To solve the system of equations, two factorizations were found; Type 1 & Type 2. Type 2 solves the system iteratively. This is a generalization of the original 1970 Bjork-Pereyra algorithm to the complex plane. Type 2 solves the system recursively. The factorization of Type 2 holds the subtraction to a single value for the calculation of lower triangular matrices. This reduces the point floating point error in the sequent calculations and even leads to more accurate algorithm.

**Sparsity Factorization: Type 1**

\[
\begin{align*}
V_{\omega_{n}}(\omega_{k}) = & \frac{\omega_{k}^{n}}{\prod_{j=1}^{n}(\omega_{k} - \omega_{j})} \\
L_{k} = & \frac{1}{\prod_{j=1}^{n}(\omega_{k} - \omega_{j})} \\
\end{align*}
\]

**Algorithms for Type 1 and Type 2**

Input: \(n, f_k, x_k = \frac{-n-k}{k} \), for \( k = 1,2,...,n \)

**Recursion:**

\[
\begin{align*}
\beta_k &= f_k & \text{for } k = 1,2,...,n \\
\end{align*}
\]

\[
\begin{align*}
\beta_1 &= f_1 \\
\beta_2 &= f_2 \\
\beta_3 &= f_3 \\
\vdots & & \vdots \\
\beta_{n-1} &= f_{n-1} \\
\end{align*}
\]

**Sparse Factorization: Type 2**

\[
\begin{align*}
L_k &= \frac{1}{\prod_{j=1}^{n}(\omega_k - \omega_j)} \\
h_k &= \frac{1}{\prod_{j=1}^{n}(\omega_k - \omega_j)} \\
D_k &= \frac{1}{\prod_{j=1}^{n}(\omega_k - \omega_j)} \\
\end{align*}
\]

**RESULTS**

**Forward Error Bound for Both Algorithms**

\[
\frac{|\beta - \beta_o|}{|\beta_o|} \leq (1 - \lambda) \sum_{k=2}^{n} |\beta_k|
\]

**Numerical Results**

<table>
<thead>
<tr>
<th>Size</th>
<th>BP-Type 1</th>
<th>BP-Type 2</th>
<th>Gaussian Elimination</th>
<th>BP-Type 1</th>
<th>BP-Type 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10×10</td>
<td>1.491E-06</td>
<td>1.843E-08</td>
<td>2.645E-08</td>
<td>1.640E-07</td>
<td>1.894E-08</td>
</tr>
<tr>
<td>20×20</td>
<td>3.876E-04</td>
<td>1.619E-06</td>
<td>2.391E-06</td>
<td>2.422E-07</td>
<td>2.445E-08</td>
</tr>
<tr>
<td>30×30</td>
<td>8.302E-02</td>
<td>7.940E-09</td>
<td>2.517E-08</td>
<td>5.506E-07</td>
<td>1.955E-08</td>
</tr>
<tr>
<td>35×35</td>
<td>3.425E+00</td>
<td>2.266E-09</td>
<td>1.735E-07</td>
<td>5.675E-07</td>
<td>2.933E-08</td>
</tr>
<tr>
<td>40×40</td>
<td>2.443E+02</td>
<td>1.338E-07</td>
<td>3.314E-07</td>
<td>5.992E-06</td>
<td>1.262E-06</td>
</tr>
<tr>
<td>50×50</td>
<td>7.089E+06</td>
<td>3.193E-05</td>
<td>1.327E-05</td>
<td>3.224E-05</td>
<td>4.013E-07</td>
</tr>
<tr>
<td>60×60</td>
<td>1.657E+12</td>
<td>1.389E-04</td>
<td>3.916E-04</td>
<td>7.088E-04</td>
<td>1.433E-05</td>
</tr>
<tr>
<td>70×70</td>
<td>9.646E+16</td>
<td>3.264E-03</td>
<td>7.088E-03</td>
<td>1.104E-02</td>
<td>2.476E-05</td>
</tr>
<tr>
<td>80×80</td>
<td>1.089E+21</td>
<td>3.533E-02</td>
<td>3.991E-02</td>
<td>3.741E-02</td>
<td>2.226E-05</td>
</tr>
<tr>
<td>90×90</td>
<td>6.079E+26</td>
<td>6.264E+09</td>
<td>1.160E-01</td>
<td>3.155E-01</td>
<td>4.361E-06</td>
</tr>
<tr>
<td>100×100</td>
<td>7.919E+31</td>
<td>9.359E+14</td>
<td>2.844E-01</td>
<td>3.256E-01</td>
<td>1.474E-06</td>
</tr>
<tr>
<td>105×105</td>
<td>6.272E+34</td>
<td>1.781E+17</td>
<td>6.272E+06</td>
<td>3.342E+06</td>
<td>1.467E-06</td>
</tr>
<tr>
<td>150×150</td>
<td>8.979E+60</td>
<td>6.321E+16</td>
<td>1.558E-06</td>
<td>1.138E-05</td>
<td>1.506E-05</td>
</tr>
<tr>
<td>200×200</td>
<td>3.171E+91</td>
<td>9.046E+40</td>
<td>1.708E-06</td>
<td>1.433E-08</td>
<td>1.506E-05</td>
</tr>
<tr>
<td>250×250</td>
<td>1.593E+122</td>
<td>1.680E+52</td>
<td>1.433E-08</td>
<td>1.433E-08</td>
<td>1.433E-08</td>
</tr>
<tr>
<td>300×300</td>
<td>1.087E+182</td>
<td>1.841E+68</td>
<td>1.433E-08</td>
<td>1.433E-08</td>
<td>1.433E-08</td>
</tr>
</tbody>
</table>

**CONCLUSION**

Results on polynomial interpolation from real to complex plane leads to:

- Fast \(O(n^2)\) Algorithms
- Sparse Factorizations
- Iterative Algorithms
- Stable Algorithms with Leja Ordering
- Accurate Algorithms beyond Gaussian Elimination for \(n > 100\)

**REFERENCES**