Wavelet Based PDE Valuation of Derivatives

M.A.H. Dempster

Centre for Financial Research

Judge Institute of Management

University of Cambridge

&

Cambridge Systems Associates Limited

Co-workers: A. Eswaran, D.G. Richards & G.W.P Thompson

Cambridge University Finance Seminar, 16 March, 2001

Research partially sponsored by Citicorp (Schroder Salomon Smith Barney)

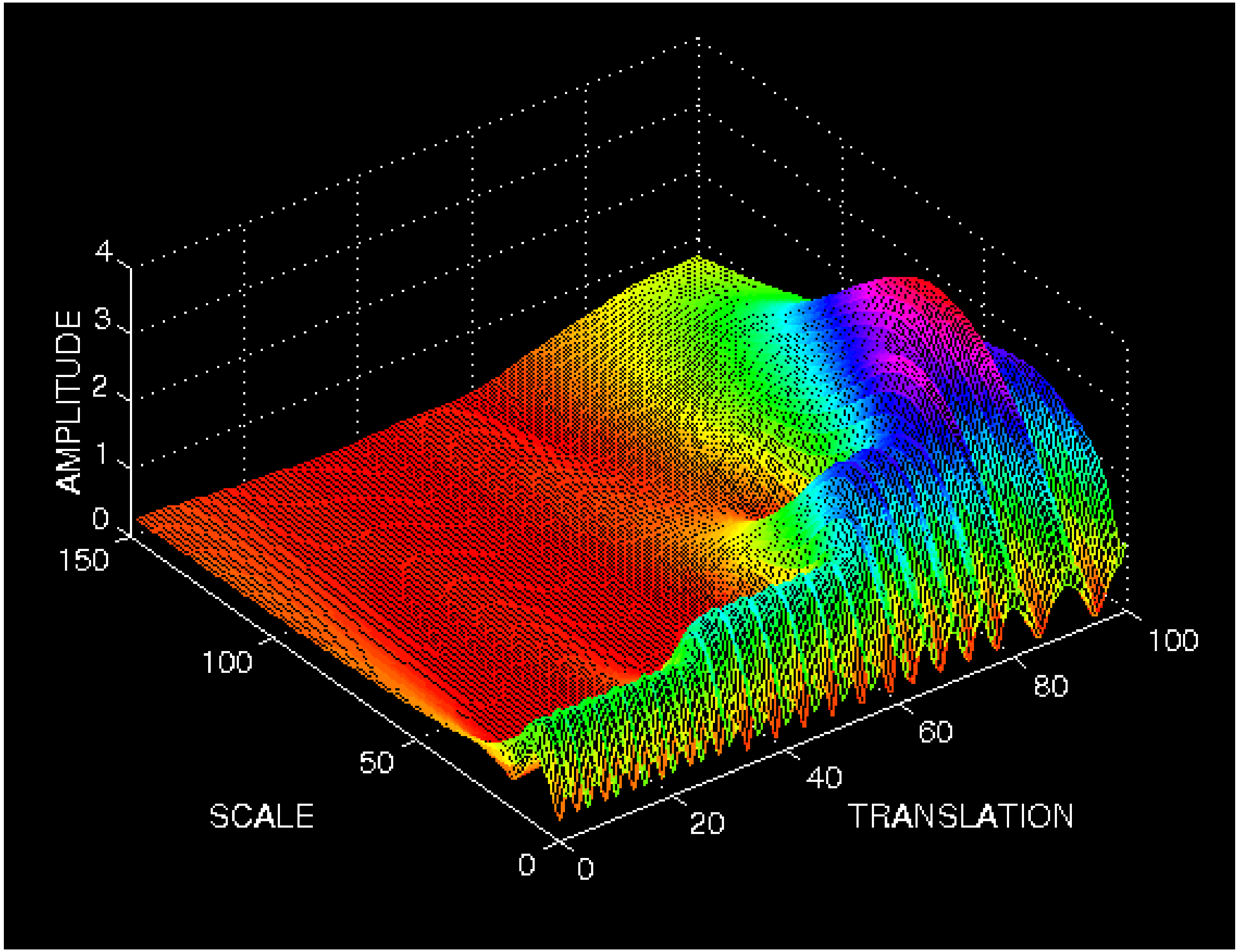
Outline

1. Introduction
2. Wavelet Transforms
3. Wavelets and PDE's
4. European Option Valuation
5. Cross-currency Swap Valuation
6. 3-Factor Interest Rate Swap Valuation
7. Simulation Methods for Bermudan Swaptions
8. Conclusions and Further Work

1. Introduction

- What are wavelets ?
- Wavelets are nonlinear functions which can be scaled and translated to form a basis for the Hilbert space $L^2(\mathbb{R})$ of square integrable functions
- Wavelets generalize the trigonometric functions e^{ist} $(s \in \mathbb{R})$ which generate the classical Fourier basis for L^2
- Hence there are wavelet -- and fast wavelet transforms -- which generalize the time to frequency map of the Fourier transform to pick up both the space and time behavior of a function


30 Hz
20 Hz
10 Hz
5 Hz
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
200
400
600
800
1000
Time, ms



Source: Robi Polikar's wavelet tutorial

Applications of wavelets

- Digital image compression
- Signal processing
- De-noising of signals --filtering
- Numerical analysis

Wavelets and PDEs

A wavelet based approach to the solution of PDEs has been studied by

- Beylkin (1993)
- Vasilyev, Yuen, Pao (1997)
- Monasse and Perrier (1998)
- Prosser and Cant (1999, 2000)
- Dahmen *et al* (1999)
- Cohen *et al* (2000)

Advantages of using wavelets to solve derivative valuation PDEs

- Wavelet PDE methods combine the advantages of both spectral (Fourier) and finite-difference methods and allow both space and time dependent coefficients
- Large classes of operators and functions are sparse or sparse to high accuracy when transformed into the wavelet domain
- Wavelets are suitable for problems with the multiple spatial scales common in financial problems
- Wavelets can be used for nonlinear terms
- Wavelets accurately represent the PDE solution in regions of sharp transitions

2. Wavelet Transforms

Scaling Functions and Wavelets

- The scaling function ϕ is the solution of a dilation equation

$$\phi(\mathrm{x}) = \sqrt{2}\sum_{k=0}^{\infty} h_k \phi(2x-k)$$

where ϕ is normalised i.e. $\int_{-\infty}^{\infty}\phi(x)dx = 1$

- The wavelet ψ is defined in terms of the scaling function as

$$\psi(x) = \sqrt{2}\sum_{k=0}^{\infty} g_k \phi(2x-k)$$

- The basis functions $\phi_{j,k}$ and $\psi_{j,k}$ are generated from ϕ and ψ through scaling and translation as

$$\phi_{j,k}(x) := 2^{-j/2}\phi(2^{-j}x - k) = 2^{-j/2}\phi\left(\frac{x - 2^j k}{2^j}\right)$$

$$\psi_{j,k}(x) := 2^{-j/2}\psi(2^{-j}x - k) = 2^{-j/2}\psi\left(\frac{x - 2^j k}{2^j}\right)$$

- $h := \{h_k\}_{k=0}^{\infty}$ and $g := \{g_k\}_{k=0}^{\infty}$ are chosen so that dilations and translations of the wavelet ψ form an orthonormal basis of $L^2(\mathbb{R})$ *i.e.*

$$\int_{-\infty}^{\infty} \psi_{j,k}(x)\ \psi_{j',k'}(x)\ dx = \delta_{j,j'}\ \delta_{k,k'}$$

where δ denotes the Kronecker delta

- The two sequences of coefficients *h* and *g* are known from the signal processing literature as *filters*

- For any function $f \in L^2(\mathbb{R})$ there exists a matrix $\{d_{jk}\}$ such that

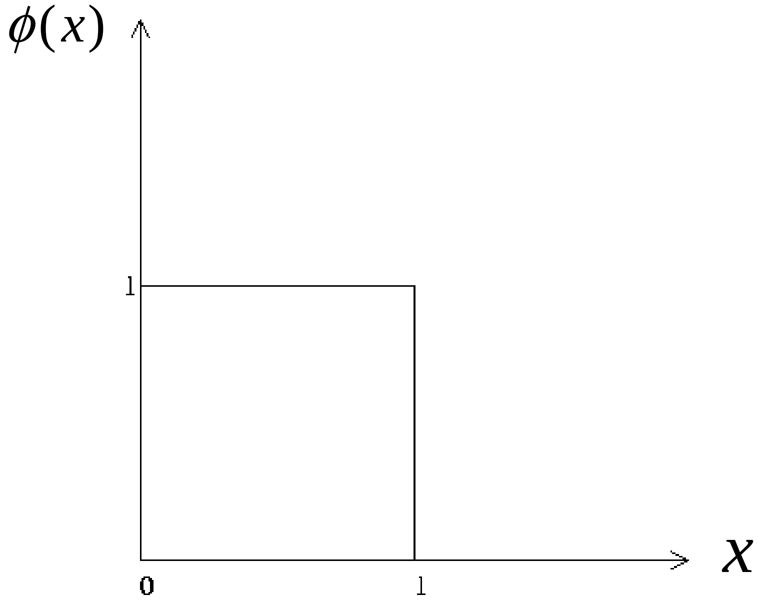
$$f(x) = \sum_{j \in \mathbb{Z}_+} \sum_{k \in \mathbb{Z}_+} d_{jk} \psi_{j,k}(x)$$

where

$$d_{jk} = \int_{-\infty}^{\infty} f(x) \psi_{j,k}(x) dx$$

- For example the Haar wavelet is defined by $h_0 = h_1 = 1/\sqrt{2}$
- For the Haar filter the solution to the dilation equation gives the unit box function for ϕ *i.e.*

$\phi(x) = 1$ for $x \in [0,1]$ and 0 otherwise



- The spaces spanned by $\phi_{j,k}$ and $\psi_{j,k}$ over the location parameter k with the scale parameter j fixed are usually denoted by

$$V_j = span_{k \in \mathbb{Z}_+} \phi_{j,k}$$

$$W_j = span_{k \in \mathbb{Z}_+} \psi_{j,k}$$

- To implement wavelet analysis on a computer we must have a smallest scale and a largest scale and thus we limit the range of the scale and location parameters j and k

- The spaces V_j and W_j are termed approximation subspaces and scaling function subspaces respectively

- These parameter ranges are chosen according to a required grid size in numerical computation
- The orthogonal wavelet approximation to a continuous function f is given by

$$f(x) \approx \sum_k d_{0,k}\psi_{0,k}(x) + \cdots + \sum_k d_{J-1,k}\psi_{J-1,k}(x) + \sum_k s_{0,k}\phi_{0,k}(x)$$

where J is the number of resolution scales and k ranges from 0 to the number of coefficients in the specified component

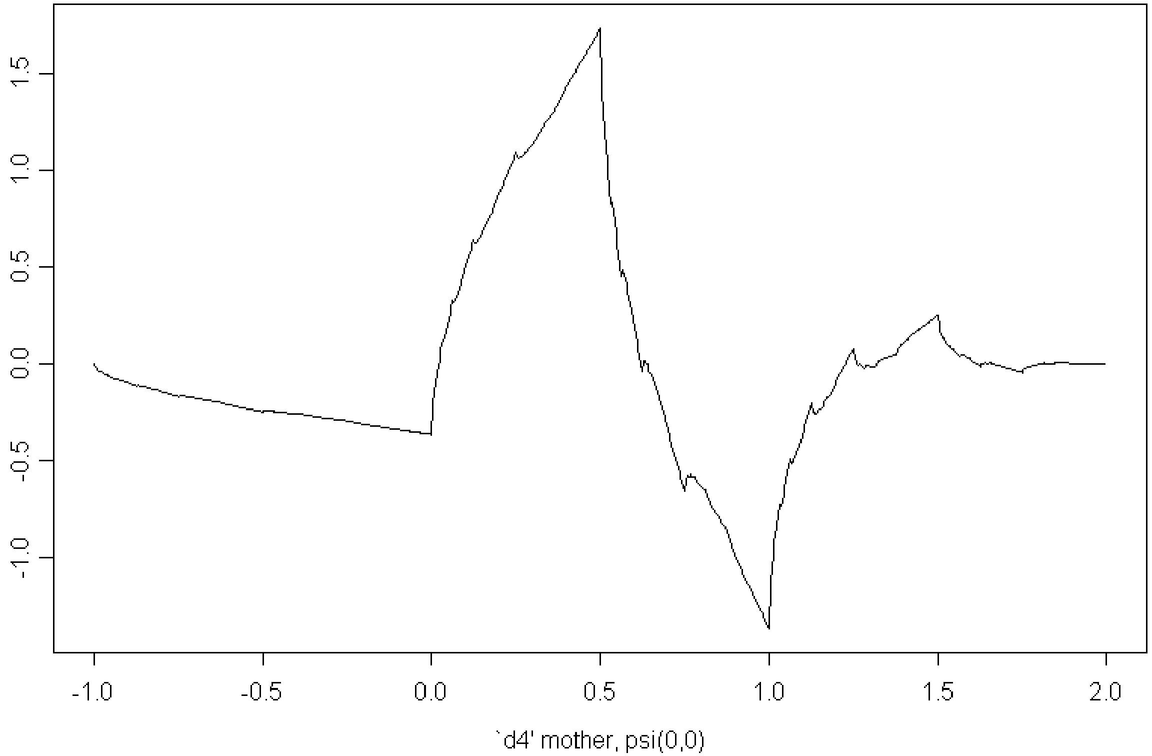
- For Daubechies wavelets the length L of the filters h and g is related to exact $(M-1)^{th}$ degree polynomial approximation for a fixed number J of resolution scales

- This is guaranteed by requiring the wavelet ψ to have $M := L/2$ vanishing moments i.e.

$$\int_{-\infty}^{\infty} \psi(x) x^m dx = 0$$

for $m=0,\dots,M-1$

Daubechies wavelet ψ



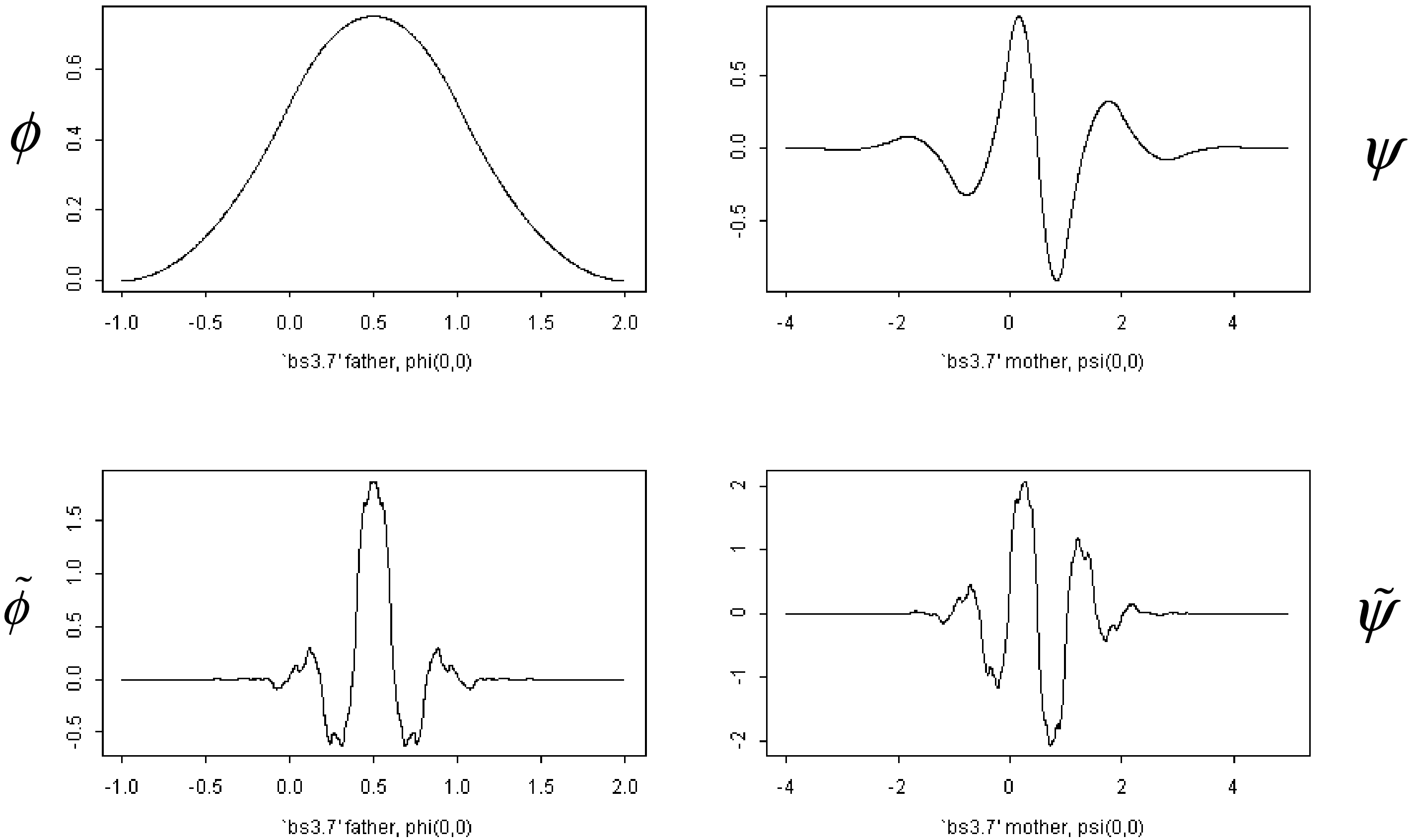
Biorthogonal wavelets

Cohen, Daubechies and Feauveau (1992)

- Four basic function types -- two primals and two duals
- The biorthogonal wavelet approximation is expressed in terms of the dual wavelet functions

$$f(x) \approx \sum_k d_{0,k} \tilde{\psi}_{0,k}(x) + \cdots + \sum_k d_{J-1,k} \tilde{\psi}_{J,k}(x) + \sum_k s_{0,k} \tilde{\phi}_{0,k}(x)$$

Biorthogonal B-spline wavelets



- Biorthogonal wavelets are not orthogonal but they satisfy the biorthogonality relationships

$$\int \phi_{j,k}(x)\ \tilde{\phi}_{j',k'}(x)\ dx = \delta_{j,j'}\delta_{k,k'}$$

$$\int \phi_{j,k}(x)\ \tilde{\psi}_{j',k}(x)\ dx = 0$$

$$\int \psi_{j,k}(x)\ \tilde{\phi}_{j,k'}(x)\ dx = 0$$

$$\int \psi_{j,k}(x)\ \tilde{\psi}_{j',k'}(x)\ dx = \delta_{j,j'}\delta_{k,k'}$$

Biorthogonal approach

- Biorthogonal wavelets are symmetrical and generalize the orthogonal wavelet approximation
- Basis functions for biorthogonal wavelet spaces are generated from the primal scaling function ϕ and the dual scaling function $\tilde{\phi}$
- Biorthogonal systems are thus derived from a paired hierarchy of approximation subspaces

$$\cdots V_{j-1} \subset V_j \subset V_{j+1} \cdots$$

$$\cdots \tilde{V}_{j-1} \subset \tilde{V}_j \subset \tilde{V}_{j+1} \cdots$$

- Wavelet innovation spaces W_j and $\tilde{W}_j$ are defined by

$$\mathrm{V}_{j+1} = \mathrm{V}_j \oplus \mathrm{W}_j$$
$$\tilde{\mathrm{V}}_{j+1} = \tilde{\mathrm{V}}_j \oplus \tilde{\mathrm{W}}_j$$

with $\tilde{V}_j \perp W_j \quad V_j \perp \tilde{W}_j$

- Basis functions for the wavelet innovation spaces are generated by the primal and dual wavelets ψ and $\tilde{\psi}$

- Projections of a function *f* onto finite dimensional scaling function space V_J or wavelet space W_J are given by

$$P_{V_j} f(x) = \sum_k \left\langle f(u), \tilde{\phi}_{j,k}(x) \right\rangle \phi_{j,k}(x)$$
$$P_{W_j} f(x) = \sum_k \left\langle f(u), \tilde{\psi}_{j,k}(x) \right\rangle \psi_{j,k}(x)$$

Biorthogonal interpolating wavelet transforms

Deslauriers and Dubuc (1989) Donoho (1992)

- The biorthogonal interpolating wavelet transform has basis functions of the form

$$\phi_{j,k}(x) = \phi(2^j x - k)$$

$$\psi_{j,k}(x) = \phi(2^{j+1} x - 2k - 1)$$

$$\tilde{\phi}_{j,k}(x) = \delta(x - x_{j,k})$$

where δ is the Dirac delta function and $x_{j,k}$ is a grid point in the spatial dimension at scale level *j*

- An explicit form for $\tilde{\psi}$ is unknown but the corresponding wavelet coefficients can be derived using the biorthogonality conditions
- These wavelets are interpolating because the primal scaling function $\phi(x)$ is equal to 1 for $x = 0$ and 0 for $x \neq 0$ leading to polynomial interpolation of up to order *M-1* between grid points

Fast interpolating wavelet transform algorithm

- The projection of a function *f* onto a finite dimensional scaling function space V_j is given by

$$\begin{aligned} P_{V_j} f(x) &= \sum_k \left\langle f(u), \tilde{\phi}_{j,k}(x) \right\rangle \phi_{j,k}(x) \\ &= \sum_k f(k/2^j)\phi_{j,k}(x) \\ &:= \sum_k s^f_{j,k}\phi_{j,k}(x) \end{aligned}$$

Fast interpolating wavelet transform algorithm structure

$$\left\{s^f_{J,0}, s^f_{J,1}, s^f_{J,2} \cdots\cdots s^f_{J,2^J-1}\right\}^T$$

Finest: 2^6=64

$$\downarrow$$

$$\left\{d^f_{J-1,0}, \cdots, d^f_{J-1,2^{J-1}-1} \mid s^f_{J-1,0}, s^f_{J-1,1}, s^f_{J-1,2} \cdots\cdots s^f_{J-1,2^{J-1}-1}\right\}^T$$

$$\downarrow$$

$$\left\{d^f_{J-1,0}, \cdots, d^f_{J-1,2^{J-1}-1} \mid d^f_{J-2,0}, \cdots, d^f_{J-2,2^{J-2}-1} \mid s^f_{J-2,0}, \cdots\cdots s^f_{J-2,2^{J-2}-1}\right\}^T$$

$$\downarrow$$

$$\left\{d^f_{J-1,0}, \cdots, d^f_{J-1,2^{J-1}-1} \mid \cdots d^f_{J-P-1,0}, \cdots, d^f_{J-P-1,2^{J-P-1}-1} \mid s^f_{J-P-1,0} \cdots\cdots s^f_{J-P-1,2^{J-P-1}-1}\right\}^T$$

Coarsest: 2^3=8

$$\downarrow$$

$$W_{J-1} \oplus W_{J-2} \oplus W_{J-3} \oplus \cdots \oplus V_{J-P}$$

Example: J := 6 P := 3

Interpolating biorthogonal wavelet transform complexity

- The number of operations required for the transform algorithm for P resolution levels is $2M\sum_{i=J-P}^{J}2^i = 2^{J-P}M\left(2^{P+1}-1\right)$ as $2M$ filter coefficients define the primal scaling function which spans the space of polynomial of degree less than $M-1$
- Calculation of the wavelet coefficients $d_{j,k}^{f}$ for a given resolution level j can be accomplished in $2(M-1)+1$ floating point operations and the sub-sampling process for the scaling function coefficients requires a further 2^j operations for a total of $2^{j+1}M$ operations required per resolution level j
- For fixed J and P the fast interpolating wavelet transform algorithm is $O(M)$ for exact $(M-1)^{st}$ order polynomial approximation
- Since the finest resolution in a spatial grid of N points is $J=log_2N$ for fixed M and P the complexity of the transform is $O(N)$

3. Wavelets and PDEs

Decomposition of differential operators

- Define $\partial_J^{(n)}$ such that

$$\partial_J^{(n)} f(x) := P_{V_J} \frac{d^n}{dx^n} P_{V_J} f(x)$$

- Repeated application of the approximation subspace decomposition gives

$$\partial_J^{(n)} f(x) := \left(P_{V_{J-P}} + \sum_{i=J-P}^{J-1} P_{W_i} \right) \frac{d^n}{dx^n} \left(P_{V_{J-P}} + \sum_{i=J-P}^{J-1} P_{W_i} \right) f(x)$$

- For example the decomposition of the first derivative operator $\frac{d}{dx}$ is given by

$$\partial_J = \left(P_{V_{J-P}} + \sum_{i=J-P}^{J-1} P_{W_i} \right) \frac{d}{dx} \left(P_{V_{J-P}} + \sum_{i=J-P}^{J-1} P_{W_i} \right)$$

- Alternatively $\partial_J = W \partial_J^1 W^{-1}$ where $\partial_J^1 = P_{V_J} \frac{d}{dx} P_{V_J}$ and W and W^{-1} are orthogonal matrices denoting the forward and inverse transforms with $W = W' = W^{-1}$

- Using the sampling nature of the dual scaling function ∂_J^1 can be written as to take account of domain boundaries as

$$\partial_J^1 = \begin{cases} 2^J \sum_{\alpha,k} s_{J,k}^f \dfrac{d\phi^L}{dx}(\alpha - k)\ \phi_{J,\alpha}^L & k = 0,\cdots,M-1 \\ 2^J \sum_{\alpha,k} s_{J,k}^f \dfrac{d\phi}{dx}(\alpha - k)\ \phi_{J,\alpha} & k = M,\cdots,2^J - M \\ 2^J \sum_{\alpha,k} s_{J,k}^f \dfrac{d\phi^R}{dx}(\alpha - k)\ \phi_{J,\alpha}^R & k = 2^J - M + 1,\cdots,2^J \end{cases}$$

- The entire operator ∂_J^1 can thus be determined provided the values of $r_{\alpha-k}^{(1)} = d\phi(\alpha-k)/dx$ can be obtained
- An approach to determining filter coefficients $r_k^{(n)}$ for a derivative of order n is given by Prosser and Cant (1999)
- Their development provides analytic expressions for the resulting scaling function ϕ and its derivative filter coefficients

Non constant coefficients

- Wavelet-based PDE methods have mainly dealt with constant coefficients but financial PDEs frequently have non-constant coefficients
- Canonical transformation of the variables is usually undesirable or impractical
- Our solution methodology should be able to handle a wide range of PDEs with non-constant coefficients

Pseudo-spectral technique

- Traditional way of handling this problem in the wavelet PDE literature
  - Do an inverse transform at each time step
  - Evaluate the product in physical space
- Advantages
  - Straightforward to implement
- Disadvantages
  - Computationally expensive

Combined operator approach

- We construct a new differential operator which combines non-constant coefficients and derivative terms
- Start with the usual projection P_J of f onto a wavelet (or scaling) space W_J

$$P_J f(x) = \sum_k s^f_{J,k} \phi_{J,k}(x)$$

- Multiplying by a nonlinear function gives

$$g(x)[P_J f(x)] = \sum_k s^f_{J,k} [g(x) \phi_{J,k}(x)]$$

- Since $g(x)\phi_{J,k}(x)$ is not a basis for V_J project again to obtain

$$P_J(g(x)[P_J f(x)]) = \sum_{\alpha}\sum_{k} s^{f}_{J,k} < g(x)\phi_{J,k}(x), \tilde{\phi}_{J,\alpha} > \phi_{J,\alpha}(x)$$

- To determine the inner product in the case of a differential operator recall the interpolating nature of the dual scaling function to yield

$$< g(x)\phi_{J,k}(x), \tilde{\phi}_{J,\alpha} >= 2^{J} g(2^{-J}\alpha)\phi'(\alpha - \kappa)$$

- We apply the standard decomposition to the above expression to get the combined differential operator

PDEs in 1 space dimension

- Consider a first order nonlinear hyperbolic PDE defined over an interval $\Omega = [x_l, x_r]$

$$\frac{\partial u}{\partial t} = \frac{\partial u}{\partial x} + S^{u/\rho} \qquad x \notin \partial\Omega$$

$$\frac{\partial u}{\partial t} = -x^L(t) \qquad x = x_l$$

$$\frac{\partial u}{\partial t} = -x^R(t) \qquad x = x_r$$

- Its wavelet transformed counterpart is

$$\frac{\partial}{\partial t}\wp_{J-P}^{J-1}(u) = -\partial_J^{(1)} u + \wp_{J-P}^{J-1} S^{u/\rho} \qquad x \notin \partial\Omega$$

where $\wp_{J-P}^{J-1} = \left(P_{V_{J-P}} + \sum_{i=J-P}^{J-1} P_{W_i} \right)$ and $\partial_J^{(1)}$ is the standard decomposition of $\frac{d}{dx}$ defined as $\wp_{\text{J-P}}^{\text{J-1}} \frac{d}{dx} \wp_{\text{J-P}}^{\text{J-1}}$

- Using the multiresolution strategy to discretize the problem we represent the domain *P+1* times for a number *P* of different resolutions of the discretization

- There are P wavelet spaces and the coarse resolution scaling function space V_{J-P} $(\mathrm{P} \geq 1)$

- In the transform domain each representation of the solution defined at some resolution P must be supplemented by boundary conditions

Wavelet method of lines

- A traditional finite difference scheme replaces partial derivatives with algebraic approximations at grid points and solves the system of algebraic equations to obtain a numerical solution of the PDE
- The method of lines transforms the PDE into a vector ODE by replacing the spatial partial derivatives with their wavelet approximations but retaining the time derivatives
- The vector ODE is solved in time using a stiff ODE solver
- A method based on the backward differentiation formula (LSODE) from Lawrence Livermore Laboratories and an Euler method have been implemented in C/Fortran 90 on an IBM RS6000/590 and an Athlon 650
- The complexity of the method is $O(N\tau)$ for time discretization τ

PDEs in *d* space dimensions

- The entire multiresolution wavelet machinery presented so far can be extended to several space dimensions *d* by taking straight forward Cartesian products of appropriate approximation and scaling subspaces -- i.e. tensor products of appropriate wavelet bases -- to result in a fast wavelet transform of $O(N)$ for $N := n^d$ for spatial discretization n
- The imposition of boundary conditions for nonlinearly bounded domains is nontrivial but these are fortunately rare in PDE derivative valuation problems which usually are Cauchy problems on a strip

4. European Option Valuation

- The Black Scholes PDE is

$$\frac{\partial C}{\partial t}+\frac{1}{2}\sigma^2 S^2\frac{\partial^2 C}{\partial S^2}+rS\frac{\partial C}{\partial S}-rC=0$$

where S is stock price, σ is volatility and r is the risk free rate of interest

- Transform to the heat diffusion equation

$$\frac{\partial u}{\partial \tau}=\frac{\partial^2 u}{\partial x^2} \quad \text{for} \quad -\infty < x < \infty, \tau > 0$$

using $S = Xe^x$ and $t = T - 2\tau / \sigma^2$

- Then $C(S,T) := \max(S-X\ , 0)$ becomes for $k := 2r/\sigma^2$

$$C(x,\tau)=e^{-\frac{1}{2}(k-1)x-\frac{1}{2}(k+1)^2\tau}Xu(x,\tau)$$

European vanilla call

- For a vanilla European call option the boundary conditions are

$$C(0,t)=0,\ C(S,t)\sim S \text{ as } S\rightarrow\infty$$

- The boundary conditions for the transformed PDE are

$$u(x,\tau)=0 \text{ as } x\rightarrow-\infty$$

$$u(x,\tau)\sim e^{\frac{1}{2}(k+1)x+\frac{1}{4}(k+1)^2\tau} \text{ as } x\rightarrow\infty$$

$$u(x,0)=\max\left\{e^{\frac{1}{2}(k+1)x}-e^{\frac{1}{2}(k-1)x},0\right\}$$

- The heat equation solution can be transformed back to original variables as

$$C(S,t)=X^{\frac{1}{2}(k+1)}S^{\frac{1}{2}(1-k)}e^{\frac{1}{8}(k+1)^2\sigma^2(T-t)}u\left(\log(S/X),\frac{1}{2}\sigma^2(T-t)\right)$$

where $k=2r/\sigma^2$

Vanilla European Call

- S =10, strike =10, r = 5%, volatility = 20%, maturity=1 Year
- Exact value: 1.04505

Wavelet method of lines

Space Steps	Time Steps	Value	Solution time (seconds)
64	60	1.03515	.05
128	100	1.04220	.10
256	200	1.04502	.13
512	200	1.04505	.30
1024	200	1.04505	.90

Crank-Nicolson Finite Difference Method

Space Steps	Time Steps	Value	Solution time (seconds)
64	60	1.03184	.02
128	100	1.04184	.04
256	200	1.04426	.09
512	200	1.04486	.16
1024	200	1.04501	.30
2000	200	1.04505	.57

- Speedup : 1.9

Cash or nothing binary call

- The cash-or-nothing call option has a payoff

$$\Pi(S) = B\mathrm{H}(S - X)$$

where H is the Heaviside function, i.e. if at expiry the stock price $S > X$ the payoff is B

- The boundary conditions for this option in the transformed domain are

$$u(x,\tau) = 0 \text{ as } x \to -\infty$$

$$u(x,\tau) \sim \frac{B}{X} e^{\frac{1}{2}(k-1)x + \frac{1}{4}(k+1)^2 \tau} \text{ as } x \to \infty$$

$$u(x,0) = e^{\frac{1}{2}(k-1)x} \frac{B}{X} H\left(e^{\frac{1}{2}(k+1)x} - e^{\frac{1}{2}(k-1)x}, 0 \right)$$

Cash-or-nothing Call

- Same parameters as before, cash given B=3
- Exact value 1.59297

Wavelet method of lines

Space Steps	Time Steps	Value	Solution time (seconds)
128	100	1.49683	.10
256	200	1.54904	.13
512	200	1.59216	.30
1024	400	1.59288	1.02

Crank-Nicolson Finite Difference Method

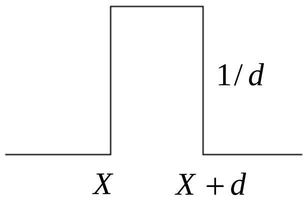
Space Steps	Time Steps	Value	Solution time (seconds)
128	200	1.46296	.04
256	400	1.53061	.10
512	400	1.56391	.18
1024	400	1.58046	.31
2048	800	1.58872	1.35
4096	800	1.59285	2.56

- Speedup : 2.5

Supershare binary call

- The supershare binary call option pays an amount *1/d* if the stock price lies between *X* and *X+d* at expiry
- The payoff of this option is

$$\Pi(S) = \frac{1}{d}\left(H(S-X) - H(S-X-d)\right)$$



which becomes the Dirac delta function in the limit as $d \searrow 0$

- The boundary conditions for this option in the transformed domain are

$$u(x,\tau) = 0 \text{ as } x \to -\infty$$

$$u(x,\tau) \sim 0 \text{ as } x \to \infty$$

$$u(x,0) = e^{\frac{1}{2}(k-1)x}\left(H(Xe^x - X) - H(Xe^x - X - d)\right)/dX$$

Supershare Binary Call

- Same parameters as before, parameter d=3
- Exact value 0.13855

Wavelet method of lines

Space Steps	Time Steps	Value	Solution time (seconds)
128	100	.12796	.10
256	200	.13310	.14
512	200	.13808	.30
1024	400	.13848	1.04

Crank-Nicolson Finite Difference Method

Space Steps	Time Steps	Value	Solution time (seconds)
128	200	.12369	.04
256	400	.13290	.09
512	400	.13435	.16
1024	400	13666	.34
2048	800	.13787	1.35
4096	800	.13800	2.56
8000	800	.13835	5.11

- Speedup : 4.9

5. Cross-currency Swap Valuation

- We consider a 10 year cross-currency cancellable fixed-fixed swap with quarterly reset dates, a 2 day decision period and exchange of principal on first and last days
- Similar to the floating-floating deals discussed in Dempster & Hutton (1997) and JP Morgan/Risk(1999)
- 1-factor extended Vasicek yield curve models lead to a 3D parabolic PDE of the form

$$\frac{\partial V}{\partial t} - \frac{1}{2}\nabla[\Lambda(t)(\nabla V)'\,] = 0$$

- Solved for the normalized deal value 39 times in backwards time with a backwards dynamic programming reset of the initial condition at each quarter
- Boundary conditions are set at 3σ of the state distributions and the computed deal is symmetric --value 0-- with the same initial term structures, mean reversions and forward volatility specification in both economies
- A 3D wavelet method of lines code in C/Fortran 90 and an explicit finite difference method in C have been implemented on an IBM RS6000/590 and an Athlon 650

Cross Currency Swap

- Domestic Fixed Rate=10%, Foreign Fixed Rate=10%
- Exact value : 0

Wavelet method of lines

Discretization	Value	Solution time (seconds)
20 x 8 x 8 x 8	-0.00082	1.2
20 x 16 x16 x16	-0.00052	6.54
20 x 32 x32 x 32	-0.00047	40.40
40 x 64 x 64 x 64	-0.00034	410.10
100 x 128 x 128 x 128	-0.00028	4240.30
160 x 256 x 256 x 256	-0.00025	53348.10

Explicit Finite Difference Method

	Value	Solution time (seconds)
20 x 8 x 8 x 8	-0.00109	0.28
20 x 16 x16 x16	-0.00101	1.70
20 x 32 x32 x 32	-0.00074	16.82
40 x 64 x 64 x 64	-0.00058	188.10
100 x 128 x 128 x 128	-0.00046	2421.6
160 x 256 x 256 x 256	-0.00038	33341.8

- Speedup : 81+

6. 3-Factor Interest Rate Swap Valuation

3D Gaussian Model

- The short rate is given by

$$\boldsymbol{r}(t) = s(t) + \boldsymbol{X}_1(t) + \boldsymbol{X}_2(t) + \boldsymbol{X}_1(t)$$

where $s(t)$ is a deterministic function

$$d\boldsymbol{X}_1 = \mu_1 \boldsymbol{X}_1 dt + \sigma_1 d\boldsymbol{W}_1$$

$$d\boldsymbol{X}_2 = \mu_2 \boldsymbol{X}_2 dt + \sigma_2 d\boldsymbol{W}_2$$

$$d\boldsymbol{X}_3 = \mu_3 \boldsymbol{X}_3 dt + \sigma_3 d\boldsymbol{W}_3$$

and $\boldsymbol{E}[d\boldsymbol{W}_i d\boldsymbol{W}_j] = \rho_{ij} dt$

- Bond prices have a closed form in this model

- If V denotes the value of the derivative security under the spot measure $\exp(-\int_0^t \boldsymbol{r}(u)du)\, V(\boldsymbol{X}_{1t}, \boldsymbol{X}_{2t}, \boldsymbol{X}_{3t}, t)$ is a martingale.
- Using Ito's lemma the PDE satisfied by the value function is

$$\mu_1 X_1 \frac{\partial V}{\partial X_1} + \mu_2 X_2 \frac{\partial V}{\partial X_2} + \mu_3 X_3 \frac{\partial V}{\partial X_3} + \frac{1}{2}\sigma_1^2 \frac{\partial^2 V}{\partial X_1^2} + \frac{1}{2}\sigma_2^2 \frac{\partial^2 V}{\partial X_2^2} + \frac{1}{2}\sigma_3^2 \frac{\partial^2 V}{\partial X_3^2}$$

$$+\rho_{12}\sigma_1\sigma_2 \frac{\partial^2 V}{\partial X_1 X_2} + \rho_{13}\sigma_1\sigma_3 \frac{\partial^2 V}{\partial X_1 X_3} + \rho_{23}\sigma_2\sigma_3 \frac{\partial^2 V}{\partial X_2 X_3} - (s(t) + X_1 + X_2 + X_3)V + \frac{\partial V}{\partial t} = 0$$

- This PDE must be solved with appropriate boundary conditions

Deal Specification

Fixed for floating LIBOR swap

- Payment at end of each period is $p_j = Z\delta_j[L(t_{j-1}, t_j) - r^*]$ where r^* is the fixed rate, δ_j is the accrual factor, Z is the notional principal and $L(t_{j-1}, t_j)$ is the LIBOR rate for period t_{j-1}, t_j
- Payment $B(t_{j-1}, t_j)p_j$ at the end of each period
- Value of the swap at $t < T$ is the sum of the expected present values of all future payments
- PDE solved with terminal condition for each period given by $V(t_{j-1}^-) = B(t_{j-1}, t_j)p_j + V(t_{j-1})$ over the spatial grid points

Bermudan Swaption

- The counterparty has the option to enter the swap at the beginning of every period
- The terminal condition for each period is given by

$$V(t_j^-) = \max\{\Sigma, V(t_j)\}$$

where Σ represents the sum of the conditional expected present values of all future payments after t_j

- The PDE is solved for each period with the terminal condition given above

Bermudan Swaption

- 1 Year, 2 Periods
- Fixed Rate=5%, Initial Flat Term Structure=5%
- MC value : 0.09921

Wavelet method of lines

Discretization	Value	Solution time (seconds)
100 x 32 x 16 x 8	.10302	.80
100 x 64 x 32 x 16	.10091	16.47
100 x 128 x 64 x 16	.09966	130.05
100 x 256 x 128 x 16	.09952	702.87

Dufort Frankel Explicit Finite Difference Method

Discretization	Value	Solution time (seconds)
100 x 32 x 16 x 8	.10451	.15
100 x 64 x 32 x 16	.10105	3.23
100 x 128 x 64 x 16	.09987	24.80
100 x 256 x 128 x 16	.09970	148.26

7. Benchmark Simulation methods for Bermudan Swaptions

- Can we use simulation methods to price Bermudan swaptions?
  - Choose a (sub-optimal) exercise rule and use normal Monte-Carlo to get a negatively biased estimate
  - Construct an approximation to the option value at a mesh of points determined from randomly generated sample paths working backwards
- Three approaches:
  - Andersen (1999)
    - Define a score function *f* (e.g. the immediate value exercise value) and an exercise rule 'exercise at *t* if $f(\boldsymbol{S}_t)>H_t$ a threshold determined by solving an optimization problem`
  - Broadie and Glasserman (1997) Stochastic Mesh
    - Generates consistent, but *positively* and negatively biased estimators
  - Longstaff and Schwartz (1998) (similar to Pedersen (1999))
    - Approximate the value-of-continuation as a linear combination of user supplied basis functions

Longstaff & Schwartz method

- Simulate a set of random paths for $\boldsymbol{S}$
  - Denote by S_t^i the value of $\boldsymbol{S}_t$ at time t on the ith path and by $D(S_t^i,t)$ the stochastic discount factor to apply from $(S_{t-1}^i,t-1)$ to (S_t^i,t)
- Construct approximation to the swaption value at these points recursively
  - Set $V(s_T^i,T)$ to be the value of exercise at the final time T
  - Choose a set of basis functions $\{b_j(s,t)\}$ and use least-squares to construct an approximation $W(s,T\text{-}1)=\Sigma\, a_j(T\text{-}1)\, b_j(s,T\text{-}1)$ to the value of continuation at time T-1 by regressing $D(S_T^i,T)\, V(S_T^i,T)$ on $b_j(S_{T-1}^i,T\text{-}1)$
  - Yields $W(s,T\text{-}1)$ which can be used to define an exercise rule in an obvious way
  - Finally define $V(S_{T-1}^i,T\text{-}1)$ to be the value of immediate exercise if we choose to exercise and $D(S_T^i,T)\, V(S_T^i,T)$ if we choose to continue
- The time-0 option value is obtained by averaging the $D(S_1^i,1)V(S_1^i,1)$

Longstaff & Schwartz method

- Many improvements are possible
  - Use European swaption prices as *outer* control variates in the final Monte-Carlo calculation
  - Use the value of the remaining swap as an *inner* control variable when performing the least-squares regression (cf. Hedged Monte Carlo of Potters, Bouchaud & Sestovic (2000))
  - Improve the exercise strategy using extra information:
    - Do not exercise if the current value of the remaining swap is below zero
    - (stronger) Do not exercise if the value of one of the remaining European swaptions is more than current value of remaining swap
  - Since the computations spend virtually all time pricing European swaptions use a very fast approximation for these. For example for a 20NC10, we can price more than 100,000/s on a 650Mhz Athlon with an error less than 0.01 b.p.

7. Conclusions and Further Work

- *O(N)* wavelet based PDE methods generalize $O(N \log_2 N)$ spectral methods without their drawbacks
- Wavelets are ideally suited for complex derivative valuations which involve several space scales e.g. due to payoff curvature or discontinuities and result in greater accuracy at a given discretization level with substantial speedups -- using prototype code -- over optimized finite difference codes in dimensions up to 3
- We are currently developing a thresholded 3D wavelet code which should improve both speedup and memory use by an order of magnitude through sparse wavelet representation
- However fast Monte Carlo techniques are currently the method to beat!
- On to 2 and 3 factor cross currency swap valuation in 5 and 7D…