

EAFIT UNIVERSITY
DEPARTMENT OF INFORMATICS AND SYSTEMS
PROJECT CHOICE

Second Report
September 27, 2022

Course: Numerical Analysis
Teacher: Edwar Samir Posada Murillo
Semester: 2022-2

Jacobo Rave Londoño
Juan David Echeverri Villada
Kevin Alejandro Sossa Chavarria
Juan Sebastián Guerra Hernández

Next, you will find recreated methods in pseudocode:

1: Bisection

```

1: function Bisection(f,left,right,tol,niter)
2:    $fRight \leftarrow f(right)$ 
3:    $fLeft \leftarrow f(left)$ 
4:   if  $fRight = 0$  then
5:     WRITE "right is a root"
6:   else if  $fLeft = 0$  then
7:     WRITE "left is a root"
8:   else if  $fLeft * fRight < 0$  then
9:      $mid \leftarrow (left + right)/2$ 
10:     $fmid \leftarrow f(mid)$ 
11:     $counter \leftarrow 1$ 
12:     $error \leftarrow tol + 1$ 
13:    while  $error > tol$  AND  $fmid \neq 0$  AND  $counter < niter$  do
14:      if  $fLeft * fmid < 0$  then
15:         $right \leftarrow mid$ 
16:         $fRight \leftarrow fmid$ 
17:      else
18:         $left \leftarrow mid$ 
19:         $fLeft \leftarrow fmid$ 
20:      end if
21:       $aux \leftarrow mid$ 
22:       $mid \leftarrow (right + left)/2$ 
23:       $fmid \leftarrow f(mid)$ 
24:       $error \leftarrow |mid - aux|$ 
25:       $Counter \leftarrow counter + 1$ 
26:    end while
27:    if  $fmid = 0$  then
28:      WRITE "mid is a root"
29:    else if  $error < tol$  then
30:      WRITE "mid is an approximation with tolerance"
31:    else
32:      WRITE "The method fails in niter iterations"
33:    end if
34:  end if
35: end function

```

2: Newton

```

1: function Newton(f, fder, tol, x0, niter)
2:    $fx \leftarrow f(x0)$ 
3:    $dfx \leftarrow fder(x0)$ 
4:    $counter \leftarrow 1$ 
5:    $error \leftarrow tol + 1$ 
6:   while  $error > tolerance$  &  $fx \neq 0$  &  $counter < niter$  do
7:      $x1 \leftarrow x0 - (fx/dfx)$ 
8:      $fx \leftarrow f(x1)$ 
9:      $dfx \leftarrow fder(x1)$ 
10:     $error \leftarrow |x1 - x0|$ 
11:     $x1 \leftarrow x0$ 
12:     $counter \leftarrow counter + 1$ 
13:  end while
14:  if  $fx = 0$  then
15:    WRITE "x0 is a root of f"
16:  else if  $error < tolerance$  then
17:    WRITE "x1 is a root approximation with tolerance tol"
18:  else

```

```

19:     WRITE "The method failed at niter iteration"
20:   end if
21: end function

```

3: Incremental Search

```

1: function IncrementalSearch(f,x0,delta,niter)
2:    $fx0 \leftarrow f(x0)$ 
3:   if  $fx0 = 0$  then
4:     answer  $\leftarrow$  x0 is a root
5:   else
6:      $x1 \leftarrow x0 + \text{delta}$ 
7:     counter  $\leftarrow 1$ 
8:      $fx1 \leftarrow f(x1)$ 
9:     while  $fx0 * fx1 > 0$  AND counter  $>$  niter do
10:       $x0 \leftarrow x1$ 
11:       $fx0 \leftarrow fx1$ 
12:       $x1 \leftarrow x0 + \text{delta}$ 
13:       $fx1 \leftarrow f(x1)$ 
14:      counter  $\leftarrow$  counter + 1
15:    end while
16:    if  $fx1 = 0$  then
17:      answer  $\leftarrow$  x1 is a root
18:    else if  $fx0 * fx1 < 0$  then
19:      answer  $\leftarrow$  There is at least one root between x0 and x1
20:    else
21:      answer  $\leftarrow$  The method fails in niter iterations
22:    end if
23:  end if
24: end function

```

4: Fixed Point

```

1: function Fixed point(f, g, tol, x0, niter)
2:    $fx \leftarrow f(x0)$ 
3:    $gx \leftarrow \text{convergent form of } f(x)$ 
4:   counter  $\leftarrow 1$ 
5:   error  $\leftarrow \text{tol} + 1$ 
6:   while error  $>$  tolerance &  $f(x0) \neq 0$  & counter  $<$  niter do
7:      $x1 \leftarrow g(x0)$ 
8:     error  $\leftarrow |x1 - x0|$ 
9:      $x0 \leftarrow x1$ 
10:    counter  $\leftarrow$  counter + 1
11:  end while
12:  if  $fx = 0$  then
13:    WRITE "x1 is a root approximation with tolerance tol"
14:  else if error  $<$  toleranc then
15:    WRITE "x1 is a root approximation with tolerance tol"
16:  else
17:    WRITE "The method failed at niter iteration"
18:

```

5: Multiple Root

```

1: function MultipleRoot(f,f1,f2,x0,tolerance,nMax)
2:    $xi \leftarrow x0$ 
3:    $fxi \leftarrow f(xi)$ 
4:   if  $fxi = 0$  then
5:     WRITE "A root was found: xi"
6:   else
7:     counter  $\leftarrow 0$ 

```

```

8:    $f1xi \leftarrow f1(xi)$ 
9:    $f2xi \leftarrow f2(xi)$ 
10:   $error \leftarrow tolerance + 1$ 
11:   $det \leftarrow (f1xi^2) - (fxi * f2xi)$ 
12:  while  $fxi \neq 0$  &  $error > tolerance$  &  $counter < nMax$  &  $det \neq 0$  do
13:     $xiAux \leftarrow xi$ 
14:     $x1 \leftarrow x1 - ((fxi * f1xi) / ((f1xi^2) - (fxi * f2xi)))$ 
15:     $fxi \leftarrow f(xi)$ 
16:     $f1xi \leftarrow f1(xi)$ 
17:     $f2xi \leftarrow f2(xi)$ 
18:     $error \leftarrow |xi - xiAux|$ 
19:     $det \leftarrow (f1xi^2) - (fxi * f2xi)$ 
20:     $counter \leftarrow counter + 1$ 
21:  end while
22:  if  $f1 = 0$  then
23:    WRITE "A root was found:  $xi$ "
24:  else if  $error \leq tolerance$  then
25:    WRITE "One approach is:  $xi$ "
26:  else if  $det = 0$  then
27:    WRITE "Method failure"
28:  else
29:    WRITE "The method fails with the maximum number of iterations given"
30:  end if
31: end if
32:  $x \leftarrow xi$ 
33: end function=0

```

6: Secant

```

1: function Secant( $x0, x1, tol, iter, f$ )
2:    $y0 \leftarrow f(x0)$ 
3:   if  $y0 = 0$  then
4:     WRITE " $x0$  is a root of  $f$ "
5:   else
6:      $y1 \leftarrow f(x1)$ 
7:      $d \leftarrow y1 - y0$ 
8:      $error \leftarrow tol + 1$ 
9:      $cont \leftarrow 0$ 
10:    while  $y1 \neq 0$  &  $error > tol$  &  $cont < iter$  &  $d \neq 0$  do
11:       $x2 \leftarrow x1 - ((y1 * (x1 - x0)) / (d))$ 
12:       $error \leftarrow |x2 - x1|$ 
13:       $x0 \leftarrow x1$ 
14:       $y0 \leftarrow y1$ 
15:       $y1 \leftarrow f(x2)$ 
16:       $x1 \leftarrow x2$ 
17:       $d \leftarrow y1 - y0$ 
18:       $counter \leftarrow counter + 1$ 
19:    end while
20:    if  $y1 = 0$  then
21:      WRITE " $x1$  is a root of  $f$ "
22:    else
23:      if  $error < tol$  then
24:        WRITE " $x1$  is an approximation to a root with a tolerance  $tol$ "
25:      else
26:        if  $d = 0$  then
27:          WRITE "denominator is zero, FAILURE"
28:        else
29:          WRITE "failure in  $iter$  iterations"
30:        end if
31:      end if

```

```

32:     end if
33: end if
34: end function

```

7: False Rule

```

1: function falseRule(Fun, Xi, Xs, Tol, Iter)
2:    $Y_i \leftarrow f(X_i)$ 
3:    $Y_s \leftarrow f(X_s)$ 
4:   if  $Y_i = 0$  then
5:     WRITE Xi is the root
6:   else
7:     if  $Y_s = 0$  then
8:       WRITE Xs is the root
9:     else
10:      if  $Y_i * Y_s < 0$  then
11:         $X_m \leftarrow (X_i) - ((f(X_i) * (X_i - X_s)) / (f(X_i) - f(X_s)))$ 
12:         $Y_m \leftarrow f(X_m)$ 
13:         $Error \leftarrow Tol + 1$ 
14:         $Cont \leftarrow 1$ 
15:        while  $Y_m \neq 0 \ \& \ Error > Tol \ \& \ Cont < Iter$  do
16:          if  $Y_i * Y_m < 0$  then
17:             $X_s \leftarrow X_m$ 
18:             $Y_s \leftarrow Y_m$ 
19:          else
20:             $X_i \leftarrow X_m$ 
21:             $Y_i \leftarrow Y_m$ 
22:          end if
23:           $X_{aux} \leftarrow X_m$ 
24:           $X_m \leftarrow (X_i) - ((f(X_i) * (X_i - X_s)) / (f(X_i) - f(X_s)))$ 
25:           $Y_m \leftarrow f(X_m)$ 
26:           $Error \leftarrow |X_m - X_{aux}| / X_m$ 
27:           $Cont \leftarrow Cont + 1$ 
28:        end while
29:        if  $Y_m = 0$  then
30:          WRITE "Xm is a root of f"
31:        else
32:          if  $Error < Tol$  then
33:            WRITE "Xm is an approximation to a root with a tolerance Tol"
34:          else
35:            WRITE "failure in Iter iterations"
36:          end if
37:        end if
38:        WRITE Iterations | Xi | Xs | Xm | Ym | Error
39:      else
40:        WRITE "The interval is inadequate"
41:      end if
42:    end if
43:  end if
44: end function

```

8: GaussSimple

```

1: function GaussSimple(A,b,n,delta,niter)
2:    $A \leftarrow Concat(A, b)$ 
3:   for  $i \leftarrow 1, n - 1$  do
4:     if  $A_{i,i} = 0$  then
5:       WRITE Mathematical Error! Stop
6:     end if
7:     for  $j \leftarrow i + 1, n$  do
8:        $ratio \leftarrow A_{j,i} / A_{i,i}$ 

```

```

9:         for  $k \leftarrow 1, n+1$  do
10:              $A_{j,k} \leftarrow A_{j,k} - \text{ratio} * A_{i,k}$ 
11:         end for
12:     end for
13: end for
14:  $X_n \leftarrow A_{n,n+1} / A_{n,n}$ 
15: for  $i \leftarrow n-1, 1, \text{step} = -1$  do
16:      $X_i \leftarrow A_{i,n+1}$ 
17:     for  $j \leftarrow i+1, n$  do
18:          $X_i \leftarrow X_i - A_{i,j} * X_j$ 
19:     end for
20:      $X_i \leftarrow X_i / A_{i,i}$ 
21: end for
22: WRITE "Answer vector"
23: for  $i \leftarrow 1, n$  do
24:     WRITE  $X_i$ 
25: end for
26: end function

```

9: GaussPartial

```

1: function GaussPartial(A,b,n,delta,niter)
2:      $A \leftarrow \text{Concat}(A,b)$ 
3:     for  $i \leftarrow 1, n-1$  do
4:         WRITE changeRows(A,i)
5:         if  $A_{i,i} = 0$  then
6:             WRITE Mathematical Error! Stop
7:         end if
8:         for  $j \leftarrow i+1, n$  do
9:              $\text{ratio} \leftarrow A_{j,i} / A_{i,i}$ 
10:            for  $k \leftarrow 1, n+1$  do
11:                 $A_{j,k} \leftarrow A_{j,k} - \text{ratio} * A_{i,k}$ 
12:            end for
13:        end for
14:    end for
15:     $X_n \leftarrow A_{n,n+1} / A_{n,n}$ 
16:    for  $i \leftarrow n-1, 1, \text{step} = -1$  do
17:         $X_i \leftarrow A_{i,n+1}$ 
18:        for  $j \leftarrow i+1, n$  do
19:             $X_i \leftarrow X_i - A_{i,j} * X_j$ 
20:        end for
21:         $X_i \leftarrow X_i / A_{i,i}$ 
22:    end for
23:    WRITE "Answer vector"
24:    for  $i \leftarrow 1, n$  do
25:        WRITE  $X_i$ 
26:    end for
27: end function

```

10: GaussTotal

```

1: function GaussTotal(A,b,n,delta,niter)
2:      $A \leftarrow \text{Concat}(A,b)$ 
3:     for  $i \leftarrow 1, n-1$  do
4:         WRITE changeRowsAndColumns(A,i)
5:         if  $A_{i,i} = 0$  then
6:             WRITE Mathematical Error! Stop
7:         end if
8:         for  $j \leftarrow i+1, n$  do
9:              $\text{ratio} \leftarrow A_{j,i} / A_{i,i}$ 
10:            for  $k \leftarrow 1, n+1$  do

```

```

11:          $A_{j,k} \leftarrow A_{j,k} - \text{ratio} * A_{i,k}$ 
12:     end for
13: end for
14: end for
15:  $X_n \leftarrow A_{n,n+1} / A_{n,n}$ 
16: for  $i \leftarrow n-1, 1, \text{step} = -1$  do
17:      $X_i \leftarrow A_{i,n+1}$ 
18:     for  $j \leftarrow i+1, n$  do
19:          $X_i \leftarrow X_i - A_{i,j} * X_j$ 
20:     end for
21:      $X_i \leftarrow X_i / A_{i,i}$ 
22: end for
23: WRITE "Answer vector"
24: for  $i \leftarrow 1, n$  do
25:     WRITE  $X_i$ 
26: end for
27: end function

```

Repository

The repository with the evidence related to the project will be: [🔗 NumericalAnalysisProject](#)

Description

This project aims to develop a web app to calculate data using different numerical methods as well as an API that let users make request to solve their problems. Also, the web app will have the option of visualising the data in a 2D graph.

Added Values

- The project will be done in English
- The project will have its documentation in \LaTeX
- The numerical algorithms can be found in multiple programming languages.
- The project will have extra numerical methods

Members signatures

1: Jacobo Rave



2: David Echeverri



3: Kevin Sossa



4: Sebastián Guerra

