Journal of Mathematical Finance, 2019, 9, 616-636
https://www.scirp.org/journal/jmf
ISSN Online: 2162-2442
ISSN Print: 2162-2434

Introducing the Power Series Method to Numerically Approximate Contingent Claim Partial Differential Equations

Gerald W. Buetow, James Sochacki

James Madison University, Harrisonburg, Virginia, USA
Email: gwb@bfrcservices.com, sochacjs@jmu.edu

How to cite this paper: Buetow, G.W. and Sochacki, J. (2019) Introducing the Power Series Method to Numerically Approximate Contingent Claim Partial Differential Equations. *Journal of Mathematical Finance*, **9**, 616-636.
https://doi.org/10.4236/jmf.2019.94031

Received: June 27, 2019
Accepted: October 22, 2019
Published: October 25, 2019

Abstract

We introduce a previously unused numerical framework for estimating the Black-Scholes partial differential equation. The approach, known as the Power Series Method (PSM), offers several advantages over traditional finite difference methods. Our objective is to highlight the advantages of the PSM over traditionally used numerical approximation approaches. To meet this we deploy a numerical approximation scheme to illustrate the PSM. The PSM is more stable than explicit methods and thus computationally more efficient. It is as accurate as hybrid approaches like Crank Nicolson and faster to compute. It is more accurate over a far wider spectrum of time steps. Finally, and importantly, it can be expressed analytically thus offering the capability of performing comparative statics in a far more stable and accurate environment. For a more complex application this last advantage may have wide implications in producing hedge ratios for synthetic replication purposes.

Keywords

Black-Scholes-Merton Options Pricing, Partial Differential Equations, Finite Difference Methods, Crank-Nicolson Method, Power Series Methods

1. Introduction

Numerical approaches to solving contingent claim based partial differential equations (PDE) have focused on finite difference method (FDM) approaches. Specifically, these have centered on explicit FDM (EFDM) approaches and Crank-Nicholson methodologies (CNM) and modification of these methods. Here we consider the approach introduced by Parker and Sochacki (1996, 2000) [1] [2] that uses a power series approximation based upon the Picard method.

The former methods provide solutions at grid points determined by the user. The method introduced by Parker and Sochacki provides a function that is defined over an entire time interval and is an accurate approximation to the solution to the contingent claim problem. Since one has a function that approximates the solution accurately, one can differentiate, integrate or use this function to solve at what time a certain value is achieved. These advantages become more apparent as the contingent claim becomes more complicated. The objective here is to illustrate the advantages relative to plain vanilla options in an effort to present why this approach may be a superior framework for complicated, nonlinear, or less stable contingent claim PDE's. In the literature this method is also known as automatic differentiation (Gofen [3] (2009) and Neidinger [4] (2010)), differential transform method (Mirzaee [5] (2011)), the Parker-Sochacki method (Stewart and Bair [6] (2009) and Rudmin [7] (1998)) and the power series method. In this article, we will use the terminology power series method (PSM) for the method that was discovered by Parker and Sochacki through Picard iteration.

Our paper follows Anwar and Andallah [8] (2018) and Cen and Le [9] (2011) by introducing PSM as an alternative and improved numerical scheme to estimate the Black Scholes PDE. The PSM approach offers advantages over traditional FDM that produces both operational and estimation improvements. Moreover, it enables the user to better approximate solutions to the PDE in far greater levels of precision. Also, PSM does not have to be limited to an FD mesh. We demonstrate this below in some examples that highlight these points. The highlights will be emphasized in our examples. We also point out that PSM is easy to program in both Matlab and Maple and is a natural extension of EFDM. In future papers we will show that it works for variable rates, dividends and volatility including those that produce a nonlinear PDE.

Since this method is not common in the mathematical financial community, we will demonstrate it through some examples and compare it to EFDM and CNM. As the reader goes through the examples, some important concepts should become clearer. One is that we are determining the coefficients of the power series solution to the differential equations by two approaches. One method is simpler and easier to program than the other. However, the second method is also valid and gives us the ability to do mathematical analysis on the differential equation, the solution and the properties of the approximate solution. By having a polynomial estimate, we are able to analyze the discontinuity in the derivative at the strike price and study oscillations in difference methods noted by many researchers, including Cen and Le [9]. One can also generate comparative statics commonly referred to as the Greeks using calculus. The polynomial is easily differentiated at any combination of underlying variables and thus offers a framework where analytics on the contingent claim valuation is greatly facilitated.

In this article, our goal is to show that the method is a powerful method for obtaining approximate solutions to contingent claims and is superior to discrete methods like EFDM and CNM in many cases. The method also allows thorough analysis of the solution and its properties.

2. Introduction of the PSM Approach to Estimating Differential Equations

In this section, we introduce the general approach of PSM to differential equations. This is necessary in order to develop the approach as it specifically applies to the contingent claim partial differential equation presented in section 3 of the paper. To introduce the framework we start with a simple asset, *V*, that increases over time (*t*) due to continuous compounding at a constant rate (*r*). The ordinary differential equation (ODE) describing this process is

$$V'(t) = rV(t),\ V(0) = V_0 \tag{1}$$

where V_0 is the initial value (IV) or initial condition. The solution to this ODE is $V(t) = V_0 e^{rt}$. Note that the solution depends on V_0, *r* and *t*. That is, we can write

$$V(t) = V_0 e^{rt} = V(t; r, V_0)$$

to highlight the dependence of the value of the solution on V_0, *r* and *t*.

In EFDM and CNM one discretizes time to approximate the solution to this problem. That is, one chooses specific time values at which an approximation for *V* will be obtained. We assume these times are $t_n = n\Delta t$ for $n = 1, 2, 3, \cdots, N$ where Δt is a fixed time interval and *N* is a counting number. Let W^n be our approximation to $V(n\Delta t)$.

The EFDM approximation to IV ODE (1) is determined from

$$\frac{W^{n+1} - W^n}{\Delta t} = rW^n.$$

Solving for the "future" gives

$$W^{n+1} = W^n + rW^n \Delta t.$$

That is, the approximation at t_{n+1} for *V* is given by W^n (approximation to $V(n\Delta t)$) plus Δt times the approximation of the slope $V' = rV$ at t_n which is rW^n. We emphasize this because one iteration of PSM gives this same result and PSM extends this result. In particular, note that

$$W^1 = W^0 + rW^0 \Delta t = V_0 + rV_0 \Delta t$$

since $W^0 = V_0$.

The CNM approximation for IV ODE (1) is obtained from

$$\frac{W^{n+1} - W^n}{\Delta t} = arW^{n+1} + (1-a) rW^n$$

which is a linear combination of the explicit and implicit FDM with $0 \le a \le 1$. For the PSM, note that if $a = 0$ the EFDM arises from CNM. To obtain W^{n+1} the approximation for $V((n+1)\Delta t)$ from CNM, one has to solve an algebraic equation. This will be highlighted and demonstrated in our numerical analysis below.

From calculus, it is well known that we can express the solution $V(t)$ for the IV ODE (1) in power series (PS) form as

$$V(t)=V_0\mathrm{e}^{rt}=\sum_{j=0}^{\infty}\frac{V_0}{j!}(rt)^j=\sum_{j=0}^{\infty}\frac{r^jV_0}{j!}t^j=\sum_{j=0}^{\infty}V_jt^j$$

where $V_j=\dfrac{r^jV_0}{j!}$ for $j=0,1,2,3,\cdots$ It is also well known that by choosing a large enough counting number J that $\sum_{j=0}^{J}V_jt^j$ will be as accurate an approximation as one desires to $V(t)=V_0\mathrm{e}^{rt}=V(t;r,V_0)$ over an interval $0\le t\le\Delta t$ for an appropriate time interval Δt. That is, one has a function of t that accurately approximates the true solution $V(t;r,V_0)$ on the interval $0\le t\le\Delta t$.

We now demonstrate two methods for obtaining the PS form for $V(t;r,V_0)$. Picard used the fact that the ODE (1) solves the integral equation (IE)

$$V(t)=V_0+\int_0^t rV(\tau)\mathrm{d}\tau$$

and vice versa by the Fundamental Theorem of Integral Calculus to generate a function solution. Picard's method was to define a sequence of functions that converge to $V(t;r,V_0)$, the solution to the ODE. First, choose $P^{[0]}(t)$ to be the constant function given by the IV, $P^{[0]}(t)=V_0$. Now define the remaining functions in the sequence by the IE as

$$P^{[k+1]}(t)=V_0+\int_0^t rP^{[k]}(\tau)\mathrm{d}\tau$$

for $k=0,1,2,3,\cdots$. To demonstrate, some iterates are presented.

$$P^{[1]}(t)=V_0+\int_0^t rP^{[0]}(\tau)\mathrm{d}\tau=V_0+\int_0^t rV_0\mathrm{d}\tau=V_0+rV_0t$$

$$P^{[2]}(t)=V_0+\int_0^t rP^{[1]}(\tau)\mathrm{d}\tau=V_0+\int_0^t r(V_0+rV_0\tau)\mathrm{d}\tau=V_0+rV_0t+\frac{r^2V_0}{2!}t^2$$

$$P^{[3]}(t)=V_0+\int_0^t rP^{[2]}(\tau)\mathrm{d}\tau=V_0+rV_0t+\frac{r^2V_0}{2!}t^2+\frac{r^3V_0}{3!}t^3.$$

One can now determine that

$$P^{[k]}(t)=\sum_{j=0}^{k}\frac{r^jV_0}{j!}t^j$$

which is the solution to the ODE (1).

The second method is to substitute $\sum_{j=0}^{\infty}V_jt^j$ into the ODE (1). Doing this leads to

$$V'(t)=\sum_{j=0}^{\infty}(j+1)V_{j+1}t^j=r\sum_{j=0}^{\infty}V_jt^j=rV(t)$$

$$\sum_{j=0}^{\infty}(j+1)V_{j+1}t^j=\sum_{j=0}^{\infty}rV_jt^j.$$

Equating like powers of t in this last equation gives the iterative sequence

$$(j+1)V_{j+1}=rV_j$$

or

$$V_{j+1} = \frac{r}{j+1} V_j \tag{2}$$

for $j = 0,1,2,3,\cdots$. Writing out a few of the iterates gives $V_1 = rV_0$, $V_2 = \frac{r}{2} V_1 = \frac{r^2}{2!} V$, $V_3 = \frac{r}{3} V_2 = \frac{r^3}{3!} V_0$ and so on. Again, one can determine that these are the coefficients of the PS for the solution to the ODE (1). The recurrence relation (2) is easily programmed in either a symbolic or numerical computing environment. Therefore, one can, in principle, generate as accurate a polynomial solution to the true solution as desired by generating enough coefficients using recurrence relation (2). That is, one code, as will be demonstrated in this article, can generate as high an order of accuracy as one desires.

One important item to note is that if $J = 1$ then the approximation for $V(t)$ using PSM is given by

$$\sum_{j=0}^{1} V_j t^j = V_0 + V_1 t$$

and the corresponding approximation to $V'(t)$ given by the derivative of the above expression is V_1. The PSM then gives the approximation

$$V_1 = r(V_0 + V_1 t) = rV_0 + rV_1 t$$

which means $V_1 = rV_0$ with error given by $rV_1 t$. Therefore, the first PSM iterate gives

$$V_0 + rV_0 t$$

as the approximation to the solution $V(t)$ over the interval $0 \le t \le \Delta t$. In particular, at $t = \Delta t$, this approximation is $V_0 + rV_0 \Delta t$ which we noted above is the EFDM approximation. This is an important result as we now can generate a single framework that includes PSM and EFDM as a special case ($J = 1$). If $J = 2$, one obtains

$$V_0 + V_1 t + V_2 t^2 = V_0 + rV_0 t + \frac{r^2}{2!} V_0 t^2$$

as the approximation of $V(t)$ over the interval $0 \le t \le \Delta t$. To extend to the time interval over the interval $\Delta t \le t \le 2\Delta t$, one updates V_0 to be $V_0 + rV_0 \Delta t + \frac{r^2}{2!} V_0 (\Delta t)^2$ (the approximation to $V(\Delta t)$) and then generates a new V_1, V_2 with this V_0 and obtains an approximation for $V(2\Delta t)$). One continues for any desired *N*. Of course, the process is similar for any *J*. The larger *J* is or the smaller Δt is the better the approximation will be. Warne P.G. *et al.* [10] use the Picard iterates and PS to derive an *a-priori* error bound for PSM using this fact. This is how a PSM approximation for a plain vanilla option will be derived in our numerical analysis below.

It is important to mention a few other points. Picard showed that his iterative method works for a large class of ODEs. That is, the Picard iterates will converge to the solution to the ODE for many ODEs. This is important for our purposes and why offering the Picard approach as a polynomial solution. Parker and Sochacki [1] used this fact to generate solutions to nonlinear ODEs. In fact, the second method also works for nonlinear ODEs which was shown by Parker and Sochacki and exploited to give convergence and error results by Carothers *et al.* [10] [11] [12]. The important point is that we now have two methods for developing algorithms and proving theorems for solutions to linear and nonlinear ODEs. Also we point out that since the two methods are computational and based off the ODE, they can compute either a symbolic solution which can be analyzed in terms of the parameters defining the ODE, like V_0 and r in ODE (1) or strictly numerically for visualization and numerical analysis. This offers tremendous advantages over the traditional FDM approaches as it enables us to work with the actual solution of the PDE not only discrete numerical approximations of the PDE. This is not a subtle distinction but a meaningful shift in how to understand the valuation dynamics of contingent claims. Now we can actually operate on a time continuum and almost any space dimension required with more accuracy. By operating in this framework we are able to more accurately understand the PDE dynamics and corresponding first and second order hedging or replication applications. This will be demonstrated for plain vanilla options using the Black-Scholes model in this article and on various other options, including nonlinear options in future papers. We now demonstrate the two methods on a nonlinear ODE in order to indicate how we will use PSM on nonlinear options in our future papers.

We modify the IV ODE (1) to

$$V'(t) = rV(t) - lV(t)^2, \quad V(0) = V_0 \tag{3}$$

for r, l positive. This dynamical system puts a bound on the growth of $V(t)$ for $V_0 < \frac{r}{l}$. The closed form analytic solution to this IV ODE is

$$V(t) = V(t; r, l, V_0) = \frac{rV_0 e^{rt}}{lV_0 e^{rt} - (lV_0 - r)}$$

Note that if $l = 0$, this is the solution to ODE (1). That is, $V(t; r, 0, V_0) = V_0 e^{rt}$. Determining the PS form of $V(t; r, l, V_0)$ is not an easy task. However, if one uses the two methods presented in our first example, it is straightforward. First, we note that if we have two PS

$$A = \sum_{j=0}^{\infty} A_j t^j, \quad B = \sum_{j=0}^{\infty} B_j t^j$$

then the PS product of A and B is given by Cauchy's formula

$$AB = \sum_{j=0}^{\infty} A_j t^j \sum_{j=0}^{\infty} B_j t^j = \sum_{j=0}^{\infty} \left(\sum_{i=0}^{j} A_i B_{j-i} \right) t^j.$$

This fact, allows us to solve nonlinear ODEs using either Picard iterates or solving for the coefficients in the PS form of the solution. For this example, this leads to

$P^{[0]}(t) = V_0$ and then

$$P^{[k+1]}(t) = V_0 + \int_0^t \left(rP^{[k]}(\tau) - lP^{[k]}(\tau)^2 \right) \mathrm{d}\tau$$

for $k = 0,1,2,3,\cdots$ for the Picard iterates. Again, writing out a few iterates produces

$$\begin{aligned} P^{[1]}(t) &= V_0 + \int_0^t \left(rP^{[0]}(\tau) - lP^{[0]}(\tau)^2 \right) \mathrm{d}\tau \\ &= V_0 + \int_0^t \left(rV_0 - lV_0^2 \right) \mathrm{d}\tau = V_0 + \left(rV_0 - lV_0^2 \right) t \end{aligned}$$

$$\begin{aligned} P^{[2]}(t) &= V_0 + \int_0^t \left(rP^{[1]}(\tau) - lP^{[1]}(\tau)^2 \right) \mathrm{d}\tau \\ &= V_0 + \int_0^t \left(r\left(V_0 + \left(rV_0 - lV_0^2\right)\tau\right) - k\left(r\left(V_0 + \left(rV_0 - lV_0^2\right)\tau\right)\right)^2 \right) \mathrm{d}\tau \\ &= V_0 + \left(rV_0 - lV_0^2\right)t + \frac{V_0\left(lV_0 - r\right)\left(2lV_0 - r\right)}{2} t^2 - \frac{lV_0^2\left(lV_0 - r\right)^2}{3} t^3 \end{aligned}$$

$$\begin{aligned} P^{[3]}(t) &= V_0 + \int_0^t \left(rP^{[2]}(\tau) - lP^{[2]}(\tau)^2 \right) \mathrm{d}\tau \\ &= V_0 + \left(rV_0 - lV_0^2\right)t + \frac{V_0\left(lV_0 - r\right)\left(2lV_0 - r\right)}{2!} t^2 \\ &\quad + \frac{V_0\left(r - lV_0\right)\left(6l^2V_0^2 - 6rlV_0 + l^2\right)}{3!} t^3 + \frac{lV_0^2\left(r - lV_0\right)^2\left(r - 2lV_0\right)}{3} t^4 \\ &\quad + \frac{lV_0^2\left(r - lV_0\right)^2\left(20rlV_0 - 3r^2 - 20l^2V_0^2\right)}{60} t^5 \\ &\quad + \frac{l^2V_0^3\left(r - lV_0\right)^3\left(2lV_0 - r\right)}{18} t^6 - \frac{l^3V_0^4\left(r - lV_0\right)^4}{63} t^7 \end{aligned}$$

We note that the Picard iterates are much more complicated than the Picard iterates in the linear IV ODE of our first example, but that a pattern is arising. Even for this complicated example, Picard showed that

$$P^{[k+1]}(t) = \sum_{j=0}^{\infty} V_j t^j$$

solves the IV ODE (3). Parker and Sochacki noted that

$$P^{[2]}(t) = P^{[1]}(t) + \text{terms higher in order than 2 in } t$$

$$P^{[3]}(t) = P^{[2]}(t) + \text{terms higher in order than 3 in } t$$

and in general that

$$P^{[k+1]}(t) = P^{[k]}(t) + \text{terms higher in order than } k \text{ in } t$$

not only for this IV ODE, but for a large class of nonlinear IV ODEs. Therefore, using a computer one can generate all the Picard iterates iteratively with a single program code.

Using the second method, one assumes

$$V(t) = \sum_{j=0}^{\infty} V_j t^j$$

and generates a recurrence relation like in Example 1 that gives all the coefficients in the PS for $V(t;r,l,V_0)$ using Cauchy's formula for the product of PS. Doing this, one has

$$V'(t)=\sum_{j=0}^{\infty}(j+1)V_{j+1}t^j=r\sum_{j=0}^{\infty}V_jt^j-l\left(\sum_{j=0}^{\infty}V_jt^j\right)^2=rV(t)-lV(t)^2.$$

Therefore, using Cauchy's formula we have that

$$\sum_{j=0}^{\infty}(j+1)V_{j+1}t^j=r\sum_{j=0}^{\infty}V_jt^j-l\left(\sum_{j=0}^{\infty}V_jt^j\right)^2=\sum_{j=0}^{\infty}\left(rV_j-l\sum_{j=0}^{j}V_iV_{j-i}\right)t^j.$$

This gives the recurrence relation

$$V_{j+1}=\frac{\left(rV_j-l\sum_{i=0}^{j}V_iV_{j-i}\right)}{j+1}.$$

(We leave it to the reader to verify that V_1,V_2,V_3 given by this recurrence relation agrees with the coefficients in the Picard iterate $P^{[3]}$ above. Therefore, the solution to the IV ODE (3) is given by

$$V(t)=V(t;r,l,V_0)=\frac{rV_0\mathrm{e}^{rt}}{lV_0\mathrm{e}^{rt}-(lV_0-r)}=\sum_{j=0}^{\infty}\frac{\left(rV_j-l\sum_{i=0}^{j}V_iV_{j-i}\right)}{j+1}t^j.$$

Again, we can approximate $V(t)$ to any accuracy desired over an interval $0\le t\le\Delta t$ for some chosen Δt and counting number J using

$$\sum_{j=0}^{J}\frac{\left(rV_j-l\sum_{i=0}^{j}V_iV_{j-i}\right)}{j+1}t^j$$

on that interval. Note that the coefficients, once again, depend on V_0, r and l.

3. Black Scholes Plain Vanilla Option

Using the development in section 2 we can now turn our attention to option valuation. In a plain vanilla option, the value of the option, V depends on the price of an underlying asset, S and time, t. Therefore, one wants to determine $V(S,t)$ for any S and t for a reasonable range $0\le S\le S_M$ and $0\le t\le T$.

Now we turn to the Black-Scholes (1973) and Merton (1973) PDE

$$\frac{\partial}{\partial t}V+\frac{1}{2}\sigma^2S^2\frac{\partial^2}{\partial S^2}V+(r-q)S\frac{\partial}{\partial S}V-rV=F(S,t);\ V(S,T)=Q(s)$$

where V represents the value of the option, σ the standard deviation of returns on the underlying asset, S, and r is the risk-free borrowing and lending costs over the life of the contingent claim, and q is cash flow emanating from S. All rates are continuous. F is the value of the PDE at any given combination of asset value or time and is zero for simple contingent claims. In this article, we will assume F is zero, but in a future paper we will consider arbitrary F. $Q(S)$ is the expiration value of the option and can be a continuous or discontinuous (binary) relationship as well as path dependent. For our purposes here it is simply $\max(0,S-K)$ where K is the strike price.

Since there is a closed form solution of this IV PDE, we will use that closed form to compare the accuracy of EFDM, CNM and PSM. We have shown how one can determine V for a time interval. We now choose an increment value ΔS for the interval $0 \le S \le S_M$. We let $S_i = i\Delta S$ for $i = 0,1,2,3,\cdots,I$ so that $S_0 = 0$ and $S_I = I\Delta S = S_M$. We will determine a PS for $V(S_i,t)$ for each *i* as was done in Example 1.

Making the standard time variable change $\tau = T - t$ into the plain vanilla option IV PDE (with $F = 0$) gives the following IV PDE

$$\frac{\partial}{\partial \tau}V = \frac{1}{2}\sigma^2 S^2 \frac{\partial^2}{\partial S^2}V + (r-q)S\frac{\partial}{\partial S}V - rV;\ V(S,0) = Q(s) \quad (4)$$

where $V(S,t) = V(S,T-\tau)$.

Numerical routines to evaluate Equation (4) can be found in Duffie [13], Brennan and Schwartz [14], Company, Lodar and Pintos [15], Forsyth and Labahn [16], Tangman, Gopaul, and Bhuruth [17], Buetow and Sochacki [18] [19] [20], Wang and Forsyth [21], Anwar and Andallah [8], and Cen and Le [9] to name a few. Nonlinearities are introduced within the diffusion term. Leland [22], Boyle and Vorst [23], and Hoggard, Whaley, and Wilmott [24] introduce transaction costs that manifest nonlinearly through the diffusion term. Risk adjustment pricing is similarly introduced into the framework. See Kratka [25], Jandacka and Sevcovic [26] and Barles and Soner [27] who combine both concepts. Kutik and Mikula [28] and Lesmana and Wang [29] use various numerical approaches to solve these types of problems. Frey [30], Frey and Patie [31] and Frey and Stremme [32] introduce illiquidity. Liu and Young [33] extend [30] [31] [32] and Bakstein and Howison [34] combine illiquidity and transaction costs into the framework.

We will compare the EFDM, CNM and PSM for IV PDE (4) to the closed form solution. Again, since PSM gives us a function of *t*, we can compare PSM to the closed form solution at any *t* for a given S_i.

As for the two IV ODEs above, Equation (4) can be written equivalently as

$$V(S,\tau) = Q(s) + \int_0^\tau \left(\frac{1}{2}\sigma^2 S^2 \frac{\partial^2}{\partial S^2}V(S,\tau) + (r-q)S\frac{\partial}{\partial S}V(S,\tau) - rV(S,\tau)\right)\mathrm{d}\tau. \quad (5)$$

One can then build a Picard iterative process for V using this IE. The Picard iterates are given by

$$\begin{aligned} &P^{[k+1]}(S,\tau) \\ &= Q(s) + \int_0^\tau \left(\frac{1}{2}\sigma^2 S^2 \frac{\partial^2}{\partial S^2}P^{[k]}(S,\tau) + (r-q)S\frac{\partial}{\partial S}P^{[k]}(S,\tau) - rP^{[k]}(S,\tau)\right)\mathrm{d}\tau \end{aligned} \quad (6)$$

for $k = 0,1,2,\cdots$ together with

$$P^{[0]}(S,\tau) = V(S,0) = Q(S).$$

As pointed out earlier, the mathematical theory for the Picard iterates $P^{[k]}$ to approximate the solution $V(S,\tau)$ is well developed. Parker and Sochacki [1] [2] have shown that if the IV PDE is polynomial then computers can easily generate $P^{[k]}(S,\tau)$ iteratively for any *k*. Equation (4) falls into this category and thus can be solved using this framework. In fact, we can enter the Picard method into any symbolic package and obtain a symbolic approximation for $P^{[k]}(S,\tau)$ very easily. An example of this is included in the appendix using Maple. We emphasize that if one does this, one has an approximation to $V(S,\tau)$ for any (S,t) in the region of interest. As mentioned above this is extremely useful in performing comparative statics including hedging ratios for synthetic trading replication.

To demonstrate, we choose $Q(S)=S-K$ where *K* is a fixed value for *S*. With these, the first four Picard iterates are

$$P^{[1]}(S,\tau)=S-K-(r-q)(S+1)\tau P^{[2]}(S,\tau)$$
$$=S-K-(r-q)(S+1)\tau+\frac{1}{2}(r-q)^2(S+1)\tau^2P^{[3]}(S,\tau)$$
$$=S-K-(r-q)(S+1)\tau+\frac{1}{2}(r-q)^2(S+1)\tau^2-\frac{1}{6}r(r-q)^3(S+1)\tau^3P^{[4]}(S,\tau)$$
$$=S-X-(r-q)(S+1)\tau+\frac{1}{2}(r-q)^2(S+1)\tau^2-\frac{1}{6}(r-q)^3(S+1)\tau^3$$
$$+\frac{1}{4!}(r-q)^4(S+1)\tau^4$$

One sees the repeating pattern in these Picard iterates and realizes that $P^{[k]}(S,\tau)$ converges to

$$S-K+\sum_{j=0}^{\infty}\frac{1}{j!}(r-q)^j(S+1)\tau^j.$$

In options pricing, this would be the solution for $S>K$ under a simplification, but this presentation demonstrates the process. One should appreciate that this approximate solution is a polynomial and one can observe the dependence of the solution on *S* and τ for any (S,τ) of interest. Now that we have this solution, we can analyze the dependence of *V* on not only on *S*, but also σ, *q*, *r* and *K*.

Note that this method will generate a power series solution (polynomial) approximation for *V* for more complicated situations, including a polynomial *F*, an *r* that depends on *S* and/or *t* or even *V* and other *Q*. Thus, this approach can be used for the aforementioned nonlinearities. This allows one to do a thorough parameter analysis on *K*, *r*, *q*, σ, *F* and *Q* in a very wide-ranging set of applications. We will leave the more complex applications to future research. Here, we want to introduce the approach and compare it to the aforementioned common numerical approaches found in the literature.

Since the EFDM and CNM are discrete methods, we let

$$V(S,t)=V(S_i,\tau)=v_i(\tau)$$

and we let W_i^n be the approximation to $v_i(n\Delta t)$. We mention that PSM can be used in the non-discretized case as shown above and an example generated in Maple will be given below.

The discretized version of Equation (4) that we consider is

$$v_i'(\tau)=\frac{1}{2}\sigma^2 S_i^2 D^2 v_i(\tau)+(r-q)S_i D v_i(\tau)-r v_i(\tau) \tag{7}$$

with $v_i(0)=Q(S_i)$, D is a discretized approximation for $\frac{\partial}{\partial S}$ and D^2 is a discretized approximation for $\frac{\partial^2}{\partial S^2}$. For example, if we use the centered difference approximations

$$Dv_i(\tau)=\frac{v_{i+1}(\tau)-v_{i-1}(\tau)}{2\Delta S},\quad D^2 v_i(\tau)=\frac{v_{i+1}(\tau)-2v_i(\tau)+v_{i-1}(\tau)}{\Delta S^2}$$

then Equation (7) in discretized form becomes the ODE

$$\begin{aligned} v_i'(\tau)=&\frac{1}{2}\sigma^2 S_i^2\left(\frac{v_{i+1}(\tau)-2v_i(\tau)+v_{i-1}(\tau)}{\Delta S^2}\right)\\ &+(r-q)S_i\left(\frac{v_{i+1}(\tau)-v_{i-1}(\tau)}{2\Delta S}\right)-r v_i(\tau)\end{aligned} \tag{8}$$

with the initial conditions $v_i(0)=Q(S_i)$. Therefore, if $i=1,2,\cdots,I$ for some counting number *I*, we have a system of *I* IV ODEs that we can solve using PSM. That is, we assume

$$v_i(\tau)=\sum\nolimits_{j=0}^{J} v_{i,j}\tau^j \tag{9}$$

for each *I* and some user chosen *J*. The larger *J* is the more accurate an approximation $v_i(\tau)$ will be to $V(S_i,\tau)$. We can use the PSM in either a symbolic or numeric computing environment. This bypasses the restriction of using only time discrete approaches like EFDM and CNM. For our purposes here, we will contrast the PSM approach to these two numerical approaches to illustrate its strengths. We will use only the PS approach for PSM because of its ease to program. Since the first PSM polynomial (PSM 1) was shown to be equivalent to EFDM, we can generate EFDM with our PSM code.

The Picard method is a useful tool for analysis, $v_i(\tau)$ can be used to determine properties of $V(S_i,\tau)$ through differentiation, integration and equation solving. It is also important to remember that the $v_{i,j}$ are functions of σ, r, q and $Q(S_i)$. Therefore, they can be analyzed to determine the impact of these variables on the solution regarding sensitive dependence, stability, growth, etc. The PSM effectively offers a more robust framework for contingent claim valuation and a corresponding analysis.[1]

We will demonstrate that the PSM framework is a powerful setting to be applied to numerically understanding contingent claim valuations and more im-

[1]In other words, Equation (6) can be used to create polynomial estimates to the solution to Equation (4) over the entire valuation spectrum. These polynomials are then easily manipulated to produce estimates to all of the Greeks throughout the same spectrum. These will be more accurate than any FDM approach and also require a single computation. FDM requires computations over several input variations to obtain the Greeks or any kind of comparative static analysis. These repeated numerical computations may introduce a propagation of errors and may result in a compounding error. PSM does not suffer from this issue.

portantly the corresponding comparative statics relationships often used in practice by risk managers and traders. This is an important development since most contingent claim-based valuation models do not have analytical solutions so having another numerical approach like PSM may prove invaluable.

The EFDM approximation to Equation (8) that will be used is

$$W_i^{n+1} = W_i^n + \left(\frac{1}{2}\sigma^2 S_i^2\left(\frac{W_{i+1}^n - 2W_i^n + W_{i-1}^n}{\Delta S^2}\right) + (r-q)S_i\left(\frac{W_{i+1}^n - W_{i-1}^n}{2\Delta S}\right) - rW_i^n\right)\Delta\tau. \quad (9)$$

This form shows the slope term multiplied by $\Delta\tau$.

The CNM approximation to Equation (8) that will be used is

$$\begin{aligned} W_i^{n+1} = W_i^n &+ (1-a)\left(\frac{1}{2}\sigma^2 S_i^2\left(\frac{W_{i+1}^{n+1} - 2W_i^{n+} + W_{i-1}^{n+1}}{\Delta S^2}\right)\right. \\ &\left. + (r-q)S_i\left(\frac{W_{i+1}^{n+1} - W_{i-1}^{n+1}}{2\Delta S}\right) - rW_i^{n+1}\right)\Delta\tau \\ &+ a\left(\frac{1}{2}\sigma^2 S_i^2\left(\frac{W_{i+1}^n - 2W_i^n + W_{i-1}^n}{\Delta S^2}\right) + (r-q)S_i\left(\frac{W_{i+1}^n - W_{i-1}^n}{2\Delta S}\right) - rW_i^n\right)\Delta\tau. \end{aligned}$$

This form shows the slope term multiplied by $\Delta\tau$. The equation we code is

$$\begin{aligned} &-\left(\frac{a}{2}\left(i^2\sigma^2 - i(r-q)\right)\Delta\tau\right)W_{i-1}^{n+1} + \left(1 + a\Delta\tau\left(i^2\sigma^2 + (r-q)\right)\right)W_i^{n+1} \\ &-\left(\frac{a}{2}\left(i^2\sigma^2 + i(r-q)\right)\Delta\tau\right)W_{i+1}^{n+1} \\ &= (1-a)\left(\left(\frac{1}{2}i^2\sigma^2 + i(r-q)\right)\Delta\tau W_{i+1}^n + \left(1 - \left(i^2\sigma^2 + ir\right)\Delta\tau\right)W_i^n\right. \\ &\left. + \left(\frac{1}{2}i^2\sigma^2 - i(r-q)\right)\Delta\tau W_{i-1}^n\right). \end{aligned} \quad (10)$$

We solve this system of linear equations using a tri-diagonal matrix solver.

We compare EFDM, CNM and PSM for polynomials of various degrees with the closed form solution

$$V(S,t) = SN(d_1) - Ke^{-(r-q)(T-\tau)}N(d_2) \quad (11a)$$

for a call ($Q(S) = \max(S-K,0)$) option and

$$V(S,t) = Ke^{-(r-q)(T-\tau)}N(-d_2) - SN(-d_1) \quad (11b)$$

for a put ($Q(S) = \max(K-S,0)$) option where

$$d_1 = \ln\left(\frac{S}{K}\right) + \frac{\left(r-q+\frac{1}{2}\sigma^2\right)(T-\tau)}{\sigma\sqrt{T-\tau}}$$

$$d_2 = \ln\left(\frac{S}{K}\right) + \frac{\left(r-q-\frac{1}{2}\sigma^2\right)(T-\tau)}{\sigma\sqrt{T-\tau}}$$

to IV PDE (4). For PSM we will determine an accurate PS solution for the ODEs in Equation (8). We will demonstrate the power of being able to choose J without writing a new code, by varying J and providing the results when this is done. Again, we point out that one does not have to discretize in S to use PSM as was shown in our third example. We are doing this in this article for comparison with EFDM and CNM. We will also present some of the coefficients that PSM produces so that one can view the approximate polynomial solutions to the "true" answer. We also point out that polynomials are straightforward to evaluate efficiently on a computer using Horner's algorithm which we actually do here.

4. Motivation behind PSM

To our knowledge the PSM approach has not been extensively used within the finance literature, if at all. The advantages of PSM are many and several of these are particularly important in the contingent claim valuation framework. The neuroscientists Stewart and Bair [6] showed that PSM was competitive with the Runge-Kutta order 4 method (RK4) and the Bulirsch-Stoer method (BSM) and, in most cases, superior for Hodgkin-Huxley type differential equations, which are nonlinear. The astrophysicists Pruett, Rudmin and Lacy [35] showed that PSM was in many cases superior to RK4 and BSM for Newton's N-Body problem. Also, the astrophysics teams Nurminskii and Buryi [36] and Pruett, Ingham and Herman [37] and the neuroscientists Synkiewicz [38] and Yudanov, Shaaban, Melton and Reznik [39] have shown that it is in general easier to parallelize PSM codes and obtain close to linear speed up than other codes, including on graphical processing units.

As outlined above the PSM can be used symbolically and/or numerically to analyze or obtain numerical results. A symbolic polynomial estimate to the PDE solution directly offers the user the ability to compute not only the value of the contingent claim but also the corresponding comparative statics of that value to changes in the underlying inputs. These are well known as the Greeks in the finance literature. Once the symbolic solution is created no further numerical estimation is required within PSM. This is very different that any FDM approach. All FDM approaches would require repeated numerical estimations to compute the Greeks. The PSM doesn't require that and so errors introduced are not propagated further like in the FDM framework. We will illustrate this below.

The numerical PSM approach also subsumes the EFDM as a special case and thus can always be at least as accurate and stable as that approach. This enables the user to create a single PSM program and be able to easily use it when the EFDM is needed. Additionally, for similar reasons, the stability requirement of the PSM is never more constraining than the EFDM approach.

The CNM approach has the advantage of most always being stable but it is particularly inefficient computationally. CNM will also not have solutions for more complicated valuation applications that result in nonlinear frameworks. PSM is both more computationally efficient and in almost all cases more accurate. It also lends itself to far more complicated problems than the CNM framework.

Computational errors within the PSM framework also tend to remain of the same order regardless of the underlying values of the inputs whereas those associated with FDM can compound (grow) throughout the FD grid.

Money [40] showed that when applied discretely to PDEs, PSM is a generalization of the Lax-Wendroff method (LWM) that is commonly used and/or modified in computational fluid dynamics that involve nonlinearities. He determined stability conditions for some linear PDEs, including the heat equation. His work was for constant coefficient PDEs, but he did show its applicability to some nonlinear PDES. As far as we know, this is the first application of PSM to PDEs with non-constant coefficients. In the Black-Scholes-Merton options modeling PDE (BSMOMPDE) the coefficients depend on the variable being discretized. In future papers, we will demonstrate the relationship between PSM and LWM in both the linear and nonlinear case and derive stability and convergence conditions. Here our goal is to introduce the method, to show its efficacy and its potential strengths for variable coefficient PDEs and show that we can improve on the stability of the EFDM and be competitive with CNM. It is not an easy matter to extend CNM to nonlinear BSMOMPDEs. However, PSM extends naturally without having to solve systems of nonlinear equations to get a numerical answer.

For the discretizations we used above the EFDM for the BSMOPMPDE has an error that is $O(\Delta t)+O(\Delta S^2)$. The CNM for the BSMOPMPDE has an error that is $O(\Delta t^2)+O(\Delta S^2)$ which is an improvement over the explicit FDM. The PSM of order k (using polynomials of degree k) has an error that is $O(\Delta t^k)+O(\Delta S^2)$ which allows analyzing the error in using PSM on the BSMOPMPDE.

Finally, even the analytic solutions to the contingent claim BSMOPMPDE requires numerical estimation since it contains the cumulative distribution function. The distribution function requires a numerical approximation to the integration. This also introduces an error. It is possible that the PSM may more accurately approximate the true solution than this approximation of the analytic solution. Also, since one can increase the order of PSM, one has the advantage of studying convergence with respect to the grid size, the order of the polynomial used in PSM and the time step. This allows the potential of the computer giving as accurate an answer as possible.

5. Numerical Results Comparing PSM to FDM

Figure 1 and Figure 2 contrast the different numerical routines for a put option at the money ($S = K = 100$), with a volatility of 10%, an interest rate of 3%, a dividend rate of 1.5%, $dS = 1$, $S_{\min} = 0$, and $S_{\max} = 200$. Figure 1 presents the error relative to the closed form solution for $dt = 0.01$ and increasing expirations. With a small time step the EFDM's linear approximation is actually better than higher order PSM's and CNM. This is most likely due to the higher order terms in the PSM that effectively introduce round off error on the polynomial coefficients.

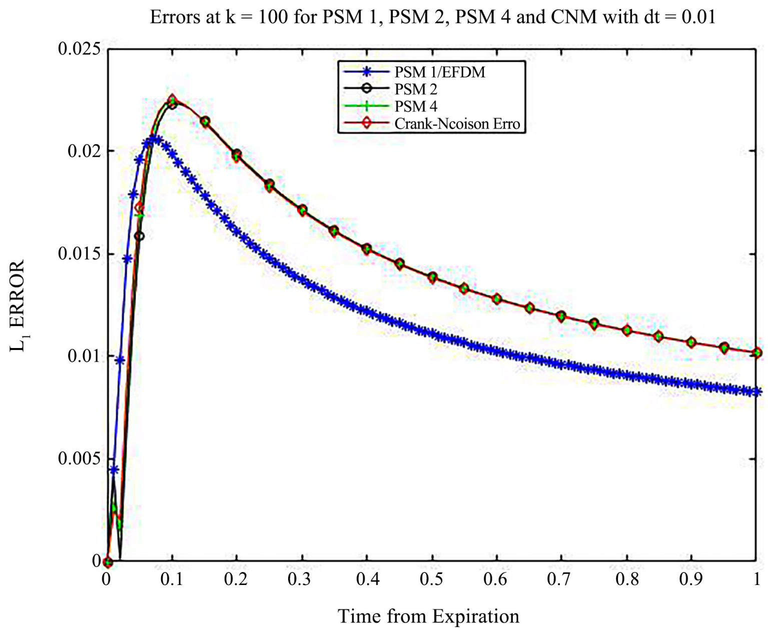

Figure 1. The absolute errors at $S = 100$ for various PSM and CNM with $dt = 0.01$.

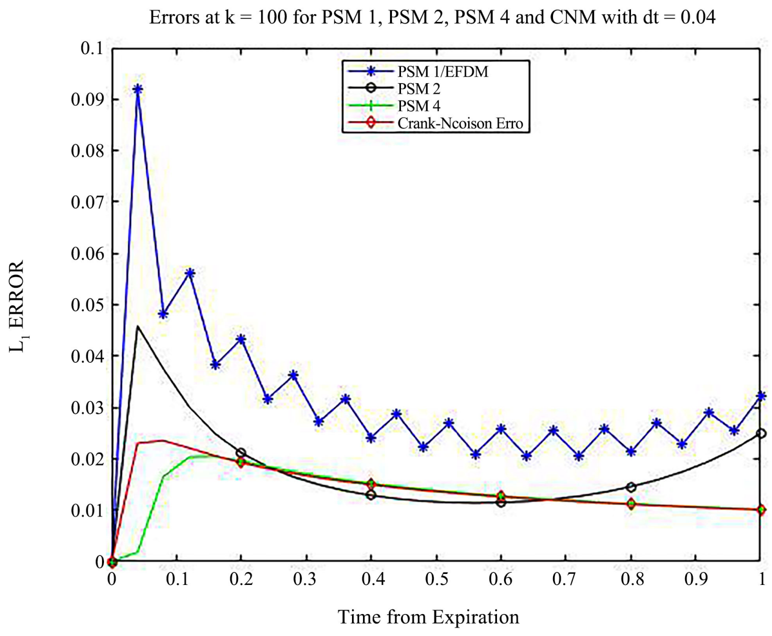

Figure 2. The absolute errors at $S = 100$ for various PSM and CNM with $dt = 0.04$.

This is the rare case where EFDM is the most accurate. Somewhat counter intuitively the increased terms in the PSM introduce a source of small round off errors when the time step is very small. However, since the EFDM is just a special case of the PSM we can determine easily at what time step does the PSM overtake the EFDM in accuracy. That is an important determination because as the time step increases the speed and stability of the PSM will dominate the tradeoff quite quickly. To illustrate this we increase the time step in the next example.

Figure 2 is similar to Figure 1 but for a slightly larger time step, $dt = 0.04$. Here we can plainly see that the EFDM is far less accurate than all the other methods and begins to oscillate. This is due to the instability of the EFDM. The second order PSM (PSM 2) and fourth order (PSM 4) is as accurate as the CNM and everywhere more accurate than the EFDM. As the time step increases and EFDM becomes less stable, and loses accuracy. PSM 4 is as accurate as CNM and close to expiration sometimes more accurate. This implies that the PSM can be computed more quickly than both CNM and EFDM without losing accuracy. This is particularly important in a trading environment or any application where resources are constrained. As we extend our research to non-linear applications this property will become even more important.

A Note on Discontinuity

Cen and Le (2011) highlight many of the difficulties within numerical routines in dealing with the non-differentiability at the strike price. In Figure 2 we can see the oscillation aspects to some degree within the EFDM framework. The general symbolic PSM method cannot be used on a non-differentiable initial condition. Here we approximate the put payoff with an infinitely differentiable function which gives us advantages over Cen and Le [9]. This is a relatively common issue throughout continuous mathematics. Whenever an application contains a discrete boundary condition difficulties will arise by the very nature of differentiability. Consequently, numerical approaches like FDM and PSM will have to deal with this issue. As we illustrate below the PSM easily addresses this problem that can negatively impact the FDM approaches.

One reason that PSM does not work in a symbolic environment for BSMOPMPDE is that the payoff functions are not differentiable at the strike price, K since the payoff for a call is given by $Q(S) = \max(S - K, 0)$ and for a put is given by $Q(S) = \max(K - S, 0)$ and to do PSM on a PDE the initial condition must be infinitely differentiable. In the Appendix, we give the second degree polynomial approximation to $V(S,\tau)$ that Maple generated (The fourth degree polynomial is too large to represent.) and the infinitely differentiable Q we used to approximately the put payoff even at the discrete boundary condition.

Figure 3 is a plot of the fourth PSM iterate given by Equation (6) for a call with $r = 0.03$, $q = 0.015$, $\sigma = 0.25$, $K = 100$ over the time interval $0 \le \tau \le 0.01$ ($T - 0.01 \le t \le T$) for $0 \le S \le 200$.

6. Conclusion

We have introduced the PSM framework to the finance literature and illustrated its advantages over traditional FDM. It is more accurate and offers a polynomial representation which can be further leveraged for both higher accuracy and more computational efficiencies. These advantages were illustrated through numerical examples. Specifically, as the time step increases (and computational

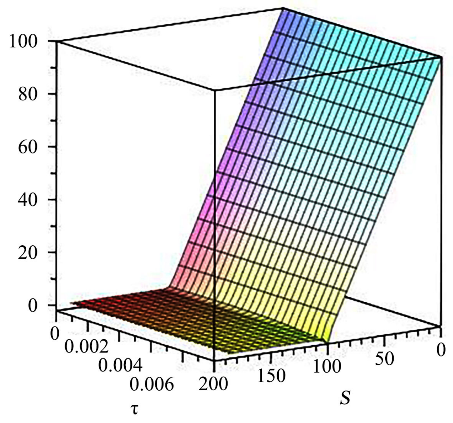

Figure 3. A Maple 3D Plot of a Put From a Symbolic PSM Fourth Degree Polynomial.

speed increases) the advantages of PSM over EFDM becomes more apparent. Additionally, the PSM is everywhere as accurate as the CNM without the computational limitations. The PSM easily addresses discrete boundary conditions as well. Finally, due to the polynomial representation of PSM, analysts can far more accurately approximate values and hedging parameters for almost any set of inputs without being constrained by a finite difference grid. Future research will include more complicated nonlinear applications that will highlight the advantages over traditional FDM even more. These future applications offer fertile ground for research. Both FDM and CNM approaches struggle with nonlinear applications. PSM offers a framework that doesn't suffer the same limitations. PSM is an excellent alternative to the numerical finance literature.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Parker, G. and Sochacki, J. (1996) Implementing the Picard Iteration. *Neural, Parallel, and Scientific Computation*, **4**, 97-112.

[2] Parker, G. and Sochacki, J. (2000) A Picard-McLaurin Theorem for Initial Value PDE's. *Abstract Analysis and Its Applications*, **5**, 47-63. https://doi.org/10.1155/S1085337500000063

[3] Gofen, A.M. (2009) The Ordinary Differential Equations and Automatic Differentiation Unified. *Complex Variables and Elliptic Equations*, **54**, 825-854. https://doi.org/10.1080/17476930902998852

[4] Neidinger, R.D. (2010) Introduction to Automatic Differentiation and Matlab Object-Oriented Programming. *SIAM Review*, **52**, 545-563. https://doi.org/10.1137/080743627

[5] Mirzaee, F. (2011) Differential Transform Method for Solving Linear and Nonlinear Systems of Ordinary Differential Equations. *Applied Mathematical Sciences*, **5**, 3465-3472.

[6] Stewart, R.D. and Bair, W. (2009) Spiking Neural Network Simulation: Numerical Integration with the Parker-Sochacki Method. *Journal of Computational Neuroscience*, **27**, 115-133. https://doi.org/10.1007/s10827-008-0131-5

[7] Rudmin, J.W. (1998) Application of the Parker-Sochacki Method to Celestial Mechanics. Technical Report, James Madison University, Harrisonburg.

[8] Anwar, M. and Andallah, L. (2018) A Study on Numerical Solution of Black-Scholes Model. *Journal of Mathematical Finance*, **8**, 372-381. https://doi.org/10.4236/jmf.2018.82024

[9] Cen, Z. and Le, A. (2011) A Robust and Accurate Finite Difference Method for a Generalized Black-Scholes Equation. *Journal of Computational and Applied Mathematics*, **235**, 3728-3733. https://doi.org/10.1016/j.cam.2011.01.018

[10] Warne, P.G., Warne, D.A., Sochacki, J.S., Parker, G.E. and Carothers, D.C. (2006) Explicit A-Priori Error Bounds and Adaptive Error Control for Approximation of Nonlinear Initial Value Differential Systems. *Computers & Mathematics with Applications*, **52**, 1695-1710. https://doi.org/10.1016/j.camwa.2005.12.004

[11] Carothers, D.C., Parker, G.E., Sochacki, J.S. and Warne, P.G. (2005) Some Properties of Solutions to Polynomial Systems of Differential Equations. *Electronic Journal of Differential Equations*, **40**, 1-17.

[12] Carothers, D.C., Lucas, S.K., Parker, G.E., Rudmin, J.D., Sochacki, J.S., Thelwell, R.J., Tongen, A. and Warne, P.G. (2012) Connections between Power Series Methods and Automatic Differentiation. *Recent Advancements in Algorithmic Differentiation*, **87**, 175-186. https://doi.org/10.1007/978-3-642-30023-3_16

[13] Duffie, D. (2006) Finite Difference Methods in Financial Engineering: A Partial Differential Equation Approach. John Willy & Sons, Hoboken. https://doi.org/10.1002/9781118673447

[14] Brennan, M. and Schwartz, E. (1978) Finite Difference Methods and Jump Processes Arising in the Pricing of Contingent Claims: A Synthesis. *Journal of Financial and Quantitative Analysis*, **13**, 461-474. https://doi.org/10.2307/2330152

[15] Company, R., Jodar, L. and Pintos, J.R. (2009) A Numerical Method for European Option Pricing with Transaction Costs Nonlinear Equation. *Mathematical and Computer Modelling*, **50**, 910-920. https://doi.org/10.1016/j.mcm.2009.05.019

[16] Forsyth, P. and Labahn, G. (2007) Numerical Methods for Controlled Hamilton-Jacobi-Bellman PDEs in Finance. *Journal of Computational Finance*, **11**, 1-44. https://doi.org/10.21314/JCF.2007.163

[17] Tangman, D., Gopaul, A. and Bhuruth, M. (2008) Numerical Pricing of Options Using Higher-Order Compact FD Schemes. *Journal of Computational and Applied Mathematics*, **218**, 270-280. https://doi.org/10.1016/j.cam.2007.01.035

[18] Buetow, G. and Sochacki, J. (1995) A Finite Difference Approach to the Pricing of Options Using Absorbing Boundary Conditions. *Journal of Financial Engineering*, **4**, 263-280.

[19] Buetow, G. and Sochacki, J. (1998) A More Accurate Finite Difference Approach to the Pricing of Contingent Claims. *Applied Mathematics and Computation*, **91**, 111-126. https://doi.org/10.1016/S0096-3003(97)10029-7

[20] Buetow, G. and Sochacki, J. (2000) The Tradeoffs between Alternative Finite Difference Techniques Used to Price Derivative Securities. *Applied Mathematics and Computation*, **115**, 177-190. https://doi.org/10.1016/S0096-3003(99)00141-1

[21] Wang, J. and Forsyth, P. (2008) Maximal Use of Central Differencing for Hamilton-Bellman PDE's in Finance. *SIAM Journal of Numerical Analysis*, **46**, 1580-1601.

https://doi.org/10.1137/060675186

[22] Leland, H. (1985) Option Pricing and Replication with Transactions Costs. *The Journal of Finance*, **40**, 1283-1301. https://doi.org/10.1111/j.1540-6261.1985.tb02383.x

[23] Boyle, P. and Vorst, T. (1992) Option Replication in Discrete Time with Transaction Costs. *The Journal of Finance*, **47**, 271-293. https://doi.org/10.1111/j.1540-6261.1992.tb03986.x

[24] Hoggard, T., Whaley, A. and Wilmott, P. (1994) Hedging Option Portfolios in the Presence of Transaction Costs. *Advances in Futures and Options Research*, **7**, 21-35.

[25] Kratka, M. (1998) No Mystery behind the Smile. *Risk*, **9**, 67-71.

[26] Jandacka, M. and Sevcovic, D. (2005) On the Risk-Adjusted Pricing-Methodology-Based Valuation of Vanilla Options and Explanation of the Volatility Smile. *Journal of Applied Mathematics*, **3**, 235-258. https://doi.org/10.1155/JAM.2005.235

[27] Barles, G. and Soner, H.M. (1998) Option Pricing with Transaction Costs and a Nonlinear Black-Scholes Equation. *Finance and Stochastics*, **2**, 369-397. https://doi.org/10.1007/s007800050046

[28] Kutik, P. and Mikula, K. (2011) Finite Volume Schemes for Solving Nonlinear Partial Differential Equations in Financial Mathematics. In: *Finite Volumes for Complex Applications VI and Perspectives*, Springer, Berlin, 643-651. https://doi.org/10.1007/978-3-642-20671-9_68

[29] Lesmana, D. and Wang, S. (2013) An Upwind Finite Difference Method for a Nonlinear Black-Scholes Equation Governing European Option Valuation under Transaction Costs. *Applied Mathematics and Computation*, **219**, 8811-8828. https://doi.org/10.1016/j.amc.2012.12.077

[30] Frey, R. (2000) Market Illiquidity as a Source of Model Risk in Dynamic Hedging. In: Gibson, R., Ed., *Model Risk*, Risk Publications, London, 125-136.

[31] Frey, R. and Patie, P. (2002) Risk Management for Derivatives in Illiquid Markets: A Simulation Study. In: Sandmann, K. and Schnbucher, P., Eds., *Advances in Finance and Stochastics*, Springer, Berlin, 137-159. https://doi.org/10.1007/978-3-662-04790-3_8

[32] Frey, R. and Stremme, A. (1997) Market Volatility and Feedback Effects from Dynamic Hedging. *Mathematical Finance*, **7**, 351-374. https://doi.org/10.1111/1467-9965.00036

[33] Liu, H. and Yong, J. (2005) Option Pricing with an Illiquid Underlying Asset Market. *Journal of Economic Dynamics and Control*, **29**, 2125-2156. https://doi.org/10.1016/j.jedc.2004.11.004

[34] Bakstein, D. and Howison, S. (2003) A Non-Arbitrage Liquidity Model with Observable Parameters for Derivatives. Working Paper, Oxford Centre for Industrial and Applied Mathematics, Oxford, 1-52.

[35] Pruett, C.D., Rudmin, J.W. and Lacy, J.M. (2003) An Adaptive N-Body Algorithm of Optimal Order. *Journal of Computational Physics*, **187**, 298-317. https://doi.org/10.1016/S0021-9991(03)00101-3

[36] Nurminskii, E. and Buryi, A. (2011) Parker-Sochacki Method for Solving Systems of Ordinary Differential Equations Using Graphics Processors. *Numerical Analysis and Applications*, **4**, 223. https://doi.org/10.1134/S1995423911030049

[37] Pruett, C.D., Ingham, W.H. and Herman, R.D. (2011) Parallel Implementation of an Adaptive and Parameter-Free n-Body Integrator. *Computer Physics Communications*, **182**, 1187-1198. https://doi.org/10.1016/j.cpc.2011.01.014

[38] Szynkiewicz, P. (2016) A Novel GPU-Enabled Simulator for Large Scale Spiking Neural Networks. *Journal Telecommunications and Information Technology*, **2**, 34-42.

[39] Yudanov, D., Shaaban, M., Melton, R. and Reznik, L. (2010) GPU-Based Simulation of Spiking Neural Networks with Real-Time Performance & High Accuracy. *The* 2010 *International Joint Conference on Neural Networks* (*IJCNN*), Barcelona, 18-23 July 2010, 1-8. https://doi.org/10.1109/IJCNN.2010.5596334

[40] Money, J.H. (2006) Variational Methods for Image Deblurring and Discretized Picard's Method. PhD Thesis, University of Kentucky, Lexington.

Appendix.

In Figure 1 and Figure 2 we showed the errors for PSM 1, PSM 2 and PSM 4 and CNM for different time steps. Below we show the fourth degree polynomial generated by PSM 4 at $S_{74} = 74$.

$$\begin{aligned} v_{74}^{[4]}(t) = {} & 26 - 1.89t + 0.036675t^2 - 0.000408374999996t^3 \\ & + 0.000003218905924t^4 \end{aligned}$$

It is important to remember that PSM generates a polynomial for each S_i and that from this polynomial we know that

$$v_{74}^{[3]}(t) = 26 - 1.89t + 0.036675t^2 - 0.000408374999996t^3$$

is generated by PSM 3 and

$$v_{74}^{[2]}(t) = 26 - 1.89t + 0.036675t^2$$

Is generated by PSM 2.

In Figure 3 we generated the results for a put option using a smooth approximation to the put payoff using Maple. We actually plotted the fourth degree polynomial that Maple generated. The second degree polynomial Maple generates for any $V(S,0) = Q(s), r, q$ and σ is

$$\begin{aligned} & \left(\frac{1}{4}\left(\sigma^2 S^2 \left(\sigma^2 \left(\frac{\mathrm{d}^2}{\mathrm{d}S^2} Q(S) \right) + 2\sigma^2 S \left(\frac{\mathrm{d}^3}{\mathrm{d}S^3} Q(S) \right) + \frac{\sigma^2 S^2 \left(\frac{\mathrm{d}^4}{\mathrm{d}S^4} Q(S) \right)}{2} \right.\right.\right. \\ & \left.\left. + 2(r-q)\left(\frac{\mathrm{d}^2}{\mathrm{d}S^2} Q(S) \right) + (r-q)S\left(\frac{\mathrm{d}^3}{\mathrm{d}S^3} Q(S) \right) - r\left(\frac{\mathrm{d}^2}{\mathrm{d}S^2} Q(S) \right) \right)\right) \\ & + \frac{1}{2}\left((r-q)S\left(\sigma^2 S \left(\frac{\mathrm{d}^2}{\mathrm{d}S^2} Q(S) \right) + \frac{\sigma^2 S^2 \left(\frac{\mathrm{d}^3}{\mathrm{d}S^3} Q(S) \right)}{2} + (r-q)\left(\frac{\mathrm{d}}{\mathrm{d}S} Q(S) \right) \right.\right. \\ & \left.\left. + (r-q)S\left(\frac{\mathrm{d}^2}{\mathrm{d}S^2} Q(S) \right) - r\left(\frac{\mathrm{d}}{\mathrm{d}S} Q(S) \right) \right)\right) \\ & \left. - \frac{r\left(\frac{\sigma^2 S^2 \left(\frac{\mathrm{d}^2}{\mathrm{d}S^2} Q(S) \right)}{2} + (r-q)S\left(\frac{\mathrm{d}}{\mathrm{d}S} Q(S) \right) - rQ(S) \right)}{2} \right) \tau^2 \\ & + \left(\frac{\sigma^2 S^2 \left(\frac{\mathrm{d}^2}{\mathrm{d}S^2} Q(S) \right)}{2} + (r-q)S\left(\frac{\mathrm{d}}{\mathrm{d}S} Q(S) \right) - rQ(S) \right)\tau + Q(S) \end{aligned}$$

We used $Q(S) = \ln(1 + \exp(S - 100)) - S + 100$ to approximate the put payoff for Figure 3. The maximum difference between this Q and the put payoff is ln2 and occurs at $S = K = 100$. A future study would involve approximating the payoff functions by even more accurate infinitely differentiable functions or at least functions that are k times continuously differentiable.