01.04 rootfinding: newtons method

aka newton-raphson. newtons is a variant of FPI.

to find root of $f(x) = 0$, start with guess x_0. draw tangent line at $f(x_0)$. ie, where $f'(x_0)$ intersects x-axis is the next iteration x_1. ie,

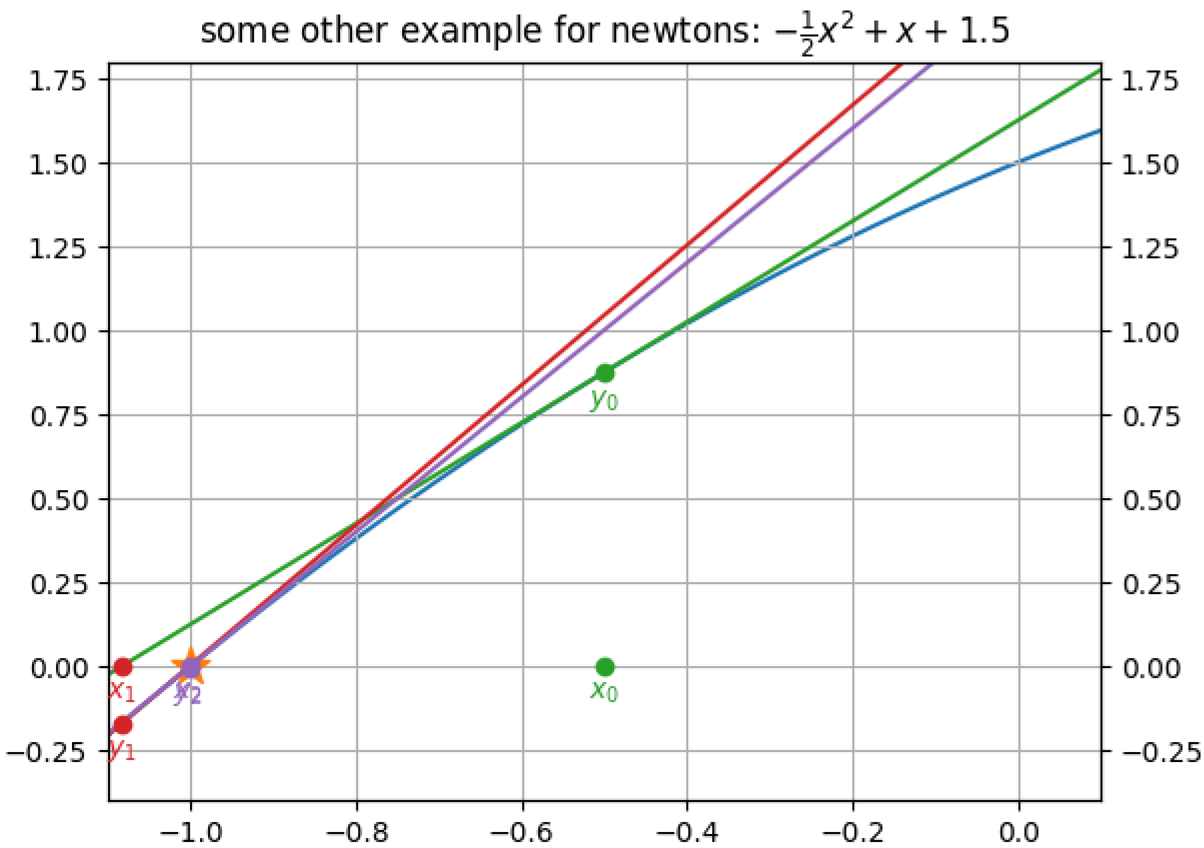
$$tan\theta = \frac{f(x_0)}{x_0 - x_1} = f'(x_0) \Rightarrow x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

code, visual: newtons

given $f(x) = -\frac{1}{2}x^2 + x + 1.5$.

```
# requires prior execution of fpi_expanded()
# repurposed bisection animation for newtons method

if __name__ == "__main__":

```

```
ani
```



Once Loop Reflect

algorithm

newtons method

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}, \quad i = 0, 1, 2, 3, \ldots$$

$$= g(x_i), \quad g(x) = x - \frac{f(x)}{f'(x)}.$$

```
function g(x)
  return x - f(x)/df(x)
```

```
function fpi(g,x0,epsilon,imax)
```

example 11

solve $f(x) = x^3 + x - 1$ using newtons.

$$f'(x) = 3x^2 + 1$$

$$\Rightarrow x_{i+1} = x_i - \frac{x_i^3 + x_i - 1}{3x_i^2 + 1}$$

$$= \frac{2x_i^3 + 1}{3x_i^2 + 1}, \quad \text{which looks strangely familiar...}$$

$$\Downarrow \quad x_0 = -0.7$$

$$x_1 = \frac{2(-0.7)^3+1}{3(-0.7)^2+1} \approx 0.1271$$
$$x_2 = \frac{2x_1^3+1}{3x_i^2+1} \approx 0.9577.$$

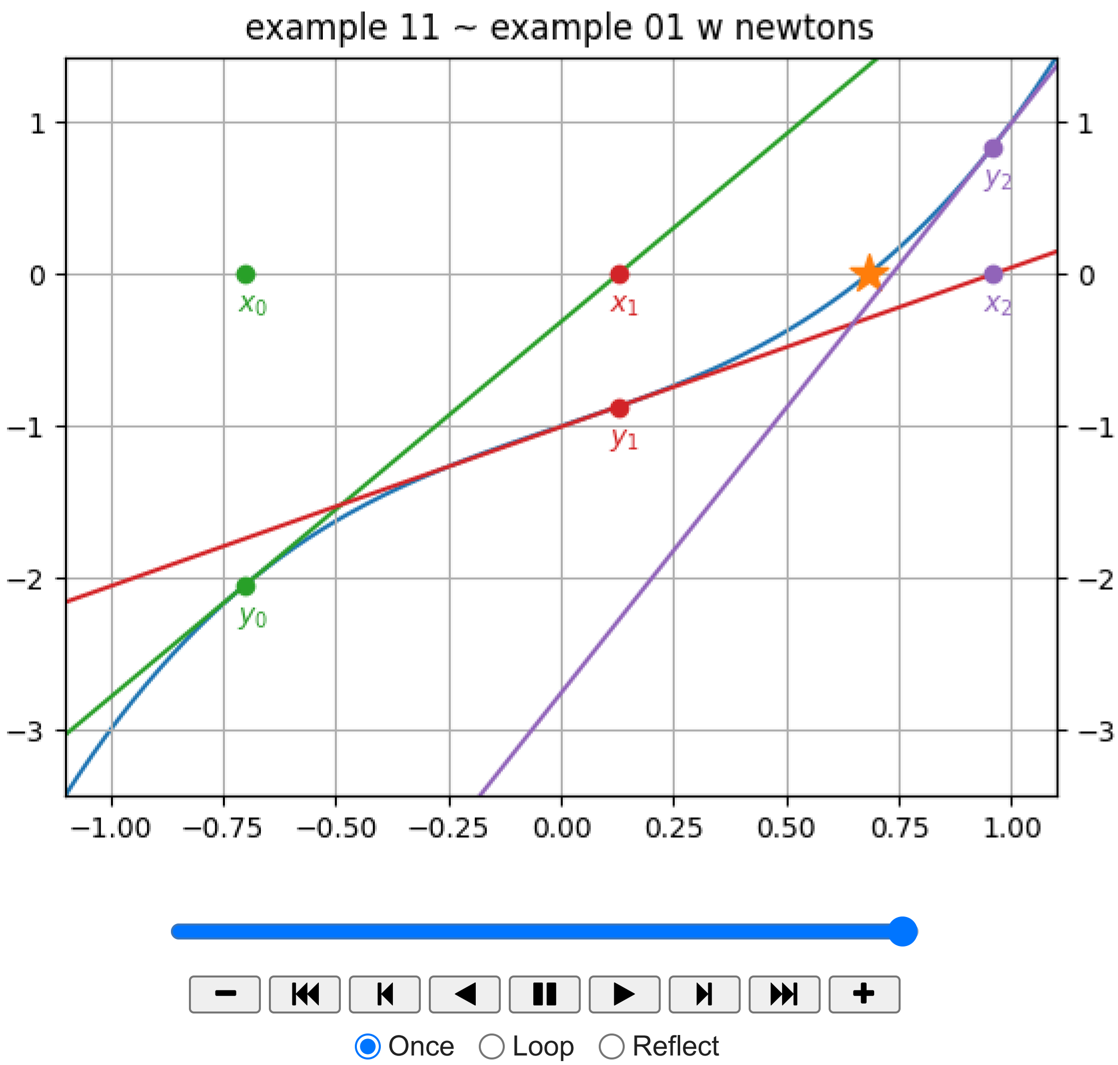
code, example 11

```
# example 11 revisits example 01 # repurposed previous code

if __name__ == "__main__":

```

```
example 11 ~ example 01 w newtons. x0 = -0.7.

|   i |        x[i] |       e[i] |   "/e[i-1]^2 |
|-----|-------------|------------|--------------|
| 000 | -0.70000000 | 1.38232780 |              |
| 001 |  0.12712551 | 0.55520230 |   0.29055555 |
| 002 |  0.95767812 | 0.27535032 |   0.89327066 |
```

```
| 003 |   0.73482779 | 0.05249999 |    0.69244945 |
| 004 |   0.68459177 | 0.00226397 |    0.82139415 |
| 005 |   0.68233217 | 0.00000437 |    0.85266556 |
| 006 |   0.68232780 | 0.00000000 |    0.85407850 |
```

```
ani
```



code, example 11, briefly

```
if __name__ == "__main__":

  f = lambda x : pow(x,3) + x - 1
  df = lambda x : 3*pow(x,2) + 1

  g = lambda x : x - f(x)/df(x) # newtons method

```

```
ws = fpi_expanded(g,x=-0.7,tol=1e-4,worksheet=Tru
iterations,root = ws[len(ws)-1]
print(f"root {root} at {iterations} iterations.")
```

```
root 0.6823321742044841 at 5 iterations.
```

1 quadratic convergence

definition 10

let e_i denote error after step i of iterative method. iteration is **quadratically convergent** if

$$M = \lim_{i \to \infty} \frac{e_{i+1}}{e_i^2} < \infty.$$

theorem 11

let f be twice continuously differentiable and $f(r) = 0$. if $f'(r) \neq 0$, then newtons is locally and quadratically convergent to r. error e_i at step i satisfies

$$\lim_{i \to \infty} \frac{e_{i+1}}{e_i^2} = M, \quad \text{where } M = \frac{f''(r)}{2f'(r)}.$$

proof

1. local convergence

note that newtons method is a particular form of FPI where

$$g(x) = x - \frac{f(x)}{f'(x)},$$

$$g'(x) = 1 - \frac{f'(x)^2 - f(x)f''(x)}{f'(x)^2} = \frac{f(x)f''(x)}{f'(x)^2}.$$

$g'(r) = 0$ so locally convergent by theorem 06. ✓

2. quadratic convergence

derive newtons method with taylors formula. at i steps,

$$f(r) = f(x_i) + (r - x_i)f'(x_i) + \frac{(r - x_i)^2}{2} f''(c_i)$$

$$\Downarrow \quad c_i \text{ between } x_i, r$$

$$0 = f(x_i) + (r - x_i)f'(x_i) + \frac{(r - x_i)^2}{2} f''(c_i)$$

$$\Downarrow$$

$$-\frac{f(x_i)}{f'(x_i)} = r - x_i + \frac{(r - x_i)^2}{2} \frac{f''(c_i)}{f'(x_i)}, \qquad f'(x_i) \neq 0$$

$$\Downarrow$$

$$x_i - \frac{f(x_i)}{f'(x_i)} - r = \frac{(r - x_i)^2}{2} \frac{f''(c_i)}{f'(x_i)}$$

$$\Downarrow \quad e_i = |x_i - r|$$

$$x_{i+1} - r = e_i^2 \frac{f''(c_i)}{f'(x_i)}$$

$$\Downarrow$$

$$e_{i+1} = e_i^2 \frac{f''(c_i)}{f'(x_i)}$$

bc c_i is between x_i, r, it converges to r as x_i converges to r and

$$\lim_{i \to \infty} \frac{e_{i+1}}{e_i^2} = \left| \frac{f''(r)}{2f'(r)} \right|,$$

which is the definition of quadratic convergence. ✓ ■

ie,

$$e_{i+1} \approx M e_i^2, \quad M = \left| \frac{f''(r)}{2f'(r)} \right|, \quad f'(r) \neq 0.$$

compare with linearly convergent methods with $e_{i+1} \approx S e_i$, where $S = |g'(r)|$ for FPI and $S = \frac{1}{2}$ for bisection. while S is critical for linearly convergent methods, M is less critical bc of the division by the square of the previous error.

- example 11, revisited

$f'''(x) = 6x$ and at $x_c \approx 0.6823$, $M \approx 0.85$ which agrees with error ratio of displayed with iteration.

- example 6, revisited

let a be positive and consider roots of $f(x) = x^2 - a$ using newtons method.

$$\begin{aligned} x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} &= x_i - \frac{x_i^2 - a}{2x_i} \\ = \frac{x_i^2 + a}{2x_i} &= \frac{x_i + \frac{a}{x_i}}{2}. \quad \checkmark \text{ babylonia!} \end{aligned}$$

wrt convergence,

$$\begin{aligned} &f'(\sqrt{a}) = 2\sqrt{a}, \quad f''(\sqrt{a}) = 2 \\ &\qquad \Downarrow \quad \text{quadratically convergent bc } f'(\sqrt{a}) = 2\sqrt{a} \neq 0 \end{aligned}$$

$$e_{i+} \approx M e_i^2, \quad M = \frac{f''(r)}{2f'(r)} = \frac{2}{2 \cdot 2\sqrt{a}} = \frac{1}{2\sqrt{a}}.$$

2 linear convergence

example 12

find root of $f(x) = x^2$ using newtons method.

obviously $r = 0$, but

$$\begin{aligned} x_{i+1} &= x_i - \frac{f(x_i)}{f'(x_i)} \\ &= x_i - \frac{x_i^2}{2x_i} \\ &= \frac{x_i}{2}. \end{aligned}$$

which gets you linear convergence of $S = \frac{1}{2}$. $\downarrow$ see?

code, example 12

```
# example 12 # mod example 11

if __name__ == "__main__":

```

```
example 12. x0 = 1.

|   i |       x[i] |       e[i] |   "/e[i-1] |
|-----|------------|------------|------------|
| 000 | 1.00000000 | 1.00000000 |            |
| 001 | 0.50000000 | 0.50000000 | 0.50000000 |
| 002 | 0.25000000 | 0.25000000 | 0.50000000 |
| 003 | 0.12500000 | 0.12500000 | 0.50000000 |
| 004 | 0.06250000 | 0.06250000 | 0.50000000 |
| 005 | 0.03125000 | 0.03125000 | 0.50000000 |
|-----|------------|------------|------------|
| 022 | 0.00000024 | 0.00000024 | 0.50000000 |
| 023 | 0.00000012 | 0.00000012 | 0.50000000 |
| 024 | 0.00000006 | 0.00000006 | 0.50000000 |
| 025 | 0.00000003 | 0.00000003 | 0.50000000 |
| 026 | 0.00000001 | 0.00000001 | 0.50000000 |
```

example 13

find root of $f(x) = x^m$ using newtons method.

again, only roor $r = 0$ and

$$x_{i+1} = x_i - \frac{x_i^m}{m\, x^{m-1}} = \tfrac{m-1}{m}\, x_i$$

$$\Downarrow \qquad e_i = |x_i - r| = |x_i - 0| = |x_i|$$

$$e_{i+1} = S\, e_i, \text{ where } S = \tfrac{m-1}{m}.$$

theorem 12

assume $(m+1)$-times continuously differentiable function f on $[a, b]$ has multiplicity m root at r. then newtons is locally convergent to r and error e_i at step i satisfies

$$\lim_{i \to \infty} \tfrac{e_{i+1}}{e_i} = S, \qquad S = \tfrac{m-1}{m}.$$

example 14

find multiplicity of root $r = 0$ of $f(x) = sin\, x + x^2\, cos\, x - x^2 - x$ and estimate iterations required using newtons method with convergence of six decimal places beginning with $x_0 = 1$.

$$
\begin{aligned}
f(x) &= sin\,x + x^2\,cos\,x - x^2 - x \\
f'(x) &= cos\,x + 2x\,cos\,x - x^2\,sin\,x - 2x - 1 \\
f''(x) &= -sin\,x + 2\,cos\,x - 4x\,sin\,x - x^2\,cos\,x - 2 \\
f'''(x) &= -cos\,x - 6\,sin\,x - 6x\,cos\,x + x^2\,sin\,x
\end{aligned}
$$

$$\Downarrow \quad r = 0$$

$$f(r) = f'(r) = f''(r) = 0, \quad f'''(r) = -1 \quad \Rightarrow \quad m = 3 \text{ and}$$

$$\Downarrow \quad \text{by theorem 12, with linear convergence}$$

$$S = \tfrac{m-1}{m} = \tfrac{2}{3} \quad \Rightarrow \quad e_{i+1} \approx \tfrac{2}{3}\,e_i.$$

$$x_0 = 1 \quad \Rightarrow \quad e_0 = 1$$

$$\Downarrow$$

$$\left(\tfrac{2}{3}\right)^n < 0.5 \times 10^{-6}$$

$$\Downarrow$$

$$n > \frac{log_{10}(0.5) - 6}{log_{10}(\frac{2}{3})} \approx 35.78 \quad \rightarrow \quad 36 \text{ iterations.}$$

code, example 14

```
# example 14 # mod example 12

if __name__ == "__main__":

```

```
example 14. x0 = 1.

|   i |       x[i] |       e[i] |   "/e[i-1] |
|-----|------------|------------|------------|
| 000 | 1.00000000 | 1.00000000 |            |
| 001 | 0.72159024 | 0.72159024 | 0.72159024 |
| 002 | 0.52137095 | 0.52137095 | 0.72253049 |
| 003 | 0.37530831 | 0.37530831 | 0.71984890 |
| 004 | 0.26836349 | 0.26836349 | 0.71504809 |
| 005 | 0.19026161 | 0.19026161 | 0.70896981 |
|-----|------------|------------|------------|
```

```
| 031 | 0.00000622 | 0.00000622 | 0.66667262 |
| 032 | 0.00000414 | 0.00000414 | 0.66666644 |
| 033 | 0.00000276 | 0.00000276 | 0.66665531 |
| 034 | 0.00000184 | 0.00000184 | 0.66668266 |
| 035 | 0.00000123 | 0.00000123 | 0.66667984 |
```

theorem 13

if $(m+1)$-times continuously differentiable function f on $[a,b]$ has multiplicity $m>1$ root at r, then **modified newtons method**

$$x_{i+1} = x_i - \frac{m\,f(x_i)}{f'(x_i)}$$

converges locally and quadratically to r.

ie, if multiplicity known, newtons can be improved.

example 14, revisited

code, example 14, revisited

```
# example 14 with modified newtons

if __name__ == "__main__":

```

```
example 14. x0 = 1.

|   i |       x[i] |       e[i] |   "/e[i-1] |
|-----|------------|------------|------------|
| 000 | 1.00000000 | 1.00000000 |            |
| 001 | 0.16477072 | 0.16477072 | 0.16477072 |
| 002 | 0.01620734 | 0.01620734 | 0.09836297 |
| 003 | 0.00024654 | 0.00024654 | 0.01521172 |
| 004 | 0.00000006 | 0.00000006 | 0.00024629 |
```

and then the conflict with machine precision wins bc its the machine. another reminder to mind the machine.

ie, backwards error is driven near ϵ_{mach} but forward error x_i is several orders of magnitude larger.

3 more fail

example 15

apply newtons method to $f(x) = 4x^4 - 6x^2 - \frac{11}{4}$ with $x_0 = \frac{1}{2}$.

the function has roots bc it is continuous and negative at $x = 0$ and goes to $+\infty$ for large positive and negative x.

$$x_{i+1} = x_i - \frac{4x_i^4 - 6x_i^2 - \frac{11}{4}}{16x_i^3 - 12x_i}.$$

however, $x_1 = -\frac{1}{2} \quad \mapsto \quad x_2 = -\frac{1}{2} \quad \mapsto$ lol.

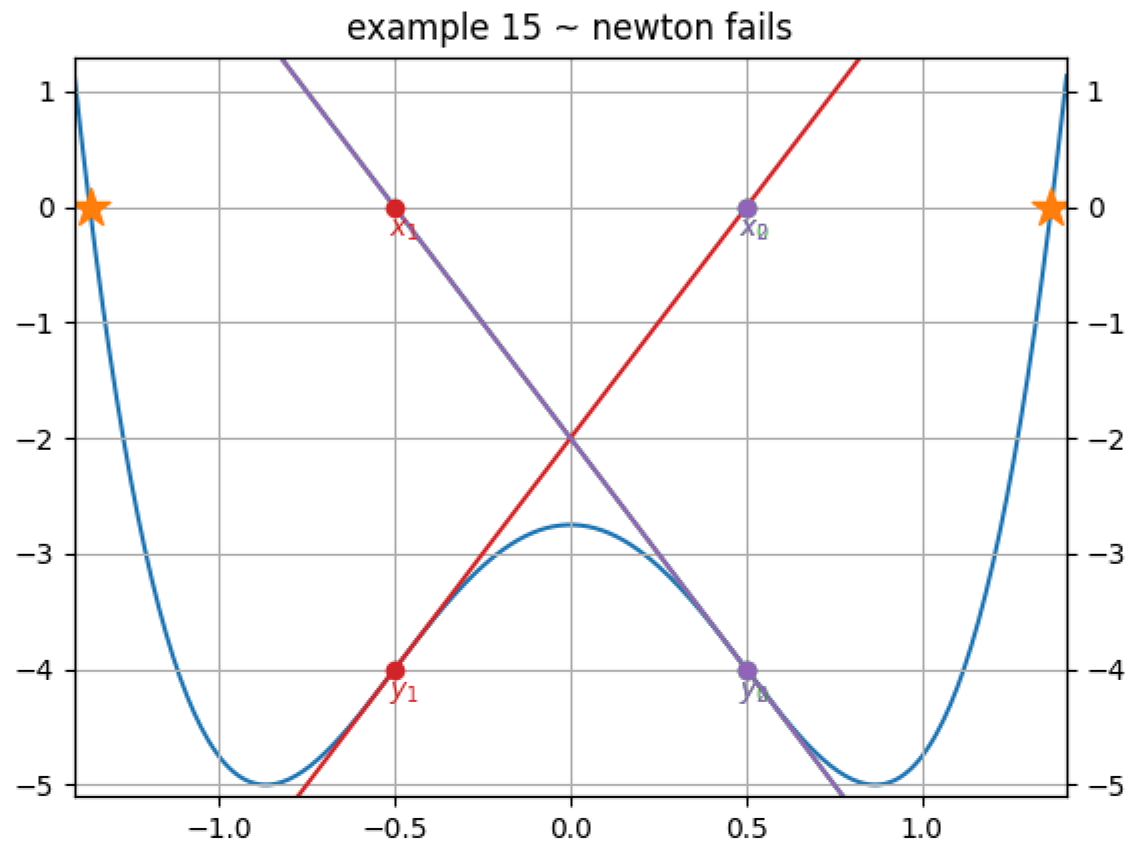
code, example 15

thats worth some code.

```
# example 15 # repurposed example 11

if __name__ == "__main__":

```

```
example 15 ~ newton fails. x0 = 0.5.

|   i |        x[i] |       e[i] |   "/e[i-1]^2 |
|-----|-------------|------------|--------------|
| 000 |  0.50000000 | 0.86676040 |              |
| 001 | -0.50000000 | 1.86676040 |   2.48479439 |
| 002 |  0.50000000 | 0.86676040 |   0.24872641 |
| 003 | -0.50000000 | 1.86676040 |   2.48479439 |
| 004 |  0.50000000 | 0.86676040 |   0.24872641 |
| 005 | -0.50000000 | 1.86676040 |   2.48479439 |
```

```
ani
```



in addition to examples 14 and 15, if $f'(x_i) = 0$. also, iterations unto infinity or mimicry of an rng. however, theorems 11 and 12 guarantee a neighborhood of initial guesses surrounding each root for which convergence to that root is assured.