Practical Algorithms for Solving the Quartic Equation
David J. Wolters
June 17, 2020

Each of the five algorithms presented here for solving the quartic equation provides:
- stable analytic solutions for any combination of real coefficients,
- formulas that convert easily to code, and
- calculations that use real numbers only.


The algorithm inputs are four real coefficients $A_3, A_2, A_1,$ and $A_0,$ and the outputs are the four values $Z_1, Z_2, Z_3$ and $Z_4$ such that
$$Z^4 + A_3 Z^3 + A_2 Z^2 + A_1 Z + A_0 = (Z-Z_1) (Z-Z_2) (Z-Z_3) (Z-Z_4) \quad \text{for all } Z.$$ The outputs are thus the four solutions of the general quartic equation
$$Z_n^4 + A_3 Z_n^3 + A_2 Z_n^2 + A_1 Z_n + A_0 = 0, \quad n = 1, 2, 3, 4. \quad (1)$$

Except for the NBS method, the algorithms begin by calculating $C = A_3 / 4$, $b_2$, $b_1$, and $b_0$. The last three of these values are coefficients of the equivalent depressed quartic equation with no cubic term:
$$T_n^4 + b_2 T_n^2 + b_1 T_n + b_0 = 0 \quad \text{for all } T_n. \quad \text{n = 1, 2, 3, 4.} \quad (2)$$
The solutions $Z_n$ of (1) are related to the solutions $T_n$ of (2) by $Z_n = T_n - C$. The coefficients $b_2$, $b_2$, and $b_0$ are calculated from $C$, $A_2$, $A_1$, and $A_0$ as shown in the algorithm tables below.

Part I of this document presents the five algorithms in both their original and modified forms. Notes explain the computational shortcoming of each original algorithm and the fix. Part II concludes with check equations to validate the set of four calculated solutions $Z_n$. Part II assesses the suitability of each algorithm for general calculation and demonstrates that all of the algorithms are mathematically equivalent to each other. Part III derives the algorithms.

Unless noted otherwise, the radical sign $\sqrt{\phantom{1}}$ denotes the principal square root. The principal square root of a positive real number is the positive square root. The principal square root of a negative real number is the positive imaginary square root. If $z$ is complex with modulus $r$ and argument $\phi$ such that $-\pi < \phi \leq \pi$, then $z = r e^{i \phi}$ and the principal square root of $z$ is $\sqrt{z} = \sqrt{r} e^{i \phi/2}$.

Analytic methods for solving quartic equations, including the algorithms presented here, require the solution(s) of a corresponding resolvent cubic equation. See the companion paper on algorithms for solving cubic equations.
PART I -- The Algorithms and Check Equations
Ferrari’s Method

<table>
<thead>
<tr>
<th>Common Algorithm</th>
<th>Modified Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Given:</strong> Real coefficients $A_3$, $A_2$, $A_1$, and $A_0$,</td>
<td></td>
</tr>
<tr>
<td><strong>Find:</strong> $Z_1$, $Z_2$, $Z_3$ and $Z_4$ such that</td>
<td></td>
</tr>
<tr>
<td>$Z^4 + A_3Z^3 + A_2Z^2 + A_1Z + A_0 = (Z-Z_1)(Z-Z_2)(Z-Z_3)(Z-Z_4)$ for all $Z$.</td>
<td></td>
</tr>
<tr>
<td><strong>Calculation:</strong> $C = \frac{A_3}{4}$, $b_2 = A_2 - 6C^2$, $b_1 = A_1 - 2A_2C + 8C^3$, $b_0 = A_0 - A_1C + A_2C^2 - 3C^4$</td>
<td></td>
</tr>
<tr>
<td>Solve this resolvent cubic equation for real $m$:</td>
<td></td>
</tr>
<tr>
<td>$m^3 + b_2m^2 + (b_2^2/4 - b_0)m - b_0^2/8 = 0$.</td>
<td></td>
</tr>
<tr>
<td>Use a real solution $m &gt; 0$ if it exists. Otherwise, $m = 0$.</td>
<td></td>
</tr>
<tr>
<td><strong>Case: $m &gt; 0$</strong></td>
<td></td>
</tr>
<tr>
<td>$Z_{1,2} = \sqrt[3]{m/2} - C \pm \sqrt{-m/2 - b_2/2 - b_1/(2\sqrt{2m})}$</td>
<td></td>
</tr>
<tr>
<td>$Z_{3,4} = -\sqrt[3]{m/2} - C \pm \sqrt{-m/2 - b_2/2 + b_1/(2\sqrt{2m})}$</td>
<td></td>
</tr>
<tr>
<td><strong>Case: $m = 0$</strong></td>
<td></td>
</tr>
<tr>
<td>$Z_{1,2} = -C \pm \sqrt{b_2/2 + \sqrt{b_2^2/4 - b_0}}$</td>
<td></td>
</tr>
<tr>
<td>$Z_{3,4} = -C \pm \sqrt{-b_2/2 - \sqrt{b_2^2/4 - b_0}}$</td>
<td></td>
</tr>
<tr>
<td>where $b_2/4 - b_0 \geq 0$ provided that no real $m &gt; 0$ exists.</td>
<td></td>
</tr>
<tr>
<td><strong>Given:</strong> Real coefficients $A_3$, $A_2$, $A_1$, and $A_0$,</td>
<td></td>
</tr>
<tr>
<td><strong>Find:</strong> $Z_1$, $Z_2$, $Z_3$ and $Z_4$ such that</td>
<td></td>
</tr>
<tr>
<td>$Z^4 + A_3Z^3 + A_2Z^2 + A_1Z + A_0 = (Z-Z_1)(Z-Z_2)(Z-Z_3)(Z-Z_4)$ for all $Z$.</td>
<td></td>
</tr>
<tr>
<td><strong>Calculation:</strong> $C = \frac{A_3}{4}$, $b_2 = A_2 - 6C^2$, $b_1 = A_1 - 2A_2C + 8C^3$, $b_0 = A_0 - A_1C + A_2C^2 - 3C^4$</td>
<td></td>
</tr>
<tr>
<td>Solve this resolvent cubic equation for real $m$:</td>
<td></td>
</tr>
<tr>
<td>$m^3 + b_2m^2 + (b_2^2/4 - b_0)m - b_0^2/8 = 0$.</td>
<td></td>
</tr>
<tr>
<td>Use a real solution $m &gt; 0$ if it exists. Otherwise, $m = 0$.</td>
<td></td>
</tr>
<tr>
<td>$\Sigma = \begin{cases} 1 &amp; \text{if } b_1 &gt; 0 \ -1 &amp; \text{otherwise} \end{cases}$</td>
<td></td>
</tr>
<tr>
<td>$R = \Sigma \sqrt{m^2 + b_2m + b_2^2/4 - b_0}$</td>
<td></td>
</tr>
<tr>
<td>$Z_{1,2} = \sqrt[3]{m/2} - C \pm \sqrt{-m/2 - b_2/2 - R}$</td>
<td></td>
</tr>
<tr>
<td>$Z_{3,4} = -\sqrt[3]{m/2} - C \pm \sqrt{-m/2 - b_2/2 + R}$</td>
<td></td>
</tr>
<tr>
<td>where the radicand in the formula for $R$ is nonnegative provided that real $m &gt; 0$ is used if it exists.</td>
<td></td>
</tr>
</tbody>
</table>

In his 1545 book *Ars Magna*, Girolamo Cardano provides the earliest known description of Ferrari’s method. Modern algebraic notation had not been invented at the time. To demonstrate the method, Cardano gave rules for solving the depressed quartic equation and then worked out sample problems. The common algorithm is a modern generalization of Cardano’s first demonstration problem, Problem V.

The quotient $b_1/(2\sqrt{2m})$ in the formulas for $Z_n$ causes the common algorithm to become computationally unstable as $m$ approaches zero. The $m$ value, calculated as the solution of the resolvent cubic equation, typically contains a small round-off error not found in $b_1$. As $b_1$ and true $m$ approach zero, the round-off error dominates the calculated $m$ value, and the algorithm becomes unstable. The appendix demonstrates a particularly severe case in which the calculated value of solution $Z_1$ suffers large error even when $m$ is several orders of magnitude greater than the round-off error.

The modified algorithm, a modern generalization of Cardano’s Problem VIII, avoids the instability by replacing $b_1/(2\sqrt{2m})$ with $R$. To check the validity of this replacement, add $b_1^2/8$ to both sides of the resolvent cubic equation, divide through by $m$, and take the square root.
Descartes' Method

Original Algorithm

Given: Real coefficients $A_3, A_2, A_1,$ and $A_0$

Find: $Z_1, Z_2, Z_3$ and $Z_4$ such that

$$Z_1 + A_3 Z_3 + A_2 Z_2 + A_1 Z + A_0 = (Z-Z_1) (Z-Z_2) (Z-Z_3) (Z-Z_4)$$

for all $Z$.

Calculation: $C = A_3/4, \quad b_2 = A_2 - 6C^2,$

$$b_1 = A_1 - 2A_2 C + 8C^3,$$

$$b_0 = A_0 - A_1 C + A_2 C^2 - 3C^4$$

Solve this resolvent cubic equation for real $y^2$:

$$y^6 + 2b_2 y^4 + (b_2^2 - 4b_0) y^2 - b_1^2 = 0.$$

Use a real solution $y^2 > 0$ if it exists. Otherwise, $y^2 = 0$. Value $y$ is the nonnegative square root of $y^2$.

Case: $y^2 > 0$

$$Z_{1,2} = y/2 - C \pm \sqrt{-y^2/4 - b_2/2 - b_1/(2y)}$$

$$Z_{3,4} = -y/2 - C \pm \sqrt{-y^2/4 - b_2/2 + b_1/(2y)}$$

Case: $y^2 = 0$

$$Z_{1,2} = -C \pm \sqrt{-b_2/2 + \sqrt{b_2^2/4 - b_0}}$$

$$Z_{3,4} = -C \pm \sqrt{-b_2/2 - \sqrt{b_2^2/4 - b_0}}$$

where $b_2^2/4 - b_0 \geq 0$ provided no real $y^2 > 0$ exists.

Modified Algorithm

Given: Real coefficients $A_3, A_2, A_1,$ and $A_0$

Find: $Z_1, Z_2, Z_3$ and $Z_4$ such that

$$Z_1 + A_3 Z_3 + A_2 Z_2 + A_1 Z + A_0 = (Z-Z_1) (Z-Z_2) (Z-Z_3) (Z-Z_4)$$

for all $Z$.

Calculation: $C = A_3/4, \quad b_2 = A_2 - 6C^2,$

$$b_1 = A_1 - 2A_2 C + 8C^3,$$

$$b_0 = A_0 - A_1 C + A_2 C^2 - 3C^4$$

Solve this resolvent cubic equation for real $y^2$:

$$y^6 + 2b_2 y^4 + (b_2^2 - 4b_0) y^2 - b_1^2 = 0.$$

Use a real solution $y^2 > 0$ if it exists. Otherwise, $y^2 = 0$. Value $y$ is the nonnegative square root of $y^2$.

$$\Sigma = \begin{cases} 1 & \text{if } b_1 > 0 \\ -1 & \text{otherwise} \end{cases}$$

$$R = \Sigma \sqrt{y^4/4 + (b_2/2)y^2 + b_2^2/4 - b_0}$$

$$Z_{1,2} = y/2 - C \pm \sqrt{-y^2/4 - b_2/2 - R}$$

$$Z_{3,4} = -y/2 - C \pm \sqrt{-y^2/4 - b_2/2 + R}$$

where the radicand in the formula for $R$ is nonnegative provided that real $y^2 > 0$ is used if it exists.

The two Descartes algorithms are similar to the corresponding Ferrari algorithms. The Descartes formulas for $Z_0$ become the corresponding Ferrari formulas by substituting $\sqrt{2m}$ for $y$. Substitute $\sqrt{2m}$ for $y$ in the Descartes resolvent cubic equation and divide through by 8 to obtain the Ferrari resolvent cubic equation.

Like the Ferrari common algorithm, the Descartes original algorithm suffers computational instability as the solution $y^2$ of the resolvent cubic equation approaches zero. The instability is avoided in the Descartes modified algorithm just as it is in the Ferrari modified algorithm.
NBS Method

### Original Algorithm

Given \( Z^4 + A_2Z^3 + A_2Z^2 + A_1Z + A_0 = 0 \), find the real root \( u_1 \) of the cubic equation
\[
u^3 - A_2u^2 + (A_1A_3 - 4A_0)u - (A_1^2 + A_0A_3^3 - 4A_0A_2) = 0
\]
and determine the four roots of the quartic as solutions of the two quadratic equations
\[
v^2 + \left[ \frac{A_3}{2} \mp \left( \frac{A_3^2}{4} + u_1 - A_2 \right) \right]^{1/2} v + \frac{u_1}{2} \mp \left[ \left( \frac{u_1}{2} \right)^2 - \frac{u_1}{2} \right] = 0.
\]
If all roots of the cubic equation are real, use the value of \( u_1 \) which gives real coefficients in the quadratic equation and select signs so that if
\[
Z^4 + A_1Z^3 + A_2Z^2 + A_1Z + A_0 = (Z^2 + p_1Z + q_1)(Z^2 + p_2Z + q_2)
\]
then
\[
p_1 + p_2 = A_1, \quad p_1p_2 + q_1 + q_2 = A_2, \quad p_1q_2 + p_2q_1 = A_3, \quad q_1q_2 = A_0.
\]

### Modified Algorithm

**Problem:** Given real coefficients \( A_3, A_2, A_1, \) and \( A_0 \), find \( Z, Z_2, Z_3 \) and \( Z_4 \) such that
\[
Z^4 + A_3Z^3 + A_2Z^2 + A_1Z + A_0 = (Z-Z_1)(Z-Z_2)(Z-Z_3)(Z-Z_4)
\]
for all \( Z \).

**Solution:** Calculate \( u_1 \) as the greatest real solution of the resolvent cubic equation
\[
u^3 - A_2u^2 + (A_1A_3 - 4A_0)u + 4A_0A_2 - A_1^2 - A_0A_3^2 = 0.
\]
\[
\Sigma_# = \begin{cases} 
1 & \text{if } A_1 - A_3u_1/2 > 0 \\
-1 & \text{otherwise}
\end{cases}
\]

\[
p_1 = A_3/2 - \sqrt{A_3^2/4 + u_1 - A_2} \quad p_2 = A_3/2 + \sqrt{A_3^2/4 + u_1 - A_2}
\]

\[
q_1 = u_1/2 + \Sigma_# \sqrt{u_1^2/4 - A_0} \quad q_2 = u_1/2 - \Sigma_# \sqrt{u_1^2/4 - A_0}
\]

\[
Z_{1,2} = -p_1/2 \pm \sqrt{p_1^2/4 - q_1} \quad Z_{3,4} = -p_2/2 \pm \sqrt{p_2^2/4 - q_2}
\]

The NBS original algorithm is unnecessarily complicated and difficult to code. The user is left to perform trial-and-error tests for two of the algorithm steps: 1) if all three solutions of the resolvent cubic equation are real, select \( u_1 \) to produce real coefficients in the quadratic equations, and 2) in the two quadratic equations, choose the correct combination of signs to be used in the expressions for the coefficients. The user faces the possibility of performing trial-and-error tests on three cubic-equation solutions and four sign combinations to arrive at the two correct quadratic equations.

The modified algorithm is easy to code and satisfies all requirements of the original algorithm. Choosing \( u_1 \) as the greatest real solution of the resolvent cubic equation always provides real coefficients: \( p_1, p_2, q_1, \) and \( q_2 \). The function \( \Sigma_# \) as defined in the modified algorithm assures that correct signs are selected.
Euler's Method

**Original Algorithm**

Given: Real coefficients $A_3, A_2, A_1,$ and $A_0$  

Find: $Z_1, Z_2, Z_3$ and $Z_4$ such that  


Calculation:  

- $C = A_3/4, \quad b_2 = A_2 – 6C^2, \quad b_1 = A_1 – 2A_2C + 8C^3, \quad b_0 = A_0 – A_1C + A_2C^2 – 3C^4$  

Find the three solutions $r_1, r_2,$ and $r_3$ of the resolvent cubic equation:  

$r_k^3 + (b_2/2) r_k^2 + [(b_2^2 – 4b_0)/16] r_k – b_1^2/64 = 0.$  

$T_1 = \sqrt[3]{r_1} + \sqrt[3]{r_2} + \sqrt[3]{r_3}$  

$T_2 = \sqrt[3]{r_1} – \sqrt[3]{r_2} – \sqrt[3]{r_3}$  

$T_3 = -\sqrt[3]{r_1} + \sqrt[3]{r_2} – \sqrt[3]{r_3}$  

$T_4 = -\sqrt[3]{r_1} – \sqrt[3]{r_2} + \sqrt[3]{r_3}$  

The signs for the $\sqrt{r_k}$ are selected so that  

$\sqrt[r_1]{r_2} \sqrt[r_3]{r_3} = -b_1/8.$  

$Z_n = T_n – C, \quad n = 1, 2, 3, 4$  

The radical in the Euler original algorithm does not imply the principal square root. Instead the user selects either square root for any two of the $\sqrt{r_k}$. The third $\sqrt{r_k}$ is selected to satisfy $\sqrt[r_1]{r_2} \sqrt[r_3]{r_3} = -b_1/8$. The user only needs to check that the two sides of this equation have the same sign. The resolvent cubic equation guarantees that the two sides have the same absolute value:  

$(r – r_1)(r – r_2)(r – r_3) = r^3 + (b_2/2) r^2 + [(b_2^2 – 4b_0)/16] r – b_1^2/64 \quad \text{for all } r  

\Rightarrow r_1r_2r_3 = b_1^2/64 \Rightarrow |\sqrt[r_1]{r_2} \sqrt[r_3]{r_3}| = |b_1|/8. \quad (3)$

The resolvent cubic equation sometimes has two solutions that are a complex conjugate pair. The original algorithm then requires operations on complex numbers, but the modified algorithm does not. In the modified algorithm, all constituents of the $T_n$ formulas are real numbers, and the inner radicand, $x_2x_3 + y_2^2$, is nonnegative.

The modified algorithm's $T_n$ formulas may be expressed more simply as  

$T_{1,2} = \sqrt[r_1]{r_2 + r_3 – 2\sqrt[r_2]{r_3}}$  

$T_{3,4} = -\sqrt[r_1]{r_2 + r_3 + 2\sqrt[r_2]{r_3}}.$  

Solutions $r_2 = x_2 + iy_2$ and $r_3 = x_3 + iy_3$ of the resolvent cubic equation are real ($y_2 = y_3 = 0$), or they form a complex conjugate pair ($x_2 = x_3, y_2 = -y_3 > 0$). In either case, the sum $r_2 + r_3$ equals $x_2 + x_3,$ and the product $r_2r_3$ equals $x_2x_3 + y_2^2.$
The DLMF methods are similar to the corresponding Euler algorithms. The Euler $T_n$ formulas become the corresponding DLMF formulas with the following substitutions:

$$ r_k = -\theta_k/4, \quad x_k = -\theta_k x_k/4, \quad y_k = -\theta_k y_k/4, \quad k = 1, 2, 3. $$

Substitute $-\theta_k/4$ for $r_k$ in the Euler resolvent cubic equation and simplify to obtain the DLMF resolvent cubic equation. Like the Euler algorithms, the DLMF original algorithm requires operations on complex numbers, but the DLMF modified algorithm does not.

The $T_n$ formulas in the DLMF modified algorithm may be expressed more simply as

$$ T_{1,2} = \frac{1}{2} \left[ \sqrt{-\theta_1} \pm \sqrt{-\theta_2 - \theta_3 - 2\Sigma \sqrt{\theta_2 \theta_3}} \right] \quad T_{3,4} = \frac{1}{2} \left[ -\sqrt{-\theta_1} \pm \sqrt{-\theta_2 - \theta_3 + 2\Sigma \sqrt{\theta_2 \theta_3}} \right] $$

(5)

Solutions $\theta_2 = \theta_2 + i\theta_{y_2}$ and $\theta_3 = \theta_3 + i\theta_{y_3}$ of the resolvent cubic equation are real ($\theta_2 = \theta_3 = 0$), or they form a complex conjugate pair ($\theta_2 = \theta_3$, $\theta_2 = \theta_3 > 0$). In either case, $-\theta_2 - \theta_3 = -\theta_2 - \theta_3$, and $\theta_2 \theta_3 = \theta_2 \theta_3 + \theta_{y_2}$. 

---

**DLMF Method**

**Original Algorithm**

**Given:** Real coefficients $A_3, A_2, A_1,$ and $A_0$

**Find:** $Z_1, Z_2, Z_3$ and $Z_4$ such that

$$ Z^3 + A_3 Z^3 + A_2 Z^2 + A_1 Z + A_0 = (Z - Z_1) (Z - Z_2) (Z - Z_3) (Z - Z_4) \text{ for all } Z. $$

**Calculation:**

$$ C = A_3/4, \quad b_2 = A_2 - 6C^2, \quad b_1 = A_1 - 2A_2 C + 8C^3, \quad b_0 = A_0 - A_1 C + A_2 C^2 - 3C^4 $$

Find the three solutions $\theta_1, \theta_2,$ and $\theta_3$ of the resolvent cubic equation:

$$ \theta_k^2 - 2b_2 \theta_k^2 + (b_2^2 - 4b_0) \theta_k + b_1^2 = 0. $$

The signs for the $\sqrt{-\theta}$ are selected so that

$$ \sqrt{-\theta_1} \sqrt{-\theta_2} \sqrt{-\theta_3} = -b_1. $$

$$ Z_n = T_n - C, \quad n = 1, 2, 3, 4 $$

**Modified Algorithm**

**Given:** Real coefficients $A_3, A_2, A_1,$ and $A_0$

**Find:** $Z_1, Z_2, Z_3$ and $Z_4$ such that

$$ Z^3 + A_3 Z^3 + A_2 Z^2 + A_1 Z + A_0 = (Z - Z_1) (Z - Z_2) (Z - Z_3) (Z - Z_4) \text{ for all } Z. $$

**Calculation:**

$$ C = A_3/4, \quad b_2 = A_2 - 6C^2, \quad b_1 = A_1 - 2A_2 C + 8C^3, \quad b_0 = A_0 - A_1 C + A_2 C^2 - 3C^4 $$

Find the three solutions $\theta_1, \theta_2,$ and $\theta_3$ of the resolvent cubic equation:

$$ \theta_k^2 - 2b_2 \theta_k^2 + (b_2^2 - 4b_0) \theta_k + b_1^2 = 0. $$

Solution $\theta_1$ is the least real solution and $-\theta_1 \geq 0$.

Solutions $\theta_2 = \theta_2 + i\theta_{y_2}$ and $\theta_3 = \theta_3 + i\theta_{y_3}$ are real ($\theta_{y_2} = \theta_{y_3} = 0$), or they form a complex conjugate pair ($\theta_{x_2} = \theta_{x_3}, -\theta_{y_2} = \theta_{y_3} > 0$).

$$ \Sigma = 1 \text{ if } b_1 > 0, \Sigma = -1 \text{ otherwise.} $$

$$ T_{1,2} = \frac{1}{2} \left[ \sqrt{-\theta_1} \pm \sqrt{-\theta_2 - \theta_3 - 2\Sigma \sqrt{\theta_2 \theta_3} + \theta_{y_2}^2} \right] $$

$$ T_{3,4} = \frac{1}{2} \left[ -\sqrt{-\theta_1} \pm \sqrt{-\theta_2 - \theta_3 + 2\Sigma \sqrt{\theta_2 \theta_3} + \theta_{y_2}^2} \right] $$

where $\theta_{x_2} \theta_{x_3} + \theta_{y_2}^2 \geq 0$.

$$ Z_n = T_n - C, \quad n = 1, 2, 3, 4 $$
Checking the Solutions
The set of calculated solutions $Z_1$, $Z_2$, $Z_3$ and $Z_4$ of the quartic equation can be checked against the requirement that

$$Z^4 + A_3Z^3 + A_2Z^2 + A_1Z + A_0 = (Z-Z_1)(Z-Z_2)(Z-Z_3)(Z-Z_4) \text{ for all } Z.$$  

Expand and simplify the right side of this equation, and then equate each coefficient to the corresponding coefficient on the left side to obtain

$$A_3 = -(Z_1+Z_2+Z_3+Z_4)$$
$$A_2 = Z_1Z_2+Z_1Z_3+Z_1Z_4+Z_2Z_3+Z_2Z_4+Z_3Z_4$$
$$A_1 = -(Z_1Z_2Z_3+Z_1Z_2Z_4+Z_1Z_3Z_4+Z_2Z_3Z_4)$$
$$A_0 = Z_1Z_2Z_3Z_4.$$  

Express each $Z_n$ as the sum of its real and imaginary components: $Z_n = X_n + iY_n$. Solutions $Z_1$ and $Z_2$ are either real ($Y_1 = Y_2 = 0$) or they form a complex conjugate pair ($X_1 = X_2$, $Y_1 = -Y_2 > 0$). Solutions $Z_3$ and $Z_4$ are either real ($Y_3 = Y_4 = 0$) or they form a complex conjugate pair ($X_3 = X_4$, $Y_3 = -Y_4 > 0$). We now have:

$$A_3 = -(X_1+X_2+X_3+X_4)$$
$$A_2 = X_1X_2+Y_1^2 + (X_1+X_2)(X_3+X_4) + X_3X_4+Y_3^2$$
$$A_1 = -(X_1X_2+Y_1^2)(X_3+X_4) + (X_3X_4+Y_3^2)(X_1+X_2)$$
$$A_0 = (X_1X_2+Y_1^2)(X_3X_4+Y_3^2).$$

Valid solutions must reproduce the input coefficients according to these check equations.
PART II -- Algorithm Assessment
Algorithm Suitability for General Calculation

Each of the modified algorithms presented here is suitable for general calculation. Each provides:

- stable analytic solutions for any combination of real coefficients,
- formulas that convert easily to code, and
- calculation with real numbers only.

The choice of one of these algorithms over the others is therefore a matter of personal preference.

For introducing students to quartic equations, instructors may prefer Ferrari’s method. It is the earliest analytic method, and its derivation relies primarily on the technique of completing the square with which the students should be familiar. Students are thus likely to find its derivation the easiest to understand and remember. Moreover, the same derivation produces both the common algorithm and the modified algorithm.

The NBS modified algorithm has the advantage of not requiring the intermediate depressed quartic equation.

The Euler and DLMF original algorithms are also suitable for users who have the ability to perform operations on complex numbers.

The Ferrari common algorithm and the Descartes original algorithm are not recommended for general calculation because they can become computationally unstable. Even so, they are mathematically useful. In the section below, the Ferrari common algorithm helps demonstrate the mathematical equivalence of the NBS algorithm to the other algorithms presented here. Heikkinen[10] uses the unstable Descartes original algorithm to derive his stable algorithm for calculating a position’s geodetic coordinates (longitude, latitude, altitude) given the position’s earth-centered, earth-fixed rectangular coordinates on the earth ellipsoid.

The NBS original and modified algorithms are not two different algorithms in the same way that the Ferrari common and modified algorithms are different from each other. The NBS original is not even an algorithm in the sense of a defined sequence of logical and mathematical operations. It is rather a set of detailed requirements. An algorithm which meets all of the requirements can solve the general quartic equation correctly. The modified version is a true algorithm that meets all of the requirements.

Mathematical Equivalence of the Algorithms

Although the algorithms here vary in their suitability for general calculation, they are all mathematically equivalent to each other. This of course must be true because they all produce the correct quartic-equation solutions in theory. This section demonstrates equivalence by showing that any of the algorithms presented can be converted to any of the others. The NBS original algorithm is excluded because it is not a true algorithm, as just pointed out. From this point on, the term “NBS algorithm” refers to the NBS modified algorithm.
The NBS algorithm requires special treatment because it is the only algorithm considered here that solves the general quartic equation directly. We refer to the other algorithms as the depressed algorithms because they solve the equivalent depressed quartic equation and then shift the results to obtain solutions of the general quartic equation. Our demonstration first shows the mathematical equivalence of all of the depressed algorithms and then addresses their equivalence to the NBS algorithm.

The Depressed Algorithm Summary Table below lists the resolvent cubic equations and \( T_n \) formulas for all of the depressed algorithms.

**Principal-Square-Root Convention for Radicals**

The table uses the principal-square-root convention for radicals, so the \( T_n \) formulas in the Euler and DLMF original algorithms are recast accordingly. This is accomplished in the Euler original algorithm by replacing \( \sqrt{r_3} \) with \(-\Sigma s\sqrt{r_3}\) where

\[
\Sigma = \begin{cases} 
1 & \text{if } b_1 > 0 \\
-1 & \text{otherwise} 
\end{cases} \quad \text{and} \quad s = \begin{cases} 
1 & \text{if } \sqrt{r_1}\sqrt{r_2}\sqrt{r_3} \geq 0 \\
-1 & \text{otherwise} 
\end{cases}.
\]

The definitions of these special functions and equation (3) above imply that

\[
b_1 = \Sigma |b_1| \quad \text{and} \quad \sqrt{r_1}\sqrt{r_2}\sqrt{r_3} = s|\sqrt{r_1}\sqrt{r_2}\sqrt{r_3}| = s|b_1|/8.
\]

The Euler original \( T_n \) formulas change

FROM: \( T_{1,2} = \sqrt{r_1} \pm (\sqrt{r_2} + \sqrt{r_3}) \quad \text{and} \quad T_{3,4} = -\sqrt{r_1} \pm (\sqrt{r_2} - \sqrt{r_3}) \)

TO: \( T_{1,2} = \sqrt{r_1} \pm (\sqrt{r_2} - \Sigma s \sqrt{r_3}) \quad \text{and} \quad T_{3,4} = -\sqrt{r_1} \pm (\sqrt{r_2} + \Sigma s \sqrt{r_3}). \) (7)

In this revised formulation, the product of terms for each \( T_n \) is

\[-\Sigma s\sqrt{r_1}\sqrt{r_2}\sqrt{r_3} = -\Sigma s^2|b_1|/8 = -\Sigma |b_1|/8 = -b_1/8\]

as required by the Euler original algorithm. The DLMF original algorithm is recast in a corresponding fashion.

The function \( s \) in (6) accommodates the condition \( \sqrt{r_1}\sqrt{r_2}\sqrt{r_3} < 0 \), which occurs when one of the \( r_k \), say \( r_1 \), is positive real and the other two \( r_k \) are negative real:

\[
\sqrt{r_2} = i\sqrt{|r_2|}, \quad \sqrt{r_3} = i\sqrt{|r_3|} \quad \Rightarrow \quad \sqrt{r_2}\sqrt{r_3} = -\sqrt{|r_2|\sqrt{|r_3|}} = -\sqrt{r_2 r_3} < 0.
\]

For the Euler and DLMF modified algorithms, the table uses the simplified \( T_n \) formulas from (4) and (5).

**Greatest Real Solution of the Resolvent Cubic Equations**

The algorithm summary table applies the following additional convention. For calculating the \( T_n \), each algorithm except DLMF employs the greatest real solution of its resolvent cubic equation. This solution is \( m \) in Ferrari, \( y^2 \) in Descartes, and \( r_1 \) in Euler. DLMF uses the least real solution \( \theta_1 \) of its resolvent cubic equation. This convention allows a direct comparison of all the algorithms.
### DEPRESSED ALGORITHM SUMMARY TABLE

Given real coefficients $b_2$, $b_1$, and $b_0$, find $T_1$, $T_2$, $T_3$, and $T_4$ such that

$$(T - T_1)(T - T_2)(T - T_3)(T - T_4) = T^4 + b_2T^2 + b_1T + b_0$$

for all $T$.

**Definition:** $\Sigma = 1$ if $b_1 > 0$, $\Sigma = -1$ otherwise.

Radical $\sqrt{}$ denotes the principal square root for all algorithms.

<table>
<thead>
<tr>
<th>Method</th>
<th>Common Algorithm ($m &gt; 0$) *</th>
<th>Modified Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ferrari’s Method</strong></td>
<td>$T_{1,2} = \sqrt{m/2} \pm \sqrt{-m/2 + b_2/2 + b_1/(2\sqrt{2m})}$</td>
<td>$T_{1,2} = \Sigma \sqrt{m^2 + b_2m + b_2^2/4 - b_0}$</td>
</tr>
<tr>
<td></td>
<td>$T_{3,4} = -\sqrt{m/2} \pm \sqrt{-m/2 - b_2/2 + b_1/(2\sqrt{2m})}$</td>
<td>$T_{3,4} = -\sqrt{m/2} \pm \sqrt{-m/2 - b_2/2 + b_0}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Descartes’ Method</strong></th>
<th>$y^2$ is greatest real solution of: $y^4 + 2b_2y^2 + (b_2^2 - 4b_0)y^2 - b_2^2 = 0$, and $y \geq 0$.</th>
<th>$r_1$ is the greatest real solution of: $r_1^4 + (b_2/2)r_1^2 + [(b_2^2 - 4b_0)/16]r_1 - b_2^2/64 = 0$, and $r_1 \geq 0$.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$T_{1,2} = y/2 \pm \sqrt{-y^2/4 - b_2/2 - b_1/(2y)}$</td>
<td>$r_1^4 + (b_2/2)r_1^2 + [(b_2^2 - 4b_0)/16]r_1 - b_2^2/64 = 0$, and $r_1 \geq 0$.</td>
</tr>
<tr>
<td></td>
<td>$T_{3,4} = -y/2 \pm \sqrt{-y^2/4 - b_2/2 + b_1/(2y)}$</td>
<td>$s = 1$ if $\sqrt{r_1}\sqrt{r_2}\sqrt{r_3} \geq 0$, $s = -1$ otherwise.</td>
</tr>
</tbody>
</table>

| **Euler’s Method**    | $\theta_1$ is the least real solution of: $\theta_1^8 - 2b_2\theta_1^6 + (b_2^2 - 4b_0)\theta_1 + b_2^2 = 0$, and $-\theta_1 \geq 0$. | $\theta_1$ is the least real solution of: $\theta_1^8 - 2b_2\theta_1^6 + (b_2^2 - 4b_0)\theta_1 + b_2^2 = 0$, and $-\theta_1 \geq 0$. |
|                      | $T_{1,2} = \frac{1}{2} \left[ \sqrt{-\theta_1} \pm \left( \sqrt{-\theta_2} - \Sigma \sqrt{-\theta_3} \right) \right]$ | $T_{1,2} = \frac{1}{2} \left[ \sqrt{-\theta_1} \pm \left( \sqrt{-\theta_2} - \Sigma \sqrt{-\theta_3} \right) \right]$ |
|                      | $T_{3,4} = \frac{1}{2} \left[ -\sqrt{-\theta_1} \pm \left( \sqrt{-\theta_2} + \Sigma \sqrt{-\theta_3} \right) \right]$ | $T_{3,4} = \frac{1}{2} \left[ -\sqrt{-\theta_1} \pm \left( \sqrt{-\theta_2} + \Sigma \sqrt{-\theta_3} \right) \right]$ |
|                      | $s = 1$ if $\sqrt{-\theta_1}\sqrt{-\theta_2}\sqrt{-\theta_3} \geq 0$, $s = -1$ otherwise. | $s = 1$ if $\sqrt{-\theta_1}\sqrt{-\theta_2}\sqrt{-\theta_3} \geq 0$, $s = -1$ otherwise. |

$*$ $T_n$ formulas for the Ferrari common algorithm and the Descartes original algorithm for the case $m= y^2 = 0$ are

$T_{1,2} = \pm \sqrt{-b_2/2 + b_2^2/4 - b_0}$ $T_{3,4} = \pm \sqrt{-b_2/2 - b_2^2/4 - b_0}$.
Resolvent Cubic Equations in the Depressed Algorithms
In the table above, each resolvent cubic equation converts to every other by applying the transformation
\[ 2m = y^2 = 4r_k = -\theta_k \]
and simplifying to standard form. The greatest real solutions in Ferrari, Descartes, and Euler and the least real solution in DLMF are related by
\[ 2m = y^2 = 4r_1 = -\theta_1. \] (8)
The algorithms use these values to calculate the \( T_n \).

The greatest real solution \( m \) of the Ferrari resolvent cubic equation is nonnegative, as we now demonstrate. The resolvent cubic equation is
\[ m^3 + b_2 m^2 + (b_2^2/4 - b_0)m - b_1^2/8 = 0. \]
The constant coefficient, \(-b_1^2/8\), is less than or equal to zero. If \( b_1 = 0 \), then \( m = 0 \) is a solution. Otherwise, the cubic on the left side of the equation is negative at \( m = 0 \), but the cubic must eventually increase to zero as \( m \) increases to a sufficiently large positive value. This \( m \) value is a positive real solution. Thus \( m \geq 0 \), and by (8) we have
\[ 2m = y^2 = 4r_1 = -\theta_1 \geq 0. \] (9)
These values that are used to calculate the \( T_n \) are all nonnegative.

\( T_n \) Formulas for Ferrari and Descartes Algorithms
Convert the Ferrari common \( T_n \) formulas for \( m > 0 \) to the Ferrari modified \( T_n \) formulas by solving the resolvent cubic equation for \( b_1^2/(8m) \) and taking the square root.
\[ m^3 + b_2 m^2 + (b_2^2/4 - b_0)m - b_1^2/8 = 0 \Rightarrow b_1^2/(8m) = m^2 + b_2 m + b_2^2/4 - b_0 \]
\[ \Rightarrow \quad b_1/(2\sqrt{2m}) = \Sigma \sqrt{m^2 + b_2 m + b_2^2/4 - b_0} = R \quad (m > 0) \]
where \( \Sigma = 1 \) if \( b_1 > 0 \), \( \Sigma = -1 \) otherwise
The term \( b_1/(2\sqrt{2m}) \) in the common \( T_n \) formulas is replaced with \( R \) to produce the Ferrari modified \( T_n \) formulas.

The case \( m = 0 \) implies that \( b_1 = 0, \Sigma = -1, \) and \( R = -\sqrt{b_2^2/4 - b_0} \). The Ferrari modified algorithm produces the same formula for \( T_n \) as the Ferrari common algorithm for \( m = 0 \). The Ferrari common and modified algorithms are therefore mathematically equivalent to each other for all \( m \).

The transform \( m = y^2/2 \) from (9) converts the Ferrari common and modified \( T_n \) formulas to the corresponding formulas in the Descartes algorithms. Thus, the Ferrari and Descartes algorithms convert to each other and are all equivalent to each other.

\( T_n \) Formulas for Euler and DLMF Algorithms
Using the principal-square-root convention for radicals, the \( T_n \) formulas for the Euler original algorithm are
\[ T_{12} = \sqrt{r_1} \pm \left( \sqrt{r_2} - \Sigma \sqrt{r_3} \right) \quad T_{34} = -\sqrt{r_1} \pm \left( \sqrt{r_2} + \Sigma \sqrt{r_3} \right) \]  

where

\[ \Sigma = \begin{cases} 1 & \text{if } b_1 > 0 \\ -1 & \text{otherwise} \end{cases} \quad \text{and} \quad s = \begin{cases} 1 & \text{if } \sqrt{r_1}\sqrt{r_2}\sqrt{r_3} \geq 0 \\ -1 & \text{otherwise} \end{cases}, \]

and where \( r_1, r_2, \) and \( r_3 \) are solutions of the resolvent cubic equation

\[ r_k^3 + (b_2/2) r_k^2 + [(b_2^2 - 4b_0)/16] r_k - b_1^2/64 = 0. \]

By (9), the greatest real solution \( r_1 \) is nonnegative, as is \( \sqrt{r_1} \):

\[ r_1 \geq 0 \quad \Rightarrow \quad \sqrt{r_1} \geq 0. \]

Equation (3) shows that \( r_1 r_2 r_3 = b_1^2/64 \geq 0 \). Therefore,

\[ r_1 \geq 0 \quad \text{and} \quad r_1 r_2 r_3 = b_1^2/64 \geq 0 \quad \Rightarrow \quad r_2 r_3 \geq 0. \]

The product \( r_2 r_3 \) is a nonnegative real number. Thus \( r_2 = x_2 + iy_2 \) and \( r_3 = x_3 + iy_3 \) are real \( y_2 = y_3 = 0 \), or they form a complex conjugate pair \( x_2 = x_3, y_2 = -y_3 \). If real, then they cannot have opposite signs. This restriction on \( r_2 \) and \( r_3 \) implies that each parenthetical expression in (10) is either real or pure imaginary.

Conversion of the \( T_n \) formulas in (10) to those in the Euler modified algorithm starts by replacing each parenthetical expression with the radical of its square:

\[ T_{12} = \sqrt{r_1} \pm \sqrt{r_2 + r_3 - 2\Sigma \sqrt{r_2 r_3}} \quad T_{34} = -\sqrt{r_1} \pm \sqrt{r_2 + r_3 + 2\Sigma \sqrt{r_2 r_3}}. \]

\( T_1 \) and \( T_2 \) in (14) each have the same value as in (10) unless \( \sqrt{r_2 - \Sigma \sqrt{r_3} \} \) happens to be either negative real or negative imaginary. In that case, \( T_1 \) in (10) is \( T_2 \) in (14) and \( T_2 \) in (10) is \( T_1 \) in (14). \( T_3 \) and \( T_4 \) are correspondingly affected by the value of \( \sqrt{r_2 + \Sigma \sqrt{r_3} \}. \)

As an option to prevent the \( T_n \) from flipping values between (10) and (14), use the following convention: select \( r_2 = x_2 + iy_2 \) and \( r_3 = x_3 + iy_3 \) so that \( |x_2| \geq |x_3| \) and \( y_2 = -y_3 \geq 0 \). The convention assures that the parenthetical expressions in (10) are either nonnegative real or nonnegative imaginary. The convention does not affect (14), which is symmetrical with respect to \( r_2 \) and \( r_3 \).

Equation (12) implies that the formula for \( s \) in (11) simplifies to

\[ s = \begin{cases} 1 & \text{if } \sqrt{r_2}\sqrt{r_3} \geq 0 \\ -1 & \text{otherwise} \end{cases} \quad \Rightarrow \quad \sqrt{r_2\sqrt{r_3}} = s \sqrt{r_2 r_3} = s \sqrt{\sqrt{r_2 r_3}}. \]

This result and (13) imply that

\[ s \sqrt{r_2\sqrt{r_3}} = s^2 \sqrt{r_2 r_3} = \sqrt{r_2 r_3} = \sqrt{r_2 r_3}. \]

The \( T_n \) formulas in (14) convert to the simplified \( T_n \) formulas in the Euler modified algorithm.

\[ T_{12} = \sqrt{r_1} \pm \sqrt{r_2 + r_3 - 2\Sigma \sqrt{r_2 r_3}} \quad T_{34} = -\sqrt{r_1} \pm \sqrt{r_2 + r_3 + 2\Sigma \sqrt{r_2 r_3}}. \]
The Euler original and modified algorithms are therefore mathematically equivalent.

The transform \( r_k = -\theta_k/4 \) converts the original and modified Euler \( T_n \) formulas to the corresponding formulas in the DLMF algorithms. Thus, the Euler and DLMF algorithms are all equivalent to each other.

**Equivalence of \( T_n \) Formulas from All Eight Depressed Algorithms**

So far, we have the eight depressed algorithms grouped into two sets of equivalent algorithms:

Set 1: Ferrari common and modified, Descartes original and modified

Set 2: Euler original and modified, and DLMF original and modified.

We show that the two sets are equivalent by converting the Euler modified \( T_n \) formulas from Set 2 to the Ferrari modified \( T_n \) formulas from Set 1. Solutions \( r_k \) of the Euler resolvent cubic equation must satisfy the requirement:

\[
(r - r_1) (r - r_2) (r - r_3) = r^3 + (b_2/2)r^2 + [(b_2^2 - 4b_0)/16] r - b_2^2/64 \quad \text{for all } r.
\]

Expand and simplify the left side, then equate coefficients with the corresponding coefficients on the right. Results for the quadratic and linear coefficients are:

\[
-(r_1 + r_2 + r_3) = b_2/2 \quad \quad r_1r_2 + r_1r_3 + r_2r_3 = (b_2^2 - 4b_0)/16.
\]

Solve these two equations for \( r_2 + r_3 \) and for \( r_2r_3 \) as functions of \( r_1 \).

\[
r_2 + r_3 = -r_1 - b_2/2 \quad \quad r_2r_3 = r_1^2 + (b_2/2)r_1 + (b_2^2 - 4b_0)/16.
\]

Substitute these two expressions into the Euler modified \( T_n \) equations and simplify.

\[
T_{1,2} = \sqrt{r_1} \pm \sqrt{-r_1 - b_2/2 - \Sigma \sqrt{4r_1^2 + 2b_2 r_1 + b_2^2/4 - b_0}}
\]

\[
T_{3,4} = -\sqrt{r_1} \pm \sqrt{-r_1 - b_2/2 + \Sigma \sqrt{4r_1^2 + 2b_2 r_1 + b_2^2/4 - b_0}}
\]

Apply the transform \( r_1 = m/2 \) from (9) to produce the Ferrari modified \( T_n \) formulas:

\[
T_{1,2} = \sqrt{m/2} \pm \sqrt{-m/2 - b_2/2 - R} \quad \quad T_{3,4} = -\sqrt{m/2} \pm \sqrt{-m/2 - b_2/2 + R}
\]

where \( R = \Sigma \sqrt{m^2 + b_2m + b_2^2/4 - b_0} \).

The Euler modified algorithm from Set 2 converts to the Ferrari modified algorithm from Set 1. Thus, all eight depressed algorithms comprising the two sets are mathematically equivalent to each other.

**Equivalence of the NBS Algorithm to the Depressed Algorithms**

The eight depressed algorithms are also mathematically equivalent to the NBS algorithm as now demonstrated by constructing the NBS algorithm from the Ferrari algorithms.
Part II -- Algorithm Assessment

The NBS algorithm is related to the Ferrari algorithms through \(u_1\), the greatest real solution of the NBS resolvent cubic equation. The NBS algorithm provides \(u_1\) in terms of \(Z_1, Z_2, Z_3,\) and \(Z_4\) as follows. The formulas for \(q_1\) and \(q_2\) show that \(q_1 + q_2 = u_1\), and the \(Z_n\) formulas show that \(Z_1Z_2 = q_1\) and \(Z_3Z_4 = q_2\). This demonstration therefore defines \(u_1\) as

\[
u_1 = Z_1Z_2 + Z_3Z_4.\tag{16}\]

The Ferrari starting point includes 1) the calculation formulas for \(C, b_2, b_1,\) and \(b_0,\) 2) the \(Z_n\)-to-\(T_n\) transform, 3) the resolvent cubic equation, and 4) the \(T_n\) formulas:

\[
C = A_3/4, \quad b_2 = A_2 - 6C^2, \quad b_1 = A_1 - 2A_2C + 8C^3, \quad b_0 = A_0 - A_1C + A_2C^2 - 3C^4, \tag{17}\]

\[
Z_n = T_n - C \iff T_n = Z_n + C, \quad n = 1, 2, 3, 4,
\]

\[
m^3 + b_2 m^2 + (b_2^2/4 - b_0)m - b_2^2/8 = 0, \tag{18}\]

\[
T_{1,2} = \sqrt{m/2} \pm \sqrt{-m/2 - b_2/2 - R}, \tag{19}\]

\[
T_{3,4} = -\sqrt{m/2} \pm \sqrt{-m/2 - b_2/2 + R}, \tag{20}\]

where

\[
R = \frac{\Sigma \sqrt{m^2 + b_2m + b_2^2/4 - b_0}}{m > 0}, \tag{21}\]

\[
\Sigma = 1 \text{ if } b_1 > 0, \quad \Sigma = -1 \text{ otherwise}, \tag{23}\]

and where \(m \geq 0\) is the greatest real solution of (18).

Equations (19) and (20) provide the following preliminary relationships.

\[
T_1 + T_2 = \sqrt{2m} \tag{24}\]

\[
T_3 + T_4 = -\sqrt{2m} \tag{25}\]

\[
T_1T_2 = m + b_2/2 + R \tag{26}\]

\[
T_3T_4 = m + b_2/2 - R \tag{27}\]

Relate \(u_1\) to \(m\) by substituting \(Z_n = T_n - C\) and (24) through (27) into (16) as follows.

\[
u_1 = (T_1 - C)(T_2 - C) + (T_3 - C)(T_4 - C) = T_1T_2 + T_3T_4 - (T_1 + T_2 + T_3 + T_4)C + 2C^2
\]

\[
u_1 = 2m + b_2 + 2C^2 \iff 2m = u_1 - b_2 - 2C^2 \tag{28}\]

Convert the resolvent cubic equation of Ferrari to that of NBS as follows. Multiply (18) through by 8 and write as

\[
(2m)^3 + 2b_2 (2m)^2 + (b_2^2/4)2m) - b_1^2 = 0.
\]

From (28), replace \(2m\) with \(u - b_2 - 2C^2\). Substitute the expressions in (17) for \(C, b_2, b_1,\) and \(b_0\). Then simplify to obtain the NBS resolvent cubic equation.

\[
u^3 - A_2u^2 + (A_1A_3 - 4A_0)u + 4A_0A_2 - A_1^2 - A_0A_3^2 = 0 \tag{29}\]
Construct the NBS $Z_n$ formulas from the Ferrari $T_n$ formulas as follows. Calculate the quantities below on the left by using (24) through (27).

$$(T_1 + T_2)^2/4 - T_1 T_2 = -m/2 - b_2/2 - R$$
$$(T_3 + T_4)^2/4 - T_3 T_4 = -m/2 - b_2/2 + R$$

The expressions on the right are the radicands in the $T_n$ formulas (19) and (20). These $T_n$ formulas thus have the form:

$$T_{1,2} = (T_1 + T_2)/2 \pm \sqrt{(T_1 + T_2)^2/4 - T_1 T_2}$$
$$T_{3,4} = (T_3 + T_4)/2 \pm \sqrt{(T_3 + T_4)^2/4 - T_3 T_4}.$$ 

Obtain the corresponding formulas for the $Z_n$ by substituting $T_n = Z_n + C$ and then simplifying.

$$Z_{1,2} = (Z_1 + Z_2)/2 \pm \sqrt{(Z_1 + Z_2)^2/4 - Z_1 Z_2}$$
$$Z_{3,4} = (Z_3 + Z_4)/2 \pm \sqrt{(Z_3 + Z_4)^2/4 - Z_3 Z_4}$$

These become the NBS $Z_n$ formulas,

$$Z_{1,2} = -p_1/2 \pm \sqrt{p_1^2/4 - q_1}, \quad Z_{3,4} = -p_2/2 \pm \sqrt{p_2^2/4 - q_2} \quad (30)$$

where

$$p_1 = -(Z_1 + Z_2) \quad p_2 = -(Z_3 + Z_4) \quad (31)$$

and

$$q_1 = Z_1 Z_2 \quad q_2 = Z_3 Z_4. \quad (32)$$

Find the NBS calculation formulas for $p_1$ and $p_2$ by substituting $T_n - C$ for the $Z_n$ in (31), and then apply (24) or (25), (28), and (17).

$$p_1 = -(Z_1 + Z_2) = 2C - (T_1 + T_2) = 2C - \sqrt{2m} = 2C - \sqrt{u_1 - b_2 - 2C}$$

$$p_1 = A_3/2 - \sqrt{A_3^2/4 + u_1 - A_2} \quad p_2 = A_3/2 + \sqrt{A_3^2/4 + u_1 - A_2} \quad (33)$$

To find the NBS calculation formulas for $q_1$ and $q_2$, start by substituting $T_n - C$ for the $Z_n$ in (32), and then apply (24) through (28).

$$q_1 = Z_1 Z_2 = (T_1 - C)(T_2 - C) = T_1 T_2 - (T_1 + T_2)C + C^2 = m + b_2/2 + R - C\sqrt{2m} + C^2$$
$$q_2 = Z_3 Z_4 = (T_3 - C)(T_4 - C) = T_3 T_4 - (T_3 + T_4)C + C^2 = m + b_2/2 - R + C\sqrt{2m} + C^2$$

or

$$q_{1,2} = m + b_2/2 + C^2 \pm (R - C\sqrt{2m}) \quad (34)$$

For the case $m > 0$, apply the formula $R = b_1/(2\sqrt{2m})$ from (22). Then apply (28) for 2m.

$$q_{1,2} = m + b_2/2 + C^2 \pm \frac{b_1 - 4Cm}{2\sqrt{2m}} = u_1/2 \pm \frac{b_1 - 2Cu_1 + 2Cb_2 + 4C^3}{\sqrt{4(u_1 - b_2 - 2C^2)}} \quad (m > 0)$$

Apply (17) for $C$, $b_1$, and $b_2$. 

6/17/2020
The expression
\[ q_{1,2} = \frac{u_1}{2} \pm \frac{A_1 - A_3 u_1 / 2}{\sqrt{4(u_1 - A_2 + A_3^2 / 4)}} \] (m > 0)

Use the NBS definition
\[ \Sigma_g = \begin{cases} 
1 & \text{if } A_1 - A_3 u / 2 > 0 \\
-1 & \text{otherwise}
\end{cases} \] (35)

so that
\[ A_1 - A_3 u_1 / 2 = \Sigma_g |A_1 - A_3 u_1 / 2| = \Sigma_g \sqrt{(A_1 - A_3 u_1 / 2)^2} \]

and \( q_{1,2} \) becomes
\[ q_{1,2} = \frac{u_1}{2} \pm \frac{\Sigma_g \sqrt{(A_1 - A_3 u_1 / 2)^2}}{\sqrt{4(u_1 - A_2 + A_3^2 / 4)}} = \frac{u_1}{2} \pm \Sigma_g \sqrt{N/D} \] (m > 0) (36)

where \( N = (A_1 - A_3 u_1 / 2)^2 \) and \( D = 4(u_1 - A_2 + A_3^2 / 4) \).

The quotient \( N/D \) is \( Q = u_1^2 / 4 - A_0 \) because \( DQ = N \), which fact we now demonstrate.
\[ DQ = 4(u_1 - A_2 + A_3^2 / 4)(u_1^2 / 4 - A_0) = (u_1 - A_2 + A_3^2 / 4)(u_1^2 - 4A_0) \]
\[ DQ = u_1^3 - A_1 A_3 u_1 + A_3^2 u_1^2 / 4 - 4A_0 u_1 + 4A_0 A_2 - A_0 A_3^2 \]

Subtract zero in the form of the left side of (29) with solution \( u_1 \) replacing \( u \). The expression for \( DQ \) becomes
\[ DQ = A_1^2 - A_1 A_3 u_1 + A_3^2 u_1^2 / 4 = (A_1 - A_3 u_1 / 2)^2 = N. \]

\( DQ = N \), so the quotient \( N/D \) is \( Q = u_1^2 / 4 - A_0 \). Thus, when \( m \) is greater than 0 the formulas for \( q_1 \) and \( q_2 \) in (36) become those of the NBS algorithm.
\[ q_{1,2} = \frac{u_1}{2} \pm \Sigma_g \sqrt{u_1^2 / 4 - A_0} \] (m > 0) (37)

The case \( m = 0 \) produces this same expression for \( q_{1,2} \) as now shown. Equations (34) and (21) give
\[ q_{1,2} = b_2 / 2 + C^2 \pm \Sigma \sqrt{b_2^2 / 4 - b_0} \] (m = 0). (38)

The resolvent cubic equation (18) implies that \( b_1 = 0 \), and from (23), \( \Sigma = -1 \). The expression for \( b_1 \) in (17) shows that
\[ A_1 = 2A_2 C - 8C^3 = (A_2 - A_3^2 / 4) A_3 / 2 \] (m = 0). (39)

This expression and those for \( C, b_2, \) and \( b_0 \) in (17) convert \( q_{1,2} \) in (38) to
\[ q_{1,2} = (A_2 - A_3^2 / 4) / 2 \pm (-1) \sqrt{(A_2 - A_3^2 / 4)^2 / 4 - A_0} \] (m = 0). (40)

Equations (17), (28), (39), and (35) produce the following.
\[ u_1 = b_2 + 2C^2 = A_2 - A_3^2 / 4, \quad A_1 - A_3 u_1 / 2 = 0, \quad \Sigma_g = -1 \] (m = 0)

The expression for \( q_{1,2} \) in (40) may thus take the form of (37).
\[ q_{1,2} = \frac{u_1}{2} \pm \Sigma_g \sqrt{u_1^2 / 4 - A_0} \] (m = 0) (41)
Together, (37) and (41) give the NBS $q_1$ and $q_2$ formulas for all $m \geq 0$.

$$q_1 = \frac{u_{1}}{2} + \sum g \sqrt{u_{1}^2 / 4 - A_0} \quad q_2 = \frac{u_{1}}{2} - \sum g \sqrt{u_{1}^2 / 4 - A_0}$$  \hspace{1cm} (42)

This completes the construction of the NBS algorithm from the Ferrari algorithms. The NBS algorithm finds $u_1$ as the greatest real solution of its resolvent cubic equation (29), and then solves (35), (33), (42), and (30) in succession to find solutions $Z_n$ of the general quartic equation. By constructing the NBS algorithm from the Ferrari algorithms, we have shown that the NBS algorithm is mathematically equivalent to the Ferrari algorithms. Because the Ferrari algorithms are mathematically equivalent to all of the algorithms described here, so is the NBS algorithm.
PART III -- Algorithm Derivations
Derivation 1: Depressed Quartic Equation

The algorithm inputs are four real coefficients $A_3, A_2, A_1,$ and $A_0,$ and the outputs are the four values $Z_1, Z_2, Z_3$ and $Z_4$ such that

$$Z^4 + A_3Z^3 + A_2Z^2 + A_1Z + A_0 = (Z-Z_1)(Z-Z_2)(Z-Z_3)(Z-Z_4) \text{ for all } Z.$$ 

The outputs are thus the four solutions of the general quartic equation

$$Z_n^4 + A_3Z_n^3 + A_2Z_n^2 + A_1Z_n + A_0 = 0, \quad n = 1, 2, 3, 4. \quad (1-1)$$

Except for the NBS method, the algorithms solve the equivalent depressed quartic equation

$$T_n^4 + b_2T_n^2 + b_1T_n + b_0 = 0, \quad n = 1, 2, 3, 4. \quad (1-2)$$

The first derivation applies a shift constant $C$ and the transform

$$Z_n = T_n - C \quad (1-3)$$

to convert (1-1) to (1-2). Substitute (1-3) into (1-1). Expand and simplify to a quartic equation for $T_n$ in standard form. Then equate the resulting coefficients to the corresponding coefficients in (1-2). Solve for $C, b_2, b_1,$ and $b_0.$ The result is

$$C = A_3/4, \quad b_2 = A_2 - 6C^2, \quad b_1 = A_1 - 2A_2C + 8C^3, \quad b_0 = A_0 - A_1C + A_2C^2 - 3C^4. \quad (1-4)$$

Except for the NBS method, the algorithms calculate $C, b_2, b_1, \text{ and } b_0$ in (1-4), solve (1-2) for the $T_n,$ and apply (1-3) to compute solutions $Z_n$ of (1-1).

Derivation 2: Ferrari Algorithms

Starting with $b_2, b_1, \text{ and } b_0$ from (1-4) as given, Ferrari (as described by Cardano\cite{1, pp 237, 253}) finds the four solutions $T_n$ of the depressed quartic equation (1-2). Ferrari applies an adjustable parameter $m$ to convert (1-2) into the equality of two perfect squares

$$A^2 = B^2 \quad \Rightarrow \quad A^2 - B^2 = 0.$$ 

$A$ is quadratic and $B$ is linear in $T_n.$ This converted quartic equation factors into two easily-solved quadratic equations:

$$A - B = 0 \quad \text{ and } \quad A + B = 0.$$ 

The first step is to add $b_2^2/4 - b_1T_n - b_0$ to both sides of (1-2) to produce a perfect-square quartic on the left side.

$$\left(T_n^2 + b_2/2\right)^2 = b_2^2/4 - b_1T_n - b_0 \quad (2-1)$$

The left side remains a perfect square if $m$ is added to $T_n^2 + b_2/2$ inside the parentheses. Do this by adding $2m(T_n^2 + b_2/2) + m^2$ to both sides of (2-1). Then express the right side as a standard-form quadratic in $T_n.$

$$\left(T_n^2 + b_2/2 + m^2 \right) = 2mT_n^2 - b_1T_n + \left(m^2 + b_2m + b_2^2/4 - b_0\right) \quad (2-2)$$

This equation is valid for all values of $m.$
The quadratic on the right side of (2-2) is a perfect square if its discriminant is zero. (The discriminant of quadratic \( ax^2 + bx + c \) is \( b^2 - 4ac \).) Setting the discriminant to zero produces

\[
(-b_1)^2 - 4(2m)(m^2 + b_2m + b_2^2/4 - b_0) = -8m^3 - 8b_2m^2 - 2(b_2^2 - 4b_0)m + b_1^2 = 0.
\]

Divide through by \(-8\) to obtain Ferrari’s resolvent cubic equation:

\[
m^3 + b_2m^2 + (b_2^2/4 - b_0)m - b_1^2/8 = 0. \tag{2-3}
\]

Any solution \( m \) of (2-3) makes the right side of (2-2) a perfect square \( B^2 \). To avoid complex-number operations, choose \( m \) as a nonnegative real solution.

\[
m \geq 0
\]

Such a solution always exists because the constant coefficient, \(-b_1^2/8\), is less than or equal to zero.

With the nonnegative real solution \( m \), equation (2-2) takes on the desired form \( A^2 = B^2 \):

\[
(T_n^2 + b_2/2 + m)^2 = (\sqrt{2m} T_n - R)^2 \tag{2-4}
\]

where

\[
A = T_n^2 + b_2/2 + m \quad \text{and} \quad B = \sqrt{2m} T_n - R \tag{2-5}
\]

\[
R^2 = m^2 + b_2m + b_2^2/4 - b_0, \tag{2-6}
\]

and

\[
2\sqrt{2m} R = b_1. \tag{2-7}
\]

Equations (2-6) and (2-7) provide two different ways to solve for \( R \). Equation (2-7) shows that \( R \) must have the same sign as \( b_1 \) if \( m > 0 \). If \( m = 0 \), then the sign of \( R \) is arbitrary in (2-4) through (2-7), but a negative \( R \) value will prove convenient later. We may therefore define the function

\[
\Sigma = \begin{cases} 
1 & \text{if } b_1 > 0 \\
-1 & \text{otherwise}
\end{cases} \tag{2-8}
\]

and write

\[
b_1 = \Sigma |b_1| \quad \text{and} \quad R = \Sigma |R|. \tag{2-9}
\]

Solving (2-7) for \( R \) produces

\[
R = b_1/(2\sqrt{2m}), \quad m > 0. \tag{2-10}
\]

Equations (2-6) and (2-9) imply that

\[
R = \Sigma \sqrt{m^2 + b_2m + b_2^2/4 - b_0}. \tag{2-11}
\]

For the case \( m > 0 \), (2-10) guarantees that \( R \) is a real number, so the radicand in (2-11) must be nonnegative.

\[
m^2 + b_2m + b_2^2/4 - b_0 \geq 0, \quad m > 0 \tag{2-12}
\]

With \( R \) from either (2-10) or (2-11), we proceed to factor quartic equation (2-4) into the two quadratic equations \( A - B = 0 \) and \( A + B = 0 \) using \( A \) and \( B \) from (2-5).
\[ T_n^2 - \sqrt{2m} T_n + b_2/2 + m + R = 0 \quad T_n^2 + \sqrt{2m} T_n + b_2/2 + m - R = 0 \] (2-13)

The solutions are:
\[ T_{1,2} = \sqrt{m/2} \pm \sqrt{-m/2 - b_2/2 - R} \quad T_{3,4} = -\sqrt{m/2} \pm \sqrt{-m/2 - b_2/2 + R} \] (2-14)

The transform \( Z_n = T_n - C \) then produces solutions of the general quartic equation.

\[
\begin{align*}
Z_{1,2} &= \sqrt{m/2} - C \pm \sqrt{-m/2 - b_2/2 - R} \\
Z_{3,4} &= -\sqrt{m/2} - C \pm \sqrt{-m/2 - b_2/2 + R}
\end{align*}
\] (Ferrari Modified Algorithm) (2-15)

These are the \( Z_n \) formulas in the Ferrari modified algorithm.

The Ferrari common \( Z_n \) formulas substitute (2-10) for \( R \) into this result for the case \( m > 0 \).

\[
\begin{align*}
Z_{1,2} &= \sqrt{m/2} - C \pm \sqrt{-m/2 - b_2/2 - b_1/(2\sqrt{2m})} \\
Z_{3,4} &= -\sqrt{m/2} - C \pm \sqrt{-m/2 - b_2/2 + b_1/(2\sqrt{2m})}
\end{align*}
\] (Ferrari Common Algorithm, \( m > 0 \)) (2-16)

Consider now the case \( m = 0 \). Equation (2-7) implies that \( b_1 \) must also be zero. The depressed quartic equation (1-2) becomes a quadratic equation in \( T_n^2 \) with solutions
\[ T_n^2 = -b_2/2 \pm \sqrt{b_2^2/4 - b_0}. \]

Take the square root of both sides and apply \( Z_n = T_n - C \) to obtain the Ferrari common \( Z_n \) formulas for the case \( m = 0 \).

\[
\begin{align*}
Z_{1,2} &= -C \pm \sqrt{-b_2/2 + \sqrt{b_2^2/4 - b_0}} \\
Z_{3,4} &= -C \pm \sqrt{-b_2/2 - \sqrt{b_2^2/4 - b_0}}
\end{align*}
\] (Ferrari Common Algorithm, \( m = 0 \)) (2-17)

These formulas match the results of the Ferrari modified \( Z_n \) formulas in (2-15) for \( m = 0 \). Equations (2-7), (2-8), and (2-11) show that \( m = 0 \Rightarrow b_1 = 0 \Rightarrow \Sigma = -1 \Rightarrow R = -\sqrt{b_2^2/4 - b_0} \Rightarrow Z_n \) formulas in (2-15) for \( m = 0 \) match those in (2-17).

If the inner integrand \( b_2^2/4 - b_0 \) in (2-17) is less than zero, then the \( Z_n \) calculation requires taking square roots of complex numbers. The following paragraph demonstrates that this situation is avoided by using (2-17) only if the greatest real solution of the resolvent cubic equation is \( m = 0 \). If some other real solution \( m > 0 \) exists, then it should be used in (2-15) or (2-16) to calculate the \( Z_n \).
Consider the case that \( m = 0 \) is the greatest real solution of the resolvent cubic equation (2-3). A solution \( m = 0 \) implies that \( b_1 = 0 \). The resolvent cubic equation (2-3) is therefore

\[
m^3 + b_2m^2 + (b_2^2/4 - b_0)m = m(m^2 + b_2m + b_2^2/4 - b_0) = 0
\]

with solutions 0 and \(-b_2/2 \pm \sqrt{b_0}\). If \( b_0 \geq 0 \), then these last two solutions are real and, by assumption, less than or equal to zero.

If \( b_0 \geq 0 \), then \(-b_2/2 + \sqrt{b_0} \leq 0 \) \( \Rightarrow \sqrt{b_0} \leq b_2/2 \) \( \Rightarrow b_0 \leq b_2^2/4 \) \( \Rightarrow b_2^2/4 - b_0 \geq 0 \).

Otherwise, \( b_0 \) is negative, which implies that \( b_2^2/4 - b_0 \geq 0 \). Therefore,

\[
b_2^2/4 - b_0 \geq 0 \text{ provided no real } m > 0 \text{ exists.} \quad (2-18)
\]

This result assures that (2-17) in the common algorithm operates on real numbers only when the greatest real solution of the resolvent cubic equation (2-3) is \( m = 0 \).

The modified algorithm applies (2-18) as well. The radicand of \( R \) in (2-11) is \( b_2^2/4 - b_0 \) when \( m = 0 \). Therefore (2-11), (2-12), and (2-18) together imply that the radicand of \( R \) is nonnegative provided that real \( m > 0 \) is used if it exists.

\[
m^2 + b_2m + b_2^2/4 - b_0 \geq 0 \text{ provided real } m > 0 \text{ is used if it exists.} \quad (2-19)
\]

This concludes the derivation. Its results are summarized as follows.

- Given coefficients \( b_2, b_1, \) and \( b_0 \) of the depressed quartic equation (1-2), solve the resolvent cubic equation (2-3). Use a real solution \( m > 0 \) if it exists. Otherwise, use \( m = 0 \).

- For the common algorithm, calculate the \( Z_n \) using (2-16) for \( m > 0 \) or (2-17) for \( m = 0 \). Inequality (2-18) assures that (2-17) operates on real numbers only.

- For the modified algorithm, calculate \( \Sigma \) using (2-8), \( R \) using (2-11), and the \( Z_n \) using (2-15). By using a real \( m > 0 \) if it exists, the inequality (2-19) assures that \( R \) is real and that (2-15) operates on real numbers only.

**Derivation 3: Descartes Algorithms**

The two versions of the Descartes algorithm are similar to the corresponding versions of the Ferrari algorithm. The Ferrari formulas for \( Z_n \) become the corresponding Descartes formulas by substituting \( y^2/2 \) for \( m \) and positive \( y \) for \( \sqrt{y^2} \). Substitute \( y^2/2 \) for \( m \) in the Ferrari resolvent cubic equation (2-3) and multiply through by 8 to obtain the Descartes resolvent cubic equation.

\[
y^6 + 2b_2 y^4 + (b_2^2 - 4b_0)y^2 - b_1^2 = 0 \quad (3-1)
\]

For the case \( m = y^2/2 > 0 \), the two quadratic equations in (2-13) and their solutions in (2-14) for Ferrari convert to those for Descartes by substituting (2-10) for \( R, y^2/2 \) for \( m, \) and positive \( y \) for \( \sqrt{y^2} \).
Part III -- Algorithm Derivations

\[ T_n^2 - yT_n + \left( y^2 + b_2 + b_1/y \right)/2 = 0 \]
\[ T_n^2 + yT_n + \left( y^2 + b_2 - b_1/y \right)/2 = 0 \quad \text{ (} y^2 > 0 \text{)} \quad (3-2) \]

\[ T_{1,2} = y/2 \pm \sqrt{-y^2/4 - b_2/2 - b_1/(2y)} \]
\[ T_{3,4} = -y/2 \pm \sqrt{-y^2/4 - b_2/2 + b_1/(2y)} \quad \text{ (} y^2 > 0 \text{)} \]

These \( T_n \) formulas give the solutions of the depressed quartic equation.

\[ T_n^4 + b_2 T_n^2 + b_1 T_n + b_0 = 0 \quad \text{ , } \quad n = 1, 2, 3, 4 \quad (3-3) \]

Rather than deriving the \( T_n \) solutions, Descartes simply asserts that the solutions \( T_n \) of the two quadratic equations (3-2) are the same as the four solutions of the depressed quartic equation (3-3) provided that \( y^2 \) is a positive real solution of the resolvent cubic equation (3-1).\[^{5, p 184} \]

The following derivation verifies Descartes’ assertion. Form the product of the two quadratic equations in (3-2).

\[ [T_n^2 - yT_n + \left( y^2 + b_2 + b_1/y \right)/2][T_n^2 + yT_n + \left( y^2 + b_2 - b_1/y \right)/2] = 0 \]

Expand and simplify to obtain

\[ T_n^4 + b_2 T_n^2 + b_1 T_n + (y^4 + 2b_2 y^2 + b_2^2 - b_1^2/y^2)/4 = 0 \quad (3-4) \]

where \( y^2 \) is a positive real solution of (3-1). Note that equations (3-3) and (3-4) differ only in the constant coefficients. Add \( 4b_0y^2 \) to both sides of the resolvent cubic equation (3-1), and then divide through by \( 4y^2 \).

\[ (y^4 + 2b_2 y^2 + b_2^2 - b_1^2/y^2)/4 = b_0 \]

This result equates the constant coefficients in (3-4) and (3-3). Thus (3-4), the product of the two quadratic equations in (3-2), is the depressed quartic equation (3-3). Therefore, the solutions of the two quadratic equations are the same as the four solutions of the depressed quartic equation provided that \( y^2 \) is a positive real solution of the resolvent cubic equation (3-1). Descartes’ assertion is verified.

**Derivation 4: NBS Modified Algorithm**

The algorithm inputs are four real coefficients \( A_3, A_2, A_1, \) and \( A_0 \), and the outputs are the four values \( Z_1, Z_2, Z_3, \) and \( Z_4 \) such that

\[ Z^4 + A_3 Z^3 + A_2 Z^2 + A_1 Z + A_0 = (Z-Z_1)(Z-Z_2)(Z-Z_3)(Z-Z_4) \] for all \( Z \). \quad (4-1) \]

The outputs are thus the four solutions of the general quartic equation

\[ Z_n^4 + A_3 Z_n^3 + A_2 Z_n^2 + A_1 Z_n + A_0 = 0. \quad (4-2) \]

The derivation comprises two sections. The first section derives the algorithm calculation equations. The second section shows that the algorithm should use the greatest real solution of the resolvent cubic equation in order to assure that the algorithm operates on real numbers only.
Calculation Equations

The NBS method expresses the left side of (4-1) as the product of two quadratics with real coefficients:

\[(Z^2 + p_1Z + q_1)(Z^2 + p_2Z + q_2) = Z^4 + A_3Z^3 + A_2Z^2 + A_1Z + A_0.\] (4-3)

Real values \(p_1, p_2, q_1,\) and \(q_2\) are calculated from the coefficients \(A_3, A_2, A_1,\) and \(A_0,\) and then the solutions of quartic equation are easily computed as the roots of the two quadratics.

\[
Z_{1,2} = -p_1/2 \pm \sqrt{p_1^2/4 - q_1} \quad Z_{3,4} = -p_2/2 \pm \sqrt{p_2^2/4 - q_2} \] (4-4)

The derivation of \(p_1, q_1, p_2,\) and \(q_2\) starts by expanding and simplifying the left side of (4-3).

\[Z^4 + (p_1+p_2)Z^3 + (p_1p_2+q_1+q_2)Z^2 + (p_1q_2+p_2q_1)Z + pq = Z^4 + A_3Z^3 + A_2Z^2 + A_1Z + A_0\]

Equate corresponding coefficients from the two sides to create the following system of equations.

\[
p_1 + p_2 = A_3 \quad \text{(4-5)}
\]
\[
p_1p_2 + q_1 + q_2 = A_2 \quad \text{(4-6)}
\]
\[
p_1q_2 + p_2q_1 = A_1 \quad \text{(4-7)}
\]
\[
q_1q_2 = A_0 \quad \text{(4-8)}
\]

Define \(u\) as \(q_1 + q_2,\) and solve (4-6) for \(p_1p_2.\)

\[
q_1 + q_2 = u \quad \text{(4-9)}
\]
\[
p_1p_2 = A_2 - u \quad \text{(4-10)}
\]

Apply the following fact to express \(p_1, p_2, q_1,\) and \(q_2\) as functions of \(u:\) given the sum \(S\) and product \(P\) of two unknowns \(x_1\) and \(x_2,\) the unknowns are found as solutions of the quadratic equation \(x^2 - Sx + P = 0, \quad m = 1, 2.\) Equations (4-5) and (4-10) give \(A_3\) and \(A_2 - u\) as the sum and product of \(p_1\) and \(p_2;\) (4-9) and (4-8) give \(u\) and \(A_0\) as the sum and product of \(q_1\) and \(q_2.\) Therefore, the quadratic equations for \(p_1\) and \(p_2\) and for \(q_1\) and \(q_2\) are

\[
p_m^2 - A_3p_m + A_2 - u = 0 \quad \text{and} \quad q_m^2 - uq_m + A_0 = 0, \quad m = 1, 2.
\]

Solve the first of these by assigning the negative radical to \(p_1.\)

\[
p_1 = A_3/2 - \sqrt{A_3^2/4 + u - A_2} \quad p_2 = A_3/2 + \sqrt{A_3^2/4 + u - A_2} \quad \text{(4-11)}
\]

We cannot yet say which of \(q_1\) or \(q_2\) gets the positive radical and which gets the negative. For now, let \(\Sigma_g\) have a value of either 1 or \(-1,\) and write

\[
q_1 = u/2 + \Sigma_g \sqrt{u^2/4 - A_0} \quad q_2 = u/2 - \Sigma_g \sqrt{u^2/4 - A_0} \quad \text{(4-12)}
\]

Equations (4-5) and (4-10) show that the radicand in (4-11) is nonnegative:

\[A_3^2/4 + u - A_2 = (p_1 + p_2)^2/4 - p_1p_2 = (p_1 - p_2)^2/4 \geq 0.\]
Equations (4-9) and (4-8) show that the radicand in (4-12) is nonnegative:
\[ u^2/4 - A_0 = (q_1 + q_2)^2/4 - q_1 q_2 = (q_1 - q_2)^2/4 \geq 0. \]

To find \( \Sigma_g \), substitute (4-11) and (4-12) into (4-7), and simplify to obtain
\[ 2 \Sigma_g \sqrt{(A_3^2/4 + u - A_2)(u^2/4 - A_0)} = A_1 - A_3 u/2. \]  
(4-13)

The radical is nonnegative, so \( \Sigma_g \) must correspond to the sign of the right side of the equation. We therefore express \( \Sigma_g \) as follows.

\[
\Sigma_g = \begin{cases} 
1 & \text{if } A_1 - A_3 u/2 > 0 \\
-1 & \text{otherwise}
\end{cases}
\]  
(4-14)

To obtain the resolvent cubic equation in \( u \), square both sides of (4-13) and simplify to a cubic equation in standard form:
\[ u^3 - A_2 u^2 + (A_1 A_3 - 4 A_0) u + 4 A_0 A_2 - A_1^2 - A_0 A_3^2 = 0. \]  
(4-15)

At least one solution, \( u_1 \), is real and is to be used for \( u \) wherever it occurs in the calculation formulas. If multiple solutions are real, then select the greatest real solution for \( u_1 \) to assure that calculations operate on real numbers only.

This completes derivation of the algorithm calculation equations. The results are summarized as follows.

- Given the real coefficients \( A_3, A_2, A_1, \) and \( A_0 \) of the general quartic equation (4-2), solve the resolvent cubic equation (4-15). Choose the greatest real solution as \( u_1 \), and apply it as the value of \( u \) to be used hereafter.
- Calculate \( \Sigma_g \) using (4-14).
- Calculate \( p_1, p_2, q_1, \) and \( q_2 \) using (4-11) and (4-12).
- Calculate the quartic-equation solutions \( Z_1, Z_2, Z_3 \) and \( Z_4 \) using (4-4).

**Greatest Real Solution of the Resolvent Cubic Equation**

This section shows that the algorithm should use the greatest real solution of the resolvent cubic equation in order to assure that the algorithm operates on real numbers only. In other words, we want to assure that solution \( u_1 \) of the resolvent cubic equation and the values \( p_1, p_2, q_1, \) and \( q_2 \) are real numbers.

The resolvent cubic equation (4-15) returns three solutions for \( u \), which is defined in (4-9) as \( q_1 + q_2 \). Quantities \( q_1 \) and \( q_2 \) are the constant coefficients in the two quadratics on the left side of (4-3). Therefore, each \( q_m \) is the product of the two roots of the corresponding quadratic. If \( Z_1 \) and \( Z_2 \) are roots of the first quadratic and \( Z_3 \) and \( Z_4 \) are roots of the second quadratic, then \( q_1 = Z_1 Z_2 \), \( q_2 = Z_3 Z_4 \), and \( u = Z_1 Z_2 + Z_3 Z_4 \). This value of \( u \) corresponds to only one of the possible pairings of the four quartic solutions: the one in which \( Z_1 \) and \( Z_2 \) are paired together as roots of a quadratic on the left side of (4-3). Solution \( Z_1 \) can also pair with \( Z_3 \) or \( Z_4 \) as roots of a quadratic. Thus, the four quartic-equation solutions \( Z_0 \) have a total of three possible pairing combinations leading to three corresponding values of \( u \).
$$u_1 = Z_1 Z_2 + Z_3 Z_4 \quad u_2 = Z_1 Z_3 + Z_2 Z_4 \quad u_3 = Z_1 Z_4 + Z_2 Z_3. \quad (4-16)$$

Because the different $u$ values correspond to different pairings of the $Z_n$, they produce different values of $p_1$, $p_2$, $q_1$, and $q_2$.

$$u_1 \Rightarrow p_1 = -(Z_1 + Z_2), \quad q_1 = Z_1 Z_2, \quad p_2 = -(Z_3 + Z_4), \quad q_2 = Z_3 Z_4 \quad (4-17)$$

$$u_2 \Rightarrow p_1 = -(Z_1 + Z_3), \quad q_1 = Z_1 Z_3, \quad p_2 = -(Z_2 + Z_4), \quad q_2 = Z_2 Z_4 \quad (4-18)$$

$$u_3 \Rightarrow p_1 = -(Z_1 + Z_4), \quad q_1 = Z_1 Z_4, \quad p_2 = -(Z_2 + Z_3), \quad q_2 = Z_2 Z_3 \quad (4-19)$$

The $p_m$ and $q_m$ values are real only if the two roots $Z_n$ of a quadratic are both real or are a complex conjugate pair. At least one of the three possible $u$ values produces such a set of real $p_m$ and $q_m$ values. We define $u_1$ as this proper choice of $u$, and solutions $Z_1$ and $Z_2$ as the proper roots of the first quadratic on the left side of (4-3).

The challenge is to design the algorithm to always select the proper $u$ value for $u_1$. If all four of the $Z_n$ are real, then any of the three $u_k$ is a good choice for $u_1$ because all of the $u_k$ are real and all produce real $p_m$ and $q_m$ values.

If the four solutions $Z_n$ are not all real, then either 1) two of the $Z_n$ are real and the other two are a complex conjugate pair, or 2) the four $Z_n$ consist of two complex conjugate pairs. The $u_2$ and $u_3$ values in these cases may be real, but the corresponding $p_m$ and $q_m$ values may not be real. The following paragraphs consider these possibilities. Each solution $Z_n$ is expressed as the sum of its real and imaginary components: $Z_n = X_n + iY_n$.

Suppose two of the $Z_n$ are real and the other two are a complex conjugate pair. Let the real $Z_n$ be roots of the first quadratic. Then $Z_1 = X_1$, $Z_2 = X_2$, $Z_3 = X_3 + iY_3$, and $Z_4 = X_3 - iY_3$ where $Y_3 > 0$. The three $u$ values in (4-16) become

$$u_1 = X_1 X_2 + X_3^2 + Y_3^2 \quad u_2 = X_3 (X_1 + X_2) + iY_3 (X_1 - X_2) \quad u_3 = X_3 (X_1 + X_2) - iY_3 (X_1 - X_2).$$

Solution $u_1$ is real, as are its corresponding $p_m$ and $q_m$ values in (4-17). If $X_1 \neq X_2$, then $u_2$ and $u_3$ are a complex conjugate pair with nonzero imaginary components. Such complex $u$ values are avoided, and the real $u$ value is selected as $u_1$. If $X_1 = X_2$, then the three $u$ values are all real, but $u_1$ is greater than $u_2$ and $u_3$:

$$u_1 = X_1^2 + X_3^2 + Y_3^2 \quad u_2 = u_3 = 2X_1 X_3,$$

$$u_1 = X_1^2 + X_3^2 + Y_3^2 > X_1^2 + X_3^2 = (X_1 - X_3)^2 + 2X_1 X_3 \geq 2X_1 X_3 = u_2 = u_3 \quad : \quad u_1 > u_2 = u_3.$$ 

Although $u_2$ and $u_3$ are real, they should be avoided because their corresponding $p_m$ and $q_m$ values in (4-18) and (4-19) are complex. For example, $p_1$ in (4-18) becomes

$$p_1 = -(Z_1 + Z_3) = -(X_1 + X_3 + iY_3) = -(X_1 + X_3) - iY_3 \quad \text{where} \quad Y_3 > 0.$$

Using $u_1$ as the greatest real solution of the resolvent cubic equation avoids such complex $p_m$ and $q_m$ values.
Now suppose that the four $Z_n$ consist of two complex conjugate pairs. Then $Z_1 = X_1 + iY_1$, $Z_2 = X_1 - iY_1$, $Z_3 = X_3 + iY_3$, and $Z_4 = X_3 - iY_3$ where $Y_1 > 0$ and $Y_3 > 0$. The three $u$ values in (4-16) become

$$u_1 = X_1^2 + Y_1^2 + X_3^2 + Y_3^2$$

$u_2 = 2(X_1X_3 - Y_1Y_3)$

$u_3 = 2(X_1X_3 + Y_1Y_3)$.

All three $u$ values are real, but $u_1$ is at least as great as $u_2$ and $u_3$.

$$u_1 = X_1^2 + Y_1^2 + X_3^2 + Y_3^2 = (X_1 - X_3)^2 + 2X_1X_3 + (Y_1 - Y_3)^2 + 2Y_1Y_3$$

$$\geq 2X_1X_3 + 2Y_1Y_3 = u_3 > 2X_1X_3 - 2Y_1Y_3 = u_2.$$  

$u_1 \geq u_3 > u_2.$

The $p_m$ and $q_m$ values corresponding to $u_1$ in (4-17) are real, but those corresponding to $u_2$ in (4-18) are complex. The $p_m$ and $q_m$ values corresponding to $u_3$ in (4-19) are also complex unless $Z_1 = Z_3$. In any case, complex $p_m$ and $q_m$ values are avoided by selecting $u_1$ as the greatest of the three real solutions of the resolvent cubic equation.

We see that selecting $u_1$ as the greatest real solution of the resolvent cubic equation for all cases assures that the algorithm operates only on real numbers: $u_1$, $p_1$, $p_2$, $q_1$, and $q_2$.

**Derivation 5: Euler Algorithms**

The Euler algorithms solve the depressed quartic equation

$$T_n^4 + b_2T_n^2 + b_1T_n + b_0 = 0, \quad n = 1, 2, 3, 4. \quad (5-1)$$

Euler’s derivation$^{[7, pp 256-257]}$ assumes that a solution of some quartic equation has the form

$$T_n = \sqrt{r_1} + \sqrt{r_2} + \sqrt{r_3} \quad (5-2)$$

where $r_1$, $r_2$, and $r_3$ are the three solutions of a cubic equation with real coefficients:

$$r_k^3 + a_2r_k^2 + a_1r_k + a_0 = 0. \quad (5-3)$$

The derivation finds that the quartic equation with solution (5-2) is a depressed quartic equation (5-1) whose coefficients $b_n$ are functions of the $a_m$ in (5-3). The derivation inverts these functions to calculate the $a_m$ from $b_n$. The cubic equation (5-3) with coefficients expressed in terms of the $b_n$ becomes the resolvent cubic equation. Its solutions $r_1$, $r_2$, and $r_3$ enable calculation of $T_n$ via (5-2).

Euler does not employ the principal-square-root convention for radicals. Instead, he initially allows each of the $\sqrt{r_k}$ in (5-2) to be either square root of $r_k$. The two square-root values for each of the three $r_k$ provide eight combinations of square-root terms in (5-2) for eight possible values for $T_n$. The derivation shows that only four of these eight are valid for any given $b_1$ value in (5-1).

The derivation begins with the requirement for the $r_k$ in (5-3):

$$(r - r_1)(r - r_2)(r - r_3) = r^3 + a_2r^2 + a_1r + a_0 \quad \text{for all } r.$$
Expand and simplify the left side. Then equate corresponding coefficients on the two sides to obtain:

\[-(r_1 + r_2 + r_3) = a_2 \]  \hspace{1cm} (5-4)
\[r_1 r_2 + r_1 r_3 + r_2 r_3 = a_1 \]  \hspace{1cm} (5-5)
\[-r_1 r_2 r_3 = a_0. \]  \hspace{1cm} (5-6)

We proceed to find the quartic equation whose solutions are given by (5-2). First square (5-2). Then apply (5-4) and rearrange.

\[T_n^2 = r_1 + r_2 + r_3 + 2\sqrt{r_1 r_2} + 2\sqrt{r_1 r_3} + 2\sqrt{r_2 r_3} \]
\[T_n^2 + a_2 = 2\left(\sqrt{r_1 r_2} + \sqrt{r_1 r_3} + \sqrt{r_2 r_3}\right) \]

Square both sides to obtain

\[T_n^4 + 2a_2 T_n^2 + a_2^2 = 4(r_1 r_2 + r_1 r_3 + r_2 r_3) + 8\sqrt{r_1 r_2} + 2\sqrt{r_1 r_3} + 2\sqrt{r_2 r_3}. \]

Apply (5-5) and (5-2). Then rearrange to standard form.

\[T_n^4 + 2a_2 T_n^2 - 8\sqrt{r_1 r_2} + 2\sqrt{r_1 r_3} + 2\sqrt{r_2 r_3} T_n + a_2^2 - 4a_1 = 0 \]  \hspace{1cm} (5-7)

Equation (5-6) shows that

\[\sqrt{r_1 r_2 r_3} = \sqrt{r_1 r_2} = \sqrt{-a_0}. \]  \hspace{1cm} (5-8)

Equation (5-7) becomes

\[T_n^4 + 2a_2 T_n^2 - 8\sqrt{-a_0} T_n + a_2^2 - 4a_1 = 0, \]  \hspace{1cm} (5-9)

which has the form of (5-1).

With this information, we can now find the resolvent cubic equation for the depressed quartic equation (5-1). Equate coefficients \(b_n\) in (5-1) to the corresponding coefficients in (5-9).

\[b_2 = 2a_2 \quad b_1 = -8\sqrt{-a_0} \quad b_0 = a_2^2 - 4a_1. \]  \hspace{1cm} (5-10)

Solve this system of equations for \(a_2, a_1,\) and \(a_0,\) and apply the results to (5-3).

\[a_2 = b_2/2 \quad a_1 = (b_2^2 - 4b_0)/16 \quad a_0 = -b_2^2/64 \]  \hspace{1cm} (5-11)

\[r_k^2 + (b_2/2)r_k^2 + [(b_2^2 - 4b_0)/16]r_k - b_2^2/64 = 0 \]  \hspace{1cm} (5-12)

This is Euler’s resolvent cubic equation.

The constant coefficient, \(a_0 = -b_2^2/64,\) provides information about the equation’s three solutions, \(r_1, r_2,\) and \(r_3.\) Because \(-b_2^2/64\) is less than or equal to zero, the equation has at least one nonnegative real solution, say \(r_1: r_1 \geq 0.\) Equations (5-6) and (5-11) for \(a_0\) combine to show that

\[r_1 r_2 r_3 = b_2^2/64 \geq 0. \]  \hspace{1cm} (5-13)
The product of all three solutions is a nonnegative real number. As a result, the product \( r_2r_3 \) is a nonnegative real number:
\[
r_1 \geq 0 \quad \text{and} \quad r_1r_2r_3 = b_1^2/64 \geq 0 \quad \Rightarrow \quad r_2r_3 \geq 0.
\]

Solutions \( r_2 \) and \( r_3 \) are real or they form a complex conjugate pair. If they are real, then they cannot have opposite signs.

Through its constant coefficient, \(-b_1^2/64\), equation (5-12) depends on the modulus of \( b_1 \), but not on its sign. Thus (5-12) is the resolvent cubic equation for two quartic equations:
\[
T_n^4 + b_2T_n^2 \pm b_1T_n + b_0 = 0.
\]

Each of these two quartic equations has four solutions for a total of eight solutions. These are the eight possible values of \( T_n \) given by (5-2):
\[
T_n = \sqrt{r_1} + \sqrt{r_2} \pm \sqrt{r_3}.
\]  

(5-2)

To determine which four of the \( T_n \) values from (5-2) are solutions of
\[
T_n^4 + b_2T_n^2 + b_1T_n + b_0 = 0,
\]  

(5-1)

combine (5-10) for \( b_1 \) with (5-8) to obtain
\[
\sqrt[4]{r_1}\sqrt[4]{r_2}\sqrt[4]{r_3} = -b_1/8.
\]  

(5-14)

Equation (5-2) produces a desired solution \( T_n \) only if the three radical terms on the right satisfy (5-14). The user is allowed to select either of two square roots for any two of the \( \sqrt{r_k} \). The third \( \sqrt{r_k} \) is selected to satisfy (5-14). The user only needs to check that the two sides of (5-14) have the same sign because (5-13) guarantees that they have the same modulus.

\[
r_1r_2r_3 = b_1^2/64 \quad \Rightarrow \quad |\sqrt[4]{r_1}\sqrt[4]{r_2}\sqrt[4]{r_3}| = |b_1/8|
\]

With a set of square roots \( \sqrt{r_k} \) that satisfies (5-14), the four solutions of (5-1) become:
\[
\begin{align*}
T_1 &= \sqrt{r_1} + \sqrt{r_2} + \sqrt{r_3} \\
T_2 &= \sqrt{r_1} - \sqrt{r_2} - \sqrt{r_3} \\
T_3 &= -\sqrt{r_1} + \sqrt{r_2} - \sqrt{r_3} \\
T_4 &= -\sqrt{r_1} - \sqrt{r_2} + \sqrt{r_3}.
\end{align*}
\]  

(5-15)-(5-18)

Each of these \( T_n \) expressions is valid because its terms are all square roots of the \( r_k \), and the product of its terms equals \(-b_1/8\) to satisfy (5-14).

In summary, Euler's original algorithm
- starts with the coefficients \( b_2, b_1, \) and \( b_0 \) of the depressed quartic equation (5-1),
- solves the resolvent cubic equation (5-12) for \( r_1, r_2, \) and \( r_3 \).
selects signs of \( \sqrt{r_1}, \sqrt{r_2}, \) and \( \sqrt{r_3} \) to satisfy (5-14),
• and calculates the four solutions \( T_n \) of (5-1) by using (5-15) through (5-18).

The conversion of Euler's original algorithm to the modified algorithm was described previously in Part II. Titles of the relevant Part II sections are underlined in the following summary. The original \( T_n \) formulas are recast to use the Principal-Square-Root Convention for Radicals, equations (6) and (7). Solution \( r_1 \) of the resolvent cubic equation is defined as the Greatest Real Solution of the Resolvent Cubic Equation. The section \( T_n \) Formulas for Euler and DLMF Algorithms shows that \( r_1 \) is nonnegative, (13), and derives the simplified form of the modified \( T_n \) formulas, (15).

\[
T_{1,2} = \sqrt{r_1} \pm \sqrt{r_2 + r_3 - 2\Sigma\sqrt{r_2 r_3}} \quad T_{3,4} = -\sqrt{r_1} \pm \sqrt{r_2 + r_3 + 2\Sigma\sqrt{r_2 r_3}}.
\]

The inner integrand \( r_2 r_3 \) is a nonnegative real number by (13). Solutions \( r_2 = x_2 + iy_2 \) and \( r_3 = x_3 + iy_3 \) of the resolvent cubic equation are real (\( y_2 = y_3 = 0 \)), or they form a complex conjugate pair \( (x_2 = x_3, y_2 = -y_3 > 0) \). In either case, the sum \( r_2 + r_3 \) equals \( x_2 + x_3 \), and the product \( r_2 r_3 \) equals \( x_2 x_3 + y_2^2 \). The modified algorithm \( T_n \) formulas become

\[
T_{1,2} = \sqrt{r_1} \pm \sqrt{x_2 + x_3 - 2\Sigma\sqrt{x_2 x_3 + y_2^2}} \quad T_{3,4} = -\sqrt{r_1} \pm \sqrt{x_2 + x_3 + 2\Sigma\sqrt{x_2 x_3 + y_2^2}}.
\]

All constituents of these \( T_n \) formulas are real numbers, and the inner integrand \( r_2 r_3 = x_2 x_3 + y_2^2 \) is a nonnegative real number. Thus, the \( T_n \) formulas require operations on real numbers only.

### Derivation 6: DLMF Algorithms
This section derives the DLMF original and modified algorithms for solving the depressed quartic equation

\[
T_n^4 + b_2 T_n^2 + b_1 T_n + b_0 = 0, \quad n = 1, 2, 3, 4. \tag{6-1}
\]

In the DLMF original algorithm and its derivation here, the radical indicates that either of the two possible square roots applies.

**The Resolvent Cubic Equation**
Express the quartic in (6-1) as the product of two quadratics with real coefficients.

\[
(T^2 + p_1 T + q_1)(T^2 + p_2 T + q_2) = T^4 + b_2 T^2 + b_1 T + b_0 \tag{6-2}
\]

Expand and simplify the left side.

\[
T^4 + (p_1 + p_2) T^3 + (p_1 p_2 + q_1 + q_2) T^2 + (p_1 q_2 + p_2 q_1) T + q_1 q_2 = T^4 + b_2 T^2 + b_1 T + b_0
\]

Equate corresponding coefficients from the two sides to create the following system of equations.

\[
p_1 + p_2 = 0 \tag{6-3}
\]
\[
p_1 p_2 + q_1 + q_2 = b_2 \tag{6-4}
\]
\[
p_1 q_2 + p_2 q_1 = b_1 \tag{6-5}
\]
Define $\theta_k$ as

$$\theta_k = p_1 p_2.$$  

Then (6-4) becomes

$$q_1 + q_2 = b_2 - \theta_k.$$  

Equations (6-3) and (6-7) give $0$ and $\theta_k$ as the sum and product of $p_1$ and $p_2$; (6-8) and (6-6) give $b_2 - \theta_k$ and $b_0$ as the sum and product of $q_1$ and $q_2$. Values of $p_1$, $p_2$, $q_1$, and $q_2$ are therefore solutions of the quadratic equations

$$p_m^2 + \theta_k = 0 \quad \text{and} \quad q_m^2 - (b_2 - \theta) q_m + b_0 = 0, \quad m = 1, 2.$$  

The solutions are

$$p_1 = -\sqrt{-\theta_k} \quad \quad q_1 = \frac{1}{2} \left[ b_2 - \theta_k + \sqrt{(b_2 - \theta_k)^2 - 4b_0} \right]$$

$$p_2 = \sqrt{-\theta_k} \quad \quad q_2 = \frac{1}{2} \left[ b_2 - \theta_k - \sqrt{(b_2 - \theta_k)^2 - 4b_0} \right]$$

where $\Sigma'$ has a value of either 1 or -1.

Substitute the expressions for $p_1$, $p_2$, $q_1$, and $q_2$ into (6-5) and simplify.

$$\Sigma' \sqrt{-\theta_k[(b_2 - \theta_k)^2 - 4b_0]} = b_1$$

Square both sides and rearrange to form the DLMF resolvent cubic equation.

$$\theta_k^3 - 2b_2 \theta_k^2 + (b_2^2 - 4b_0) \theta_k + b_1^2 = 0 \quad (6-9)$$

The Three Solutions of the Resolvent Cubic Equation

The three solutions $\theta_1$, $\theta_2$, and $\theta_3$ of (6-9) satisfy the requirement

$$(\theta - \theta_1)(\theta - \theta_2)(\theta - \theta_3) = \theta^3 - 2b_2 \theta^2 + (b_2^2 - 4b_0) \theta + b_1^2 \quad \text{for all } \theta.$$  

Expand and simplify the left side, and then equate corresponding coefficients from the two sides.

$$\theta_1 + \theta_2 + \theta_3 = 2b_2 \quad (6-10)$$

$$\theta_1 \theta_2 + \theta_1 \theta_3 + \theta_2 \theta_3 = b_2^2 - 4b_0 \quad (6-11)$$

$$-\theta_1 \theta_2 \theta_3 = b_1^2 \quad (6-12)$$

DLMF uses all three solutions $\theta_1$, $\theta_2$, and $\theta_3$ of (6-9). Each solution corresponds to its particular grouping of the four $T_n$ into two pair: each pair of $T_n$ are the roots of a quadratic on the left side (6-2).

Solution $\theta_1$ corresponds to $T_1$ paired with $T_2$ as roots of $T^2 + p_1 T + q_1$, so $T_3$ and $T_4$ are roots of $T^2 + p_2 T + q_2$. Then

$$(T - T_1)(T - T_2) = T^2 + p_1 T + q_1 \quad \text{and} \quad (T - T_3)(T - T_4) = T^2 + p_2 T + q_2$$

imply that

$$p_1 = -(T_1 + T_2) \quad p_2 = -(T_3 + T_4).$$

These expressions for $p_1$ and $p_2$ combined with (6-3) and (6-7) show that
Part III -- Algorithm Derivations

\[ T_1 + T_2 + T_3 + T_4 = 0 \]  \hspace{1cm} (6-13)

and

\[ \theta_1 = (T_1 + T_2) (T_3 + T_4) = -(T_1 + T_2)^2 = -(T_3 + T_4)^2. \]  \hspace{1cm} (6-14)

Solutions \( \theta_2 \) and \( \theta_3 \) correspond to the two alternate pairings: \( T_1 \) paired with \( T_3 \) and \( T_1 \) paired with \( T_4 \):

\[ \begin{align*}
\theta_2 &= (T_1 + T_3) (T_2 + T_4) = -(T_1 + T_3)^2 = -(T_2 + T_4)^2 \quad (6-15) \\
\theta_3 &= (T_1 + T_4) (T_2 + T_3) = -(T_1 + T_4)^2 = -(T_2 + T_3)^2. \quad (6-16)
\end{align*} \]

**Solutions of the Depressed Quartic Equation in the Original Algorithm**

The solutions \( T_n \) of the depressed quartic equation derive from (6-13) to (6-16). Equations (6-14) to (6-16) provide the following corresponding expressions.

\[ \begin{align*}
T_1 + T_2 &= -(T_3 + T_4) = \sqrt{-\theta_1} \\
T_1 + T_3 &= -(T_2 + T_4) = \sqrt{-\theta_2} \\
T_1 + T_4 &= -(T_2 + T_3) = \sqrt{-\theta_3}.
\end{align*} \]

By invoking \( T_1+T_2+T_3+T_4 = 0 \) from (6-13), these last three equations convert to the four \( T_n \) formulas as follows.

\[ \begin{align*}
T_1 &= \frac{1}{2}(T_1+T_2 + T_1+T_3 + T_1+T_4) \Rightarrow \\
T_2 &= \frac{1}{2}(T_1+T_2 + T_2+T_4 + T_2+T_3) \Rightarrow \\
T_3 &= \frac{1}{2}(T_3+T_4 + T_1+T_3 + T_2+T_3) \Rightarrow \\
T_4 &= \frac{1}{2}(T_3+T_4 + T_2+T_4 + T_1+T_4) \Rightarrow
\end{align*} \]

\[ \begin{align*}
T_1 &= \frac{1}{2}(\sqrt{-\theta_1} + \sqrt{-\theta_2} + \sqrt{-\theta_3}) \quad (6-17) \\
T_2 &= \frac{1}{2}(\sqrt{-\theta_1} - \sqrt{-\theta_2} - \sqrt{-\theta_3}) \quad (6-18) \\
T_3 &= \frac{1}{2}(-\sqrt{-\theta_1} + \sqrt{-\theta_2} - \sqrt{-\theta_3}) \quad (6-19) \\
T_4 &= \frac{1}{2}(-\sqrt{-\theta_1} - \sqrt{-\theta_2} + \sqrt{-\theta_3}) \quad (6-20)
\end{align*} \]

To qualify as solutions of the depressed quartic equation (6-1), these expressions for the four \( T_n \) must satisfy the requirement

\[ (T-T_1) (T-T_2) (T-T_3) (T-T_4) = T^4 + b_2 T^2 + b_1 T + b_0 \quad \text{for all } T. \]

That is, the \( T_n \) must satisfy the system:

\[ \begin{align*}
T_1 + T_2 + T_3 + T_4 &= 0 \quad (6-21) \\
T_1 T_2 + T_1 T_3 + T_1 T_4 + T_2 T_3 + T_2 T_4 + T_3 T_4 &= b_2 \quad (6-22) \\
-T_1 T_2 T_3 - T_1 T_2 T_4 - T_1 T_3 T_4 - T_2 T_3 T_4 &= b_1 \quad (6-23) \\
T_1 T_2 T_3 T_4 &= b_0. \quad (6-24)
\end{align*} \]

The \( T_n \) of (6-17) to (6-20) do satisfy (6-21). They also satisfy (6-22) and (6-24) as verified with (6-10) and (6-11). However, (6-23) holds only if

\[ \sqrt{-\theta_1} \sqrt{-\theta_2} \sqrt{-\theta_3} = -b_1. \]  \hspace{1cm} (6-25)
Equation (6-25) becomes a restriction on the $\sqrt{-\theta_k}$. The user selects either square root for any two of the $\sqrt{-\theta_k}$. The third $\sqrt{-\theta_k}$ is selected to satisfy (6-25). The user only needs to check that the two sides of this equation have the same sign. Equation (6-12) guarantees that the two sides have equal magnitudes:

$$-\theta_1 \theta_2 \theta_3 = b_1^2 \quad \Rightarrow \quad |\sqrt{-\theta_1 \theta_2 \theta_3}| = |\sqrt{-\theta_1} \sqrt{-\theta_2} \sqrt{-\theta_3}| = |-b_1|. \quad (6-26)$$

This completes the derivation of the DLMF original algorithm, which is summarized as follows.

- Given the coefficients $b_2, b_1,$ and $b_0$ of (6-1), the algorithm solves the resolvent cubic equation (6-9) for its three solutions $\theta_1, \theta_2, \text{ and } \theta_3$.
- Signs of the three $\sqrt{-\theta_k}$ are selected to satisfy (6-25).
- Equations (6-17) to (6-20) give the solutions $T_1, T_2, T_3 \text{ and } T_4$ of the depressed quartic equation (6-1).

**DLMF Modified Algorithm**

Conversion of the DLMF original algorithm to the modified algorithm is similar to the corresponding conversion involving the Euler algorithms.

First, the original $T_n$ formulas are recast to use the principal-square-root convention for radicals. This is accomplished by replacing $\sqrt{-\theta_3}$ with $-\Sigma s \sqrt{-\theta_3}$ where

$$\Sigma = \begin{cases} 1 & \text{if } b_1 > 0 \\ -1 & \text{otherwise} \end{cases} \quad \text{and} \quad s = \begin{cases} 1 & \text{if } \sqrt{-\theta_1 \sqrt{-\theta_2} \sqrt{-\theta_3}} \geq 0 \\ -1 & \text{otherwise} \end{cases}. \quad (6-27)$$

The definitions of these special functions and (6-26) imply that

$$b_1 = \Sigma |b_1| \quad \text{and} \quad \sqrt{-\theta_1 \sqrt{-\theta_2} \sqrt{-\theta_3}} = s |\sqrt{-\theta_1 \sqrt{-\theta_2} \sqrt{-\theta_3}}| = s |b_1|.\quad (6-26)$$

The DLMF original $T_n$ formulas change from

$$T_{1,2} = \frac{1}{2} \left[ \sqrt{-\theta_1 \pm (\sqrt{-\theta_2} + \sqrt{-\theta_3})} \right] \quad \text{and} \quad T_{3,4} = \frac{1}{2} \left[ \sqrt{-\theta_1 \pm (\sqrt{-\theta_2} - \sqrt{-\theta_3})} \right]$$

to

$$T_{1,2} = \frac{1}{2} \left[ \sqrt{-\theta_1 \pm (\sqrt{-\theta_2} - \Sigma s \sqrt{-\theta_3})} \right] \quad \text{and} \quad T_{3,4} = \frac{1}{2} \left[ \sqrt{-\theta_1 \pm (\sqrt{-\theta_2} + \Sigma s \sqrt{-\theta_3})} \right]. \quad (6-28)$$

In this revised formulation, the product of terms inside the brackets for all $T_n$ is

$$-\Sigma s \sqrt{-\theta_1 \sqrt{-\theta_2} \sqrt{-\theta_3}} = -\Sigma s^2 |b_1| = -\Sigma |b_1| = -b_1 \quad \text{as required by (6-25).}$$

The function $s$ in (6-27) accommodates the condition $\sqrt{-\theta_1 \sqrt{-\theta_2} \sqrt{-\theta_3}} < 0$, which occurs when one of the $-\theta_k$, say $-\theta_i$, is positive real and the other two $-\theta_k$ are negative real:

$$\sqrt{-\theta_2} = i \sqrt{|\theta_2|}, \quad \sqrt{-\theta_3} = i \sqrt{|\theta_3|} \quad \Rightarrow \quad \sqrt{-\theta_2} \sqrt{-\theta_3} = -\sqrt{|\theta_2| \sqrt{|\theta_3|}} = -\sqrt{|\theta_2 \theta_3|} < 0.$$
The revised DLMF algorithm specifies that $\theta_1$ is a nonpositive real solution, $\theta_1 \leq 0$, of the resolvent cubic equation (6-9). Such a solution must exist because the constant term on the left side is $b_1^2 \geq 0$. As a result, both $-\theta_1$ and its principal square root are nonnegative real numbers.

$$-\theta_1 \geq 0 \implies \sqrt{-\theta_1} \geq 0 \quad (6-29)$$

Equation (6-12) shows that $-\theta_1 \theta_2 \theta_3 = b_1^2 \geq 0$. Therefore,

$$-\theta_1 \geq 0 \quad \text{and} \quad -\theta_1 \theta_2 \theta_3 = b_1^2 \geq 0 \implies \theta_2 \theta_3 \geq 0. \quad (6-30)$$

The product $\theta_2 \theta_3$ is a nonnegative real number. Thus, $\theta_2 = \theta_{x2} + i\theta_{y2}$ and $\theta_3 = \theta_{x3} + i\theta_{y3}$ are real ($\theta_{y2} = \theta_{y3} = 0$), or they form a complex conjugate pair ($\theta_{x2} = \theta_{x3}$, $\theta_{y2} = -\theta_{y3} > 0$). If real, then they cannot have opposite signs. This restriction on $\theta_2$ and $\theta_3$ implies that each parenthetical expression in (6-28) is either real or pure imaginary.

Conversion of the $T_n$ formulas in (6-28) to those in the DLMF modified algorithm starts by replacing each parenthetical expression with the radical of its square.

$$T_{1,2} = \frac{1}{2} \left[\sqrt{-\theta_1} \pm \sqrt{-\theta_2 - \theta_3 - 2s\Sigma\sqrt{-\theta_2}\sqrt{-\theta_3}}\right] \quad (6-31)$$

$$T_{3,4} = \frac{1}{2} \left[-\sqrt{-\theta_1} \pm \sqrt{-\theta_2 - \theta_3 + 2s\Sigma\sqrt{-\theta_2}\sqrt{-\theta_3}}\right]$$

$T_1$ and $T_2$ in (6-31) each have the same value as in (6-28) unless $\sqrt{-\theta_2 - \Sigma s\sqrt{-\theta_3}}$ happens to be either negative real or negative imaginary. In that case, $T_1$ in (6-28) becomes $T_2$ in (6-31) and $T_2$ in (6-28) becomes $T_1$ in (6-31). $T_3$ and $T_4$ are correspondingly affected by the value of $\sqrt{-\theta_2 + \Sigma s\sqrt{-\theta_3}}$.

As an option to prevent the $T_n$ from flipping values between (6-28) and (6-31), use the following convention: select $\theta_2 = \theta_{x2} + i\theta_{y2}$ and $\theta_3 = \theta_{x3} + i\theta_{y3}$ so that $|\theta_{x2}| \geq |\theta_{x3}|$ and $\theta_{y2} = -\theta_{y3} \geq 0$. The convention assures that the parenthetical expressions in (6-28) are either nonreal negative or nonreal imaginary. The convention does not affect (6-31), which is symmetrical with respect to $\theta_2$ and $\theta_3$.

Equation (6-29) implies that the formula for $s$ in (6-27) simplifies to

$$s = \begin{cases} 1 & \text{if } \sqrt{-\theta_2}\sqrt{-\theta_3} \geq 0 \\ -1 & \text{otherwise} \end{cases} \implies \sqrt{-\theta_2}\sqrt{-\theta_3} = s|\sqrt{-\theta_2}\sqrt{-\theta_3}| = s|\sqrt{\theta_2\theta_3}|.$$

This result and (6-30) imply that

$$s\sqrt{-\theta_2}\sqrt{-\theta_3} = s^2|\theta_2\theta_3| = \sqrt{\theta_2\theta_3}.$$

The $T_n$ formulas in (6-31) become

$$T_{1,2} = \frac{1}{2} \left[\sqrt{-\theta_1} \pm \sqrt{-\theta_2 - \theta_3 - 2\sqrt{\theta_2\theta_3}}\right] \quad T_{3,4} = \frac{1}{2} \left[-\sqrt{-\theta_1} \pm \sqrt{-\theta_2 - \theta_3 + 2\sqrt{\theta_2\theta_3}}\right].$$
Whether the values $\theta_2 = \theta_{x_2} + i\theta_{y_2}$ and $\theta_3 = \theta_{x_3} + i\theta_{y_3}$ are real ($\theta_{y_2} = \theta_{y_3} = 0$) or form a complex conjugate pair ($\theta_{x_2} = \theta_{x_3}$, $\theta_{y_2} = -\theta_{y_3} > 0$), we have $-\theta_2 - \theta_3 = -\theta_{x_2} - \theta_{x_3}$ and $\theta_2\theta_3 = \theta_{x_2}\theta_{x_3} + \theta_{y_2}^2$. The final $T_n$ formulas for the DLMF modified algorithm become

$$T_{1,2} = \frac{1}{2} \left[ -\theta_1 \pm \sqrt{-\theta_{x_2} - \theta_{x_3} - 2\Sigma \sqrt{\theta_{x_2}\theta_{x_3} + \theta_{y_2}^2}} \right]$$

$$T_{3,4} = \frac{1}{2} \left[ -\sqrt{-\theta_1} \pm \sqrt{-\theta_{x_2} - \theta_{x_3} + 2\Sigma \sqrt{\theta_{x_2}\theta_{x_3} + \theta_{y_2}^2}} \right].$$

All constituents of these DLMF modified $T_n$ formulas are real numbers, and the inner integrand $\theta_2\theta_3 = \theta_{x_2}\theta_{x_3} + \theta_{y_2}^2$ is a nonnegative real number. The calculation therefore requires operations on real numbers only.
References


9 National Institute of Standards and Technology, *Digital Library of Mathematical Functions* (2019-03-15) DLMF Update; Version 1.0.22, [https://dlmf.nist.gov/1.11#iii](https://dlmf.nist.gov/1.11#iii).

Appendix A -- Computational Instability of the Ferrari Common Algorithm

This appendix demonstrates that the Ferrari common algorithm becomes computationally unstable as solution m of the resolvent cubic equation approaches zero. The table below gives the algorithm to find the four solutions $T_n$ ($n = 1, 2, 3, 4$) of the depressed quartic equation

$$T_n^4 + b_2 T_n^2 + b_1 T_n + b_0 = 0 \quad \text{(A-1)}$$

where the three real coefficients $b_2, b_1, \text{and } b_0$ are given.

<table>
<thead>
<tr>
<th>Solve this resolvent cubic equation for real m:</th>
<th>If m = 0, then</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m^3 + b_2 m^2 + (b_2^2/4 - b_0)m - b_2^2/8 = 0.$</td>
<td>$T_{1,2} = \pm \sqrt{-b_2/2 + \sqrt{b_2^2/4 - b_0}}$</td>
</tr>
<tr>
<td>Use a real solution $m &gt; 0$ if it exists. Otherwise, $m = 0$.</td>
<td>$T_{3,4} = \pm \sqrt{-b_2/2 - \sqrt{b_2^2/4 - b_0}}$</td>
</tr>
</tbody>
</table>

If a nonzero real solution $m$ of the resolvent cubic equation becomes sufficiently small, then the $m^3$ and $m^2$ terms vanish, and that equation becomes

$$(b_2^2/4 - b_0)m - b_2^2/8 = 0 \quad \Rightarrow \quad b_1^2/(8m) = b_2^2/4 - b_0 \quad \Rightarrow \quad b_1 = \pm (2\sqrt{2m}) \sqrt{b_2^2/4 - b_0} \quad \text{for sufficiently small } m.$$

Thus, $b_1$ approaches zero as $m$ approaches zero.

The fraction $b_1/(2\sqrt{2m})$ causes computational instability for small $m$ because the $m$ value, calculated as the solution of a cubic equation, typically contains a small round-off error $e$ not found in $b_1$. Square brackets distinguish such a calculated value $[m]$ from the true value $m$.

$$[m] = m + e \quad \text{(A-3)}$$

The calculated fraction $[b_1/(2\sqrt{2m})]$ becomes

$$
\left[ \frac{b_1}{2\sqrt{2m}} \right] = \frac{b_1}{2\sqrt{2[m]}} = \frac{b_1}{2\sqrt{2(m + e)}} = \frac{b_1}{2\sqrt{2m}} \sqrt{m + e} = \frac{b_1}{2\sqrt{2m}} (1 + e/m)^{-1/2}.
$$

When $m$ approaches the magnitude of $e$, then the calculated fraction $[b_1/(2\sqrt{2m})]$ is dominated by error. Suppose $e$ is negative. As $m$ approaches $|e|$, the factor $(1 + e/m)^{-1/2}$ becomes unbounded. The calculated fraction $[b_1/(2\sqrt{2m})]$ and the calculated solutions $[T_n]$ are then also unbounded.

The Ferrari common algorithm becomes unstable when $m$ and $b_1$ approach the order of round-off error $e$, but the instability is even worse when the constant coefficient $b_0$ also diminishes with $m$ and approaches $|e|$. The following example demonstrates. Let $\mu > 0$
be an adjustable real parameter, and let the true solutions $T_n$ of a depressed quartic equation be

$$T_1 = T_2 = \sqrt{\mu/2} \quad \text{and} \quad T_{3,4} = -\sqrt{\mu/2} \pm 2i. \quad (A-4)$$

The depressed quartic equation (A-1) is

$$(T_n - \sqrt{\mu/2})^2 [T_n - (-\sqrt{\mu/2} + 2i)][T_n - (-\sqrt{\mu/2} - 2i)] =$$

$$T_n^4 + (4 - \mu)T_n^3 - 8\sqrt{\mu/2} T_n + 2\mu + \mu^2/4 = 0.$$  

The coefficients $b_2$, $b_1$, and $b_0$ are:

$$b_2 = 4 - \mu \quad b_1 = -8\sqrt{\mu/2} \quad b_0 = 2\mu + \mu^2/4. \quad (A-5)$$

The resolvent cubic equation (A-2) is

$$m^3 + (4 - \mu)m^2 + 4(1 - \mu)m - 4\mu = 0. \quad (A-6)$$

The left side factors to $(m-\mu)(m+2)^2$, so the three solutions are $\mu$, $-2$, and $-2$. Solution $m = \mu$, the only nonnegative real solution, applies. If (A-5) and $m = \mu$ are inserted into the Ferrari common $T_n$ formulas, those formulas produce the four $T_n$ solutions in (A-4).

If the coefficients in (A-6) are given as numerical values, then the calculated solution $[m] = m + e = \mu + e$ contains a round-off error $e$, which in turn produces error in the calculated values of the $T_n$. The calculated solution for $T_1$ using the Ferrari common algorithm is

$$[T_1] = \sqrt{[m]/2} + \sqrt{-[m]/2 - b_2/2 - b_1/(2\sqrt{2[m]})}. \quad (A-7)$$

Substitute (A-3), $\mu = m$, and (A-5) into (A-7) and simplify to obtain

$$[T_1] = \frac{\sqrt{(m + e)/2}}{} + E$$

where

$$E = \sqrt{2f(e/m) - e/2} \quad \text{and} \quad f(e/m) = (1 + e/m)^{-1/2} - 1.$$ 

We consider only the case $m = \mu > |e|$. If $e$ is negative ($0 < -e < m$), then $2f(e/m) - e/2$ is positive, and $E$ is real. If $e$ is positive, then $2f(e/m) - e/2$ is negative, and $E$ is imaginary. If $e = 0$, then $E = 0$ and $[T_1] = \sqrt{m/2} = \sqrt{\mu/2} = T_1$.

The figure on the next page demonstrates graphically the instability of the Ferrari common algorithm as $m = \mu$ becomes small. The first graph plots $E$, $[T_1]$, and true $T_1$ versus $m = \mu$ for an assumed constant round-off error $e = -1 \times 10^{-16}$. This $e$ value is typical for a 64-bit operating system applied to this problem. The plot shows how error $E$ increases as $m$ diminishes. Error $E$ starts to dominate $[T_1]$ as $m$ falls to $10^{-8}$. This $m$ value is the square root of $|e|$ and $10^8$ times as great as $|e|$. As $m$ decreases further and approaches $10^{-16}$, $E$ and $[T_1]$ increase without limit whereas the true $T_1$ value approaches $10^{-8}/\sqrt{2}$.
Figure A-1  Computational Instability of Ferrari Common Algorithm for Small m
The second graph demonstrates the same effect by using a 64-bit operating system to calculate $b_2$, $b_1$, and $b_0$ from $\mu$, then calculate $[m]$ as the solution of resolvent cubic equation \((A-6)\), and finally calculate $[T_1]$ as the solution of \((A-7)\). The graph plots $[T_1]$ as its real part, its imaginary part, and its modulus. True $T_1 = \sqrt{\mu/2}$ is also plotted for reference.

In summary, the fraction $b_1/(2\sqrt{2m})$ in the Ferrari common $T_n$ formulas produces computational instability. The value $m$, calculated as the solution of the resolvent cubic equation, typically contains a small round-off error $e$ not found in $b_1$. As $m$ diminishes to the magnitude of $e$, then the calculated fraction $[b_1/(2\sqrt{2m})]$ is dominated by error. The instability is particularly severe in the case illustrated in Figure A-1 above. The calculated value $[T_1]$ suffers large error even when the $m$ value is several orders of magnitude greater than the round-off error $e$. If error $e$ is negative, then the error in $[T_1]$ can become unbounded as $m$ approaches the modulus of $e$. By reason of this instability, the Ferrari common algorithm is not recommended for general calculation.