Blog Entry © Monday to Tuesday, June 8 - 9, 2026, by James Pate Williams, Jr., Damped Newton's Method for a System of Equations

Reference: ***Elementary Numerical Analysis: An Algorithmic Approach Third Editon*** © 1980 by S. D. Conte and Carl de Boor Section 5.2 pages 216 to 222

This technique requires a system of n-equations and n-unknowns. First, we must compute initial values of the n-unknowns and then calculate the initial Jacobian of the system and its inverse. Suppose n = 2 then the Jacobian is of the form:

$$J = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

The inverse is:

$$I = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Now assume n = 3 and use Cramer's Method to compute the inverse:

$$J = \begin{bmatrix} o & p & q \\ r & s & t \\ u & v & w \end{bmatrix}\begin{matrix} o & p \\ r & s \\ u & v \end{matrix}$$

$$I_{00} = \begin{bmatrix} 1 & p & q \\ 0 & s & t \\ 0 & v & w \end{bmatrix}\begin{matrix} 1 & p \\ 0 & s \\ 0 & v \end{matrix} = sw - tv$$

$$I_{01} = \begin{bmatrix} o & 1 & q \\ r & 0 & t \\ u & 0 & w \end{bmatrix}\begin{matrix} o & 1 \\ r & 0 \\ u & 0 \end{matrix} = tu - rw$$

$$I_{02} = \begin{bmatrix} o & p & 1 \\ r & s & 0 \\ u & v & 0 \end{bmatrix}\begin{matrix} o & p \\ r & s \\ u & v \end{matrix} = rv - su$$

$$I_{10} = \begin{bmatrix} 0 & p & q \\ 1 & s & t \\ 0 & v & w \end{bmatrix}\begin{matrix} 0 & p \\ 1 & s \\ 0 & v \end{matrix} = qv - pw$$

$$I_{11} = \begin{bmatrix} o & 0 & q \\ r & 1 & t \\ u & 0 & w \end{bmatrix}\begin{matrix} o & 0 \\ r & 1 \\ u & 0 \end{matrix} = ow - qu$$

$$I_{12} = \begin{bmatrix} o & p & 0 \\ r & s & 1 \\ u & v & 0 \end{bmatrix}\begin{matrix} o & p \\ r & s \\ u & v \end{matrix} = pu - ov$$

$$I_{20} = \begin{bmatrix} 0 & p & q \\ 0 & s & t \\ 1 & v & w \end{bmatrix}\begin{matrix} 0 & p \\ 0 & s \\ 1 & v \end{matrix} = pt - qs$$

$$I_{21} = \begin{bmatrix} o & 0 & q \\ r & 0 & t \\ u & 1 & w \end{bmatrix} \begin{matrix} o & 0 \\ r & 0 \\ u & 1 \end{matrix} = qr - ot$$

$$I_{22} = \begin{bmatrix} o & p & 0 \\ r & s & 0 \\ u & v & 1 \end{bmatrix} \begin{matrix} o & p \\ r & s \\ u & v \end{matrix} = os - pr$$

Now we must divide the inverse matrix elements by the determinant of the system:

$$det = osw + ptu + qrv - prw - otv - qsu$$

$$I_{ij} = \frac{I_{ij}}{det}$$

For n >= 4, we use an approximate Jacobian matrix based on finite differences. The first computation for n = 2 is Example 5.5 pages 219 to 220 in the preceding reference. It uses an exact Jacobian and inverse.

Tabular Data

```
Jacobian Matrix

+3.1842353704e+00        +3.0499296728e+00
+2.0188782500e+00        -1.3734783534e+00

Jacobian Inverse Matrix

+1.3042345789e-01        +2.8961677717e-01
+1.9170967039e-01        -3.0237024610e-01

Identity Matrix

+1.0000000000e+00        +3.3306690739e-16
-2.2204460493e-16        +1.0000000000e+00

x[ 0] = +1.3734783534e+00        f[ 0] = -4.4408920985e-16
x[ 1] = -1.5249648364e+00        f[ 1] = +0.0000000000e+00

Iterations = 76
Runtime in Seconds = 0.000000
```

Document was last saved: Just now

The n = 3 case is Example 5.4 found on pages 217 to 218 of the reference.

Tabular Data

```
Jacobian Matrix

-2.0275027809e+00       +1.1705792763e+00       +0.0000000000e+00
+1.4774687570e+00       -3.5117378290e+00       +1.9411861235e+00
+0.0000000000e+00       +2.1705792763e+00       -4.8098069800e+00

Jacobian Inverse Matrix

-7.2922100543e-01       -4.0876993644e-01       -1.8447051130e-01
-3.2386310312e-01       -5.6094734929e-01       -2.5314543735e-01
-1.3070764883e-01       -2.2639228871e-01       -3.1007531413e-01

Identity Matrix

+1.0000000000e+00       +1.1102230246e-16       -1.1102230246e-16
+1.1102230246e-16       +1.0000000000e+00       +2.2204460493e-16
-5.5511151231e-17       +1.1102230246e-16       +1.0000000000e+00

x[ 0] = +3.3791713015e-01       f[ 0] = +5.5511151231e-17
x[ 1] = +5.8528963816e-01       f[ 1] = +2.2204460493e-16
x[ 2] = +8.0163449666e-01       f[ 2] = -2.2204460493e-16

Iterations = 14
Runtime in Seconds = 0.000000
```

The final two tests use an approximate Jacobian and inverse.

Tabular Data

```
Jacobian Matrix

-1.0612344337e+00      +9.3258734069e-01      +0.0000000000e+00      +0.0000000000e+0
+1.0380585280e+00      -2.7977620221e+00      +1.5454304503e+00      +0.0000000000e+0
+0.0000000000e+00      +1.7807977315e+00      -4.1056047451e+00      +2.1627144520e+0
+0.0000000000e+00      +0.0000000000e+00      +2.3669954885e+00      -5.0848214528e+0

Jacobian Inverse Matrix

-1.8040901062e+00      -8.8103176981e-01      -4.3937928350e-01      -1.8688009700e-0
-9.8067226759e-01      -1.0025669559e+00      -4.9999008645e-01      -2.1265953895e-0
-5.6355593753e-01      -5.7613800191e-01      -6.1002542225e-01      -2.5946059444e-0
-2.6233651939e-01      -2.6819349783e-01      -2.8396816599e-01      -3.1744321241e-0

Identity Matrix

+1.0000000000e+00      +3.3306690739e-16      +2.2204460493e-16      -3.3306690739e-1
-2.2204460493e-16      +1.0000000000e+00      +3.3306690739e-16      -4.4408920985e-1
+0.0000000000e+00      -2.2204460493e-16      +1.0000000000e+00      -2.2204460493e-1
+0.0000000000e+00      +0.0000000000e+00      +0.0000000000e+00      +1.0000000000e+0

x[ 0] = +5.0111539823e-01        f[ 0] = -4.2829727337e-01
x[ 1] = +5.7013389099e-01        f[ 1] = +1.0942760400e-01
x[ 2] = +6.9614609998e-01        f[ 2] = +8.4377208618e-02
x[ 3] = +8.5267023677e-01        f[ 3] = -3.7410554388e-04

Iterations = 1000000
Runtime in Seconds = 96.030000
```

Tabular Data

```
-5.2535753525e+00

Jacobian Inverse Matrix

-2.5152843459e+00      -1.2860042466e+00      -7.1139234413e-01      -4.0463247759e-0
-1.8128082259e-01
-1.3621520336e+00      -1.3057856550e+00      -7.2233503153e-01      -4.1085656300e-0
-1.8406929703e-01
-8.8219502579e-01      -8.4568945399e-01      -8.7689979064e-01      -4.9877137111e-0
-2.2345632011e-01
-5.4851509567e-01      -5.2581732860e-01      -5.4522272116e-01      -6.0832417037e-0
-2.7253745587e-01
-2.5965211629e-01      -2.4890761117e-01      -2.5809359581e-01      -2.8796410432e-0
-3.1935837418e-01

Identity Matrix

+1.0000000000e+00      +4.4408920985e-16      -3.3306690739e-16      -1.6653345369e-1
+0.0000000000e+00
-2.2204460493e-16      +1.0000000000e+00      +0.0000000000e+00      -5.5511151231e-1
+0.0000000000e+00
+0.0000000000e+00      +0.0000000000e+00      +1.0000000000e+00      -2.2204460493e-1
-2.2204460493e-16
-5.5511151231e-17      -1.1102230246e-16      +2.2204460493e-16      +1.0000000000e+0
+0.0000000000e+00
-5.5511151231e-17      +5.5511151231e-17      +1.1102230246e-16      +0.0000000000e+0
+1.0000000000e+00

x[ 0] = +4.9961109647e-01       f[ 0] = -4.9162743979e-01
x[ 1] = +5.0715510779e-01       f[ 1] = +1.6490248597e-01
x[ 2] = +6.1582310208e-01       f[ 2] = +6.4385244358e-02
x[ 3] = +7.5100932439e-01       f[ 3] = -2.2188705024e-04
x[ 4] = +8.7842506186e-01       f[ 4] = +1.3256179637e-04

Iterations = 1000000
Runtime in Seconds = 130.854000
```