-returns and the asset volatility, affects the skewness of the distribution and hence the shape of the implied volatility surface; the parameter ξ, the volatility of the variance, affects the kurtosis of the distribution; the mean reversion parameter κ can be interpreted as representing the degree of volatility clustering. This phenomenon has been observed repeatedly in the market; the occurrence of large price variations makes it more likely that further large price variations will follow. Additional details about the effect of these parameters are considered in [?].

Additionally, the model provides a closed-form (exact solution) for European options, making it more tractable and easier to implement than other stochastic volatility models. For these reasons, the Heston stochastic model is regarded as a more robust and flexible alternative to the Black-Scholes-Merton model, providing a more realistic framework for option pricing.

2.2 The partial differential equation form of the Heston model

Let $g(t, v, S)$ denotes the price of an option at time t, with volatility v and spot process S; then $g(t, v, S) := e^{-r(T-t)}\mathrm{E}[h(V(T), S(T))]$, where $h(V(T), S(T))$ is the payoff of the option at time T. By the Feynman-Kac Theorem, we than

have that the function $g(t, v, S)$ satisfies the PDE

$$g_t + \frac{1}{2}\xi^2 v g_{vv} + \rho\xi S v g_{Sv} + \frac{1}{2}S^2 v g_{SS} + \kappa(\theta - v)g_v + (r - q)S g_S - rg = 0 \quad (2)$$

along with the condition $g(T, v, S) = h(v, S)$ imposed at the final time T; the values $g(0, v, S)$ at the initial time are unknown and need to be determined. Equation (??) can be simplified by the changes of variable $y = \log(S/K)$ and $\tau = T - t$, resulting in

$$U_\tau - \frac{1}{2}\xi^2 v U_{vv} - \rho\xi v U_{yv} - \frac{1}{2}v U_{yy} - \kappa(\theta - v)U_v - (r - q - \frac{1}{2}v)U_y + rU = 0 \quad (3)$$

along with the initial condition $U(0, v, y) = h(v, Ke^y)$ imposed at $\tau = 0$, where $U(\tau, v, y) = g(t, v, S)$. Note that the change of variables replaces the calendar time t by the time-to-maturity $\tau = T - t$ so that resulting problem (??) has the more familiar form of an initial-value problem with the solutions to be determined at $\tau = T$.

To solve (??), boundary conditions need to be specified. Let $\Omega = \{(v, y) : v \in (v_{min}, v_{max}), \ y \in (y_{min}, y_{max})\}$ with $y_{min} = \log(S_{min}/K)$ and $y_{max} = \log(S_{max}/K))$ denote the computational domain with the boundaries

$$\begin{cases} \Gamma_1 = \left\{(v, y) : v = v_{min}, \ y \in (y_{min}, y_{max}))\right\} \\[2mm] \Gamma_2 = \left\{(v, y) : y = y_{max}, \ v \in (v_{min}, v_{max})\right\} \\[2mm] \Gamma_3 = \left\{(v, y) : v = v_{max}, \ y \in (y_{min}, y_{max})\right\} \\[2mm] \Gamma_4 = \left\{(v, y) : y = y_{min}, \ v \in (v_{min}, v_{max})\right\}. \end{cases}$$

Then the Dirichlet boundary condition and initial condition for a European vanilla option (call: $\eta = 1$; put: $\eta = -1$) are defined as

$$\begin{cases} U(t, v_{min}, y) = \left[\eta(Ke^y e^{-q\tau} - Ke^{-r\tau})\right]^+ \\[2mm] U(t, v, y_{max}) = \frac{1 + \eta}{2}\left[\eta(Ke^{y_{max}}e^{-q\tau} - Ke^{-r\tau})\right]^+ \\[2mm] U(t, v_{max}, y) = \frac{1 + \eta}{2}Ke^y e^{-q\tau} + \frac{1 - \eta}{2}Ke^{-r\tau} \quad\quad (4) \\[2mm] U(t, v, y_{min}) = \frac{1 - \eta}{2}\left[\eta(Ke^{y_{min}}e^{-q\tau} - Ke^{-r\tau})\right]^+ \\[2mm] U(0, v, y) = \left[\eta(Ke^y - K)\right]^+. \end{cases}$$

In order to describe the finite element method, it is useful to highlight the differential operators involved so that we rewrite (??) in terms of the gradient and divergence operators as

$$U_\tau - \nabla \cdot A\nabla U + b \cdot \nabla U + rU = 0, \quad\quad (5)$$

5

where

$$A = \begin{bmatrix} \dfrac{1}{2}v\xi^2 & \alpha v\rho\xi \\ (1-\alpha)v\rho\xi & \dfrac{1}{2}v \end{bmatrix} \qquad \text{and} \qquad b = \begin{bmatrix} -\kappa(\theta - v) + \dfrac{1}{2}\xi^2 \\ -(r-q) + \dfrac{1}{2}v + \alpha\rho\xi \end{bmatrix} \qquad (6)$$

for $\alpha \in [0,1]$. Note that in most of the literature, e.g., [?], these matrices are written as

$$A = \begin{bmatrix} \dfrac{1}{2}v\xi^2 & \dfrac{1}{2}v\rho\xi \\ \dfrac{1}{2}v\rho\xi & \dfrac{1}{2}v \end{bmatrix} \qquad \text{and} \qquad b = \begin{bmatrix} -\kappa(\theta - v) + \dfrac{1}{2}\xi^2 \\ -(r-q) + \dfrac{1}{2}v + \dfrac{1}{2}\rho\xi \end{bmatrix} \qquad (7)$$

which corresponds to the specific choice $\alpha = \frac{1}{2}$. However, our numerical experiments given in Section ?? show that α can be any number in the interval $[0,1]$ so that, in order to simplify the computational work, it may be beneficial to use $\alpha = 1$ or $\alpha = 0$.

3   Finite difference methods

In general, an exact solution of the Heston model cannot be obtained. One has to be content with obtaining an approximate solution through a discretization process. Previous efforts [?, ?, ?, ?, ?] in this direction use finite difference methods (FDMs) for this purpose. Here, we briefly discuss such an approach so that it can be contrasted with finite element methods that are the main concern of this paper.

To define a finite difference method for the Heston model (??) and (??), we begin by choosing two positive integers I and J and then defining the Carteisan grid $v_i = v_{min} + i\Delta v$ for $i = 0, \ldots, I$ with $\Delta v = (v_{max} - v_{min})/I$ and $y_j = y_{min} + j\Delta y$ for $j = 0, \ldots, J$ with $\Delta y = (y_{max} - y_{min})/J$. We also define a temporal grid by choosing a positive integer N and then letting $\Delta\tau = T/N$ and setting $\tau_n = n\Delta\tau$ for $n = 0, \ldots, N$. By replacing the derivative in (??) by difference quotients, we then define the discretized system for the approximation $u_{i,j}^n \approx U(v_i, y_j, \tau_n)$ given by

$$\frac{u_{i,j}^{n+1} - u_{i,j}^n}{\Delta\tau} - \theta L_h u_{i,j}^{n+1} - (1-\theta)L_h u_{i,j}^n = 0 \qquad (8)$$

for $i = 1, \ldots, I-1$, $j = 1, \ldots, J-1$, and $n = 1, \ldots, N$, where

$$L_h u_{i,j}^n = \frac{1}{2}\xi^2 v_i \frac{u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n}{(\Delta v)^2} + \rho\xi v_i \frac{u_{i+1,j+1}^n + u_{i-1,j-1}^n - u_{i-1,j+1}^n - u_{i+1,j-1}^n}{4\Delta v\Delta y}$$

$$+ \frac{1}{2}v_i \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{(\Delta y)^2} + \kappa(\theta - v_i)\frac{u_{i+1,j}^n - u_{i-1,j}^n}{2\Delta v}$$

$$+ (r - q - \frac{1}{2}v_i)\frac{u_{i,j+1}^n - u_{i,j-1}^n}{2\Delta y} - ru_{i,j}.$$

$$(9)$$

The values of $u_{0,j}^n$, $u_{I,j}^n$, $u_{i,0}^n$, $u_{i,J}^n$, and $u_{i,j}^0$ needed to close the system (**??**) are obtained from (**??**).

The parameter θ allows us to consider three common time-stepping schemes at once. Setting $\theta = 0$ results in the explicit forward-Euler FDM scheme which is conditionally stable, with a first-order convergence rate in the time step $\Delta\tau$. The value $\theta = 1$ defines the implicit backward-Euler scheme which is unconditionally stable, also with a first-order convergence rate in $\Delta\tau$. The intermediate value $\theta = 0.5$ yields the implicit Crank-Nicolson method; it shares the unconditional stability of the backward-Euler method, but has a second order convergence rate in $\Delta\tau$, and hence is the most attractive of the three options. All three options have second-order convergence rates in Δv and Δy.

The FDM method exhibited here is highly popular because it is relatively simple to implement. However, the accuracy of the method is limited by the fineness of the mesh, both in time and space; if greater accuracy is desired, the temporal and spatial meshes must be refined. When an implicit time integration approach is used, the corresponding linear system can grow rapidly in size. Those committed to using an FDM-style approach have therefore explored variations involved higher-order compact FDM methods [**?**] for the spatial derivatives, or ADI methods [**?**] for the linear system associated with the time integration. We instead consider finite element methods as an alternative to FDMs.

4 Finite element methods

Finite element methods (FEMs) are an alternative to the FDMs discussed in the previous section; FEMs are widely used, at least outside the financial community, for discretizing systems of partial differential equations. Their popularity arises in part from the fact that they can offer more and better information than FDMs for a given problem and can handle many problems for which an FDM approach would be difficult or impossible. Before describing the implementation details of an FEM approach, it is worth discussing, in greater detail, the advantages FEMs afford.

For the finite element approach, although input data is only specified at finitely many locations, the solution is returned as a (piecewise) smooth function defined over the entire domain, so that it can be evaluated, integrated, or contoured. By contrast, FDMs return the solution only at a discrete set of grid points so that to obtain values at other points, interpolation is required.

PDEs most often involve boundary and/or initial conditions. Whereas FDM can easily handle Dirichlet conditions for which the value of the solution is specified, they can be awkward in handling conditions involving a derivative, often referred to as Neumann or Robin conditions. However, this type of boundary condition is common when estimating the behavior of an option as the underlying price goes to infinity. FEMs incorporate such boundary conditions in a natural and accurate fashion.

The shape of the computational domain can also become a problem for

FDMs. Instead of the rectangular domain treated in Section **??**, irregular domains can arise, e.g., when knock-out barriers are imposed on a multiple-asset option, or in the pricing of convertible bonds, or when the PDE only needs to be solved over a portion of the domain because some parts can be determined by financial reasoning. FDMs rely on the use of a regular Cartesian grid of sample points, as was done in Section **??**; FEMs can handle irregular regions by decomposing them into the sum of simple shapes such as triangles. Moreover, FEMs make possible local refinement of the mesh. If areas of the computational domain are known to represent regions of rapid change of the solution, the grid can automatically detect this and there exist standard methods for reducing the mesh size there. This can be useful when computations are being made near the strike price or close to the barrier.

If the computational domain is semi-infinite, FEMs can be implemented using infinite elements or boundary elements, resulting in the correct treatment of the domain. This is common practice in engineering, whereas, in finance, FDMs are used, requiring the use of artificial boundary conditions applied at some large but finite value.

Whereas most papers limit their concern to pricing, the majority of practitioners are also very interested in measures of sensitivity of those prices, i.e., the degree to which the computed answers change if some input quantity is varied. Some of these measures of sensitivity, commonly called Greeks, can be obtained more easily and accurately using FEMs.

The greater flexibility and power of FEMs are associated with a more complicated computational procedure. We now consider the mathematical background and computational issues associated with the method.

4.1 Weak formulation and finite element discretization

Let Ω denote the computational domain and let Γ denote its boundary. Let $S(\Omega) \subset L^2(\Omega)$ denote the space of functions having first derivatives that also belong to $L^2(\Omega)$; here $L^2(\cdot)$ denotes the space of square integrable functions. Given $f \in L^2(\Gamma)$, define the affine space $S_f(\Omega) = \{U \in S(\Omega) \mid U = f \text{ on } \Gamma\}$ and the subspace $S_0(\Omega) = \{\phi \in S(\Omega) \mid \phi = 0 \text{ on } \Gamma\}$. We then pose the problem:

$$
\begin{aligned}
& \textit{given } f \in L^2(\Gamma), \textit{ seek } U \in S_f(\Omega) \textit{ satisfying} \\
& \int_\Omega U_\tau \cdot \phi \mathrm{d}\Omega + \mathcal{B}(U, \phi) = 0 \quad \textit{for all } \phi \in S_0(\Omega),
\end{aligned}
\tag{10}
$$

where

$$
\mathcal{B}(U, \phi) = \int_\Omega \Big(\nabla\phi \cdot (A\nabla U) + \phi(b \cdot \nabla U + rU) \Big) \mathrm{d}\Omega.
$$

Equation (**??**) is referred to as a weak formulation of (**??**) and is obtained by multiplying the latter by a general function $\phi \in S_0(\Omega)$, then integrating the result over Ω, and then integrating the term involving the matrix A by parts, taking notice that $\phi = 0$ on the boundary of Ω. The well-posedness of the equivalent version of weak formulation (**??**) has been proved in [**?**].

8

To effect discretization, let $S^h \subset S(\Omega)$ denote a family of finite-dimensional subspaces parameterized by a parameter h tending to zero; in our setting, h is a measure of the grid size. Let $\mathbf{x} = (v \ , \ y)^T$. Then for $\mathbf{x} \in \Gamma$, let $f^h(\mathbf{x})$ denote an approximation of $f(\mathbf{x})$; if $f(\mathbf{x})$ is continuous, we can choose $f^h(\mathbf{x})$ to be the interpolant of $f(\mathbf{x})$ in $S^h|_\Gamma$; otherwise, we can choose $f^h(\mathbf{x}) \in S^h|_\Gamma$ to be the $L^2(\Gamma)$ projection of $f(\mathbf{x})$. We then define the affine space $S^h_f = \{U^h \in S^h \mid U^h = f^h \text{on } \Gamma\}$ and the subspace $S^h_0 = \{\phi^h \in S^h \mid \phi^h = 0 \text{ on } \overline{\Omega}\} \subset S_0(\Omega)$. Then, the semi-discrete spatial discretization of (**??**) is defined by

$$
given\ f^h \in S^h|_\Gamma,\ seek\ U^h \in S^h_f\ satisfying
$$

$$
\int_\Omega U^h_\tau \cdot \phi^h \mathrm{d}\Omega + \mathcal{B}(U^h, \phi^h) = 0 \quad for\ all\ \phi^h \in S^h_0. \tag{11}
$$

Using the same time discretization scheme as used in Section **??**, we arrive at the fully discrete system: for $n = 0, 1, \ldots, N$,

$$
\int_\Omega \frac{U^{n+1}_h - U^n_h}{\Delta \tau} \phi_h \mathrm{d}\Omega + \theta \mathcal{B}(U^{n+1}_h, \phi^h) + (1-\theta)\mathcal{B}(U^n_h, \phi^h) = 0 \quad for\ all\ \phi^h \in S^h_0. \tag{12}
$$

The connection between θ and the three types of time-stepping schemes is the same as that in Section **??**. In Section **??**, we report the results of numerical experiments using all three of the choices for θ.

4.1.1 Continuous piecewise-linear finite element spaces

An important step in the finite element method involves choosing the finite dimensional space S^h within which the test functions ϕ^h and solution U^n_h are sought. The most common choice S^h is the space of continuous piecewise-linear polynomials defined with respect to a partition of the domain Ω into triangles. In this case, there will be three degrees of freedom associated with each triangle, with a basis function associated with each of the vertices. A example partition is shown in Figure **??**, where the vertices and elements have been numbered; the grid size parameter can be chosen to be the length of the side of the triangles. It is known that if S^h is chosen in this manner, the L^2-norm of the error in the finite element solution is of order h^2 whereas the L^2-norm of the error in the derivatives of that solution is of order h; this latter norm is often referred to as the H^1 semi-norm. The numerical experiments described in Section **??** illustrate these results.

4.1.2 Continuous piecewise-quadratic finite element spaces

More accurate approximate solutions can be constructed by choosing higher degree polynomials, e.g., by letting S^h be the space of continuous piecewise-quadratic polynomials. These can be defined with respect to the same type of partition of the domain Ω as used in Section **??**. However, now one locates grid points not only at the three vertices of the triangles, but also the three mid-side points. The additional mesh points are illustrated in Figure **??** which should
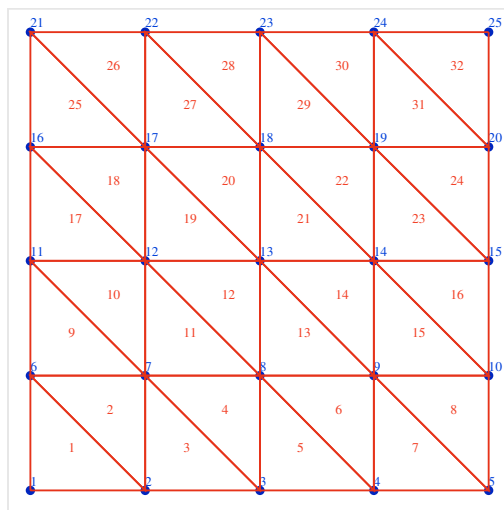
9

Figure 1: An example of two-dimensional grid and the grid numbering for continuous piecewise-linear finite elements.

be compared to Figure **??**. It is known that for this choice of finite element space, the L^2 error is now of order h^3 and the H^1-norm error is of order h^2. Thus, at the cost of a greater number of degrees of freedom, i.e., grid points, this choice produces approximate solutions having greater accuracy, compared to using piecewise-linear finite elements.

4.2 Implementation issues

We now consider some issues that arise in the implementation and testing of finite element methods.

4.2.1 Numerical integration

The integrals appearing in (**??**) are usually approximated using a quadrature rule that must be chosen to be of sufficient accuracy so that the accuracy of the finite element approximation is not compromised while also not causing an excessive computational burden. Every integral is decomposed into a sum of integrals over the elements, e.g., over the triangles, and each of the latter is approximated using a quadrature rule which takes the form of a sum of products of constant weights times the integrand evaluated at several points in the triangle. The number of evaluations may be used as measure of the cost of the rule so that one wants to choose a rule that uses as few points as possible. Depending on the finite element space chosen, it is usually appropriate to use a 3 point rule for the piecewise-linear case and a 7 point rule for the piecewise
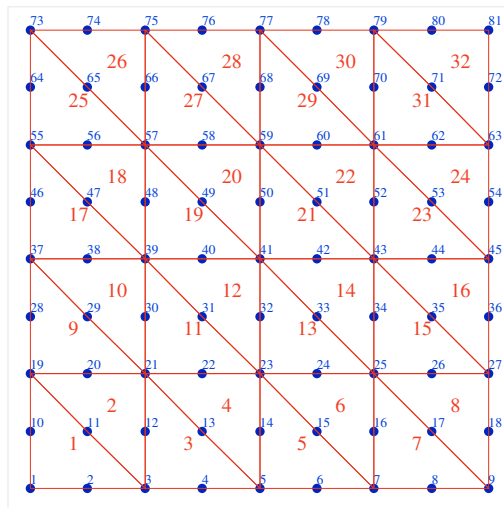
Figure 2: An example of two-dimensional grid and the grid numbering for continuous piecewise-quadratic finite elements.

linear case for the piecewise-quadratic case. A very accurate 13 point rule is always used to estimate the error norms.

4.2.2 Linear System Solver

After all the integrals have been approximated, we are left with, at every time step, a large linear algebraic system of the form $\mathbb{B}\vec{U}^n = \vec{F}^n$, where the components of $\vec{U}$ are the values of the approximate solution at the grid points; note that the coefficient matrix $\mathbb{B}$ does remains the same at all time levels. Then, we have that the approximation to the solution at time τ_n is given by $U_h^n(\mathbf{x}) = \sum (\vec{U}^n)_k \phi_k(\mathbf{x})$. In general, the matrix $\mathbb{B}$ is banded so that a compact storage and solution scheme can be applied. In our computational experiments, we use the subroutines with DGB prefix from the standard LAPACK linear algebra package [?]. Larger problems than the ones we consider here can be handled efficiently using sparse matrix storage or iterative techniques.

4.2.3 Estimates for the convergence rates of FEM approximations

FEMs, of course, produce an approximation of the solutions of (??), or equivalently (??). The analysis of FEMs allows one to estimate how the error in the approximation behaves as a function of the grid sizes h and $\Delta\tau$, even before one implements the method into a computer code. Knowing what is the expected error is a valuable tool in verifying the correctness of the code and also allows one to choose grid sizes such that the error will be smaller than a prescribed tolerance.

From the general theory of FEM for PDE, see[**?**], we may expect to have the following convergence results for both the forward Euler scheme ($\theta = 0$ in (**??**)) and the backward Euler scheme ($\theta = 1$ in (**??**)):

$$\|u(\tau_n) - U_h^n\|_0 = O(h^{k+1} + \Delta\tau). \tag{13}$$

Also for the Crank-Nicolson scheme ($\theta = 0.5$ in (**??**)), the expected result is

$$\|u(\tau_n) - U_h^n\|_0 = O(h^{k+1} + \Delta\tau^2), \tag{14}$$

where k denotes the degree of the piecewise polynomial used in the FEM. Table **??** provides convergence rates, as predicted by analyses, with respect to the spatial gird size h and temporal grid size $\Delta\tau$ for linear and quadratic FEMs.

Table 1: Summary of the general convergence rate of FEM schemes with respect to the spatial grid size h and temporal step size $\Delta\tau$.

		$\|u - u^h\|_{L^2}$	$\|u - u^h\|_{H^1}$
spatial	linear FEM	$O(h^2)$	$O(h)$
error	quadratic FEM	$O(h^3)$	$O(h^2)$
temporal	forward Euler	$O(\Delta\tau)$	$O(\Delta\tau)$
error	backward Euler	$O(\Delta\tau)$	$O(\Delta\tau)$
	Crank-Nicolson	$O(\Delta\tau^2)$	$O(\Delta\tau^2)$

5 Computational experiments

Unless otherwise noted, the parameters use for the numerical experiments are given in Table **??**.

5.1 Manufactured solution and code verification

Before applying our methodology for pricing an option, we first test the methodology and the code that implements it in order to verify that we obtain the convergence rates predicted by the analysis; see Table **??**. To this end, we use the method of manufactured solutions to define a problem for which the exact solution is known. Specifically, we use the solution

$$\widehat{U}(v, y, \tau) = \cos(\pi v)\cos(\pi y)e^{-\tau}. \tag{15}$$

This exact solution is smooth and satisfies inhomogeneous Dirichlet boundary conditions. Note that this type of boundary condition is applied below when we apply our methodology to option pricing. Also, note that to accommodate this exact solution, we must add a right-hand side forcing function to (**??**) or,

Table 2: Default model parameters used in the computational experiments.

Initial spot price	$S_0=100.0$
Initial variance	$v_0=0.25$
Mean reversion rate	$\kappa=1.0$
Mean variance	$\theta = 0.09$
Risk free interest rate	$r = 0.05$
Dividend yield rate	$q = 0.01$
Volatility of variance	$\xi = 0.4$
Correlation	$\rho = -0.7$
Time to maturity	$T = 1.0$
Call strike	$K_c = 110.0$
Put strike	$K_p = 90.0$

equivalently, to (??); this function is determined by substituting $\widehat{U}(v, y, \tau)$ in to the left-hand side.

To simplify the problem, we choose the spatial domain $[0, 1] \times [0, 1]$ and temporal interval $[0, 0.25]$, with spatial resolution $h = \Delta v = \Delta y$ and temporal step size $\Delta \tau$. We present results of computational experiments for the model problem having exact solution $\widehat{U}(v, y, \tau)$ for each of the three time discretization schemes discussed in Section ?? and for both the linear and quadratic FEMs. In the tables and figures, we provide the $L^2(\Omega)$ norms and $H^1(\Omega)$ semi-norms of the error and the corresponding rates of convergence for a sequence of grid sizes.

We begin with the *continuous piecewise-linear finite element* discretization discussed in Section ??. Specifically, for the smooth exact solution $\widehat{U}(v, y, \tau)$, we provide, respectively, in Tables ??, ??, ??, and ??, results for the forward Euler (FE) (unstable and stable cases), backward Euler (BE), and Crank-Nicolson (CN) time discretization schemes. From these tables, we make the following observations. For smooth solutions such as $\widehat{U}(v, y, \tau)$, using continuous piecewise-linear finite element approximations of the model (??) or equivalently (??):

- The forward Euler time discretization scheme converges at the optimal rates, i.e., the L^2 norm of errors are roughly of $O(\Delta \tau + h^2)$ and the H^1 semi-norm of errors are roughly of $O(h)$. However, the scheme is only conditionally stable, i.e., the L^2 norm of errors and the H^1 semi-norm of the errors may blow up if the chosen $\Delta \tau$ is not sufficiently small.

- The backward Euler time discretization scheme is unconditionally stable and converges at the optimal rates, i.e., the L^2 norm of errors are roughly of $O(\Delta \tau + h^2)$ and the H^1 semi-norm of errors are roughly of $O(h)$.

- The Crank-Nicolson time discretization scheme is unconditionally stable and converges at the optimal rates, i.e., the L^2 norm of errors are roughly of $O(\Delta \tau^2 + h^2)$ and the H^1 semi-norm of errors are roughly of $O(h)$.

Table 3: Errors and convergence rates of continuous piecewise-linear approximations and the forward Euler scheme with $(2\Delta\tau)^{1/2} = h$ for the smooth exact solution $\widehat{U}(v, y, \tau)$. This choice for $\Delta\tau$ leads to an unstable approximation.

h	$||U - U^h||_{L^2}$		$|U - U^h|_{H^1}$	
	error	rate	error	rate
2^{-2}	5.0409E-002	–	6.6377E-001	–
2^{-3}	1.3430E-002	–	3.3792E-001	–
2^{-4}	6.3602E+014	–	3.8775E+016	–
2^{-5}	1.0769E+108	–	1.3574E+110	–

Table 4: Errors and convergence rates of continuous piecewise-linear approximations and the forward Euler scheme with $(3\Delta\tau)^{1/2} = h$ for the smooth exact solution $\widehat{U}(v, y, \tau)$. This choice for $\Delta\tau$ leads to a stable approximation.

h	$||U - U^h||_{L^2}$		$|U - U^h|_{H^1}$	
	error	rate	error	rate
2^{-2}	5.0267E-002	–	6.6349E-001	–
2^{-3}	1.3392E-002	1.9082	3.3788E-001	0.9736
2^{-4}	3.4164E-003	1.9708	1.6967E-001	0.9938
2^{-5}	8.6030E-004	1.9896	8.4911E-002	0.9987

Table 5: Errors and convergence rates of continuous piecewise-linear approximations and the backward Euler scheme with $\Delta\tau^{1/2} = h$ for the smooth exact solution $\widehat{U}(v, y, \tau)$.

h	$||U - U^h||_{L^2}$		$|U - U^h|_{H^1}$	
	error	rate	error	rate
2^{-2}	4.9320E-002	–	6.6179E-001	–
2^{-3}	1.3105E-002	1.9121	3.3769E-001	0.9707
2^{-4}	3.3379E-003	1.9731	1.6963E-001	0.9933
2^{-5}	8.3961E-004	1.9911	8.4905E-002	0.9985

Next, we repeat the experiments, but now use the *continuous piecewise-quadratic finite element* discretization discussed in Section **??**; the results are given in Tables **??**, **??**, and **??**; note that we omit results for unstable choices of the time step for the forward Euler method. For all three time stepping

14

Table 6: Errors and convergence rates of continuous piecewise-linear approximations and the Crank-Nicolson scheme with $\Delta\tau = h$ for the smooth exact solution $\widehat{U}(v, y, \tau)$.

h	$\lVert U - U^h \rVert_{L^2}$		$\lvert U - U^h \rvert_{H^1}$	
	error	rate	error	rate
2^{-2}	5.0844E-002	–	6.6563E-001	–
2^{-3}	1.3384E-002	1.9256	3.3788E-001	0.9782
2^{-4}	3.4083E-003	1.9734	1.6966E-001	0.9939
2^{-5}	8.5779E-004	1.9904	8.4910E-002	0.9986

schemes, we observe that the L^2 norm and H^1 semi-norm of the errors converge at roughly the optimal rates, namely:

- Forward Euler: $O(\Delta\tau + h^3)$ and $O(h^2)$, if $\Delta\tau$ is small enough;
- Backward Euler: $O(\Delta\tau + h^3)$ and $O(h^2)$;
- Crank-Nicolson: $O(\Delta\tau^2 + h^3)$ and $O(h^2)$.

Table 7: Errors and convergence rates of continuous piecewise-quadratic approximations and the forward Euler scheme with $(3\Delta\tau)^{1/3} = h$ for the smooth exact solution $\widehat{U}(v, y, \tau)$.

h	$\lVert U - U^h \rVert_{L^2}$		$\lvert U - U^h \rvert_{H^1}$	
	error	rate	error	rate
2^{-1}	2.8554E-002	–	3.9581E-001	–
2^{-2}	3.6179E-003	2.9805	1.0636E-001	1.8959
2^{-3}	4.4290E-004	3.0300	2.6699E-002	1.9941
2^{-4}	5.4576E-005	3.0206	6.6293E-003	2.0099

5.2 European vanilla option pricing

In this section, we present the results of applying the finite element method to European option pricing. For this purpose, we use the higher-order convergence piecewise-quadratic finite element basis functions for spatial discretization and the Crank-Nicolson time discretization scheme so that we expect good convergence behavior. With the faster convergence of this scheme, we can choose the spatial computational domain to be $v \in [0.0, 4.0]$ and $S \in [Ke^{-2}, Ke^2]$; then, we may have the same spatial resolutions N_x for both the v and $y = \log(S/K)$ directions. Also, for the temporal resolution, we will set $N_T = \lceil T \times (N_x/4)^{1.5} \rceil$

Table 8: Errors and convergence rates of continuous piecewise-quadratic approximations and the backward Euler scheme with $\Delta\tau^{1/3} = h$ for the smooth exact solution $\widehat{U}(v, y, \tau)$.

h	$\|\|U - U^h\|\|_{L^2}$		$|U - U^h|_{H^1}$	
	error	rate	error	rate
2^{-1}	2.6392E-002	–	3.8679E-001	–
2^{-2}	3.5225E-003	2.9054	1.0622E-001	1.8645
2^{-3}	4.3757E-004	3.0090	2.6709E-002	1.9917
2^{-4}	5.4381E-005	3.0083	6.6308E-003	2.0101

Table 9: Errors and convergence rates of continuous piecewise-quadratic approximations and the Crank-Nicolson scheme with $\Delta\tau^{2/3} = h$ for the smooth exact solution $\widehat{U}(v, y, \tau)$.

h	$\|\|U - U^h\|\|_{L^2}$		$|U - U^h|_{H^1}$	
	error	rate	error	rate
2^{-1}	2.9630E-002	–	4.1040E-001	–
2^{-2}	3.5908E-003	3.0447	1.0546E-001	1.9603
2^{-3}	4.4203E-004	3.0221	2.6687E-002	1.9825
2^{-4}	5.4500E-005	3.0198	6.6292E-003	2.0092
2^{-5}	6.7711E-006	3.0088	1.6509E-003	2.0056

to match the spatial resolutions predicted by the FEM convergence theory introduced in (??).

5.2.1 Exact vanilla option value simulation for Heston model

As mentioned earlier in Section ??, the Heston model provides a closed-form (exact solution) for European options; however, this closed-form involves the evaluation of an integral, which must be approximated numerically. Specifically, by using the original Heston formulation [?, ?], the involved integrand is

$$f(\phi_j) = \text{Re}\left[\frac{e^{-i\phi_j \ln K} f_k(\phi_j; x, v)}{i\phi_j}\right], \quad k = 1, 2. \tag{16}$$

where $f_k(\phi_j; x, v)$ $(k = 1, 2)$ are the characteristic functions in the Heston model; additional details for this formula are given in [?].

In order to obtain benchmark option prices for later comparisons, we use four different quadrature rules to approximate the integral in the exact solution with a relative error less than 10^{-11} for each case. For instance, the exact solution

for the Vanilla call option $(T = 1; K_c = 110)$ is given by 13.856740; see Table **??**.

Table 10: 1 year maturity Vanilla call option closed-form results by Newton-Cotes Formulas with 4 different quadrature rules.

Quadrature Formula	Closed-Form Results
Mid-Point Rule	13.85674022071691
Trapezoidal Rule	13.85674022071776
Simpson's Rule	13.85674022071719
Simpson's 3/8 Rule	13.85674022071720

5.2.2 Vanilla call option pricing

In Table **??**, we present the results of estimating the European vanilla call option price for the case of a single fixed strike price $K_c = 110$ and for several spatial and temporal resolutions. By increasing the spatial and temporal resolutions, we see that the relatively coarse 64×64 spatial resolution has already reduced the relative error to 10^{-4}.

Table 11: European call option results of the Crank-Nicolson piecewise-quadratic scheme for several spatial and temporal resolutions.

Spatial Resolution N_x	Temporal Resolution N_T	C-N Quadratic Results	Closed-Form Results	Relative Error
4	$\lceil (4/4)^{1.5} \rceil = 1$	10.157151		2.6699E-001
8	$\lceil (8/4)^{1.5} \rceil = 3$	13.375594		3.4723E-002
16	$\lceil (16/4)^{1.5} \rceil = 8$	14.014920	13.856740	1.1415E-002
32	$\lceil (32/4)^{1.5} \rceil = 23$	13.865808		6.5441E-004
64	$\lceil (64/4)^{1.5} \rceil = 64$	13.854340		1.7320E-004

In Tables **??** and **??**, we consider several strike prices and maturity times, respectively. For both, the Crank-Nicolson piecewise-quadratic scheme is able to produce results having relative error on the order of 10^{-5}, using the same relatively coarse spatial and temporal resolutions $(64 \times 64 \times 64)$ used for the last row of Table **??**.

5.2.3 Vanilla put option pricing

We now consider the problem of estimating the European vanilla put option price; we present the results of estimating the European vanilla put option price

Table 12: European call option results of the Crank-Nicolson piecewise-quadratic scheme for several strike prices.

Strike Price K_c	C-N Quadratic Results	Closed-Form Results	Relative Error
105.0	15.936498	15.938426	1.2097E-004
110.0	13.854340	13.856740	1.7320E-004
115.0	11.976440	11.979461	2.5218E-004
130.0	7.4757386	7.4832223	1.0001E-003
150.0	3.6868859	3.7017824	4.0241E-003

Table 13: European call option results of the Crank-Nicolson piecewise-quadratic scheme for several maturity times.

Maturity Time T	C-N Quadratic Results	Closed-Form Results	Relative Error
1/12	2.1903889	2.1805420	4.5158E-003
1/4	5.7981669	5.7929266	9.0460E-004
1/2	9.3185161	9.3178780	6.8481E-005
1	13.854340	13.856740	1.7320E-004

for the case of a single fixed strike price $K_p = 90$ and for several spatial and temporal resolutions. By increasing the spatial and temporal resolutions, we see that the relatively coarse 64×64 spatial resolution has already reduced the relative error to 10^{-5}.

Table 14: European put option results of the Crank-Nicolson piecewise-quadratic scheme for several spatial and temporal resolutions.

Spatial Resolution N_x	Temporal Resolution N_T	C-N Quadratic Results	Closed-Form Result	Relative Error
4	$\lceil (4/4)^{1.5} \rceil = 1$	9.4915439		5.7457E-002
8	$\lceil (8/4)^{1.5} \rceil = 3$	10.670848		5.9652E-002
16	$\lceil (16/4)^{1.5} \rceil = 8$	10.212524	10.070148	1.4138E-002
32	$\lceil (32/4)^{1.5} \rceil = 23$	10.070997		8.4308E-005
64	$\lceil (64/4)^{1.5} \rceil = 64$	10.069421		7.2194E-005

In Tables **??** and **??**, we fix the spatial and temporal resolutions to be those of the last row of Table **??** and consider several values of the strike price and

maturity times, respectively, resulting in relative errors of about 10^{-5}.

Table 15: European put option results of the Crank-Nicolson piecewise-quadratic scheme for several strike prices.

Strike Price K_p	C-N Quadratic Results	Closed-Form Results	Relative Error
95.0	12.099818	12.100795	8.0738E-005
90.0	10.069421	10.070148	7.2194E-005
85.0	8.2562793	8.2568093	6.4189E-005
80.0	6.6576145	6.6580024	5.8261E-005
70.0	4.0796191	4.0798814	6.4291E-005
50.0	1.0927921	1.0928219	2.7269E-005

Table 16: European put option results of the Crank-Nicolson piecewise-quadratic scheme for several maturity times.

Maturity Time T	C-N Quadratic Results	Closed-Form Results	Relative Error
1/12	1.8359884	1.8299753	3.2859E-003
1/4	4.7677247	4.7625281	1.0911E-003
1/2	7.3419447	7.3412199	9.8730E-005
1	10.069421	10.070148	7.2194E-005

6 Concluding remarks

The numerical results provided in Section **??** demonstrate that finite element methods are a very effective means for solving option pricing problems, efficiently producing high-accuracy approximations whose convergence behavior can be theoretically predicted. Especially, once the optimal convergence rate is obtained by the specific finite element scheme, the numerical vanilla option values are fairly reliable and accurate. In particular, we make the following observations from the experiments. Among the different schemes tested, the combination of continuous piecewise-quadratic finite element spatial approximations and Crank-Nicolson temporal discretization scheme is found to be especially effective. For example, for European vanilla option pricing, this combination results, for wide-ranging values of the strike price and maturity time, in a relative error of order 10^{-5} using only a coarse $64 \times 64 \times 64$ spatial and temporal resolutions.

The results presented in this paper demonstrate that finite element methods deserve consideration by financial analysts as a powerful alternative to standard techniques, e.g., finite difference methods. In future work, we will further exploit other advantageous features of finite element methods by considering more complicated geometries and boundary conditions and by using local mesh refinements at locations where greater accuracy is needed, resulting in a desired accuracy level with greatly reduced computational complexity.