# Notes on the BENCHOP implementations for RBF–PUM

Victor Shcherbakov (victor.shcherbakov@it.uu.se)

**Abstract**

This text describes the radial basis function partition of unity method (RBF–PUM) and its implementation for the BENCHOP-project.

All problems considered in the BENCHOP-project can be represented as

$$\frac{\partial V}{\partial t} - \mathcal{L}V = 0, \quad x \in \Omega,\, t \in (0, T]\,, \tag{0.1}$$

where $V$ is the value of the option, $s = (s_1, \dots, s_d)$ defines the spot prices of the $d$ underlying assets, $t$ is the backward time, i.e., time to maturity, $T$ is the maturity time of the option, and $\Omega \subset \mathbb{R}^d$.

For example for the European option on one underlying asset under the Black-Scholes model the spatial operator $\mathcal{L}$ takes form

$$\mathcal{L} = \frac{1}{2}\sigma^2 s^2 \frac{\partial^2}{\partial s^2} + rs\frac{\partial}{\partial x} - r, \tag{0.2}$$

where $\sigma$ is the volatility and $r$ is the risk-free interest rate. For other models it has a similar form.

## 1 Radial basis function method

An RBF approximation of function $u$ on $N$ scattered nodes $x_1, \dots, x_N \in \Omega \subset \mathbb{R}^d$ reads as

$$\mathcal{J}_u(x) = \sum_{j=1}^{N} \lambda_j \phi(\|x - x_j\|), \quad x \in \Omega, \tag{1.1}$$

where $\lambda_j$ is an unknown coefficient, $\|\cdot\|$ is the Euclidian norm and $\phi(r)$ is a real-valued radial basis function. In order to determine $\lambda_j$, $j = 1, \ldots, N$, we enforce the interpolation conditions $\mathcal{J}_u(x_j) = u(x_j)$ and as a result we obtain a linear system

$$A\lambda = u, \tag{1.2}$$

where $A_{ij} = \phi(\|x_i - x_j\|)$, $\lambda = [\lambda_1, \ldots, \lambda_N]^T$, $u = [u(x_1), \ldots, u(x_N)]^T$.

If the approximated function is time dependent, we let $\lambda_j$ be time-dependent, such that

$$\mathcal{J}_u(x,t) = \sum_{j=1}^{N} \lambda_j(t)\phi(\|x - x_j\|), \quad x \in \Omega, t \geq 0. \tag{1.3}$$

## 1.1 RBF partition of unity methods

The idea of the radial basis function partition of unity method (RBF–PUM) is to split the domain $\Omega$ into $M$ overlapping partitions and construct a local interpolant in each subdomain. This allows to significantly sparsify the linear system.

We define a partition of unity $\{w_i\}_{i=1}^M$, subordinated to the open cover $\{\Omega_i\}_{i=1}^M$ of $\Omega$, i.e., $\Omega \subseteq \bigcup_{i=1}^M \Omega_i$, such that

$$\sum_{i=1}^{M} w_i(x) = 1, \quad x \in \Omega. \tag{1.4}$$

Now, for each subdomain we construct a local RBF interpolant $\mathcal{J}_u^i$, and then form the global interpolant for the entire domain $\Omega$:

$$\mathcal{J}_u(x) = \sum_{i=1}^{M} w_i(x)\mathcal{J}_u^i(x) = \sum_{i=1}^{M} w_i(x) \sum_{j=1}^{N_i} \lambda_j^i \phi(\|x - x_j^i\|), \quad x \in \Omega. \tag{1.5}$$

The partition of unity functions $w_i$ can be constructed in the following way

$$w_i(x) = \frac{\varphi_i(x)}{\sum_{k=1}^{M} \varphi_k(x)}, \quad i = 1, \ldots, M, \tag{1.6}$$

where $\varphi_i(x)$ is a $C^2$ compactly supported on $\Omega_i$ Wendland function

$$\varphi(r) = \begin{cases} (1-r)^4(4r+1), & \text{if } 0 \leq r \leq 1 \\ 0, & \text{if } r > 1. \end{cases} \tag{1.7}$$

Each $\Omega_i$ is chosen as a circular patch. Therefore, each Wendland function will be scaled as

$$\varphi_i(x) = \varphi\left(\frac{\|x - c_i\|}{r_i}\right), \quad i = 1, \ldots, M, \tag{1.8}$$

where $r_i$ is the radius of the patch $\Omega_i$ and $c_i$ is its center point.

# 2 Time discretisation

For the time discretisation we use a modified version of the backward differential formula of the second order (BDF-2), which requires just one LU-factorisation.

We divide the time interval $[0, T]$ into $N_t$ steps of length $k^n = t^n - t^{n-1}$, $n = 1, \ldots, N_t$. The BDF-2 scheme then has the form

$$\begin{aligned} (E - \beta_0^n L)\, V_I^1 &= V_I^0, && (2.1) \\ (E - \beta_0^n L)\, V_I^n &= \beta_1^n V_I^{n-1} - \beta_2^n V_I^{n-2}, \quad n = 2, \ldots, N_t, && (2.2) \end{aligned}$$

where $V_I^n$ is the solution in the interior, $L$ is the discretised spatial operator, and $E$ is an identity operator and

$$\beta_0^n = k^n \frac{1 + \omega_n}{1 + 2\omega_n}, \quad \beta_1^n = \frac{(1 + \omega_n)^2}{1 + 2\omega_n}, \quad \beta_2^n = \frac{\omega_n^2}{1 + 2\omega_n}, \tag{2.3}$$

where $\omega_n = k^n / k^{n-1}$, $n = 2, \ldots, N_t$.

## 2.1 Boundary treatment

We enforce the following boundary conditions

$$V_B^n = f_B^n, \quad n = 1, \ldots, N_t, \tag{2.4}$$

where $f_B^n$ is the asymptotic solution.

Thus, the linear system becomes

$$\begin{pmatrix} E_I - \beta_0 L_{II} & -\beta_0 L_{IB} \\ 0 & E_B \end{pmatrix} \begin{pmatrix} V_I^n \\ V_B^n \end{pmatrix} = \begin{pmatrix} f_I^n \\ f_B^n \end{pmatrix}, \tag{2.5}$$

where

$$f_I^n = \beta_1^n V_I^{n-1} - \beta_2^n V_I^{n-2}. \tag{2.6}$$

# 3 American options

We solve American option problems by a penalty approach, which allows us to remove the free boundary and solve the problem on a fixed domain. The used penalty function is

$$P = \frac{erK}{V + e - q}, \tag{3.1}$$

where $e$ is the penalty parameter, $r$ is the risk-free interest rate, $K$ is the strike price, $V$ is the option price, and $q = K - s$. This penalty is non-linear and in order to avoid non-linear iterations we treat the penalty explicitly. It will result into the same linear system as (2.5), but

$$f_I^n = \beta_1^n V_I^{n-1} - \beta_2^n V_I^{n-2} - \beta_0^n P(V_I^{n-1}). \tag{3.2}$$

The explicit treatment of the penalty function will put a limit on the time step size

$$\Delta t \leq \frac{e}{rK}. \tag{3.3}$$

# 4 Implementation delails

The type of RBFs used for all experiments is multiquadric $\phi(r) = \sqrt{1 + \varepsilon^2 r^2}$. The discretisation parameters for each test problem can be found in the attached sourse code files. The only thing which is worth to mention here is for some problems we used meshes with points clustered around the strike price. This allowed for an essential increase in the accuracy. The information about used grids and amount of their clustering can be as well found in the code files.

# References

[1] A. Safdari-Vaighani, A. Heryudono and E. Larsson, *A radial basis function partition of unity collocation method for convectiondiffusion equations arising in financial applications*, Journal of Scientific Computing, 2014.

[2] V. Shcherbakov, E. Larsson, *Radial basis function partition of unity methods for pricing vanilla basket options*, Tech. Rep. 2015-001, Division of Scientific Computing, Department of Information Technology, Uppsala University, 2015.