Comparing
High-Order Multivariate
AD Methods

Prof. Richard D. Neidinger
Ben Altman, as undergraduate
(now at Goldman Sachs)
Davidson College, NC, USA

AD2016
Oxford University, Sept. 15, 2016
Multivariate Taylor Methods

\[ f(x) \approx \sum_{|i| \leq d} F(i)(x - a)^i \]

for \( x, a \in \mathbb{R}^n \) and

\[ \binom{n+d}{d} \]

multi-indices \( i = (i_1, \ldots, i_n) \) in \( \mathbb{N}_0^n \)

Our goal: full set of coefficients in this massive, non-rectangular data structure \( F \), that I will call a corner; corresponding to unique derivative values

\[ F(i) = D_i(f) / i! \]
Four implementations of three methods

• **Direct Forward Method**
  operations on corners of series coefficients (AD)

• **Interpolation Methods**
  Compute univariate Taylor coefficients \( u_j \) in

\[
f(a + tr_j) \approx \sum_{k=0}^{d} u_j(k) t^k \quad \text{for directions } r_j, j=1 \text{ to } \#(\|j\|=d)
\]

then reconstruct multivariate \( F(i) \) for all \( |i| = k \leq d \)

• **GUW directions**: all directions used

• **Nested directions**: only use enough directions for \( k \)
  • Two different implementations: Divided-difference and LU
Test function – partials to 25\textsuperscript{th} order

function f = tennis(params)
%horizontal range of a serve with initial params
a = params(1); % angle in degrees
v = params(2); % speed in ft/sec
h = params(3); % height in ft
rad = a*pi/180;
tana = tan(rad);
vhor = (v*cos(rad))^2;
f = (vhor/32)*(tana + sqrt(tana^2+64*h/vhor));

- Overloaded in four versions of AD in
- MATLAB 2015b on
- Windows 7 Laptop.
## Accuracy Comparisons

**Tennis serve function** \((n=3, d=25)\)

<table>
<thead>
<tr>
<th></th>
<th>Direct Forward Corner</th>
<th>GUW Interpol</th>
<th>Nested Interpol Div-Diff</th>
<th>Nested Interpol LU</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Max abs error</strong></td>
<td>1.75e-15</td>
<td>3.60e-06</td>
<td>1.86e-06</td>
<td>1.51e-06</td>
</tr>
<tr>
<td><strong>Max relative error</strong></td>
<td>6.39e-06</td>
<td>3.39e+15</td>
<td>6.03e+16</td>
<td>5.37e+16</td>
</tr>
<tr>
<td><strong>Fix order=25: max error / max value</strong></td>
<td>1.30e-15</td>
<td>7.91e-03</td>
<td>4.09e-03</td>
<td>3.32e-03</td>
</tr>
<tr>
<td></td>
<td>(max 25(^{th}) order partial is 4.55e-04)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
log (max error / max value) by order

-20
-15
-10
-5
0
5
10
to power

max value
direct
GUW
NestDD
NestLU

order d
Verify error is from the methods

- Simple example:
  
  \[ f(x, y) = \exp(x + y) \text{ at (0,0)} \]

  compute \( D_i(f) = 1 \) up through \( |i| \leq d = 9 \) or \( d = 25 \).

- Direct Forward AD implementation has full numerical accuracy (max error: \( 2 \times 10^{-16} \) for \( d = 9 \) and \( 5 \times 10^{-16} \) for \( d = 25 \)).

- Do interpolation methods by matrix operations in Wolfram Mathematica to show source of numerical error.
GUW interpolation as matrix multiplication

\[ H_k \begin{bmatrix} u_j(k) : \text{directions } |j| = d \end{bmatrix} = \begin{bmatrix} D_{ijf} : |i| = k \end{bmatrix} \]

\( H_k \) only depends on \( n \) and \( d \) [GUW]. Use exact \( u_j(k) \) for \( \exp(x+y) \).

<table>
<thead>
<tr>
<th>Same Mathematica code for ( H_k )</th>
<th>( d = 9 ) error</th>
<th>( d = 25 ) error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using ( H_k ) exact rationals</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>N[( H_k )] exact to double</td>
<td>7.0e-14</td>
<td>4.4e-7</td>
</tr>
<tr>
<td>Condition Number of ( H_k )</td>
<td>10^3</td>
<td>10^{10}</td>
</tr>
<tr>
<td>( H_k ) doubles throughout</td>
<td>1.5e-11</td>
<td>1.6e+1</td>
</tr>
<tr>
<td>My MATLAB GUW</td>
<td>1.1e-11</td>
<td>1.4e+0</td>
</tr>
</tbody>
</table>
Nested interpolation as matrix solve

\[ M_k^{-1} \left[ u_j(k) : \text{first } \#(|i| = k) \text{ directions} \right] = \left[ F(i) : |i| = k \right] \]

Square \( M_k \) only depends on \( n \) and \( d \). Use exact \( u_j(k) \) for \( \exp(x+y) \).

<table>
<thead>
<tr>
<th>Same Mathematica code for ( M_k )</th>
<th>( d = 9 ) error</th>
<th>( d = 25 ) error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Using ( M_k^{-1} ) exact rationals</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \text{N}[M_k^{-1}] ) exact to double</td>
<td>( 9.8e-13 )</td>
<td>( 7.0e-3 )</td>
</tr>
<tr>
<td>Condition Number of ( M_k )</td>
<td>( 10^4 )</td>
<td>( 10^{13} )</td>
</tr>
<tr>
<td>( (M_k)^{-1} ) doubles throughout</td>
<td>( 1.2e-11 )</td>
<td>( 2.4e+2 )</td>
</tr>
<tr>
<td>My MATLAB Nested LU</td>
<td>( 1.3e-12 )</td>
<td>( 2.3e+1 )</td>
</tr>
</tbody>
</table>
### Efficiency Comparisons

**Tennis serve function \((n=3, d=25)\)**

<table>
<thead>
<tr>
<th></th>
<th>Direct Forward Corner</th>
<th>GUW Interpol</th>
<th>Nested Interpol Div-Diff</th>
<th>Nested Interpol LU</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Evaluation time (sec)</strong></td>
<td>0.55</td>
<td>0.52</td>
<td>1.18</td>
<td>0.48</td>
</tr>
<tr>
<td><strong>Rank fewest arith. flops</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td><strong>One-time globals (sec)</strong></td>
<td>0.31</td>
<td>9.47</td>
<td>0.0008</td>
<td>2.75</td>
</tr>
<tr>
<td><strong>largest global (MB)</strong></td>
<td>5.96</td>
<td>8.77</td>
<td>0.005</td>
<td>5.46</td>
</tr>
</tbody>
</table>

(one corner is 0.025 or 3,276 partials)
Direct Forward Implementation

- Direct Forward Method consistently gives full numerical accuracy in these examples!
- How do we make it efficient?
- Fix $n$ variable and up to maximum order $d$
- Each multidimensional corner (corresponding to Taylor coefficients) is stored in a linear array in increasing degree (order).
- **Global reference** (cell) array: For each linear index (corresponding to one multi-index in a corner) store a different size 2-dimensional matrix.
- Each 2-D matrix array stores linear indices of the $n$-D sub-box of multi-indices.
Direct coef of \((x-a_1)^2\) \((y-a_2)^0\) \((z-a_3)^3\)
in a \(n=3\), \(d=5\) corner needs multi-indices \(\leq 203\) (linear index 45)

<table>
<thead>
<tr>
<th></th>
<th>000 (1)</th>
<th>001 (4)</th>
<th>002 (10)</th>
<th>003 (20)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 (2)</td>
<td>101 (7)</td>
<td>102 (16)</td>
<td>103 (30)</td>
<td></td>
</tr>
<tr>
<td>200 (5)</td>
<td>201 (13)</td>
<td>202 (26)</td>
<td>203 (45)</td>
<td></td>
</tr>
</tbody>
</table>

For \(h = u \ast v\), \(h_{203 \ (45)} =
\begin{align*}
u[ & 1 \ 4 \ 10 \ 20 ] \ast v[ & 45 \ 26 \ 13 \ 5 ]^T \\
+ u[ & 2 \ 7 \ 16 \ 30 ] \ast v[ & 30 \ 16 \ 7 \ 2 ]^T \\
+ u[ & 5 \ 13 \ 26 \ 45 ] \ast v[ & 20 \ 10 \ 4 \ 1 ]^T
\end{align*}
For $h(x,y,z) = \exp(u(x,y,z))$, use

$$h_x = u_x \ast h$$

$$h_{203\ (45)} = \frac{1}{2} (0 \ast u[1\ 4\ 10\ 20] \ast h[45\ 26\ 13\ 5]^T + 1 \ast u[2\ 7\ 16\ 30] \ast h[30\ 16\ 7\ 2]^T + 2 \ast u[5\ 13\ 26\ 45] \ast h[20\ 10\ 4\ 1]^T)$$

$x$ is chosen as the smallest nonzero in $(2,0,3)$.

By only distinguishing the smallest nonzero, any $n$-dim sub-box is listed in rows of a 2-dim matrix.
Efficient generation of reference array

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>001</td>
<td>002</td>
<td>003</td>
</tr>
<tr>
<td>100</td>
<td>101</td>
<td>102</td>
<td>103</td>
</tr>
<tr>
<td>200</td>
<td>201</td>
<td>202</td>
<td>203</td>
</tr>
</tbody>
</table>

For matrix 45 combine matrix 30:               and           matrix  26:

```
1 4 10 20
2 7 16 30
5 13 26
```

1 4 10
2 7 16

But how do we find the linear indices 30 and 26?
Linear index of multi-index down 1

\[(i_1, i_2, i_3) = (2, 0, 3)\]
\[k = 45\]
\[s = 2 + 0 + 3 - 1\]
for \(j = 1\) to \(3\)

\[k = k - \binom{n - j + s}{s}\]
\[s = s - i_j\]

Results in linear indices \(k = 45, 30, 27, 26\)
2-D matrix for location 213(70)

- All entries in 203(45) matrix sorted into first row

- All entries in 113(49) and 212(44) matrices corresponding to $i_2=1$ are inserted into second row

\[
\begin{array}{cccccccccccc}
1 & 2 & 4 & 5 & 7 & 10 & 13 & 16 & 20 & 26 & 30 & 45 \\
3 & 6 & 9 & 12 & 15 & 19 & 25 & 29 & 34 & 44 & 49 & 70
\end{array}
\]
References:

### Accuracy Comparisons $n=8$, $d=8$

\[
t = x_1^2 + 2x_2^2 + 3x_3^2 + 4x_4^2 + 5x_5^2 + 6x_6^2 + 7x_7^2 + 8x_8^2
\]

\[
f(x) = \exp(-\sqrt{t}) \sin(t \cdot \ln(1 + t))
\]

<table>
<thead>
<tr>
<th></th>
<th>Direct Forward Corner</th>
<th>GUW Interpol</th>
<th>Nested Interpol Div-Diff</th>
<th>Nested Interpol LU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max abs error</td>
<td>2.81e-01</td>
<td>1.15e+01</td>
<td>1.72e+00</td>
<td>3.41e+00</td>
</tr>
<tr>
<td>Max relative error</td>
<td>8.57e-12</td>
<td>4.20e-02</td>
<td>7.12e-04</td>
<td>7.05e-04</td>
</tr>
<tr>
<td>Fix order=8: max error / max value</td>
<td>9.70e-15</td>
<td>3.98e-13</td>
<td>5.94e-14</td>
<td>1.17e-13</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(max 10th order partial is 2.90e+13)</td>
<td></td>
</tr>
</tbody>
</table>
Efficiency Comparisons $n=8, d=8$

$$t = x_1^2 + 2x_2^2 + 3x_3^2 + 4x_4^2 + 5x_5^2 + 6x_6^2 + 7x_7^2 + 8x_8^2$$

$$f(x) = \exp(-\sqrt{t}) \sin(t \cdot \ln(1 + t))$$

<table>
<thead>
<tr>
<th></th>
<th>Direct Forward Corner</th>
<th>GUW Interpol</th>
<th>Nested Interpol Div-Diff</th>
<th>Nested Interpol LU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation time (sec)</td>
<td>1.65</td>
<td>14.29</td>
<td>12.25</td>
<td>11.64</td>
</tr>
<tr>
<td>One-time globals (sec)</td>
<td>0.79</td>
<td>312.1</td>
<td>0.0002</td>
<td>218.4</td>
</tr>
<tr>
<td>largest global (MB)</td>
<td>7.0</td>
<td>631.8</td>
<td>0.0005</td>
<td>434.0</td>
</tr>
<tr>
<td></td>
<td>(one corner is 0.098</td>
<td></td>
<td>or 12,870</td>
<td></td>
</tr>
<tr>
<td></td>
<td>or 12,870</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Accuracy Comparisons $n=5$, $d=10$

$$t = x_1^2 + 2x_2^2 + 3x_3^2 + 4x_4^2 + 5x_5^2$$

$$f(x) = \exp(-\sqrt{t}) \sin(t \cdot \ln(1 + t))$$

<table>
<thead>
<tr>
<th></th>
<th>Direct Forward Corner</th>
<th>GUW Interpol</th>
<th>Nested Interpol Div-Diff</th>
<th>Nested Interpol LU</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Max abs error</strong></td>
<td>4.36e-07</td>
<td>6.37e-03</td>
<td>8.01e-03</td>
<td>2.65e-01</td>
</tr>
<tr>
<td><strong>Max relative error</strong></td>
<td>4.39e-13</td>
<td>2.45e-08</td>
<td>7.25e-10</td>
<td>2.76e-08</td>
</tr>
<tr>
<td><strong>Fix order=10:</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>max error / max value</td>
<td>6.61e-16</td>
<td>9.66e-12</td>
<td>1.22e-11</td>
<td>4.02e-10</td>
</tr>
<tr>
<td>(max 10th order partial is 6.59e+08)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Efficiency Comparisons $n=5, d=10$

$$t = x_1^2 + 2x_2^2 + 3x_3^2 + 4x_4^2 + 5x_5^2$$

$$f(x) = \exp(-\sqrt{t}) \sin(t \cdot \ln(1+t))$$

<table>
<thead>
<tr>
<th></th>
<th>Direct Forward Corner</th>
<th>GUW Interpol</th>
<th>Nested Interpol Div-Diff</th>
<th>Nested Interpol LU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation time (sec)</td>
<td>0.38</td>
<td>1.64</td>
<td>1.63</td>
<td>1.40</td>
</tr>
<tr>
<td>One-time globals (sec)</td>
<td>0.22</td>
<td>12.64</td>
<td>0.0008</td>
<td>7.30</td>
</tr>
<tr>
<td>largest global (MB)</td>
<td>1.73</td>
<td>22.9</td>
<td>0.0008</td>
<td>14.8</td>
</tr>
</tbody>
</table>

(one corner is 0.023 or 3,003 partials)
GUW Interpolation Formula

\[ H_k \left[ u_j(k) : \text{directions } |j| = d \right] = \left[ D_i f : |i| = k \right] \]

\[ H_k = \left[ c_{i,j} \right] \text{ where} \]

\[ c_{i,j} = \sum_{0<p\leq i} (-1)^{|i-p|} \left( \frac{|p|}{d} \right)^i \binom{i}{p} \left( \frac{dp}{|p|} \right)^j \]
Nested interpolation as matrix solve

- For $n=2$ and $d=9$ use 10 directions:
  \[ \mathbf{r}_j = (1, w_j) \text{ for } j = 0 \ldots 9 \]
  \[ w_j = 0, 1, -1, \frac{1}{2}, -\frac{1}{2}, \frac{1}{4}, -\frac{1}{4}, \frac{3}{4}, -\frac{3}{4}, \frac{1}{8} \]
  \[ u_j(t) = f \left( (0,0) + t \mathbf{r}_j \right) = f(t, tw_j) \]

- Multivariate: \( f(x, y) = \sum \sum F(i_1, i_2) x^{i_1} y^{i_2} \)

- Univariate: \( u_j(t) = \sum \sum F(i)(w_j)^{i_2} t^k \)

- Relation: \[ \sum_{|i|=k} (w_j)^{i_2} F(i) = u_j(k) \]

- Forms \( (k+1) \times (k+1) \) matrix \( M_k \) such that
  \[ M_k \left[ F(i) \right]_{|i|=k} = \left[ u_j(k) \right]_{j=0}^k \]
## Evaluation Time
### Theoretical Operation Counts

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of mults</th>
<th>( n=10, \ d=5 )</th>
<th>( n=3, \ d=20 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct Forward</td>
<td>( N \cdot \text{ops}_f )</td>
<td>( N=53,130 )</td>
<td>( N=230,230 )</td>
</tr>
<tr>
<td>Univariate Forward</td>
<td>( q \cdot N \cdot \text{ops}_f )</td>
<td>( q=.79 )</td>
<td>( q=.23 )</td>
</tr>
<tr>
<td>GUW Interpolation</td>
<td>( p ) nonzeros (or mat size)</td>
<td>118,502 3,002\times 2,002</td>
<td>275,370 1,770\times 231</td>
</tr>
<tr>
<td>Nested Interpolation</td>
<td>( dd + ct )</td>
<td>( 12,285 ) +10,230</td>
<td>( 17,710 ) +67,298</td>
</tr>
</tbody>
</table>