UPPSALA UNIVERSITET

Radial Basis Functions generated Finite Differences (RBF-FD) for Solving High-Dimensional PDEs in Finance

S. Milovanović, L. von Sydow

Uppsala University
Department of Information Technology
Division of Scientific Computing

SIAM
Student Computational Finance Day
TU Delft, Netherlands, 2016-May-23

Overview

UPPSALA UNIVERSITET

Introduction
- Option Pricing

Methods
- Radial Basis Functions generated Finite Difference methods

Results
- Numerical Experiments

Conclusion
- Further Research

Option Pricing

UPPSALA UNIVERSITET

- Black-Scholes-Merton model in higher dimensions

$$\begin{cases} dB(t) = rB(t)dt, \\ dS_1(t) = \mu_1 S_1(t)dt + \sigma_1 S_1(t)dW_1(t), \\ dS_2(t) = \mu_2 S_2(t)dt + \sigma_2 S_2(t)dW_2(t), \\ \vdots \\ dS_D(t) = \mu_D S_D(t)dt + \sigma_D S_D(t)dW_D(t). \end{cases} \tag{1}$$

- Option price

$$u(S_1(t), \ldots, S_D(t), t) = e^{-r(T-t)}\mathbb{E}_t^Q[g(S_1(T), \ldots, S_D(T))].$$

- The Black-Scholes-Merton equation

$$\begin{cases} \frac{\partial u}{\partial t} + r\sum_i^D s_i \frac{\partial u}{\partial s_i} + \frac{1}{2}\sum_{i,j}^D \rho_{ij}\sigma_i\sigma_j s_i s_j \frac{\partial^2 u}{\partial u_{s_i s_j}} - ru = 0, \\ u(s_1, s_2, \ldots, s_D, T) = g(s_1, s_2, \ldots, s_D). \end{cases} \tag{2}$$

BENCHOP – The BENCHmarking project in Option Pricing*

UPPSALA UNIVERSITET

- Monte Carlo methods converge slowly but are increasingly competitive in higher dimensions.
- Fourier-methods are fast, especially the COS-method.
- Finite difference methods are reasonably fast in lower dimensions but suffer from the curse of dimensionality.
- Methods based on approximations with Radial Basis Functions have the potential to be competitive in higher dimensions.

* L. von Sydow, L.J. Höök, E. Larsson, E. Lindström, S. Milovanovic, J. Persson, V. Shcherbakov, Y. Shpolyanskiy, S. Sirén, J. Toivanen, J. Waldén, M. Wiktorsson, J. Levesley, J. Li, C.W. Oosterlee, M.J. Ruijter, A. Toropov, and Y. Zhao. In International Journal of Computer Mathematics, volume 92, pp 2361-2379, 2015.

Radial Basis Functions Method

- Discretize space using N nodes.
- Approximate solution

$$u(s,t) \approx \sum_{k=1}^{N} \lambda_k(t)\phi(\varepsilon\|s - s_k\|),\ k = 1, 2, \ldots, N, \tag{3}$$

  where ϕ is a radial basis function and ε is a shape parameter.
- The linear combination coefficients λ_k are found by enforcing interpolation conditions.

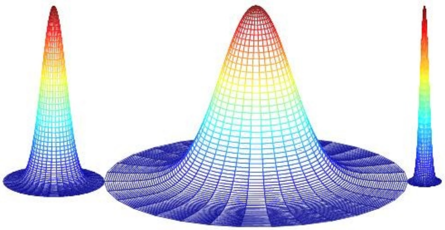


Figure 1: Gaussian RBFs.

Local Radial Basis Functions Methods

UPPSALA
UNIVERSITET

- The global approximation leads to a dense linear system of equations which tends to be ill-conditioned when ε is small
  $\Rightarrow$ Local RBF approximations might be a better approach!

Local Radial Basis Functions Methods

UPPSALA
UNIVERSITET

- The global approximation leads to a dense linear system of equations which tends to be ill-conditioned when ε is small
  $\Rightarrow$ Local RBF approximations might be a better approach!

- Ideas
  - Come up with a localization which leads to a sparse linear system of equations.
    - Radial Basis Functions Partition of Unity (RBF-PU)
    - Radial Basis Functions generated Finite Differences (RBF-FD)
  - Come up with basis transformation which will heal the ill-conditioning.
    - RBF-QR
    - RBF-GA

Radial Basis Functions generated Finite Differences

UPPSALA UNIVERSITET

- Try to exploit the best properties from both FD and RBF with minimal loss.
- For each point s_i in space, define its neighborhood of $M-1$ points and observe it as a stencil.
- Approximate the differential operator at every point

$$[Lu(s)]_i \approx \sum_{k=1}^{M} w_k^{(i)} u_k^{(i)}. \tag{4}$$

Radial Basis Functions generated Finite Differences

UPPSALA UNIVERSITET

- Try to exploit the best properties from both FD and RBF with minimal loss.
- For each point s_i in space, define its neighborhood of $M-1$ points and observe it as a stencil.
- Approximate the differential operator at every point

$$[Lu(s)]_i \approx \sum_{k=1}^{M} w_k^{(i)} u_k^{(i)}. \tag{4}$$

- Compute the RBF-FD weights and place them in matrix W

$$\begin{bmatrix} \phi(\|s_1^{(i)} - s_1^{(i)}\|) & \dots & \phi(\|s_1^{(i)} - s_M^{(i)}\|) \\ \vdots & \ddots & \vdots \\ \phi(\|s_M^{(i)} - s_1^{(i)}\|) & \dots & \phi(\|s_M^{(i)} - s_M^{(i)}\|) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_M \end{bmatrix} = \begin{bmatrix} [L\phi(\|s - s_1^{(i)}\|)]_{s=s_i} \\ \vdots \\ [L\phi(\|s - s_M^{(i)}\|)]_{s=s_i} \end{bmatrix}.$$

Adding polynomial terms

UPPSALA UNIVERSITET

$$\begin{bmatrix} \phi(\|s_1^{(i)} - s_1^{(i)}\|) & \dots & \phi(\|s_1^{(i)} - s_M^{(i)}\|) & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \phi(\|s_M^{(i)} - s_1^{(i)}\|) & \dots & \phi(\|s_M^{(i)} - s_M^{(i)}\|) & 1 \\ 1 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_M \\ w_{M+1} \end{bmatrix} = \begin{bmatrix} [L\phi(\|s - s_1^{(i)}\|)]_{s=s_i} \\ \vdots \\ [L\phi(\|s - s_M^{(i)}\|)]_{s=s_i} \\ L1 \end{bmatrix}.$$

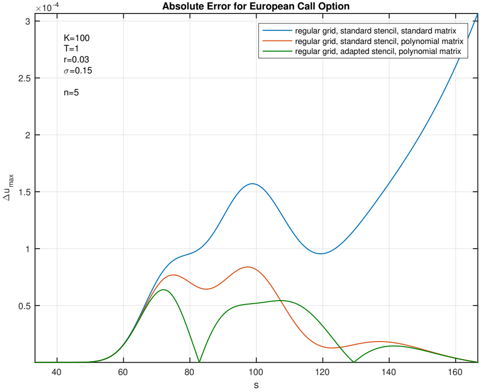


Figure 2: Error in the solution.

Disctretization

UPPSALA UNIVERSITET

- Discretize the Black-Scholes-Merton equation operator in space using RBF-FD

$$u_t = -\left[r\sum_{i}^{D} s_i u_{s_i} + \frac{1}{2}\sum_{i,j}^{D} \rho_{ij}\sigma_i\sigma_j s_i s_j u_{s_i s_j} - ru\right] \approx Wu.$$

- Integrate in time using the standard implicit schemes
  - BDF-1,
  - BDF-2.

$$\Rightarrow$$

$$Au^{n+1} = b^n$$

Solution of linear systems

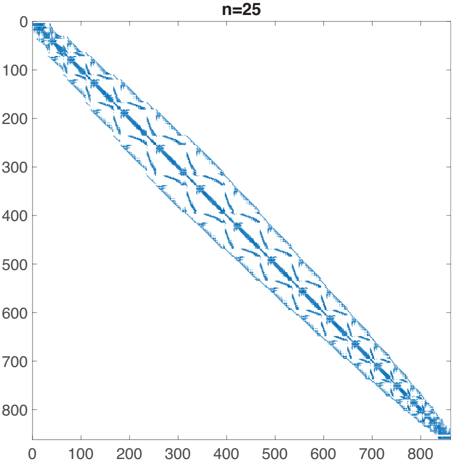


Figure 3: Structure of A.

Grids

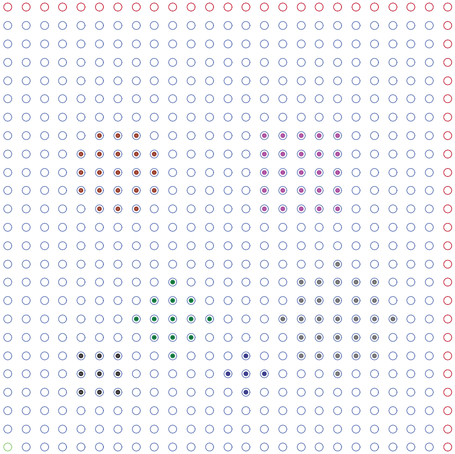


Figure 4: The nearest-neighbor based stencils used for approximating the differential operator.

Error distribution

UPPSALA UNIVERSITET

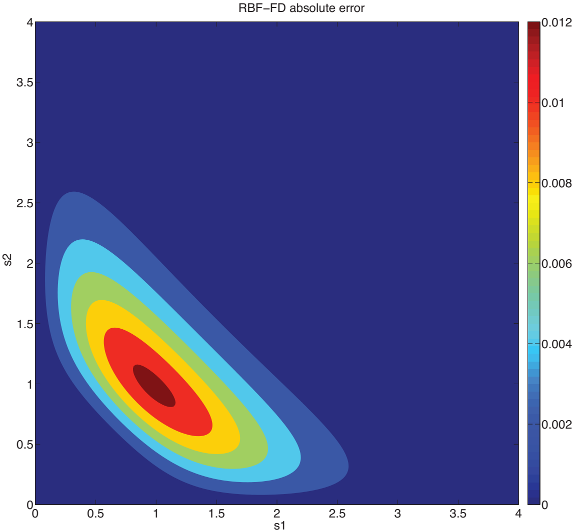


Figure 5: The absolute error distribution computed using 41 point in each dimension and a 5-point stencil on a full regular grid.

Grids

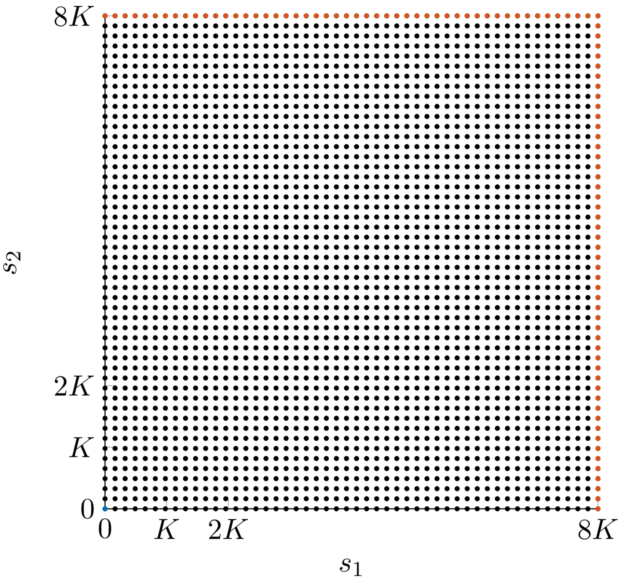


Figure 6: Square domain.

Grids

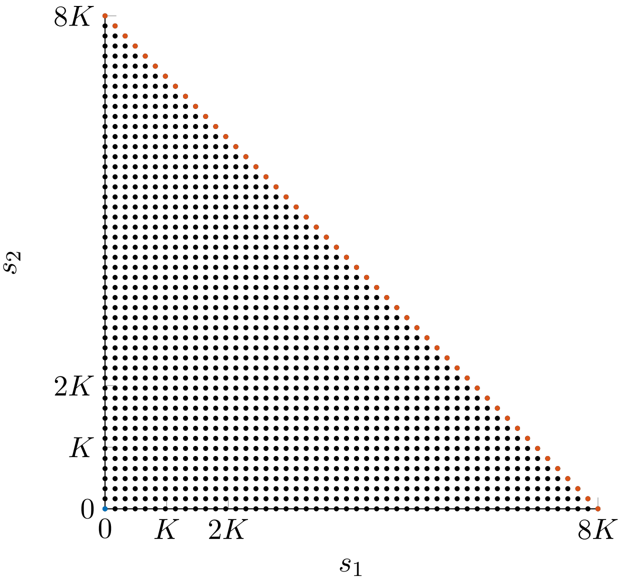


Figure 7: Triangular grid.

Grids

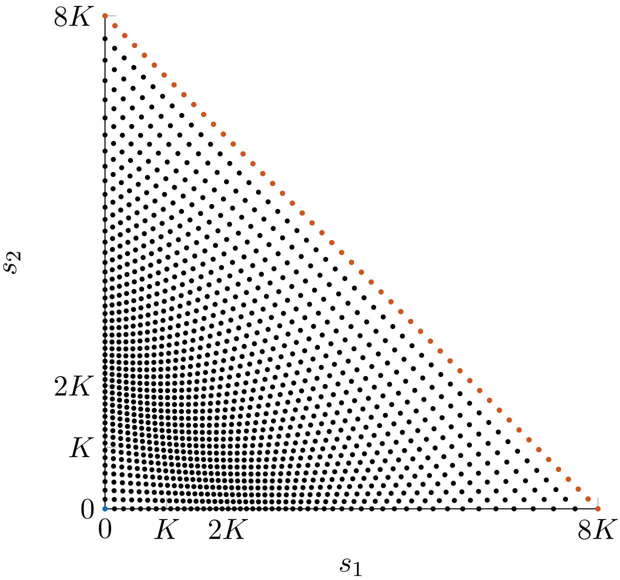


Figure 8: Adapted grid.

Grids

UPPSALA
UNIVERSITET

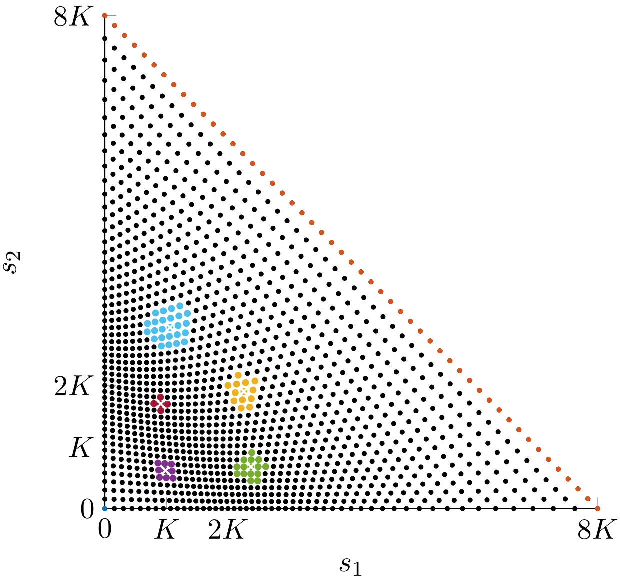


Figure 9: Adapted grid with some possible stencils.

Experiment

UPPSALA UNIVERSITET

- European-style basket option with
  $T = 1$,
  $K = 1$,
  $r = 0.03$,
  $\sigma_1 = \sigma_2 = 0.15$,
  $\rho = 0.5$.
- Payoff function

$$u(s_1, s_2, T) = \left[\frac{s_1(T) + s_2(T)}{2} - K\right]^+. \tag{5}$$

Details

UPPSALA
UNIVERSITET

- Boundary conditions

$$u(0, 0, t) = 0, \tag{6}$$

$$u(s_1^*, s_2^*, t) = \frac{1}{2}\left[s_1^*(t) + s_2^*(t)\right] - e^{-r(T-t)}K. \tag{7}$$

- RBF choice: Gaussian

$$\phi(r) = e^{-(\varepsilon \cdot r)^2}. \tag{8}$$

- Error measured on a subdomain: $(s_1, s_2) \in [\frac{1}{3}K, \frac{5}{3}K]$.

Solution

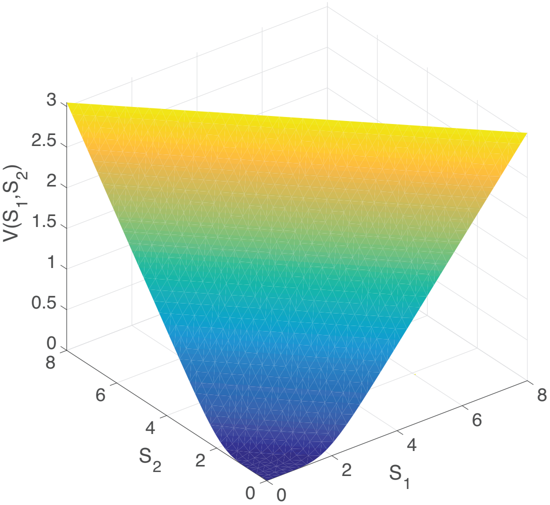


Figure 10: The computed solution on a regular triangular grid.

Choice of ε, $n = 3$.

UPPSALA
UNIVERSITET

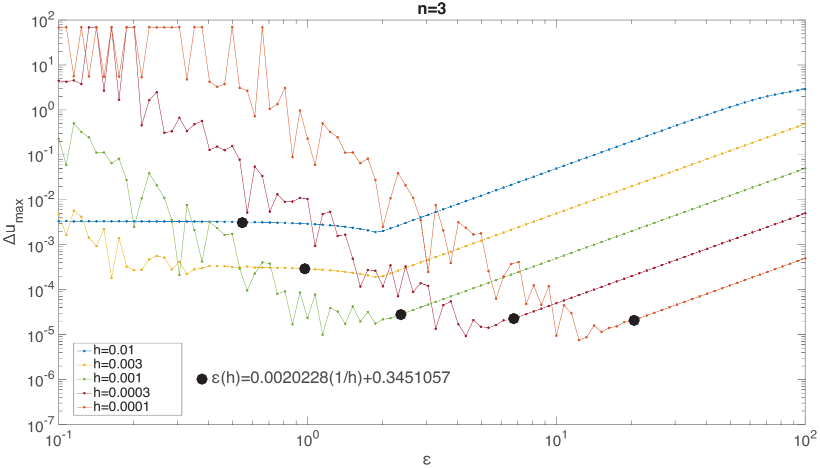


Figure 11: Error as a function of ε.

Choice of ε, $n = 7$.

UPPSALA UNIVERSITET

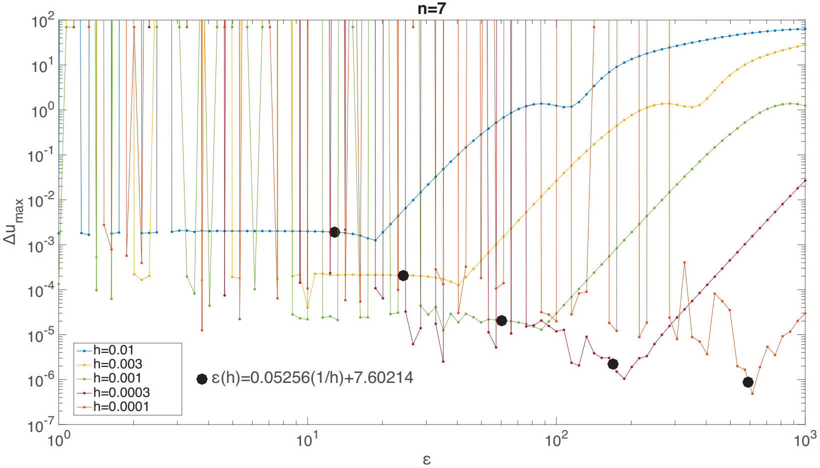


Figure 12: Error as a function of ε.

Results

UPPSALA UNIVERSITET

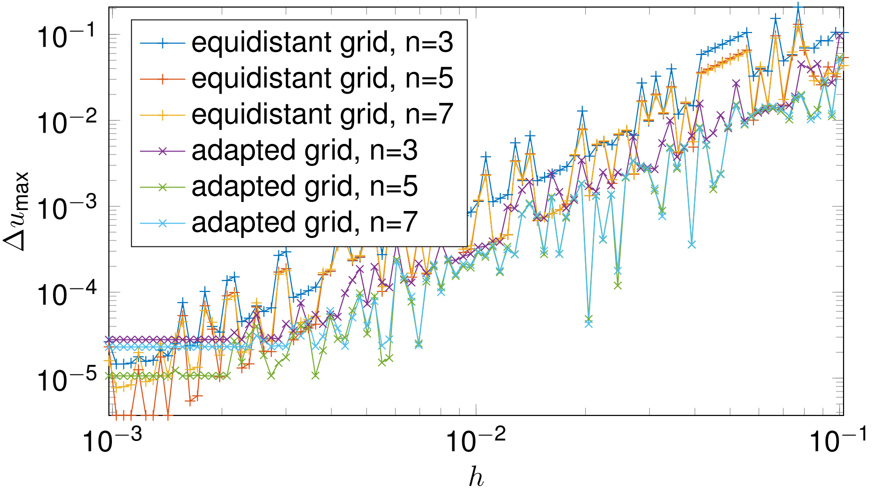


Figure 13: Spatial Convergence.

Results

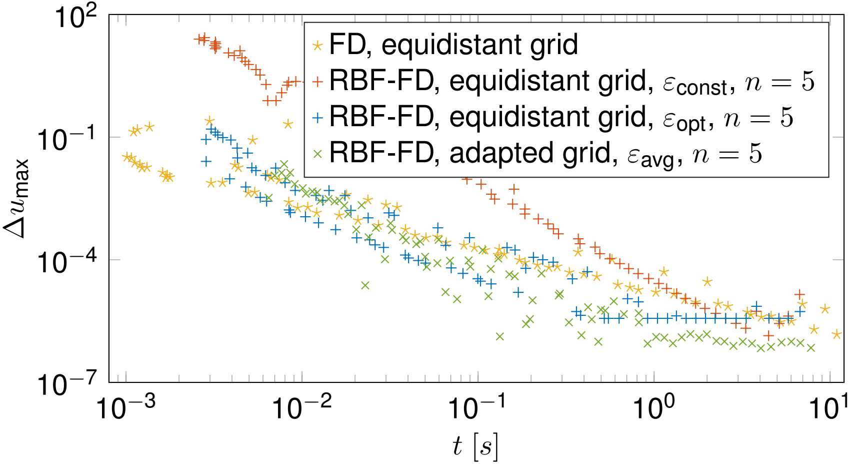


Figure 14: Computational performance.

Conclusions

UPPSALA
UNIVERSITET

Proposed method:

- Works equally well with American options using operator splitting.
- Allows for local grid refinement which may deliver high accuracy in desired regions of the computational domain.
- Performs well due to sparsity and reduction in the computational domain allowed by the mesh-free discretization.
- Promises extendability to higher dimensional problems as well as for other models in the field.

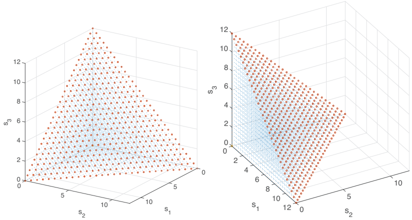
Future Research

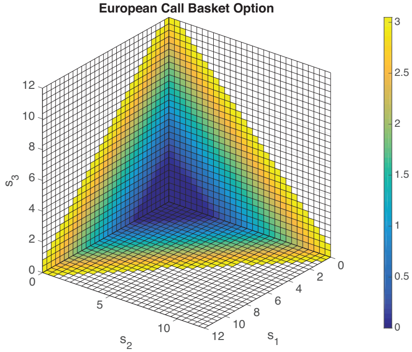
UPPSALA UNIVERSITET

- Finalize study of shape parameter behavior.
- Developing adaptivity both in node placement and stencil size (hp-adaptivity).
- Optimizing stencils close to the boundaries.
- Smoothing of the terminal condition.
- Least squares instead of interpolation.
- Solving advanced financial models.

Future Research

- Higher dimensional problems.





Thank you for your attention.

UPPSALA
UNIVERSITET