Numerical solution of surface PDEs with Radial Basis Functions

Andriy Sokolov

Institut für Angewandte Mathematik (LS3)
TU Dortmund
andriy.sokolov@math.tu-dortmund.de

TU Dortmund
June 1, 2017

technische universität dortmund

fakultät für mathematik

Outline

1. **Radial Basis Functions (RBFs)**
2. **FD-RBF approximation of differential operators**
   (Finite Difference Radial Basis Functions)
3. **FD-RBF level set method for surface PDEs**
4. **Outlook**

Outline

Outline

Outline

1. **Radial Basis Functions (RBFs)**
2. **FD-RBF approximation of differential operators**
   (Finite Difference Radial Basis Functions)
3. **FD-RBF level set method for surface PDEs**
4. **Outlook**

Problem

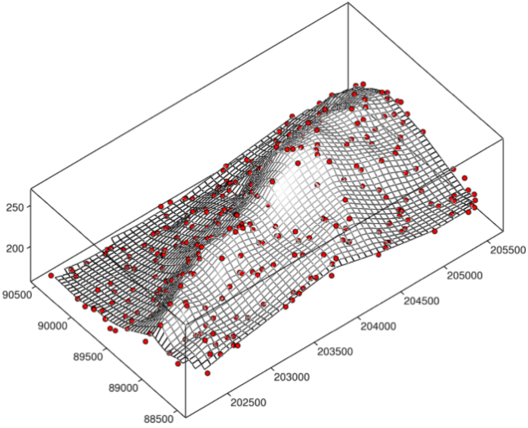


Figure: Scattered data interpolation

Idea 1: polynomial interpolation

Given **points** $a < \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N < b$
and **data values** $f_1, f_2, \ldots, f_N$
construct a polynomial p of degree $N-1$ s.t. $p(\boldsymbol{x}_i) = f_i$.

Idea 1: polynomial interpolation

Given **points** $a < \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N < b$
and **data values** $f_1, f_2, \ldots, f_N$
~~construct a polynomial p of degree $N-1$ s.t. $p(\boldsymbol{x}_i) = f_i$.~~

Idea 1: polynomial interpolation

Given **points** $a < \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N < b$
and **data values** $f_1, f_2, \ldots, f_N$
~~construct a polynomial p of degree $N-1$ s.t. $p(\boldsymbol{x}_i) = f_i$.~~

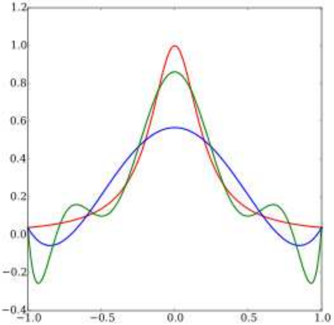


Figure: Runge phenomenon

Idea 2: splines

Cubic splines:

$$\mathcal{S}_3(X) = \{s \in C^2[a,b] \,:\, s_{|_{[x_i,x_{i+1}]}} \in \mathbb{P}_3(\mathbb{R}),\;\; 0 \le i \le N\}.$$

$\dim \mathcal{S}_3(X) = N + 4$, but N interpolation conditions $s(x_i) = f_i$.

Idea 2: splines

Cubic splines:

$$\mathcal{S}_3(X) = \{s \in C^2[a,b] \,:\, s_{|_{[x_i,x_{i+1}]}} \in \mathbb{P}_3(\mathbb{R}),\ \ 0 \le i \le N\}.$$

$\dim \mathcal{S}_3(X) = N + 4$, but N interpolation conditions $s(x_i) = f_i$.

Natural splines:

$$\mathcal{N}_3(X) = \{s \in \mathcal{S}_3(X) \,:\, s_{|_{[a,x_1]}}, s_{|_{[x_N,b]}} \in \mathbb{P}_1(\mathbb{R})\}.$$

Here, **the initial interpolation problem has a unique solution**

Idea 2: splines, 2D case

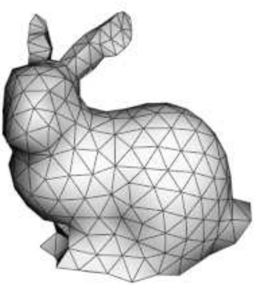


Figure: Stanford Bunny

- In the subdivision of a region by triangles, **the dimension of the spline space is in general unknown**
- Even if great progresses have been made in the 2D setting, **the method is not suited for general dimensions**

Idea 2: splines, 2D case

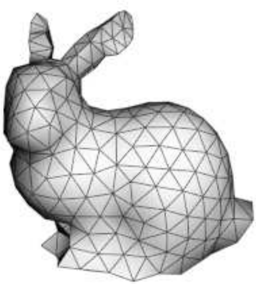


Figure: Stanford Bunny

- In the subdivision of a region by triangles, **the dimension of the spline space is in general unknown**
- Even if great progresses have been made in the 2D setting, **the method is not suited for general dimensions**

Idea 2: splines, 2D case

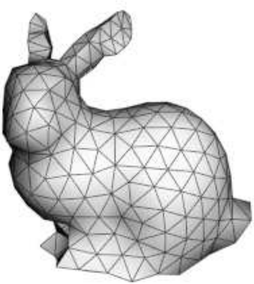


Figure: Stanford Bunny

- In the subdivision of a region by triangles, **the dimension of the spline space is in general unknown**
- Even if great progresses have been made in the 2D setting, **the method is not suited for general dimensions**

Interpolation with RBFs

Idea:

Let us search interpolant as a linear combination of basis functions $\{\varphi_j\}_{j=1}^N$:

$$s(x) = \sum_{j=1}^{N} \lambda_j \varphi_j(x).$$

Interpolation with RBFs

Idea:

Let us search interpolant as a linear combination of basis functions $\{\varphi_j\}_{j=1}^N$:

$$s(x) = \sum_{j=1}^{N} \lambda_j \varphi_j(x).$$

Applying interpolation conditions

$$s(x_i) = f_i, \quad i = 1, \ldots, N$$

we obtain a system of linear equations

$$\underbrace{\begin{pmatrix} \varphi_1(x_1) & \ldots & \varphi_N(x_1) \\ \vdots & \ldots & \vdots \\ \varphi_1(x_N) & \ldots & \varphi_N(x_N) \end{pmatrix}}_{=\mathbf{A}} \begin{pmatrix} \lambda_1 \\ \ldots \\ \lambda_N \end{pmatrix} = \begin{pmatrix} f_1 \\ \ldots \\ f_N \end{pmatrix}.$$

Idea:

Let us search interpolant as a linear combination of basis functions $\{\varphi_j\}_{j=1}^N$:

$$s(x) = \sum_{j=1}^{N} \lambda_j \varphi_j(x).$$

Applying interpolation conditions

$$s(x_i) = f_i, \quad i = 1, \dots, N$$

we obtain a system of linear equations

$$\underbrace{\begin{pmatrix} \varphi_1(x_1) & \dots & \varphi_N(x_1) \\ \vdots & \dots & \vdots \\ \varphi_1(x_N) & \dots & \varphi_N(x_N) \end{pmatrix}}_{=\mathbf{A}} \begin{pmatrix} \lambda_1 \\ \dots \\ \lambda_N \end{pmatrix} = \begin{pmatrix} f_1 \\ \dots \\ f_N \end{pmatrix}.$$

Is this SLE solvable?

The answer: in general **NO**

The answer: in general **NO**

Definition

Let the finite-dimensional linear function space $\Phi \subset \mathcal{C}(\Omega)$, have a basis $\{\phi_1, \phi_2, \dots, \phi_N\}$. Then Φ is a **Haar space** on Ω if

$$\det(\mathbf{A}) \neq 0$$

for any set of distinct points $x_1, \dots, x_N \in \Omega$. Here, $\mathbf{A} = \{\phi_j(x_i)\}_{ij=1,N}$.

Interpolation with RBFs

The answer: in general **NO**

Definition

Let the finite-dimensional linear function space $\Phi \subset \mathcal{C}(\Omega)$, have a basis $\{\phi_1, \phi_2, \ldots, \phi_N\}$. Then Φ is a **Haar space** on Ω if

$$\det(\mathbf{A}) \neq 0$$

for any set of distinct points $x_1, \ldots, x_N \in \Omega$. Here, $\mathbf{A} = \{\phi_j(x_i)\}_{ij=1,N}$.

Theorem (Mairhuber-Curtis)

Let $\Omega \subseteq \mathbb{R}^d$, $d \geq 2$ contains an interior point, then
there exist no Haar space
of continuous functions except for the 1-dimensional case.

Interpolation with RBFs

The answer: in general **NO**

Definition

Let the finite-dimensional linear function space $\Phi \subset \mathcal{C}(\Omega)$, have a basis $\{\phi_1, \phi_2, \ldots, \phi_N\}$. Then Φ is a **Haar space** on Ω if

$$\det(\mathbf{A}) \neq 0$$

for any set of distinct points $x_1, \ldots, x_N \in \Omega$. Here, $\mathbf{A} = \{\phi_j(x_i)\}_{ij=1,N}$.

Theorem (Mairhuber-Curtis)

Let $\Omega \subseteq \mathbb{R}^d$, $d \geq 2$ contains an interior point, then

there exist no Haar space

of continuous functions except for the 1-dimensional case.

Let ϕ_i be dependent on x_i!

Interpolation with RBFs

Approach

- shift the function $\varphi_j(x)$ to the point x_j

$$\varphi_j(\|x - x_j\|)$$

- Choose a radially symmetric function

$$\varphi(\mathbf{r}) = \varphi(\|x\|)$$

Approach

- shift the function $\varphi_j(x)$ to the point x_j

$$\varphi_j(\|x - x_j\|)$$

- Choose a radially symmetric function

$$\varphi(\mathbf{r}) = \varphi(\|x\|)$$

Remark 1: Every natural spline s has the representation

$$s(x) = \sum_{j=1}^{N} a_j \varphi(\|x - x_j\|) + p(x), \;\; x \in \mathbb{R}$$

where $\varphi(r) = r^3, \;\; r \geq 0$ and $p \in \mathbb{P}_1(\mathbb{R})$.

Interpolation with RBFs

Radial Basis Functions (RBFs)

$$s(x) = \sum_{j=1}^{N} \lambda_j \varphi(\|x - x_j\|),$$

Radial Basis Functions (RBFs)

$$s(x) = \sum_{j=1}^{N} \lambda_j \varphi(\|x - x_j\|),$$

where λ_j's are calculated to satisfy $s(x_i) = f_i$:

$$\underbrace{\begin{pmatrix} \varphi(\|x_1 - x_1\|) & \varphi(\|x_1 - x_2\|) & \dots & \varphi(\|x_1 - x_N\|) \\ \varphi(\|x_2 - x_1\|) & \varphi(\|x_2 - x_2\|) & \dots & \varphi(\|x_2 - x_N\|) \\ \vdots & \vdots & \dots & \vdots \\ \varphi(\|x_N - x_1\|) & \varphi(\|x_N - x_2\|) & \dots & \varphi(\|x_N - x_N\|) \end{pmatrix}}_{=\mathbf{A} \text{ distance matrix}} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \dots \\ \lambda_N \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \dots \\ f_N \end{pmatrix}.$$

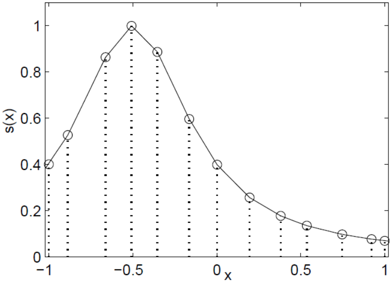
Radial Basis Functions (RBFs)

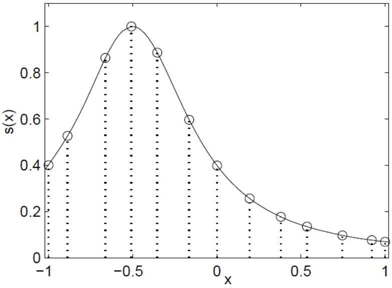
$$s(x) = \sum_{j=1}^{n} \lambda_j \varphi(\|x - x_j\|),$$

- dimension independent,
- sufficiently smooth,
- easy to find (multiple-) derivatives,
- the corresponding SLE is well-posed, i.e. $\det(\mathbf{A}) \neq 0$,
- meshfree,
- other advantages to discuss later.

Linear vs Multiquadratic RBF:

$$s(x) = \sum_{j=1}^{N} \lambda_j \|x - x_j\| \qquad s(x) = \sum_{j=1}^{N} \lambda_j \sqrt{c^2 + \|x - x_j\|^2}$$

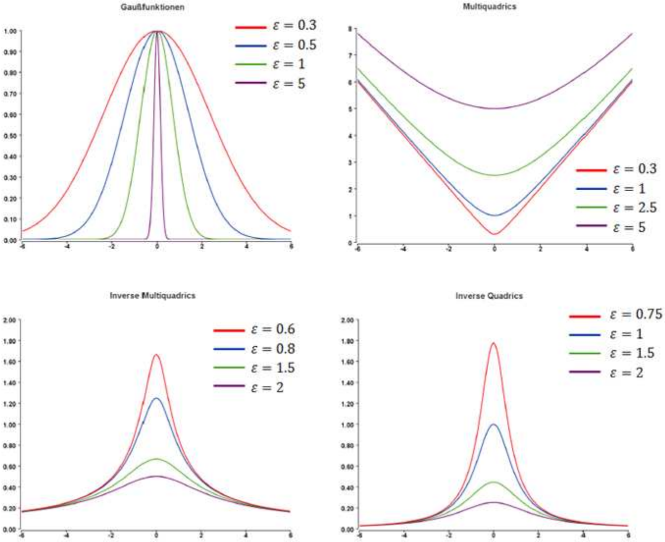




Types of RBFs:

Infinitely smooth RBFs	$\varphi(\mathbf{r}) \ \ (\mathbf{r} \geq 0)$
Gaussian (GA)	$\mathbf{e}^{-(\varepsilon \mathbf{r})^2}$
Inverse quadratic (IQ)	$\dfrac{1}{1+(\varepsilon \mathbf{r})^2}$
Inverse multiquadratic (IMQ)	$\dfrac{1}{\sqrt{1+(\varepsilon \mathbf{r})^2}}$
Multiquadratic (MQ)	$\sqrt{1+(\varepsilon \mathbf{r})^2}$
Piecewise smooth RBFs	
Linear	$\mathbf{r}$
Cubic	$\mathbf{r}^3$
Thin plate spline (TPS)	$\mathbf{r}^2 \ \log \ \mathbf{r}$

Interpolation with RBFs



Solvability of the SLE :

$$\underbrace{\begin{pmatrix} \varphi(\|x_1 - x_1\|) & \varphi(\|x_1 - x_2\|) & \dots & \varphi(\|x_1 - x_N\|) \\ \varphi(\|x_2 - x_1\|) & \varphi(\|x_2 - x_2\|) & \dots & \varphi(\|x_2 - x_N\|) \\ \vdots & \vdots & \dots & \vdots \\ \varphi(\|x_N - x_1\|) & \varphi(\|x_N - x_2\|) & \dots & \varphi(\|x_N - x_N\|) \end{pmatrix}}_{=\mathbf{A}\ \text{ distance matrix}} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \dots \\ \lambda_N \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \dots \\ f_N \end{pmatrix}$$

Solvability of the SLE :

$$\underbrace{\begin{pmatrix} \varphi_\varepsilon(\|x_1 - x_1\|) & \varphi_\varepsilon(\|x_1 - x_2\|) & \dots & \varphi_\varepsilon(\|x_1 - x_N\|) \\ \varphi_\varepsilon(\|x_2 - x_1\|) & \varphi_\varepsilon(\|x_2 - x_2\|) & \dots & \varphi_\varepsilon(\|x_2 - x_N\|) \\ \vdots & \vdots & \dots & \vdots \\ \varphi_\varepsilon(\|x_N - x_1\|) & \varphi_\varepsilon(\|x_N - x_2\|) & \dots & \varphi_\varepsilon(\|x_N - x_N\|) \end{pmatrix}}_{=\mathbf{A}(\varepsilon)\ \text{distance matrix}} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \dots \\ \lambda_N \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \dots \\ f_N \end{pmatrix}$$

Solvability of the SLE :

$$\underbrace{\begin{pmatrix} \varphi_\varepsilon(\|x_1 - x_1\|) & \varphi_\varepsilon(\|x_1 - x_2\|) & \dots & \varphi_\varepsilon(\|x_1 - x_N\|) \\ \varphi_\varepsilon(\|x_2 - x_1\|) & \varphi_\varepsilon(\|x_2 - x_2\|) & \dots & \varphi_\varepsilon(\|x_2 - x_N\|) \\ \vdots & \vdots & \dots & \vdots \\ \varphi_\varepsilon(\|x_N - x_1\|) & \varphi_\varepsilon(\|x_N - x_2\|) & \dots & \varphi_\varepsilon(\|x_N - x_N\|) \end{pmatrix}}_{=\mathbf{A}(\varepsilon) \text{ distance matrix}} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \dots \\ \lambda_N \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \dots \\ f_N \end{pmatrix}$$

Solvability of the SLE :

$$\begin{pmatrix} \varphi_\varepsilon(\|x - x_1\|) \\ \varphi_\varepsilon(\|x - x_2\|) \\ \vdots \\ \vdots \\ \varphi_\varepsilon(\|x - x_N\|) \end{pmatrix}$$

Solvability of the SLE :

$$\begin{pmatrix} \varphi_\varepsilon(\|x-x_1\|) \\ \varphi_\varepsilon(\|x-x_2\|) \\ \vdots \\ \vdots \\ \varphi_\varepsilon(\|x-x_N\|) \end{pmatrix} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & C & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} \varepsilon^0 & & & & \\ & \varepsilon^2 & & & \\ & & \ddots & & \\ & & & \varepsilon^4 & \\ & & & & \ddots \end{pmatrix} \begin{pmatrix} Y_0^0(x) \\ Y_1^{-1}(x) \\ \vdots \\ \vdots \\ \vdots \end{pmatrix},$$

where C is a matrix with entries of size $O(1)$.

Solvability of the SLE :

$$\begin{pmatrix} \varphi_\varepsilon(\|x-x_1\|) \\ \varphi_\varepsilon(\|x-x_2\|) \\ \vdots \\ \vdots \\ \varphi_\varepsilon(\|x-x_N\|) \end{pmatrix} = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & C & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} \varepsilon^0 & & & & \\ & \varepsilon^2 & & & \\ & & \ddots & & \\ & & & \varepsilon^4 & \\ & & & & \ddots \end{pmatrix} \begin{pmatrix} Y_0^0(x) \\ Y_1^{-1}(x) \\ \vdots \\ \vdots \\ \vdots \end{pmatrix},$$

where C is a matrix with entries of size $O(1)$.

Remedy: special QR-decomposition $\Rightarrow$ **work of Markus Verkely**

Error estimation for the RBF interpolation:
(Gaussian and multiquadratic-like functions)

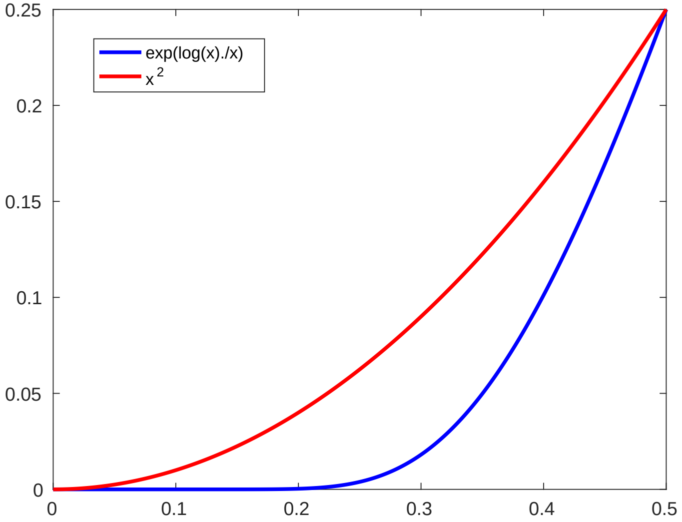
$$\text{if } |f^{(l)}| \le l! M^l \;\Rightarrow\; \|f - s_{f,X}\|_{L_\infty(\Omega)} \le e^{-c/h_{X,\Omega}} |f|_{\mathcal{N}(\Omega)},$$

$$\text{if } |f^{(l)}| \le M^l \;\Rightarrow\; \|f - s_{f,X}\|_{L_\infty(\Omega)} \le e^{c \log h_{X,\Omega}/h_{X,\Omega}} |f|_{\mathcal{N}(\Omega)},$$

where

$$h_{X,\Omega} = \sup_{x \in \Omega} \min_{x_j \in X} \|x - x_j\|_2$$

is the **fill distance**.


0.25
0.2
0.15
0.1
0.05
0
0
0.1
0.2
0.3
0.4
0.5
exp(log(x)./x)
x 2

Outline

Finite Difference Radial Basis Function (FD-RBF)

Finite difference (notation $u(x_i) = u_i$):

$$\begin{aligned}
\text{Forward: } (u_x)_i &= \frac{u_{i+1} - u_i}{h} + R_F(h, \cdot) \\
\text{Backward: } (u_x)_i &= \frac{u_i - u_{i-1}}{h} + R_B(h, \cdot) \\
\text{Central: } (u_x)_i &= \frac{u_{i+1} - u_{i-1}}{2h} + R_C(h, \cdot)
\end{aligned}$$

Finite Difference Radial Basis Function (FD-RBF)

Finite difference (notation $u(x_i) = u_i$):

$$
\begin{aligned}
\text{Forward:}\ (u_x)_i &= \frac{1}{h} u_{i+1} + \frac{-1}{h} u_i + R_F(h, \cdot) \\
\text{Backward:}\ (u_x)_i &= \frac{1}{h} u_i + \frac{-1}{h} u_{i-1} + R_B(h, \cdot) \\
\text{Backward:}\ (u_x)_i &= \frac{1}{2h} u_{i+1} + \frac{-1}{2h} u_{i-1} + R_C(h, \cdot)
\end{aligned}
$$

Finite difference (notation $u(x_i) = u_i$):

$$\begin{aligned}
\text{Forward: } (u_x)_i &= \omega_{i+1} u_{i+1} + \omega_i u_i + R_F(h, \cdot) \\
\text{Backward: } (u_x)_i &= \omega_i u_i + \omega_{i-1} u_{i-1} + R_B(h, \cdot) \\
\text{Central: } (u_x)_i &= \omega_{i+1} u_{i+1} + \omega_{i-1} u_{i-1} + R_C(h, \cdot)
\end{aligned}$$

Finite Difference Radial Basis Function (FD-RBF)

Finite difference (notation $u(x_i) = u_i$):

$$\begin{aligned}
\text{Forward: } (u_x)_i &= \sum_{i\in\Xi_F} \omega_i u_i + R_F(h,\cdot) \\
\text{Backward: } (u_x)_i &= \sum_{i\in\Xi_B} \omega_i u_i + R_B(h,\cdot) \\
\text{Central: } (u_x)_i &= \sum_{i\in\Xi_C} \omega_i u_i + R_C(h,\cdot)
\end{aligned}$$

Finite Difference Radial Basis Function (FD-RBF)

Approximation of the Laplace operator:

$$\Delta u = f, \quad \Omega \subset \mathbb{R}^d$$

Finite Difference Radial Basis Function (FD-RBF)

Approximation of the Laplace operator:

$$\Delta u = f, \quad \Omega \subset \mathbb{R}^d$$

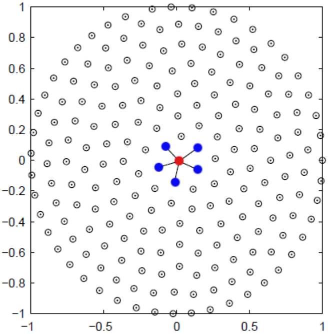
$$u(x) \approx s(x) = \sum_{i=1}^{N} c_i \varphi(\|x - x_i\|)$$

Finite Difference Radial Basis Function (FD-RBF)

Approximation of the Laplace operator:

$$\Delta u = f, \quad \Omega \subset \mathbb{R}^d$$

$$\Delta s(\zeta) \approx \sum_{\xi \in \Xi_\zeta} \omega_\xi u(\xi), \quad \Xi_\zeta = \{\zeta, \xi_1, \xi_2, \ldots, \xi_K\}.$$



Finite Difference Radial Basis Function (FD-RBF)

Approximation of the Laplace operator:

$$\Delta u = f, \quad \Omega \subset \mathbb{R}^d$$

$$\begin{aligned}
\Delta s(\zeta) &= \sum_{j=0}^{K} \omega_j u(\xi_j), \\
\Delta \sum_{i=0}^{N} c_i \varphi(\|\zeta - \xi_i\|) &= \sum_{j=0}^{K} \omega_j u(\xi_j), \\
\sum_{i=0}^{N} c_i \Delta\varphi(\|\zeta - \xi_i\|) &= \sum_{i=0}^{N} c_i \sum_{j=0}^{K} \omega_j \varphi(\|\xi_j - \xi_i\|) \\
\underbrace{\Delta\varphi(\|\zeta - \xi_i\|)}_{rhs} &= \sum_{j=0}^{K} \underbrace{\omega_j}_{\omega} \underbrace{\varphi(\|\xi_j - \xi_i\|)}_{\mathbf{\Phi}} \\
\mathbf{\Phi}\,\omega &= rhs(\Delta)
\end{aligned}$$

Finite Difference Radial Basis Function (FD-RBF)

Approximation of a differential operator:

$$\mathcal{L}u = f, \quad \Omega \subset \mathbb{R}^d$$

$$\begin{aligned}
\Delta s(\zeta) &= \sum_{j=0}^{K} \omega_j u(\xi_j), \\
\mathcal{L} \sum_{i=0}^{N} c_i \varphi(\|\zeta - \xi_i\|) &= \sum_{j=0}^{K} \omega_j u(\xi_j), \\
\sum_{i=0}^{N} c_i \mathcal{L} \varphi(\|\zeta - \xi_i\|) &= \sum_{i=0}^{N} c_i \sum_{j=0}^{K} \omega_j \varphi(\|\xi_j - \xi_i\|) \\
\underbrace{\mathcal{L}\varphi(\|\zeta - \xi_i\|)}_{rhs} &= \sum_{j=0}^{K} \underbrace{\omega_j}_{\omega} \underbrace{\varphi(\|\xi_j - \xi_i\|)}_{\mathbf{\Phi}} \\
\mathbf{\Phi}\,\omega &= rhs(\mathcal{L})
\end{aligned}$$

Derivative(s) of the Gaussian RBF:

$$\varphi(\underbrace{\|\mathbf{x}-\xi\|}_{\mathbf{r}(\mathbf{x})}) = \varphi(\mathbf{r}(\mathbf{x})) = \mathbf{e}^{-\varepsilon^2 \mathbf{r}^2(\mathbf{x})},$$

Derivative(s) of the Gaussian RBF:

$$\varphi(\underbrace{\|\mathbf{x}-\xi\|}_{\mathbf{r}(\mathbf{x})}) = \varphi(\mathbf{r}(\mathbf{x})) = \mathbf{e}^{-\varepsilon^2 \mathbf{r}^2(\mathbf{x})},$$

$$\begin{aligned}
\mathbf{s}(\mathbf{x}) &= \frac{\mathbf{r}^2(\mathbf{x})}{2} \quad \Rightarrow \varphi(\mathbf{s}(\mathbf{x})) = \mathbf{e}^{-2\,\varepsilon^2 \mathbf{s}(\mathbf{x})}, \\
\varphi_{x_i}(\mathbf{s}(\mathbf{x})) &= \underbrace{\varphi_{\mathbf{s}}}_{-2\varepsilon^2\varphi} \quad \underbrace{\mathbf{s}_{x_i}}_{(x_i-\xi_i)} = -2\varepsilon^2\varphi\left[x_i-\xi_i\right], \\
\varphi_{x_i x_i}(\mathbf{s}(\mathbf{x})) &= \varphi_{\mathbf{ss}} \underbrace{\mathbf{s}_{x_i}\mathbf{s}_{x_i}}_{(x_i-\xi_i)^2} + \varphi_{\mathbf{s}} \underbrace{\mathbf{s}_{x_i x_i}}_{1} \\
\varphi_{x_i x_j}(\mathbf{s}(\mathbf{x})) &= \varphi_{\mathbf{ss}}(x_i-\xi_i)(x_j-\xi_j), \quad \text{since } s_{x_i x_j} = 0.
\end{aligned}$$

Consistency + Stability $\Rightarrow$ convergence

Consistency + Stability $\Rightarrow$ convergence

Consistency: "Error Bounds for Kernel-Based Numerical Differentiation" by O. Davydov and R. Schaback, Numer. Math., **132** (2016), 243–269.

Stability: "Error Analysis of Nodal Meshless Methods" by R. Schaback, Numerical Analysis, arXiv:1612.07550 (2016) **... + ???**

Finite Difference Radial Basis Function (FD-RBF)

Poisson equation:

$$\begin{aligned} -\nabla \cdot (\nabla u(\boldsymbol{x})) &= f(\boldsymbol{x}) \quad \text{in } \Omega = [0,1]^2, \\ u_{\text{analyt}} &= \sin(\pi x) \sin(\pi y). \end{aligned}$$

Finite Difference Radial Basis Function (FD-RBF)

Poisson equation:

$$\begin{aligned} -\nabla \cdot (\nabla u(\boldsymbol{x})) &= f(\boldsymbol{x}) \quad \text{in } \Omega = [0,1]^2, \\ u_{\text{analyt}} &= \sin(\pi x) \sin(\pi y). \end{aligned}$$

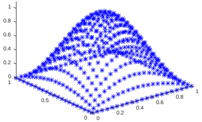


Figure: Numerical solution, h=1/20.

	E_{l_2}	EOC(l_2)	$E_{\max}$	EOC(max)
h=1/5	13019	–	22401	–
h=1/10	330	4,967	726	4,947
h=1/20	89	1,891	188	1,949
h=1/40	23	1,952	47	2,000
h=1/80	5,87	1,970	11,89	1,983

Table: 5-points stencil (including the main node).

Finite Difference Radial Basis Function (FD-RBF)

Anisotropic diffusion:

$$-\nabla\left(\boldsymbol{A}\nabla u(\boldsymbol{x})\right) = f(\boldsymbol{x}) \qquad \text{in} \qquad \Omega = [0,1]^2,$$

where

$$\boldsymbol{A} = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \varepsilon \end{pmatrix} \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix}.$$

Finite Difference Radial Basis Function (FD-RBF)

Anisotropic diffusion:

$$-\nabla\left(\boldsymbol{A}\nabla u(\boldsymbol{x})\right) = f(\boldsymbol{x}) \qquad \text{in} \qquad \Omega = [0,1]^2,$$

where

$$\boldsymbol{A} = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \varepsilon \end{pmatrix} \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix}.$$

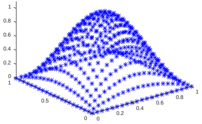


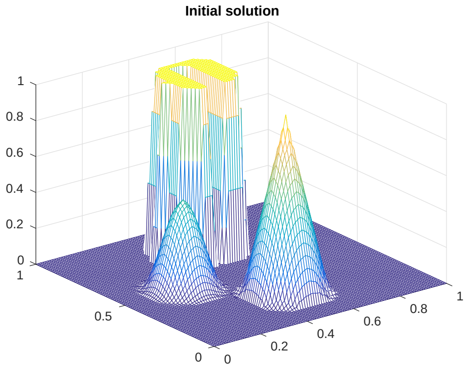
Figure: Numerical solution, h=1/20.

	E_{l_2}	EOC(l_2)	$E_{\max}$	EOC(max)
Stencil=5, $\varepsilon = 10^{-6}$, $\phi = \pi/6$				
h=1/5	58180	–	143297	–
h=1/10	71389	diverges	152334	diverges
h=1/20	76663	diverges	155059	diverges
Stencil=9, $\varepsilon = 10^{-6}$, $\phi = \pi/6$				
h=1/5	4895	–	10783	–
h=1/10	1882	1.379	4025	1.421
h=1/20	560	1.748	1150	1.807
Stencil=25, $\varepsilon = 10^{-6}$, $\phi = \pi/6$				
h=1/5	609	–	1529	–
h=1/10	47	3.695	85	4.168
h=1/20	3	3.969	12	2.824

Table: Numerical experiments.

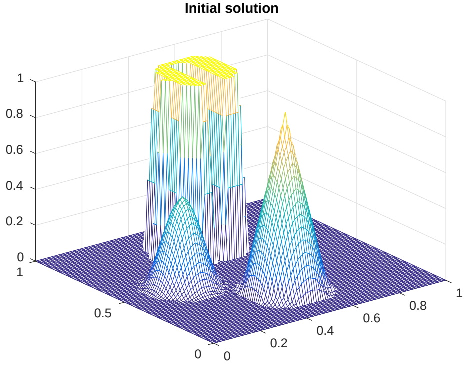
Transport equation (the solid body rotation):

$$u_t + \mathbf{v} \cdot \nabla u = 0 \quad \text{in } \Omega = [0,1]^2.$$



Transport equation (the solid body rotation):

$$u_t + \mathbf{v} \cdot \nabla u - 0.0008\,\Delta\, u = 0 \quad \text{in } \Omega = [0,1]^2.$$



Finite Difference Radial Basis Function (FD-RBF)

Transport equation (the solid body rotation):

$$u_t + \mathbf{v} \cdot \nabla u - 0.0008\, \Delta\, u = 0 \quad \text{in } \Omega = [0,1]^2.$$

Stabilization: "Stabilization of RBF-generated finite difference methods for convective PDEs" by Bengt Fornberg and Erik Lehto, Journal of Computational Physics, 2010.

Finite Difference Radial Basis Function (FD-RBF)

Transport equation (the solid body rotation):

$$u_t + \mathbf{v} \cdot \nabla u - 0.0008\,\Delta\,u = 0 \quad \text{in } \Omega = [0,1]^2.$$

Stabilization: "Stabilization of RBF-generated finite difference methods for convective PDEs" by Bengt Fornberg and Erik Lehto, Journal of Computational Physics, 2010.

... or ???

Reaction-Convection-Diffusion Equation:

$$u_t + \mathbf{v} \cdot \nabla u - \nabla \left(\boldsymbol{A}(\boldsymbol{x}) \nabla u(\boldsymbol{x}) \right) = f \quad \text{in } \Omega \subset \mathbb{R}^2.$$

Finite Difference Radial Basis Function (FD-RBF)

Reaction-Convection-Diffusion Equation:

$$u_t + \mathbf{v} \cdot \nabla u - \nabla \left(\boldsymbol{A}(\boldsymbol{x}) \nabla u(\boldsymbol{x}) \right) = f \quad \text{in } \Omega \subset \mathbb{R}^2.$$

More? $\Rightarrow$ **work of Ufuk Erkul**

Outline

Laplace-Beltrami:

$$\Delta_{\Gamma} u = u(\mathbf{x}) + f \quad \text{on } \Gamma \subset \mathbb{R}^d,$$

where Γ is sufficiently smooth, closed hypersurface.

Laplace-Beltrami:

$$\nabla_\Gamma \cdot (\nabla_\Gamma u(\mathbf{x})) = u(\mathbf{x}) + f \quad \text{on } \Gamma \subset \mathbb{R}^d,$$

where Γ is sufficiently smooth, closed hypersurface.

- **How to treat the surface?**
- **How to treat $\nabla_\Gamma \cdot$?**

Laplace-Beltrami:

$$\nabla_\Gamma \cdot (\nabla_\Gamma u(\mathbf{x})) = u(\mathbf{x}) + f \quad \text{on } \Gamma \subset \mathbb{R}^d,$$

where Γ is sufficiently smooth, closed hypersurface.

- **How to treat the surface?**
- **How to treat $\nabla_\Gamma \cdot$?**

Laplace-Beltrami:

$$\nabla_\Gamma \cdot (\nabla_\Gamma u(\mathbf{x})) = u(\mathbf{x}) + f \quad \text{on } \Gamma \subset \mathbb{R}^d,$$

where Γ is sufficiently smooth, closed hypersurface.

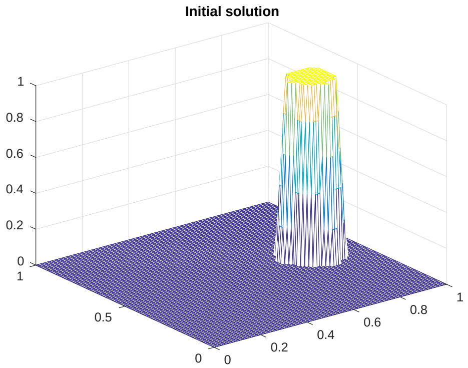
- **How to treat the surface?**
- **How to treat $\nabla_\Gamma \cdot$?**

The phase-field method:

$$u_t - \nabla \cdot (\nabla_\Gamma u(\mathbf{x})) = u(\mathbf{x}) + f \quad \text{on } \Gamma \subset \mathbb{R}^d,$$

The phase-field method:

$$\color{red}{B(\phi)}\, u_t - \nabla \cdot (\color{red}{B(\phi)}\nabla u(\mathbf{x})) = \color{red}{B(\phi)}\,(u(\mathbf{x}) + f) \quad \text{in } \Omega_{\color{red}{\varepsilon}} \subset \mathbb{R}^d,$$



The level set based method:

$$-D\nabla_{\Gamma(t)} \cdot \left(\nabla_{\Gamma(t)} u\right) = f(\boldsymbol{u}), \text{ on } \Gamma(t)$$

The level set based method:

$$-D\nabla_{\Gamma(t)} \cdot \left(\nabla_{\Gamma(t)} u\right) = f(\boldsymbol{u}), \text{ on } \Gamma(t)$$

Introducing the level-set function

$$\phi(\mathbf{x}) = \begin{cases} -dist(\mathbf{x}, \Gamma) & \text{if } \mathbf{x} \text{ is inside } \Gamma \\ 0 & \text{if } \mathbf{x} \in \Gamma \\ dist(\mathbf{x}, \Gamma) & \text{if } \mathbf{x} \text{ is outside } \Gamma \end{cases}$$

The level set based method:

$$-D\nabla_{\Gamma(t)} \cdot \left(\nabla_{\Gamma(t)} u\right) = f(\boldsymbol{u}), \text{ on } \Gamma(t)$$

Introducing the level-set function

$$\phi(\mathbf{x}) = \begin{cases} -dist(\mathbf{x}, \Gamma) & \text{if } \mathbf{x} \text{ is inside } \Gamma \\ 0 & \text{if } \mathbf{x} \in \Gamma \\ dist(\mathbf{x}, \Gamma) & \text{if } \mathbf{x} \text{ is outside } \Gamma \end{cases}$$

we obtain

$$P_\Gamma\, u = \left(I - \frac{\nabla\phi}{\|\nabla\phi\|} \otimes \frac{\nabla\phi}{\|\nabla\phi\|}\right) \nabla u.$$

The level set based method:

$$\nabla_{\Gamma(t)} u = \begin{pmatrix} (\mathbf{e}^x - n^x \mathbf{n}) \cdot \nabla \\ (\mathbf{e}^y - n^y \mathbf{n}) \cdot \nabla \\ (\mathbf{e}^z - n^z \mathbf{n}) \cdot \nabla \end{pmatrix} u = \begin{pmatrix} \mathbf{p}^x \cdot \nabla \\ \mathbf{p}^y \cdot \nabla \\ \mathbf{p}^z \cdot \nabla \end{pmatrix} u = \begin{pmatrix} \mathcal{G}^x \\ \mathcal{G}^y \\ \mathcal{G}^z \end{pmatrix} u$$

The level set based method:

$$\nabla_{\Gamma(t)} u = \begin{pmatrix} (\mathbf{e}^x - n^x \mathbf{n}) \cdot \nabla \\ (\mathbf{e}^y - n^y \mathbf{n}) \cdot \nabla \\ (\mathbf{e}^z - n^z \mathbf{n}) \cdot \nabla \end{pmatrix} u = \begin{pmatrix} \mathbf{p}^x \cdot \nabla \\ \mathbf{p}^y \cdot \nabla \\ \mathbf{p}^z \cdot \nabla \end{pmatrix} u = \begin{pmatrix} \mathcal{G}^x \\ \mathcal{G}^y \\ \mathcal{G}^z \end{pmatrix} u$$

$$\begin{aligned} \Delta_{\Gamma(t)} u &= \nabla_{\Gamma(t)} \cdot \nabla_{\Gamma(t)} u = \nabla_{\Gamma(t)} \cdot \begin{pmatrix} \mathcal{G}^x \\ \mathcal{G}^y \\ \mathcal{G}^z \end{pmatrix} u \\ &= \begin{pmatrix} \mathcal{G}^x & \mathcal{G}^y & \mathcal{G}^z \end{pmatrix} \begin{pmatrix} \mathcal{G}^x \\ \mathcal{G}^y \\ \mathcal{G}^z \end{pmatrix} u = \left(\mathcal{G}^x \mathcal{G}^x + \mathcal{G}^y \mathcal{G}^y + \mathcal{G}^z \mathcal{G}^z \right) u \end{aligned}$$

The level set based method:

$$
\begin{aligned}
(\mathcal{G}^x \, I_\phi u(\mathbf{x}))\,|_{\mathbf{x}=\mathbf{x}_i} &= \sum_{j=1}^{N} c_j \, (\mathcal{G}^x \, \phi(\mathbf{r}_j(\mathbf{x})))\,|_{\mathbf{x}=\mathbf{x}_i} \\
&= \sum_{j=1}^{N} c_j \, [\; ((1 - n_i^x n_i^x)(x_i - x_j) - \\
&\qquad - n_i^y n_i^y (y_i - y_j) \\
&\qquad - n_i^z n_i^z (z_i - z_j)) \, \frac{\phi_{\mathbf{r}}^{'}(\mathbf{r}_j(\mathbf{x}_i))}{\mathbf{r}_j(\mathbf{x}_i)} \;] \, .
\end{aligned}
$$

The level set based method:

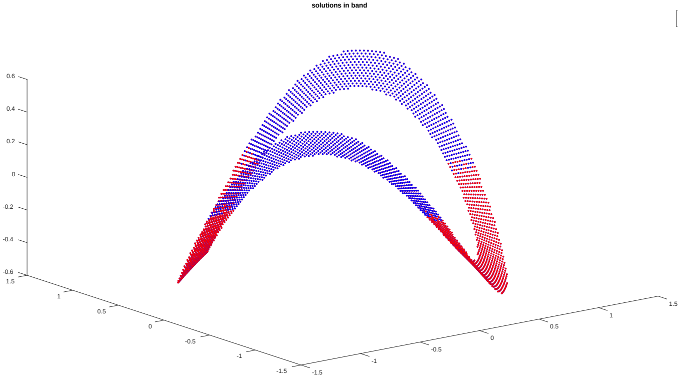
$$u(\mathbf{x}) - \Delta_{\Gamma} u(\mathbf{x}) = f(\mathbf{x}) \qquad \text{on} \qquad \Gamma = \{\boldsymbol{x} : |\boldsymbol{x}| = 1\}.$$

$$u_{\text{analyt}} = \frac{1}{|\mathbf{x}|^5} \frac{26|\mathbf{x}|^2}{|\mathbf{x}|^2 + 25}(x_1^5 - 10x_1^3x_2^2 + 5x_1x_2^4)$$

$$f = \frac{26}{|\mathbf{x}|^5}(x_1^5 - 10x_1^3x_2^2 + 5x_1x_2^4).$$

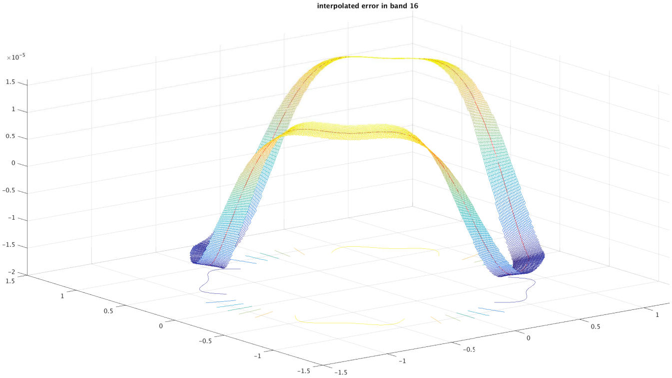
The level set based method:

$$u(\mathbf{x}) - \Delta_{\Gamma} u(\mathbf{x}) = f(\mathbf{x}) \qquad \text{on} \qquad \Gamma = \{\boldsymbol{x} : |\boldsymbol{x}| = 1\}.$$



The level set based method:

$$u(\mathbf{x}) - \Delta_\Gamma u(\mathbf{x}) = f(\mathbf{x}) \qquad \text{on} \qquad \Gamma = \{\boldsymbol{x} : |\boldsymbol{x}| = 1\}.$$



The level set based method:

$$u(\mathbf{x}) - \Delta_{\Gamma} u(\mathbf{x}) = f(\mathbf{x}) \qquad \text{on} \qquad \Gamma = \{\boldsymbol{x} : |\boldsymbol{x}| = 1\}.$$

	E($\varepsilon - \mathrm{band}$)	EOC($\varepsilon - \mathrm{band}$)
h=3/10	0.003001	–
h=3/20	0.001026	1.5484
h=3/40	0.000242	2.0840
h=3/80	0.000051	2.2464
h=3/160	0.000013	1.9720

Table: 9-points stencil (including the main node), ε-band = 0.1.

$$\mathrm{E}(\varepsilon - \mathrm{band}) = \|u_{\mathrm{num}} - u_{\mathrm{analyt}}\|_{l2(\varepsilon-\mathrm{band})} / \sqrt{\#nodes}$$

The level set based method:

$$u(\mathbf{x}) - \Delta_{\Gamma} u(\mathbf{x}) = f(\mathbf{x}) \qquad \text{on} \qquad \Gamma = \{\boldsymbol{x} : |\boldsymbol{x}| = 1\}.$$

	E($\varepsilon - \mathrm{band}$)	EOC($\varepsilon - \mathrm{band}$)
h=3/10	0.003001	–
h=3/20	0.001026	1.5484
h=3/40	0.000242	2.0840
h=3/80	0.000051	2.2464
h=3/160	0.000013	1.9720

Table: 9-points stencil (including the main node), ε-band $= 0.1$.

$$\mathrm{E}(\varepsilon - \mathrm{band}) = \|u_{\mathrm{num}} - u_{\mathrm{analyt}}\|_{l2(\varepsilon-\mathrm{band})} / \sqrt{\#nodes}$$

More? $\Rightarrow$ **work of David Borringo**

The level set based method for $\Gamma(t)$:

$$\partial_t^* u = D\Delta_{\Gamma(t)} u(\boldsymbol{x}, t) + f(\boldsymbol{x}, t) \qquad \text{on} \quad \Gamma = \Gamma(t),$$

where $\Gamma(t) = \{\boldsymbol{x} \,:\, \phi(t, \boldsymbol{x}) = 0\}$ and

$$\phi(t, \boldsymbol{x}) = |\boldsymbol{x}| - 0.75 + \sin(4\,t)(|\boldsymbol{x}| - 0.5)(1 - |\boldsymbol{x}|).$$

The level set based method for $\Gamma(t)$:

$$\underbrace{\overbrace{u_t(\boldsymbol{x},t) + \mathbf{v}\cdot\nabla u}^{\partial_t^{\bullet} u} + u\nabla_\Gamma\cdot\mathbf{v}}_{\partial_t^{*} u} = D\nabla_{\Gamma(t)}\cdot\big(\nabla_{\Gamma(t)}u(\boldsymbol{x},t)\big) + f(\boldsymbol{x},t) \text{ on } \Gamma = \Gamma(t),$$

where $\mathbf{v} = V\boldsymbol{n} + \boldsymbol{v}_S$ and

$$V = \mathbf{v}\cdot\boldsymbol{n} = -\frac{\phi_t}{|\nabla\phi|}.$$

The level set based method for $\Gamma(t)$:

$$\underbrace{\overbrace{u_t(\boldsymbol{x},t)+\mathbf{v}\cdot\nabla u}^{\partial_t^{\bullet}u}+u\nabla_\Gamma\cdot\mathbf{v}}_{\partial_t^{*}u}=D\nabla_{\Gamma(t)}\cdot\big(\nabla_{\Gamma(t)}u(\boldsymbol{x},t)\big)+f(\boldsymbol{x},t)\ \text{ on }\ \Gamma=\Gamma(t),$$

where $\mathbf{v}=V\boldsymbol{n}+\boldsymbol{v}_S$ and

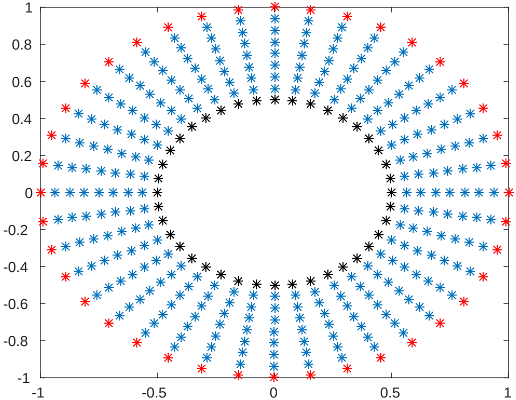
$$V=\mathbf{v}\cdot\boldsymbol{n}=-\frac{\phi_t}{|\nabla\phi|}.$$

The analytical solution is chosen to be

$$u(\boldsymbol{x},t)=e^{-t/|\boldsymbol{x}|^2}\frac{x_1}{|\boldsymbol{x}|}.$$

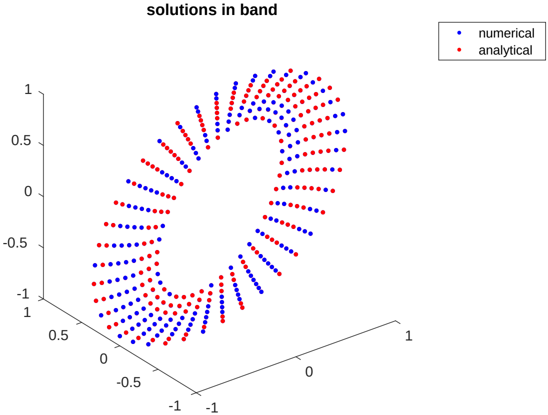
The level set based method for $\Gamma(t)$:

$$\partial_t^* u = D\Delta_{\Gamma(t)} u(\boldsymbol{x}, t) + f(\boldsymbol{x}, t) \qquad \text{on} \quad \Gamma = \Gamma(t).$$



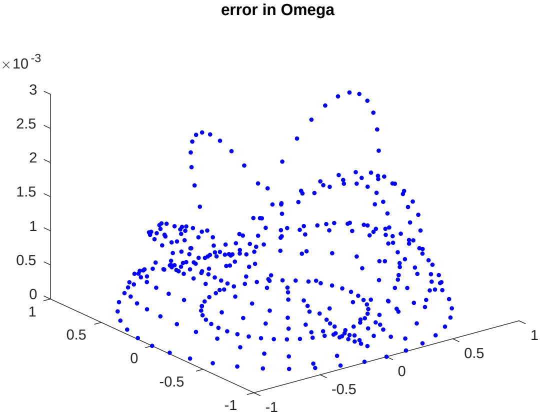
The level set based method for $\Gamma(t)$:

$$\partial_t^* u = D\Delta_{\Gamma(t)} u(\boldsymbol{x}, t) + f(\boldsymbol{x}, t) \qquad \text{on} \quad \Gamma = \Gamma(t).$$



The level set based method for $\Gamma(t)$:

$$\partial_t^* u = D\Delta_{\Gamma(t)} u(\boldsymbol{x}, t) + f(\boldsymbol{x}, t) \qquad \text{on} \quad \Gamma = \Gamma(t).$$



The level set based method for $\Gamma(t)$:

$$\partial_t^* u = D\Delta_{\Gamma(t)} u(\boldsymbol{x}, t) + f(\boldsymbol{x}, t) \qquad \text{on} \quad \Gamma = \Gamma(t).$$

	E($\varepsilon - \mathrm{band}$)	EOC($\varepsilon - \mathrm{band}$)
N=20	0.010179	–
N=40	0.003626	1.4891
N=80	0.002065	0.8122
N=160	0.001584	0.3875

Table: 9-points stencil (including the main node). ε-band = 0.1

The level set based method for $\Gamma(t)$:

$$\partial_t^* u = D\Delta_{\Gamma(t)} u(\boldsymbol{x}, t) + f(\boldsymbol{x}, t) \qquad \text{on} \quad \Gamma = \Gamma(t).$$

	E($\varepsilon - \mathrm{band}$)	EOC($\varepsilon - \mathrm{band}$)
N=20	0.010179	–
N=40	0.003626	1.4891
N=80	0.002065	0.8122
N=160	0.001584	0.3875

Table: 9-points stencil (including the main node). ε-band $= 0.1$

More? $\Rightarrow$ **work in progress...**

Conclusion

- **(FD)RBF is a new rapidly developing approach**
- **It it possible to use it for real-world applications**
- **Mesh adaptivity**

Conclusion

- **(FD)RBF is a new rapidly developing approach**
- **It it possible to use it for real-world applications**
- **Mesh adaptivity**

Conclusion

- (FD)RBF is a new rapidly developing approach
- It it possible to use it for real-world applications
- Mesh adaptivity

- **(FD)RBF is a new rapidly developing approach**
- **It it possible to use it for real-world applications**
- **Mesh adaptivity**
- **Lots of new and unexplored fields**

Conclusion

- **(FD)RBF is a new rapidly developing approach**
- **It it possible to use it for real-world applications**
- **Mesh adaptivity**
- **Lots of new and unexplored fields**

Acknowledgements

- Prof. Dr. Oleg Davydov, University of Giessen
- Prof. Dr. Dmitri Kuzmin, TU Dortmund
- Prof. Dr. Stefan Turek, TU Dortmund

Thank you very much
for your attention!