B-Spline Sparse Grids for Eddy-Current Testing Inverse Problems

Sándor Bilicz and József Pávó

Department of Broadband Infocommunications and Electromagnetic Theory
Budapest University of Technology and Economics

ENDE 2017, Saclay, France
Framework:
Surrogate modeling in eddy-current nondestructive testing (EC-NdT)

- parametric defect model $\mathbf{x} = [x_1, \ldots, x_N]$

- impedance signal $\mathbf{y} = [\Delta Z_1, \ldots, \Delta Z_M]$

- heavy numerical simulations $\mathbf{x} \rightarrow \mathbf{y} = f(\mathbf{x})$
  /integral equations, finite elements, .../

- data-fit surrogate model: $\mathbf{y} \approx \hat{f}_n(\mathbf{x})$ based on $\mathbb{D}_n = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$

- generation of $\mathbb{D}$: “off-line”

- use of $\hat{f}_n(\mathbf{x})$ “on-line”

- a very efficient method: interpolation on sparse grids
Sparse grids in a nutshell (1/3): ingredients & hierarchical basis

Ingredients of sparse grid interpolation:

1. hierarchical linear interpolant in 1-dimension
2. expansion to N-dimension by tensor product
3. truncation of the basis

Illustration in 1-dimension:

\[ f(x) \approx \tilde{f}(x) = \sum_{\ell=0}^{d} \sum_{i=1}^{m_{\ell}} c_{i}^{(\ell)} \psi_{i}^{(\ell)}(x) \]

Terminology: \( \ell \): level, \( d \): depth, \( c_{i}^{(\ell)} \): coefficients (“surplus”) and \( \psi_{i}^{(\ell)}(x) \): basis functions (here: linear “hat” → B-spline later on)

Important: number of basis functions = number of function calls (samples)
Sparse grids in a nutshell (2/3): extension to N-dimension

- all 1-dim basis fun. @level $\ell$:
  \[ \Psi_{\ell}(x) = \left\{ \psi_{i}^{(\ell)}(x) \mid i = 1, 2, \ldots, m_{\ell} \right\} \]

- sparse N-dim basis @level $\ell$:
  \[ \Phi_{\ell}(x) = \left\{ \Psi_{\ell_{1}}(x_{1}) \otimes \Psi_{\ell_{2}}(x_{2}) \otimes \cdots \otimes \Psi_{\ell_{N}}(x_{N}) \mid \sum_{i=1}^{N} \ell_{i} = \ell \right\} \]

- interpolation in N-dim:
  \[ \hat{f}(x) = \sum_{\ell=0}^{d} \sum_{i=1}^{m^{(N)}_{\ell}} c_{i}^{(\ell)} \phi_{i}^{(\ell)}(x) \]
Sparse grids in a nutshell (3/3)

- number of nodes: \( n \)
- interpolation error: \( \varepsilon = |\hat{y} - y| \)
- nodes per dimension: \( K = 2^d + 1 \), \( d \): depth

For smooth function \( \left| \frac{\partial^2 f}{(\partial x_i \partial x_j)} \right| \leq B < \infty \) using linear basis functions:

<table>
<thead>
<tr>
<th></th>
<th>Full grid</th>
<th>Sparse grid</th>
<th>Gain</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n )</td>
<td>( K^N )</td>
<td>( \mathcal{O}\left{K (\log K)^{N-1}\right} )</td>
<td>( \mathcal{O}\left{\frac{K^{N-1}}{(\log K)^{N-1}}\right} )</td>
<td>( \mathcal{O}\left{(\log K)^{N-1}\right} )</td>
</tr>
<tr>
<td>( \varepsilon )</td>
<td>( \mathcal{O}\left{\frac{1}{K^2}\right} )</td>
<td>( \mathcal{O}\left{\frac{(\log K)^{N-1}}{K^2}\right} )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ \text{Gain} \times \text{Loss} = K^{N-1} \gg 1 \]

Illustration for the number of nodes \( n \):

<table>
<thead>
<tr>
<th>depth (( d ))</th>
<th>( N = 2 )</th>
<th>( N = 9 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d )</td>
<td>sparse</td>
<td>full</td>
</tr>
<tr>
<td>2</td>
<td>13</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>65</td>
<td>289</td>
</tr>
<tr>
<td>6</td>
<td>321</td>
<td>4225</td>
</tr>
<tr>
<td>8</td>
<td>1537</td>
<td>( 6.6 \times 10^4 )</td>
</tr>
</tbody>
</table>
Past & present of sparse grids in EC-NdT

- 1963 Smolyak: tensor product and truncation, applied for quadrature [1]
- 1980s hierarchical basis
- 1990s solution of PDEs by sparse grids

- 2015 solution of PDE of 24 variables (macro-economy model) [2]
- 2015... application as a surrogate model in electromagnetics [3], [4] in cooperation with:
  - Laboratoire des Signaux et Systèmes (L2S), Gif-sur-Yvette, France
  - Laboratoire Génie électrique et électronique de Paris (GeePs), Gif-sur-Yvette, France
  - CEA-LIST, Saclay, France

- Today: B-spline basis functions [5] ⇒ smooth interpolant
  - gradient-based optimization (inversion), sensitivity analysis
  - using the gradient (if available) to construct $\hat{f}(x)$ (quasi-interpolation?)

B-splines as basis functions

**Cardinal B-spline (of order** \( p = 3 \):**

\[
b(x) = \begin{cases} 
\frac{1}{4}x^3 & 0 \leq x < 1 \\
-\frac{3}{4}x^3 + 3x^2 - 3x + 1 & 1 \leq x < 2 \\
\frac{3}{4}x^3 - 6x^2 + 15x - 11 & 2 \leq x < 3 \\
-\frac{1}{4}x^3 + 3x^2 - 12x + 16 & 3 \leq x < 4 
\end{cases}
\]

**Hierarchical B-spline (of order** \( p = 3 \):**

Transformation of \( x \rightarrow \) all basis functions at level \( \ell (\ell > 0) \):

\[
\psi_i^{(\ell)}(x) = b\left(\frac{x}{h_{\ell}} + 2 - i\right), \quad h_{\ell} = 2^{-\ell}, \quad i = 1, 3, 5, \ldots 2^\ell - 1
\]

for \( \ell = 0, 1, 2 & 3 \):
Comparison of the linear and the B-spline basis

Linear basis functions:

Pros & cons:
- not smooth ✗
- coefficients: surpluses, local error estimators ✓
- easier to implement ✓

B-spline basis functions:

Pros & cons:
- smoothness, gradient ✓
- coefficients: lin. sys. of eq. ✗
- more complicated implementation ✗
Illustration by the 2-dim Branin test function
Inversion in EC-NdT via the minimization of a misfit function

- measured data \( \hat{y} \) and simulated data \( y = f(x) \)
  - e.g., impedance variation at \( M \) coil positions (line/surface scan)
  - let \( y \) be a real row vector (\( y := [\text{Re}(y), \text{Im}(y)] \))

- quadratic misfit: \( u(x) = (\hat{y} - f(x))^2 = [\hat{y} - f(x)][\hat{y} - f(x)]^T \)

- gradient: \( \nabla u(x) = 2[\hat{y} - f(x)][\nabla f(x)]^T \), \( \nabla f(x) = \left( \frac{\partial f_i}{\partial x_j} \right)_{ij} \)

- \( f \) is approximated by \( \hat{f} \) based on the surrogate model and \( \nabla \hat{f} \) is available in closed form

- approximate gradient: \( \nabla \hat{u}(x) = 2[\hat{y} - \hat{f}(x)][\nabla \hat{f}(x)]^T \)

- optimization by a gradient method, e.g., quasi-Newton (Matlab: `fminunc`)

- constraints on \( x \): \( 0 \leq x_i \leq 1 \rightarrow x_i = (\arctan \xi_i)/\pi + 1/2, \ -\infty \leq \xi_i \leq \infty \)
Example: the WFNDEC 2008 benchmark problem with 4 parameters

- non-magnetic plate
  \[ t = 1.55 \text{ mm}, \sigma = 1 \text{ MS/m} \]
- absolute mode coil
  \[ f = 150 \text{ kHz}, \text{ lift-off } h = 0.303 \text{ mm} \]
- surface scan:
  \[ x_c = \{-15 : 1 : 15\}\text{mm} \]
  \[ y_c = \{-5 : 1 : 5\}\text{mm} \]
  \[ \rightarrow M = 341 \text{ complex samples} \]
- narrow crack of rectangular shape

<table>
<thead>
<tr>
<th>parameter</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>length</td>
<td>( a ) [4, 22]mm</td>
</tr>
<tr>
<td>width</td>
<td>( b ) [0.001, 0.3]mm</td>
</tr>
<tr>
<td>depth</td>
<td>( d ) [10, 100]% (of ( t ))</td>
</tr>
<tr>
<td>ligament</td>
<td>( l ) [0, ( t - d )]</td>
</tr>
</tbody>
</table>
Example: test of the interpolation accuracy

- 4000 random test samples $y_i = y(x_i)$, length of $y = 2 \times 341$
  (341 complex impedance measurements)

- $\text{rms}$ interpolation error at the $i$th test point:
  \[ \varepsilon_i = \left( \frac{1}{M} |y_i - \hat{f}(x_i)|^2 \right)^{1/2} \]

- linear and spline basis functions are compared

\[ \text{MAX}(\varepsilon_i) \quad \text{RMS}(\varepsilon_i) \quad [\Omega] \]

No. of nodes by levels in dim=4:

<table>
<thead>
<tr>
<th>$\ell$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_\ell^{(4)}$</td>
<td>1</td>
<td>8</td>
<td>32</td>
<td>96</td>
<td>264</td>
<td>704</td>
<td>1824</td>
</tr>
</tbody>
</table>
Results of WFNDEC 4-parameter inversion

Comparison using a SG surrogate model (dim=4, depth=6, sample no.=2929)

- the proposed gradient method based on $\nabla \tilde{u}(x)$, quasi-Newton algorithm
- a grad-free direct optimization, simplex algorithm

Example 1:

<table>
<thead>
<tr>
<th>true &amp; found parameters [mm]</th>
<th>$a$</th>
<th>$b$</th>
<th>$d$</th>
<th>$l$</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>16.04</td>
<td>0.290</td>
<td>0.258</td>
<td>0.413</td>
</tr>
<tr>
<td>grad</td>
<td>16.30</td>
<td>0.012</td>
<td>0.474</td>
<td>0.324</td>
</tr>
<tr>
<td>direct</td>
<td>20.42</td>
<td>0.118</td>
<td>0.348</td>
<td>0.389</td>
</tr>
</tbody>
</table>

Example 2:

<table>
<thead>
<tr>
<th>true &amp; found parameters [mm]</th>
<th>$a$</th>
<th>$b$</th>
<th>$d$</th>
<th>$l$</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>15.10</td>
<td>0.157</td>
<td>0.525</td>
<td>0.942</td>
</tr>
<tr>
<td>grad</td>
<td>14.96</td>
<td>0.188</td>
<td>0.507</td>
<td>0.957</td>
</tr>
<tr>
<td>direct</td>
<td>16.20</td>
<td>0.137</td>
<td>0.478</td>
<td>0.850</td>
</tr>
</tbody>
</table>

- faster convergence & better (local) minimum found by the “grad” method
- a multi-start method might improve both scheme
Conclusion, perspectives

Findings:

- sparse grids: efficient in surrogate modeling in EC-NdT (and maybe in other contexts as well)
- curse-of-dimensionality is overcome to some extent
- linear interpolation might be more accurate
- spline interpolation yields a smooth surface → gradient-based optimization

Left to do / in progress:

- further applications in sensitivity analysis and inversion
- **adaptive** sparse grid generation with spline basis functions
- closed-form expression of the Hessian
- in case of $\nabla f$ is available → use when constructing $\hat{f}(x)$