Method of False Position (or Regula Falsi Method)

nalib

The method of **false position** is a hybrid of bisection and the secant method. It incorporates the bracketing of the **bisection** method with the **secant** method. You begin with two initial approximations p_0 and p_1 which bracket the root and have $f(p_0) f(p_1) < 0$. We stay with our original problem.

```
> restart;
> libname:="c:/nalib",libname;
```

libname := "/nalib", "/Library/Frameworks/Maple.framework/Versions/15/lib",
"/Library/Frameworks/Maple.framework/Versions/15/toolbox/NAG/lib"

```
> with(numanal);
```

[*SOR*, *SOR_dir*, *adaptq*, *adaptq_dir*, *bezier*, *bezier_dir*, *bisection*, *bisection_dir*, *chop*, *chop_dir*, *clamped_spline*, *clamped_spline_dir*, *divided_diff*, *divided_diff_dir*, *extrap*, *extrap_dir*, *falseposition*, *falseposition_dir*, *fixedpoint*, *fixedpoint_dir*, *gaussseidel*, *gaussseidel_dir*, *hermite*, *hermite_dd*, *hermite_dd_dir*, *hermite_dir*, *horner*, *horner_dir*, *jacobi*, *jacobi_dir*, *muller*, *muller_dir*, *natural_spline*, *natural_spline_dir*, *newton*, *newton_dir*, *romberg*, *romberg_dir*, *secant*, *secant_dir*, *steffensen*, *steffensen_dir*]

We continue to work with the following function, which has roots near .12, .3, and 4.8. Suppose we wish to find all the roots of the following function accurate to within 10^{-6}.

```
> f:=4*x^3-20*x^2+3*x+2+ln(x);
```

$$f := 4\,x^3 - 20\,x^2 + 3\,x + 2 + \ln(x)$$

We check the directions and then find the first root. Let's bracket the root with .1 and .4 as initial approximations.

```
> falseposition_dir();
falseposition returns a root of the given function.

The arguments for falseposition are:
(1)function expression in x
(2)first initial approximation
(3)second initial approximation
(4)tolerance
(5)maximum number of iterations
(6)variable for returning root

If assigning the result to a variable, have the
variable and the 6th argument the same.

If r is the variable for returning the root and has already been given
a value, the procedure should be preceded by the statement:
r:='r'
> r1:=falseposition(f,.1,.4,.000001,100,r1);
Error, (in falseposition) You must have f(p[0])*f(p[1]) < 0
```

OK, we missed on the second requirement, namely that $f(p_0) f(p_1) < 0$. Let's try again and bracket the root with .1 and .2.

```
> r1:='r1';
```

$$r1 := r1$$

```
> r1:=falseposition(f,.1,.2,.000001,100,r1);
  i   p                 f(p)
  -   -                 ----
  0   1.00000000e-01   -1.9858509e-01
  1   2.00000000e-01    2.2256209e-01
  2   1.47153371e-01    1.0484383e-01
  3   1.30860462e-01    2.5432133e-02
  4   1.27356948e-01    5.1762930e-03
  5   1.26661980e-01    1.0157670e-03
  6   1.26526298e-01    1.9789700e-04
  7   1.26499890e-01    3.8502000e-05
  8   1.26494753e-01    7.4890000e-06
  9   1.26493754e-01    1.4570000e-06

The approximate solution is r1 =  0.12649375
with f(r1) =   0.00000146
```

$$r1 := 0.1264937538$$

Notice that starting with p_2, each approximation is between the most recent giving a positive value and the most recent giving a negative value. Let's bracket the second root between .2 and .4.

```
> r2:=falseposition(f,.2,.4,.000001,100,r2);
  i   p                 f(p)
  -   -                 ----
  0   2.00000000e-01    2.2256209e-01
  1   4.00000000e-01   -6.6029073e-01
  2   2.50418843e-01    1.7525886e-01
  3   2.81793908e-01    8.0152839e-02
  4   2.94589689e-01    2.8197202e-02
  5   2.98906796e-01    9.0150990e-03
  6   3.00268453e-01    2.7943100e-03
  7   3.00688732e-01    8.5780200e-04
  8   3.00817583e-01    2.6254900e-04
  9   3.00857005e-01    8.0285000e-05
 10   3.00869058e-01    2.4544000e-05
 11   3.00872743e-01    7.5030000e-06
 12   3.00873869e-01    2.2950000e-06
 13   3.00874214e-01    7.0100000e-07

The approximate solution is r2 =  0.30087421
with f(r2) =   0.00000070
```

$$r2 := 0.3008742137$$

Finally, let's bracket the third root between 4 and 5.

```
> r3:=falseposition(f,4,5,.000001,100,r3);
  i   p                 f(p)
  -   -                 ----
  0   4.00000000e+00   -4.8613706e+01
  1   5.00000000e+00    1.8609438e+01
  2   4.72316918e+00   -6.9805422e+00
  3   4.79868425e+00   -5.7869447e-01
  4   4.80475573e+00   -4.5511605e-02
  5   4.80523206e+00   -3.5640830e-03
  6   4.80526935e+00   -2.7924200e-04
  7   4.80527228e+00   -2.2064000e-05
  8   4.80527251e+00   -1.7260000e-06

The approximate solution is r3 =  4.80527251
with f(r3) =  -0.00000173
```

$r3 := 4.805272507$

NumericalAnalysis

There is a procedure called FalsePosition in the **NumericalAnalysis** subpackage of the **Student** package. We use it to find the third root above.

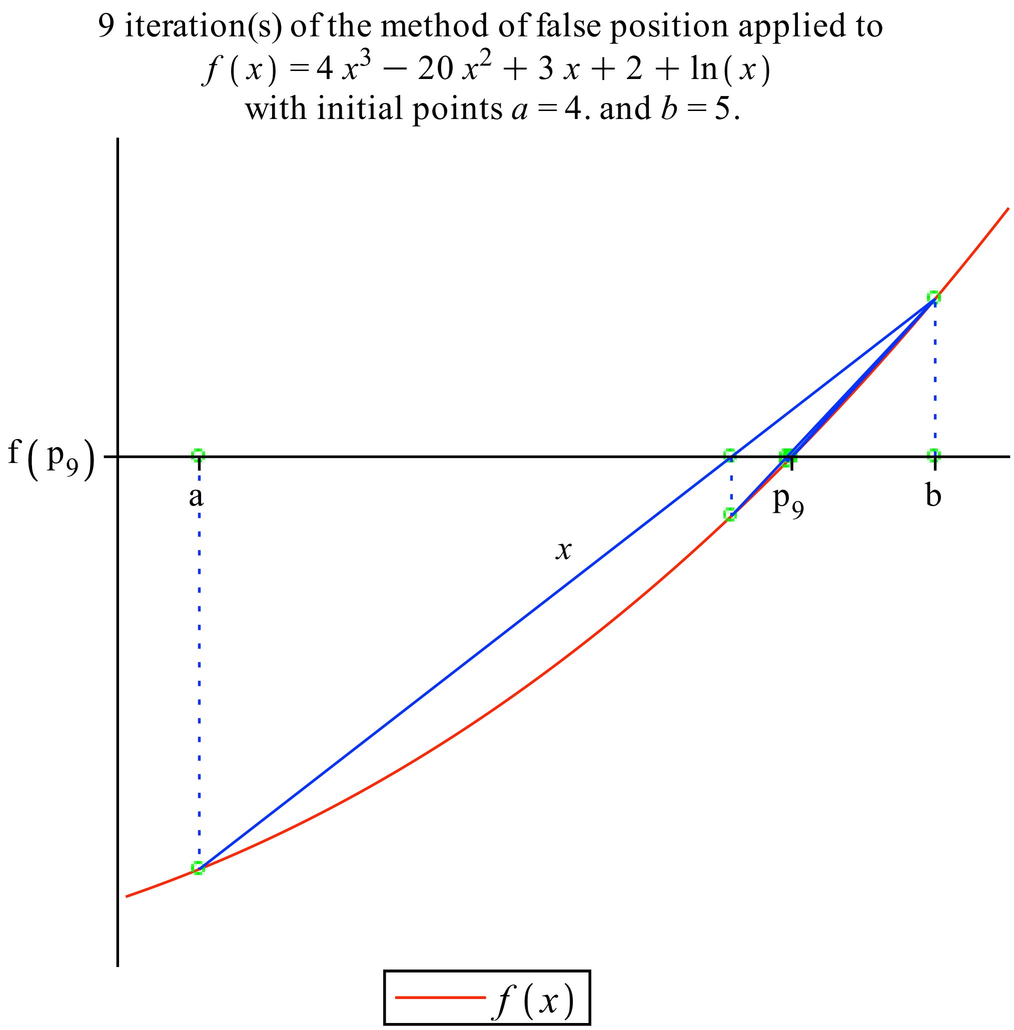
```
> with(Student[NumericalAnalysis]);
```

[*AbsoluteError*, *AdamsBashforth*, *AdamsBashforthMoulton*, *AdamsMoulton*, *AdaptiveQuadrature*, *AddPoint*, *ApproximateExactUpperBound*, *ApproximateValue*, *BackSubstitution*, *BasisFunctions*, *Bisection*, *CubicSpline*, *DataPoints*, *Distance*, *DividedDifferenceTable*, *Draw*, *Euler*, *EulerTutor*, *ExactValue*, *FalsePosition*, *FixedPointIteration*, *ForwardSubstitution*, *Function*, *InitialValueProblem*, *InitialValueProblemTutor*, *Interpolant*, *InterpolantRemainderTerm*, *IsConvergent*, *IsMatrixShape*, *IterativeApproximate*, *IterativeFormula*, *IterativeFormulaTutor*, *LeadingPrincipalSubmatrix*, *LinearSolve*, *LinearSystem*, *MatrixConvergence*, *MatrixDecomposition*, *MatrixDecompositionTutor*, *ModifiedNewton*, *NevilleTable*, *Newton*, *NumberOfSignificantDigits*, *PolynomialInterpolation*, *Quadrature*, *RateOfConvergence*, *RelativeError*, *RemainderTerm*, *Roots*, *RungeKutta*, *Secant*, *SpectralRadius*, *Steffensen*, *Taylor*, *TaylorPolynomial*, *UpperBoundOfRemainderTerm*, *VectorLimit*]

```
> FalsePosition(f,x=[4,5],stoppingcriterion=absolute,tolerance=10^
  (-6),output=information,maxiterations=100);
```

n	a_n	b_n	p_n	$f(p_n)$	*absolute error*
1	4.	5.	4.723169181	-6.980542249	0.276830819
2	4.723169181	5.	4.798684254	-0.578694473	0.201315746
3	4.798684254	5.	4.804755731	-0.045511605	0.195244269
4	4.804755731	5.	4.805232059	-0.003564083	0.194767941
5	4.805232059	5.	4.805269354	-0.000279242	0.194730646
6	4.805269354	5.	4.805272276	-0.000022064	0.194727724
7	4.805272276	5.	4.805272507	-0.000001726	0.194727493
8	4.805272507	5.	4.805272525	$1.38\ 10^{-7}$	0.194727475
9	4.805272507	4.805272525	4.805272524	$-7.2\ 10^{-8}$	$1.\ 10^{-9}$

```
> FalsePosition(f,x=[4,5],stoppingcriterion=absolute,tolerance=10^
  (-6),output=plot,maxiterations=100);
```


9 iteration(s) of the method of false position applied to
f(x) = 4 x^3 − 20 x^2 + 3 x + 2 + ln(x)
with initial points a = 4. and b = 5.
f(p9)
a
p9
b
x
f(x)