Proper Orthogonal Decomposition for Pricing Options

Olivier Pironneau *

Accepted in Journal of Computational Finance (August 2011)

Abstract

In a paper that appeared in volume 2 (2011) of SIAM Financial Mathematics by R. Cont, N. Lantos and the author, it was shown that by writing the solution of the Black-Scholes partial differential equation on a small set of basis functions the computing time can be dramatically reduced. In this study we show that it is in fact a P.O.D. method and in some other variable it is also a spectral method. It allows us to find a good preconditioning matrix to minimize the ill conditioned linear system, and even have explicit solutions.

Keywords Option pricing, Galerkin basis, Reduced basis, POD.

AMS 37M25, 65N99

Introduction

Derivatives like European calls and puts on a single or compound assets are computed a very large number of times every day. When a closed form solution is not available, an alternative to Monte-Carlo simulation is to solve the Black-Scholes partial differential equation (more details can be found in [2, 9, 1], for example).

A complement to the Finite Element or Finite Difference Method for a numerical solution of the problem is to construct an appropriate basis, smaller in size but with larger support. Proper Orthogonal Decomposition is the usual tool for it [5]. At the cost of some preliminary calculations it allows very fast computations, thereby being well fitted to cases where one needs to solve the problem many times with different data, which is exactly the situation in financial engineering.

In [4] it was shown that a set of rescaled calls with constant volatilities forms a reduced basis with similar performance as POD. In this article we shall show that it is in fact a variant of POD.

*UPMC-ParisVI, LJLL, 4 Place Jussieu, Paris, F-75005 (pironneau@ann.jussieu.fr)

If τ is the time to the maturity T, K is the strike, r the interest rate, it is easy to see that by using the forward moneyness, $y = e^{r\tau}\frac{S}{K}$, as variable, the pricing of a put $P(S,t)$ comes to solve in $\mathbf{R}^+ \times (\mathbf{0}, \mathbf{T})$:

$$\partial_\tau v_\sigma - \frac{\sigma^2 y^2}{2}\partial_{yy} v_\sigma = 0, \quad v_\sigma(y,0) = (1-y)^+ \tag{1}$$

where $v_\sigma(y,\tau) = e^{r\tau}P(Kye^{-r\tau}, T-\tau)/K$.

When σ is constant, the Black-Scholes formula provides an analytical solution

$$v_\sigma(y,\tau) = \frac{y}{2}\left(1 + \mathrm{erf}(\frac{\ln y}{\sigma\sqrt{2\tau}} + \sigma\sqrt{\frac{\tau}{8}})\right) - \frac{1}{2}\left(1 + \mathrm{erf}(\frac{\ln y}{\sigma\sqrt{2\tau}} - \sigma\sqrt{\frac{\tau}{8}})\right) + 1 - y$$
$$\text{and } \mathrm{erf}(y) = \frac{2}{\sqrt{\pi}}\int_0^y e^{-x^2}\mathrm{d}x \tag{2}$$

Notice that only $\sigma\sqrt{\tau}$ appears; so *a change in the volatility and a change in* τ *have the same effect*. Hence a basis of snapshots in time for (1) can also be

$$v_\sigma(y,\tau) = v_\Sigma(y,\tau) + \sum_{i=1}^{I} a_i(\tau)\bar{w}_i(y) \text{ with } \bar{w}_i := v_{\sigma_i}(y,T) - v_\Sigma(y,T) \tag{3}$$

where $\{\sigma_i\}_{i\in I}$ is an appropriate set of constant volatilities and Σ is a chosen reference volatilities. The PDE verified by $u := v_\sigma - v_\Sigma$ is

$$\partial_\tau u - \frac{\sigma^2 y^2}{2}\partial_{yy} u = -(\partial_\tau v_\Sigma - \frac{\sigma^2 y^2}{2}\partial_{yy} v_\Sigma), \quad u(0) = 0. \tag{4}$$

Applying Galerkin's method to (4) with (3) will give the $a_i(\tau)$. For POD it is also common practice to use derivative of snapshots rather than the snapshots themselves. Here too it is numerically much easier to work with $\hat{w}_i := \partial_{yy} v_{\sigma_i}(y,T)$ instead of $\bar{w}_i = v_{\sigma_i}(y,T) - v_\Sigma(y,T)$ and since ∂_{yy} is a bijection with appropriate boundary conditions it shouldn't make much of a difference!

As we shall see a good choice is $\sigma_i = (2c\ T\ i)^{-\frac{1}{2}},\ i = 1, 2...$ where c is a chosen constant. A rapid computation shows that $\hat{w}_i$ is proportional to

$$w_i := \sqrt{y}e^{-\alpha_i \ln^2 y} \text{ with } \alpha_i = c\ i.$$

This will be the basis chosen here, and $c = 1$ in the numerical tests.

The POD algorithm In this framework POD means

1. Choose I (in the range 5 to 10)
2. Apply the Gram-Schmidt algorithm to orthogonalize the $\{w_i\}_{i\in I}$ with respect to the $L^2(\mathbf{R}^+)$ or $H^1(\mathbf{R}^+)$ scalar product $(\cdot,\cdot)$. Let $\{p_i\}_{i\in I}$ be the result.
3. Solve (4) on this basis by Galerkin's method

4. If the result is not precise enough, add a new element in I and go back to Step 2.

The "not precise enough" of Step 4 needs in principle an a posteriori estimate to become mathematically sound. The selection of a "good new basis vector" is also difficult in Step 4 and one way is to order the vectors p_i so as to have a decreasing sequence of eigenvalues of the correlation matrix (w_i, w_j) [6].

Remark 1 *For a fixed I, instead of Gram-Schmidt orthogonalization, it is equivalent to solve the linear system of the Galerkin discretization, by a singular value decomposition algorithm.*

In [4] it was found that the above algorithm is subject to instabilities due to round-off errors when $I > 15$. In practice it isn't a problem because beyond $I \sim 5$ the error is very small and doesn't decay further. Nevertheless there is a need to understand why the linear systems are so ill conditioned for larger I.

Note that all spectral methods have the same problem and require orthonormal basis like the Legendre polynomials [3]. So here too we shall demonstrate that a similar cure is available. To do so we will work with the variable $z = e^{-\ln^2 y}$, because, $\{y^{-\frac{1}{2}} w^i\}_{i\in I} = \{z^i\}_{i\in I}$ and the Legendre polynomials $P^i(z)$ are the orthogonal basis of $\{z^i\}_{i\in I}$; Equivalently, going back to the variable y, $P^i(e^{\sqrt{-\ln z}})$ is an orthogonal basis equivalent to the one that the POD algorithm would construct, effectively implementing the above POD algorithm.

Then one may wonder why not work directly with the variable z? Two reasons:

1. The mapping $y \to z$ is not one-to-one on $\mathbf{R}^+$.

2. Equation (4) written in z is not very simple; it is, for some appropriate f

$$\partial_\tau u - 2\sigma^2 z^2 \ln\frac{1}{z}\partial_{zz} u + \sigma^2 z(1 + \sqrt{\ln\frac{1}{z}} - 2\ln\frac{1}{z})\partial_z u = f, \quad u(z,0) = 0. \quad (5)$$

   and its coefficients have singularities at $z = 0$ and $z = 1$ and it is not clear that the problem is well posed.

We will show that it is possible to work in z, provided some symmetry about $y = 1$ exists in the solution so as to work in $z \in (0, 1)$, and we will propose a finite element solution of the above, for possible use in the future. But here the numerical results on (5) are not good.

1 Galerkin Approximation on the Basis

1.1 Mathematical Result

Decomposition (3) gives a v_σ which satisfies the initial and boundary conditions of the Black-Scholes equations (1); indeed it is zero at $y = 0$, it has also the

right exponential behavior when $y \to \infty$ and it is zero at $t = 0$. Consider $y \to u(y)/\sqrt{y}$ and as before $u = v_\sigma - v_\Sigma$.

Note that $e^{-ic\ln^2 y} = f(y)^i$ with $f(y) = e^{-c\ln^2 y}$. By the Stone-Weierstrass theorem $\{f(y)^n\}_{n=0,1..}$ is a basis if $y \to f(y)$ is a separating function (i.e. $y \to f(y)$ is one to one, continuous); however $e^{-c\ln^2 y}$ is a separating function only on the interval $(0,1)$ or on $(1,\infty)$ but not on $(0,\infty)$. So we need a symmetry assumption about $y = 1$: $u(\frac{1}{y}) = \frac{1}{y}u(y)$ for (3) to work. Fortunately if $\sigma(\frac{1}{y},\tau) = \sigma(y,\tau)$ then it is easy to show that u has the required symmetry. If symmetry does not hold then another set of vectors must be added in the basis, such as $\{\sqrt{y}w^i(y)\}_{i=1,2..}$. More details can be found in [4].

1.2 Implementation

Note that

$$\partial_{yy}w^i = \frac{e^{-\alpha_i \ln^2 y}}{y\sqrt{y}}(4\alpha_i^2 \ln^2 y - \frac{1}{4} - 2\alpha_i)$$
$$\partial_\tau v_\Sigma - \frac{\sigma^2 y^2}{2}\partial_{yy}v_\Sigma = \frac{(\Sigma^2-\sigma^2)y^2}{2}\partial_{yy}v_\Sigma = \frac{e^{-\frac{\Sigma^2\tau}{8}}}{2\Sigma\sqrt{2\pi\tau}}(\Sigma^2-\sigma^2)\sqrt{y}e^{-\frac{\ln^2 y}{2\Sigma^2\tau}} \quad (6)$$

So (3) is

$$\sum_1^I \dot{a}_i\sqrt{y}e^{-\alpha_i\ln^2 y} - \sum_1^I a_i e^{-\alpha_i\ln^2 y}\sigma^2\sqrt{y}(2\alpha_i^2\ln^2 y - \frac{1}{8} - \alpha_i)$$
$$= -\frac{e^{-\frac{\Sigma^2\tau}{8}}}{2\Sigma\sqrt{2\pi\tau}}(\Sigma^2-\sigma^2)\sqrt{y}e^{-\frac{\ln^2 y}{2\Sigma^2\tau}} \quad (7)$$

The Galerkin method requires to multiply this equation by $\gamma(y)\sqrt{y}e^{-\alpha_j\ln^2 y}$ (the extra factor $\gamma(y)$ is a weight) and integrate over $\mathbf{R}^+$. We obtain an ODE system for $a = (a_1(\tau),..,a_I(\tau))^T$:

$$M\dot{a} + Ba = F \text{ with}$$
$$M_{ij} := \int_0^\infty e^{-(\alpha_i+\alpha_j)\ln^2 y}\gamma(y)y\mathrm{d}y$$
$$B_{ij} := -\int_0^\infty e^{-(\alpha_i+\alpha_j)\ln^2 y}\sigma^2(2\alpha_i^2\ln^2 y - \frac{1}{8} - \alpha_i)\gamma(y)y\mathrm{d}y$$
$$F_j := -\frac{e^{-\frac{\Sigma^2\tau}{8}}}{2\Sigma\sqrt{2\pi\tau}}\int_0^\infty e^{-(\alpha_j+\frac{1}{2\Sigma^2\tau})\ln^2 y}(\Sigma^2-\sigma^2)\gamma(y)y\mathrm{d}y \quad (8)$$

The ODE will then be discretized by an implicit Euler scheme.

1.2.1 The Case σ Constant

When σ is constant and $\gamma(y) = y^{-2}$:

$$M_{ij} = \sqrt{\frac{\pi}{\alpha_i+\alpha_j}}$$

$$B_{ij} = \frac{\sqrt{\pi}\sigma^2}{(\alpha_i+\alpha_j)^{\frac{1}{2}}}(\frac{\alpha_i\alpha_j}{\alpha_i+\alpha_j}+\frac{1}{8})$$
$$F_j = -\frac{e^{-\frac{\Sigma^2\tau}{8}}(\Sigma^2-\sigma^2)}{2\sqrt{2\alpha_j\Sigma^2\tau+1}} \tag{9}$$

1.2.2 The Non Constant Case

If σ is a function of y and τ, with $\sigma(\frac{1}{y},\tau)=\sigma(y,\tau)$ it is best to express it on an exponential basis as:

$$\sigma(x,t) = \sigma_0 + \sum_1^J \sigma_j(t)e^{-\alpha_j \ln^2 y} \text{ or } \sigma(x,t) = \sigma_0 + \sum_1^J \sigma_j(t)y^j \tag{10}$$

because then all integrals can be computed analytically.

1.2.3 Another choice for γ

Later on we will change variables to $z = e^{-\ln^2 y}$; then in view of the fact that $\mathrm{d}z = \frac{2}{y}e^{-\ln^2 y}\ln\frac{1}{y}\mathrm{d}y$ we will need to choose

$$\gamma(y) = \frac{2}{y^2}|\ln y| \text{ so as to have } e^{-\ln^2 y}y\gamma(y)\mathrm{d}y = e^{-\ln^2 y}\frac{2}{y}|\ln y|\mathrm{d}y = \mathrm{d}z.$$

Then with σ constant:

$$\begin{aligned} M_{ij} &= \frac{1}{\alpha_i+\alpha_j} \\ B_{ij} &= \frac{\sigma^2}{\alpha_i+\alpha_j}[\frac{1}{8}+\alpha_i-\frac{2\alpha_i^2}{\alpha_i+\alpha_j}] \\ F_j &= -(\Sigma^2-\sigma^2)\frac{e^{-\frac{\Sigma^2\tau}{8}}\sqrt{2\Sigma^2\tau}}{2\sqrt{\pi}(1+2\Sigma^2\tau\alpha_j)} \end{aligned} \tag{11}$$

The computation of the integrals relies on the following:

$$\begin{aligned} \int_0^\infty e^{-k\ln^2 y}\gamma(y)y\mathrm{d}y &= 2\int_0^1 z^k\mathrm{d}z = \frac{2}{k+1} \\ \int_0^\infty e^{-k\ln^2 y}\ln^2\frac{1}{y}\gamma(y)y\mathrm{d}y &= 2\int_0^1 z^k\ln\frac{1}{z}\mathrm{d}z = \frac{2}{(k+1)^2} \end{aligned} \tag{12}$$

1.3 Numerical Tests

The method was thoroughly tested in [4] but since we are concerned with a minor variation of the original we retested the method for the computation of a vanilla call of volatility $\sigma=\sqrt{0.9}$ using $\Sigma=\sqrt{0.5}$ and I ranging from 5 to 30 (using multi precision arithmetics when $I>20$). The maturity is $T=2$ and 10 time steps are used to integrate (8); with 20 time steps the precision is not significantly improved.

Figure 1 shows that the precision of the method is well within the 3 digits required by banks. Calls are computed from puts by the put-call parity relation. The linear system is solved with the SVD module of [7]; however as we shall see, ill conditioning is such that even SVD fails when I is larger that 20 or so.

2 Finite Element Solution of the Problem in variable z

Change of Variable

When u satisfies $u(\frac{1}{y},\tau)=\frac{1}{y}u(y,\tau)$ equation (4) can be studied on the interval $(0,1)$ instead of $\mathbf{R}^+$. The boundary condition at $y=1$ is obtained from the symmetry condition. Indeed, $u(\frac{1}{y})=\frac{1}{y}u(y)$ implies that $2\partial_y u=u$ at $y=1$.

Let $z=e^{-\ln^2 y}$, i.e. $y=e^{-\sqrt{-\ln z}}$.

2.1 The Partial Differential Equation in z

Written in z for $u=v_\sigma-v_\Sigma$ the Black-Scholes equation is :$\forall z\in(0,1)$,

$$\partial_\tau u-2\sigma^2z^2\ln\frac{1}{z}\partial_{zz}u+\sigma^2z(1+\sqrt{\ln\frac{1}{z}}-2\ln\frac{1}{z})\partial_z u=f,\quad u(z,0)=0. \tag{13}$$

with $f=\frac{\sigma^2-\Sigma^2}{\Sigma^2}\partial_\tau v_\Sigma$. At $z=0$ no boundary condition is needed; at $z=1$, since

$$\partial_y u=2z\sqrt{\ln\frac{1}{z}}e^{\sqrt{-\ln z}}\partial_z u$$

we must have, for some c, $\partial_z u\sim c/\sqrt{\ln\frac{1}{z}}$, giving $\partial_z u\sim u/(4\sqrt{\ln\frac{1}{z}})$ at $z=1$. Calling the conormal derivative $\partial_\nu u:=2\sigma^2z^2\ln\frac{1}{z}\partial_z u$ we obtain:

$$\partial_\nu u\sim u\frac{\sigma^2}{2}\sqrt{\ln\frac{1}{z}}\text{ at }z=1\quad\Rightarrow\quad\partial_\nu u(1,\tau)=0.$$

Therefore u is given by

$$\begin{aligned}&\partial_\tau u-2\partial_z(\sigma^2z^2\ln\frac{1}{z}\partial_z u)+\beta z\partial_z u=f,\quad u(z,0)=0,\quad\partial_\nu u(1)=0\\&\text{with }\beta=2z\ln\frac{1}{z}\partial_z\sigma^2+\sigma^2(\sqrt{\ln\frac{1}{z}}-1+2\ln\frac{1}{z})\end{aligned} \tag{14}$$

This problem is well posed in variational form : find $u\in V$ such that $\forall\hat w\in V$:

$$\int_0^1[\hat w\partial_\tau u+2\sigma^2z^2\ln\frac{1}{z}\partial_z u\partial_z\hat w+\beta z\hat w\partial_z u]\mathrm{d}z=\int_0^1 f\hat w\mathrm{d}z \tag{15}$$

where

$$V=\{v\in L^2(0,1)\ :\ z\sqrt{-\ln z}\partial_z v\in L^2(0,1)\}.$$

However the Hilbert structure of V remains to be proved.

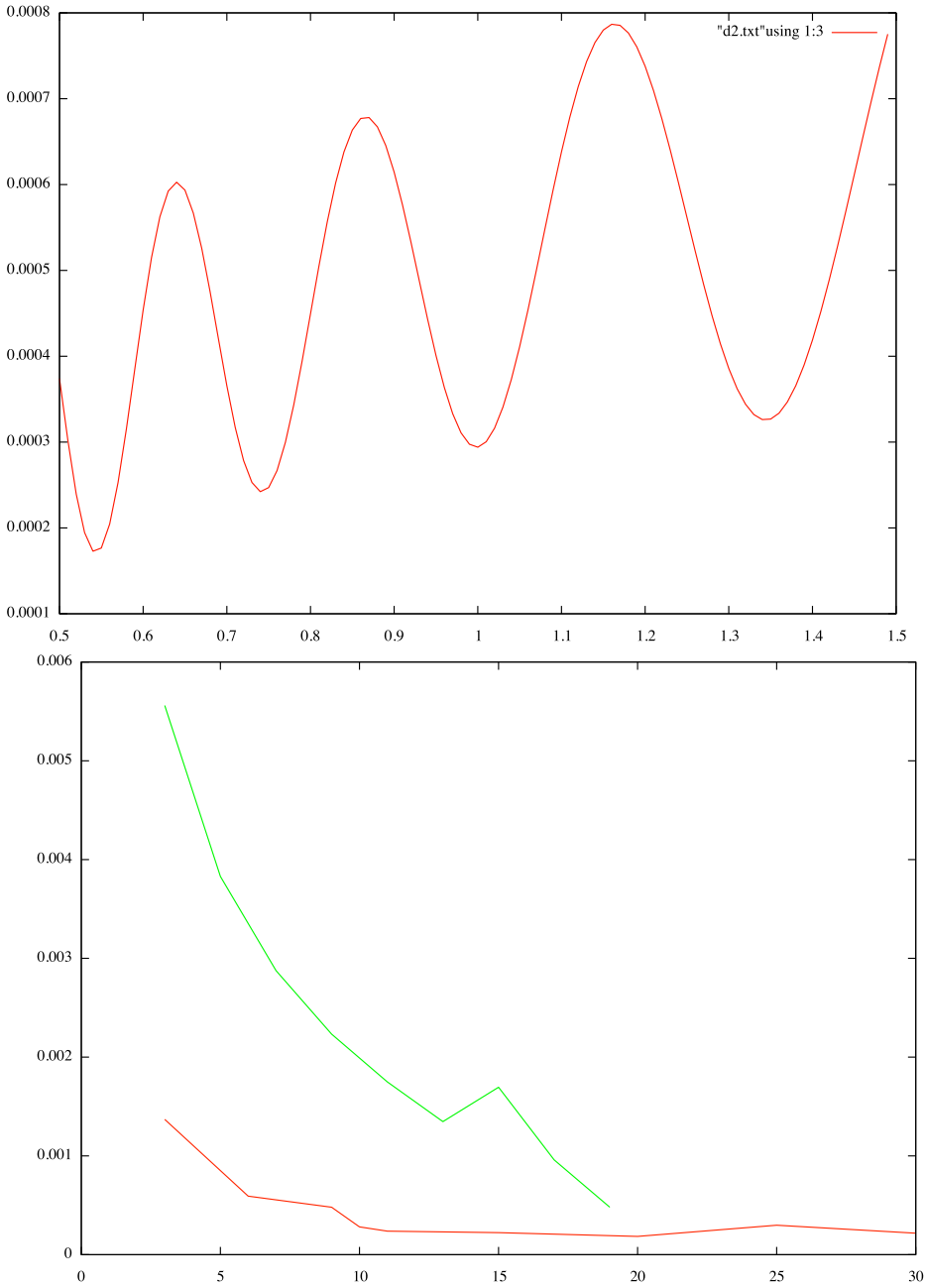

Figure 1: Top figure: errors with $I = 10$ (difference between the analytical Black-Scholes formula and this method) in the interval $y \in (0.5, 1.5)$ is shown to be below the 0.1% accepted standard. Bottom figure: The highest curve is the L^1 error versus ten times the time to maturity τ. The lowest curve is the L^1 error at $\tau = T$ versus the number of basis functions I.

2.2 Numerical Solution

We have discretized the problem with the linear finite element method on a mesh, refined at $z = 0$ and $z = 1$ and with an implicit Euler scheme in time. Using `freefem++` the following code implements the method. The source code is given to show the reader how simple it is to solve even 1D problems (`freefem++` was designed primarily for 2 and 3D problems).

```
real dt=0.02, T=2, sig2=0.09, Sig2=0.25;
mesh Th=square(100,1,[x,y/10]);
fespace Vh(Th,P1,periodic=[[1,x],[3,x]]);

Vh uold=0,u,v;
func alpha=-2*sig2*x^2*log(x);
func beta=sig2*x*(sqrt(-log(x))-1-2*log(x));
func rhs=(sig2-Sig2)*sqrt(x)*exp(-Sig2*t/8-log(x)^2/(2*Sig2*t))
                                              /(2*sqrt(2*Sig2*t*pi));
for(t=dt;t<T;t+=dt){
 solve aa(u,v)=
         int1d(Th,1)(u*v+dt*alpha*dx(u)*dx(v)+dt*beta*v*dx(u))
                   -int1d(Th,1)(uold*v + dt*v*rhs) +on(4,u=0);
  uold=u;
}
```

Figure 2 shows the precision obtained by this method. The precision is poor and the computing time is not small compared to a similar numerical solution of the Black-Scholes PDE written in y.

3 A Basis based on Legendre Polynomials

As a function of z, $\frac{u}{\sqrt{y}}$ can be expanded by Taylor's formula:

$$\frac{u}{\sqrt{y}} = \sum_{i=0}^{I} a_i z^i + \frac{C}{(I+1)!} z^{I+1} \tag{16}$$

Equivalently we can use Legendre polynomials p^i instead of z^i

$$\frac{u}{\sqrt{y}} = \sum_i \underline{a}_i(\tau) p^i(y) \tag{17}$$

However due to the exponential behavior of $u \to 0$ near $y = 0$ we must keep only those p^i which satisfy $p^i(0) = 0$, so i must be an odd integer greater than 0 and less than I.

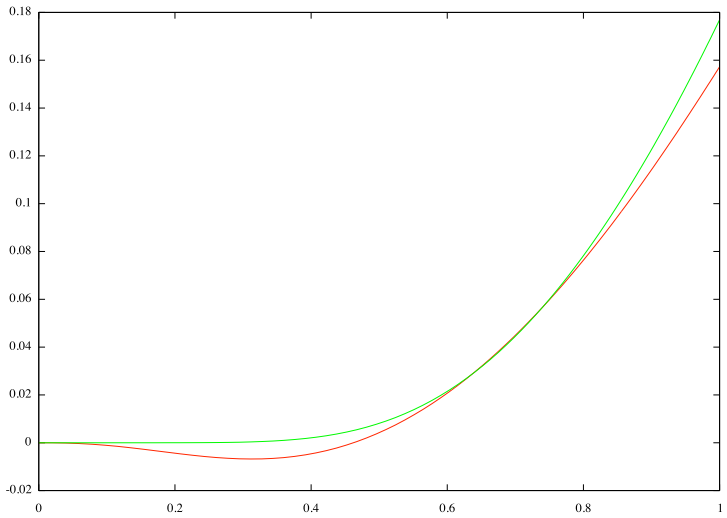

Figure 2: The solution obtained with the finite element program in z with 100 elements and 100 time steps (the curve that goes slightly below zero). For comparison v_σ computed with Black-Scholes formula is also plotted. The precision is poor. Here $\sigma = 0.3$, $\Sigma = 0.5$, $T = 2$.

3.1 The New Basis

With (17), to obtain a mass matrix $\int_0^1 p_i p_j \mathrm{d}z$ from the first term of the PDE it suffices to multiply (4) by $p^j \frac{\mathrm{d}z}{\sqrt{y}}$ and integrate. Hence

$$\sum_{i\in\mathcal{I}} \dot{\underline{a}}_i \int_0^1 p^i p^j \mathrm{d}z = \sum_{i\in\mathcal{I}} \underline{a}_i \int_0^1 \frac{\sigma^2 y^2}{2} (\partial_{yy}(p^i\sqrt{y})) p^j \frac{\mathrm{d}z}{\sqrt{y}} + \int_0^1 f p^j \frac{\mathrm{d}z}{\sqrt{y}}$$

Knowing that

$$p^j = \sum_k q_{jk} \frac{w^k}{\sqrt{y}} \tag{18}$$

and that $\mathrm{d}z = \frac{2}{y} e^{-\ln^2 y} \ln\frac{1}{y}\mathrm{d}y$ we obtain,

$$\begin{aligned}
\frac{\dot{\underline{a}}_j}{2j+1} &= \sum_{i,k,m} \underline{a}_i q_{im} q_{jk} \int_0^1 \frac{\sigma^2 y^2}{2} \partial_{yy} w^m \frac{w^k}{\sqrt{y}} \frac{\mathrm{d}z}{\sqrt{y}} + \sum_k q_{jk} \int_0^1 f \frac{w^k}{\sqrt{y}} \frac{\mathrm{d}z}{\sqrt{y}} \\
&= \sum_{i,k,m} \underline{a}_i q_{im} q_{jk} \int_0^1 \sigma^2 \partial_{yy} w^m w^k e^{-\ln^2 y} \ln\frac{1}{y}\mathrm{d}y \\
&+ \sum_k q_{jk} \int_0^1 \frac{2 f w^k}{y^2} e^{-\ln^2 y} \ln\frac{1}{y}\mathrm{d}y, \quad 1 \le j \le I
\end{aligned} \tag{19}$$

because $\int_0^1 p^i p^j = 1/(2j+1)$. This is a system of the type

$$\underline{M}\ \dot{\underline{a}} + \underline{B}\ \underline{a} = \underline{F}$$

Remark 2 *Because $\underline{M}$ is diagonal we can use an explicit Euler scheme for time discretization and assert explicitly the stability condition for $\delta\tau$. Numerical tests show that it is not a restrictive condition; hence the method becomes very fast.*

3.2 Computation of Q, B and F

One of the definitions of p^n is

$$(n+1)p^{n+1}(z) = (2n+1)zp^n(z) - np^{n-1}(z), \text{ starting at } p^0 = 1,\ p^1 = z.$$

Hence, starting with $q_{0,0} = 1, q_{1,0} = 0, q_{1,1} = 1$, and with the convention that $q_{n,m} = 0$ if $m > n$ or $m < 0$ we obtain:

$$(n+1)q_{n+1,j} = (2n+1)q_{n,j-1} - nq_{n-1,j}, \quad j = 0, ..n+1; \quad n = 1, 2. \tag{20}$$

By choosing the weight γ as in section 1.2.3 we obtain the right formulae for $\underline{M}$, so the relations between $\underline{B}, \underline{F}, \underline{a}(\tau)$ and $B, F, a(\tau)$ are

$$\underline{M} = QMQ^T, \quad \underline{B} = QBQ^T, \quad \underline{F} = QF, \quad a = Q^T\underline{a} \tag{21}$$

3.3 Numerical Results

Freefem++ is a public domain package to solve partial differential equations (see `www.freefem.org`); it uses a language close to the mathematical formulation and it allows to manipulate matrices. The following `freefem++` script should be self explanatory:

```
real t, dt=0.002, T=2, sig2=0.1, Sig2=0.25;
int n2=26, id=2, n=n2/id;
real [int,int] Q2(n2,n2),  B(n,n),  C(n,n), Q(n,n), QT(n,n);
real [int] a(n),b(n),f(n);
complex[int] vp(n);
complex[int,int] VP(n,n);
                  // compute Q
Q2(0,0)=1;Q2(1,0)=0; Q2(1,1)=1;
for(int m=1;m<n2-1;m++)
 for(int j=0;j<m+2;j++)
  Q2(m+1,j) = ((2*m+1)*((j>0)?Q2(m,j-1):0.)
           -m*((m>0 && j<m)?Q2(m-1,j):0.))/(m+1);
              // Compute B
for(int m=0;m<n2;m+=id)
 for(int k=0;k<n2;k+=id){
```

```
  B(k/id,m/id) = -sig2*2*((m+1)/(m+k+3.0))^2-(m+1.125)/(m+k+3.0);
  Q(m/id,k/id) =Q2(m,k);
  QT(m/id,k/id)=Q2(k,m);
 }
dgeev(B,vp,VP); // eigen solver before multiplication by Q
C=Q*B; B=C*QT;
dgeev(B,vp,VP); // eigen solver after multiplication by Q
```

This program computes the eigenvalues of B given by (11) and those of the matrix which appears in (19), namely QB^TQ^T.

Table 1: Eigen values of B and QB^TQ^T for $n = 12$

i	$R_e\lambda_i$	$I_m\lambda_i$	i	$R_e\lambda_i$	$I_m\lambda_i$
0	0.0203252	0.127462	0	0.627759	0.124794
1	0.0203252	-0.127462	1	0.627759	-0.124794
2	0.00381098	0.00349785	2	0.33515	0.0775545
3	0.00381098	-0.00349785	3	0.33515	-0.0775545
4	0.00013659	0	4	0.159081	0.0463727
5	2.18895e-05	0	5	0.159081	-0.0463727
6	4.95924e-07	0	6	0.0603271	0.0275838
7	3.06291e-08	0	7	0.0603271	-0.0275838
8	4.38722e-10	0	8	0.0195863	0.00830954
9	1.13289e-11	0	9	0.0195863	-0.00830954
10	7.37279e-14	0	10	0.00291485	0
11	4.86674e-16	0	11	0.000395566	0

Table 1 shows that the condition number of the matrix of the linear system has been reduced by a factor of 10^{12} by switching from polynomials in z^n to Legendre polynomials p^n. The new linear system is still not ideally conditioned and for $I > 30$ the eigenvalue solver of Lapack fails, but such I is unnecessarily too large anyway. However this is a known problem for spectral solvers as well, for which solutions have been found [3],[8].

Table 2: First 6 coefficients on the basis versus n at $T = 2$

n	a_1	a_2	a_3	a_4	a_5	a_6
4	-2.699e-2	-1.376e-2	-2.434e-3	-3.732e-4		
6	-2.707e-2	-1.384e-2	-2.495e-3	-3.782e-4	-1.770e-4	-7.748e-05
8	-2.710e-2	-1.386e-2	-2.515e-3	-3.959e-4	-1.917e-4	-7.287e-05
12	-2.711e-2	-1.387e-2	-2.529e-3	-4.078e-4	-2.012e-4	-8.110e-05

Table 3: Condition numbers of B and QBQ^T versus n

n	$\frac{\max_i \|\lambda_i(B)\|}{\min_i \|\lambda_i(B)\|}$	$\frac{\max_i \|\lambda_i(QBQ^T)\|}{\min_i \|\lambda_i(QBQ^T)\|}$
4	6.69e02	6.98
6	4.50e05	7.45
8	3.51e08	3.12e02
12	2.70e14	1.60e03

Conclusion

By using the variable z we have shown here that the reduced order model proposed in [4] is a spectral method and that the POD algorithm amounts to using the Legendre polynomials instead of the canonical basis attached to $\{z^i\}_i$. However using variable z has its own difficulties as demonstrated by the direct FEM solution of the Black-Scholes equation in z. So we have translated this information back into the moneyness variable y by a matrix multiplication similar to a preconditioning, which makes the mass matrix diagonal; coupled with an explicit Euler time scheme the method requires no linear solver and consequently it is extremely fast. This analysis explains also why so few basis functions are needed and why the matrices in the y-formulation are so ill conditioned (Hilbert matrices $\int_0^1 z^i z^j$).

The analysis requires a symmetry condition about $y = 1$. In absence of such property one needs to work with all the w_i, i even and odd. This will be the topic of a forthcoming paper.

References

[1] Yves Achdou and O. Pironneau *Numerical Methods for Option Pricing* SIAM series, Philadelphia, USA, 2005.

[2] F. Black, M. Scholes: *The pricing of options and corporate liabilities*, *J. Political Econ.* 81, pp. 637-659 (1973) .

[3] C. Canuto, M.Y. Hussaini, A. Quarteroni and T.A. Zang Spectral Methods, Fundamentals in single domains. Springer 2006.

[4] R. Cont and N. Lantos and O. Pironneau: A reduced basis for option pricing. SIAM on mathematical Finance, vol. 2, pp 287-316, 2011.

[5] problems. Numerische Mathematik, 90:117?148, 2001.

[6] C. Prud'homme, D. Rovas, K. Veroy, Y. Maday, A.T. Patera, G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. Journal of Fluid Engineering, 124(1):70-80, 2002.

[7] W.H. Press, S.A. Teukolsky, W.T. Vettering and B.P. Flannery Numerical Recipes in C, 2nd edition, Cambridge University Press, 1992.

[8]

[9] Paul Wilmott, Sam Howison, and Jeff Dewynne: *The mathematics of financial derivatives*. Cambridge University Press, Cambridge, 1995. A student introduction.