Golden Search Pseudocode:

function $N = \text{Golden\_Search}(tol)$

↗ output # of iterations

↑ input error tolerance

- set tau, search step
- establish a, b (initial endpts of our interval)
- compute initial error, $err = |b-a|$
- initialize # of iterations, N, at 0

while $\underbrace{err > tol}_{\text{run until error is less than tolerance}}$

- choose 2 new x values in our interval x_1 & x_2, that are tau distance from a and b.
- make sure $x_1 < x_2$, if not, switch them! (use if statement)
- calculate corresponding y values using function $f(x) = 0.5 - xe^{-x^2}$
- pick x w/ larger y value to be new endpoint that replaces a (if x_1) or b (if x_2) → use if/else statement
- compute new error, $err = |b-a|$
- compute approx min $= \frac{a+b}{2}$
- add to # of iterations, $N = N+1$

end while loop

end function

Successive Parabolic Interpolation PseudoCode:

function N = successive_Parabolic_Interpolation(tol)

output # of iterations

input error tolerance

- pick 3 initial x values in the interval [0,2], x_1, x_2, x_3
- est. initial error, $err = |x_3 - x_1|$
- initialize # of iterations, $N = 0$

while $\underbrace{err > tol}_{\text{run until error < tolerance}}$

- find the corresponding y values to each x-value using function $f(x) = 0.5 - xe^{-x^2}$
- use formula $A = inv\left(\begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \end{bmatrix}\right) * \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$

to find the a,b,c vals such that:

$$\left.\begin{aligned} ax_1^2 + bx_1 + c &= y_1 \\ ax_2^2 + bx_2 + c &= y_2 \\ ax_3^2 + bx_3 + c &= y_3 \end{aligned}\right\}$$

creates a parabola that goes through all 3 pts: $(x_1,y_1), (x_2,y_2), (x_3,y_3)$

- find the minimum of the parabola:

$$x_p = \frac{-b}{2a}$$

- redefine pts, & replace x_1 w/ x_p:

$x_3 = x_2$
$x_2 = x_1$
$x_1 = x_p$

- recalculate error, $err = |x_3 - x_1|$
- add to the # of iterations, $N = N+1$

end while loop

- find the approx minimum

$$min = \frac{x_1 + x_2 + x_3}{3}$$

end function