PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
SCHOOL OF ENGINEERING

A PARALLEL ALGORITHM FOR PRICING AMERICAN OPTIONS UNDER STOCHASTIC VOLATILITY

MAURICE ANDRE DATTAS GAMBOA

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Advisor:
GONZALO CORTÁZAR

Santiago de Chile, October 2015

PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
SCHOOL OF ENGINEERING

A PARALLEL ALGORITHM FOR PRICING AMERICAN OPTIONS UNDER STOCHASTIC VOLATILITY

MAURICE ANDRE DATTAS GAMBOA

Members of the Committee:
GONZALO CORTÁZAR
TOMÁS REYES
LORENZO NARANJO
ROLANDO REBOLLEDO

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Santiago de Chile, October 2015

To my parents,

ALPHONSE *and* SONIA.

ACKNOWLEDGEMENTS

I would like to thank my advisor Gonzalo Cortázar for his guidance, support, and opportunity to work in RiskAmerica during the development of this thesis. Also, I want to thank Lorenzo Naranjo for all his guidance, advice, patience, support and trust he had in me which resulted in this thesis. I want to thank Leonardo Medina for his contributions and help in writing this article. I would like to thank Fondecyt for the support to this research project. Finally, I would like to thank my girlfriend, family, friends, and ex colleagues from RiskAmerica for all the support they gave me during the years I spent in this work.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In this thesis a simple, fast and accurate iterative algorithm for pricing American options and solve for its early exercise boundary under a stochastic volatility setup is proposed. The algorithm is based on a fixed-point iteration that is derived from the early exercise representation of American options, and is equivalent to a multivariate Newton iteration that converges globally. This thesis is an extension of the work done by Medina (2013) on the application of the algorithm on the pricing of American options under the Black & Scholes model. The method is stable, robust, converges monotonically, and is well suited for parallel calculations and programming. The algorithm is tested using the Heston stochastic volatility model, and find that the procedure outperforms existing methodologies in terms of pricing efficiency and precision in the calculation of the early exercise frontier. In addition to a CPU implementation similar to what was done in Medina (2013) for the Black & Scholes model, we take advantage of the method's parallel nature and do a GPU implementation under the Heston model in order to assess its real benefits and surpass other benchmark methods.

Keywords: American Options, Parallel Computing, Stochastic Volatility, Fixed-Point Iteration

RESUMEN

En esta tesis, se propone un algoritmo iterativo simple, rápido y preciso para valorizar opciones Americanas y resolver su frontera de ejercicio óptimo bajo un modelo de volatilidad estocástica. El algoritmo está basado en una iteración de punto fijo que deriva de la representación del ejercicio óptimo para opciones Americanas y es equivalente una iteración multivariable de Newton que converge globalmente. Esta tesis es una extensión al trabajo realizado por Medina (2013) en la aplicación del algoritmo en la valorización de opciones Americanas bajo el modelo de Black & Scholes. El método es estable, robusto, converge monotónicamente, y es muy adecuado para cálculos y programación paralela. El algoritmo es probado utilizando el modelo de volatilidad estocástica de Heston y se halla que el procedimiento supera metodologías existentes en términos de eficiencia de valorización y precisión en el cálculo de la frontera de ejercicio óptimo. Además de hacer una implementación en una *CPU* (similar a lo hecho por Medina (2013) con el modelo de Black & Scholes), se toma ventaja de la naturaleza paralela del método y se hace una implementación *GPU* bajo el modelo de Heston para poder evaluar sus beneficios reales y superar a los otros métodos comparables.

Palabras Clave: Opciones Americanas, Programación Paralela, Volatilidad Estocástica, Iteración de Punto Fijo

I. ARTICLE BACKGROUND

1.1 Introduction

Options are financial instruments that allow holders to exercise the right to buy (Call option) or sell (Put option) a defined underlying asset at a specified strike price within a specified time frame. A European option gives the right to exercise at maturity only, whereas an American option can be exercised at any time before maturity, making the latter more difficult to model and value. This is mainly explained by the fact that to price American options the early exercise boundary is needed to determine the optimal hold and exercise policy and finally value this type of options. Due to this difficulty, American options do not have a closed form analytical solution.

Given this inherent complexity in the American option pricing, Kim (1990), Jacka (1991), and Carr et al. (1992), based on different approaches, derived an integral equation to determine the early exercise boundary for this type of options. The authors showed that the American option price is equal to the corresponding European price plus an early exercise premium, which depends on the early exercise boundary. Despite of their work, approximation and numerical techniques have to be applied in order to price these options. The fact that no closed form analytical solution to the American option pricing problem exists has encouraged researchers to come up with numerical methods to price American options focusing on both the pricing precision and efficiency.

The need of more sophisticated models to value options and better understand its drivers resulted in several models that enhance the geometric Brownian motion usually assumed for the underlying asset when options are priced. In particular, the ones that assume stochastic instead of constant volatility have proved to be useful to improve the pricing and modelling by increasing the degree of complexity of the problem. Among the several that exist, the one proposed by Heston (1993) stands out.

In this thesis, it is proved that the Fixed Point Iteration algorithm (FPI) initially proposed in Medina (2013) and applied to a more sophisticated American option mathematical model (Heston stochastic volatility model) continues to be stable, accurate, efficient, converges monotonically to the true option price as we increase time-steps, and remains as a real alternative to other stochastic volatility numerical methods given that it outperforms them. Based on numerical results, this thesis shows that FPI is capable of generating a smoother policy compared to the Transformation Procedure algorithm (TP), resulting in an efficient method in terms of grid size and pricing. It is important to note that the GPU implementation of the Fixed Point Iteration proves to be useful and efficient for more complex models as calculation times for the algorithm in the CPU increase signifcantly compared to the Black & Scholes implementation.

The rest of the thesis is organized as follows. Section 1.2 defines the main hypothesis. Section 1.3 defines the main objectives of this work. Section 1.4 introduces the theoretical framework and numerical methodologies related to American option pricing under the Heston model. Section 1.5 introduces the proposed parallel algorithm for pricing American options (Fixed Point Iteration). Section 1.6 presents the perspectives for future research. Chapter II contains the main article of this thesis, written in colaboration with Gonzalo Cortázar, Leonardo Medina, and Lorenzo Naranjo. This article contains some chapters based on the work of Medina (2013). Within this, Section 2.1 introduces. In Section 2.2 we define the Fixed Point Iteration procedure. Section 2.3 applies and implements the methodology to the pricing of American options under the Black & Scholes model and compares its performance with some numerical methods present in the literature. It is important to note that most of this Section was done entirely by Leonardo Medina. In Section 2.4 the same analysis as in Section 2.3 is done, but applied and implemented for American options under the Heston model, and compared with numerical methods that address stochastic volatility option pricing. Section 2.5 concludes.

1.2 Hypothesis

The hypothesis of this work is that the new iterative and parallel method presented in Medina (2013) can be adapted and implemented on a Graphical Processing Unit (GPU) or any multi-core processing system to exploit its parallel characteristic and obtain a novel, fast, simple, and precise procedure to solve the early exercise boundary and pricing of American options under the stochastic volatility model of Heston (1993). Under this setup, the algorithm should surpass other numerical methodologies present in the literature to price this type of options in terms of speed, precision and efficiency.

1.3 Main Objectives

The main objective of this thesis is to present, explain, develop and extend the application of the Fixed Point Iteration algorithm (FPI) to a stochastic volatility layout. This should be achieved by applying it to price American options, solve its early exercise boundary explicitly under the Heston model, and prove through its implementation the real benefits in terms of speed and precision of exploiting the parallel nature of the procedure. Within this, the thesis has three specific objectives: First, this work intends to establish a theoretical framework that supports the utilization and the main features of the methodology. The second objective is to adapt the Fixed Point Iteration to the Heston stochastic volatility setup. Finally, the third objective is to implement: a) the proposed method under the aforementioned model and b) implement some numerical methodologies or benchmarks present in the literature in order to: Establish the stability and convergence of the proposed algorithm, and compare these methods in terms of speed, accuracy, efficiency, and precision in the resolution of the early exercise frontier. Given that one of the proposed benchmark methods calculates the early exercise boundary as part of its algorithm (Transformation Procedure or TP), a combined method is proposed and implemented (TP-FPI) to assess the possibility and real benefits of combining pricing methods.

1.4 Methodology

To accomplish the aforementioned objectives, it was necessary to solve the pricing option model using the expression proposed by Rutkowski (1994) to formulate and solve the early exercise frontier and the option's value. The Fixed Point Iteration procedure and the benchmark methods were implemented using Matlab on a two-core 2.00 GHz Intel Core i7 with 8.00 GB RAM, and additional CUDA implementations on a Nvidia GTX Titan GPU (2,688 CUDA cores and a 837 MHz base clock) for the FPI and TP-FPI procedures. In particular, GPU routines for the modified Bessel function and a numerical adaptive integration algorithm based on Matlab's procedure were programmed. The other numerical methods that were implemented to price Heston American options and compare them with our algorithm were: Projected Succesive Over-Relaxation - PSOR (Cryer (1971)) and Componentwise Splitting - CS (Ikonen & Toivanen (2007*a*)), which were implemented based on Ikonen & Toivanen (2007*b*), and the Transformation Procedure - TP (Chockalingam & Muthuraman (2011)). Time and Root Mean Square Error (RMSE) were measured against the true values accepted in the literature. Given that both FPI and TP allow to calculate the critical price boundary explicitly: a) Early exercise boundaries were calculated using both methods and 2) a combined method was implemented and compared in terms of performance against the other procedures.

1.5 Literature Review

1.5.1 American options and the Heston stochastic volatility model

American option pricing under the constant volatility model proposed by Black & Scholes (1973) has been deeply studied in the literature (Broadie & Detemple (2004) does an extensive revision of this). However, its study is not as extensive when these class of options are modeled under diffusion models that are more consistent with reality and actual prices, as would, among others, be the case when stochastic volatility in the underlying asset is included (Broadie et al. (2000), Scott (1987)). This is mainly due

to the complexity involved in pricing these type of options. Different models have been proposed in the literature to approach stochastic volatility (Stein & Stein (1991), Scott (1987), and Hull & White (1987), among others), however, the one proposed by Heston (1993) is currently the most well known and commonly studied.

Given that an optimal exercise policy has to be determined in order to price American options by defining an optimal early exercise boundary that separates the price-time region into hold and exercise regions, the American option pricing becomes a free boundary problem. As a result, closed-form solutions for this type of options do not exist, even assuming the most basic case of constant volatility. The difficulty of valuing American options stems from the fact that the optimal exercise rule is unknown ex-ante, and must be computed simultaneously with the price of the option. In the simple case when the underlying asset follows a geometric Brownian motion (GBM), Kim (1990), Jacka (1991), and Carr et al. (1992) derived an integral equation that provides an explicit solution to the optimal early exercise boundary for American options. Based on these works, Medina (2013) provided a novel iterative method to price American options under a GBM and the theoretical background for this thesis. Rutkowski (1994) extended the analysis and provided an expression for the optimal exercise boundary in the more general case when the underlying asset follows a diffusion.

1.5.2 Numerical Methods for American options under the Heston model

For the case of American options based on a stochastic volatility model (Heston (1993), among others), there are several methodologies that stand out, especially the ones based on the resolution of the corresponding partial differential equation that arises. Given the early exercise possibility of American options, they can be modeled as a time dependent linear complementarity problem (LCP). Among the different existing methods that solve the pricing problem by the LCP approach, the projected successive overrelaxation method (PSOR) proposed by Cryer (1971) is the most well known and widely used for pricing American options. The projected full approximation scheme (PFAS) multigrid

method by Brandt & Cryer (1983), which is a procedure based on LCPs has been applied on American options by Clarke & Parrot (1999) and Osterlee (2003), where the method is enhanced with a projected line Gauss-Siedel smoother. In addition to these, the operator splitting method by Ikonen & Toivanen (2004), the penalty method of Zvan et al. (1998), and the Componenwise Splitting method (CS) of Ikonen & Toivanen (2007*a*) are deeply studied by Ikonen & Toivanen (2007*b*). Here, the authors compared the five methods for American option pricng under stochastic volatility, and show that, although these procedures are comparable in terms of precision, CS is the best method in terms of speed, while PSOR is the slowest, but the easiest to implement. Another significant contribution in terms of methodologies that solve the underlying PDE was the Transformation Procedure (TP) proposed by Chockalingam & Muthuraman (2011)) in which the American option problem can be transformed into a series of European options where its main characteristic is that the exercise policy is known ex-ante, thus becoming a sequence of fixed-boundary problems. For more details regarding the benchmark methods implemented in this thesis see Appendix E.

Another set of methods capable of solving American options under stochastic volatility are the ones based on simulation of multi dimensional American options where the least-squares error algorithm of Longstaff & Schwartz (2001), the primal-dual method of Andersen & Broadie (2004), and the stochastic mesh of Broadie & Glasserman (2004) are highlighted. Although most of these procedures were originally created to solve American options under constant volatility, its versatility allows them to adapt to the stochastic volatility case. It is important to mention that there are not many methods that use the resolution of the stochastic volatility integral equation as an initial approach mainly due to its complexity (Chiarella & Ziogas (2005) in which the authors solve the equation using the method of Kallast & Kivinukk (2003)).

1.6 Future Research

The Fixed Point Iteration algorithm defined in this thesis shows some promising features for future research given its capacity of solving smooth and accurate early exercise boundaries explicitly, its parallel nature which can improve efficiency and performance by doing a GPU implementation, and its flexibility to adapt to more complex diffusion models. For instance, research could be focused on better understanding the drivers behind the early exercise frontiers and pricing of American options under more complex diffusion setups. Furthermore, the adaptation of this method to solve more complex pricing models could be compared in terms of performance with other numerical methods for these particular models.

II. A PARALLEL ALGORITHM FOR PRICING AMERICAN OPTIONS

2.1 Introduction

Notwithstanding impressive advances in the pricing of American options for constant volatility diffusions (see e.g. Broadie & Detemple 2004), the study of American options under stochastic volatility has remained elusive, even though it is an important feature of many financial markets (Broadie et al. 2000, Scott 1987). In this paper, we propose a simple, fast and accurate iterative algorithm for pricing American options and solve for its early-exercise boundary when the volatility is stochastic.

The difficulty of valuing American options stems from the fact that the optimal exercise rule is unknown ex-ante, and must be computed simultaneously with the price of the option. In the simple case when the underlying asset follows a geometric Brownian motion, Kim (1990), Jacka (1991), and Carr et al. (1992) derive an integral equation that provides an explicit solution to the optimal early-exercise boundary for American options. Rutkowski (1994) extends the analysis and provides an expression for the optimal early-exercise boundary for general diffusions.

Our algorithm is based on a fixed-point iteration (FPI) that is derived from the early-exercise representation of American options. As we show in the paper, the FPI is equivalent to a multivariate Newton iteration that converges globally in which the Jacobian is diagonal, simple to compute, and hence numerically cheap to invert. Hence, the FPI inherits the fast quadratic convergence of the Newton method.

We test empirically our algorithm using the Black & Scholes (1973) and the Heston (1993) stochastic volatility models, and find that our methodology outperforms existing approaches in terms of efficiency and also in the precision of the early-exercise frontier. We find that the method is stable, robust, converges monotonically to its true value, and is well suited for parallel calculations and programming, increasing substantially the speed

of execution. The performance of our method can be improved further by the use of Richardson extrapolation.

Iterative methods are interesting alternatives to other numerical techniques because they are well-suited for parallel implementations. In recent years, parallel algorithms have become attractive with the advent of multi-core processors and graphic processing units (GPUs), which can be installed on any modern PC at a reasonable cost. We find that using these novel features of modern hardware significantly improves the efficiency of the FPI.

2.1.1 The Black & Scholes Model

A robust method to solve the early-exercise boundary equation of Kim (1990) in the case of a Geometric Brownian motion (GBM) was already studied by Kallast & Kivinukk (2003). The authors show that solving such equation directly yields a fast and stable way to compute the early-exercise boundary. Their approach works **sequentially** in that for a given solution of the boundary up to maturity T, they compute the boundary up to maturity $T + \Delta T$.

In contrast, our approach operates in **parallel**. We iterate over a series of approximating functions $B^{(k)}(\tau)$, where $\tau \in [0, T]$ represents time-to-maturity, in order to compute the early-exercise boundary $S_c(\tau)$. In each iteration a new approximation of the **whole** early-exercise boundary $B^{(k+1)}$ is obtained as the result of applying an operator Υ to the previous approximation $B^{(k)}$, i.e. $B^{(k+1)} = \Upsilon(B^{(k)})$. The operator is derived from the equation that determines the early-exercise boundary, and is equivalent to a multivariate Newton iteration. For each maturity τ, the new value of the early-exercise boundary at that point is given by $B^{(k+1)}(\tau) = \Phi_\tau(B^{(k)})$, where Φ_τ is the functional that defines the operator Υ for each $\tau \in [0, T]$.

The resulting algorithm requires an initial guess of the early-exercise boundary, and a numerical rule to evaluate the integral part of Kim's (1990) equation. We explore three

different alternatives for implementing the method. First, we estimate the integrals appearing in the early-exercise boundary equation by use of the trapezoidal rule as in Kallast & Kivinukk (2003). We start the iterations using two different priors: the flat guess of Kim et al. (2013), and the initial guess of Barone-Adesi & Whaley (1987). Second, we follow Kim et al. (2013) and estimate the integrals appearing in the boundary equation by interpolating a few discretized points, and using the more advanced quadrature procedure such as Gauss-Kronrod.

We perform a thorough empirical analysis, and compare the different implementations of our method with several standard methods found in the literature. The numerical results show that the implementations of our method that use the trapezoidal rule achieves the best performance among all the studied algorithms. In particular, the best performance is obtained by using the smart initial guess of Barone-Adesi & Whaley (1987).

In a recent paper, Kim et al. (2013) propose an iterative procedure based on an equation developed by Little et al. (2000) to solve for the early-exercise boundary. Even though the method of Kim et al. (2013) and ours are both iterative procedures, we show in the paper that our implementation is more efficient in terms of speed, accuracy, and stability. In addition, it is not immediately clear how to extend their approach to a setup with stochastic volatility as we do in our paper. Furthermore, the authors do not prove the convergence of their method while in this paper we prove that our methodology converges fast and globally. Finally, our results are consistent with previous findings of Kallast & Kivinukk (2003) who show that solving the equation of Kim (1990) directly yields stable results.

2.1.2 The Heston Model

Several methodologies have been proposed to price American options under stochastic volatility, either by solving the resulting PDE (Cryer 1971, Brandt & Cryer 1983, Clarke & Parrot 1999, Osterlee 2003, Ikonen & Toivanen 2004, Zvan et al. 1998, Ikonen & Toivanen 2007*a*,*b*, Chockalingam & Muthuraman 2011), or by simulation (Longstaff & Schwartz 2001, Andersen & Broadie 2004, Broadie & Glasserman 2004). However, only

one previous study solves the early-exercise equation directly (Chiarella & Ziogas 2005) by using the method of Kallast & Kivinukk (2003).

As in the case of constant volatility, in the Heston model our algorithm iterates over a series of approximating functions $B^{(k)}(\tau, v)$, where $\tau \in [0, T]$ represents the time-to-maturity, $v \in D$ is the variance and D denotes the domain of v, in order to compute the early-exercise surface boundary $S_c(\tau, v)$. In each iteration a new approximation of the early-exercise policy $B^{(k+1)}$ is obtained as the result of applying an operator Υ to the previous approximation $B^{(k)}$, i.e. $B^{(k+1)} = \Upsilon(B^{(k)})$. The operator is derived from the equation that determines the early-exercise boundary. For each time-to-maturity τ and variance v, the new value of the early-exercise surface at that point is given by $B^{(k+1)}(\tau, v) = \Phi_{\tau,v}(B^{(k)})$, where $\Phi_{\tau,v}$ is a functional that defines the operator Υ at each $\tau \in [0, T]$ and $v \in D$.

In the Heston implementation of our algorithm, we use the Gauss-Konrod quadrature to invert the characteristic function arising from the model, and a spline interpolation to price the option once we have found the early-exercise boundary. Due to the significant increase in computation time, we take advantage of the parallel nature of our algorithm and implement the solution on a GPU. As a technical contribution, we ported Matlab's adaptive numerical integration algorithm and the numerical evaluation of modified Bessel functions of first kind for complex arguments to CUDA.

Empirically, we compare our method with the projected successive over-relaxation method (PSOR) of Cryer (1971), and the component-wise splitting algorithm (CS) proposed by Ikonen & Toivanen (2007*a*) (both implemented as in Ikonen & Toivanen 2007*b*), as well as the transformation procedure (TP) of Chockalingam & Muthuraman (2011). Given that the TP also solves the early-exercise boundary explicitly as in our method, we compare both policies to analyze their performance. We also implement a combined FPI-TP in which the option pricing algorithm of Chockalingam & Muthuraman (2011) uses as an input the early-exercise policy calculated in the first stage of our method.

Our results suggest that our method continues to be stable, accurate, and efficient when applied to the Heston model. We show that the FPI is capable of generating a smoother early-exercise policy compared to the TP. Furthermore, the GPU implementation of the FPI proves to be useful and efficient for more complex models as calculation times in the CPU increase significantly compared to the GBM implementation.

The rest of the paper is organized as follows. In Section 2.2 we introduce our FPI algorithm and recall some standard results for American options. We also show the equivalence between the FPI and the multivariate Newton method and prove its convergence. Section 2.3 implements our method for American options for GBM, and compares its performance with several standard methods proposed in the literature. In Section 2.4 we repeat the analysis of Section 2.3 for the Heston model. Section 2.5 concludes.

2.2 Solving for the Early Exercise Boundary

In this section we introduce our FPI to price American options and solve for its early-exercise boundary under stochastic volatility. It should be noted that the analysis presented in this section can be extended to any process for which the early-exercise representation is valid, which is for example the case of diffusions.

2.2.1 Early Exercise Representation of American Put Options

Throughout the paper we consider a continuous-time economy in which a complete probability space $(\Omega, \mathscr{F}, \mathcal{P})$ and a filtration $\mathbb{F} = \{\mathscr{F}_t; t \geq 0\}$ are defined satisfying the usual conditions (see, e.g., Protter 2005). The dynamics of the risky asset are characterized by the following process under the pricing measure $\mathcal{Q}$, equivalent to $\mathcal{P}$:

$$\frac{\mathrm{d}S_t}{S_t} = (r - q)\,\mathrm{d}t + \sigma(Y_t)\,\mathrm{d}W_{1,t}, \tag{2.1}$$

$$\mathrm{d}Y_t = \mu(Y_t)\,\mathrm{d}t + v(Y_t)\,\mathrm{d}W_{2,t}, \tag{2.2}$$

where W_1 and W_2 are two $\mathcal{Q}$-correlated Brownian motions such that $(\mathrm{d}W_{1,t}) \cdot (\mathrm{d}W_{2,t}) = \rho \mathrm{d}t$, r is the risk-free rate, q denotes the dividend yield; $\sigma \colon D \to \mathbb{R}^+$, $v \colon D \to \mathbb{R}^+$, and $\mu \colon D \to \mathbb{R}^+$ are real-valued functions where $D \subset \mathbb{R}$ denotes the domain of Y.

We first consider an American put option written on S with maturity T and exercise price K. In the following, we denote by $\mathbb{E}(\cdot)$ the expectation under $\mathcal{Q}$ conditional on S_0 and Y_0, and define $x^+ = \max(0, x)$. The price P of the American put is then given by (see e.g. Schroder 1999):

$$P = \sup_{\tilde{\tau} \in [0,T]} \mathbb{E}\left(e^{-r\tilde{\tau}}[K - S_{\tilde{\tau}}]^+\right), \tag{2.3}$$

where the supremum is taken over all stopping times $\tilde{\tau} \in [0, T]$. The price p of an equivalent European put option satisfies:

$$p = p(S_0, Y_0, T; K) = \mathbb{E}\left(e^{-rT}[K - S_T]^+\right). \tag{2.4}$$

By defining $X_T = S_T/S_0$, we can re-write equation (2.4) as:

$$p = K\,\mathbb{E}\left(e^{-rT}\mathbf{1}_{\{X_T \leq \frac{K}{S_0}\}}\right) - S_0\,\mathbb{E}\left(e^{-rT}X_T\mathbf{1}_{\{X_T \leq \frac{K}{S_0}\}}\right). \tag{2.5}$$

The difference between P_0 and p_0 is called the early-exercise premium e_0, i.e.

$$P_0 = p_0 + e_0. \tag{2.6}$$

Applying the results of Rutkowski (1994), in our setting we have that:

$$\begin{aligned} e_0 &= e(S_0, Y_0, T; K, S_c), \\ &= \mathbb{E}\left(\int_0^T e^{-r\tau}(rK - qS_\tau)\mathbf{1}_{\{S_\tau \leq S_c(T-\tau, Y_\tau)\}}\,\mathrm{d}\tau\right), \\ &= \int_0^T rKe^{-r\tau}\,\mathbb{E}\left(\mathbf{1}_{\{X_\tau \leq \frac{S_c(T-\tau,Y_\tau)}{S_0}\}}\right) - qS_0e^{-r\tau}\,\mathbb{E}\left(X_\tau\mathbf{1}_{\{X_\tau \leq \frac{S_c(T-\tau,Y_\tau)}{S_0}\}}\right)\mathrm{d}\tau. \end{aligned} \tag{2.7}$$

In Equation (2.7), $S_c(T-\tau, Y_\tau)$ denotes the critical spot price that triggers early-exercise conditional on Y_τ and time-to-maturity being $T-\tau$. For American put options, early-exercise is optimal whenever $S_\tau \leq S_c(T-\tau, Y_\tau)$. Hence, we can make explicit the dependence on initial conditions and the early-exercise boundary by writing:

$$P(S_0, Y_0, T; K, S_c) = p(S_0, Y_0, T; K) + e(S_0, Y_0, T; K, S_c). \tag{2.8}$$

For each time-to-maturity T and $Y_0 \in D$, the optimal exercise price $S_c(T, Y_0)$ satisfies the following equation:

$$K - S_c(T, Y_0) = p(S_c(T, Y_0), Y_0, T; K) + e(S_c(T, Y_0), Y_0, T; K, S_c), \tag{2.9}$$

since it must be the case that $P(S_c(T, Y_0), Y_0, T; K, S_c) = K - S_c(T, Y_0)$.

2.2.2 The Fixed Point Iteration

We start re-writing (2.9) as:

$$S_c(T, Y_0) U_{T,Y_0}(S_c) - K V_{T,Y_0}(S_c) = 0, \tag{2.10}$$

where

$$U_{T,Y_0}(S_c) = 1 - \mathbb{E}\left(e^{-rT} X_T \mathbf{1}_{\{X_T \leq \frac{K}{S_c(T,Y_0)}\}}\right) - \int_0^T q e^{-r\tau}\, \mathbb{E}\left(X_\tau \mathbf{1}_{\{X_\tau \leq \frac{S_c(T-\tau,Y_\tau)}{S_c(T,Y_0)}\}}\right) \mathrm{d}\tau, \tag{2.11}$$

$$V_{T,Y_0}(S_c) = 1 - \mathbb{E}\left(e^{-rT} \mathbf{1}_{\{X_T \leq \frac{K}{S_c(T,Y_0)}\}}\right) - \int_0^T r e^{-r\tau}\, \mathbb{E}\left(\mathbf{1}_{\{X_\tau \leq \frac{S_c(T-\tau,Y_\tau)}{S_c(T,Y_0)}\}}\right) \mathrm{d}\tau. \tag{2.12}$$

In the Appendix we show that $U_{T,Y_0}(\cdot) > 0$, which allows us to write:

$$S_c(T, Y) = K \frac{V_{T,Y}(S_c)}{U_{T,Y}(S_c)}, \quad \forall T \in [0, \infty), Y \in D. \tag{2.13}$$

We propose to use (2.13) to perform a fixed-point iteration in order to find $S_c(T, Y)$ for all $Y \in D$ and all $T \in [0, \infty)$. Starting from an initial guess $B^{(0)}$ of the whole

early-exercise boundary, we obtain a new approximation $B^{(1)}$ as follows:

$$B^{(1)}(T,Y) = K\frac{V_{T,Y}(B^{(0)})}{U_{T,Y}(B^{(0)})}, \quad \forall T \in [0,\infty), Y \in D.$$

Hence, given an approximation of the whole early-exercise frontier $B^{(k)}$ after k iterations, we can find a new approximation $B^{(k+1)}$:

$$B^{(k+1)}(T,Y) = K\frac{V_{T,Y}(B^{(k)})}{U_{T,Y}(B^{(k)})}, \quad \forall T \in [0,\infty), Y \in D. \tag{2.14}$$

Note that, at each step k, the new approximation $B^{(k+1)}(t_1, y_1)$ for a given $T = t_1$ and $Y = y_1$, can be computed independently from the new approximation $B^{(k+1)}(t_2, y_2)$ corresponding to $T = t_2$ and $Y = y_2$. This feature of the fixed-point iteration is convenient from a numerical point of view since it allows to compute the refinement of $B^{(k)}$ at each point (T, Y) in parallel.[1]

2.2.3 The Method as a Newton Iteration

In numerical applications, we estimate a discretized version of the early-exercise boundary $S_c(T,Y)$ where $T \in \{T_0, T_1, \ldots, T_M\}$ and $Y \in \{Y_0, Y_1, \ldots, Y_N\}$. We denote by $b_{i,j}^{(k)}$ the approximation at stage k of $S_c(T_i, Y_j)$, and

$$B^{(k)} = \begin{pmatrix} b_{0,0}^{(k)} & b_{0,1}^{(k)} & \cdots & b_{0,N}^{(k)} \\ b_{1,0}^{(k)} & b_{1,1}^{(k)} & \cdots & b_{1,N}^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ b_{M,0}^{(k)} & b_{M,1}^{(k)} & \cdots & b_{M,N}^{(k)} \end{pmatrix}.$$

Also, define

$$F_{i,j}(B^{(k)}) = b_{i,j}^{(k)} U_{i,j}(B^{(k)}) - K V_{i,j}(B^{(k)}),$$

[1] Note, however, that the method is not asynchronous.

where $U_{i,j}(B^{(k)})$ and $V_{i,j}(B^{(k)})$ denote the discretized versions of (2.11) and (2.12). The numerical approximation B of S_c should then solve $F_{i,j}(B) = 0$ for all $i = 0, 1, \cdots, M$ and $j = 0, 1, \cdots, N$.

The numerical problem consists in solving a system of $M \times N$ equations in $M \times N$ variables. In order to simplify notation, define $\boldsymbol{B} = (B)$ and $\boldsymbol{F} = (F)$ where $(\cdot)$ represents the vectorization operator that stacks the columns of a matrix, and switch from the double index notation (i, j) to a single index notation, such that $b_{i,j}^{(k)}$ becomes $b_h^{(k)}$, $b_i^{(k)}$, or $b_j^{(k)}$ depending on the case. We do the same for $F_{i,j}(\cdot)$. The problem then becomes finding $\boldsymbol{B}$ such that $\boldsymbol{F}(\boldsymbol{B}) = 0$.

The multivariate Newton method defines a series of iterations that approximate the solution of the system of equations. Given an estimate $\boldsymbol{B}^{(k)}$, we define a new estimate $\boldsymbol{B}^{(k+1)}$ as:

$$\boldsymbol{B}^{(k+1)} = \boldsymbol{B}^{(k)} - [J_{\boldsymbol{F}}(\boldsymbol{B}^{(\boldsymbol{k})})]^{-1}\boldsymbol{F}(\boldsymbol{B}^{(\boldsymbol{k})}),$$

where $J_{\boldsymbol{F}}(\boldsymbol{B}^{(\boldsymbol{k})})$ is the Jacobian matrix whose (i, j) element is given by $\partial F_i(\boldsymbol{B}^{(\boldsymbol{k})})/\partial b_j^{(k)}$.

We first note that $\partial F_i(\boldsymbol{B}^{(\boldsymbol{k})})/\partial b_j^{(k)} \approx 0$ for $i \neq j$ since the derivatives will be taken with respect to integrands appearing in equations (2.11) and (2.12). If we use for example the Trapezoidal method to approximate the integrals with respect to τ and y (assuming that we use the density $g(y)$ of Y_τ to compute the expectation), then $\partial F_i(\boldsymbol{B}^{(\boldsymbol{k})})/\partial b_j^{(k)}$ for $i \neq j$ will be of order $\Delta\tau\Delta y$, which can be made arbitrarily small as we refine the grid for τ and y.

Second, we note that:

$$\frac{\partial F_h(\boldsymbol{B}^{(\boldsymbol{k})})}{\partial b_h^{(k)}} = U_h(\boldsymbol{B}^{(k)}) + b_h^{(k)}\frac{\partial U_h(\boldsymbol{B}^{(k)})}{\partial b_h^{(k)}} - K\frac{\partial V_h(\boldsymbol{B}^{(k)})}{\partial b_h^{(k)}}.$$

However, if the asset exhibits constant returns to scale, the ratio S_c/K is independent of K. This means that we can fix the level of K arbitrarily small, and that choice will not affect the behavior of the ratio S_c/K. We could imagine then solving for the early-exercise

boundary for $K = \varepsilon$, where ε is an arbitrarily small number, in which case we will have:

$$\frac{\partial F_h(\boldsymbol{B^{(k)}})}{\partial b_h^{(k)}} \approx U_h(\boldsymbol{B}^{(k)}) > 0.^2$$

Plunging-in these results into the Newton iteration gives:

$$b_h^{(k+1)} = b_h^{(k)} - \frac{b_h^{(k)} U_h(\boldsymbol{B}^{(k)}) - K V_h(\boldsymbol{B}^{(k)})}{U_h(\boldsymbol{B}^{(k)})} = K \frac{V_h(\boldsymbol{B^{(k)}})}{U_h(\boldsymbol{B^{(k)}})},$$

which is the FPI defined in (2.14).

In the Appendix we show that $U_h(\boldsymbol{B}^{(k)})$ is increasing in $b_h^{(k)}$, which implies that:

$$\frac{\partial^2 F_h(\boldsymbol{B^{(k)}})}{\partial b_h^{(k)} \partial b_h^{(k)}} > 0,$$

for a discretization of the integrals accurate enough.

Hence, if we assume that $\left(b_1^{(k)}, b_2^{(k)}, \ldots, b_{h-1}^{(k)}\right)$ have converged, i.e. $F_1(\boldsymbol{B^{(k)}}) = 0, F_2(\boldsymbol{B^{(k)}}) = 0, \ldots, F_{h-1}(\boldsymbol{B^{(k)}}) = 0$, Newton's method is guaranteed to converge for $b_h^{(k)}$ since $F_h(\boldsymbol{B^{(k)}})$ is increasing and convex in $b_h^{(k)}$. Therefore, a simple induction argument reveals that the method will eventually converge globally for all $\boldsymbol{B}^{(k)}$. We collect the previous remarks in the following proposition.

PROPOSITION 1. *The iteration defined as:*

$$B^{(k+1)}(T, Y) = K \frac{V_{T,Y}(B^{(k)})}{U_{T,Y}(B^{(k)})}, \quad \forall T \in [0, \infty), Y \in D$$

is equivalent to a Newton iteration and converges globally to the solution of

$$S_c(T, Y_0) U_{T,Y_0}(S_c) - K V_{T,Y_0}(S_c) = 0, \quad \forall T \in [0, \infty), Y \in D.$$

---

[2]Note that we could prove this by induction as follows. Set $b_h^{(0)} = K = \varepsilon$. Assume that $\frac{\partial F_h(\boldsymbol{B^{(k)}})}{\partial b_h^{(k)}} = U_h(\boldsymbol{B}^{(k)})$ for some k. Then we have that $b_h^{(k+1)} = K \frac{V_h(\boldsymbol{B^{(k)}})}{U_h(\boldsymbol{B^{(k)}})} \propto \varepsilon$.

2.2.4 American Call Options

The value of the early-exercise premium of an American call option with the same characteristics can be obtained in a similar manner. We provide these results for completeness. Indeed, we have that:

$$C_0 = \sup_{\widetilde{\tau}\in[0,T]} \mathbb{E}\left(e^{-r\tilde{\tau}}(S_{\widetilde{\tau}} - K)^+\right) = c_0 + \breve{e}_0, \tag{2.15}$$

where

$$\begin{aligned}\breve{e}_0 &= \mathbb{E}\left(\int_0^T e^{-r\tau}(q_\tau S_\tau - r_\tau K)\mathbf{1}_{\{S_\tau \geq S_c(T-\tau,Y_\tau)\}}\,\mathrm{d}\tau\right), \\ &= \int_0^T qS_0\,\mathbb{E}\left(e^{-r\tau}X_\tau \mathbf{1}_{\{X_\tau \geq \frac{S_c(T-\tau,Y_\tau)}{S_0}\}}\right) - rK\,\mathbb{E}\left(e^{-r\tau}\mathbf{1}_{\{X_\tau \geq \frac{S_c(T-\tau,Y_\tau)}{S_0}\}}\right)\,\mathrm{d}\tau. \quad (2.16)\end{aligned}$$

and

$$\begin{aligned}c_0 &= c(S_0, Y_0, T; K), \\ &= \mathbb{E}\left(e^{-rT}[K - S_T]^+\right) \quad (2.17) \\ &= S_0\,\mathbb{E}\left(e^{-rT}X_T\mathbf{1}_{\{X_T \geq \frac{K}{S_0}\}}\right) - K\,\mathbb{E}\left(e^{-rT}\mathbf{1}_{\{X_T \geq \frac{K}{S_0}\}}\right), \quad (2.18)\end{aligned}$$

For American call options, early-exercise is optimal whenever $S_\tau \geq S_c(T-\tau, Y_\tau)$. Hence, we can make explicit the dependence on initial conditions and the early-exercise boundary by writing:

$$C(S_0, Y_0, T; K, S_c) = c(S_0, Y_0, T; K) + \breve{e}(S_0, Y_0, T; K, S_c). \tag{2.19}$$

For each time-to-maturity T and $Y_0 \in D$, the optimal exercise price $S_c(T, Y_0)$ satisfies the following equation:

$$S_c(T, Y_0) - K = c(S_c(T, Y_0), Y_0, T; K) + \tilde{e}(S_c(T, Y_0), Y_0, T; K, S_c), \tag{2.20}$$

since it must be the case that $C(S_c(T, Y_0), Y_0, T; K, S_c) = S_c(T, Y_0) - K$.

We can re-write (2.20) as:

$$S_c(T, Y_0)\breve{U}_{T,Y_0}(S_c) - K\breve{V}_{T,Y_0}(S_c) = 0, \tag{2.21}$$

where

$$\breve{U}_{T,Y_0}(S_c) = 1 - \mathbb{E}\left(e^{-rT}X_T\mathbf{1}_{\{X_T \geq \frac{K}{S_c(T,Y_0)}\}}\right) - \int_0^T qe^{-r\tau}\,\mathbb{E}\left(X_\tau\mathbf{1}_{\{X_\tau \geq \frac{S_c(T-\tau,Y_\tau)}{S_c(T,Y_0)}\}}\right)\,\mathrm{d}\tau,$$

$$\breve{V}_{T,Y_0}(S_c) = 1 - \mathbb{E}\left(e^{-rT}\mathbf{1}_{\{X_T \geq \frac{K}{S_c(T,Y_0)}\}}\right) - \int_0^T re^{-r\tau}\,\mathbb{E}\left(\mathbf{1}_{\{X_\tau \geq \frac{S_c(T-\tau,Y_\tau)}{S_c(T,Y_0)}\}}\right)\,\mathrm{d}\tau.$$

Therefore, we can also write:

$$S_c(T, Y) = K\frac{\breve{V}_{T,Y}(S_c)}{\breve{U}_{T,Y}(S_c)}, \quad \forall T \in [0, \infty), Y \in D. \tag{2.22}$$

2.3 American Options Under The Black & Scholes Model

In the Black & Scholes economy, the risk-free rate r is constant, the risky asset S pays a constant dividend yield q and follows a geometric Brownian motion process with constant volatility under the pricing measure $\mathcal{Q}$, equivalent to $\mathcal{P}$:

$$\frac{dS_t}{S_t} = (r - q)dt + \sigma dW_t.$$

In the setup described above, it is well known that we can derive a closed-form expression for the early-exercise premium (see e.g. Kim 1990, Jacka 1991, Carr et al. 1992). Let:

$$e_0 = \int_0^T rKe^{-r\tau}N(-d_2(S_0, S_c(T-\tau), \tau)) - qS_0e^{-q\tau}N(-d_1(S_0, S_c(T-\tau), \tau))\,\mathrm{d}\tau, \tag{2.23}$$

$$p_0 = Ke^{-rT}N(-d_2(S_0, K, T)) - S_0e^{-qT}N(-d_1(S_0, K, T)), \tag{2.24}$$

where:

$$d_1(x, y, \tau) = \frac{\log(x/y) + (r - q + 0.5\sigma^2)\tau}{\sigma\sqrt{\tau}},$$
$$d_2(x, y, \tau) = d_1(x, y, \tau) - \sigma\sqrt{\tau},$$

and $N(x) = \mathcal{Q}(X \leq x)$, where X is a standard normally distributed random variable under $\mathcal{Q}$. The price the American put is then given by (2.6). Hedge parameters can be computed in closed-form as shown in Huang et al. (1996, p. 284).

2.3.1 Numerical Implementation

We calculate the price of an American option in three steps. First, we start with an initial guess of the early-exercise boundary. Second, the initial guess is updated using Equation (2.14). We repeat this procedure until the maximum difference between two estimated curves during two sequential iterations is below a specified tolerance. Finally, the price of an American option is obtained by using the estimated early-exercise boundary along with Equation (2.6).

In order to implement our numerical method we need to estimate the integrals appearing in the numerator and denominator of Equation (2.14), as well as the integral appearing in Equation (2.6). We proceed by using two different methods: the trapezoidal rule, as implemented in Kallast & Kivinukk (2003), and a numerical quadrature method similar to the one used by Kim et al. (2013).

a) Trapezoidal Rule

We first divide the time interval $[0, T]$ into N_T subintervals of length $\Delta t = T/N_T$. We keep the number of points fixed through all iterations. Therefore, by varying the number of initial points it is possible to increase the accuracy. However, the trapezoidal method is also parallelizable, so its computational cost does not increase much with the grid size. Our implementation of the trapezoidal rule takes advantage of this feature.

In the following, we describe the iteration when applied to American put options. The case of American call options can be handled in a similar fashion. In our empirical tests, we initialize the method for American put options either by setting:

$$b_i^{(0)} = K \min(1, r/q), \quad \forall i = 0, \ldots, N_T \tag{2.25}$$

as in Kim et al. (2013), or by using the initial guess of Barone-Adesi & Whaley (1987):

$$b_i^{(0)} = b_\infty + (K - b_\infty)\left(1 - e^{-\left[(r-q)i\Delta t - 2\sigma\sqrt{i\Delta t})K/(K-b_\infty)\right]}\right), \quad \forall i = 0, \ldots, N_T \tag{2.26}$$

where b_∞ represents the critical price of a perpetual American put option.[3]

Also, since we know that $S_c(0+) = K \min(1, r/q)$ (see e.g. Kim 1990), we fix $b_0^{(k)} = K \min(1, r/q)$. We do this for convenience since our method naturally converges to this limit if we set $b_0^{(k)} = K$ and let $\Delta t \to 0$. Moreover, the value of the integral appearing in (2.23) is the same whether we start from 0 or $0+$.

Given a set of estimates $b_i^{(k)}$ or all $i \in 1, 2, \ldots, N_T$ obtained after k iterations, we find a new set of estimates $b_i^{(k+1)}$ by using Equation (2.14):

$$b_i^{(k+1)} = K \frac{V_i(B^{(k)})}{U_i(B^{(k)})}, \quad \forall i = 1, 2, \ldots, N_T, \tag{2.27}$$

[3]The critical price of the perpetual American put option is equal to $b_\infty = \dfrac{K}{1 - 1/q_\infty^1}$, where $q_\infty^1 = -\dfrac{1}{2}(N-1) - \dfrac{1}{2}\sqrt{(N-1)^2 + 4M}$, $M = \dfrac{2r}{\sigma^2}$ and $N = \dfrac{2(r-q)}{\sigma^2}$.

where

$$U_i(B^{(k)}) = 1 - e^{-qi\Delta t}N(-d_1(b_i^{(k)}, K, i\Delta t)) - q\Delta t\sum_{j=1}^{i-1} e^{-qj\Delta t}N(-d_1(b_i^{(k)}, b_{i-j}^{(k)}, j\Delta t))$$
$$-\frac{q\Delta t}{2}\left[N(-d_1(b_i^{(k)}, b_i^{(k)}, 0)) + e^{-qi\Delta t}N(-d_1(b_i^{(k)}, b_0^{(k)}, i\Delta t))\right],$$
$$V_i(B^{(k)}) = 1 - e^{-ri\Delta t}N(-d_2(b_i^{(n)}, K, i\Delta t)) - r\Delta t\sum_{j=1}^{i-1} e^{-rj\Delta t}N(-d_2(b_i^{(k)}, b_{i-j}^{(k)}, j\Delta t))$$
$$-\frac{r\Delta t}{2}\left[N(-d_2(b_i^{(k)}, b_i^{(k)}, 0)) + e^{-ri\Delta t}N(-d_2(b_i^{(k)}, b_0^{(k)}, i\Delta t))\right],$$

$N(-d_1(b_i^{(k)}, b_i^{(k)}, 0)) = N(-d_2(b_i^{(k)}, b_i^{(k)}, 0)) = 0.5$, and $B^{(k)}$ is the vector containing the elements $b_i^{(k)}$ for $i = 1, \ldots, N_T$. We continue the process until:

$$\max_{i=1,\ldots,N_T}\left|\frac{b_i^{(k+1)} - b_i^{(k)}}{K}\right| \leq \varepsilon,$$

where ε is a relative tolerance threshold. The method usually converges very fast, taking for example between 5 to 6 iterations when $\varepsilon = 10^{-3}$.

Figure 1 presents an example of how the method works. In this example we use 20 time-intervals to approximate the early-exercise boundary. The strike price is $K = 100$, the time-to-maturity is $T = 1$, the risk-free rate is $r = 0.04$, the dividend rate is $q = 0.08$, and the constant volatility is $\sigma = 0.2$. In this example the method converges in 5 iterations.

Once an estimate of the early-exercise boundary $B_{N_T} = (\beta_0, \beta_1, \ldots, \beta_{N_T})$ is determined as in the previous example, we compute the premium of an American put option with spot S_0, strike K and maturity T using Equation (2.6). We follow Kallast & Kivinukk (2003) and use Simpson's rule to approximate the integral:

$$P_0 = p(S_0, K, T) + \frac{\Delta t}{3}\left[f(S_0, \beta_{N_T}, 0) + 4f(S_0, \beta_{N_T-1}, \Delta t) + 2f(S_0, \beta_{N_T-2}, 2\Delta t) + \cdots\right.$$
$$\left.\cdots + 2f(S_0, \beta_2, (N_T-2)\Delta t) + 4f(S_0, \beta_1, (N_T-1)\Delta t) + f(S_0, \beta_0, N_T\Delta t)\right], \quad (2.28)$$

where

$$f(x, y, \tau) = rKe^{-r\tau}N(-d_2(x, y, \tau)) - qxe^{-q\tau}N(-d_1(x, y, \tau)).$$

Note that we implicitly assume that N_T is even.

b) Numerical Quadrature

We also implement our method by estimating the integrals using a Gauss-Kronrod adaptive procedure. The iteration is adapted in a similar manner as it was explained for the trapezoidal rule.

We work with a discrete set of estimated points from the early-exercise curve denoted by $b_i^{(k)}$. In this implementation we follow Kim et al. (2013) and initialize the method only with the flat prior given in Equation (2.25). Given a set of estimates $b_i^{(k)}$ for $i = 1, 2, \ldots, N_T$ obtained after k iterations, we build a curve $B^{(k)}(\tau)$ for $\tau \in [0, T]$ by using a spline that goes through the points $b_i^{(k)}$ for all $i = 0, 1, \ldots, N_T$.

We find a new estimate $B^{(k+1)}$ for each point $b_i^{(k+1)}$ by using Equation (2.27) where U_i and V_i are replaced by:

$$U_{t_i}(B^{(k)}) = 1 - e^{-qt_i}N(-d_1(B^{(k)}(t_i), K, t_i)) - q\int_0^{t_i} e^{-q\tau}N(-d_1(B^{(k)}(t_i), B^{(k)}(t_i - \tau), \tau))\,\mathrm{d}\tau,$$

$$V_{t_i}(B^{(k)}) = 1 - e^{-rt_i}N(-d_2(B^{(k)}(t_i), K, t_i)) - r\int_0^{t_i} e^{-r\tau}N(-d_2(B^{(k)}(t_i), B^{(k)}(t_i - \tau), \tau))\,\mathrm{d}\tau,$$

and the integrals are estimated using the adaptive Gauss-Kronrod procedure. We continue the process until:

$$\max_{i=1,\ldots,N_T} \left| \frac{b_i^{(k+1)} - b_i^{(k)}}{K} \right| \leq \varepsilon,$$

where ε is a relative tolerance threshold.

Once an estimate of the early-exercise boundary $B(i)$ is determined for each $i = 1, 2, \ldots, N_T$, we build a curve $B(\tau)$ for $\tau \in [0, T]$ using splines, and compute the premium of an American put option with spot S_0, strike K and maturity T using Equations (2.23) and (2.24) into (2.6), and the adaptive Gauss-Kronrod procedure.

2.3.2 Numerical Results

In this section we present numerical tests that compare the pricing accuracy and the speed of our proposed method with other numerical techniques that have been studied in the literature. We implement the iteration in three different ways: using the trapezoidal rule with a flat initial guess [FPI-F], using the trapezoidal rule with the initial guess of Barone-Adesi & Whaley (1987) [FPI-BAW], and using the Gauss-Kronrod quadrature with splines and starting from a flat initial guess [FPI-GK].

We compare our results with standard methods that have been studied in the literature and whose abbreviated names are listed below. BAW denotes the quadratic approximation of Barone-Adesi & Whaley (1987). BIN is the binomial tree method of Cox et al. (1979). BIN-BS is the binomial model using Black & Scholes at the last time-step of Broadie & Detemple (1996). CARR is the six-point randomization method of Carr (1998). HSY is the six-point recursive integration method of Huang et al. (1996). IBN is the tree-point modified recursive integration method of Ibáñez (2003). JZ is the refined quadratic approximation of Ju & Zhong (1999). KJK is the iterative method of Kim et al. (2013). KK is the recursive method of Kallast & Kivinukk (2003). LS is the least-squares Monte Carlo approach of Longstaff & Schwartz (2001). LUBA is the lower and upper bound approximation method of Broadie & Detemple (1996). TRI the trinomial tree method of Boyle (1988). Numbers next to some methods specify the number of time-steps.

In order to assess the accuracy of all our computations, the true value is computed with a binomial tree model with 15,000 time steps in which the Black & Scholes formula is used at the penultimate node. Broadie & Detemple (1996) show that the traditional binomial method can have an oscillatory convergence whereas the binomial tree with Black

& Scholes at the end converges faster to the true price.[4] We use the root mean squared error (RMSE) and the relative RMSE[5] as the main measures of errors.

All codes were programmed in Matlab and all tests were performed using the same hardware. For every method we recalculate the early-exercise frontier when pricing each option. We do not take advantage of the fact that for some methods (such as ours) once the early-exercise frontier is known, it is possible to price options with different spot prices without having to recompute it. We do this to make all methods comparable.

a) Pricing Accuracy of Our Method

In this section we analyze the pricing accuracy of our methodology. Table I compares the pricing performance of FPI-F with the true American option price. In the table we fix the following parameters: strike price $K = 100$, time-to-maturity $T = 3$, and volatility $\sigma = 0.2$. We generate 12 different examples by using different spot prices (80, 100, and 120), interest rates (0.04 and 0.08) and dividend yields (0.04 and 0.12), as shown in the table. Column (1) reports the "true" value of the American put option calculated using a binomial tree with 15,000 steps in which the Black & Scholes formula is used at the last time-step (Broadie & Detemple 1996). Columns (2)$-$(8) report American put option prices calculated using our FPI-F method where the number of time-steps vary from 20 to 140 in increments of 20, while columns (9)$-$(11) report results using 200, 300 and 400 time-steps respectively.

The table shows that the method using a simple trapezoidal rule to estimate the integral requires a small number of time-steps to attain high precision, even when the maturity of the option is large as in the example. Using 20 time-steps we obtain a relative RMSE

[4] Kallast & Kivinukk (2003) use the binomial method with 10,000 steps as their benchmark. Broadie & Detemple (1996) use the convergent binomial method proposed by Amin & Khanna (1994) with 15,000 steps as their benchmark. We tried both but found that the binomial tree with Black & Scholes at the end presents the best convergence among all three of them.

[5] We define the relative RMSE as the root mean squared percentage errors.

less than 10^{-4}. With 60 time-steps we achieve a relative RMSE less than 10^{-5}. The last two columns show relative RMSEs less than 10^{-6}. We can also see that errors decrease monotonically as the number of time-steps increase.

b) Comparison with Other Methods

In this section we compare the performance of our proposed implementations with other methods that have been studied in the literature. We first compare all methods described at the beginning of this section across twelve combinations of the spot price (80, 100, and 120), time-to-maturity (0.5 and 3), and volatility (0.2 and 0.5). The following parameters are fixed throughout the evaluations: $K = 100$, $\delta = 0.04$, and $r = 0.04$.

Tables II and III report the results where the methods are sorted from left to right by decreasing relative RMSE (RRMSE). We find that BAW has the worst RRMSE. JZ attains a better accuracy than BAW since it is a refined version of the later, reaching close to a tenth of the RRMSE in BAW (Ju & Zhong 1999). LUBA has a RRMSE close to but better than BIN with 1000 time-steps, which is consistent with the accuracy reported in Broadie & Detemple (1996). CARR has around a third of the accuracy of LUBA, and around double of the accuracy of HSY, which is consistent with results reported in Ju (1998). We also find that IBN gains an extra decimal in accuracy over HSY, despite the fact that IBN uses half the points of HSY. This is consistent with the findings reported in Ibáñez (2003) who introduces the use of Richardson extrapolation applied to American option methods. Finally, our approach with only 60 time-steps achieves better accuracy than BAW, LS, JZ, HSY, CARR, BIN 500, TRI 500, BIN 1000, BIN BS 500, LUBA, BIN BS 1000, BIN 2500, IBN and TRI 2500. Also, our method FPI-F with 400 time-steps reaches the same accuracy as KK with the same number of time-steps, which is expected since both methods use the same equation to solve for the early-exercise boundary.

We also perform a comprehensive analysis in which we compare the pricing performance of each method using the following different combinations of parameters: spot price $S = 75, 80, ..., 120, 125$ (11 values); maturity $T = 1/12, 3/12, 6/12, 9/12, 1, 2, 3$;

volatility $\sigma = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6$; risk-free rate $r = 0.02, 0.04, 0.06, 0.08, 0.1$ and dividend yield $\delta = 0.0, 0.04, 0.08, 0.12$. This generates a set of 9,240 different combinations. Without loss of generality, the strike price is fixed at $K = 100$. In the analysis we only focus on American put options since the put-call symmetry identity of McDonald & Schroder (1998) suggests that similar results should be obtained for American call options. In unreported results, we verify that this is indeed the case.

We exclude from our tests options with prices of less than 50 cents and we do not include samples in which a certain method computes a negative early-exercise premia.[6] After applying these filters, we obtain a total of 7,865 sample points over which we test the accuracy and speed of all methods.

Table IV reports summary statistics of the empirical performance of each method. As a measure of accuracy we report the root mean squared error (RMSE). We also report the average time in seconds for each method to price the 7,865 options. To compare the speed-accuracy trade-off across different methods, we define a new measure of efficiency:

$$\text{Efficiency} = -\log(\text{RMSE} \times \text{Time}).$$

According to this definition, a method performs better the higher its Efficiency.

When compared using our measure of efficiency, the best method is FPI-F 60, followed by FPI-BAW 400, FPI-F 400, LUBA[7], and JZ (in that order). It is interesting to note that FPI-GK 24 and KJK 24 score much lower, even though they are also iterative

[6] Binomial and trinomial methods give in some cases negative early-exercise premia. This is probably due to the oscillatory convergence of these kind of methods. On the other hand, our method never computes negative early-exercise premia under the same set of parameters.

[7] We use the same parameters λ_1 and λ_2 as calibrated in Broadie & Detemple (1996). Even though these parameters were calibrated using a dividend rate $\delta = [0, ..., 0.1]$, we test LUBA under same constraints of δ and obtain similar RMSEs compared with our testing set that has in addition δ = 0.12.

methods. Also, KK 400 also scores lower, even though the method solves the same equation as FPI-F or FPI-BAW. Finally, it is surprising to note that the highly popular LS method is the second worst in terms of RMSE, and the worst in terms of speed.

Table IV also reports information on the dispersion of the pricing accuracy among the methods. Columns (4) to (6) report the percentage of options for which the absolute error (AE) is lower than the corresponding threshold. It is interesting to note that FPI-BAW 400 achieves the highest percentage of options priced with an absolute error less than 10^{-5}. It is followed by FPI-F 400, KK 400, FPI-GK 24, and KJK 24 (in that order).

We continue our analysis by plotting the speed-accuracy trade-off as shown in Figure 2. In the figure we measure accuracy as the root mean squared error (RMSE), and speed as the number of options priced per second. In the figure the axis is in $\log_{10}$-scale. Hence, the performance of a given method increases as we move towards the north-east.

Consistent with Table IV, Figure 2 shows that FPI-F and FPI-BAW outperform all other methods. We can also observe that for lower precision LUBA and BIN-BS are efficient. Also, note that FPI-BAW, FPI-F, KK, FPI-GK, BIN-BS, TRI and BIN converge monotonically in accuracy at the expense of a lower speed.

Figure 2 confirms the convergence of our methods in terms of pricing errors. As a general result, the more time-steps we use in the computation of the early-exercise frontier, the higher the accuracy we obtain in pricing the options. Furthermore, our approach seems to dominate in terms of speed-accuracy trade-off all other methods including KK, even though the later is based on solving the same equation. Our method is fast because the iterations are performed in parallel. Hence, the speed could be increased even further without sacrificing the accuracy by the use of modern hardware such as graphic processing units (GPUs). As is shown in the next section, this feature of our method is crucial when solving the same problem with stochastic volatility.

Finally, it is also worth noticing that methods such as BIN, TRI and BIN-BS, among others, need to recalculate all steps if one wishes to price an option with a different spot price. On the other hand, in the case of our method we could compute the early-exercise frontier up to a time-to-maturity T, say, for fixed r, δ, and σ. We could then use the same early-exercise boundary to price options with different spot and strike prices, increasing the speed even more. We choose not to exploit this natural ability of our method in order to make the benchmark comparable across different methods.

c) Richardson Extrapolation

Several authors in the literature have proposed the use of Richardson extrapolation to improve the pricing accuracy (see e.g. Geske & Johnson 1984, Bunch & Johnson 1992, Ibáñez 2003). Let P_i denote the option price obtained using N_i time-steps. The 2-point Richardson extrapolation (Bunch & Johnson 1992) is equal to $P = 2P_2 - P_1$, whereas the 3-point Richardson extrapolation (Geske & Johnson 1984) is obtained as $P = P_3 + 7/2P_2 - 1/2P_1$.

Richardson extrapolation works on methods on which convergence can be improved by using more time-steps. Therefore, the extrapolation cannot be used on methods such as LUBA, BAW and JZ, for example. We decide to check if Richardson extrapolation would improve the efficiency of our method and compare the results to two other methods that improve their efficiency as the number of time-steps increases, namely BIN-BS and KK.

Table V reports the results. We try five different combinations of 2 and 3-point Richardson extrapolation for BIN-BS, FPI-F and KK. Overall, FPI-F improves its efficiency considerably and in all five cases achieves a higher efficiency than BIN-BS and KK, with FPI-F 200/150/100 achieving the highest efficiency. The reason why we are able to improve the efficiency by using Richardson extrapolation is that the RMSE decreases linearly in that region with the number of time-steps (see Ibáñez 2003), as observed in Figure 2.

For BIN-BS, the 2-point Richardson extrapolation performs better than the 3-point Richardson Extrapolation in terms of efficiency, which is consistent with the results reported by Broadie & Detemple (1996). The 2-point Richardson extrapolation of BIN-BS shows a significant improvement over the regular BIN-BS. On the other hand, KK does not benefit from the use of Richardson extrapolation.

2.4 American Options Under The Heston Model

In this section we analyze the problem of pricing American options when the underlying asset follows a geometric Brownian motion with stochastic volatility as in Heston (1993):

$$\begin{aligned}\frac{\mathrm{d}S_t}{S_t} &= (r-q)\,\mathrm{d}t + \sqrt{v_t}\,\mathrm{d}W_{1,t},\\ \mathrm{d}v_t &= \kappa(\theta - v_t)\,\mathrm{d}t + \sigma\sqrt{v_t}\,\mathrm{d}W_{2,t},\end{aligned}$$

where $\mathrm{d}W_{1,t}\,\mathrm{d}W_{2,t} = \rho\,\mathrm{d}t$.

We first show how to solve for the early-exercise premium in the case of the American put option. This problem was previously analyzed by Chiarella & Ziogas (2005), where they solve for the early-exercise premium in Heston's (1993) model. Their approach uses McKean's (1965) incomplete Fourier transform method to derive a generalization of Jamshidian's (1992) representation of the free boundary value problem for American call options in the case of stochastic volatility.

In this paper we derive the early-exercise premium by inverting in closed-form the joint characteristic function of $\log(S_t)$ and v_t with respect to v_t. Therefore, our paper makes a technical contribution to the literature by providing a novel solution to this problem. Detailed proofs of all results can be found in Appendix C.

PROPOSITION 2. *The early-exercise premium for an American put option is given by:*

$$e_0 = \int_0^T rKe^{-r\tau}\,\mathbb{E}\left(\mathbf{1}_{\{S_\tau \le S_c(T-\tau,v_\tau)\}}\right) - qS_0e^{-q\tau}\,\mathbb{E}^*\left(\mathbf{1}_{\{S_\tau \le S_c(T-\tau,v_\tau)\}}\right)\,\mathrm{d}\tau, \qquad (2.29)$$

where

$$\mathbb{E}\left(\mathbf{1}_{\{S_\tau \le S_c(T-\tau,v_\tau)\}}\right) = \frac{1}{2} - \frac{1}{\pi}\int_{v=0}^{\infty}\int_{\phi=0}^{\infty} \mathrm{Re}\left(\frac{e^{i\phi\left(\log\left(\frac{S_0}{S_c(T-\tau,v)}\right)+(r-q)\tau\right)}G_\tau(\phi,v)}{i\phi}\right)\,\mathrm{d}\phi\,\mathrm{d}v,$$

$$\mathbb{E}^*\left(\mathbf{1}_{\{S_\tau \le S_c(T-\tau,v_\tau)\}}\right) = \frac{1}{2} - \frac{1}{\pi}\int_{v=0}^{\infty}\int_{\phi=0}^{\infty} \mathrm{Re}\left(\frac{e^{i\phi\left(\log\left(\frac{S_0}{S_c(T-\tau,v)}\right)+(r-q)\tau\right)}G_\tau(\phi-i,v)}{i\phi}\right)\,\mathrm{d}\phi\,\mathrm{d}v,$$

and $G_\tau(\phi,v)$ is defined in Equation (C.10) *in the Appendix.*

PROOF. See Result C.12 in Appendix C. □

The proposition shows that, given the early-exercise frontier $S_c(\cdot,\cdot)$, it is possible to compute analytically the early-exercise premium by evaluating a triple integral. We will use this result in Equation (2.14) to solve for the early-exercise frontier.

Before moving to the next result, it is important to understand the behavior of the early-exercise frontier at the boundary. It is well known that $S_c(T,v)$ satisfies the boundary condition $\lim_{T\to 0^+} S_c(T,v) = K\min(1,r/q)$ (see e.g. Kim 1990, Huang et al. 1996). However, as pointed out by Chockalingam & Muthuraman (2011), $\lim\limits_{v\to 0^+} S_c(T,v)$ is not always a trivial expression. In the Heston model, when $v_t = 0$ we have that $dv_t = \kappa\theta\mathrm{d}t$, making $v_{t+dt} > 0$. Hence, even when the variance is zero it is optimal to wait and not exercise as soon as the option is in the money. Our numerical procedure will allow us to compute this boundary accurately.

In Heston's (1993) model, it is well known how to compute the closed-form expression of the European put option. The following proposition reports this result as a reference.

PROPOSITION 3 (Heston 1993). *The European put premium is:*

$$p_0 = Ke^{-rT}\,\mathbb{E}\left(\mathbf{1}_{\{S_T\leq K\}}\right) - S_0e^{-qT}\,\mathbb{E}^*\left(\mathbf{1}_{\{S_T\leq K\}}\right), \tag{2.30}$$

where

$$\mathbb{E}\left(\mathbf{1}_{\{S_T\leq K\}}\right) = \frac{1}{2} - \frac{1}{\pi}\int_{\phi=0}^{\infty} \mathrm{Re}\left(\frac{e^{i\phi\left(\log\left(\frac{S_0}{K}\right)+(r-q)T\right)}H_T(\phi)}{i\phi}\right)\,\mathrm{d}\phi,$$

$$\mathbb{E}^*\left(\mathbf{1}_{\{S_T\leq K\}}\right) = \frac{1}{2} - \frac{1}{\pi}\int_{\phi=0}^{\infty} \mathrm{Re}\left(\frac{e^{i\phi\left(\log\left(\frac{S_0}{K}\right)+(r-q)T\right)}H_T(\phi-i)}{i\phi}\right)\,\mathrm{d}\phi,$$

and $H_T(\phi)$ is defined in Equation (C.13) *in the Appendix.*

PROOF. See Result C.14 in Appendix C. □

Finally, we provide expressions for the early-exercise premium and the European option premium for call options.

PROPOSITION 4. *The early-exercise premium for an American call option is given by:*

$$\breve{e}_0 = \int_0^T qS_0e^{-q\tau}\,\mathbb{E}^*\left(\mathbf{1}_{\{S_\tau\geq S_c(T-\tau,v_\tau)\}}\right) - rKe^{-r\tau}\,\mathbb{E}\left(\mathbf{1}_{\{S_\tau\geq S_c(T-\tau,v_\tau)\}}\right)\,\mathrm{d}\tau, \tag{2.31}$$

where

$$\mathbb{E}\left(\mathbf{1}_{\{S_\tau\geq S_c(T-\tau,v_\tau)\}}\right) = \frac{1}{2} + \frac{1}{\pi}\int_{v=0}^{\infty}\int_{\phi=0}^{\infty} \mathrm{Re}\left(\frac{e^{i\phi\left(\log\left(\frac{S_0}{S_c(T-\tau,v)}\right)+(r-q)\tau\right)}G_\tau(\phi,v)}{i\phi}\right)\,d\phi\,dv,$$

$$\mathbb{E}^*\left(\mathbf{1}_{\{S_\tau\geq S_c(T-\tau,v_\tau)\}}\right) = \frac{1}{2} + \frac{1}{\pi}\int_{v=0}^{\infty}\int_{\phi=0}^{\infty} \mathrm{Re}\left(\frac{e^{i\phi\left(\log\left(\frac{S_0}{S_c(T-\tau,v)}\right)+(r-q)\tau\right)}G_\tau(\phi-i,v)}{i\phi}\right)\,\mathrm{d}\phi\,\mathrm{d}v.$$

PROOF. See Result C.15 in Appendix C. □

PROPOSITION 5. *The European call premium is:*

$$c_0 = S_0e^{-qT}\,\mathbb{E}^*\left(\mathbf{1}_{\{S_T\geq K\}}\right) - Ke^{-rT}\,\mathbb{E}\left(\mathbf{1}_{\{S_T\geq K\}}\right), \tag{2.32}$$

where

$$\mathbb{E}\left(\mathbf{1}_{\{S_T \geq K\}}\right) = \frac{1}{2} + \frac{1}{\pi}\int_{\phi=0}^{\infty} \mathrm{Re}\left(\frac{e^{i\phi\left(\log\left(\frac{S_0}{K}\right)+(r-q)T\right)}H_T(\phi)}{i\phi}\right)\,\mathrm{d}\phi,$$

$$\mathbb{E}^*\left(\mathbf{1}_{\{S_T \geq K\}}\right) = \frac{1}{2} + \frac{1}{\pi}\int_{\phi=0}^{\infty} \mathrm{Re}\left(\frac{e^{i\phi\left(\log\left(\frac{S_0}{K}\right)+(r-q)T\right)}H_T(\phi-i)}{i\phi}\right)\,\mathrm{d}\phi.$$

PROOF. See Result C.16 in Appendix C. □

2.4.1 Numerical Implementation

We calculate the price of an American put option in three steps. First, we start with an initial guess of the early-exercise boundary. In particular, the use of the Heston model implies that the boundary is a surface dependent on the time-to-maturity and the current variance.

In the numerical implementation, we truncate the domain of the variance to $[0, V]$, where $V = 1$, consistent with the probability density function obtained for the Heston model being almost zero and numerical methods present in the literature. Second, the initial guess is updated using the FPI algorithm established by Equation (2.14) in combination with Heston's put price given by the European and early-exercise premiums established in Propositions 2 and 3. We iterate and update our early-exercise policy until the maximum difference among two estimated curves is below a specified tolerance, giving us a reasonable estimation of the early-exercise surface boundary. Finally, the price of the American option is obtained by using the estimated early-exercise boundary along with Propositions 2 and 3 in Equation (2.6).

To calculate the integrals present in these equations, we use Matlab's integral computation algorithm based on the work done by Shampine (2008) and Shampine (2010) on vectorized adaptive quadrature and the Gauss-Kronrod recursive quadrature procedure. In particular, the algorithm uses a 7-point Gauss rule with a 15-point Kronrod rule. As a way to improve the accuracy of our calculations, the final step of our procedure is improved by

doing a smooth spline interpolation on the already calculated optimal exercise surface. By doing this, we increase its discretization without sacrificing execution time in order to do a finer integration in Propositions 2 and 3.

2.4.2 Numerical Results

In this section, we present numerical results that compare the pricing accuracy and speed of our FPI method with other numerical techniques that have been studied in the literature and applied to the Heston model. We implement the FPI starting from a flat initial guess in a uniform (τ, v) grid and using the aforementioned Matlab's integral algorithm [FPI-CPU]. In order to assess the benefits of combining our proposed FPI procedure and Matlab's vectorized integration method with parallel computing (something that is straightforward considering the parallelizable nature of both algorithms), we also implement the most computing intensive part of the algorithm on a Graphical Processing Unit (GPU) through the CUDA programming language [FPI-GPU].

We compare our results with standard methods that have been studied before and whose abbreviated names are listed below. PSOR denotes the Projected Succesive Over-relaxation Method of Cryer (1971), and CS stands for Componentwise Splitting Method of Ikonen & Toivanen (2007*a*). Both methods were implemented using a non-uniform grid as proposed in Ikonen & Toivanen (2007*b*). TP is the Transformation Procedure of Chockalingam & Muthuraman (2011) implemented on a uniform grid. We also use the FPI method as the initial step for other algorithms requiring the critical price S_c as an input. This is the case of TP, where we implement a two-step scheme (which we denote TP-FPI-CPU and TP-FPI-GPU) to assess the benefits of using one numerical method (TP) to do the option pricing using the early-exercise surface obtained through the FPI algorithm.

The comparisons were carried out using Matlab implementations on a two-core 2.00 GHz Intel Core i7 with 8.00 GB RAM for all methods, and additional CUDA implementations on a Nvidia GTX Titan GPU (2,688 CUDA cores and a 837 MHz base clock) for

the FPI-GPU and TP-FPI-GPU procedures. For each algorithm we recalculate the early-exercise frontier when pricing each option. We do not take advantage of the fact that for some methods (such as ours), once the early-exercise border is known, it is possible to price options with different spot prices and volatilities without having to recompute it. The objective of this is to make all studied methods comparable.

As a way to be consistent and to evaluate the accuracy of all our computations, we use as true values the ones listed in Ikonen & Toivanen (2007*b*), which were obtained by using the CS method with a very fine grid. In the referred paper, the authors compare their calculated true values to other prices reported in the literature to validate their precision. The parameters used in our analysis are: strike price K = 10, time-to-maturity T = 3 months, risk-free rate of interest r = 0.1 per annum, mean rate of reversion $\kappa = 5$, long-term mean variance $\theta = 0.16$, volatility of the volatility process $\sigma = 0.9$, and correlation $\rho = 0.1$ We use the root mean squared error (RMSE) as the main measure of error and accuracy.

a) Pricing Accuracy of Our Method

In this section we analyze the pricing accuracy and efficiency of our procedure. Tables VI and VII compare the pricing performance of our iterative method FPI with the true American Heston put option price and the performance of the other methods introduced before. We generate 10 prices by using different initial stock prices ($S_0 = 8, 9, 10, 11,$ and 12) and initial variances ($v_0 = 0.0625$ and $v_0 = 0.25$) for various grid sizes. Column (2) reports the grid size used in each method. For PSOR, CS and TP, grid size refers to (n_{S_0}, n_{v_0}, n_T), as these methods calculate put option prices for each initial stock price S_0 and variance v_0 at a defined time-to-maturity T, whereas for the FPI method we refer to the early-exercise policy grid size defined by (n_τ, n_v), as this method calculates an option price given S_0 and v_0 by solving the optimal exercise policy $S_c(\tau, v)$. Also, as a way of assessing the precision of our method depending on the grid size of S_c after using spline interpolation to calculate the early-exercise premium of Proposition 2, we report

the number of discretization points for τ and v on the policy grid used in the last step of FPI in Column (3). Columns (4)–(8) report American put option prices, while columns (9) and (10) report the average runtimes and RMSEs achieved in each price computation for the various grid sizes. Tables VIII and IX report performance statistics for the combined TP-FPI method. In this case, grid size refers to (n_{S_0}, n_{v_0}, n_T) and (n_τ, n_v) grids, as this combined method calculates put option prices for each initial stock price S_0 and variance v_0 at a defined time-to-maturity $T = 0.25$ using the TP option pricing procedure and having the optimal early-exercise policy $S_c(\tau, v)$ calculated using the initial steps of our FPI algorithm.

Figures 3 and 4 plot the relationship between root-mean-square errors (RMSE) and runtimes for the three benchmark methods (PSOR, CS and TP), different variations of our method (FPI-CPU, FPI-GPU), and our hybrid procedure combining the optimal exercise price surface obtained through FPI and the TP algorithm for option pricing calculation (TP-FPI-CPU, TP-FPI-GPU). Each of the graphs were plotted for $v_0 = 0.0625$ and $v_0 = 0.25$, respectively.

As can be observed, the performance-gap between PSOR and CS is consistent with the findings reported in Ikonen & Toivanen (2007*b*). Our implementation of the TP method (under the same hardware and software conditions as CS and PSOR) shows that TP performs worst compared to the other benchmarks. This is explained by the fact that the speed and accuracy of this method, as stated by Chockalingam & Muthuraman (2011), is highly dependent on the choice of the numerical procedure used to solve for the fixed-boundary problem (in our case, we use Matlab and its GMRES method). However, it is important to point out that TP benefits from its easiness of implementation and adaptability to other fixed-boundary PDE solvers.

For our proposed method and different variations, several findings can be highlighted. Our results show that the FPI method's performance on a CPU lies between PSOR and CS, where the flexibility in choosing the time-variance grid's precision allows FPI to achieve

significant accuracy. We note however, that FPI-CPU 40×10 has almost the same accuracy as FPI-CPU 40×20 which is nearly 3 times slower. Also, both figures reveal that the performance slope of our CPU methods is capped by the exercise policy grid calculation once these grids are sufficiently large (40x10, 40x20, 20x20). For smaller grids (5x5, 10x10, 20x5), slopes are driven by the calculation of the early-exercise premium integral.

In the case of FPI implemented on the GPU, we obtain important benefits as we eliminate the exercise policy calculation time cap as restriction for runtimes (allowing us to use finer grids without losing much calculation time) and we reduce the time employed for the early-exercise premium integral, proving the relevance of our algorithm's parallel nature and adaptability, and the importance of parallel computing in complex models. As we can see in the figures, FPI-GPU's performance excels all the other implemented methods in most part of the RMSE spectrum. If we compare our method with CS, we can see that we achieve similar errors for much less computational runtimes. However, both methods tend to converge in performance for very small errors (10^{-4} and 10^{-3} neighborhood). The results show the trend exhibited in the CPU implementation, where a finer time-discretization is more relevant than the variance-discretization. (FPI-GPU $X \times 10$ is the best performer).

The analysis of our hybrid TP-FPI method also highlights some interesting findings. TP-FPI 5×5 is the worst performer as its RMSE is capped to more than 10^{-2}, evidencing a lack of convergence. This shows that a 5×5 FPI exercise policy surface does not couple well with the TP, revealing large computation errors. For larger FPI grid sizes, we can observe convergence in the TP-FPI method both for the CPU and GPU implementations, where the latter clearly surpasses the clean TP method when more accuracy is required. Another interesting thing to note is that for less accurate results, the runtimes for this method (both CPU and GPU) are driven by the calculation of the optimal exercise surface, whereas after a certain RMSE threshold, results are driven by the TP algorithm and the fixed-boundary solver used (Matlab GMRES for our implementations).

b) Optimal Early Exercise Policy Accuracy of Our Method

Figures 5, 6, 7, and 8 plot the exercise policies calculated by FPI and TP using both coarse and fine grid sizes. The parameters used to elaborate these surfaces were the ones defined in Section 2.4.2. It is interesting to note how both iterative algorithms converge to the same optimal exercise policy through different mathematical procedures. As we can see, and mostly explained by the discrete nature of the algorithm, the estimated early-exercise surface obtained by TP shows non-smooth characteristics that are reduced once the grid size is increased, allowing for convergence in the pricing accuracy of the method. On the other hand, FPI is capable of generating a much smoother early-exercise surface compared to TP even with coarser grids. As a result, this procedure is capable of generating efficient option pricing with smaller grids compared to TP and the other methods.

2.5 Concluding Remarks

In this paper we propose a simple, fast and accurate iterative algorithm for pricing American options, and solve for its early-exercise boundary. Our algorithm is based on a fixed-point iteration derived from the early-exercise representation for American options, and is equivalent to a multivariate Newton iteration that converges globally. The method is stable, robust, and well suited for parallel implementations. We test empirically our algorithm using the Black and Scholes (1973) and Heston (1993) stochastic volatility models, and find that our methodology outperforms existing approaches in terms of efficiency and accuracy when computing the early-exercise frontier.

Our results show that the trapezoidal rule in the Black & Scholes model achieves the best performance among all existing methods analyzed in the paper. Performance can be improved even further by using a smart guess for the initial early-exercise boundary as in Barone-Adesi & Whaley (1987). We find that our FPI converges monotonically to the true American option price as we increase the number of time-steps in the approximation,

regardless of the method employed to estimate the integrals. Finally, we find that the performance of our method can be improved further by the use of Richardson extrapolation.

Furthermore, the results for the Heston implementation reveal that the FPI remains competitive in terms of speed and accuracy on its CPU implementation compared to other computational procedures. However, our procedure generates smoother early-exercise boundaries compared to other methods such as the TP of Chockalingam & Muthuraman (2011) that can in turn be used as an input for their method.

Our analysis shows that a direct solution of the early-exercise representations of Kim (1990) and Rutkowski (1994) is robust and efficient. This point was already raised by Kallast & Kivinukk (2003) in the case of GBM, and Chiarella & Ziogas (2005) in the case of stochastic volatility, by solving these equations sequentially. Our results confirm that the method of Kallast & Kivinukk (2003) and the one we propose share the same Newton-type convergence property. On the other hand, by solving the early-exercise boundary equation as a FPI as in Kim et al. (2013) we can accelerate the process through the parallelization of iterations, and simplify its implementation by avoiding the computation on complex derivatives.

In summary, our work shows that the FPI is superior to all commonly used algorithms to price American options under the Black & Scholes model, and to other computational methods used to price American options under the Heston model. Additionally, the calculation of the critical price frontier by our procedure could be used to develop new numerical methodologies that incorporate the FPI as part of a combined procedure as shown with the TP-FPI. The algorithm seems promising for solving early exercise boundaries and option prices of more sophisticated models considering the significant improvements in performance this method can achieve through GPU parallel calculation, and the inherent flexibility of our procedure.

REFERENCES

Amin, K. & Khanna, A. (1994), 'Convergence of American option values from discrete- to continuous-time financial models', *Mathematical Finance* **4**(4), 289–304.

Amos, D. E. (1985), A subroutine package for bessel functions of a complex argument and nonnegative order, Technical report, Sandia National Laboratories.

Amos, D. E. (1986), 'Algorithm 644: A portable package for bessel functions of a complex argument and nonnegative order', *ACM Transactions on Mathematical Software (TOMS)* **12**(3), 265–273.

Andersen, L. & Broadie, M. (2004), 'Primal-dual simulation algorithm for pricing multidimensional american options', *Management Science* **9**(50), 1222–1234.

Barone-Adesi, G. (2005), 'The saga of the American put', *Journal of Banking and Finance* **29**, 2909–2918.

Barone-Adesi, G. & Whaley, R. E. (1987), 'Efficient analytic approximation of American option values', *Journal of Finance* **42**(2), 301–320.

Black, F. & Scholes, G. (1973), 'The pricing of options and corporate liabilities', *Journal of Political Economy* **81**(3), 637–659.

Boyle, P. P. (1988), 'A lattice framework for option pricing with two state variables', *Journal of Financial and Quantitative Analysis* **23**(3), 1–12.

Brandt, A. & Cryer, C. (1983), 'Multigrid algorithms for the solution of linear complementarity problems arising from free boundary problems', *SIAM Journal on Scientific and Statistical Computing* **4**(4), 655–684.

Brennan, M. J. & Schwartz, E. S. (1977), 'The valuation of American put options', *Journal of Finance* **32**(2), 449–462.

Broadie, M. & Detemple, J. (1996), 'American option valuation: New bounds, approximations, and a comparison of existing methods', *Review of Financial Studies* **9**(4), 1211–1250.

Broadie, M. & Detemple, J. (2004), 'Option pricing: Valuation models and applications', *Management Science* **50**(9), 1145–1177.

Broadie, M., Detemple, J., Ghysels, E. & Torés, O. (2000), 'American options with stochastic dividends and volatility', *Journal of Econometrics* **94**(1–2), 53–92.

Broadie, M. & Glasserman, P. (2004), 'A stochastic mesh method for pricing high-dimensional american options', *Journal of Computational Finance* **4**(7), 35–72.

Bunch, D. S. & Johnson, H. (1992), 'A simple and numerically efficient valuation method for American puts using a modified Geske-Johnson approach', *Journal of Finance* **47**(2), 809–816.

Carr, P. (1998), 'Randomization and the American put', *Review of Financial Studies* **11**(3), 597–626.

Carr, P., Jarrow, R. & Myneni, R. (1992), 'Alternative characterizations of American put options', *Mathematical Finance* **2**(2), 87–106.

Chiarella, C. & Ziogas, A. (2005), Pricing american options under stochastic volatility, Computing in Economics and Finance 77, Society for Computational Economics.

Chockalingam, A. & Muthuraman, K. (2011), 'American options under stochastic volatility', *Operations research* **59**(4), 793–809.

Clarke, N. & Parrot, K. (1999), 'Multigrid for american option pricing with stochastic volatility', *Applied Mathematical Finance* **6**(3), 177–195.

Cox, J. C., Ross, S. A. & Rubinstein, M. (1979), 'Option pricing: A simplified approach', *Journal of Financial Economics* **7**(1), 229–263.

Cryer, C. W. (1971), 'The solution of a quadratic programming problem using systematic overelaxation', *SIAM Journal on Control* **9**(3), 385–392.

Duffie, D., Pan, J. & Singleton, K. (2000), 'Transform analysis and asset pricing for affine jump-diffusions', *Econometrica* **68**(6), 1343–1376.

Flannery, B. P., Press, W. H., Teukolsky, S. A. & Vetterling, W. (1992), *Numerical recipes in C*, Press Syndicate of the University of Cambridge, New York.

Geske, R. & Johnson, H. E. (1984), 'The American put option valued analytically', *Journal of Finance* **39**(5), 1511–1524.

Heston, S. L. (1993), 'A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options', *Review of Financial Studies* **6**(2), 327–343.

Huang, J.-Z., Subrahmanyam, M. G. & Yu, G. G. (1996), 'Pricing and hedging American options: A recursive integration method', *Review of Financial Studies* **9**(1), 277–300.

Hull, J. & White, A. (1987), 'The pricing of options on assets with stochastic volatilities', *Journal of Finance* **2**(42), 281–300.

Ibáñez, A. (2003), 'Robust pricing of the American put option: A note on Richardson extrapolation and the early exercise premium', *Management Science* **49**(9), 1210–1228.

Ikonen, S. & Toivanen, J. (2004), 'Operator splitting methods for american option pricing', *Applied Mathematics Letters* **17**(7), 809–814.

Ikonen, S. & Toivanen, J. (2007*a*), 'Componentwise splitting methods for pricing american options under stochastic volatility', *International Journal of Theoretical and Applied Finance* **10**(2), 331–361.

Ikonen, S. & Toivanen, J. (2007*b*), 'Efficient numerical methods for pricing american options under stochastic volatility', *Numerical Methods for Partial Differential Equations* **24**(1), 104–126.

Jacka, S. D. (1991), 'Optimal stopping and the American put', *Mathematical Finance* **1**(2), 1–14.

Jamshidian, F. (1992), 'An analysis of american options', *Review of Futures Markets* **11**(1), 72–80.

Johnson, N. L., Kotz, S. & Balakrishnan, N. (1995), *Contunuous Univariate Distributions*, Vol. 2, second edn, John Wiley & Sons.

Ju, N. (1998), 'Pricing an American option by approximating its early exercise boundary as a piece-wise exponential', *Review of Financial Studies* **11**(3), 627–46.

Ju, N. & Zhong, R. (1999), 'An approximate formula for pricing American options', *Journal of Derivatives* **7**(2), 31–40.

Kallast, S. & Kivinukk, A. (2003), 'Pricing and hedging American options using approximations by Kim integral equations', *European Finance Review* **7**(3), 361–383.

Kim, I. J. (1990), 'The analytic valuation of American options', *Review of Financial Studies* **3**(4), 547–572.

Kim, I. J., Jang, B.-G. & Kim, K. T. (2013), 'A simple iterative method for the valuation of American options', *Quantitative Finance* **13**(6), 885–895.

Little, T., Pant, V. & Hou, C. (2000), 'A new integral representation of the early exercise boundary for American put options', *Journal of Computational Finance* **3**, 73–96.

Longstaff, F. A. & Schwartz, E. S. (2001), 'Valuing American options by simulation: A simple least-squares approach', *Review of Financial Studies* **14**(1), 113–147.

McDonald, R. L. & Schroder, M. D. (1998), 'A parity result for American options', *Journal of Computational Finance* **1**(3), 5–13.

McKean, H. P. (1965), 'A free-boundary problem for the heat-equation arising from a problem of mathematical economics', *IMR-Industrial Management Review* **6**(2), 32–39.

Medina, L. (2013), A new iterative method for pricing american option, Dissertation, Pontificia Universidad Católica de Chile, Santiago de Chile.

Olver, F. W. J., Lozier, D. W., Boisvert, R. F. & Clark, C. W., eds (2010), *NIST Handbook of Mathematical Functions*, Cambridge University Press.

Osterlee, C. (2003), 'On multigrid for linear complementarity problems with application to american-style options', *Electronic Transactions on Numerical Analysis* (15), 165–185.

Protter, P. (2005), *Stochastic Integration and Differential Equations*, 2nd edn, Springer-Verlag, Heidelberg.

Rutkowski, M. (1994), 'The early exercise premium representation of foreign market American options', *Mathematical Finance* **4**(4), 313–325.

Schroder, M. (1999), 'Changes of numeraire for pricing futures, forwards, and options', *Review of Financial Studies* **12**(5), 1143–1163.

Scott, L. (1987), 'Option pricing when the variance changes randomly: Theory, estimation and an application', *Journal of Financial and Quantitative Analysis* **22**(4), 419–438.

Shampine, L. (2008), 'Vectorized adaptive quadrature in matlab', *Journal of Computational and Applied Mathematics* **211**(2), 131–140.

Shampine, L. F. (2010), 'Weighted quadrature by change of variable', *Neural, Parallel and Scientific Computations* **18**(2), 195–206.

Stein, E. & Stein, J. (1991), 'Stock price distributions with stochastic volatility: An analytic approach', *Rev. Financial Studies* **4**(4), 727–752.

Sullivan, M. A. (2000), 'Valuing American put options using Gaussian quadrature', *Review of Financial Studies* **13**(1), 75–94.

Thompson, I. & Barnett, A. (1987), 'Modified bessel functions iv (z) and kv (z) of real order and complex argument, to selected accuracy', *Computer Physics Communication* **47**(2), 245–257.

Zvan, R., Forsyth, P. & Vetzal, K. (1998), 'Penalty methods for american options with stochastic volatility', *Journal of Computational and Applied Mathematics* **2**(91), 199–218.

APPENDIX

A. EXISTING METHODS TO PRICE AMERICAN OPTIONS UNDER GBM

There are numerous studies in the literature that propose methods to price American options in the case of GBM.[1] Brennan & Schwartz (1977) were the first to solve numerically a partial differential equation (PDE) to price American options. Another popular method that discretizes the time space and the asset price is the binomial method of Cox et al. (1979). Both methods are still widely used because of their simplicity.

Following the same ideas, other researchers have tried to improve the lattice approach in order to increase the accuracy and/or reduce the computation time, such as the trinomial method of Boyle (1988) and the improved binomial method presented in Broadie & Detemple (1996). Longstaff & Schwartz (2001) developed a novel method to value options by simulation that determines the conditional expected payoff by least-squares. Although all of these methods are flexible and easily adapted to many kinds of options, their main drawback is that they are time consuming.

A different approach to improve the speed at the expense of precision are the so-called quasi-analytical approximation methods. One of the first of such methods is the quadratic approximation of Barone-Adesi & Whaley (1987). The idea of the method is to solve an approximate version of the PDE governing the price of the American option that yields a closed-form solution. Ju & Zhong (1999) refine the derivation of the quadratic approximation of Barone-Adesi & Whaley (1987) by making a similar approximation to the PDE. Even though Ju & Zhong (1999) appears to be more accurate than Barone-Adesi & Whaley (1987), a main drawback of both methods is that the approximation works well for very short and very long maturity options, but present difficulties when applied to medium-term maturity options. Additionally, these methods do not converge to the true value so the estimation cannot be made arbitrarily small. Broadie & Detemple (1996) develop a method along these lines based on a lower and an upper bound. The lower-bound price is computed from a closed-form solution of a capped option, while the upper-bound

[1] Barone-Adesi (2005) presents a comprehensive survey of existing methods to price American options.

price is based on the integral equations of Kim (1990). The price is finally obtained as a weighted average between the lower and upper-bound prices, where the weights are estimated as a function of model parameters.

In the literature there are also methods that use Richardson extrapolation in order to improve the accuracy of the computations. For example, Geske & Johnson (1984) find an exact representation of an American put and introduce the Richardson extrapolation to the pricing problem. Huang et al. (1996) approximate the American option as a Bermudan option. Carr (1998) randomizes the time-to-maturity of the American option and introduces a feasible distribution in order to find a simple solution. Ju (1998) prices the option approximating the early-exercise boundary as a multipiece exponential function. Ibáñez (2003) refines the recursive method of Huang et al. (1996) by making the Bermudan option monotonically convergent to the true American option as the number of exercise times increases.

Finally, other methods use quadrature formulas in order to price the option. Sullivan (2000) approximates the early-exercise premium by using Gaussian quadrature. Kallast & Kivinukk (2003) use the trapezoidal rule in order to approximate the integral part of the equation of Kim (1990).

B. AUXILIARY RESULTS

LEMMA B.1.

$$\mathbb{E}\left(e^{-rT}X_T + \int_0^T e^{-r\tau} q X_\tau \,\mathrm{d}\tau\right) = 1.$$

PROOF. An asset that pays continuously a dividend of $qX_\tau \,\mathrm{d}\tau$ at times $\tau \in [0,T]$ and X_T at time T, can be replicated by buying the asset at time $t = 0$ for its value $X_0 = 1$. Therefore:

$$\mathbb{E}\left(e^{-rT}X_T + \int_0^T e^{-r\tau} q X_\tau \,\mathrm{d}\tau\right) = X_0 = 1,$$

which completes the proof. □

LEMMA B.2. *Let* $\phi\colon [0,\infty) \times D \to (0,\infty)$. *Then:*

$$1 - e^{-rT}\,\mathbb{E}\left(\mathbf{1}_{\{X_T \le \frac{K}{\phi(T,Y_0)}\}}\right) - \int_0^T r e^{-r\tau}\,\mathbb{E}\left(\mathbf{1}_{\{X_\tau \le \frac{\phi(T-\tau,Y_\tau)}{\phi(T,Y_0)}\}}\right) \mathrm{d}\tau > 0,$$

and

$$1 - e^{-rT}\,\mathbb{E}\left(X_T\mathbf{1}_{\{X_T \le \frac{K}{\phi(T,Y_0)}\}}\right) - \int_0^T q e^{-r\tau}\,\mathbb{E}\left(X_\tau\mathbf{1}_{\{X_\tau \le \frac{\phi(T-\tau,Y_\tau)}{\phi(T,Y_0)}\}}\right) \mathrm{d}\tau > 0.$$

PROOF. We have that:

$$\begin{aligned} 1 - e^{-rT}\,\mathbb{E}\left(\mathbf{1}_{\{X_T \le \frac{K}{\phi(T,Y_0)}\}}\right) - \int_0^T r e^{-r\tau}\,\mathbb{E}\left(\mathbf{1}_{\{X_\tau \le \frac{\phi(T-\tau,Y_\tau)}{\phi(T,Y_0)}\}}\right) \mathrm{d}\tau \\ > 1 - e^{-rT} - \int_0^T r e^{-r\tau}\,\mathrm{d}\tau = 0, \end{aligned}$$

and by Lemma B.1

$$\begin{aligned} 1 - e^{-rT}\,\mathbb{E}\left(X_T\mathbf{1}_{\{X_T \le \frac{K}{\phi(T,Y_0)}\}}\right) - \int_0^T q e^{-r\tau}\,\mathbb{E}\left(X_\tau\mathbf{1}_{\{X_\tau \le \frac{\phi(T-\tau,Y_\tau)}{\phi(T,Y_0)}\}}\right) \mathrm{d}\tau \\ > 1 - \mathbb{E}\left(e^{-rT}X_T - \int_0^T e^{-r\tau} q X_\tau \,\mathrm{d}\tau\right) = 0. \end{aligned}$$

□

LEMMA B.3. *Let*

$$U_{T,Y_0}(S_c) = 1 - \mathbb{E}\left(e^{-rT} X_T \mathbf{1}_{\{X_T \leq \frac{K}{S_c(T,Y_0)}\}}\right) - \int_0^T q e^{-r\tau}\, \mathbb{E}\left(X_\tau \mathbf{1}_{\{X_\tau \leq \frac{S_c(T-\tau,Y_\tau)}{S_c(T,Y_0)}\}}\right).$$

Then

$$\frac{\partial U_{T,Y_0}}{\partial S_c(T,Y_0)} > 0.$$

PROOF. We note that as $S_c(T,Y_0)$ increases, the ratios $\frac{K}{S_c(T,Y_0)}$ and $\frac{S_c(T-\tau,Y_\tau)}{S_c(T,Y_0)}$ decrease, and also the measures of the events $\left\{X_T \leq \frac{K}{S_c(T,Y_0)}\right\}$ and $\left\{X_\tau \leq \frac{S_c(T-\tau,Y_\tau)}{S_c(T,Y_0)}\right\}$. Hence, both expectations in the above expression decrease (since $X_t > 0$) as $S_c(T,Y_0)$ increases, making $U_{T,Y_0}(S_c)$ to increase. □

C. EARLY EXERCISE REPRESENTATION IN THE HESTON MODEL

Definition C.1. Let $x_\tau = \log(S_\tau)$, and denote by $f_\tau(x, v)$ the joint probability density function of x_τ and v_τ conditional on x_0 and v_0. The characteristic function $\hat{f}_\tau(\phi, \xi)$ of the joint distribution of x_τ and v_τ is defined as:

$$\hat{f}_\tau(\phi, \xi) = \mathbb{E}\left(e^{i(\phi x_\tau + \xi v_\tau)}\right) = \int_{v=0}^{\infty}\int_{x=-\infty}^{\infty} e^{i(\phi x + \xi v)} f_\tau(x, v)\,\mathrm{d}x\,\mathrm{d}v.$$

LEMMA C.2.

$$\hat{f}_\tau(\phi, \xi) = \exp(i\phi(x_0 + (r-q)\tau) + \delta(v_0 + \kappa\theta\tau)) \times \left(1 - \frac{\sigma^2}{2\gamma}(e^{\gamma\tau} - 1)(i\xi - \delta)\right)^{-\frac{2\kappa\theta}{\sigma^2}} \exp\left(\frac{v_0 e^{\gamma\tau}(i\xi - \delta)}{1 - \frac{\sigma^2}{2\gamma}(e^{\gamma\tau} - 1)(i\xi - \delta)}\right), \quad \text{(C.1)}$$

where $\gamma, \delta \in \mathbb{C}$ *are defined as:*

$$\gamma = \sqrt{\kappa^2 + (1-\rho^2)\sigma^2\phi^2 + i(\sigma - 2\kappa\rho)\sigma\phi}, \quad \text{(C.2)}$$

$$\delta = \frac{\kappa + \gamma - i\rho\sigma\phi}{\sigma^2}. \quad \text{(C.3)}$$

PROOF. Since $S_t = e^{x_t}$, we have that:

$$\mathrm{d}x_t = (r - q - 1/2v_t)\,\mathrm{d}t + \sqrt{v_t}\,\mathrm{d}W_{1,t},$$
$$\mathrm{d}v_t = \kappa(\theta - v_t)\,\mathrm{d}t + \sigma\sqrt{v_t}\,\mathrm{d}W_{2,t}.$$

Let $g(x_t, v_t, t) = \mathbb{E}_{x_t, v_t}\left(e^{i(\phi x_\tau + \xi v_\tau)}\right)$. Since $g(x_t, v_t, t)$ is a martingale, it must be the case that $\mathbb{E}_{x_t, v_t}(\mathrm{d}g(x_t, v_t, t)) = 0$, implying that:

$$\frac{\partial g}{\partial x_t}(r - q - 1/2v_t) + \frac{\partial g}{\partial v_t}\kappa(\theta - v_t) + \frac{1}{2}\frac{\partial^2 g}{\partial x_t^2}v_t + \frac{1}{2}\frac{\partial^2 g}{\partial v_t^2}\sigma^2 v_t + \frac{\partial^2 g}{\partial x_t \partial v_t}\rho\sigma v_t + \frac{\partial g}{\partial t} = 0.$$

Let $T = \tau - t$ and assume that:

$$g(x_t, v_t, t) = e^{A(T) + B(T)x_t + C(T)v_t}. \quad \text{(C.4)}$$

We obtain that:

$$(r-q-1/2v_t)B+\kappa(\theta-v_t)C+\frac{1}{2}v_tB^2+\frac{1}{2}\sigma^2v_tC^2+\rho\sigma v_tBC-(A'+B'x_t+C'v_t)=0,$$

which implies that:

$$\begin{aligned}
A'(T) &= (r-q)B(T)+\kappa\theta C(T),\\
B'(T) &= 0,\\
C'(T) &= -\frac{1}{2}B(T)-\kappa C(T)+\frac{1}{2}B(T)^2+\frac{1}{2}\sigma^2C(T)^2+\rho\sigma B(T)C(T),
\end{aligned}$$

subject to the boundary conditions $A(0)=0$, $B(0)=i\phi$, and $C(0)=i\xi$. It can then be verified that:

$$\begin{aligned}
A(T) &= i(r-q)\phi T+\kappa\theta\left(\delta T-\frac{2}{\sigma^2}\log\left(1-\frac{\sigma^2}{2\gamma}(e^{\gamma T}-1)(i\xi-\delta)\right)\right),\\
B(T) &= i\phi,\\
C(T) &= \delta+\frac{e^{\gamma T}(i\xi-\delta)}{1-\frac{\sigma^2}{2\gamma}(e^{\gamma T}-1)(i\xi-\delta)},
\end{aligned}$$

is a solution for the previous system of differential equations, where:

$$\begin{aligned}
\gamma &= \sqrt{\kappa^2+(1-\rho^2)\sigma^2\phi^2+i(\sigma-2\kappa\rho)\sigma\phi},\\
\delta &= \frac{\kappa+\gamma-i\rho\sigma\phi}{\sigma^2}.
\end{aligned}$$

By inserting the above expressions for $A(T)$, $B(T)$, and $C(T)$ into (C.4), factorizing, and noting that $\hat{f}_\tau(\phi,\xi)=g(x_0,v_0,0)$, we obtain the desired result. □

LEMMA C.3. *Let $f(v)\xrightarrow{\mathcal{F}}\hat{f}(\xi)$ denote the Fourier transformation of $f(v)$ into $\hat{f}(\xi)$ defined as $\hat{f}(\xi)=\int_{-\infty}^{\infty}e^{i\xi v}f(v)\,\mathrm{d}v$. Then given parameters $\nu\in\mathbb{R}$ and $\alpha,\beta,\lambda\in\mathbb{C}$, we*

have that:

$$\frac{1}{\beta}\exp\left(\left(\alpha-\frac{1}{\beta}\right)v-\frac{\lambda}{\beta}\right)\left(\frac{v}{\lambda}\right)^{\frac{\nu-1}{2}}I_{\nu-1}\left(2\sqrt{\frac{\lambda v}{\beta^2}}\right)$$
$$\xrightarrow{\mathcal{F}}\left(\frac{1}{1-\beta(i\xi+\alpha)}\right)^{\nu}\exp\left(\frac{\lambda(i\xi+\alpha)}{1-\beta(i\xi+\alpha)}\right), \quad \text{(C.5)}$$

where $I_\nu(\cdot)$ denotes the modified Bessel function of first kind of order ν.

PROOF. For $\nu\in\mathbb{R}$ and $z\in\mathbb{C}$ we have that $I_\nu(\cdot)$ can be written as:

$$I_\nu(z)=\sum_{j=0}^{+\infty}\frac{1}{j!\Gamma(j+\nu+1)}\left(\frac{z}{2}\right)^{2j+\nu}.$$

Hence for $v\in\mathbb{R}^+$ and $\hat{\lambda}\in\mathbb{C}$ we have that:

$$\frac{1}{2}e^{-\frac{v+\hat{\lambda}}{2}}\left(\frac{v}{\hat{\lambda}}\right)^{\frac{\nu-1}{2}}I_{\nu-1}\left(\sqrt{\hat{\lambda}v}\right)=\frac{1}{2}e^{-\frac{v+\hat{\lambda}}{2}}\left(\frac{v}{\hat{\lambda}}\right)^{\frac{\nu-1}{2}}\sum_{j=0}^{+\infty}\frac{1}{j!\Gamma(j+\nu)}\left(\frac{\sqrt{\hat{\lambda}v}}{2}\right)^{2j+\nu-1},$$
$$=e^{-\lambda/2}\sum_{j=0}^{\infty}\frac{(\hat{\lambda}/2)^j}{j!}\frac{e^{-v/2}v^{j+\nu-1}}{2^{j+\nu}\Gamma(j+\nu)}.$$

We also have that:

$$\begin{aligned}\int_0^\infty e^{i\xi v}\frac{e^{-v/2}v^{j+\nu-1}}{2^{j+\nu}\Gamma(j+\nu)}\mathrm{d}v&=\frac{1}{2^{j+\nu}\Gamma(j+\nu)}\int_0^\infty e^{-1/2(1-2i\xi)v}v^{j+\nu-1}\mathrm{d}v,\\&=\frac{1}{2^{j+\nu}\Gamma(j+\nu)}\frac{2^{j+\nu-1}}{(1-2i\xi)^{j+\nu-1}}\frac{2}{1-2i\xi}\int_0^\infty e^{-y}y^{j+\nu-1}\mathrm{d}y,\\&=\frac{1}{(1-2i\xi)^{j+\nu}},\end{aligned}$$

where in the second line we made the change of variables $y=\frac{1}{2}(1-2i\xi)v$ and used the fact that $\Gamma(j+\nu)=\int_0^\infty e^{-y}y^{j+\nu-1}\mathrm{d}y$.

Therefore, we can conclude that:

$$\frac{1}{2}\exp\left(-\frac{v+\hat{\lambda}}{2}\right)\left(\frac{v}{\hat{\lambda}}\right)^{\frac{\nu-1}{2}} I_{\nu-1}\left(\sqrt{\hat{\lambda}v}\right)$$
$$\xrightarrow{\mathcal{F}} e^{-\lambda/2}\sum_{j=0}^{\infty}\frac{(\hat{\lambda}/2)^j}{j!}\frac{1}{(1-2i\xi)^{j+\nu}} = \left(\frac{1}{1-2i\xi}\right)^{\nu}\exp\left(\frac{i\hat{\lambda}\xi}{1-2i\xi}\right). \quad \text{(C.6)}$$

In the case in which $\hat{\lambda} \in \mathbb{R}$ and $\nu \in \mathbb{N}$, the left-hand side in Equation (C.6) represents the probability density function of a non-central χ^2-distribution with 2ν degrees of freedom and non-centrality parameter $\hat{\lambda}$, whereas the right-hand side represents its characteristic function. Therefore, Equation (C.6) is an extension of the well-known relationship for non-central χ^2 distributions to the more general case in which $\nu \in \mathbb{R}$ and $\hat{\lambda} \in \mathbb{C}$ (see e.g. Johnson et al. 1995, Chapter 29) .

Now, given $\alpha, c, d \in \mathbb{C}$, we also have that:

$$\frac{1}{2}\exp((\alpha+c)v+d)\exp\left(-\frac{v+\hat{\lambda}}{2}\right)\left(\frac{v}{\hat{\lambda}}\right)^{\frac{\nu-1}{2}} I_{\nu-1}\left(\sqrt{\hat{\lambda}v}\right)$$
$$\xrightarrow{\mathcal{F}} \left(\frac{1}{1-2i(\xi+(\alpha+c)/i)}\right)^{\nu}\exp\left(\frac{i\hat{\lambda}(\xi+(\alpha+c)/i)}{1-2i(\xi+(\alpha+c)/i)}+d\right).$$

If we insert $\hat{\lambda} = \frac{4\lambda}{\beta^2}$, $c = \frac{1}{2} - \frac{1}{\beta}$, and $d = -\frac{2\lambda}{\beta}c$ into the above expression, we find that:

$$\frac{1}{2}\left(\frac{\beta}{2}\right)^{\nu-1}\exp\left(\left(\alpha+\frac{1}{\beta}\right)v-\frac{\lambda}{\beta}\right)\left(\frac{v}{\lambda}\right)^{\frac{\nu-1}{2}} I_{\nu-1}\left(2\sqrt{\frac{\lambda v}{\beta^2}}\right)$$
$$\xrightarrow{\mathcal{F}} \left(\frac{\beta/2}{1-\beta(i\xi+\alpha)}\right)^{\nu}\exp\left(\frac{\lambda(i\xi+\alpha)}{1-\beta(i\xi+\alpha)}\right),$$

which implies:

$$\frac{1}{\beta}\exp\left(\left(\alpha+\frac{1}{\beta}\right)v-\frac{\lambda}{\beta}\right)\left(\frac{v}{\lambda}\right)^{\frac{\nu-1}{2}} I_{\nu-1}\left(2\sqrt{\frac{\lambda v}{\beta^2}}\right)$$
$$\xrightarrow{\mathcal{F}} \left(\frac{1}{1-\beta(i\xi+\alpha)}\right)^{\nu}\exp\left(\frac{\lambda(i\xi+\alpha)}{1-\beta(i\xi+\alpha)}\right).$$

This completes the proof. □

Definition C.4. Let

$$\bar{f}_\tau(\phi,v)=\int_{-\infty}^{\infty} e^{i\phi x} f_\tau(x,v)\,\mathrm{d}x$$

denote the Fourier transform of $f_\tau(x,v)$ with respect to x.

Remark C.5. Note that $\bar{f}_\tau(0,v)$ is the marginal density of v_τ conditional on x_0 and v_0. Furthermore, we also have that

$$\bar{f}_\tau(\phi,v)=\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\xi v}\hat{f}_\tau(\phi,\xi)\,\mathrm{d}\xi,$$

i.e. $\bar{f}_\tau(\phi,v)$, as a function of v, is the inverse Fourier transform of $\hat{f}_\tau(\phi,\xi)$, as a function of ξ.

LEMMA C.6.

$$\bar{f}_\tau(\phi,v)=\frac{1}{\zeta}\exp\left(i\phi(x_0+(r-q)\tau)+\left(\delta\kappa\theta-\frac{\gamma(\nu+1)}{2}\right)\tau\right.$$
$$\left.+\delta(v_0-v)-\frac{1}{\zeta}(ve^{-\gamma\tau}+v_0)+\frac{\nu-1}{2}\log\left(\frac{v}{v_0}\right)\right)\times I_{\nu-1}\left(2\sqrt{\frac{v_0 v e^{-\gamma\tau}}{\zeta^2}}\right), \quad \text{(C.7)}$$

where γ is defined as in (C.2)*, δ is defined as in* (C.3)*,* $\zeta=\dfrac{\sigma^2}{2\gamma}(1-e^{-\gamma\tau})$*, and* $\nu=\dfrac{2\kappa\theta}{\sigma^2}$.

PROOF. By defining $\beta = \frac{\sigma^2}{2\gamma}(e^{\gamma\tau} - 1)$, $\lambda = v_0 e^{\gamma\tau}$, $\alpha = -\delta$ and $\nu = \frac{2\kappa\theta}{\sigma^2}$, we can re-write Equation (C.1) as:

$$\hat{f}_\tau(\phi, \xi) = \exp(i\phi(x_0 + (r-q)\tau) + \delta(v_0 + \kappa\theta\tau)) \times (1 - \beta(i\xi + \alpha))^{-\nu} \exp\left(\frac{\lambda(i\xi - \delta)}{1 - \beta(i\xi + \alpha)}\right). \quad \text{(C.8)}$$

Since the bottom half of (C.8) correspond to the right-hand side of (C.5), it must be the case that:

$$\bar{f}_\tau(\phi, v) = \exp(i\phi(x_0 + (r-q)\tau) + \delta(v_0 + \kappa\theta\tau)) \times \frac{1}{\beta} \exp\left(\left(\alpha - \frac{1}{\beta}\right) v - \frac{\lambda}{\beta}\right) \left(\frac{v}{\lambda}\right)^{\frac{\nu-1}{2}} I_{\nu-1}\left(2\sqrt{\frac{\lambda v}{\beta^2}}\right).$$

By inserting $\beta = e^{\gamma\tau}\zeta$, $\alpha = -\delta$, and $\lambda = v_0 e^{\gamma\tau}$ into the above expression we obtain (C.7). □

Remark C.7. Note that (C.7) is expressed in terms of $e^{-\gamma\tau}$ instead of $e^{\gamma\tau}$ as in (C.1). This is necessary because the function written like this does not overflow.

LEMMA C.8.

$$\mathbb{E}\left(\mathbf{1}_{\{S_\tau \leq S_c(T-\tau, v_\tau)\}}\right) = \frac{1}{2} - \frac{1}{\pi}\int_{v=0}^{\infty}\int_{\phi=0}^{\infty} \mathrm{Re}\left(\frac{e^{i\phi\left(\log\left(\frac{S_0}{S_c(T-\tau,v)}\right) + (r-q)\tau\right)} G_\tau(\phi, v)}{i\phi}\right) \mathrm{d}\phi\, \mathrm{d}v, \quad \text{(C.9)}$$

where

$$G_\tau(\phi, v) = \frac{1}{\zeta} \exp\left(\left(\delta\kappa\theta - \frac{\gamma(\nu+1)}{2}\right)\tau + \delta(v_0 - v) - \frac{1}{\zeta}(ve^{-\gamma\tau} + v_0) + \frac{\nu-1}{2}\log\left(\frac{v}{v_0}\right)\right) \times I_{\nu-1}\left(2\sqrt{\frac{v_0 v e^{-\gamma\tau}}{\zeta^2}}\right),$$

γ is defined as in (C.2)*, δ is defined as in* (C.3)*, $\zeta = \dfrac{\sigma^2}{2\gamma}(1 - e^{-\gamma\tau})$, and $\nu = \dfrac{2\kappa\theta}{\sigma^2}$.*

PROOF. Let $x_c(T-\tau, v) = \log(S_c(T-\tau, v))$. Then we have:

$$\begin{aligned}
\mathbb{E}\left(\mathbf{1}_{\{S_\tau \geq S_c(T-\tau,v_\tau)\}}\right) &= \mathbb{E}\left(\mathbf{1}_{\{x_\tau \geq x_c(T-\tau,v_\tau)\}}\right), \\
&= \int_{v=0}^{\infty}\int_{x=-\infty}^{\infty} \mathbf{1}_{\{x \geq x_c(T-\tau,v)\}} f_\tau(x,v)\,\mathrm{d}x\,\mathrm{d}v, \\
&= \int_{v=0}^{\infty}\int_{x=-\infty}^{\infty} \mathbf{1}_{\{x \geq x_c(T-\tau,v)\}} \frac{1}{2\pi}\int_{\phi=-\infty}^{\infty} e^{-i\phi x}\bar{f}_\tau(\phi,v)\,\mathrm{d}\phi\,\mathrm{d}x\,\mathrm{d}v, \\
&= \int_{v=0}^{\infty}\frac{1}{2\pi}\int_{\phi=-\infty}^{\infty}\left(\int_{x=-\infty}^{\infty} \mathbf{1}_{\{x \geq x_c(T-\tau,v)\}} e^{-i\phi x}\,\mathrm{d}x\right)\bar{f}_\tau(\phi,v)\,\mathrm{d}\phi\,\mathrm{d}v, \\
&= \int_{v=0}^{\infty}\frac{1}{2\pi}\int_{\phi=-\infty}^{\infty} e^{-i\phi x_c(T-\tau,v)}\pi\left(\frac{1}{i\pi\phi} + \delta(\phi)\right)\bar{f}_\tau(\phi,v)\,\mathrm{d}\phi\,\mathrm{d}v, \\
&= \int_{v=0}^{\infty}\frac{1}{2\pi}\left(\pi\bar{f}_\tau(0,v) + \int_{\phi=-\infty}^{\infty}\frac{e^{-i\phi x_c(T-\tau,v)}\bar{f}_\tau(\phi,v)}{i\phi}\,\mathrm{d}\phi\right)\mathrm{d}v, \\
&= \frac{1}{2} + \frac{1}{\pi}\int_{v=0}^{\infty}\int_{\phi=0}^{\infty}\mathrm{Re}\left(\frac{e^{-i\phi x_c(T-\tau,v)}\bar{f}_\tau(\phi,v)}{i\phi}\right)\mathrm{d}\phi\,\mathrm{d}v,
\end{aligned}$$

where in the fourth line $\delta(\cdot)$ denotes the Dirac delta function. Since we also have that:

$$\mathbb{E}\left(\mathbf{1}_{\{S_\tau \leq S_c(T-\tau,v_\tau)\}}\right) = 1 - \mathbb{E}\left(\mathbf{1}_{\{S_\tau \geq S_c(T-\tau,v_\tau)\}}\right),$$

we can conclude that:

$$\mathbb{E}\left(\mathbf{1}_{\{S_\tau \leq S_c(T-\tau,v_\tau)\}}\right) = \frac{1}{2} - \frac{1}{\pi}\int_{v=0}^{\infty}\int_{\phi=0}^{\infty}\mathrm{Re}\left(\frac{e^{-i\phi x_c(T-\tau,v)}\bar{f}_\tau(\phi,v)}{i\phi}\right)d\phi\,dv.$$

Re-factorizing the term inside the real-part of above expression yields the desired result.

□

Remark C.9. In order to highlight how v_0 affects $G_\tau(\phi, v)$, in the following we write $G_\tau(\phi, v; v_0)$ for $G_\tau(\phi, v)$. We note that $G_\tau(\phi, v; v_0)$ is not well defined when $v_0 = 0$. The

limit at $v_0 = 0$, however, exists and is given by:

$$\lim_{v_0 \to 0} G_\tau(\phi, v; v_0) = \frac{1}{\zeta} e^{(\delta\kappa\theta - \gamma\nu)\tau + \delta(v_0 - v) - \frac{1}{\zeta}(ve^{-\gamma\tau} + v_0)} v^{\nu-1} \left(\frac{1}{\zeta^2}\right)^{\frac{\nu-1}{2}} \frac{1}{\Gamma(\nu)},$$

where $\Gamma(\nu)$ denotes the gamma function of order ν. Event though the integrals in (C.9) are well defined, in numerical applications it is convenient to evaluate $G_\tau(\phi, v; v_0)$ when $v_0 = 0$. Hence, for the remainder of this Appendix we extend the function $G_\tau(\phi, v; v_0)$ by continuity as:

$$G_\tau(\phi, v; v_0) = \begin{cases} \frac{1}{\zeta} e^{(\delta\kappa\theta - \gamma\nu)\tau + \delta(v_0 - v) - \frac{1}{\zeta}(ve^{-\gamma\tau} + v_0)} v^{\nu-1} \left(\frac{1}{\zeta^2}\right)^{\frac{\nu-1}{2}} \frac{1}{\Gamma(\nu)} & \textit{if} \quad v_0 = 0, \\ \frac{1}{\zeta} e^{\left(\delta\kappa\theta - \frac{\gamma(\nu+1)}{2}\right)\tau + \delta(v_0 - v) - \frac{1}{\zeta}(ve^{-\gamma\tau} + v_0)} \left(\frac{v}{v_0}\right)^{\frac{\nu-1}{2}} I_{\nu-1}\left(2\sqrt{\frac{v_0 v e^{-\gamma\tau}}{\zeta^2}}\right) & \textit{if} \quad v_0 > 0. \end{cases} \tag{C.10}$$

PROOF.

$$\begin{aligned} \lim_{v_0 \to 0} G_\tau(\phi, v; v_0) &= \frac{1}{\zeta} e^{\left(\delta\kappa\theta - \frac{\gamma(\nu+1)}{2}\right)\tau + \delta(v_0 - v) - \frac{1}{\zeta}(ve^{-\gamma\tau} + v_0)} \left(\frac{v}{v_0}\right)^{\frac{\nu-1}{2}} I_{\nu-1}\left(2\sqrt{\frac{v_0 v e^{-\gamma\tau}}{\zeta^2}}\right), \\ &= \frac{1}{\zeta} e^{\left(\delta\kappa\theta - \frac{\gamma(\nu+1)}{2}\right)\tau + \delta(v_0 - v) - \frac{1}{\zeta}(ve^{-\gamma\tau} + v_0)} \left(\frac{v}{v_0}\right)^{\frac{\nu-1}{2}} \left(\frac{v_0 v e^{-\gamma\tau}}{\zeta^2}\right)^{\frac{\nu-1}{2}} \frac{1}{\Gamma(\nu)}, \\ &= \frac{1}{\zeta} e^{\left(\delta\kappa\theta - \frac{\gamma(\nu+1)}{2}\right)\tau + \delta(v_0 - v) - \frac{1}{\zeta}(ve^{-\gamma\tau} + v_0)} v^{\nu-1} \left(\frac{e^{-\gamma\tau}}{\zeta^2}\right)^{\frac{\nu-1}{2}} \frac{1}{\Gamma(\nu)}, \\ &= \frac{1}{\zeta} e^{\left(\delta\kappa\theta - \frac{\gamma(\nu+1)}{2} - \frac{\gamma(\nu-1)}{2}\right)\tau + \delta(v_0 - v) - \frac{1}{\zeta}(ve^{-\gamma\tau} + v_0)} v^{\nu-1} \left(\frac{1}{\zeta^2}\right)^{\frac{\nu-1}{2}} \frac{1}{\Gamma(\nu)}, \\ &= \frac{1}{\zeta} e^{(\delta\kappa\theta - \gamma\nu)\tau + \delta(v_0 - v) - \frac{1}{\zeta}(ve^{-\gamma\tau} + v_0)} v^{\nu-1} \left(\frac{1}{\zeta^2}\right)^{\frac{\nu-1}{2}} \frac{1}{\Gamma(\nu)}, \end{aligned}$$

where we have used the fact that $I_\nu(z) \sim \frac{1}{\Gamma(\nu+1)} \left(\frac{1}{2}z\right)^\nu$ when $z \to 0$. □

Remark C.10. Furthermore, we also have that:

$$\lim_{v \to 0} G_\tau(\phi, v) = \begin{cases} 0 & \textit{if} \quad \nu > 1, \\ \infty & \textit{if} \quad \nu \leq 1. \end{cases}$$

In other words, the density function of v_τ given v_0 explodes at $v_\tau = 0$ when the Feller condition for the variance is violated.

LEMMA C.11.

$$\mathbb{E}^*\left(\mathbf{1}_{\{S_\tau \le S_c(v_\tau, T-\tau)\}}\right) = \frac{1}{2} - \frac{1}{\pi}\int_{v=0}^{\infty}\int_{\phi=0}^{\infty} \mathrm{Re}\left(\frac{e^{i\phi\left(\log\left(\frac{S_0}{S_c(T-\tau,v)}\right)+(r-q)\tau\right)}G_\tau(\phi - i, v)}{i\phi}\right)\,\mathrm{d}\phi\,\mathrm{d}v, \tag{C.11}$$

where $\mathbb{E}^*(\cdot)$ denotes the expectation with respect to the measure $\mathcal{Q}^*$ defined by its Radon-Nikodym derivative as:

$$\frac{\mathrm{d}\mathcal{Q}^*}{\mathrm{d}\mathcal{Q}} = \frac{S_\tau}{S_0 e^{(r-q)\tau}},$$

and $G_\tau(\phi, v; v_0)$ is defined in (C.10)*.*

PROOF. Let $f^*_\tau(x, v)$ be the probability density function (conditional on x_0 and v_0) of x_τ and v_τ under $\mathcal{Q}^*$, and $\hat{f}^*_\tau(\phi, \xi) = \mathbb{E}^*(e^{i\phi x_\tau + i\xi v_\tau})$. Then we have that:

$$\begin{aligned}
\hat{f}^*_\tau(\phi, \xi) &= \mathbb{E}\left(e^{i\phi x_\tau + i\xi v_\tau}\frac{\mathrm{d}\mathcal{Q}^*_\tau}{\mathrm{d}\mathcal{Q}}\right), \\
&= \mathbb{E}\left(e^{i\phi x_\tau + i\xi v_\tau}\frac{S_\tau}{S_0 e^{(r-q)\tau}}\right), \\
&= \frac{1}{S_0 e^{(r-q)\tau}}\,\mathbb{E}\left(e^{i\phi x_\tau + x_\tau + i\xi v_\tau}\right), \\
&= \frac{1}{S_0 e^{(r-q)\tau}}\,\mathbb{E}\left(e^{i(\phi - i)x_\tau + i\xi v_\tau}\right), \\
&= \frac{1}{S_0 e^{(r-q)\tau}}\hat{f}_\tau(\phi - i, \xi).
\end{aligned}$$

Also, define

$$\bar{f}^*_\tau(\phi, v) = \int_{-\infty}^{\infty} e^{i\phi x} f^*_\tau(x, v)\,\mathrm{d}x = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\xi v}\hat{f}^*_\tau(\phi, \xi)\,\mathrm{d}\xi,$$

which implies that:

$$\bar{f}^*_\tau(\phi, v) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\xi v}\hat{f}_\tau(\phi - i, \xi)\,\mathrm{d}\xi = \frac{1}{S_0 e^{(r-q)\tau}}\bar{f}_\tau(\phi - i, v).$$

Moreover, if we follow the same steps as in Lemma C.8 we find that:

$$\begin{aligned}
\mathbb{E}^*\left(\mathbf{1}_{\{S_\tau \leq S_c(v_\tau, T-\tau)\}}\right) &= \frac{1}{2} - \frac{1}{\pi}\int_{v=0}^{\infty}\int_{\phi=0}^{\infty} \mathrm{Re}\left(\frac{e^{-i\phi x_c(v,T-\tau)}\bar{f}^*_\tau(\phi, v)}{i\phi}\right) \mathrm{d}\phi\, \mathrm{d}v, \\
&= \frac{1}{2} - \frac{1}{\pi}\int_{v=0}^{\infty}\int_{\phi=0}^{\infty} \frac{1}{S_0 e^{(r-q)\tau}} \mathrm{Re}\left(\frac{e^{-i\phi x_c(v,T-\tau)}\bar{f}_\tau(\phi - i, v)}{i\phi}\right) \mathrm{d}\phi\, \mathrm{d}v, \\
&= \frac{1}{2} - \frac{1}{\pi}\int_{v=0}^{\infty}\int_{\phi=0}^{\infty} \mathrm{Re}\left(\frac{e^{i\phi(\log\left(\frac{S_0}{S_c(T-\tau,v)}\right)+(r-q)\tau)} G_\tau(\phi - i, v)}{i\phi}\right) \mathrm{d}\phi\, \mathrm{d}v,
\end{aligned}$$

which completes the proof. □

Proof C.12 (Proof of Proposition 2)**.** In the Heston (1993) model, Equation (2.7) becomes:

$$e_0 = \int_0^T rKe^{-r\tau}\,\mathbb{E}\left(\mathbf{1}_{\{S_\tau \leq S_c(T-\tau, v_\tau)\}}\right) - qS_0 e^{-q\tau}\,\mathbb{E}\left(\frac{S_\tau}{S_0 e^{(r-q)\tau}}\mathbf{1}_{\{S_\tau \leq S_c(T-\tau, v_\tau)\}}\right) \mathrm{d}\tau.$$

Applying the change of measure introduced in Lemma C.11 allows us to write:

$$\mathbb{E}\left(\frac{S_\tau}{S_0 e^{(r-q)\tau}}\mathbf{1}_{\{S_\tau \leq S_c(T-\tau, v_\tau)\}}\right) = \mathbb{E}^*\left(\mathbf{1}_{\{S_\tau \leq S_c(v_\tau, T-\tau)\}}\right),$$

implying that:

$$e_0 = \int_0^T rKe^{-r\tau}\,\mathbb{E}\left(\mathbf{1}_{\{S_\tau \leq S_c(T-\tau, v_\tau)\}}\right) - qS_0 e^{-q\tau}\,\mathbb{E}^*\left(\mathbf{1}_{\{S_\tau \leq S_c(T-\tau, v_\tau)\}}\right) \mathrm{d}\tau.$$

By inserting (C.9) and (C.11) into the above expression we obtain the desired result.

Remark C.13. Note that we can rewrite (C.1) as:

$$\begin{aligned}
\hat{f}_\tau(\phi, \xi) &= e^{i\phi(x_0+(r-q)\tau)+\delta(v_0+\kappa\theta\tau)-\nu\log\left(1-\frac{\sigma^2}{2\gamma}(e^{\gamma\tau}-1)(i\xi-\delta)\right)+\left(\frac{v_0 e^{\gamma\tau}(i\xi-\delta)}{1-\frac{\sigma^2}{2\gamma}(e^{\gamma\tau}-1)(i\xi-\delta)}\right)} \\
&= e^{i\phi(x_0+(r-q)\tau)+\delta\kappa\theta\tau-\nu\log\left(1-\frac{\sigma^2}{2\gamma}(e^{\gamma\tau}-1)(i\xi-\delta)\right)+\left(\delta+\frac{e^{\gamma\tau}(i\xi-\delta)}{1-\frac{\sigma^2}{2\gamma}(e^{\gamma\tau}-1)(i\xi-\delta)}\right)v_0} \\
&= e^{i\phi(x_0+(r-q)\tau)+(\delta\kappa\theta-\gamma\nu)\tau-\nu\log\left(e^{-\gamma\tau}-\frac{\sigma^2}{2\gamma}(1-e^{-\gamma\tau})(i\xi-\delta)\right)+\left(\delta+\frac{(i\xi-\delta)}{e^{-\gamma\tau}-\frac{\sigma^2}{2\gamma}(1-e^{-\gamma\tau})(i\xi-\delta)}\right)v_0} \\
&= e^{i\phi(x_0+(r-q)\tau)} h_\tau(\phi, \xi),
\end{aligned}$$

where

$$h_\tau(\phi,\xi) = e^{(\delta\kappa\theta-\gamma\nu)\tau-\nu\log\left(e^{-\gamma\tau}-\frac{\sigma^2}{2\gamma}(1-e^{-\gamma\tau})(i\xi-\delta)\right)+\left(\delta+\frac{(i\xi-\delta)}{e^{-\gamma\tau}-\frac{\sigma^2}{2\gamma}(1-e^{-\gamma\tau})(i\xi-\delta)}\right)v_0}. \quad \text{(C.12)}$$

We note that expressing (C.1) in terms of $e^{-\gamma\tau}$ instead of $e^{\gamma\tau}$ is necessary in numerical applications for the function not to overflow.

Proof C.14 (Proof of Proposition 3)**.** The European put Equation (2.5) becomes

$$p_0 = Ke^{-rT}\,\mathbb{E}\left(\mathbf{1}_{\{S_T\le K\}}\right) - S_0e^{-qT}\,\mathbb{E}^*\left(\mathbf{1}_{\{S_T\le K\}}\right).$$

Proceeding as before, we obtain that:

$$\mathbb{E}\left(\mathbf{1}_{\{S_T\le K\}}\right) = \frac{1}{2} - \frac{1}{\pi}\int_{\phi=0}^{\infty}\mathrm{Re}\left(\frac{e^{-i\phi\log(K)}\hat{f}_T(\phi,0)}{i\phi}\right)\mathrm{d}\phi,$$

$$\mathbb{E}^*\left(\mathbf{1}_{\{S_T\le K\}}\right) = \frac{1}{2} - \frac{1}{\pi}\int_{\phi=0}^{\infty}\frac{1}{S_0e^{(r-q)T}}\mathrm{Re}\left(\frac{e^{-i\phi\log(K)}\hat{f}_T(\phi-i,0)}{i\phi}\right)\mathrm{d}\phi,$$

where $\hat{f}_T(\phi,\xi)$ is defined in (C.1). Hence, we can re-write the above expectations as:

$$\mathbb{E}\left(\mathbf{1}_{\{S_T\le K\}}\right) = \frac{1}{2} - \frac{1}{\pi}\int_{\phi=0}^{\infty}\mathrm{Re}\left(\frac{e^{i\phi\left(\log\left(\frac{S_0}{K}\right)+(r-q)T\right)}H_T(\phi)}{i\phi}\right)\mathrm{d}\phi,$$

$$\mathbb{E}^*\left(\mathbf{1}_{\{S_T\le K\}}\right) = \frac{1}{2} - \frac{1}{\pi}\int_{\phi=0}^{\infty}\mathrm{Re}\left(\frac{e^{i\phi\left(\log\left(\frac{S_0}{K}\right)+(r-q)T\right)}H_T(\phi-i)}{i\phi}\right)\mathrm{d}\phi,$$

where

$$H_T(\phi) = h_T(\phi,0), \quad \text{(C.13)}$$

and $h_T(\phi,\xi)$ is defined in (C.12).

Note that we could have also used (Duffie et al. 2000, Equation 2.12):

$$\mathbb{E}\left(e^{-rT}\mathbf{1}_{\{S_T\leq K\}}\right) = \frac{\psi_T(0,0)}{2} - \frac{1}{\pi}\int_{\phi=0}^{\infty} \mathrm{Re}\left(\frac{e^{-i\phi\log(K)}\psi_T(\phi,0)}{i\phi}\right)\,\mathrm{d}\phi,$$

$$\mathbb{E}\left(e^{-rT}S_T\mathbf{1}_{\{S_T\leq K\}}\right) = \frac{\psi_T(-i,0)}{2} - \frac{1}{\pi}\int_{\phi=0}^{\infty} \mathrm{Re}\left(\frac{e^{-i\phi\log(K)}\psi_T(\phi-i,0)}{i\phi}\right)\,\mathrm{d}\phi,$$

where $\psi_T(\phi,\xi) = e^{-rT}\hat{f}_T(\phi,\xi)$. Re-factorizing and dividing the second equation by S_0e^{-qT} yields the desired result.

Proof C.15 (Proof of Proposition 4)**.** Since

$$e_0 = \int_0^T qS_0e^{-q\tau}\,\mathbb{E}^*\left(\mathbf{1}_{\{S_\tau\geq S_c(T-\tau,v_\tau)\}}\right) - rKe^{-r\tau}\,\mathbb{E}\left(\mathbf{1}_{\{S_\tau\geq S_c(T-\tau,v_\tau)\}}\right)\,\mathrm{d}\tau,$$

the results follows from Proposition 3 by noting that

$$\mathbb{E}\left(\mathbf{1}_{\{S_\tau\geq S_c(T-\tau,v_\tau)\}}\right) = 1 - \mathbb{E}\left(\mathbf{1}_{\{S_\tau\leq S_c(T-\tau,v_\tau)\}}\right),$$
$$\mathbb{E}^*\left(\mathbf{1}_{\{S_\tau\geq S_c(T-\tau,v_\tau)\}}\right) = 1 - \mathbb{E}^*\left(\mathbf{1}_{\{S_\tau\leq S_c(T-\tau,v_\tau)\}}\right).$$

Proof C.16 (Proof of Proposition 5)**.** Since

$$c_0 = S_0e^{-qT}\,\mathbb{E}^*\left(\mathbf{1}_{\{S_T\geq K\}}\right) - Ke^{-rT}\,\mathbb{E}\left(\mathbf{1}_{\{S_T\geq K\}}\right),$$

the results follows from Proposition 3 by noting that

$$\mathbb{E}\left(\mathbf{1}_{\{S_\tau\geq K\}}\right) = 1 - \mathbb{E}\left(\mathbf{1}_{\{S_\tau\leq K\}}\right),$$
$$\mathbb{E}^*\left(\mathbf{1}_{\{S_\tau\geq K\}}\right) = 1 - \mathbb{E}^*\left(\mathbf{1}_{\{S_\tau\leq K\}}\right).$$

Summary C.17. We now provide a summary of all formulas needed to price American options under the Heston model. The American put option premium is given by:

$$P_0 = p_0 + e_0,$$

where

$$e_0 = \int_0^T rKe^{-r\tau}\,\mathbb{E}\left(\mathbf{1}_{\{S_\tau \le S_c(T-\tau,v_\tau)\}}\right) - qS_0e^{-q\tau}\,\mathbb{E}^*\left(\mathbf{1}_{\{S_\tau \le S_c(T-\tau,v_\tau)\}}\right)\,\mathrm{d}\tau,$$

$$\mathbb{E}\left(\mathbf{1}_{\{S_\tau \le S_c(T-\tau,v_\tau)\}}\right) = \frac{1}{2} - \frac{1}{\pi}\int_{v=0}^{\infty}\int_{\phi=0}^{\infty} \mathrm{Re}\left(\frac{e^{i\phi\left(\log\left(\frac{S_0}{S_c(T-\tau,v)}\right)+(r-q)\tau\right)}G_\tau(\phi,v)}{i\phi}\right)\,\mathrm{d}\phi\,\mathrm{d}v,$$

$$\mathbb{E}^*\left(\mathbf{1}_{\{S_\tau \le S_c(v_\tau,T-\tau)\}}\right) = \frac{1}{2} - \frac{1}{\pi}\int_{v=0}^{\infty}\int_{\phi=0}^{\infty} \mathrm{Re}\left(\frac{e^{i\phi\left(\log\left(\frac{S_0}{S_c(T-\tau,v)}\right)+(r-q)\tau\right)}G_\tau(\phi-i,v)}{i\phi}\right)\,\mathrm{d}\phi\,\mathrm{d}v,$$

$$G_\tau(\phi,v;v_0) = \begin{cases} \frac{1}{\zeta}e^{(\delta\kappa\theta-\gamma\nu)\tau+\delta(v_0-v)-\frac{1}{\zeta}(ve^{-\gamma\tau}+v_0)}v^{\nu-1}\left(\frac{1}{\zeta^2}\right)^{\frac{\nu-1}{2}}\frac{1}{\Gamma(\nu)} & \textit{if}\quad v_0=0,\\ \frac{1}{\zeta}e^{\left(\delta\kappa\theta-\frac{\gamma(\nu+1)}{2}\right)\tau+\delta(v_0-v)-\frac{1}{\zeta}(ve^{-\gamma\tau}+v_0)}\left(\frac{v}{v_0}\right)^{\frac{\nu-1}{2}}I_{\nu-1}\left(2\sqrt{\frac{v_0ve^{-\gamma\tau}}{\zeta^2}}\right) & \textit{if}\quad v_0>0, \end{cases}$$

$$\gamma = \sqrt{\kappa^2+(1-\rho^2)\sigma^2\phi^2+i(\sigma-2\kappa\rho)\sigma\phi},\ \delta = \frac{\kappa+\gamma-i\rho\sigma\phi}{\sigma^2},\ \zeta = \frac{\sigma^2}{2\gamma}(1-e^{-\gamma\tau}),\ \nu = \frac{2\kappa\theta}{\sigma^2},$$

and

$$p_0 = Ke^{-rT}\,\mathbb{E}\left(\mathbf{1}_{\{S_T\le K\}}\right) - S_0e^{-qT}\,\mathbb{E}^*\left(\mathbf{1}_{\{S_T\le K\}}\right),$$

$$\mathbb{E}\left(\mathbf{1}_{\{S_T\le K\}}\right) = \frac{1}{2} - \frac{1}{\pi}\int_{\phi=0}^{\infty}\mathrm{Re}\left(\frac{e^{i\phi\left(\log\left(\frac{S_0}{K}\right)+(r-q)T\right)}H_T(\phi)}{i\phi}\right)\,\mathrm{d}\phi,$$

$$\mathbb{E}^*\left(\mathbf{1}_{\{S_T\le K\}}\right) = \frac{1}{2} - \frac{1}{\pi}\int_{\phi=0}^{\infty}\mathrm{Re}\left(\frac{e^{i\phi\left(\log\left(\frac{S_0}{K}\right)+(r-q)T\right)}H_T(\phi-i)}{i\phi}\right)\,\mathrm{d}\phi,$$

$$H_T(\phi) = h_\tau(\phi,0),$$

$$h_\tau(\phi,\xi) = e^{(\delta\kappa\theta-\gamma\nu)\tau-\nu\log\left(e^{-\gamma\tau}-\frac{\sigma^2}{2\gamma}(1-e^{-\gamma\tau})(i\xi-\delta)\right)+\left(\delta+\frac{(i\xi-\delta)}{e^{-\gamma\tau}-\frac{\sigma^2}{2\gamma}(1-e^{-\gamma\tau})(i\xi-\delta)}\right)v_0}.$$

The American call option premium is given by:

$$C_0 = c_0 + \breve{e}_0,$$

where

$$\breve{e}_0 = \int_0^T qS_0 e^{-q\tau}\,\mathbb{E}^*\left(\mathbf{1}_{\{S_\tau \geq S_c(T-\tau,v_\tau)\}}\right) - rKe^{-r\tau}\,\mathbb{E}\left(\mathbf{1}_{\{S_\tau \geq S_c(T-\tau,v_\tau)\}}\right)\,\mathrm{d}\tau,$$

$$\mathbb{E}\left(\mathbf{1}_{\{S_\tau \geq S_c(T-\tau,v_\tau)\}}\right) = \frac{1}{2} + \frac{1}{\pi}\int_{v=0}^{\infty}\int_{\phi=0}^{\infty} \mathrm{Re}\left(\frac{e^{i\phi\left(\log\left(\frac{S_0}{S_c(T-\tau,v)}\right)+(r-q)\tau\right)}G_\tau(\phi,v)}{i\phi}\right)\,d\phi\,dv,$$

$$\mathbb{E}^*\left(\mathbf{1}_{\{S_\tau \geq S_c(T-\tau,v_\tau)\}}\right) = \frac{1}{2} + \frac{1}{\pi}\int_{v=0}^{\infty}\int_{\phi=0}^{\infty} \mathrm{Re}\left(\frac{e^{i\phi\left(\log\left(\frac{S_0}{S_c(T-\tau,v)}\right)+(r-q)\tau\right)}G_\tau(\phi-i,v)}{i\phi}\right)\,\mathrm{d}\phi\,\mathrm{d}v,$$

and

$$c_0 = S_0 e^{-qT}\,\mathbb{E}^*\left(\mathbf{1}_{\{S_T \geq K\}}\right) - Ke^{-rT}\,\mathbb{E}\left(\mathbf{1}_{\{S_T \geq K\}}\right),$$

$$\mathbb{E}\left(\mathbf{1}_{\{S_T \geq K\}}\right) = \frac{1}{2} + \frac{1}{\pi}\int_{\phi=0}^{\infty} \mathrm{Re}\left(\frac{e^{i\phi\left(\log\left(\frac{S_0}{K}\right)+(r-q)T\right)}H_T(\phi)}{i\phi}\right)\,\mathrm{d}\phi,$$

$$\mathbb{E}^*\left(\mathbf{1}_{\{S_T \geq K\}}\right) = \frac{1}{2} + \frac{1}{\pi}\int_{\phi=0}^{\infty} \mathrm{Re}\left(\frac{e^{i\phi\left(\log\left(\frac{S_0}{K}\right)+(r-q)T\right)}H_T(\phi-i)}{i\phi}\right)\,\mathrm{d}\phi.$$

D. IMPLEMENTATION OF THE METHODOLOGY IN THE GPU

In order to compute $I_\nu(z)$ in the GPU for $\nu \in \mathbb{R}$ and $z \in \mathbb{C}$, we code in the CUDA programming language two numerical methods that perform well for a wide range of arguments z and orders ν.

The first method that we implement is the one proposed by Thompson & Barnett (1987), that can be found coded in C in Flannery et al. (1992, Chapter 6, p. 248). We use this method whenever $|z| \leq 30$. The translation into CUDA is straightforward, but relies heavily on the CUSP library that defines numbers of complex type. In the case in which $|z| > 30$, we use instead an asymptotic expansion for large argument that can be found in Olver et al. (2010, eq. 10.40.5, p. 255).

Our implementation is different from the one found in Matlab which is programmed in Fortran 77 and based on Amos (1985, 1986). We do allow, however, for the possibility of scaling the function internally by $e^{-|\operatorname{Re}(z)|}$ to avoid overflow, as in Matlab. It turns out that this feature is crucial in the numerical implementation of our methodology.

E. EFFICIENT NUMERICAL METHODS FOR PRICING AMERICAN PUT OPTIONS UNDER HESTON'S STOCHASTIC VOLATILITY MODEL

E.1 Option Pricing Model and Linear Complementary Problem (LCP)

As stated by Ikonen & Toivanen (2007*b*), due to the early-exercise possibility, the option pricing model under Heston's stochastic volatility can be solved as a time dependent linear complementary problem (LCP). For the American put option, its LCP is defined by:

$$\begin{cases} L\,p \geq 0,\; p \geq g, \\ (L\,p)\,(p-g) = 0, \end{cases}$$

where a generalized Black-Scholes operator is defined by

$$Lp = \frac{\partial p}{\partial \tau} - \frac{1}{2}\,v\,S^2\,\frac{\partial^2 p}{\partial S^2} - \rho\,\gamma\,v\,S\,\frac{\partial^2 p}{\partial S \partial v} - \frac{1}{2}\,\gamma^2\,v\,\frac{\partial^2 p}{\partial v^2} - r\,S\,\frac{\partial p}{\partial S} - \{\kappa\,(\theta - v) - \vartheta\,\gamma\,\sqrt{v}\}\,\frac{\partial p}{\partial v} + r\,p,$$

in a domain $\{(S, v, \tau)\,|\,S \geq 0, v \geq 0, \tau \in [0, T]\}$, where T is the expiry time. Price at maturity is given by the payoff function g. The market price of risk is denoted by ϑ and it is assumed to be zero. For this model, the payoff function is

$$g(S, v) = \max\{K - S, 0\},$$

The early-exercise possibility of the American option leads to the constraint

$$p(S, v, t) \geq g(S, v),$$

In the region where the constraint is inactive, the price p satisfies the partial differential equation

$$L\,p = 0,$$

The initial and boundary conditions are defined by

$$p(S, v, 0) = 0,$$

$$p(0, v, \tau) = g(0, v),$$

$$v = 0 \rightarrow \begin{cases} L\,u = \frac{\partial p}{\partial \tau} - r\,S\,\frac{\partial p}{\partial S} - \kappa\,\theta\,\frac{\partial p}{\partial v} + r\,p\ \geq 0,\ p \geq g, \\ (L\,p)\,(p - g) = 0, \end{cases}$$

The asymptotic behavior of p satisfies the conditions

$$\begin{cases} \lim_{S \to \infty} \frac{\partial p(S,v,\tau)}{\partial S} = 0, \\ \lim_{v \to \infty} \frac{\partial p(S,v,\tau)}{\partial v} = 0. \end{cases}$$

E.2 Proposed Discretization for Nonuniform Grids

The authors contribute by proposing a finite difference space discretization on a nonuniform grid resulting in an M-matrix (noted A by the authors). The objective of using nonuniform grids is to obtain the same accuracy in terms of prices with fewer grid points with these grids than with uniform grids. This is achieved by creating finer grids in parts of the computational domain where additional grid points increase the accuracy. The cross-derivative term is approximated with a special finite difference scheme while the first-order and second-order partial derivatives are approximated using usual finite differences. In order to obtain nonpositive codiagonal elements, they restrict the grid step sizes and use one-sided differences for the first-order derivative terms in a small part of the domain. Grid points are computed using grid generating functions which concentrate more points near the zone where accuracy is required the most for the option pricing (around the strike price) giving the model a Nonuniform characteristic.

Ikonen & Toivanen (2007*b*) propose a seven point discetization stencil, where they use central finite differences as much as possible, but when this leads to a positive codiagonal

element, they employ first-order accurate one-sided differences for the spatial first-order partial derivative terms. With certain grid step size restrictions, they obtain a coefficient M-matrix. In order to avoid oscillations in the resolution of the pricing model, the authors establish boundaries for the grid step sizes to ensure an M-matrix. In order to obtain good damping properties and second-order accuracy, Ikonen & Toivanen (2007*b*) use the Rannacher-time stepping method, which combines both the implicit Euler and the Crank-Nicolson methods. The finite difference approximation is constructed on a grid

$$(S_i, v_j, \tau_k) \in [S_0 = 0, ..., S_m = X] \times [v_0 = 0, ..., S_n = Y] \times [\tau_0 = 0, ..., \tau_l = T],$$

The discrete LCP problem proposed by the authors can be written down as:

$$B\,p^{k+1} \geq C\,p^k,\; p^{k+1} \geq g,$$

$$(B\,p^{k+1} - C\,p^k)^T\,(p^{k+1} - g) = 0$$

$$where\; B = (I + \frac{1}{2}\,\Delta\,\tau\,A),\; k = 0,\; ...\,, l-1$$

$$C = I,\; k = 0, 1, 2, 3$$

$$C = I - \frac{1}{2}\,\Delta\,\tau\,A,\; k = 4,\; ...\,, l-1$$

Where I is an Identity Matrix and A is an M-matrix. To simplify the notation, Ikonen & Toivanen (2007*b*) refer to the problem for a given k by

$$LCP(B, p^{k+1}, C\,p^k, g)$$

E.3 Projected Succesive Over-relaxation Method (PSOR)

This method to solve LCP problems was proposed by Cryer in 1971 and it is a variant of the Gauss-Siedel Iterative Method. The method is widely used for pricing American options. For simplicity, the algorithm which performs one PSOR iteration for the problem

$LCP(B, p, f, g)$ is presented.

$$
\begin{aligned}
&Algorithm\, PSOR(B, p, f, g)\\
&Do\ i = 1,\ ...,\ dim B\\
&r_i = f_i - \sum_j B_{i,j}\, p_j\\
&p_j = \max\left\{p_i + \frac{\omega\, r_i}{B_{i,i}}, g_i\right\}\\
&End\, Do
\end{aligned}
$$

The stopping criterion used in the numerical experiments is based on the vector r generated by the algorithm. For this method in particular, the stopping criterion is:

$$\|r\| \leq \frac{0.1}{m\, n}\|f\|$$

$$Where\, r = \begin{cases} 0 & \textit{if} \qquad\quad p_i = g_i, \\ r_i & \textit{otherwise} \end{cases}$$

The convergence rate of the PSOR method depends on the choice of the relaxation parameter ω. In our implementation, we use the values proposed by the authors based on numerical experiments made by them.

E.4 Componentwise Splitting Method

This method is based on a decomposition of the M-matrix resulting from the aforementioned space discretization. This matrix is decomposed into three matrices as

$$A = A_S + A_{Sv} + A_v$$

Where each of the matrices obtained from the decomposition have a simpler structure than A. $A_S, A_{S,v}$, and A_v contain couplings of the finite difference stencil in the S-direction,

Sv-direction and in the v-direction, respectively. After a suitable permutation of rows and columns each of these matrices are block diagonal with tridiagonal blocks. The authors affirm in a previous work done by them Ikonen & Toivanen (2007a) that a combination of discretized LCP and symmetric componentwise splitting can be second-order accurate. Due to this, the componentwise splitting method proposed has five sub-problems at each time step defined by

$$for\, k = 0, ..., l-1 \begin{cases} LCP(B_{S/2}, p^{k+\frac{1}{5}}, C_{S/2}\, p^{k}, g), \\ LCP(B_{v/2}, p^{k+\frac{2}{5}}, C_{v/2}\, p^{k+\frac{1}{5}}, g), \\ LCP(B_{Sv}, p^{k+\frac{3}{5}}, C_{Sv}\, p^{k+\frac{2}{5}}, g), \\ LCP(B_{v/2}, p^{k+\frac{4}{5}}, C_{v/2}\, p^{k+\frac{3}{5}}, g), \\ LCP(B_{S/2}, p^{k+1}, C_{S/2}\, p^{k+\frac{4}{5}}, g), \end{cases}$$

The matrices in this method depend on the time discretization scheme. In our implementation, as well as in the authors' case, the Rannacher time step is the chosen time scheme. This means:

$$\begin{aligned} B_{S/2} &= (I + \tfrac{1}{4}\,\Delta\tau\, A_S), \quad k = 0, ..., l-1 \\ B_{Sv} &= (I + \tfrac{1}{2}\,\Delta\tau\, A_{Sv}), \quad k = 0, ..., l-1 \\ B_{v/2} &= (I + \tfrac{1}{4}\,\Delta\tau\, A_v), \quad k = 0, ..., l-1 \\ C_{S/2} &= C_{Sv} = C_{v/2} = I \quad k = 0, 1, 2, 3 \\ C_{S/2} &= (I - \tfrac{1}{4}\,\Delta\tau\, A_S), \quad k = 4, ..., l-1 \\ B_{S/2} &= (I - \tfrac{1}{2}\,\Delta\tau\, A_{Sv}), \quad k = 4, ..., l-1 \\ B_{S/2} &= (I - \tfrac{1}{4}\,\Delta\tau\, A_v), \quad k = 4, ..., l-1 \end{aligned}$$

Each of the matrices has its rows and columns permuted in order to obtain block diagonal matrices with tridiagonal blocks, which are M-matrices. It is important to note that in the permutations, there has to be consistency in the early-exercise region associated with each of the option prices in order to use the Brennan and Schwartz algorithm to solve the LCPs, avoiding the use of an iterative method.

E.4.1 Brennan and Schwartz Algorithm

Introduced by Brennan & Schwartz (1977) in order to price a put option using the one-dimensional Black Scholes PDE discretized with finite differences, this method solves the LCP by using a UL decomposition with a projection in the backsubsitution step, where U and L are bidiagonal upper and lower triangular matrices, respectively. The following simplified algorithm describes the method for solving a $LCP(T, x, b, g)$:

$$
\begin{aligned}
&Algorithm\, BS(T, x, b, g)\\
&Form\, decomposition\, UL = T\\
&y = U^{-1}\, b\\
&x_1 = \max\{\frac{y_1}{L_{1,1}}, g_1\}\\
&Do\, i = 2, ..., dim\, T\\
&x_i = \max\{\frac{y_i - L_{i,i-1}\, x_{i-1}}{L_{i,i}}, g_i\}\\
&End\, Do
\end{aligned}
$$

Additionally, sufficient conditions for the algorithm to yield the correct solution are that T is an M-matrix and an integer k exists such that for the solution x it holds that $x_i = g_i$ for $i \leq k$ and $x_i > g_i$ for $i > k$.

E.5 The Transformation Procedure

As the authors state in Chockalingam & Muthuraman (2011), equations for the Heston option pricing model are satisfied by a price function $p(\tau, S, v)$ and an optimal exercise policy $b(\tau, v)$. The main argument behind this method is that, as the optimal price is a function of the optimal-exercise policy, one can say that an arbitrary exercise policy $b^0(\tau, v)$ defines an associated value $p^0(\tau, S, v)$ which satisfies $p^0(\tau, S, v) \leq p(\tau, S, v)$ for

any b^0 if both the arbitrary user defined exercise policy and related price solve the Heston equations.

The goal is to use a procedure that converges and provides the option price at convergence. Starting from an initial guess b^0, the method provides a sequence of policies $b^0, b^1, ...$ that are monotonic increasing and bounded above. To achieve this, it is required that the initial guess policy has to be such that complies with the following condition

$$b^0(\tau, v) < b(\tau, v) \rightarrow \left.\frac{\partial p^0}{\partial S}\right|_{(\tau, b^0(\tau,v)+, v)} < -1$$

Once the condition is satisfied, we can see that it is straightforward to determine that the exercise policy and option price can be improved. On the boundary $b^0, p^0 = \max\{K - S, 0\}$ and immediately above our initial policy, the derivative of the option price with respect to S is less than -1. This implies that the option price will be less than the payoff function $\max\{K - S, 0\}$ in a region right above the exercise policy guess. Then, the policy can be improved because an immediate exercise of the option will improve its price. Following this rationale, the authors iterate the exercise policy and related price based on:

$$b^{n+1}(\tau, v) = \left\{ \sup_{(b^n(\tau,v),\infty)} S \mid \left.\frac{\partial p^n}{\partial S}\right|_{(\tau,S,v)} < -1 \right\}$$

In their paper, the authors give theoretical support and proofs that this method effectively converges to $\frac{\partial p}{\partial S} = -1$ at the limit near the exercise boundary (Smooth-pasting condition), implying optimality in the option price.

Chockalingam & Muthuraman (2011) explains that the proposed algorithm can be analyzed as a decomposition of the American option pricing problem into a sequence of European-style option pricing problems (The exercise policy is known beforehand and one can solve in each iteration the pricing equations as a European flavored option, removing the free boundary inherent complexity).

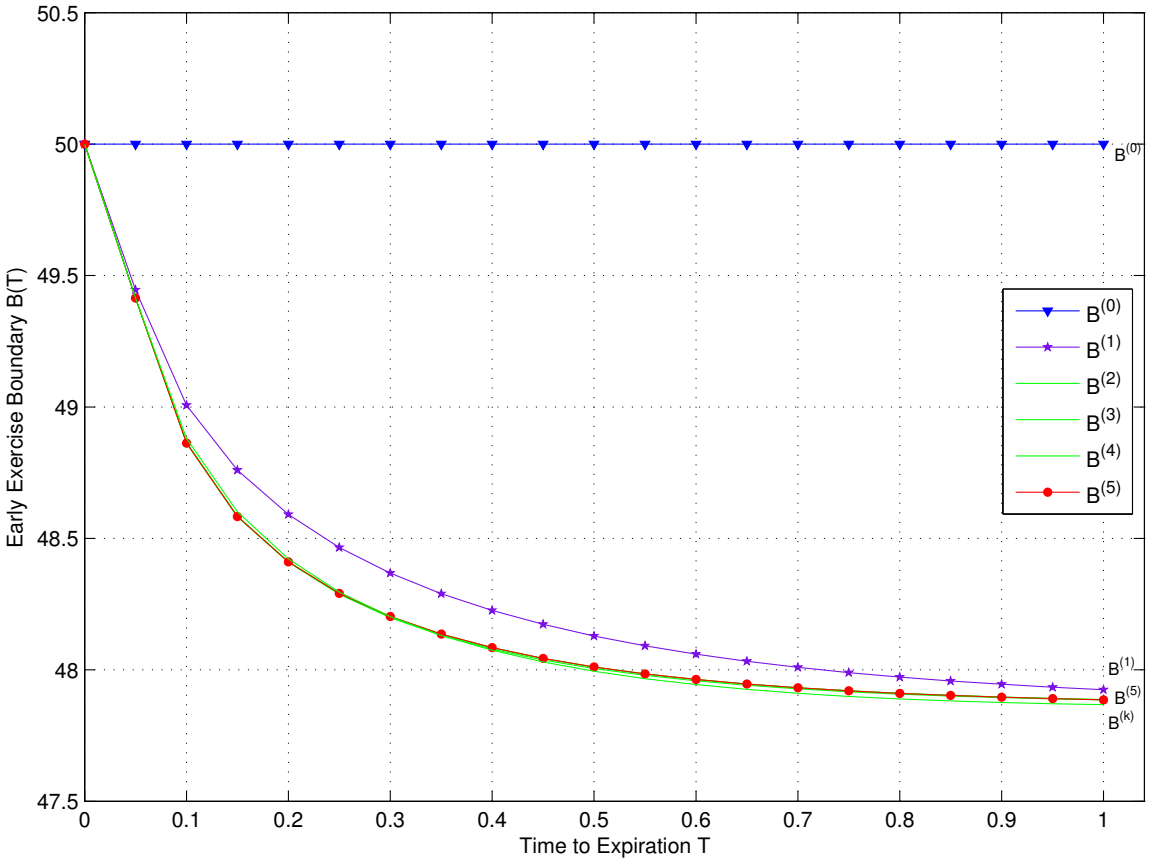

Figure 1 – The figure shows different iterations of the early exercise curve $B^{(k)}$ using our functional iterative method with a flat-prior and the trapezoidal rule. We use 20 time-intervals to discretize the time-to-maturity. In the figure, the strike price is $K = 100$, the time-to-maturity is $T = 1$, the risk-free rate is $r = 0.04$, the dividend rate is $q = 0.08$, and the volatility is $\sigma = 0.2$. Convergence is obtained after 5 iterations.

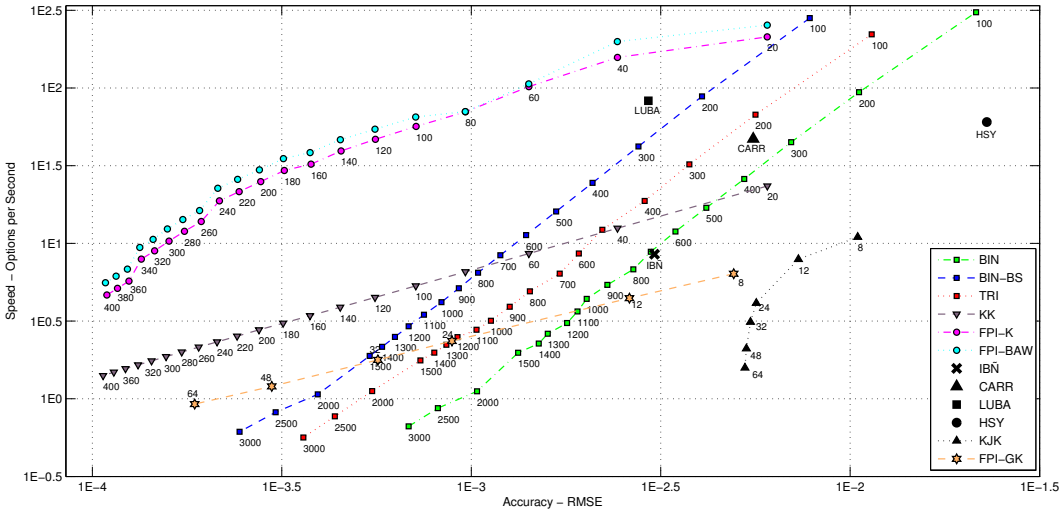

Figure 2 – Comparison of RMSE and computing time for American put prices under the Black & Scholes option pricing model and calculated using several numerical methods. Speed is measured as the average number of options calculated per second. Testing set corresponds to 7,865 options. Combination of parameters are initial stock price $S_0 = 75, 80, ..., 120, 125$, strike price $K = 100$; $T = 1/12, 3/12, 6/12, 9/12, 1, 2, 3$, risk-free rate of interest per annum $r = 0.02, ..., 0.1$, volatility $\sigma = 0.1, ..., 0.6$, and dividend rate $\delta = 0, 0.04, 0.08, 0.12$. We exclude options with prices of less than 50 cents and negative early exercise premia. Numbers below each point represent the time-steps used in each method.

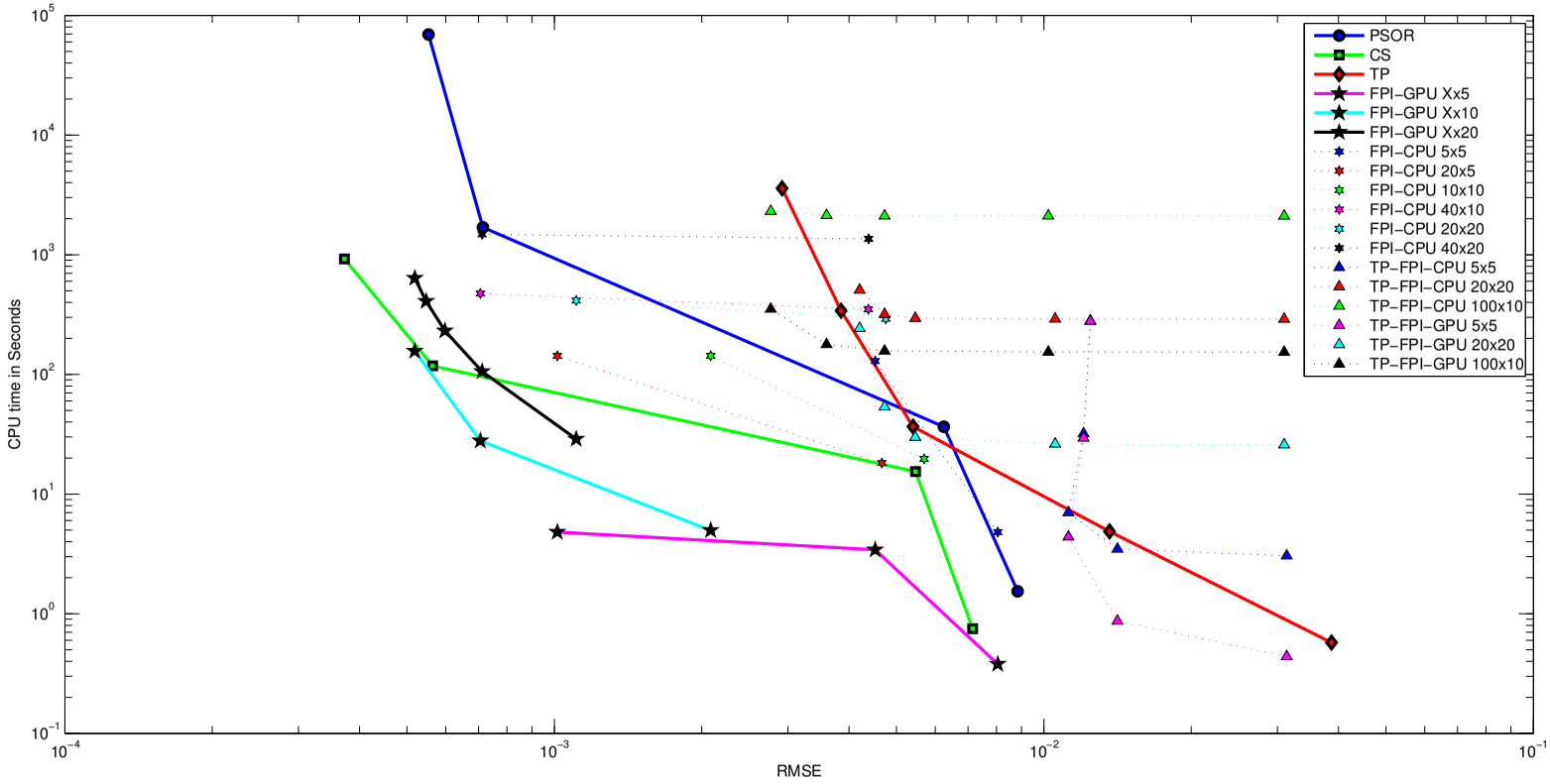

Figure 3 – Comparison of RMSE and computing time for American put prices under the Heston option pricing model and calculated using several numerical methods. Initial stock price $S_0 = 8, ..., 12$, and initial variance $\boldsymbol{v_0 = 0.0625}$. Strike price K = 10, time-to-maturity T = 3 months, risk-free rate of interest r = 0.1 per annum, mean rate of reversion $\kappa = 5$, long-term mean variance $\theta = 0.16$, volatility of the volatility process $\sigma = 0.9$, and correlation $\rho = 0.1$.

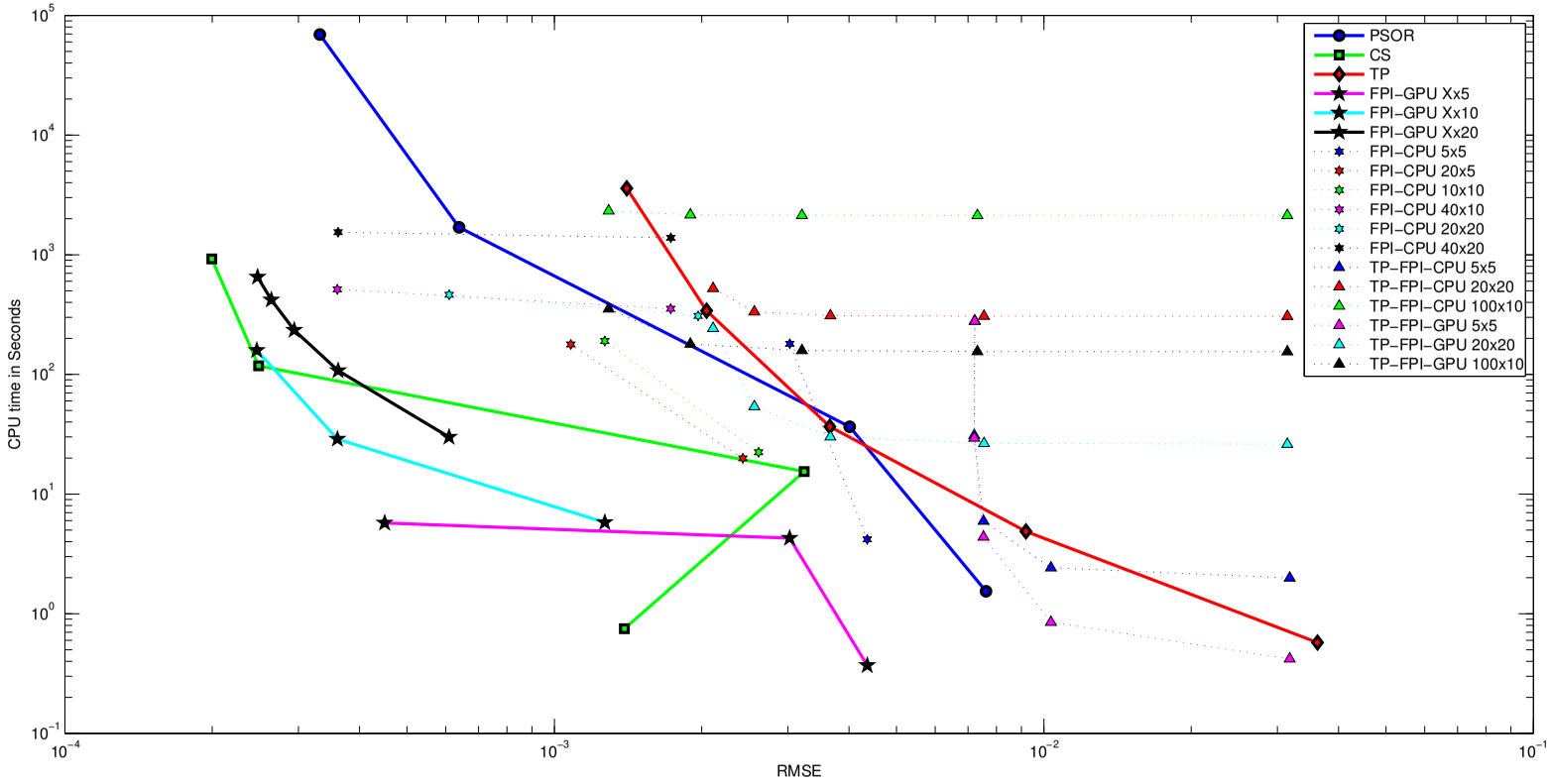

Figure 4 – Comparison of RMSE and computing time for American put prices under the Heston option pricing model and calculated using several numerical methods. Initial stock price $S_0 = 8, ..., 12$, and initial variance $\boldsymbol{v_0 = 0.25}$. Strike price K = 10, time-to-maturity T = 3 months, risk-free rate of interest r = 0.1 per annum, mean rate of reversion $\kappa = 5$, long-term mean variance $\theta = 0.16$, volatility of the volatility process $\sigma = 0.9$, and correlation $\rho = 0.1$.

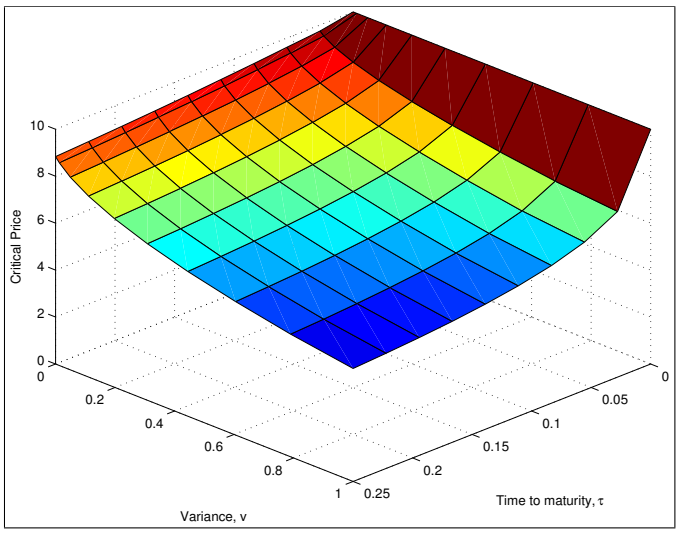

Figure 5 – Optimal exercise policy under Heston model and calculated using FPI and a 10x10 grid. Strike price K = 10, time-to-maturity T = 3 months, risk-free rate of interest r = 0.1 per annum, mean rate of reversion $\kappa = 5$, long-term mean variance $\theta = 0.16$, volatility of the volatility process $\sigma = 0.9$, and correlation $\rho = 0.1$.

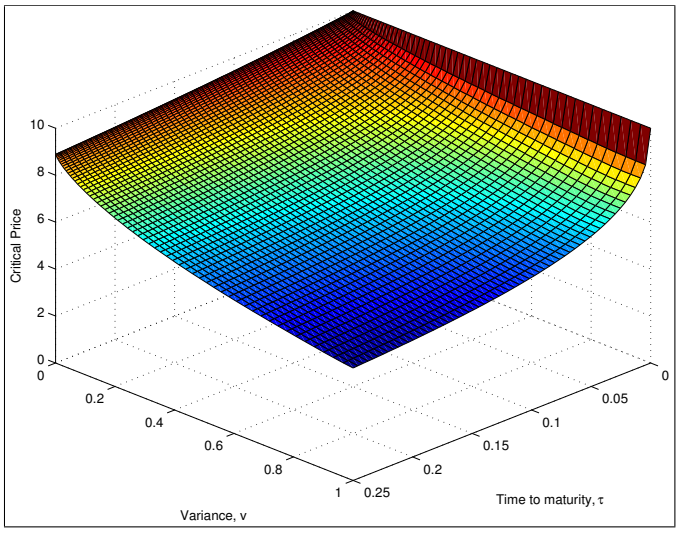

Figure 6 – Optimal exercise policy under Heston model and calculated using FPI and a 60x60 grid. Strike price K = 10, time-to-maturity T = 3 months, risk-free rate of interest r = 0.1 per annum, mean rate of reversion $\kappa = 5$, long-term mean variance $\theta = 0.16$, volatility of the volatility process $\sigma = 0.9$, and correlation $\rho = 0.1$.

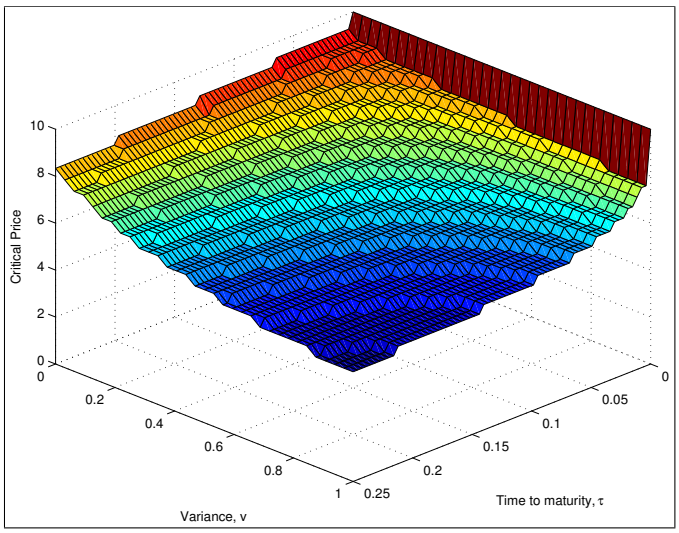

Figure 7 – Optimal exercise policy under Heston model and calculated using TP and a 60x32x66 grid. Strike price K = 10, time-to-maturity T = 3 months, risk-free rate of interest r = 0.1 per annum, mean rate of reversion $\kappa = 5$, long-term mean variance $\theta = 0.16$, volatility of the volatility process $\sigma = 0.9$, and correlation $\rho = 0.1$.

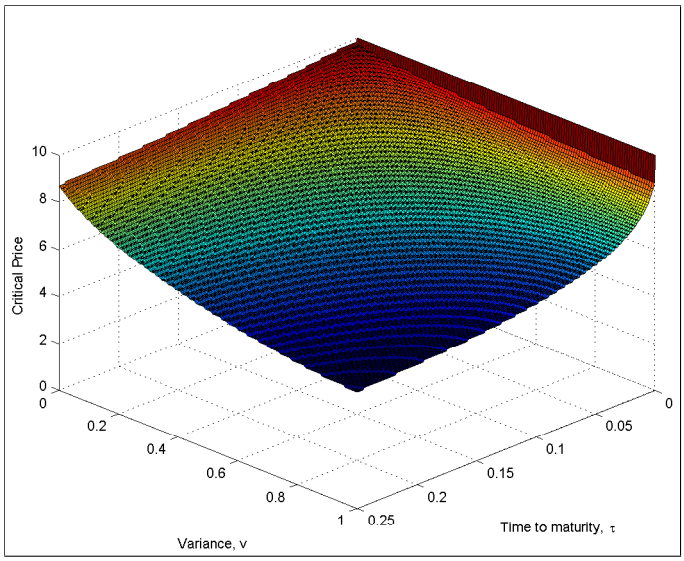

Figure 8 – Optimal exercise policy under Heston model and calculated using TP and a 240x128x258 grid. Strike price K = 10, time-to-maturity T = 3 months, risk-free rate of interest r = 0.1 per annum, mean rate of reversion $\kappa = 5$, long-term mean variance $\theta = 0.16$, volatility of the volatility process $\sigma = 0.9$, and correlation $\rho = 0.1$.

Table I – Pricing Accuracy of the Functional Iterative Method FPI-F ($K = 100$, $T = 3$, $\sigma = 0.2$)

	Spot	(1) True	(2) $N_T = 20$	(3) $N_T = 40$	(4) $N_T = 60$	(5) $N_T = 80$	(6) $N_T = 100$	(7) $N_T = 120$	(8) $N_T = 140$	(9) $N_T = 200$	(10) $N_T = 300$	(11) $N_T = 400$
$r = 0.04$	80	23.22837	23.23249	23.22990	23.22921	23.22892	23.22876	23.22866	23.22860	23.22850	23.22843	23.22840
$\delta = 0.04$			*0.00412*	*0.00153*	*0.00084*	*0.00055*	*0.00039*	*0.00029*	*0.00023*	*0.00013*	*0.00006*	*0.00003*
	100	12.60529	12.60836	12.60644	12.60592	12.60568	12.60556	12.60548	12.60543	12.60534	12.60529	12.60526
			0.00308	*0.00116*	*0.00063*	*0.00040*	*0.00027*	*0.00019*	*0.00014*	*0.00006*	*0.00000*	*0.00002*
	120	6.48247	6.48442	6.48323	6.48289	6.48274	6.48266	6.48261	6.48257	6.48252	6.48248	6.48246
			0.00194	*0.00075*	*0.00042*	*0.00027*	*0.00018*	*0.00013*	*0.00010*	*0.00004*	*0.00000*	*0.00002*
$r = 0.04$	80	33.90208	33.90228	33.90216	33.90213	33.90212	33.90211	33.90210	33.90210	33.90210	33.90209	33.90209
$\delta = 0.12$			*0.00019*	*0.00008*	*0.00005*	*0.00003*	*0.00003*	*0.00002*	*0.00002*	*0.00001*	*0.00001*	*0.00001*
	100	22.83353	22.83359	22.83357	22.83357	22.83356	22.83356	22.83356	22.83356	22.83356	22.83356	22.83356
			0.00006	*0.00004*	*0.00003*	*0.00003*	*0.00003*	*0.00003*	*0.00003*	*0.00002*	*0.00002*	*0.00002*
	120	14.50205	14.50215	14.50215	14.50215	14.50215	14.50215	14.50215	14.50215	14.50215	14.50215	14.50215
			0.00010	*0.00010*	*0.00010*	*0.00010*	*0.00010*	*0.00010*	*0.00010*	*0.00010*	*0.00010*	*0.00010*
$r = 0.08$	80	20.35002	20.34699	20.35250	20.35163	20.35080	20.35031	20.35003	20.34990	20.34982	20.34993	20.35000
$\delta = 0.04$			*0.00303*	*0.00248*	*0.00161*	*0.00078*	*0.00029*	*0.00001*	*0.00012*	*0.00020*	*0.00009*	*0.00003*
	100	8.94399	8.94509	8.94445	8.94427	8.94419	8.94414	8.94411	8.94409	8.94405	8.94402	8.94401
			0.00110	*0.00045*	*0.00027*	*0.00019*	*0.00014*	*0.00011*	*0.00009*	*0.00006*	*0.00003*	*0.00001*
	120	3.89743	3.90030	3.89855	3.89808	3.89787	3.89775	3.89768	3.89763	3.89755	3.89749	3.89747
			0.00287	*0.00111*	*0.00064*	*0.00044*	*0.00032*	*0.00025*	*0.00020*	*0.00012*	*0.00006*	*0.00003*
$r = 0.08$	80	25.65774	25.66244	25.65946	25.65870	25.65838	25.65821	25.65811	25.65804	25.65793	25.65786	25.65783
$\delta = 0.12$			*0.00470*	*0.00172*	*0.00096*	*0.00064*	*0.00047*	*0.00036*	*0.00029*	*0.00019*	*0.00011*	*0.00009*
	100	15.49841	15.50063	15.49924	15.49887	15.49871	15.49863	15.49858	15.49854	15.49849	15.49845	15.49844
			0.00222	*0.00082*	*0.00046*	*0.00030*	*0.00021*	*0.00016*	*0.00013*	*0.00008*	*0.00004*	*0.00003*
	120	8.88548	8.88646	8.88587	8.88571	8.88564	8.88560	8.88558	8.88556	8.88554	8.88552	8.88552
			0.00098	*0.00038*	*0.00022*	*0.00015*	*0.00012*	*0.00009*	*0.00008*	*0.00005*	*0.00004*	*0.00003*
	RMSE		2.530E-03	1.142E-03	6.799E-04	3.978E-04	2.502E-04	1.808E-04	1.486E-04	1.039E-04	5.950E-05	4.308E-05
	RRMSE		2.723E-04	1.102E-04	6.416E-05	4.183E-05	2.977E-05	2.273E-05	1.826E-05	1.102E-05	5.833E-06	3.820E-06

Note – Column (1) reports the true value of an American put option based on a binomial tree with 15 000 time-steps that uses Black & Scholes at the last time-step. Columns (2) to (11) report prices of American put options calculated using the FPI-F method with different number of time-steps (N_T). Absolute errors between the true value and each American option price are displayed in italics. The last two rows report the root mean squared error (RMSE) and the relative root mean squared error (RRMSE).

Table II – Prices of American Put Options ($K = 100$, $\delta = 0.04$, $r = 0.04$)

	Spot	True	BAW	LS	JZ	HSY	CARR	BIN 500	TRI 500	BIN 1000	BIN-BS 500	LUBA
$T = 0.5$	80	20.14372	20.12517	20.13108	20.13987	20.13870	20.14286	20.14385	20.14333	20.14344	20.14360	20.14433
$\sigma = 0.2$			*0.01854*	*0.01264*	*0.00384*	*0.00502*	*0.00086*	*0.00013*	*0.00039*	*0.00027*	*0.00012*	*0.00061*
	100	5.54634	5.55141	5.52130	5.54328	5.54794	5.54499	5.54378	5.54501	5.54504	5.54735	5.54600
			0.00507	*0.02504*	*0.00306*	*0.00160*	*0.00135*	*0.00256*	*0.00133*	*0.00130*	*0.00101*	*0.00034*
	120	0.70724	0.71067	0.70584	0.70776	0.70738	0.70686	0.70776	0.70747	0.70747	0.70697	0.70719
			0.00342	*0.00140*	*0.00052*	*0.00014*	*0.00039*	*0.00051*	*0.00023*	*0.00023*	*0.00028*	*0.00005*
$T = 0.5$	80	24.67736	24.67381	24.61927	24.65983	24.67603	24.67401	24.67449	24.67513	24.67528	24.67871	24.67858
$\sigma = 0.5$			*0.00355*	*0.05810*	*0.01754*	*0.00133*	*0.00335*	*0.00287*	*0.00223*	*0.00208*	*0.00135*	*0.00122*
	100	13.80581	13.81879	13.73791	13.79824	13.80981	13.80246	13.79944	13.80249	13.80258	13.80824	13.80651
			0.01298	*0.06790*	*0.00757*	*0.00400*	*0.00335*	*0.00637*	*0.00332*	*0.00323*	*0.00243*	*0.00070*
	120	7.28738	7.30229	7.25333	7.28621	7.28808	7.28518	7.28982	7.28716	7.28720	7.28901	7.28743
			0.01491	*0.03405*	*0.00117*	*0.00070*	*0.00220*	*0.00244*	*0.00022*	*0.00018*	*0.00163*	*0.00005*
$T = 3$	80	23.22837	23.31946	23.12495	23.17341	23.24958	23.22514	23.22921	23.22781	23.22864	23.22922	23.22219
$\sigma = 0.2$			*0.09109*	*0.10342*	*0.05496*	*0.02121*	*0.00323*	*0.00084*	*0.00056*	*0.00027*	*0.00085*	*0.00618*
	100	12.60529	12.76321	12.56193	12.58919	12.59927	12.60001	12.60044	12.60236	12.60282	12.60732	12.60296
			0.15792	*0.04335*	*0.01610*	*0.00602*	*0.00528*	*0.00484*	*0.00293*	*0.00246*	*0.00204*	*0.00233*
	120	6.48247	6.62559	6.41858	6.49419	6.49209	6.47798	6.48133	6.48398	6.48423	6.48385	6.48009
			0.14312	*0.06390*	*0.01171*	*0.00961*	*0.00449*	*0.00115*	*0.00151*	*0.00176*	*0.00137*	*0.00239*
$T = 3$	80	37.97483	38.31502	37.98111	37.90752	38.02363	37.96429	37.97287	37.97120	37.97255	37.97786	37.97472
$\sigma = 0.5$			*0.34019*	*0.00628*	*0.06731*	*0.04880*	*0.01054*	*0.00196*	*0.00363*	*0.00228*	*0.00302*	*0.00011*
	100	30.74247	31.13286	30.70746	30.71458	30.72989	30.73039	30.73046	30.73526	30.73639	30.74631	30.74201
			0.39039	*0.03502*	*0.02790*	*0.01258*	*0.01208*	*0.01201*	*0.00721*	*0.00609*	*0.00383*	*0.00046*
	120	25.21333	25.61935	25.20967	25.21672	25.20719	25.20052	25.22383	25.21716	25.21810	25.21707	25.21079
			0.40603	*0.00365*	*0.00339*	*0.00614*	*0.01281*	*0.01050*	*0.00384*	*0.00477*	*0.00375*	*0.00253*
	RMSE		2.016E-01	4.819E-02	2.758E-02	1.633E-02	6.517E-03	5.367E-03	3.016E-03	2.751E-03	2.162E-03	2.206E-03
	RRMSE		9.962E-03	4.148E-03	1.162E-03	6.721E-04	3.757E-04	3.639E-04	1.877E-04	1.858E-04	1.789E-04	1.486E-04

Note – Each column in the table reports the price of an American put option using a different method. The true value is based on a binomial tree with 15 000 time-steps that uses Black & Scholes at the last time-step. BAW is the quadratic approximation of Barone-Adesi & Whaley (1987). LS is the least square Monte Carlo approach of Longstaff & Schwartz (2001). JZ is the refined quadratic approximation of Ju & Zhong (1999). HSY is the six-point recursive integration method of Huang et al. (1996). CARR is the six-point randomization method of Carr (1998). BIN is the binomial tree method of Cox et al. (1979). TRI is the trinomial tree method of Boyle (1988). BIN-BS is the binomial model using Black & Scholes at the last time-step of Broadie & Detemple (1996). LUBA is the lower and upper bound approximation method of Broadie & Detemple (1996). Numbers below some methods indicate the number of time-steps. Absolute errors between the true value and each American option price are displayed in italics.The last two rows report the root mean squared error (RMSE) and the relative root mean squared error (RRMSE).

Table III – Prices of American Put Options ($K = 100$, $\delta = 0.04$, $r = 0.04$)

	Spot	True	BIN-BS 1000	BIN 2500	IBN	TRI 2500	FPI-F 60	BIN-BS 2500	KJK 32	FPI-GK 32	KK 400	FPI-F 400
$T = 0.5$	80	20.14372	20.14366	20.14376	20.14094	20.14369	20.14384	20.14370	20.14355	20.14373	20.14373	20.14373
$\sigma = 0.2$			*0.00005*	*0.00004*	*0.00277*	*0.00003*	*0.00012*	*0.00002*	*0.00016*	*0.00001*	*0.00001*	*0.00001*
	100	5.54634	5.54683	5.54580	5.54661	5.54605	5.54638	5.54652	5.54633	5.54633	5.54631	5.54631
			0.00049	*0.00054*	*0.00027*	*0.00029*	*0.00004*	*0.00018*	*0.00001*	*0.00001*	*0.00003*	*0.00003*
	120	0.70724	0.70711	0.70722	0.70727	0.70730	0.70727	0.70720	0.70727	0.70726	0.70725	0.70725
			0.00013	*0.00003*	*0.00003*	*0.00006*	*0.00002*	*0.00005*	*0.00003*	*0.00002*	*0.00001*	*0.00001*
$T = 0.5$	80	24.67736	24.67802	24.67764	24.67861	24.67761	24.67757	24.67760	24.67722	24.67737	24.67733	24.67733
$\sigma = 0.5$			*0.00066*	*0.00028*	*0.00125*	*0.00025*	*0.00021*	*0.00023*	*0.00014*	*0.00001*	*0.00003*	*0.00003*
	100	13.80581	13.80699	13.80447	13.80647	13.80508	13.80592	13.80623	13.80579	13.80580	13.80574	13.80574
			0.00118	*0.00134*	*0.00066*	*0.00073*	*0.00011*	*0.00042*	*0.00002*	*0.00001*	*0.00007*	*0.00007*
	120	7.28738	7.28818	7.28695	7.28775	7.28754	7.28745	7.28767	7.28742	7.28739	7.28733	7.28733
			0.00080	*0.00043*	*0.00037*	*0.00016*	*0.00007*	*0.00029*	*0.00004*	*0.00001*	*0.00005*	*0.00005*
$T = 3$	80	23.22837	23.22880	23.22817	23.22796	23.22830	23.22921	23.22853	23.22855	23.22848	23.22840	23.22840
$\sigma = 0.2$			*0.00043*	*0.00020*	*0.00041*	*0.00007*	*0.00084*	*0.00016*	*0.00018*	*0.00011*	*0.00002*	*0.00003*
	100	12.60529	12.60629	12.60426	12.60514	12.60465	12.60592	12.60565	12.60548	12.60544	12.60526	12.60526
			0.00100	*0.00103*	*0.00015*	*0.00064*	*0.00063*	*0.00036*	*0.00019*	*0.00015*	*0.00003*	*0.00002*
	120	6.48247	6.48314	6.48242	6.48245	6.48284	6.48289	6.48272	6.48268	6.48264	6.48246	6.48246
			0.00066	*0.00005*	*0.00002*	*0.00036*	*0.00042*	*0.00025*	*0.00020*	*0.00017*	*0.00002*	*0.00002*
$T = 3$	80	37.97483	37.97633	37.97666	37.97435	37.97426	37.97655	37.97536	37.97528	37.97519	37.97484	37.97485
$\sigma = 0.5$			*0.00150*	*0.00183*	*0.00048*	*0.00057*	*0.00172*	*0.00053*	*0.00044*	*0.00036*	*0.00001*	*0.00002*
	100	30.74247	30.74437	30.73995	30.74213	30.74093	30.74404	30.74316	30.74297	30.74290	30.74245	30.74245
			0.00190	*0.00252*	*0.00034*	*0.00154*	*0.00157*	*0.00069*	*0.00049*	*0.00043*	*0.00002*	*0.00002*
	120	25.21333	25.21518	25.21458	25.21311	25.21429	25.21475	25.21401	25.21386	25.21380	25.21330	25.21330
			0.00186	*0.00126*	*0.00022*	*0.00096*	*0.00142*	*0.00068*	*0.00054*	*0.00047*	*0.00003*	*0.00003*
	RMSE		1.066E-03	1.108E-03	9.350E-04	6.387E-04	8.558E-04	3.857E-04	2.714E-04	2.243E-04	3.228E-05	3.225E-05
	RRMSE		8.677E-05	5.942E-05	5.078E-05	4.453E-05	3.783E-05	3.102E-05	1.750E-05	1.376E-05	5.118E-06	5.112E-06

Note – Each column in the table reports the price of an American put option using a different method. The true value is based on a binomial tree with 15 000 time-steps that uses Black & Scholes at the last time-step. BIN-BS is the binomial model using Black & Scholes at the last time-step of Broadie & Detemple (1996). BIN is the binomial tree method of Cox et al. (1979). IBN is the tree-point modified recursive integration method of Ibáñez (2003). TRI is the trinomial tree method of Boyle (1988). FPI-F is the functional iterative method using a flat guess and trapezoidal rule. KJK is the iterative method of Kim et al. (2013). FPI-GK is the functional iterative method using a flat guess and Gauss-Kronrod adaptive integral. KK is the recursive method of Kallast & Kivinukk (2003). Numbers below some methods indicate the number of time-steps. Absolute errors between the true value and each American option price are displayed in italics. The last two rows report the root mean squared error (RMSE) and the relative root mean squared error (RRMSE). Our method is highlighted in gray.

Table IV – Performance Statistics

	(1) RMSE	(2) Time	(3) Efficiency	(4) AE $< 10^{-3}$	(5) AE $< 10^{-4}$	(6) AE $< 10^{-5}$
BAW	2.189E-01	0.001	8.43	16.63%	11.57%	8.30%
BIN 500	4.172E-03	0.059	8.31	33.18%	10.64%	7.31%
BIN 1000	2.017E-03	0.228	7.68	49.28%	13.94%	7.57%
BIN 2500	8.166E-04	1.154	6.97	80.34%	22.75%	9.07%
BIN-BS 500	1.675E-03	0.063	9.16	48.75%	13.48%	7.27%
BIN-BS 1000	8.345E-04	0.239	8.52	75.04%	19.26%	8.05%
BIN-BS 2500	3.047E-04	1.232	7.89	99.87%	33.87%	10.40%
CARR	5.542E-03	0.021	9.06	46.34%	24.11%	8.19%
FPI-BAW 400	1.083E-04	0.176	10.87	99.91%	96.13%	32.21%
FPI-F 60	1.419E-03	0.010	11.16	82.20%	42.10%	13.81%
FPI-F 400	1.092E-04	0.209	10.69	99.92%	96.01%	32.17%
FPI-GK 24	9.636E-04	0.452	7.74	99.06%	69.43%	24.44%
HSY	2.294E-02	0.017	7.85	40.90%	26.19%	10.96%
IBN	3.044E-03	0.117	7.94	81.84%	39.86%	12.68%
JZ	3.665E-02	0.001	10.21	24.31%	15.07%	8.96%
KJK 24	6.101E-03	0.242	6.52	81.72%	50.21%	20.20%
KK 400	1.067E-04	0.711	9.49	99.92%	96.21%	32.02%
LS	7.737E-02	1.780	1.98	2.28%	0.17%	0.01%
LUBA	2.933E-03	0.012	10.25	64.28%	36.39%	12.97%
TRI 500	2.218E-03	0.082	8.61	47.74%	12.22%	7.20%
TRI 1000	1.127E-03	0.315	7.94	70.39%	17.30%	8.01%
TRI 2500	4.364E-04	1.354	7.43	96.07%	30.63%	10.44%

Note – The table reports performance statistics for each numerical method over 7 865 option values. BAW is the quadratic approximation of Barone-Adesi & Whaley (1987). BIN is the binomial tree method of Cox et al. (1979). BIN-BS is the binomial model using Black & Scholes at the last time-step of Broadie & Detemple (1996). CARR is the six-point randomization method of Carr (1998). FPI-BAW is the functional iterative method with the initial guess of Barone-Adesi & Whaley (1987) and the trapezoidal rule. FPI-F is the functional iterative method with a flat initial guess and the trapezoidal rule. FPI-GK is is the functional iterative method with a flat initial guess and the Gauss-Kronrod quadrature. HSY is the six-point recursive integration method of Huang et al. (1996). IBN is the tree-point modified recursive integration method of Ibáñez (2003). JZ is the refined quadratic approximation of Ju & Zhong (1999). KJK is the iterative method of Kim et al. (2013). KK is the recursive method of Kallast & Kivinukk (2003). LS is the least-squares Monte Carlo approach of Longstaff & Schwartz (2001). LUBA is the lower and upper bound approximation method of Broadie & Detemple (1996). TRI the trinomial tree method of Boyle (1988). Numbers next to some methods specify the number of time-steps. Column (1) reports the root mean squared error (RMSE) with respect to the true value that was computed with a binomial tree with 15 000 time-steps that uses Black & Scholes at the last time-step. Column (2) reports the average time in seconds needed to value the 7 865 options. Efficiency in (3) is computed as $-\log(\text{RMSE} \times \text{Time})$. The last three columns (4) to (6) report the percentage of options for which the absolute error (AE) is lower than the corresponding threshold. Our method is highlighted in gray.

Table V – Performance Statistics Using Richardson Extrapolation

	(1) RMSE	(2) Time	(3) Efficiency
BIN-BS $600/400/200$	9.234E-04	0.140	8.95
BIN-BS $400/200$	5.890E-04	0.052	10.39
BIN-BS $600/300$	3.456E-04	0.112	10.16
BIN-BS $800/400$	2.874E-04	0.195	9.79
BIN-BS $1000/500$	2.264E-04	0.301	9.59
FPI-F $100/50$	4.936E-04	0.035	10.97
FPI-F $150/100/50$	1.771E-04	0.073	11.26
FPI-F $200/150/100$	1.257E-04	0.086	11.44
FPI-F $250/200/150$	1.061E-04	0.124	11.24
FPI-F $300/250/200$	8.978E-05	0.182	11.02
KK $100/50$	4.951E-04	0.286	8.86
KK $160/80$	2.861E-04	0.444	8.97
KK $200/100$	2.238E-04	0.547	9.01
KK $300/150$	1.498E-04	0.809	9.02
KK $150/100/50$	1.813E-04	0.560	9.20

Note – The table reports performance statistics for three different methods that are accelerated with the use of Richardson extrapolation. BIN-BS is the binomial model using Black & Scholes at the last time-step of Broadie & Detemple (1996). FPI-F is the functional iterative method with a flat initial guess and the trapezoidal rule. KK is the recursive method of Kallast & Kivinukk (2003). The notation $N_3/N_2/N_1$ stands for the 3-point Richardson extrapolation (see e.g. Geske & Johnson 1984) where the price of the option is found as $P = P_3 + 7/2P_2 - 1/2P_1$. The notation N_2/N_1 stands for the 2-point Richardson extrapolation (see e.g. Bunch & Johnson 1992) where the price of the option is found as $P = 2P_2 - P_1$. In the formulas P_1, P_2, and P_3 are the prices obtained using N_1, N_2, and N_3 time-steps, respectively. Column (1) reports the root mean squared error (RMSE) with respect to the true value that was computed with a binomial tree with 15,000 time-steps that uses Black & Scholes at the last time-step. Column (2) reports the average time in seconds needed to value the 7,865 options. Efficiency in (3) is computed as $-\log(\text{RMSE} \times \text{Time})$. Our method is highlighted in gray.

Table VI – American Put Prices and Performance Statistics for Option Pricing Methods Under The Heston Model and $v_0 = 0.0625$

			S_0						
Method	Grid Size	Integration Points	8	9	10	11	12	Time (s)	RMSE
PSOR	(40,16,8)		1.9989	1.1005	0.5138	0.2120	0.0830	1.54	0.88%
	(60,32,66)		1.9991	1.1041	0.5162	0.2117	0.0815	36.50	0.62%
	(120,64,130)		1.9998	1.1070	0.5195	0.2135	0.0821	1.693.53	0.07%
	(240,128,258)		2.0000	1.1074	0.5198	0.2135	0.0820	69.166.69	0.06%
CS	(40,16,8)		2.0010	1.1053	0.5165	0.2130	0.0832	0.75	0.72%
	(60,32,66)		1.9992	1.1047	0.5166	0.2119	0.0816	15.43	0.55%
	(120,64,130)		1.9999	1.1072	0.5197	0.2136	0.0821	117.92	0.06%
	(240,128,258)		2.0000	1.1075	0.5198	0.2136	0.0820	922.42	0.04%
TP	(40,16,8)		1.9825	1.0756	0.4897	0.2019	0.0830	0.57	3.87%
	(60,32,66)		1.9928	1.0985	0.5116	0.2093	0.0810	4.88	1.36%
	(120,64,130)		2.0000	1.1045	0.5171	0.2120	0.0815	36.61	0.54%
	(240,128,258)		2.0000	1.1064	0.5186	0.2127	0.0815	340.47	0.39%
	(480,256,520)		2.0000	1.1070	0.5192	0.2130	0.0816	3.585.60	0.29%
FPI-CPU	(5, 5)	(100, 100)	1.9981	1.1090	0.5224	0.2155	0.0833	4.81	0.81%
	(5, 5)	(3200, 100)	1.9985	1.1083	0.5218	0.2149	0.0827	129.35	0.45%
	(20, 5)	(100, 100)	1.9989	1.1086	0.5210	0.2145	0.0828	18.17	0.47%
	(20, 5)	(3200, 100)	1.9994	1.1080	0.5203	0.2139	0.0822	142.89	0.10%
	(10, 10)	(100, 100)	1.9992	1.1084	0.5213	0.2148	0.0830	19.69	0.57%
	(10, 10)	(3200, 100)	1.9997	1.1078	0.5207	0.2142	0.0824	143.18	0.21%
	(40, 10)	(100, 100)	1.9995	1.1083	0.5208	0.2145	0.0828	350.10	0.44%
	(40, 10)	(3200, 100)	2.0000	1.1076	0.5202	0.2138	0.0822	474.18	0.07%
	(20, 20)	(100, 100)	1.9993	1.1083	0.5209	0.2146	0.0828	291.94	0.48%
	(20, 20)	(3200, 100)	1.9998	1.1077	0.5203	0.2139	0.0822	414.31	0.11%
	(40, 20)	(100, 100)	1.9994	1.1083	0.5208	0.2145	0.0828	1.359.36	0.44%
	(40, 20)	(3200, 100)	1.9999	1.1077	0.5202	0.2138	0.0822	1.482.69	0.07%
FPI-GPU	(5, 5)	(100, 100)	1.9981	1.1090	0.5224	0.2155	0.0833	0.38	0.81%
	(5, 5)	(3200, 100)	1.9985	1.1083	0.5218	0.2149	0.0827	3.42	0.45%
	(20, 5)	(3200, 100)	1.9994	1.1080	0.5203	0.2139	0.0822	4.81	0.10%
	(10, 10)	(3200, 100)	1.9997	1.1078	0.5207	0.2142	0.0824	4.97	0.21%
	(40, 10)	(3200, 100)	2.0000	1.1076	0.5202	0.2138	0.0822	27.93	0.07%
	(100, 10)	(3200, 100)	2.0000	1.1076	0.5201	0.2138	0.0821	156.75	0.05%
	(20, 20)	(3200, 100)	1.9998	1.1077	0.5203	0.2139	0.0822	28.87	0.11%
	(40, 20)	(3200, 100)	1.9999	1.1077	0.5202	0.2138	0.0822	105.79	0.07%
	(60, 20)	(3200, 100)	1.9999	1.1076	0.5201	0.2138	0.0821	231.18	0.06%
	(80, 20)	(3200, 100)	1.9999	1.1076	0.5201	0.2138	0.0821	408.62	0.05%
	(100, 20)	(3200, 100)	1.9999	1.1076	0.5201	0.2138	0.0821	636.47	0.05%
True Value			**2.0000**	**1.1076**	**0.5200**	**0.2137**	**0.0820**		

Note – The table reports American put prices and performance statistics for numerical methods under the Heston's stochastic volatility option pricing model. FPI was implemented using an adaptive integration method based on the Gauss-Kronrod quadrature. Column (2) reports the grid size used for each method. For PSOR, CS and TP, grid size refers to (n_{S_0}, n_{v_0}, n_T), as these methods calculate put option prices for each initial stock price S_0 and variance v_0 at a defined time to maturity $T = 0.25$, whereas for the FPI method we refer to the early exercise policy grid size defined by (n_τ, n_v), as this method calculates an option price given S_0 and v_0 by solving the optimal exercise policy $S_c(\tau, v)$. Column (3) reports the number of discretization points for τ and v on the policy grid used in the last step of FPI, as this method uses spline algorithms to improve the calculation of put prices. Columns (4) to (8) report put prices for each initial stock price and $v_0 = 0.0625$. Column (9) reports time in seconds and column (10) refers to the root mean square error. True value makes reference to the put prices used by Ikonen & Toivanen (2007*b*) as benchmarks. K = 10, T = 3 months, r = 0.1 per annum, $\kappa = 5$, $\theta = 0.16$, $\sigma = 0.9$, and $\rho = 0.1$.

Table VII – American Put Prices and Performance Statistics for Option Pricing Methods Under The Heston Model and $v_0 = 0.25$

			S_0						
Method	Grid Size	Integration Points	8	9	10	11	12	Time (s)	RMSE
PSOR	(40,16,8)		2.0735	1.3266	0.7893	0.4438	0.2406	1.54	0.76%
	(60,32,66)		2.0764	1.3307	0.7929	0.4460	0.2414	36.50	0.40%
	(120,64,130)		2.0780	1.3331	0.7955	0.4479	0.2426	1.693.53	0.06%
	(240,128,258)		2.0782	1.3334	0.7958	0.4481	0.2427	69.166.69	0.03%
CS	(40,16,8)		2.0797	1.3325	0.7943	0.4474	0.2430	0.75	0.14%
	(60,32,66)		2.0770	1.3313	0.7935	0.4464	0.2417	15.43	0.32%
	(120,64,130)		2.0783	1.3334	0.7957	0.4481	0.2428	117.92	0.02%
	(240,128,258)		2.0784	1.3335	0.7959	0.4482	0.2427	922.42	0.02%
TP	(40,16,8)		2.0611	1.3063	0.7668	0.4265	0.2310	0.57	3.62%
	(60,32,66)		2.0725	1.3263	0.7888	0.4430	0.2397	4.88	0.92%
	(120,64,130)		2.0765	1.3310	0.7933	0.4462	0.2415	36.61	0.37%
	(240,128,258)		2.0777	1.3325	0.7947	0.4472	0.2420	340.47	0.20%
	(480,256,520)		2.0780	1.3330	0.7952	0.4476	0.2422	3.585.60	0.14%
FPI-CPU	(5, 5)	(100, 100)	2.0802	1.3361	0.7986	0.4506	0.2446	4.19	0.44%
	(5, 5)	(3200, 100)	2.0796	1.3355	0.7980	0.4499	0.2440	180.18	0.30%
	(20, 5)	(100, 100)	2.0786	1.3342	0.7967	0.4491	0.2436	19.88	0.24%
	(20, 5)	(3200, 100)	2.0780	1.3336	0.7961	0.4485	0.2430	178.10	0.11%
	(10, 10)	(100, 100)	2.0796	1.3350	0.7974	0.4496	0.2439	22.39	0.26%
	(10, 10)	(3200, 100)	2.0790	1.3344	0.7968	0.4490	0.2433	190.51	0.13%
	(40, 10)	(100, 100)	2.0791	1.3344	0.7968	0.4491	0.2436	353.51	0.17%
	(40, 10)	(3200, 100)	2.0785	1.3338	0.7962	0.4485	0.2430	514.43	0.04%
	(20, 20)	(100, 100)	2.0792	1.3346	0.7969	0.4492	0.2437	308.16	0.20%
	(20, 20)	(3200, 100)	2.0786	1.3340	0.7963	0.4486	0.2431	463.24	0.06%
	(40, 20)	(100, 100)	2.0791	1.3344	0.7968	0.4491	0.2436	1.387.85	0.17%
	(40, 20)	(3200, 100)	2.0785	1.3338	0.7962	0.4485	0.2430	1.543.09	0.04%
FPI-GPU	(5, 5)	(100, 100)	2.0802	1.3361	0.7986	0.4505	0.2446	0.37	0.44%
	(5, 5)	(3200, 100)	2.0796	1.3355	0.7980	0.4499	0.2440	4.28	0.30%
	(20, 5)	(3200, 100)	2.0780	1.3336	0.7961	0.4485	0.2430	5.74	0.05%
	(10, 10)	(3200, 100)	2.0790	1.3344	0.7968	0.4490	0.2433	5.79	0.13%
	(40, 10)	(3200, 100)	2.0785	1.3338	0.7962	0.4485	0.2430	28.85	0.04%
	(100, 10)	(3200, 100)	2.0785	1.3337	0.7961	0.4484	0.2429	158.75	0.02%
	(20, 20)	(3200, 100)	2.0786	1.3340	0.7963	0.4486	0.2431	29.99	0.06%
	(40, 20)	(3200, 100)	2.0785	1.3338	0.7962	0.4485	0.2430	107.55	0.04%
	(60, 20)	(3200, 100)	2.0785	1.3338	0.7961	0.4484	0.2429	234.27	0.03%
	(80, 20)	(3200, 100)	2.0785	1.3338	0.7961	0.4484	0.2429	419.95	0.03%
	(100, 20)	(3200, 100)	2.0785	1.3337	0.7961	0.4484	0.2429	651.64	0.02%
True Value			**2.0784**	**1.3336**	**0.7960**	**0.4483**	**0.2428**		

Note – The table reports American put prices and performance statistics for numerical methods under the Heston's stochastic volatility option pricing model. FPI was implemented using an adaptive integration method based on the Gauss-Kronrod quadrature. Column (2) reports the grid size used for each method. For PSOR, CS and TP, grid size refers to (n_{S_0}, n_{v_0}, n_T), as these methods calculate put option prices for each initial stock price S_0 and variance v_0 at a defined time to maturity $T = 0.25$, whereas for the FPI method we refer to the early exercise policy grid size defined by (n_τ, n_v), as this method calculates an option price given S_0 and v_0 by solving the optimal exercise policy $S_c(\tau, v)$. Column (3) reports the number of discretization points for τ and v on the policy grid used in the last step of FPI, as this method uses spline algorithms to improve the calculation of put prices. Columns (4) to (8) report put prices for each initial stock price and $v_0 = 0.25$. Column (9) reports time in seconds and column (10) refers to the root mean square error. True value makes reference to the put prices used by Ikonen & Toivanen (2007*b*) as benchmarks. K = 10, T = 3 months, r = 0.1 per annum, $\kappa = 5$, $\theta = 0.16$, $\sigma = 0.9$, and $\rho = 0.1$.

Table VIII – American Put Prices and Performance Statistics for the combined TP and FPI Option Pricing Method under The Heston Model for $v_0 = 0.0625$

		S_0						
Method	Grid Size	8	9	10	11	12	Time (s)	RMSE
TP-FPI-CPU	(40,16,8) - (5,5)	2.0000	1.0928	0.4956	0.2038	0.0836	3.05	3.13%
	(60,32,66) - (5,5)	2.0000	1.1006	0.5120	0.2091	0.0807	3.48	1.41%
	(120,64,130) - (5,5)	2.0000	1.1037	0.5154	0.2105	0.0806	7.01	1.12%
	(240,128,258) - (5,5)	2.0000	1.1045	0.5160	0.2106	0.0803	32.03	1.21%
	(480,256,520) - (5,5)	2.0000	1.1048	0.5161	0.2106	0.0802	280.96	1.24%
	(40,16,8) - (20,20)	2.0000	1.0934	0.4959	0.2039	0.0837	290.52	3.10%
	(60,32,66) - (20,20)	2.0000	1.1021	0.5135	0.2101	0.0813	290.96	1.05%
	(120,64,130) - (20,20)	2.0000	1.1054	0.5174	0.2120	0.0814	294.55	0.55%
	(240,128,258) - (20,20)	2.0000	1.1064	0.5185	0.2125	0.0814	318.33	0.47%
	(480,256,520) - (20,20)	2.0000	1.1068	0.5188	0.2127	0.0814	507.60	0.42%
	(40,16,8) - (100,10)	2.0000	1.0934	0.4959	0.2039	0.0837	2.111.93	3.10%
	(60,32,66) - (100,10)	2.0000	1.1022	0.5137	0.2102	0.0813	2.112.25	1.02%
	(120,64,130) - (100,10)	2.0000	1.1056	0.5177	0.2122	0.0815	2.115.51	0.47%
	(240,128,258) - (100,10)	2.0000	1.1067	0.5188	0.2127	0.0815	2.136.21	0.36%
	(480,256,520) - (100,10)	2.0000	1.1071	0.5193	0.2130	0.0816	2.310.21	0.28%
TP-FPI-GPU	(40,16,8) - (5,5)	2.0000	1.0928	0.4956	0.2038	0.0836	0.44	3.13%
	(60,32,66) - (5,5)	2.0000	1.1006	0.5120	0.2091	0.0807	0.87	1.41%
	(120,64,130) - (5,5)	2.0000	1.1037	0.5154	0.2105	0.0806	4.39	1.12%
	(240,128,258) - (5,5)	2.0000	1.1045	0.5160	0.2106	0.0803	29.41	1.21%
	(480,256,520) - (5,5)	2.0000	1.1048	0.5161	0.2106	0.0802	278.34	1.24%
	(40,16,8) - (20,20)	2.0000	1.0934	0.4959	0.2039	0.0837	25.83	3.10%
	(60,32,66) - (20,20)	2.0000	1.1021	0.5135	0.2101	0.0813	26.27	1.05%
	(120,64,130) - (20,20)	2.0000	1.1054	0.5174	0.2120	0.0814	29.86	0.55%
	(240,128,258) - (20,20)	2.0000	1.1064	0.5185	0.2125	0.0814	53.65	0.47%
	(480,256,520) - (20,20)	2.0000	1.1068	0.5188	0.2127	0.0814	242.92	0.42%
	(40,16,8) - (100,10)	2.0000	1.0934	0.4959	0.2039	0.0837	153.83	3.10%
	(60,32,66) - (100,10)	2.0000	1.1022	0.5137	0.2102	0.0813	154.16	1.02%
	(120,64,130) - (100,10)	2.0000	1.1056	0.5177	0.2122	0.0815	157.42	0.47%
	(240,128,258) - (100,10)	2.0000	1.1067	0.5188	0.2127	0.0815	178.12	0.36%
	(480,256,520) - (100,10)	2.0000	1.1071	0.5193	0.2130	0.0816	352.12	0.28%
True Value		**2.0000**	**1.1076**	**0.5200**	**0.2137**	**0.0820**		

Note – The table reports American put prices and performance statistics for the TP-FPI method under the Heston's stochastic volatility option pricing model. TP-FPI-CPU and TP-FPI-GPU are the CPU and GPU based implementations. Column (2) reports the grid size used in each calculation. Grid size refers to (n_{S_0}, n_{v_0}, n_T) and (n_τ, n_v) grids, as this combined method calculates put option prices for each initial stock price S_0 and variance v_0 at a defined time to maturity $T = 0.25$, based on the optimal early exercise policy $S_c(\tau, v)$. Columns (3) to (7) report put prices for each initial stock price and $v_0 = 0.0625$. Column (9) reports time in seconds and column (10) refers to the root mean square error. True value makes reference to the put prices used by Ikonen & Toivanen (2007*b*) as benchmarks. Strike price K = 10, time-to-maturity T = 3 months, risk-free rate of interest r = 0.1 per annum, mean rate of reversion $\kappa = 5$, long-term mean variance $\theta = 0.16$, volatility of the volatility process $\sigma = 0.9$, and correlation $\rho = 0.1$.

Table IX – American Put Prices and Performance Statistics for the combined TP and FPI Option Pricing Method under The Heston Model for $v_0 = 0.25$

		S_0						
Method	Grid Size	8	9	10	11	12	Time (s)	RMSE
TP-FPI-CPU	(40,16,8) - (5,5)	2.0670	1.3123	0.7710	0.4288	0.2322	1.99	3.18%
	(60,32,66) - (5,5)	2.0741	1.3268	0.7886	0.4423	0.2390	2.42	1.03%
	(120,64,130) - (5,5)	2.0759	1.3295	0.7912	0.4441	0.2398	5.94	0.75%
	(240,128,258) - (5,5)	2.0764	1.3301	0.7918	0.4444	0.2398	30.96	0.72%
	(480,256,520) - (5,5)	2.0766	1.3303	0.7919	0.4445	0.2398	279.89	0.72%
	(40,16,8) - (20,20)	2.0671	1.3125	0.7713	0.4290	0.2324	306.39	3.14%
	(60,32,66) - (20,20)	2.0752	1.3285	0.7903	0.4438	0.2402	306.83	0.75%
	(120,64,130) - (20,20)	2.0771	1.3315	0.7935	0.4462	0.2414	310.42	0.37%
	(240,128,258) - (20,20)	2.0777	1.3324	0.7945	0.4469	0.2418	334.21	0.26%
	(480,256,520) - (20,20)	2.0779	1.3327	0.7948	0.4472	0.2419	523.47	0.21%
	(40,16,8) - (100,10)	2.0671	1.3125	0.7713	0.4290	0.2324	2.133.14	3.14%
	(60,32,66) - (100,10)	2.0753	1.3286	0.7904	0.4439	0.2403	2.133.46	0.73%
	(120,64,130) - (100,10)	2.0772	1.3317	0.7938	0.4464	0.2416	2.136.73	0.32%
	(240,128,258) - (100,10)	2.0778	1.3327	0.7949	0.4473	0.2420	2.157.42	0.19%
	(480,256,520) - (100,10)	2.0781	1.3331	0.7953	0.4476	0.2423	2.331.42	0.13%
TP-FPI-GPU	(40,16,8) - (5,5)	2.0670	1.3123	0.7710	0.4288	0.2322	0.42	3.18%
	(60,32,66) - (5,5)	2.0741	1.3268	0.7886	0.4423	0.2390	0.85	1.03%
	(120,64,130) - (5,5)	2.0759	1.3295	0.7912	0.4441	0.2398	4.38	0.75%
	(240,128,258) - (5,5)	2.0764	1.3301	0.7918	0.4444	0.2398	29.40	0.72%
	(480,256,520) - (5,5)	2.0766	1.3303	0.7919	0.4445	0.2398	278.33	0.72%
	(40,16,8) - (20,20)	2.0671	1.3125	0.7713	0.4290	0.2324	26.11	3.14%
	(60,32,66) - (20,20)	2.0752	1.3285	0.7903	0.4438	0.2402	26.54	0.75%
	(120,64,130) - (20,20)	2.0771	1.3315	0.7935	0.4462	0.2414	30.13	0.37%
	(240,128,258) - (20,20)	2.0777	1.3324	0.7945	0.4469	0.2418	53.92	0.26%
	(480,256,520) - (20,20)	2.0779	1.3327	0.7948	0.4472	0.2419	243.19	0.21%
	(40,16,8) - (100,10)	2.0671	1.3125	0.7713	0.4290	0.2324	154.98	3.14%
	(60,32,66) - (100,10)	2.0753	1.3286	0.7904	0.4439	0.2403	155.31	0.73%
	(120,64,130) - (100,10)	2.0772	1.3317	0.7938	0.4464	0.2416	158.57	0.32%
	(240,128,258) - (100,10)	2.0778	1.3327	0.7949	0.4473	0.2420	179.26	0.19%
	(480,256,520) - (100,10)	2.0781	1.3331	0.7953	0.4476	0.2423	353.27	0.13%
True Value		**2.0784**	**1.3336**	**0.7960**	**0.4483**	**0.2428**		

Note – The table reports American put prices and performance statistics for the TP-FPI method under the Heston's stochastic volatility option pricing model. TP-FPI-CPU and TP-FPI-GPU are the CPU and GPU based implementations. Column (2) reports the grid size used in each calculation. Grid size refers to (n_{S_0}, n_{v_0}, n_T) and (n_τ, n_v) grids, as this combined method calculates put option prices for each initial stock price S_0 and variance v_0 at a defined time to maturity $T = 0.25$, based on the optimal early exercise policy $S_c(\tau, v)$. Columns (3) to (7) report put prices for each initial stock price and $v_0 = 0.25$. Column (9) reports time in seconds and column (10) refers to the root mean square error. True value makes reference to the put prices used by Ikonen & Toivanen (2007*b*) as benchmarks. Strike price K = 10, time-to-maturity T = 3 months, risk-free rate of interest r = 0.1 per annum, mean rate of reversion $\kappa = 5$, long-term mean variance $\theta = 0.16$, volatility of the volatility process $\sigma = 0.9$, and correlation $\rho = 0.1$.