**EAFIT UNIVERSITY**

DEPARTMENT OF SYSTEMS AND COMPUTING

# Jaba Methods Calculator

*Subject: Workshop 2*

Responsible teacher:
Edwar Samir Posada Murillo

Deadline: 31/11/2022

## Objective

Principal and first methods that we are going to introduce in the Jaba calculator, such as: Incremental search, Bisection, false position, fixed point, Newton, secant, multiple roots, Gauss simple, Gauss partial, Gauss total.

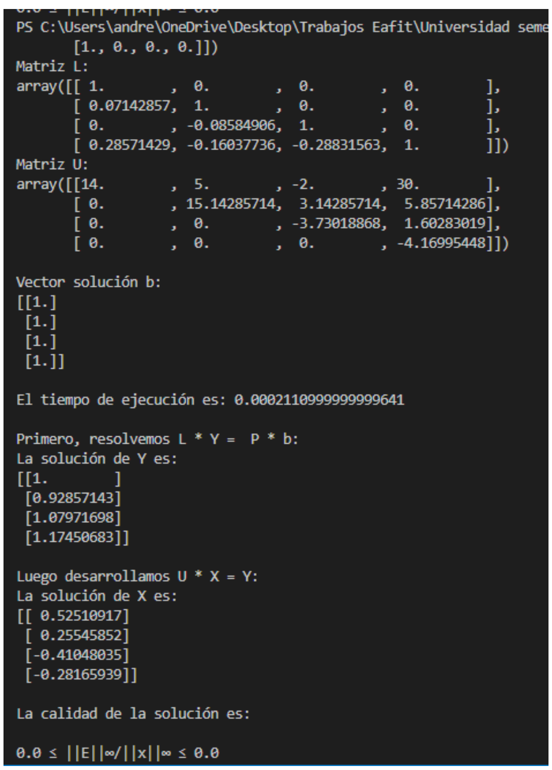
## Tested methods

**Functions to use:**
$A = [[4, -1, 0, 3], [1, 15.5, 3, 8], [0, -1.3, -4, 1.1], [14, 5, -2, 30]]$
$b = [[1], [1], [1], [1]]$
$Xo = [[0], [0], [0], [0]]$
$Tol = 1 * e^-7$
$Nmax = 100$

| | | | | |
|---|---|---|---|---|
| x | -1 | 0 | 3 | 4 |
| y | 15.5 | 3 | 8 | 1 |

**Gaussian LU**

**LU partial pivoting**



```
PS C:\Users\andre\OneDrive\Desktop\Trabajos Eafit\Universidad seme
       [1., 0., 0., 0.]])
Matriz L:
array([[ 1.        ,  0.        ,  0.        ,  0.        ],
       [ 0.07142857,  1.        ,  0.        ,  0.        ],
       [ 0.        , -0.08584906,  1.        ,  0.        ],
       [ 0.28571429, -0.16037736, -0.28831563,  1.        ]])
Matriz U:
array([[14.        ,  5.        , -2.        , 30.        ],
       [ 0.        , 15.14285714,  3.14285714,  5.85714286],
       [ 0.        ,  0.        , -3.73018868,  1.60283019],
       [ 0.        ,  0.        ,  0.        , -4.16995448]])

Vector solución b:
[[1.]
 [1.]
 [1.]
 [1.]]

El tiempo de ejecución es: 0.0002110999999999641

Primero, resolvemos L * Y =  P * b:
La solución de Y es:
[[1.        ]
 [0.92857143]
 [1.07971698]
 [1.17450683]]

Luego desarrollamos U * X = Y:
La solución de X es:
[[ 0.52510917]
 [ 0.25545852]
 [-0.41048035]
 [-0.28165939]]

La calidad de la solución es:

0.0 ≤ ||E||∞/||x||∞ ≤ 0.0
```

**Crout**

```
PS C:\Users\andre\OneDrive\Desktop\Trabajos Eafit\Universidad semestre 5\Analisis numerico\numerical-analysis> &
ktop/Trabajos Eafit/Universidad semestre 5/Analisis numerico/numerical-analysis/methods/python-methods/crout.py"
Matriz L
[[ 1.          0.          0.          0.        ]
 [ 0.25        1.          0.          0.        ]
 [ 0.         -0.08253968  1.          0.        ]
 [ 3.5         0.53968254  0.96446701  1.        ]]
Matriz U
[[ 4.         -1.          0.          3.        ]
 [ 0.         15.75        3.          7.25      ]
 [ 0.          0.         -3.75238095  1.6984127 ]
 [ 0.          0.          0.         13.94923858]]
Solucion
[[ 0.52510917]
 [ 0.25545852]
 [-0.41048035]
 [-0.28165939]]
Verificacion Ax=B:
[[1.]
 [1.]
 [1.]
 [1.]]
```

**Doolittle**

```
PS C:\Users\andre\OneDrive\Desktop\Trabajos Eafit\Universidad semestre 5\Analisis numerico\numerical-analysis> &
ktop/Trabajos Eafit/Universidad semestre 5/Analisis numerico/numerical-analysis/methods/python-methods/doolittle.
Matriz A:
[[4, -1, 0, 3], [1, 15.5, 3, 8], [0, -1.3, -4, 1.1], [14, 5, -2, 30]]
Matriz B:
[[1], [1], [1], [1]]
Matriz L
[[ 1.          0.          0.          0.        ]
 [ 0.25        1.          0.          0.        ]
 [ 0.         -0.08253968  1.          0.        ]
 [ 3.5         0.53968254  0.96446701  1.        ]]
Matriz U
[[ 4.         -1.          0.          3.        ]
 [ 0.         15.75        3.          7.25      ]
 [ 0.          0.         -3.75238095  1.6984127 ]
 [ 0.          0.          0.         13.94923858]]
Solucion
[[ 0.52510917]
 [ 0.25545852]
 [-0.41048035]
 [-0.28165939]]
Verificacion Ax=B:
[[1.]
 [1.]
 [1.]
 [1.]]
```

**Cholesky**

```
PS C:\Users\andre\OneDrive\Desktop\Trabajos Eafit\Universidad semestre 5\Analisis numerico\numerical-analysis> &
ktop/Trabajos Eafit/Universidad semestre 5/Analisis numerico/numerical-analysis/methods/python-methods/cholesky.p
[[ 2.  0.  0.]
 [ 6.  1.  0.]
 [-8.  5.  3.]]
[[  4.  12. -16.]
 [ 12.  37. -43.]
 [-16. -43.  98.]]
[ 0.38981311  0.12912144 -0.33139314 -0.143377  ]
```

**Jacobi**

```
PS C:\Users\andre\OneDrive\Desktop\Trabajos Eafit\Universidad semestre 5\Analisis numerico\numerical-analysis> &
ktop/Trabajos Eafit/Universidad semestre 5/Analisis numerico/numerical-analysis/methods/python-methods/jacobi.py"

Matriz A:
array([[ 4. , -1. ,  0. ,  3. ],
       [ 1. , 15.5,  3. ,  8. ],
       [ 0. , -1.3, -4. ,  1.1],
       [14. ,  5. , -2. , 30. ]])

Vector b:
array([[1.],
       [1.],
       [1.],
       [1.]])

Para la iteración 1: X = [[ 0.25       0.0645161 -0.25       0.0333333]]        Error: 0.25
Para la iteración 2: X = [[ 0.241129   0.0795699 -0.2618011 -0.1107527]]        Error: 0.14408602150537633
Para la iteración 3: X = [[ 0.352957   0.1567933 -0.3063172 -0.1099086]]        Error: 0.11182795698924733
Para la iteración 4: X = [[ 0.3716298  0.1577589 -0.3311827 -0.1779333]]        Error: 0.06802468493467452
Para la iteración 5: X = [[ 0.4228897  0.1964764 -0.3502033 -0.1884659]]        Error: 0.05125992889351366
Para la iteración 6: X = [[ 0.4404685  0.2022869 -0.365683  -0.2201082]]        Error: 0.031642257318578615
Para la iteración 7: X = [[ 0.4656528  0.2204804 -0.376273  -0.230312 ]]        Error: 0.025184318823917695
Para la iteración 8: X = [[ 0.4778541  0.2261717 -0.3849919 -0.2458029]]        Error: 0.01549092296161414
Para la iteración 9: X = [[ 0.4908951  0.2350674 -0.3911016 -0.2530267]]        Error: 0.013041039930310916
Para la iteración 10: X = [[ 0.4985369  0.239137  -0.3959792 -0.2610024]]       Error: 0.007975744291039588
Para la iteración 11: X = [[ 0.505536   0.2437045 -0.3994952 -0.265572 ]]       Error: 0.006999195277753278
Para la iteración 36: X = [[ 0.5250924  0.2554484 -0.4104711 -0.281646 ]]       Error: 5.469097211441287e-06
Para la iteración 37: X = [[ 0.5250966  0.2554509 -0.4104734 -0.2816493]]       Error: 4.121568323633973e-06
Para la iteración 38: X = [[ 0.5250997  0.2554528 -0.4104751 -0.2816518]]       Error: 3.1053813239756423e-06
Para la iteración 39: X = [[ 0.525102   0.2554542 -0.4104764 -0.2816536]]       Error: 2.3401223205876676e-06
Para la iteración 40: X = [[ 0.5251038  0.2554552 -0.4104774 -0.2816551]]       Error: 1.763228152595886e-06
Para la iteración 41: X = [[ 0.5251051  0.2554561 -0.4104781 -0.2816561]]       Error: 1.328675495959608e-06
Para la iteración 42: X = [[ 0.5251061  0.2554567 -0.4104787 -0.2816569]]       Error: 1.0011492709294245e-06
Para la iteración 43: X = [[ 0.5251069  0.2554571 -0.4104791 -0.2816575]]       Error: 7.544001092707475e-07
Para la iteración 44: X = [[ 0.5251074  0.2554575 -0.4104794 -0.281658 ]]       Error: 5.684435093744611e-07
Para la iteración 45: X = [[ 0.5251079  0.2554577 -0.4104796 -0.2816583]]       Error: 4.28337356805919e-07
Para la iteración 46: X = [[ 0.5251082  0.2554579 -0.4104798 -0.2816586]]       Error: 3.2275630768019425e-07
Para la iteración 47: X = [[ 0.5251084  0.2554581 -0.4104799 -0.2816588]]       Error: 2.432041344979652e-07
Para la iteración 48: X = [[ 0.5251086  0.2554582 -0.41048   -0.2816589]]       Error: 1.8325742001668033e-07
Para la iteración 49: X = [[ 0.5251087  0.2554583 -0.4104801 -0.281659 ]]       Error: 1.380881480272933e-07
Para la iteración 50: X = [[ 0.5251089  0.2554583 -0.4104802 -0.2816591]]       Error: 1.0405143124891936e-07
Para la iteración 51: X = [[ 0.5251089  0.2554584 -0.4104802 -0.2816592]]       Error: 7.840470261921695e-08

El vector 'x' es:
[[ 0.52510893]
 [ 0.25545837]
 [-0.41048022]
 [-0.2816592 ]]

El numero de iteraciones es: 51
```

**Gauss-seidel**

```
PS C:\Users\andre\OneDrive\Desktop\Trabajos Eafit\Universidad semestre 5\Analisis numerico\numerical-analysis> & 
ktop/Trabajos Eafit/Universidad semestre 5/Analisis numerico/numerical-analysis/methods/python-methods/gauss-seide
Matriz A:
 [[ 4.   -1.    0.    3. ]
 [ 1.   15.5  3.    8. ]
 [ 0.   -1.3 -4.    1.1]
 [14.    5.   -2.   30. ]]

Vector b:
 [1, 1, 1, 1]

La solución converge en x: [0.26190476191775647 ,0.04761904761820926 ]
Luego de 6 iteraciones
```

**SOR**

```
PS C:\Users\andre\OneDrive\Desktop\Trabajos Eafit\Universidad semestre 5\Analisis numerico\numerical-analysis>
ktop/Trabajos Eafit/Universidad semestre 5/Analisis numerico/numerical-analysis/methods/python-methods/SOR.py"
[0 0 0 0]
```

**Vandermonde**

```
PS C:\Users\andre\OneDrive\Desktop\Trabajos Eafit\Universidad semestre 5\Analisis numerico\numerical-analysis> &
ktop/Trabajos Eafit/Universidad semestre 5/Analisis numerico/numerical-analysis/methods/python-methods/vandermond
Vandermonde:
[[-1.  1. -1.  1.]
 [ 0.  0.  0.  1.]
 [27.  9.  3.  1.]
 [64. 16.  4.  1.]]
Polynomial coefficients:
[-1.14166667  5.825      -5.53333333  3.        ]
Interpolation polynomial:
-1.14166666666667*x**3 + 5.825*x**2 - 5.53333333333333*x + 3.0


                    3          2
- 1.14166666666667⋅x  + 5.825⋅x  - 5.53333333333333⋅x + 3.0
```

**Newton**

```
PS C:\Users\andre\OneDrive\Desktop\Trabajos Eafit\Universidad semestre 5\Analisis numerico\numerical-analysis> &
ktop/Trabajos Eafit/Universidad semestre 5/Analisis numerico/numerical-analysis/methods/python-methods/divided-di
Divided differences Newton
[['i   ', 'xi  ', 'fi  ', 'F[1]', 'F[2]', 'F[3]', 'F[4]']]
[[  0.      -1.      15.5    -12.5      3.5417  -1.1417   0.    ]
 [  1.       0.       3.       1.6667  -2.1667   0.       0.    ]
 [  2.       3.       8.      -7.       0.       0.       0.    ]
 [  3.       4.       1.       0.       0.       0.       0.    ]]
dDividida:
[-12.5      3.5417  -1.1417   0.    ]
Polynomial:
-1.14166666666667*x*(x - 3)*(x + 1) + 3.54166666666667*x*(x + 1) - 12.5*x + 3.0
Simplified polynomial:
-1.14166666666667*x**3 + 5.825*x**2 - 5.53333333333333*x + 3.0
```

**Lagrange**

```
PS C:\Users\andre\OneDrive\Desktop\Trabajos Eafit\Universidad semestre 5\Analisis numerico\numerical-analysis> &
ktop/Trabajos Eafit/Universidad semestre 5/Analisis numerico/numerical-analysis/methods/python-methods/lagrange.p
    fi values:  [15.5  3.   8.   1. ]
dividers L(i):  [-20.  12. -12.  20.]

Lagrange polynomial, expressions
-0.775*x*(x - 4)*(x - 3) - 0.666666666666667*x*(x - 4)*(x + 1) + 0.05*x*(x - 3)*(x + 1) + 0.25*(x - 4)*(x - 3)*(x

Lagrange polynomial
-1.14166666666667*x**3 + 5.825*x**2 - 5.53333333333333*x + 3.0
```

**Lineal Spline**

```
PS C:\Users\andre\OneDrive\Desktop\Trabajos Eafit\Universidad semestre 5\Analisis numerico\numerical-analysis> &
ktop/Trabajos Eafit/Universidad semestre 5/Analisis numerico/numerical-analysis/methods/python-methods/lineal spl
3.0 - 12.5*x
1.66666666666667*x + 3.0
29.0 - 7.0*x
```

**Cuadratic Spline**

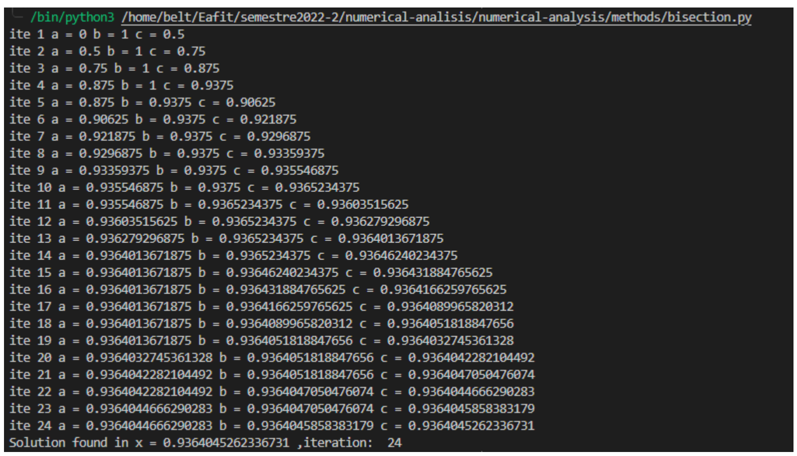
```
PS C:\Users\andre\OneDrive\Desktop\Trabajos Eafit\Universidad semestre 5\Analisis numerico\numerical-analysis> &
ktop/Trabajos Eafit/Universidad semestre 5/Analisis numerico/numerical-analysis/methods/python-methods/quadratic-
-1 <= x <= 0
1.77635683940025e-14*x**2 - 12.5*x + 3.0
0 <= x <= 3
4.72222222222222*x**2 - 12.5*x + 3.0
3 <= x <= 4
-22.8333333333333*x**2 + 152.833333333333*x - 245.0
```

**Cubic Spline**

```
PS C:\Users\andre\OneDrive\Desktop\Trabajos Eafit\Universidad semestre 5\Analisis numerico\numerical-analysis> &
ktop/Trabajos Eafit/Universidad semestre 5/Analisis numerico/numerical-analysis/methods/python-methods/cubic-spli
Polinomios por tramos:
 x = [-1,0]
2.53333333333333*x**3 + 7.6*x**2 - 7.43333333333333*x + 3.0
 x = [0,3]
-1.52222222222222*x**3 + 7.6*x**2 - 7.43333333333333*x + 3.0
 x = [3,4]
2.03333333333333*x**3 - 24.4*x**2 + 88.5666666666667*x - 93.0
```

## Project web address (repository)

https://github.com/bened18/numerical-analysis



```
/bin/python3 /home/belt/Eafit/semestre2022-2/numerical-analisis/numerical-analysis/methods/bisection.py
ite 1 a = 0 b = 1 c = 0.5
ite 2 a = 0.5 b = 1 c = 0.75
ite 3 a = 0.75 b = 1 c = 0.875
ite 4 a = 0.875 b = 1 c = 0.9375
ite 5 a = 0.875 b = 0.9375 c = 0.90625
ite 6 a = 0.90625 b = 0.9375 c = 0.921875
ite 7 a = 0.921875 b = 0.9375 c = 0.9296875
ite 8 a = 0.9296875 b = 0.9375 c = 0.93359375
ite 9 a = 0.93359375 b = 0.9375 c = 0.935546875
ite 10 a = 0.935546875 b = 0.9375 c = 0.9365234375
ite 11 a = 0.935546875 b = 0.9365234375 c = 0.93603515625
ite 12 a = 0.93603515625 b = 0.9365234375 c = 0.936279296875
ite 13 a = 0.936279296875 b = 0.9365234375 c = 0.9364013671875
ite 14 a = 0.9364013671875 b = 0.9365234375 c = 0.93646240234375
ite 15 a = 0.9364013671875 b = 0.93646240234375 c = 0.936431884765625
ite 16 a = 0.9364013671875 b = 0.936431884765625 c = 0.9364166259765625
ite 17 a = 0.9364013671875 b = 0.9364166259765625 c = 0.9364089965820312
ite 18 a = 0.9364013671875 b = 0.9364089965820312 c = 0.9364051818847656
ite 19 a = 0.9364013671875 b = 0.9364051818847656 c = 0.9364032745361328
ite 20 a = 0.9364032745361328 b = 0.9364051818847656 c = 0.9364042282104492
ite 21 a = 0.9364042282104492 b = 0.9364051818847656 c = 0.9364047050476074
ite 22 a = 0.9364042282104492 b = 0.9364047050476074 c = 0.9364044666290283
ite 23 a = 0.9364044666290283 b = 0.9364047050476074 c = 0.9364045858383179
ite 24 a = 0.9364044666290283 b = 0.9364045858383179 c = 0.9364045262336731
Solution found in x = 0.9364045262336731 ,iteration:  24
```

Figure 1: Caption