Stencil selection methods for RBF-FD approximations

Anton Artemov

Division of Scientific Computing, Uppsala University
anton.artemov@it.uu.se

25.10.2017
About RBF generated finite differences

Suppose we want to approximate

$$Du(x_c) \approx \sum_{k=1}^{n} w_k u(x_k),$$  \hspace{1cm} (1)

where $D$ is a differential operator:

$$Du = \sum_{\alpha \in \mathbb{Z}^d, |\alpha| \leq k} c_{\alpha} \partial^{\alpha} u, \hspace{0.5cm} \partial^{\alpha} = \frac{\partial |\alpha|}{\partial x_1^{\alpha_1} \ldots \partial x_d^{\alpha_d}}$$  \hspace{1cm} (2)

$$|\alpha| = \alpha_1 + \ldots + \alpha_d.$$
About RBF generated finite differences

Weights $w_k$ are found by asking exactness for RBF-interpolants

$$
\begin{bmatrix}
\phi_1(x_1) & \phi_1(x_2) & \ldots & \phi_1(x_n) \\
\phi_2(x_1) & \phi_2(x_2) & \ldots & \phi_2(x_n) \\
\vdots & \vdots & \ddots & \vdots \\
\phi_n(x_1) & \phi_n(x_2) & \ldots & \phi_n(x_n)
\end{bmatrix}
\begin{bmatrix}
w_1 \\
w_2 \\
\vdots \\
w_n
\end{bmatrix} =
\begin{bmatrix}
D\phi_1(x_c) \\
D\phi_2(x_c) \\
\vdots \\
D\phi_n(x_c)
\end{bmatrix}
$$

(3)

- Once weights are found, a row of the global matrix $A$ is filled in.
- Note that $x_c$ might not belong to stencil. We assume it does.
Weights $w_k$ are found by asking exactness for RBF-interpolants

$$
\begin{bmatrix}
\phi_1(x_1) & \phi_1(x_2) & \ldots & \phi_1(x_n) \\
\phi_2(x_1) & \phi_2(x_2) & \ldots & \phi_2(x_n) \\
\vdots & \vdots & \ddots & \vdots \\
\phi_n(x_1) & \phi_n(x_2) & \ldots & \phi_n(x_n)
\end{bmatrix}
\begin{bmatrix}
w_1 \\
w_2 \\
\vdots \\
w_n
\end{bmatrix}
= 
\begin{bmatrix}
D\phi_1(x_c) \\
D\phi_2(x_c) \\
\vdots \\
D\phi_n(x_c)
\end{bmatrix}
$$

(3)

- Once weights are found, a row of the global matrix $A$ is filled in.
- Note that $x_c$ might not belong to stencil. We assume it does.

How does one pick the points for the stencil?
Point selection criteria - the simplest one

- Assume one has to find $K$ points for the stencil out of a set of $N$ points, $N \ll N_{total}$.
- The most trivial way is to pick $K - 1$ closest points to a given one.
Point selection criteria - the expensive one

Idea: consider all possible stencils containing $x_c$ and compare them with respect to "quality measurement"

$$M = \left( \sum_{k=2}^{K} \frac{|w_k|}{\sqrt{K}} \right)^{-1}. \quad (4)$$

Note: this method is extremely expensive, since one has to solve

$$\binom{N - 1}{K - 1}$$

linear systems and pick the "best" stencil.
Point selection criteria - the expensive one

Point distribution, $K = 9$, $N = 13$
Point selection criteria - the least squares I


▶ Assume $X = \{x_1, \ldots, x_N\} \in \mathbb{R}^d$ is a unisolvent set for $\Pi^d_q$, $z$ is an evaluation point.

▶ Then the weighed least squares polynomial $L^\theta_{X,q}$, $\theta = [\theta_1, \ldots, \theta_N]^T$, $\theta_j > 0$ is uniquely defined by the condition:

$$\|(L^\theta_{X,q}u - u)|_X\|_{2,\theta} = \min\{\|p - u\|_{2,\theta} : p \in \Pi^d_q\}, \quad (5)$$

where

$$\|v\|_{2,\theta} = \sqrt{\sum_{j=1}^{N} \theta_j v_j^2}.$$
Point selection criteria - the least squares I

- Since $L_{\Theta_X,q}^\theta u$ depends on $u|_X$ linearly, application of $D$ to $L_{\Theta_X,q}^\theta u$ gives

$$Du(z) \approx DL_{\Theta_X,q}^\theta u = \sum_{j=1}^{N} w_{j}^{2,\theta} u(x_j)$$

of exactness of order $q$.

- Authors suggested $\theta_j = \|x_j - z\|^{-2\mu}_2$, $j = 1, \ldots, N$, $\mu > 0$ and then $w^{2,\mu}$ computed with these weights is minimal w.r.t.

$$\|w\|_{2,\mu} := \sqrt{\sum_{j=1}^{N} w_j^2 \|x_j - z\|_2^{2\mu}}$$

where $\mu$ is the consistency order. Preferable choice is $\mu = q$. 

Point selection criteria - the least squares I

You may wonder why one has to deal with all this stuff?
Point selection criteria - the least squares I

You may wonder why one has to deal with all this stuff?

- Error bounds of differentiation formulas are expressed in terms of growth functions:

\[ \rho_{q,D}(z, X, \| \cdot \|) := \sup \{ Dp(z) : p \in \Pi_q^d, \| p \|_{X}^* \leq 1 \}, \]

where \( \| \cdot \|^* \) is dual seminorm.

- Error bound:

\[
\left| Du(z) - \sum_{j=1}^{N} w_j^{2,\mu} u(x_j) \right| \leq \sqrt{N} \rho_{q,D}(z, X, 2, \mu) | u |_{\infty, \mu, \Omega}
\]

- With weights \( \theta_j = \| x_j - z \|_2^{-2\mu} \) we obtain

\[
\rho_{q,D}(z, X, 2, \mu) := \rho_{q,D}(z, X, \| \cdot \|_{2,\mu})
\]

and

\[
\rho_{q,D}(z, X, 2, \mu) = \| w^{2,\mu} \|_{2,\mu}
\]
Point selection criteria - the least squares I

How to compute those weights?

1. Remove $z = x_c$ from $X$ set.

2. Rescale set $X$ into unit disc:
   $$ Y := h_{z,X}^{-1}(X - z), \ h_{z,X} = \max_{x \in X} \|z - x\|_2. $$

3. Obtain vector $v$ for $D_z L_{Y,q}^\theta u(0) = \sum_{j=1}^N v_j f(y_j), \ \theta_j = \|y_j\|_2^{-2\mu}$

4. Scale back: $w = h_{z,X}^k v$, $k$ is the degree of $D$.

One can show that

$$ v^T = [\alpha! c_\alpha(z)]^T (\delta W)^+ \delta, $$

where $W = [y_j^\alpha]_{j,\alpha}, \ \delta = \text{diag}(\sqrt{\theta_1}, \ldots, \sqrt{\theta_N}), \ A^+$ is pseudoinverse of $A$. 

Point selection criteria - the least squares I

How to compute those weights?

1. Remove \( z = x_c \) from \( X \) set.
2. Rescale set \( X \) into unit disc:
   \[
   Y := h_{z,X}^{-1}(X - z), \quad h_{z,X} = \max_{x \in X} \| z - x \|_2.
   \]
3. Obtain vector \( v \) for \( D_z L_{Y,q}^\theta u(0) = \sum_{j=1}^N v_j f(y_j), \quad \theta_j = \| y_j \|_2^{-2\mu} \)
4. Scale back: \( w = h_{z,X}^k v \), \( k \) is the degree of \( D \).

One can show that

\[

v^T = [\alpha! c_\alpha(z)]_{\alpha}^T (\delta W)^+ \delta,
\]

where \( W = [y_j^\alpha]_{j,\alpha}, \quad \delta = diag(\sqrt{\theta_1}, \ldots, \sqrt{\theta_N}), \quad A^+ \) is pseudoinverse of \( A \).

Finally, take \( K - 1 \) points with highest weights and add \( x_c \).
Point selection criteria - the least squares

$$-\Delta u = f, \ q = 2, \ \mu = 2$$
Point selection criteria - the least squares II

Another idea proposed by E. Larsson. Utilizes the same exactness approach:

\[ \sum_{j=1}^{N} w_j p(x_j) = Dp(x_c), \quad \forall p \in \Pi^d_q. \]

It leads to an underdetermined system

\[ Aw = c, \]

which can be solved in least squares sense using QR decomposition:

\[ QR = A^T \implies R^T (Q^T w) = c, \]

\[ y = Q^T w = R^{-T} c \implies w = Qy. \]

Or with "normal" equations:

\[ w = A^T (AA^T)^{-1} c. \]
Point selection criteria - the least squares II

**Drawback:** If $x_c$ is in the set, then it gets the lowest weight $\Rightarrow$ excluded from the stencil.

Point distribution, $K = 9$, $N = 18$
Point selection criteria - the least squares II

**Drawback:** If $x_c$ is in the set, then it gets the lowest weight $\Rightarrow$ excluded from the stencil.

**Remedy:** compute weights without $x_c$, add it afterwards.
Point selection criteria - the least squares II

**Drawback:** with global polynomials the information is maximized by taking distant points, while the correlation of the data is the highest with the near data points.

\[ W_n, n = \text{diag}(w_1, \ldots, w_n) = W^{-1} A^T \left( AW^{-1} A^T \right)^{-1} c. \]
Point selection criteria - the least squares II

**Drawback:** with global polynomials the information is maximized by taking distant points, while the correlation of the data is the highest with the near data points.

**Remedy:** introduce weights, mark near points as more important.

\[
[W]_{n,n} = \text{diag}(w_1, \ldots, w_n) \implies w = W^{-1} A^T (AW^{-1} A^T)^{-1} c.
\]
Point selection criteria - the least squares II

Compare:

Figure: LS without weights (left), LS with weights \( w_j = e^{- (20 \| x_c - x_j \|_2)^2} \)
All four methods compared

Point distribution, $K=9$, $N=13$

Closest points (top left), best quality (top right), LS-I (bottom left), LS-II (bottom right)
All four methods compared

Test problem:

\[ -\Delta u = f, \Omega = [0, 1]^2, u|_{\partial \Omega} = 1 + x^2 + 2y^2, f = -6 \]

Solved with Gaussian RBF with \( \varepsilon h_{\min} = 0.00025 \) on "ugly" set of nodes, \( N_{\text{nodes}} = 140 \):
All four methods compared

Use intentionally small number of points in stencils, namely 6. Methods were to pick them out of 12.

<table>
<thead>
<tr>
<th></th>
<th>Error norm</th>
<th>Average quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Closest points</td>
<td>1.0324</td>
<td>0.0026</td>
</tr>
<tr>
<td>Best quality</td>
<td>1.0837e+03</td>
<td>0.0156</td>
</tr>
<tr>
<td>LS-I</td>
<td>0.0117</td>
<td>0.0046</td>
</tr>
<tr>
<td>LS-II</td>
<td>6.5315e-04</td>
<td>0.0064</td>
</tr>
</tbody>
</table>
All four methods compared

If set is refined with $N_{\text{nodes}} = 260$, the same distribution:

<table>
<thead>
<tr>
<th>Method</th>
<th>Error norm</th>
<th>Average quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Closest points</td>
<td>7.6923</td>
<td>0.0012</td>
</tr>
<tr>
<td>Best quality</td>
<td>5.4119</td>
<td>0.0083</td>
</tr>
<tr>
<td>LS-I</td>
<td>0.2199</td>
<td>0.0027</td>
</tr>
<tr>
<td>LS-II</td>
<td>2.8012e-04</td>
<td>0.0049</td>
</tr>
</tbody>
</table>
Conclusions

It makes sense to consider presented methods provided that the distribution of points does not cover the domain “evenly” and/or a small stencil size is preferred.
Conclusions

It makes sense to consider presented methods provided that the distribution of points does not cover the domain “evenly” and/or a small stencil size is preferred.

- Closest neighbours approach is not the best, although obvious.
- Best quality approach really does compute stencils with the best average quality, but the error is huge.
- LS methods work quite well, but LS-II requires the proper choice of weights.
The End

Thank you for your attention!
Any Questions?