University of Illinois at Urbana-Champaign (UIC)
Department of Electrical and Computer Engineering
Center for Computational Electromagnetics (CCEM)

A Sparse Data Fast Fourier Transform —
Algorithm and Implementation
Outline

• Conclusions
• Parabolic Reflector Problem
• Implementation
• The Algorithm in Higher Dimensions
• SDEFT in 1D
• Introduction
Introduction
transformation from the denser spatial grid in (a) to the sparser one in (b).

Letting the data, i.e., by applying the phase shifts corresponding to the

Update the Fourier data at each spectral point in (b) by spatially trans-

denser spectral grid in (b).

Interpolate the Fourier data in the sparse spectral grid in (a) to the the relevant phases.

the corresponding sparse spectral grid by aggregating all input data with

Start with the dense spatial grid in (a) and compute the Fourier data on

```
SDFFT in 1D
```
\( (4 N \log (\gamma N + 4 N)) O \) is the computational complexity of \( H \)-space points. Hence, the \( H \)-space points and ends with \( O(1) \) \( H \)-space data mapped to \( \gamma N \) \( H \)-space and \( \gamma N \) \( H \)-space data mapped to \( \gamma N \) \( H \)-space. Also, the process starts with \( N \). Thus, the procedure arrives at \( f \) after a series of aggregations and interpolations. In summary, starting with \( a \), the procedure is performed recursively until the desired spectral data are obtained.
which is interpolated to the denser mesh represented by \( k \), yielding \( f(\bar{r}_1) \).

The algorithm starts by computing

\[
\chi \in A \mathbb{K}_0 \in \mathbb{R} \prod_0^1 \mathbb{R} = (0 \mathbb{K}_0 f)
\]

Hence, we can compute the Fourier transform in a multi-level fashion. Given multidimensional vectors \( r \) and spectral points, we have \( k \) to denote spatial and spectral

\[
\chi \in A \mathbb{K} \in \mathbb{R} \prod_0^1 \mathbb{R} = (k_1 \mathbb{K}_1 f)
\]

The algorithm in Higher Dimensions
The interpolation and aggregation

\[ (4) \quad \forall k, f \in \mathcal{F}, \quad \sum_{i=1}^{k} (k + i) f_i = (k + 1) f \]

\[ (3) \quad \forall k, f \in \mathcal{F}, \quad f_{i-1} = f_i \]

\[ (2) \quad \forall k, f \in \mathcal{F}, \quad \sum_{i=1}^{k} (k + i) f_i = (k + 1) f \]

Thus, the algorithm builds cover-sets over the desired spectral points as shown below.
The $k$-space tree as we go up the $r$-space tree, $\forall$ becomes accordingly smaller as we go down. Our implementation involves two possibly unbalanced trees for the two domains as above. In the $k$-space tree, the SDFFT process starts at a level governed by the Nyquist criterion. While $\forall$ becomes larger.
Parabolic Reflector Problem
The field can be rewritten as:

In order to compute broadband data as in the figure, we need to find the $k_0$ dependence of the transfor.
the latter term may be neglected. However, since $k_R \ll 1$,
to be computed for each dimension. Because of the
transforms in terms of a shifted $z$-component, it
suffices to perform the Fourier

$\mathbf{k}_1 = \mathbf{K}^0 R^0 \mathbf{z}$ to the integral is due to $\mathbf{K} R^0$.
A stationary phase analysis reveals that the main contribution
<table>
<thead>
<tr>
<th>CPU Time (s)</th>
<th>Memory (MB)</th>
<th>Max. Frequency (MHz)</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>346</td>
<td>373</td>
<td>1200</td>
<td>31415</td>
</tr>
<tr>
<td>156</td>
<td>181</td>
<td>600</td>
<td>30397</td>
</tr>
<tr>
<td>87</td>
<td>118</td>
<td>300</td>
<td>65798</td>
</tr>
</tbody>
</table>

N
costs related to 3D interpolation. The spectral domain has been presented.

Conclusions.