Using radial basis functions for option pricing

Elisabeth Larsson

Division of Scientific Computing
Department of Information Technology
Uppsala University

Actuarial Mathematics Workshop, March 19, 2013,
University of Leicester
Outline

An introduction to RBFs

The option pricing test problem

Basic RBF approximation method

Variations of the theme

Summary
Introduction and motivation

**Basis functions:** $\phi_j(x) = \phi(\|x - x_j\|)$. Translates of one single function rotated around a center point.

**Example:** Gaussians

$\phi(\varepsilon r) = \exp(-\varepsilon^2 r^2)$

**Approximation:**

$s_\varepsilon(x) = \sum_{j=1}^{N} \lambda_j \phi_j(x)$

**Solution:**

Collocation with data yields $\{\lambda_j\}_{j=1}^{N}$. 

---

E. Larsson, Leicester, March 19, 2013  (3 : 23)
Why use RBFs for option pricing?

Advantages

- Flexibility wrt to the computational domain.
- Allows adaptive node placement
- As easy in $d$ dimensions.
- Spectral accuracy / exponential convergence.
- Allows direct calculation of $\Delta$ and $\Gamma$.

Challenges

- Parameter selection strategies
- Ill-conditioning
- Computational cost
The basic option pricing test problem

**Purpose:** To determine the current value of an option.

**Example:** European basket call option.

Expiration date: \( T = \text{Dec 30, 2013} \)
Strike price: \( K = 200 \text{ SEK} \)
Dimensions: \( d = 3 \) the number of underlying assets

The multi-dimensional Black-Scholes equation:

\[
\frac{\partial u}{\partial t} = r \sum_{i=1}^{d} x_i \frac{\partial u}{\partial x_i} + \frac{1}{2} \sum_{i,j=1}^{d} \left[ \sigma \sigma^T \right]_{ij} x_i x_j \frac{\partial^2 u}{\partial x_i \partial x_j} - ru.
\]

**Variables:** \( x \in \mathbb{R}^d \) asset prices, \( t \) time left to expiration.

**Parameters:** \( \sigma \) volatility matrix, \( r \) risk free interest rate.
Initial and boundary conditions

Contract function

\[ \Phi(x) = \max \left( 0, \frac{1}{d} \sum_{i=1}^{d} x_i - K \right) \]

Boundary conditions

\[ u(x, t) \to \frac{1}{d} \sum_{i=1}^{d} x_i - Ke^{-rt} \]

as \( \|x\| \to \infty \).

S. Jansson and J. Tysk, 2006: Feynman-Kac formulas for Black-Scholes type operators
Discretization in time and space

Solution form: \( u(x, t) \approx \sum_{j=1}^{N} \lambda_j(t) \phi_j(x) \)

Black-Scholes: \( \sum_{j=1}^{N} \lambda_j'(t) \phi_j(x) = \sum_{j=1}^{N} \lambda_j(t) \mathcal{L} \phi_j(x) \)

Time:
\[
\frac{1}{k} \sum_{j=1}^{N} (\lambda_j^{n+1} - \lambda_j^n) \phi_j(x) = \sum_{j=1}^{N} (\alpha \lambda_j^{n+1} + (1 - \alpha) \lambda_j^n) \mathcal{L} \phi_j(x)
\]

Boundary: \( \sum_{j=1}^{N} \lambda_j^{n+1} \phi_j(x_i) = g(x_i, t^{n+1}), \quad x_i \text{ at } \partial \Omega \)

Interior: \( \sum_{j=1}^{N} \lambda_j^{n+1} a_j(x_i) = \sum_{j=1}^{N} \lambda_j^n b_j(x_i), \quad x_i \text{ in } \Omega \)

Initially: \( \sum_{j=1}^{N} \lambda_j^0 \phi_j(x_i) = \Phi(x_i) \)
How can we play with the nodes?

- Square domain, \( N = 1165 \)

- Triangular domain, \( N = 603 \)

- Adapted nodes, \( N = 599 \)

- Change of domain \( \Rightarrow N \rightarrow N/d ! \)

- Redistribution improves local accuracy
How can we play with the nodes?

- Square domain, $N = 1165$
- Triangular domain, $N = 603$
How can we play with the nodes?

- Square domain, \( N = 1165 \)
- Triangular domain, \( N = 603 \)
How can we play with the nodes?

- Square domain, $N = 1165$
- Triangular domain, $N = 603$
- Adapted nodes, $N = 599$
How can we play with the nodes?

- Square domain, $N = 1165$
- Triangular domain, $N = 603$
- Adapted nodes, $N = 599$

- Change of domain $\Rightarrow N \rightarrow N/d!$
- Redistribution improves local accuracy
Numerical results for different node sets

- Error measured in the region of interest.
- Triangle and square same accuracy.
- Adaptive an order of magnitude better.

Pettersson, Larsson, Marcusson, Persson, 2008: Improved radial basis function methods for multi-dimensional option pricing.
Another game of nodes

Domain and interior nodes are invariant with respect to
- 90 degree rotations,
- reflections in diagonals and axes.

Employ the generalized Fourier transform to reduce memory and computational cost.

However, operator must have same invariance.
Transforming the Black-Scholes equation into the heat equation

\[
\frac{\partial u}{\partial t} = r \sum_{i=1}^{d} x_i \frac{\partial u}{\partial x_i} + \frac{1}{2} \sum_{i,j=1}^{d} \left[ \sigma \sigma^T \right]_{ij} x_i x_j \frac{\partial^2 u}{\partial x_i \partial x_j} - ru.
\]

- Change variables: \( x = \exp(A^T(Q^Ty + b)) \)
- Change function: \( u(t, y) = e^{\gamma t + \xi^T y} p(t, y) \)

\[
\frac{\partial p}{\partial t} = \Delta_y p
\]

⋆ \( A \) is computed from \( A^T A = \frac{1}{2} \sum_{k=1}^{d} (\sigma_k \sigma_k^T) \).
⋆ \( Q \) and \( b \) are arbitrary.
⋆ \( \gamma \) and \( \xi \) depend on \( \sigma \), \( r \), \( A \) and \( Q \).
What happens with the domain and the nodes?

The transformation leads to automatically adapted node placement.
Numerical results with the generalized Fourier transform

Red Uni square, Black GFT, Blue Adapted tri

- Given \( N \) \( \Rightarrow \) Same accuracy for square and RBF-GFT.
- Lower cost for RBF-GFT. 2D \( \rightarrow \) 48, 3D \( \rightarrow \) 864.
- Adapted is more efficient at least in 2D.

Larsson, Åhlander, Hall, 2008: Multi-dimensional option pricing using radial basis functions and the generalized Fourier transform
Exploiting the spectral accuracy

- Parabolic PDE.
- We need to get low frequencies right.
- Result sensitive to node placement.

⇒ We use more nodes than we would like.
Exploiting the spectral accuracy (cont.)

Collocation
Least squares

RBFs in finance

Outline
An introduction to RBFs
Model problem
Basic method
Examples
Summary

E. Larsson, Leicester, March 19, 2013 (15 : 23)
A multi-level least-squares RBF approach

- Coarse grid with small $\varepsilon$ for smooth part.
- Fine grid with larger $\varepsilon$ for initial non-smoothness.
- Boundary conditions are collocated (necessary).
- Computational cost turns out to be less than for a collocation approach.

**Larsson, Gomes, 2013: A least squares multi-level radial basis method with applications in finance**
Errors for the two-grid least-squares method

**Error evolution**

- **blue:** \( N = 40, \varepsilon = 10 \)
- **red:** \( N = 21, \varepsilon = 0.8 \)
- **black:** Two level

150 least-squares evaluation points.
Two-dimensional problem

\[ N_f = 320, \ N_c = 76, \ N_e = 932, \ \varepsilon_f = 4, \ \varepsilon_c = 1 \]

Summary

- Flexibility wrt the computational domain ⇒ Easy way to save computations by going to simplex.
- Adaptivity, least squares or a multi-level/multi-scale approach is needed.
- Any of the discussed approaches can be used in higher dimensions, but cost becomes an issue.
- Future direction: Adaptive partition of unity RBF-methods.
Currently: Partition of unity RBF-methods

*(RBF-PU for interpolation Wendland 2002)*

Global approximant:

\[ \tilde{u}(x) = \sum_{i=1}^{M} w_i(x) u_i(x), \]

where \( w_i(x) \) are weight functions.

Local RBF approximants:

\[ u_i(x) = \sum_{j=1}^{N_i} \lambda_j^i \phi_j(x). \]

Applying operators:

\[ \Delta \tilde{u}(x) = \sum_{i=1}^{M} \Delta w_i u_i + 2 \nabla w_i \cdot \nabla u_i + w_i \Delta u_i \]

- Sparsity reduces memory and computational cost.
- Subdomain approach introduces parallelism.
Practical challenges in RBF approximation

Conditioning for small $\varepsilon$ and large $N$

- Spectral convergence with $N$ requires fixed $\varepsilon$.
- For smooth solutions, the best $\varepsilon$ is small.
- We need to compute in the red region.

Computational cost

- Coefficient matrices are typically dense (for infinitely smooth RBFs).
- Direct methods are $O(N^3)$ and known fast methods are most efficient for larger $\varepsilon$. 

E. Larsson, Leicester, March 19, 2013 (21 : 23)
The RBF-QR method: Idea

- The Gaussian RBFs are expanded in terms of

\[ \begin{align*}
T^c_{j,m}(x) &= e^{-\varepsilon^2 r^2} r^{2m} T_{j-2m}(r) \cos((2m + p)\theta), \\
T^s_{j,m}(x) &= e^{-\varepsilon^2 r^2} r^{2m} T_{j-2m}(r) \sin((2m + p)\theta),
\end{align*} \]

leading to \( \Phi(x) = C \cdot D \cdot T(x) \), where \( c_{ij} \) is \( O(1) \) and
\( D = \text{diag}(O(\varepsilon^0, \varepsilon^2, \varepsilon^2, \varepsilon^4, \varepsilon^4, \varepsilon^6, \ldots)) \).

- Then \( C \) is QR-factorized so that

\[ \Phi(x) = Q \cdot \begin{bmatrix} R_1 & R_2 \end{bmatrix} \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} \cdot T(x) \]

- Form a new basis (in the same space)

\[ \Psi(x) = D_1^{-1} R_1^{-1} Q^H \Phi(x) = \begin{bmatrix} I & \tilde{R} \end{bmatrix} \cdot T(x). \]
Stable computation as $\varepsilon \to 0$ and $N \to \infty$

The RBF-QR method allows stable computations for small $\varepsilon$. (*Fornberg, Larsson, Flyer 2011*)

Consider a finite non-periodic domain.

**Theorem (Platte, Trefethen, and Kuijlaars 2010):**
Exponential convergence on equispaced nodes $\Rightarrow$ exponential ill-conditioning.

**Solution #1:**
Cluster nodes towards the domain boundaries.

![Graph](image)