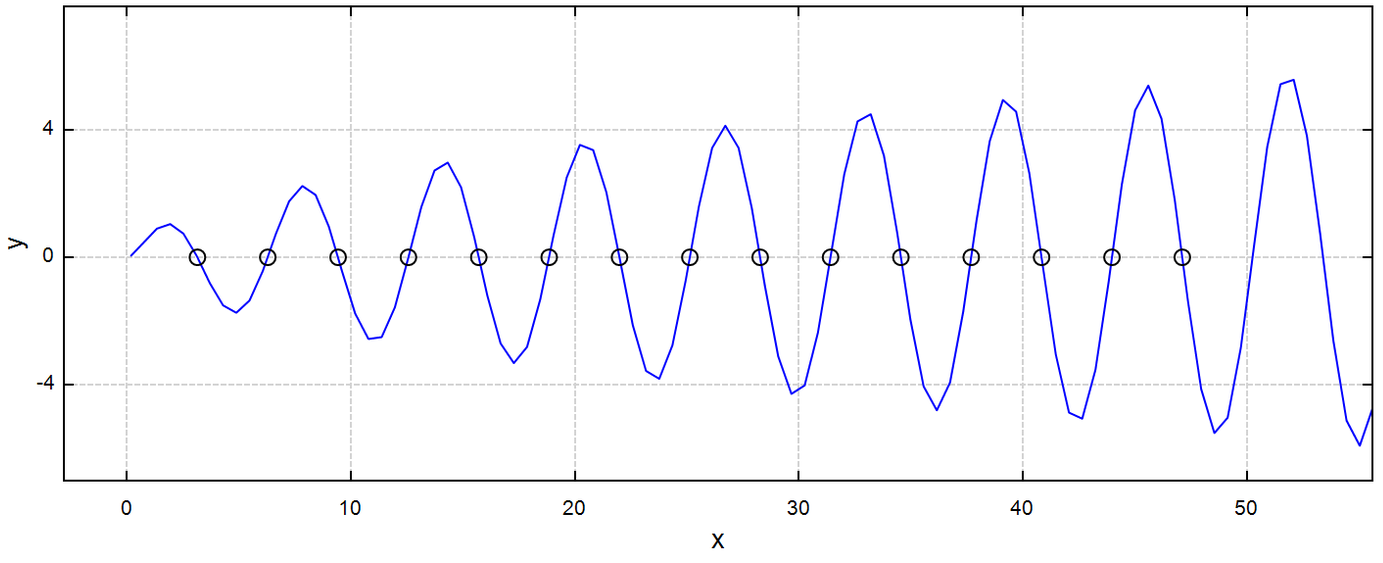
$$F(\mu) := \mu \cdot \sqrt{\frac{2}{\pi \cdot \mu}} \cdot \sin(\mu) - 0.008 \cdot \sqrt{\frac{2}{\pi \cdot \mu}} \cdot \cos(\mu)$$

$$check(\mu) := \overrightarrow{-\operatorname{trunc}\left(\log 10\left(\left|F(\mu)\right|\right)\right)}$$ (Bigger is better)

$$\mu n := \operatorname{solve}(F(\mu) = 0,\ \mu,\ 0,\ 50) = \begin{bmatrix} 3.14413652563747 \\ 6.28445825887869 \\ 9.42562670216452 \\ 12.5670072011288 \\ 15.7084725466544 \\ 18.849980325127 \\ 21.9915123518179 \\ 25.1330595345618 \\ 28.2746164218472 \\ 31.4161811817262 \\ 34.5577503834108 \\ 37.6993240484474 \\ 40.8409004068643 \\ 43.9824790408622 \\ 47.1240595682305 \end{bmatrix} \qquad check(\mu n) = \begin{bmatrix} 6 \\ 7 \\ 7 \\ 8 \\ 8 \\ 10 \\ 10 \\ 11 \\ 5 \\ 10 \\ 5 \\ 9 \\ 6 \\ 10 \\ 8 \end{bmatrix}$$



$$\begin{cases} F(\mu) \\ \operatorname{augment}(\mu n, \overrightarrow{F(\mu n)}, \text{"}\circ\text{"}) \end{cases}$$

First you can check the solvers in the NonLinearSolvers plugin. You can read about them in the Martin Kraska handbook in the Plugin / Interactive books plugin or here https://en.smath.com/forum/yaf_poststt1508p4_NonlinearSolvers-plugin.aspx

Second, you can try al_nleqsolve, which is the most poerfull pure numerical solver avaible for now in SMath.

Third, you can write your own routines

$$feval(\#fs\#, \#x\#) := \left| \operatorname{str2num}(\operatorname{concat}(\operatorname{num2str}(\#fs\#), \text{"("}, \operatorname{num2str}(\#x\#), \text{")"})) \right.$$

$$Bis(f\#, ao\#, bo\#, \varepsilon y\#, \varepsilon x\#) := \left| \begin{matrix} [a\#\ b\#] := [ao\#\ bo\#] \\ [ya\#\ yb\#] := [feval(f\#, a\#)\ feval(f\#, b\#)] \end{matrix} \right.$$

```
[ya# yb#]:=[feval(f#, a#) feval(f#, b#)]
if sign(ya#) = sign(yb#)
  error("Bad interval")
else if |ya#| ≤ εy# · UnitsOf(ya#)
  c# := a#
else if |yb#| ≤ εy# · UnitsOf(yb#)
  c# := b#
else
  N# := 1 + round( ln( | (b# − a#) / UnitsOf(b# − a#) | ) − ln(εx#) ) / ln(2) , 0 )
  for iter# ∈ [1..N#]
    [c# := eval(0.5·(a# + b#)) yc# := feval(f#, c#)]
    if |yc#| ≤ εy# · UnitsOf(yc#)
      break
    else
      if sign(yb#)·sign(yc#) > 0
        [b# yb#]:=[c# yc#]
      else
        [a# ya#]:=[c# yc#]
c#
```

$NR_1(f\#, a\#, \varepsilon y\#, \varepsilon x\#, \delta x\#, MI\#) :=$

```
t# := a#
for iter# ∈ [1..MI#]
  [x# := t# y# := feval(f#, x#)]
  if |y#| ≤ εy# · UnitsOf(y#)
    break
  else
    t# := eval( x# − δx# · UnitsOf(x#) · y# / ( feval(f#, x# + δx# · UnitsOf(x#)) − y# ) )
    if |x# − t#| < εx# · UnitsOf(x# − t#)
      break
if iter# = MI#
  error("MaxIters rached.")
else
  t#
```

Usual values $\varepsilon y := 0$ $\varepsilon x := 10^{-15}$ $\delta x := 10^{-12}$

$k := [1..\text{length}(\mu n)]$ use the results of solve as interval for Bis or guess values for NR

$$\mu b_k := Bis(F, \mu n_k - 0.5, \mu n_k + 0.5, \varepsilon y, \varepsilon x) = \begin{bmatrix} 3.14413706643403 \\ 6.28445828812908 \\ 9.42562671049443 \\ 12.5670072017972 \\ 15.7084725472111 \\ 18.8499803251392 \\ 21.991512351822 \\ 25.1330595345624 \\ 28.2746168215903 \\ 31.4161811817374 \\ 34.5577506860319 \\ 37.6993240484707 \\ 40.8409003787322 \\ 43.9824790408665 \\ 47.1240595685063 \end{bmatrix} \quad check(\mu b) = \begin{bmatrix} 14 \\ 13 \\ 14 \\ 12 \\ 13 \\ 12 \\ 13 \\ 12 \\ 13 \\ 12 \\ 12 \\ 12 \\ 13 \\ 12 \\ 12 \end{bmatrix}$$

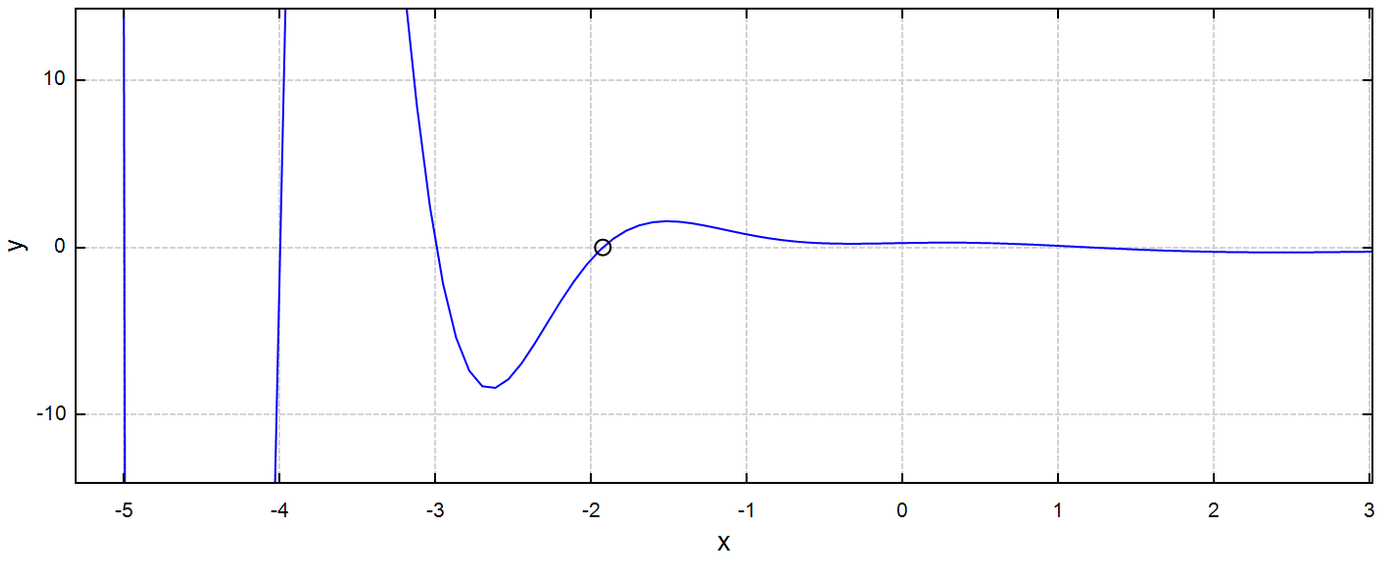
$$\mu r_k := NR_1\left(F, \mu n_k, \varepsilon y, \varepsilon x, \delta x, 10\right) = \begin{bmatrix} 3.14413706643404 \\ 6.28445828812909 \\ 9.42562671049443 \\ 12.5670072017973 \\ 15.7084725472111 \\ 18.8499803251393 \\ 21.991512351822 \\ 25.1330595345624 \\ 28.2746168215903 \\ 31.4161811817373 \\ 34.5577506860319 \\ 37.6993240484706 \\ 40.8409003787322 \\ 43.9824790408664 \\ 47.1240595685063 \end{bmatrix} \qquad check(\mu r) = \begin{bmatrix} 14 \\ 14 \\ 14 \\ 13 \\ 13 \\ 13 \\ 13 \\ 12 \\ 13 \\ 13 \\ 12 \\ 12 \\ 13 \\ 13 \\ 12 \end{bmatrix}$$

solve, and the solvers in NonLinearSolvers can't work with functions defined in plugins, like Bessel ones.

$$Y(v) := v \cdot \text{BesselJ}(v, 1) - 1.01 \cdot \text{BesselJ}(v, 3)$$

$$vo := Bis\left(Y, -2.5, -1, 0, \varepsilon x\right) = -1.92464715696971$$

$$vo := NR_1\left(Y, -2, \varepsilon y, \varepsilon x, \delta x, 10\right) = -1.9246471569697$$



$$\begin{cases} Y(v) \\ \text{augment}\left(vo, Y(vo), \text{"o"}\right) \end{cases}$$

Alvaro