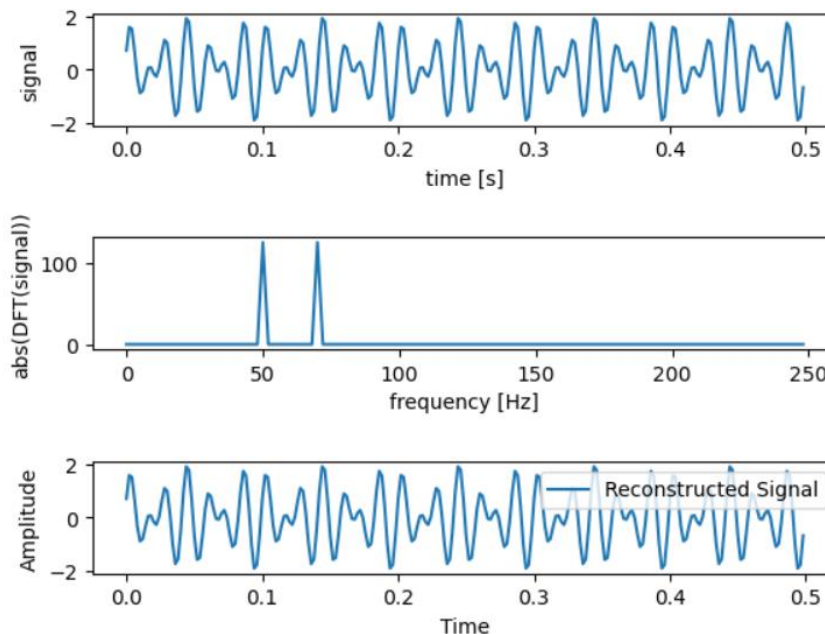


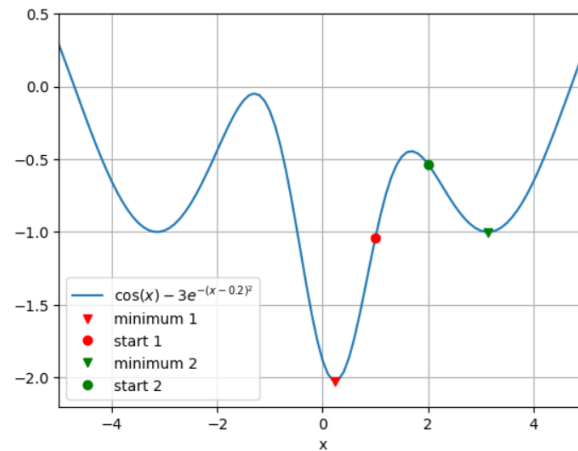
## LL7. Utilizarea librărilor NumPy, SciPy și SymPy pentru efectuarea diferitor calcule în limbajul Python

1. Calculați  $\sqrt{2}$  cu 100 de zecimale.
2. Calculați  $\frac{1}{2} + \frac{1}{3}$  în aritmetica rațională.
3. Calculați forma extinsă a expresiei  $(x + y)^6$ .
4. Simplificați expresia trigonometrică  $\sin(x)/\cos(x)$ .
5. Calculați  $\lim_{x \rightarrow 0} \sin(x)/x$ .
6. Calculați derivata pentru  $\log(x)$  pentru  $x$ .
7. Rezolvați sistemul de ecuații  $2x + 3y = 5$ ,  $4x - 3y = -4$ .
8. Există valori booleene  $x, y$  care fac expresia  $(x \vee \neg y) \wedge (y \vee \neg x)$  adevărată? Argumentați răspunsul, folosiți `sym.satisfiable`.
9. Rezolvați ecuația diferențială a lui Bernoulli  $x \frac{df(x)}{dx} + f(x) - f(x)^2 = 0$ . Rezolvați aceeași ecuație folosind `hint='Bernoulli'`. Ce observați?
10. Folosind funcția `quad()` din librăria `scipy`, scrieți un program care rezolvă următoarea integrală numerică:  $I = \int_0^1 \cos(2\pi x) dx$ . De ce este important să avem o estimare a preciziei (sau a erorii) integralei numerice?
11. Creați un semnal ca o suprapunere a unei unde sinusoidale de 50 Hz și 70 Hz (cu o ușoară schimbare de fază între ele). Apoi transformați semnalul Fourier și trasați valoarea absolută a coeficienților (complexi) discreți de transformare Fourier în funcție de frecvență (observați vârfuri la 50Hz și 70Hz).



12. Găsiți valoarea minimă a lui  $x$  pentru optimizarea expresiei  $\cos(x) - 3e^{-(x-0.2)^2}$ . Apelați funcția `scipy.optimize.fmin` care ia ca argument o funcție  $f$  pentru a minimiza și o valoare inițială  $x0$  de la care să pornească căutarea pentru minim și care returnează valoarea lui  $x$  pentru care  $f(x)$  este (local) minimizat. Repetați căutarea minimului pentru două valori

( $x_0 = 1.0$  și, respectiv,  $x_0 = 2.0$ ) pentru a demonstra că în funcție de valoarea de pornire putem găsi minime diferite ale funcției  $f$ .



13. Efectuați diverse manipulări asupra unei imagini: schimbați orientarea, rezoluția (`scipy.ndimage` oferă manipularea tablourilor n-dimensionale ca imagini). Generați zgomot asupra imaginii, apoi folosiți pe rând filtrele Gaussian, median, Wiener. Observați eficacitatea fiecărui filtru.