Notes on the BENCHOP implementations for the RBF-MLT method

Yangzhang Zhao(yz177@le.ac.uk)
Supervisor: Jeremy Levesley(jl1@le.ac.uk)

March 16, 2015

Abstract

This text describes the RBF-MLT method and its implementation for the BENCHOP-project.

1 Treating time as a spatial dimension

The single asset Black-Scholes equation is

$$\frac{\partial U}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 U}{\partial S^2} + rS\frac{\partial U}{\partial S} - rU = 0, \tag{1}$$

is time dependent, where U stands for option price, S means stock price, t is time, σ denotes volatility and r is the risk-free rate. When implementing a Radial Basis Function (RBF) method we approximate in space using an RBF, and in time we use a time-stepping method, such as the θ-method. .

Instead, let us treat this equation as a two-dimensional equation in space. If we do this we would like to scale the time variable so that the equation is defined on a square:

$$k\frac{\partial U}{\partial \tau} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 U}{\partial S^2} + rS\frac{\partial U}{\partial S} - rU = 0, \tag{2}$$

where $\tau = kt$. This can be rewritten as

$$\mathcal{L}U = 0. \tag{3}$$

In this way, we convert the problem of solving the partial differential equation (PDE) by one dimensional approximation and time-stepping, to collocation in two dimensions.

We approximate the solution U on a uniform grid of size h, which we call X_h. Then, the approximation to U is of the form

$$U = \sum_{x \in X_h} \lambda_x \phi(\|(\tau,\ S) - x\|_2), \tag{4}$$

where ϕ is the multiquadric RBF

$$\phi(r) = \sqrt{r^2 + c^2}. \tag{5}$$

In our algorithm we choose the shape parameter c to be $2h$ or $3h$, which depends on the specific problem. We observe that the larger shape parameter is useful when solving the barrier option problem, and we will explore this issue in future work.

Substituting our approximation into (2), and collocating, we can construct a underspecified linear system

$$P\lambda = \begin{pmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,N} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ P_{N,1} & P_{N,2} & \cdots & P_{N,N} \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{6}$$

where

$$P = k\Phi_\tau + \frac{1}{2}\sigma^2\Phi_{SS} + r\Phi_S - r\Phi. \tag{7}$$

Here $\Phi[x,y] = \phi(\|x-y\|)$, $\Phi_\tau[x,y] = \frac{x_\tau - y_\tau}{\phi(\|x-y\|)}$, $\Phi_S[x,y] = x_S \frac{x_S - y_S}{\phi(\|x-y\|)}$, $\Phi_{SS}[x,y] = x_S^2\left(\frac{1}{\phi(\|x-y\|)} - \frac{(x_S-y_S)^2}{\phi^3(\|x-y\|)}\right)$, $x \in X_h/(\partial\Omega), y \in X_h$, where $\partial\Omega$ is the boundary on which the boundary conditions are specified.

We implement the boundary conditions as follows:

$$\begin{aligned} U(kT,S) &= g(kT,S) \text{ (payoff function)} \\ U(\tau,\overline{S}) &= f(\tau,\overline{S}),\ \overline{S}\ on\ \partial\Omega \text{ (boundary conditions).} \end{aligned}$$

This gives a square linear system for the solution coefficients λ_x

$$\overline{P}\lambda = \begin{pmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,N} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,N} \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{i,1} & \phi_{i,2} & \cdots & \phi_{i,N} \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{j,1} & \phi_{j,2} & \cdots & \phi_{j,N} \\ \phi_{j+1,1} & \phi_{j+1,2} & \cdots & \phi_{j+1,N} \\ \vdots & \vdots & \vdots & \vdots \\ \phi_{N,1} & \phi_{N,2} & \cdots & \phi_{N,N} \end{pmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_i \\ \vdots \\ \lambda_j \\ \lambda_{j+1} \\ \vdots \\ \lambda_N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ f(X_i) \\ \vdots \\ f(X_j) \\ g(X_{j+1}) \\ \vdots \\ g(X_N). \end{pmatrix} \tag{8}$$

After solving the above linear system, we have an estimate $U^1 \approx U$. We create a new approximation by solving for the residual on a grid with half the grid size $h/2$.

$$\begin{aligned} \mathcal{L}U^2 &= \mathcal{L}(U - U^1) = -\mathcal{L}(U^1) \\ U^2(kT,S) &= g(kT,S) - U^1(kT,S) \\ U^2(\tau,\overline{S}) &= f(\tau,\overline{S}) - U^1(\tau,\overline{S}),\ \overline{S}\ on\ \partial\Omega \end{aligned}$$

Similarly, we can estimate a solution $U^2 \approx U - U^1$. We can continue this process to find generate increasingly small residuals:

1st Level: $U^1 \approx U$,

2nd Level: $U^2 \approx U - U^1$,

3rd Level: $U^3 \approx U - U^1 - U^2$.

Having set an initial tolerance, we iterate until the residual falls below this tolerance. This takes $N(\epsilon)$ iterations. The final approximation to U is $\sum_{i=1}^{N(\epsilon)} U^i$.

2 Parameter setting for Problem 1

For the standard parameter setting in Problem 1, we should get option values when the stock prices are at 90, 100 and 110. For the challenging parameter setting, we should calculate option prices at 97, 98 and 99. According to different input values, different domain intervals are determined by the code. We give the parameter settings that are used in this paper as follows:

SP: standard parameter setting
CP: challenging parameter setting
M: points in time direction
N: points in spatial direction
$S_{\min}$: minimum in interval of S
$S_{\max}$: maximum in interval of S
c: shape parameter
h: minimum number between $\frac{S_{\max}-S_{\min}}{M}$ and $\frac{S_{\max}-S_{\min}}{N}$

1a SP	M	N	c
Level 1	5	5	$2h$
Level 2	9	9	$2h$
Level 3	17	17	$2h$
Level 4	33	33	$2h$
Level 5	65	65	$2h$

Table 1: This setting is for U, Δ, Γ and $\mathcal{V}$ at points [90, 100, 110]. For U and $\mathcal{V}$, $S_{\min} = 36$, $S_{\max} = 165$. For Δ and Γ, $S_{\min} = 56$, $S_{\max} = 145$.

1a CP	M	N	c
Level 1	9	9	$2h$
Level 2	17	17	$2h$
Level 3	33	33	$2h$
Level 4	65	65	$2h$
Level 5	129	129	$2h$

Table 2: This setting is just for U and $\mathcal{V}$ at points [97, 98, 99] and $S_{\min} = 95$, $S_{\max} = 102$.

1a CP	M	N	c
Level 1	3	3	$2h$
Level 2	5	5	$2h$
Level 3	9	9	$2h$
Level 4	17	17	$2h$
Level 5	33	33	$2h$
Level 6	65	65	$2h$
Level 7	129	129	$2h$

Table 3: This setting is just for Δ and Γ at points [97, 98, 99] and $S_{\min} = 95$, $S_{\max} = 101$.

1c SP	M_1	N_1	M_2	N_2	M_3	N_3	c
Level 1	5	7	5	9	5	11	$3h$
Level 2	9	13	9	17	9	21	$3h$
Level 3	17	25	17	33	17	41	$3h$
Level 4	33	49	33	65	33	81	$3h$
Level 5	65	97	65	129	65	161	$3h$

Table 4: This setting is just for U at points [90, 100, 110] and $S_{\min} = 56$, $S_{\max} = 125$. M_1, N_1 are for input value less than 95, M_3, N_3 are for input value greater than 105, M_2, N_2 are for others.

1c CP	M	N	c
Level 1	9	17	$3h$
Level 2	17	33	$3h$
Level 3	33	65	$3h$
Level 4	65	129	$3h$
Level 5	129	257	$3h$

Table 5: This setting is just for U at points [97, 98, 99] and $S_{\min} = 91$, $S_{\max} = 125$.