UPPSALA
UNIVERSITET

Option pricing using the discontinuous Galerkin method for time integration

Emil Larsson

Project in Computational Science: Report

January 2013

PROJECT REPORT

Institutionen för informationsteknologi

Option pricing using the discontinuous Galerkin method for time integration

Emil Larsson*
Uppsala University
Department of Scientific Computing

February 13, 2013

Abstract

The discontinuous Galerkin (dG) method for time integration of the parabolic problem obtained when pricing options using the Black-Scholes partial differential equation is presented. By decoupling of the large system of equations needed to be solved in each time step, the time of computation and memory requirement is significantly reduced. Even though there is a discontinuity in the first derivative of the initial condition, numerical experiments show convergence of order $2r + 1$ in time, where r is the order of the approximating polynomials in the dG method.

Keywords: discontinuous Galerkin, option pricing, decoupling, Black-Scholes model

1 Introduction

An option is a financial instrument that specifies a contract between two parties, giving the right to buy or sell an asset at a pre-specified price within a certain time period. The seller, often a bank, usually referred to as the writer, determines the terms and the price of the option. The other party is called the holder and pays the seller the market price, which is called the premium. The time at which the contract expires is called the time of maturity T, and at this time the holder can choose to either sell or buy the underlying asset depending on the type of the option. A call option gives the holder the right to buy the underlying asset for the pre-determined strike price K at time of maturity and similarly, a put option gives the holder the right to sell the underlying asset for the price K at time T. There are many different types of options but one of the most common is the European

*E-mail address: emil.larsson.2805@student.uu.se

option, where exercise is only possible at time of maturity. This is also the type of option that is considered in this paper.

At the time of maturity the holder of a European call option has two alternatives, either to exercise the option or to allow it expire unexercised. The choice depends on the spot price S of the underlying asset. Assuming the holder wishes to make a profit, the option is only exercised if $S > K$, after which the asset can immediately be sold at market price for a gain of $S - K$ per share. If $S < K$ the option will not be exercised since the asset can be bought on the market for the lower spot price S. In summary, the value of the option V at time of maturity, also called the payoff function $\Phi(S)$, can be seen in Figure 1 and is defined by

$$\Phi(S) = V(S_T, T) = \max(S_T - K, 0). \tag{1}$$

Similarly, a European put option is only exercised when $S < K$ at time of maturity, resulting in the payoff function

$$\Phi(S) = V(S_T, T) = \max(K - S_T, 0). \tag{2}$$

The objective of option pricing is to determine the arbitrage free price for every $t < T$ given this certain outcome at time of maturity. An arbitrage opportunity arises when there is a possibility to make a risk-free profit at zero cost.

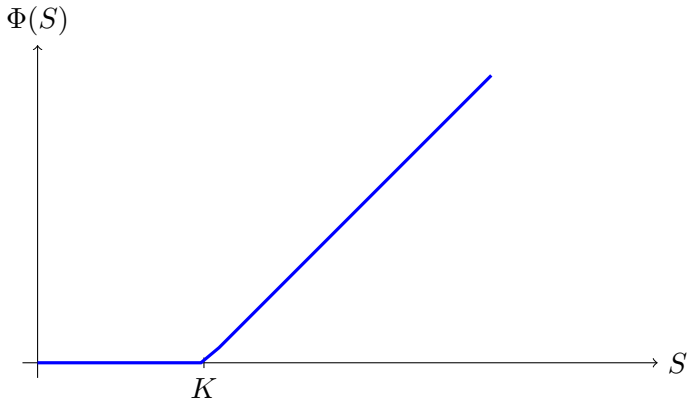

Figure 1: The value of a European call option at time of maturity, also called the payoff function.

A widely used method for option pricing is to solve the Black-Scholes equation

$$\begin{gathered} \frac{\partial V(S,t)}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V(S,t)}{\partial S^2} + rS\frac{\partial V(S,t)}{\partial S} - rV(S,t) = 0, \\ V(S_T, T) = \Phi(S), \end{gathered} \tag{3}$$

where σ is the volatility of S, r is the short rate of interest and $\Phi(S)$ is the payoff function which depends on the option type. The partial differential equation (3) was first introduced in 1973 by Fischer Black and Myron S. Scholes in [1] and Robert C. Merton in [5], where it is derived using Itô (stochastic) calculus, by first making a set of assumptions about the behavior of the financial market. It is often beneficial to make the transformation from a final value problem to a non dimensional initial value problem so that standard time integration techniques can be applied. A transformation which gives the desired properties, presented in [7], is given by

$$\begin{aligned} &S = Kx, \quad \hat{r} = \frac{r}{\sigma^2}, \quad \hat{t} = \frac{1}{2}\sigma^2(T-t), \\ &KP(x,\hat{t}) = V(S,t), \quad K\Psi(x) = \Phi(S). \end{aligned} \tag{4}$$

Applying this transformation on (3) results in the partial differential equation

$$\begin{aligned} \frac{\partial P}{\partial \hat{t}} &= 2\hat{r}x\frac{\partial P}{\partial x} + \hat{\sigma}^2 x^2 \frac{\partial^2 P}{\partial x^2} - 2\hat{r}P, \\ P(x,0) &= \Psi(x), \end{aligned} \tag{5}$$

and by letting $\mathcal{L}$ be the operator

$$\mathcal{L} = 2\hat{r}x\frac{\partial}{\partial x} + \hat{\sigma}^2 x^2 \frac{\partial^2}{\partial x^2} - 2\hat{r} \tag{6}$$

the equation (5) can be written as

$$\frac{\partial P}{\partial \hat{t}} = \mathcal{L}P. \tag{7}$$

Options contracts have been known for many centuries but it was not until 1973, when they started being issued in a standardized form and traded through a guaranteed clearinghouse at the Chicago Board Options Exchange, both trading activity and academic interest saw a massive increase. In fact, options have undergone a transformation from specialized and obscure securities to an essential component of the portfolios of not only large fund managers, but of ordinary individual investors. Essential ingredients of any successful modern investment strategy include the ability to generate income streams and reduce risk, as well as some level of speculation, all of which can be accomplished by effective use of options. Naturally, practitioners require an accurate method of pricing options and since today's market conditions evolve very rapidly, they also need to be able to obtain the price estimates quickly.

The goal of the work presented in this paper, which builds on results from [6], has thus been to develop a numerical method for pricing options, which is at the same time both accurate and efficient. The discontinuous Galerkin (dG) method for time integration has been used extensively in [9]

[10] [11] for parabolic problems similar to the one arising in option pricing, which made it a natural choice for further study.

The outline of this paper is as follows. In Section 2 we present an adaptive technique for the spatial semi-discretization using centered second order finite differences. In Section 3 the dG method for time integration is formulated. In Section 4 we present a technique for decoupling the large linear system of equations, obtained by the dG method, by choosing the temporal shape functions to be normalized Legendre polynomials. In Section 5 different methods for achieving convergence using the dG method are presented and the implementations of these are explained in Section 6. Finally, in Section 7 and 8 the results from the numerical experiments and the conclusions drawn from them are presented.

2 Space discretization

The spatial semi-discretization is made by using centered second order finite differences on a structured but not equidistant grid. To keep the number of grid points N to a minimum while still keeping the spatial discretization error at a pre-described level, an adaptive technique has been used. The technique, which is thoroughly explained in [7], can be broken down into three steps:

(i) Solve the problem once with a coarse spatial grid using large time steps, giving low accuracy.

(ii) Create a new grid in space, to give required accuracy, by estimating the local truncation error based on the result from the solution acquired using the coarse grid.

(iii) Solve the problem once again using the adapted grid in space.

This adaptive finite difference technique results in the system of ordinary partial differential equations

$$\frac{dP}{d\hat{t}} = AP, \tag{8}$$

where A is the finite difference discretization of the spatial operator $\mathcal{L}$. The large and sparse matrix A also includes the boundary conditions, which can be chosen in several different ways. Here, the condition used on all boundaries is

$$\frac{\partial^2 P(x,\hat{t})}{\partial x^2} = 0, \tag{9}$$

which implies that the price of the option is nearly linear with respect to the spot price at the boundaries. It is worth noting that having imperfect boundary conditions is typically not a great concern in financial problems

of this type since there often is a dominant diffusion term in the equation. This means that disturbances reflected back into the computational domain are smoothed out, reducing the destructive effect on the solution.

3 Time discretization

In this section we establish the dG formulation for a system of ordinary differential equations on the form

$$\begin{aligned} \dot{u}(t) &= Au(t) + f(t), \quad 0 \le t \le T \\ u(0) &= u_0 \end{aligned} \tag{10}$$

where A is the discretization of the spatial operator, u_0 the initial datum and f the forcing term. Let $\mathcal{M}$ be a partition of $(0,T)$ into M subintervals $\{I_m = (t_{m-1}, t_m)\}_{m=1}^{M}$ of size $k_m = t_m - t_{m-1}$. Define $\mathcal{P}^r(I_m)$ as the space of polynomials of degree r or less on the interval I_m and $\mathbb{U} = \{U : U_m \in \mathcal{P}^r(I_m)\}$ to be the finite element space containing the piecewise polynomials. In the dG method, which is thoroughly explained in [3], the finite element solution U is continuous within each time interval I_m, but may be discontinuous at the nodes $t_1, \ldots, t_{M-1}$. In [4], it is shown that the dG approach provides a unified framework to derive various time-stepping schemes, such as low order one-step methods, high order Runge-Kutta methods and multistep methods. Since the limiting values of the approximative solution from the left and the right at each time node t_m usually differ, we define the one-sided limits of a piecewise continuous function $u(t)$ as

$$\begin{aligned} u_m^+ &:= \lim_{s \to 0^+} u(t_m + s), \quad 0 \le m \le M-1 \\ u_m^- &:= \lim_{s \to 0^+} u(t_m - s), \quad 1 \le m \le M \end{aligned}$$

and the "jump" in $u(t)$ across t_m as

$$[u_m] := u_m^+ - u_m^-, \quad 1 \le m \le M-1.$$

This discontinuous behavior is illustrated in Figure 2. The dG method of degree r reads as follows: Find $U \in \mathbb{U}$, satisfying $U_0^- = u_0$, such that

$$\sum_{m=1}^{M} \int_{I_m} (\dot{U} - AU) w(t)\, dt + \sum_{m=1}^{M} [U_{m-1}] w(t_{m-1}) = \sum_{m=1}^{M} \int_{I_m} f w(t)\, dt \tag{11}$$

for all $w(t) \in \mathbb{U}$. In practice U can be computed in each interval

$$\int_{I_m} (\dot{U}_m - AU_m) w(t)\, dt + [U_{m-1}] w(t_{m-1}) = \int_{I_m} f w(t)\, dt \tag{12}$$

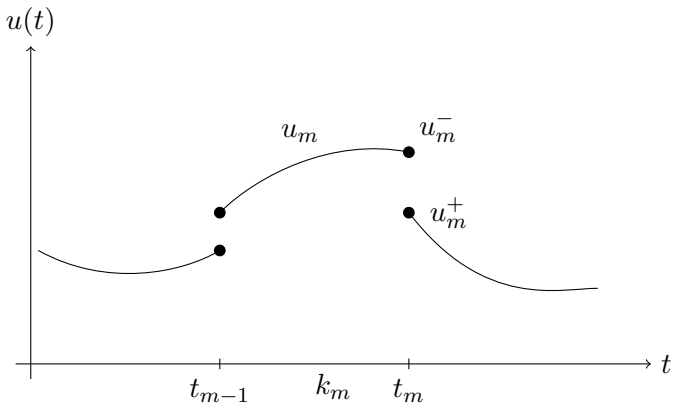

Figure 2: Illustration of a discontinuous function $u(t)$ with the one sided limits shown.

for $m = 1, \ldots, M$. Equation (12) results in having to solve a linear system of equations of size $(r+1)N$ in each of the M time steps. Let $\{\varphi\}_{j=0}^{r_m}$ be a basis of the polynomial space $\mathcal{P}_{r_m}(-1,1)$ and let time shape functions on time interval I_m be given by $\varphi_j \circ F_m^{-1}$, where the mapping $F_m : (-1,1) \to I_m$ is given by

$$t = F_m(\tau) = \frac{1}{2}(t_{m-1} + t_m) + \frac{1}{2}k_m\tau, \quad k_m = t_m - t_{m-1}, \quad \tau \in (-1,1).$$

Since the dG approximation U_m in each time interval I_m is in the polynomial space $\mathcal{P}_{r_m}(I_m)$, it can uniquely be expressed in the basis $\{\varphi\}_{j=0}^{r_m}$ as

$$U_m = \sum_{j=0}^{r_m} u_{m,j}(\varphi_j \circ F_m^{-1}). \tag{13}$$

Inserting (13) into (12), and letting the test function $w(t)$ be the basis $\{\varphi\}_{j=0}^{r_m}$, gives

$$\sum_{j=0}^{r_m}\sum_{i=0}^{r_m}\left(\int_{I_m}(\varphi_j' \circ F_m^{-1})(\varphi_i \circ F_m^{-1}) - A(\varphi_j \circ F_m^{-1})(\varphi_i \circ F_m^{-1})\,dt\right)u_{m,j} +$$
$$\sum_{j=0}^{r_m}\sum_{i=0}^{r_m}(\varphi_j(t_{m-1}) \circ F_m^{-1})(\varphi_i(t_{m-1}) \circ F_m^{-1})u_{m,j} -$$
$$\sum_{j=0}^{r_m}\sum_{i=0}^{r_m}(\varphi_j(t_{m-1}) \circ F_m^{-1})(\varphi_i(t_m) \circ F_m^{-1})u_{m-1,j} =$$
$$\sum_{i=0}^{r_m}\int_{I_m} f(\varphi_i \circ F_m^{-1})\,dt.$$

This can be expressed in a more compact form by using the mapping F_m^{-1} to transform the integration domain from I_m to $(-1,1)$, resulting in

$$\sum_{i,j=0}^{r_m} \left(C_{ij} - \frac{k_m}{2} G_{ij} \cdot A \right) u_{m,j} = \sum_{i=0}^{r_m} \left(\frac{k_m}{2} f_{m,i}^1 + f_{m,i}^2 \right), \tag{14}$$

where

$$f_{m,i}^1 := \int_{-1}^{1} (f \circ F_m) \varphi_i \, d\tau, \quad f_{m,i}^2 := \varphi_i(-1) \sum_{j=0}^{r_m} \varphi_j(1) u_{m-1,j}$$
$$C_{ij} := \int_{-1}^{1} \varphi_j' \varphi_i \, d\tau + \varphi_j(-1)\varphi_i(-1), \quad G_{ij} = \int_{-1}^{1} \varphi_j \varphi_i \, d\tau.$$

Dropping the subscript m for sake of readability and representing (14) in matrix form results in

$$\left(\mathbf{C} \otimes \mathbf{I} - \frac{k}{2} \mathbf{G} \otimes \mathbf{A} \right) \mathbf{u} = \frac{k}{2} \mathbf{f^1} + \mathbf{f^2}, \tag{15}$$

where $\otimes$ is the Kronecker product and $\mathbf{u}$ denotes the coefficient vector of U_m, that is

$$\mathbf{u} = \begin{pmatrix} \mathbf{u}_{m,0} \\ \vdots \\ \mathbf{u}_{m,r_m} \end{pmatrix}.$$

4 Decoupling

The coupled system (15) is costly to solve numerically in terms of computational time since a linear system of size $(r+1)N$ has to be solved in each time step. As observed in [9] however, it is possible to reduce the system (15) and solve $r+1$ linear systems of size N by choosing the temporal shape functions to be the normalized Legendre polynomials. Using these temporal shape functions given by $\varphi_i(\tau) = (i+1/2)^{1/2} L_i(\tau)$, with L_i denoting the i-th Legendre polynomial on $(-1,1)$, gives that $\mathbf{G} = \mathbf{I}$ and

$$C_{ij} = \sigma_{ij} \left(i + \frac{1}{2} \right)^{1/2} \left(j + \frac{1}{2} \right)^{1/2}, \quad \sigma_{ij} = \begin{cases} (-1)^{i+j} & \text{if } j < i \\ 1 & \text{otherwise} \end{cases}, \tag{16}$$

where $i, j = 0, \ldots, r$. According to [11], numerical experiments show that the matrix $\mathbf{C}$ is diagonalizable in $\mathbb{C}$ at least for $0 \leq r \leq 100$, and thus there exists a matrix $\mathbf{Q} \in \mathbb{C}^{(r+1)\times(r+1)}$ such that $\mathbf{Q}^{-1}\mathbf{C}\mathbf{Q} = \mathbf{\Lambda} = \operatorname{diag}(\lambda_0, \ldots, \lambda_r)$. Multiplying (15) by $\mathbf{Q}^{-1} \otimes \mathbf{I}$ from the left gives

$$\left(\mathbf{T} \otimes \mathbf{M} - \frac{k}{2} \mathbf{I} \otimes \mathbf{A} \right) \mathbf{w} = \mathbf{g} \tag{17}$$

with

$$\mathbf{w} := (\mathbf{Q}^{-1} \otimes \mathbf{I})\mathbf{u}, \quad \mathbf{g} := (\mathbf{Q}^{-1} \otimes \mathbf{I}) \left(\frac{k}{2}\mathbf{f^1} + \mathbf{f^2} \right). \tag{18}$$

The system (17) is block-diagonal and completely decouples into

$$\left(\lambda_j \mathbf{M} - \frac{k}{2}\mathbf{A} \right) \mathbf{w}_j = \mathbf{g}_j, \quad j = 0, \ldots, r. \tag{19}$$

Hence, in each time step we have to solve the $r+1$ linear systems in (19) of size N, which greatly reduces the time of computation compared to solving the whole system of size $(r+1)N$. Another advantage of this decomposition is that each of the decoupled systems can be assigned to a separate processor and solved in parallel. However, for large r, the matrix $\mathbf{Q}$ becomes ill-conditioned which may introduce a magnification in the round-off error. In practice, r is expected to vary only in a small range, say $0 \leq r \leq 10$, which according to [9] means that the decomposition does not affect the solution in a negative way.

5 Convergence in time

In addition to decoupling of the large system of equation resulting from the dG method, different techniques for obtaining convergence have been investigated. It is shown in [2] that the optimal convergence rate for the dG method is $2r + 1$. The choice of technique needed to obtain optimal convergence rates is thoroughly explained in [9] [10] [11] and depends on if the there exists time singularities in the solution, induced by non-smooth initial data or discontinuities in the forcing term f. In our option pricing problem the initial data is non-smooth because of the discontinuity in the payoff function at the strike price. This suggest the need to use a version of the dG method capable of converging despite singularities in the solution.

5.1 The *h*-version of the dG method

In the h-version of the dG method, convergence is achieved by refining the temporal mesh $\mathcal{M}$ by increasing the number of time steps in the interval $J = (0, T)$, while keeping the time approximation order r fixed for all time steps. However, if the solution is not arbitrarily smooth in time, a graded temporal mesh has to be used to recover the optimal convergence rate. It is shown in [9] that for some $0 < \theta \leq 1$, the temporal mesh $\mathcal{M}$ must be algebraically graded as

$$t_m = h\left(\frac{mT}{M} \right), \quad m = 0, \ldots, M, \quad h(t) = t^{(2r+3)/\theta} \tag{20}$$

to recover the optimal convergence rate where, as before, M is the number of time steps.

5.2 The *p*-version of the dG method

In the p-version of the dG method, convergence is obtained by increasing the polynomial degree r uniformly on all time steps while keeping the temporal mesh $\mathcal{M}$ fixed. According to [9], this results in exponential rates of convergence in r if the solution is analytic. If on the other hand the exact solution is not analytic in time, the hp-version has to be applied in order to still obtain exponential convergence.

5.3 The *hp*-version of the dG method

In the hp-version of the dG method, a geometrically refined temporal mesh is used together with linearly increasing approximation orders. A mesh $\{I_m\}_{m=1}^M$ in $J = (0,T)$ is geometric with M time steps $I_m = (t_{m-1}, t_m)$, $m = 1, \ldots, M$, with grading factor $\sigma \in (0,1)$ if

$$t_0 = 0, \quad t_m = T\sigma^{M-m}, \quad 1 \le m \le M,$$
$$k_m = \lambda t_{m-1}, \quad \lambda = (1-\sigma)/\sigma, \quad 2 \le m \le M.$$

We denote this geometric mesh, which is shown schematically in Figure 3, by $\mathcal{M}_{M,\sigma}$. A polynomial degree vector $\vec{r} = \{r_m\}_{m=1}^M$ is called linear with

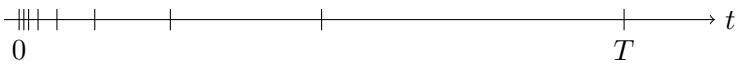

Figure 3: The partition of $(0,T)$ geometrically refined near $t = 0$.

slope $u < 0$ on the geometric mesh $\mathcal{M}_{M,\sigma}$ if $r_m = \lfloor \mu m \rfloor$ for $1 \le m \le M$. Figure 4 illustrates an example of such a linearly increasing degree vector. According to [10], combining a geometrically refined mesh $\mathcal{M}_{M,\sigma}$ with a linear approximation order vector makes it possible to recover exponential rate of convergence in r, even when singularities are present.

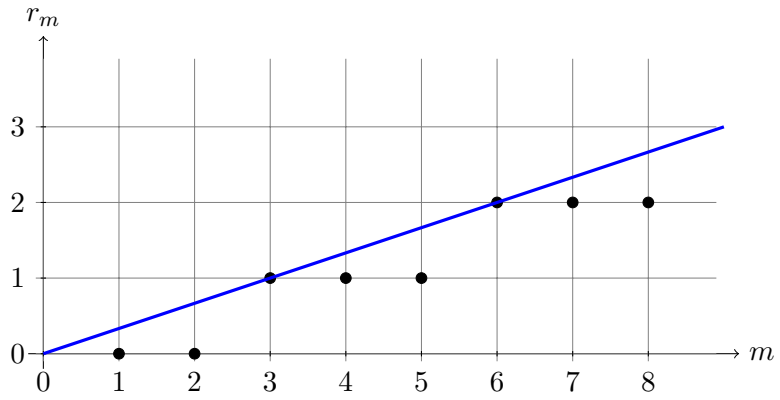

Figure 4: Example of a linearly increasing polynomial degree vector with slope $\mu = 1/3$ on 8 time steps, resulting in $\vec{r} = (0, 0, 1, 1, 1, 2, 2, 2)$.

6 Implementation

The implementation was made in MATLAB and the sparse linear systems of equations (19) that arise in each time step are solved using the *Generalized Minimal Residual Method* (GMRES), with incomplete LU-factorization as a pre-conditioner. The matrices $\mathbf{Q}$, $\mathbf{Q}^{-1}$ and $\mathbf{\Lambda}$, obtained in the decoupling procedure presented in Section 4, does not vary during the time stepping, which means they can be pre-computed and stored in memory for efficiency. To further reduce the time of computation, the Kronecker products in (18) have been replaced by the MATLAB function `bsxfun` after making some algebraic manipulations.

In Section 4 it is mentioned that the decoupled system (19) can be solved in parallel to further reduce the time of computation. An attempt at this was made but, since there is a significant amount of parallelization overhead when using the Parallel Computing Toolbox in MATLAB, no speedup was obtained. The time of computation is less than one second even when solving a fairly large problem in serial, making it unsuitable for parallelization in MATLAB. However, if the problem were to be implemented in for example C, one would most likely experience a significant speedup.

All of the serial experiments presented in the following section were run on a Macbook pro with a 2,3 GHz dual-core i5 processor and 4GB (2 x 2GB) of 1333MHz DDR3 SDRAM.

7 Numerical results

In this section we report on the numerical experiments which have been carried out to investigate the performance of the dG method for time inte-

gration. We consider a European call option with a short rate of interest r of 0.05, a volatility σ of 0.3, a strike price K of 30 and a time of maturity T set to 2.22 years.

The option pricing problem is defined for an arbitrarily large price of the underlying asset but for computational reasons we need to restrict the spatial domain. The general rule of thumb, also employed in [7] and [8], is to truncate the domain at four times the strike price K, that is, setting $S_{max} = 4K$. Furthermore, in options trading, the region of most interest is when the price of the underlying asset is close to the strike price. With this in mind we define the interval $[\frac{K}{3}, \frac{5K}{3}]$ and let this be the domain in which the error is calculated. The error reported is always the L_1-norm of the difference between the numerical and exact solution.

The tolerance for the local discretization error in the adaptive technique described in Section 2 is selected to be 10^{-8} in order to obtain a fine spatial mesh, so that the overall error is dominated by the error of the time discretization. This results in a total of $N = 12823$ grid points in space and a spatial error that is approximately $3 \cdot 10^{-7}$, which can be seen in Figure 7.

In Figure 5 we plot the relative errors against the number of time steps used for the dG solution with the order of the temporal polynomials r in the range $0 \leq r \leq 2$. The dashed lines corresponding to the slopes -1, -3 and -5 clearly show that the h-version of the dG method manages to obtain optimal convergence rates of $2r + 1$, even though there is a discontinuity in the initial condition.

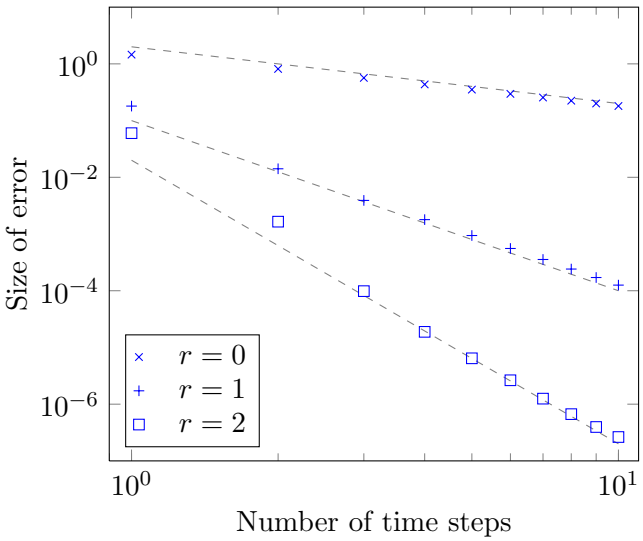

Figure 5: Convergence rates for the h-version of the dG method. The dashed lines give the slopes $-1, -3, -5$, which are the optimal convergence rate for the corresponding polynomial order r.

By decoupling the large system of equations that has to be solved in each time step in the dG method, a significant reduction in the time of computation can be obtained. This can be seen in Figure 6, where the time of computation is plotted for a number of time steps, both before and after decoupling.

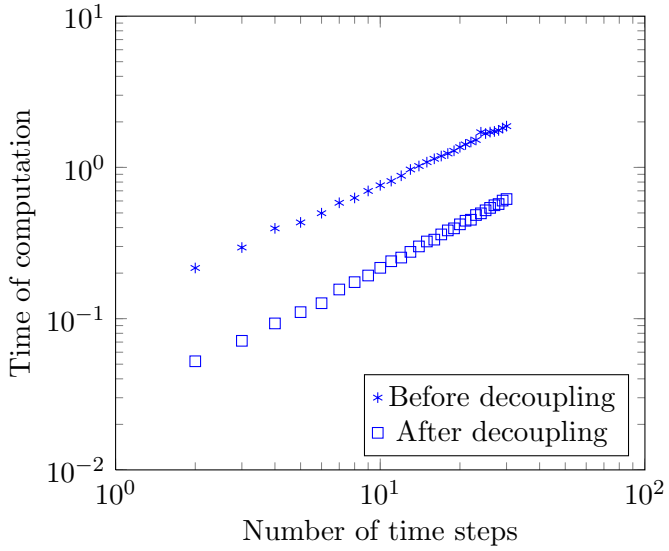

Figure 6: The time of computation for the dG method with $r = 1$ before and after the decoupling of the system of equations that has to be solved in each time step.

In Figure 7, a comparison is made between the dG method with decoupling of the resulting system of equations and the commonly used finite difference method *backwards differentiation formula of order 2* (BDF2). The time of computation is plotted as a function of the size of the error, and it is clear that for a given error, the dG method outperforms BDF2. It is also worth noticing that the dG method only requires very few time steps to obtain a very small temporal error, e.g. 3 time steps when $r = 2$ gives an error from the time discretization that is only about 10^{-4}. As mentioned, the spatial error is approximately $3 \cdot 10^{-7}$ which is the reason for the shape of the plot for the dG method with $r = 2$ in Figure 7. The total error cannot be reduced further, by increasing the number of time steps, due to the spatial error.

Even though optimal convergence rates were obtained with the h-version of the dG method, the hp-version, which can be seen in Figure 8, was investigated. We employ a geometrical grading factor $\sigma = 0.24$ and set the approximation order in time step m to $r_m = \lfloor \mu m \rfloor$ with a slope $\mu > 0$. The plot clearly shows exponential rates of convergence in r, with $\mu = 1.5$ as the best slope.

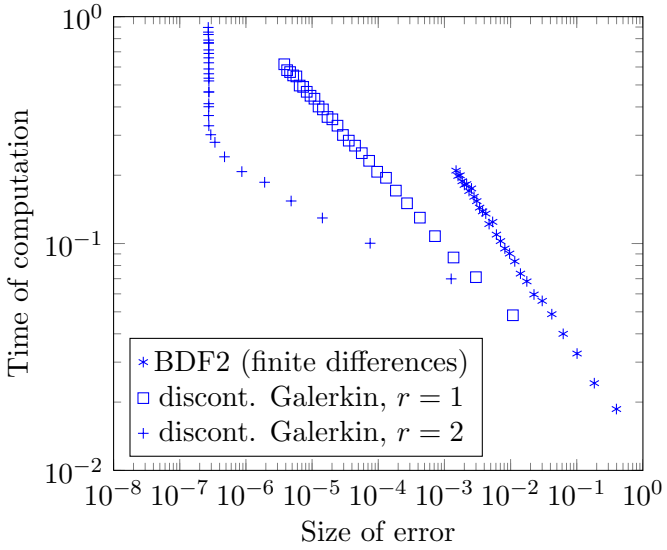

Figure 7: Comparison between the commonly used second order finite difference method BDF2 and the dG method with polynomial order $r = 1$ and $r = 2$ respectively. The number of time steps used are (from the right) 2 up to 30.

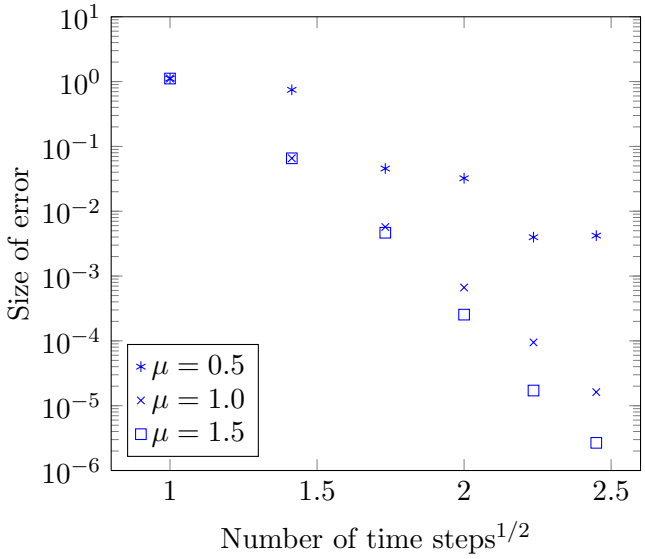

Figure 8: Convergence rates for the *hp*-version of the dG method with $\sigma = 0.24$ and different values of the slope μ.

8 Conclusion

In this work, we have employed a dG method for time integration of the parabolic problem obtained when pricing options using the Black-Scholes model. The dG method is an implicit single step scheme which allows for arbitrary variations in the size of the time steps and the corresponding approximation order. The method leads to having to solve a large system of equations of size $(r+1)N$ in each time step. By choosing the temporal shape functions to be normalized Legendre polynomials it is possible to decouple this system into $r+1$ independent systems of size N, at the price of switching over to complex arithmetic. This decoupling process is of great value both in terms of time of computation and memory requirements. Additionally, since the $r+1$ systems are completely independent, these can be assigned to separate processors and solved in parallel to further decrease the time of computation.

We have investigated different versions of the dG method that can be used to obtain convergence, and concluded that optimal convergence rates of $2r+1$ can be obtained with the h-version, even though there is a discontinuity in the initial condition from the payoff function. This is briefly discussed in [10] and is most likely due to the fact that the solution operator of the parabolic problem is an analytic semigroup which increases the smoothness of the solution for $t > 0$. The hp-version of the dG method is shown to result in exponential convergence in r, but since optimal convergence rates can be obtained with the simpler h-version, it is not recommended for the option pricing problem. The use of non-uniform time steps as well as variable polynomial order in the hp-version increases the complexity of the implementation which results in a significant increase in time of computation compared to the h-version. To increase the accuracy of the dG method it is much more beneficial to uniformly increase the polynomial order or increase the number of time steps used.

The dG method for time integration in combination with the adaptive finite difference operator in space is thus an efficient way to price options. From the numerical results we observe that only a very few number time steps are needed in order to obtain a high temporal accuracy.

References

[1] F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 81(635-654):1, 1973.

[2] M. Delfour, W. Hager, and F. Trochu. Discontinuous galerkin methods for ordinary differential equations. *Mathematics of Computation*, 36(154):pp. 455–473, 1981.

[3] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson. *Computational differential equations*. Studentlitteratur AB, Lund, 1996.

[4] S. Gottlieb, G. W. Wei, and S. Zhao. A unified discontinuous galerkin framework for time integration. 2010.

[5] R. C. Merton. Theory of rational option pricing. *The Bell Journal of Economics and Management Science*, 1973.

[6] V. Nilsson. Diskontinuerliga galerkinmetoder för initialvärdesproblem och prissättning av optioner. Bachelor Thesis TVE 12036, Uppsala University, 2012.

[7] J. Persson and L. von Sydow. Pricing european multi-asset options using a space-time adaptive fd-method. *Computing and visualization in science*, 10(4):173, 2007.

[8] U. Pettersson, E. Larsson, G. Marcusson, and J. Persson. Improved radial basis function methods for multi-dimensional option pricing. *Journal of Computational and Applied Mathematics*, 222(1):82 – 93, 2008.

[9] D. Schötzau and C. Schwab. Time discretization of parabolic problems by the hp-version of the discontinuous galerkin finite element method. *SIAM journal on numerical analysis*, 38(3):837, 2000.

[10] T. von Petersdorff and C. Schwab. Numerical solution of parabolic equations in high dimensions. *Esaim. Mathematical Modelling And Numerical Analysis*, 38(1):93, 2004.

[11] T. Werder, K. Gerdes, D. Schötzau, and C. Schwab. hp-discontinuous galerkin time stepping for parabolic problems. *Computer methods in applied mechanics and engineering*, 190(49):6685, 2001.