Algorithm 1 BISECTION

```
\overline{\textbf{Input: } f, a, b, tol, Nmax}
Output: x, iter, err
 1: fa \leftarrow f(a)
 2: fpm \leftarrow (a+b)/2
 3: fpm \leftarrow f(pm)
 4: E \leftarrow 1000
 5: cont \leftarrow 1
 6: While (E > tol) & (cont < Nmax) do
       if (fa * fpm < 0) Then
          b \leftarrow pm
 8:
       else if
 9:
          a \leftarrow pm
10:
       End if
11:
12:
       p0 \leftarrow pm
13:
       pm \leftarrow (a+b)/2
14:
       fpm \leftarrow f(pm)
       E \leftarrow |pm - p0|
15:
       cont \leftarrow cont + 1
16:
17: End While
18: x \leftarrow pm
19: iter \leftarrow cont
20: err \leftarrow E
21: Return x, iter, err
```

Algorithm 2 FALSE RULE

```
Input: f, a, b, tol, nax
Output: x, iter, err
 1: fa \leftarrow f(a)
 2: fb \leftarrow f(b)
 3: fpm \leftarrow f(pm)
 4: E \leftarrow 1000
 5: cont \leftarrow 1
 6: While (E > tol) & (cont < Nmax) do
       if (fa*fmp<0) Then
          b \leftarrow pm
 8:
 9:
       else if
10:
          a \leftarrow pm
       End if
11:
12:
       p0 \leftarrow pm
      pm \leftarrow (f(b) * a - f(a) * b) / f(b) - f(a))
13:
       fpm \leftarrow f(pm)
       E \leftarrow |pm - p0|
15:
       cont \leftarrow cont + 1
16:
17: End While
18: x \leftarrow pm
19: iter \leftarrow cont
20: err \leftarrow E
21: Return x, iter, err
```

Algorithm 3 FIXED POINT

```
Input: g, x0, tol, Nmax

Output: x, iter, err

1: xant \leftarrow x0

2: E \leftarrow 1000

3: cont \leftarrow 1

4: While (E > tol) & (cont < Nmax) do

5: xact \leftarrow g(xant)

6: E \leftarrow |xact - xant|

7: cont \leftarrow cont + 1

8: xant \leftarrow xact

9: End While

10: x \leftarrow xact

11: iter \leftarrow cont

12: err \leftarrow E

13: Return x, iter, err
```

${\bf Algorithm~4~\rm NEWTON}$

```
Input: f, df, x0, tol, Nmax
Output: x, iter, err
 1: xant \leftarrow x0
 2: fant \leftarrow f(xant)
 3: E \leftarrow 1000
 4: cont \leftarrow 0
 5: While (E > tol) & (cont < Nmax) do
       xact \leftarrow (xant - fant)/(df(xant))
       fact \leftarrow f(xact)
 7:
       E \leftarrow |xact - xant|
 8:
       cont \leftarrow cont + 1
10:
       xant \leftarrow xact
       fant \leftarrow fact
11:
12: End While
13: x \leftarrow xact
14: iter \leftarrow cont
15: err \leftarrow E
16: Return x, iter, err
```

Algorithm 5 SEC

```
Input: f, x0, x1, tol, Nmax
Output: x, iter, err
 1: f0 \leftarrow f(x0)
 2: f1 \leftarrow f(x1)
 3: E \leftarrow 1000
 4: cont \leftarrow 1
 5: While (E > tol) & (cont < Nmax) do
 6:
       xact \leftarrow x1 - f1 * (x1 - x0)/(f1 - f0)
       fact \leftarrow f(xact)
 7:
       E \leftarrow |xact - x1|
 8:
 9:
       cont \leftarrow cont + 1
       x0 \leftarrow x1
10:
       f0 \leftarrow f1
11:
12:
       x1 \leftarrow xact
       f1 \leftarrow fact
14: End While
15: x \leftarrow xact
16: iter \leftarrow cont
17: err \leftarrow E
18: Return x, xact, err
```

Algorithm 6 MULTIPLE ROOTS

```
Input: f, f', f'', x0, tol, Nmax
Output: x, iter, err
 1: xant \leftarrow x0
 2: fant \leftarrow f(act)
 3: E \leftarrow 1000
 4:\ cont \leftarrow 0
 5: While (E > tol) & (cont < Nmax) do
       xact \leftarrow (xant - fant) * f''(xant)/(f'(xant)^2 - fant * f''(xant))
       fact \leftarrow f(xact)
       E \leftarrow |xact - xant|
 8:
       cont \leftarrow cont + 1
       xant \leftarrow xact
11:
       fant \leftarrow fact
12: End While
13: x \leftarrow xact
14: iter \leftarrow cont
15: err \leftarrow E
16: Return x, iter, err
```

Algorithm 7 GAUSSIAN ELIMINATION

```
Input: A, b

Output: x

1: n \leftarrow size(A, 1)

2: M \leftarrow [Ab]

3: For i \leftarrow 0, n - 1 do

4: For j \leftarrow i + 1, n do

5: if M(j, i) \neq 0 Then

6: M(j, i : n + 1) \leftarrow M(j, i : n + 1) - (M(j, i)/M(i, i)) * M(i, i : n + 1)

7: End if

8: End For

9: End For

10: x \leftarrow backsubstitution(M)

11: Return x
```

Algorithm 8 GAUSSIAN ELIMINATION WITH PARCIAL PIVOTING

```
Input: A, b
Output: x
  n \leftarrow size(A,1)
  M \leftarrow [Ab]
  For i \leftarrow 1, n-1 do
     [aux0, aux] \leftarrow max|M(i+1:n,1)|
     if aux0 > |M(i,i)| Then
       aux2 \leftarrow M(i + aux, i : n + 1)
       M(aux+i,i:n+1) \leftarrow M(i,i:n+1)
        M(i, i: n+1) \leftarrow aux2
     End if
     For j \leftarrow i+1, n do
       if M(j,i) \neq 0 Then
          M(j, i: n+1) \leftarrow M(j, i: n+1) - (M(j, i)/M(i, i)) * M(i, i: n+1)
       End if
     End For
  End For
  x \leftarrow backsubstitution(M)
  \textbf{Return} \ \ x
```

Algorithm 9 GAUSSIAN ELIMINATION WITH TOTAL PIVOTING

```
Input: A, b
Output: x
  n \leftarrow size(A, 1)
  M \leftarrow [Ab]
  cambi \leftarrow []
  For i \leftarrow 1, n-1 do
     [a,b] \leftarrow find(|M(i:n,i:n)|) \equiv max(|M(i:n,i:n)|) Cambio de columna
     if b(1) + i - 1 \neq i Then
        cambi \leftarrow + = [ib(1) + i + 1]
        aux2 \leftarrow M(:, b(1) + i + 1)
        M(:,b(1)+i-1) \leftarrow
        M(:,i) \leftarrow
     End if Cambio de fila
     if a(1) + i \neq i Then
        aux2 \leftarrow M(i + a(1) - 1, i : n + 1)
        M(a(1) + i - 1, i : n + 1) \leftarrow M(i, i : n + 1)
        M(i, i: n+1) \leftarrow aux2
     End if
     For j \leftarrow i+1, n do
       if M(j,i) \neq 0 Then
          M(j, i: n+1) \leftarrow M(j, i: n+1) - (M(j, i)/M(i, i)) * M(i, i: n+1)
        End if
     End For
  End For
  x \leftarrow backsubstitution(M)
  Return x
```