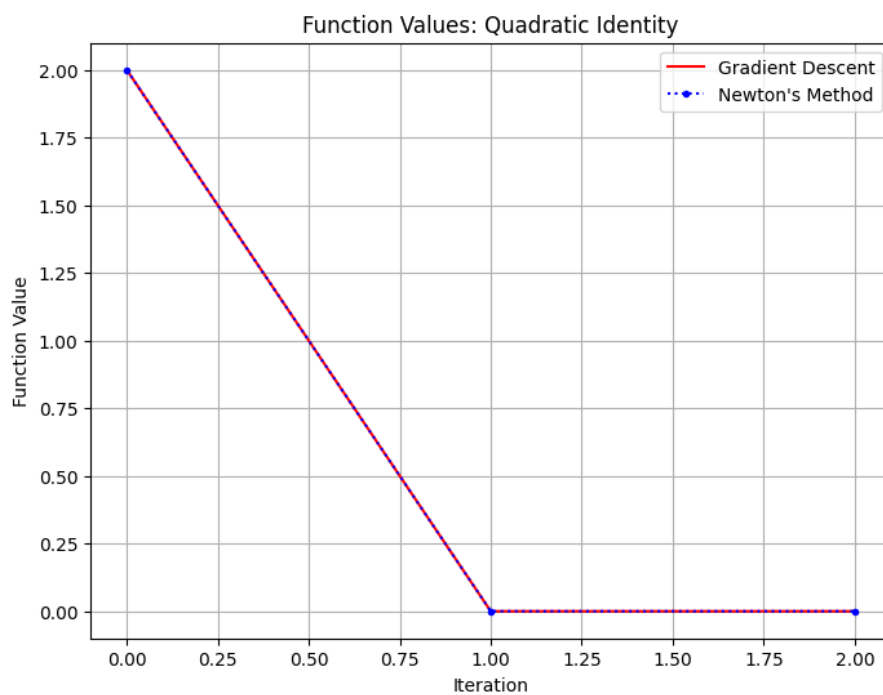
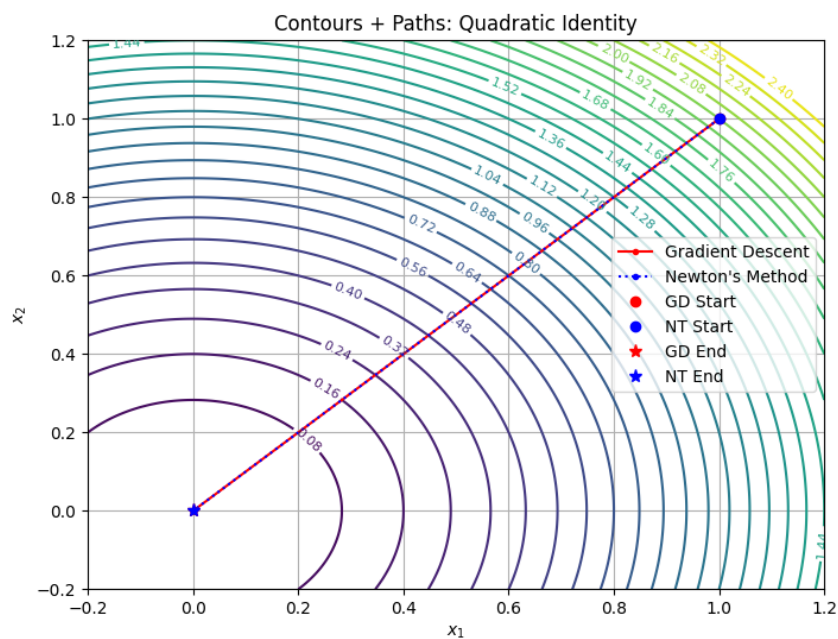


Eral Edree – Numerical Optimization in Python – python ex1.

Link to github repo: [https://github.com/EralEdree/numerical\\_optimization\\_ex\\_1](https://github.com/EralEdree/numerical_optimization_ex_1)

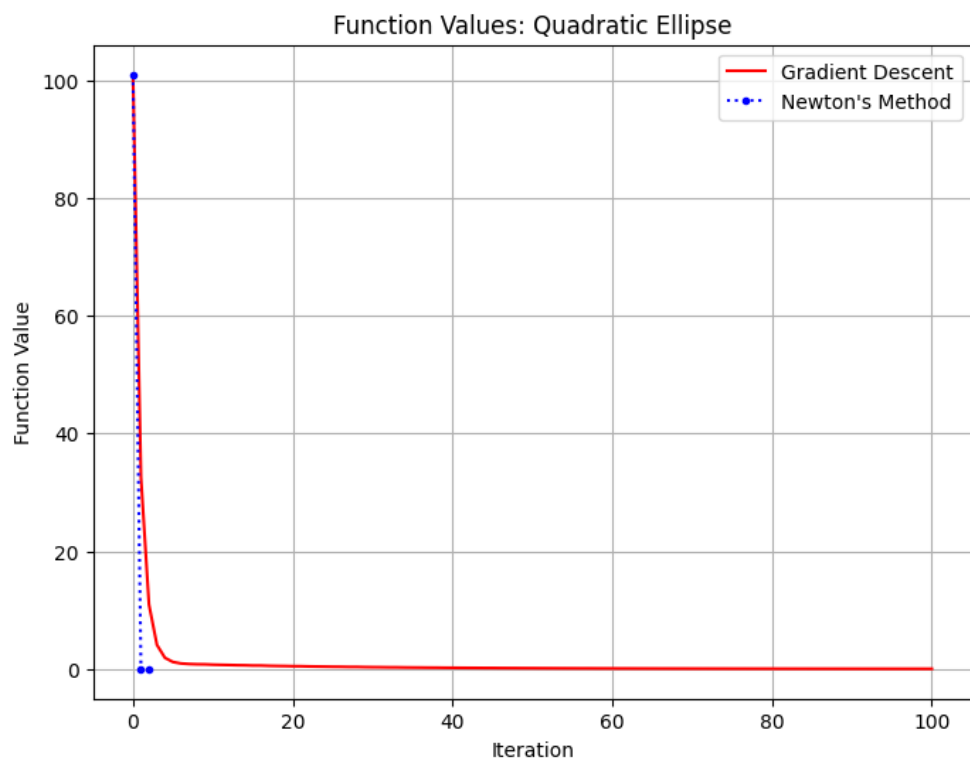
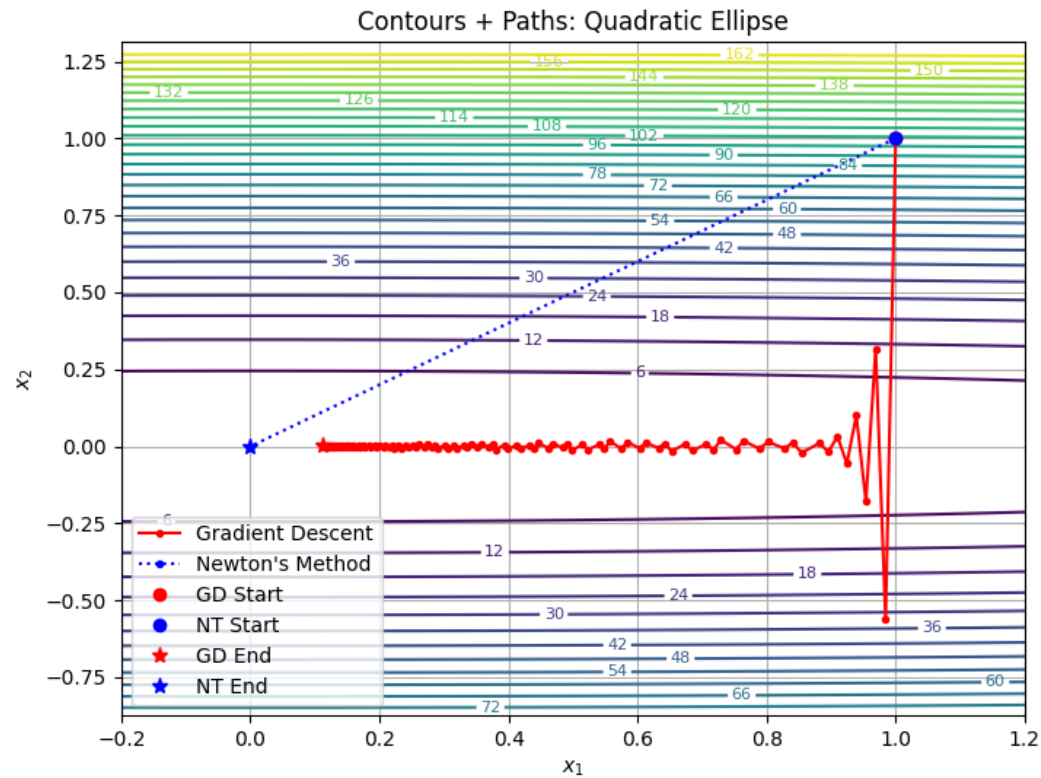
quadratic\_identity, method: **gradient\_descent**, Iter 1:  $x = [0.0.]$ ,  $f(x) = 0.0$ , success = True

function: quadratic\_identity, method: **newton\_method**, Iter 1:  $x = [0.0.]$ ,  $f(x) = 0.0$ , success = True



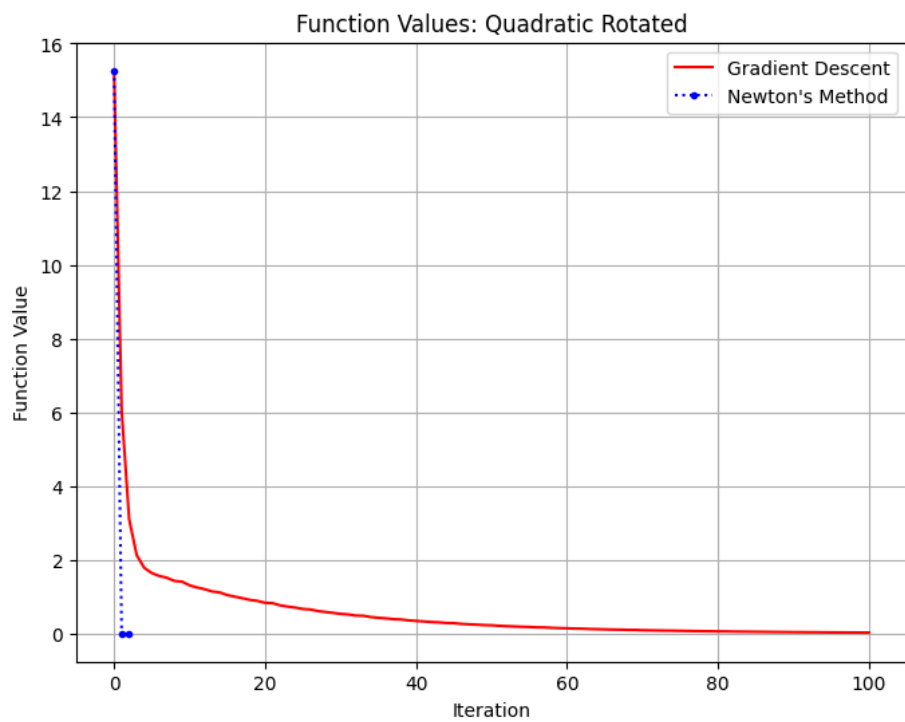
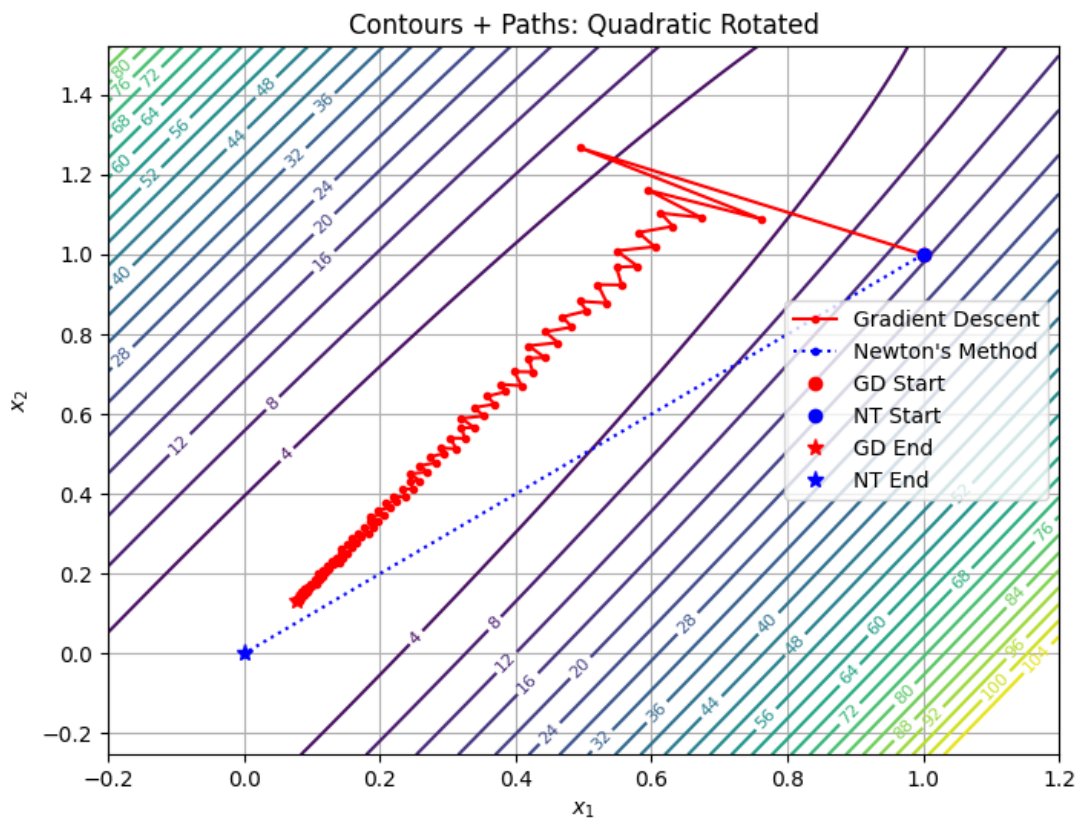
function: quadratic\_ellipse, method: **gradient\_descent**, lter 100:  $x = [0.11271997 \ 0.0008856]$ ,  $f(x) = 0.012784220095399295$ , success = False

function: quadratic\_ellipse, method: **newton\_method**, lter 1:  $x = [0 \ 0.]$ ,  $f(x) = 0.0$ , success = True:



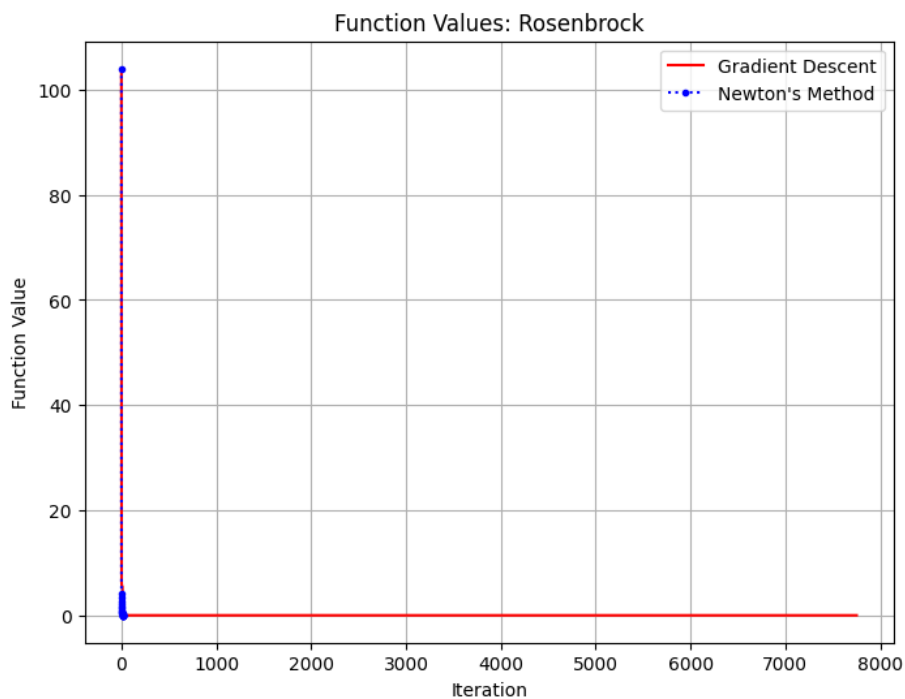
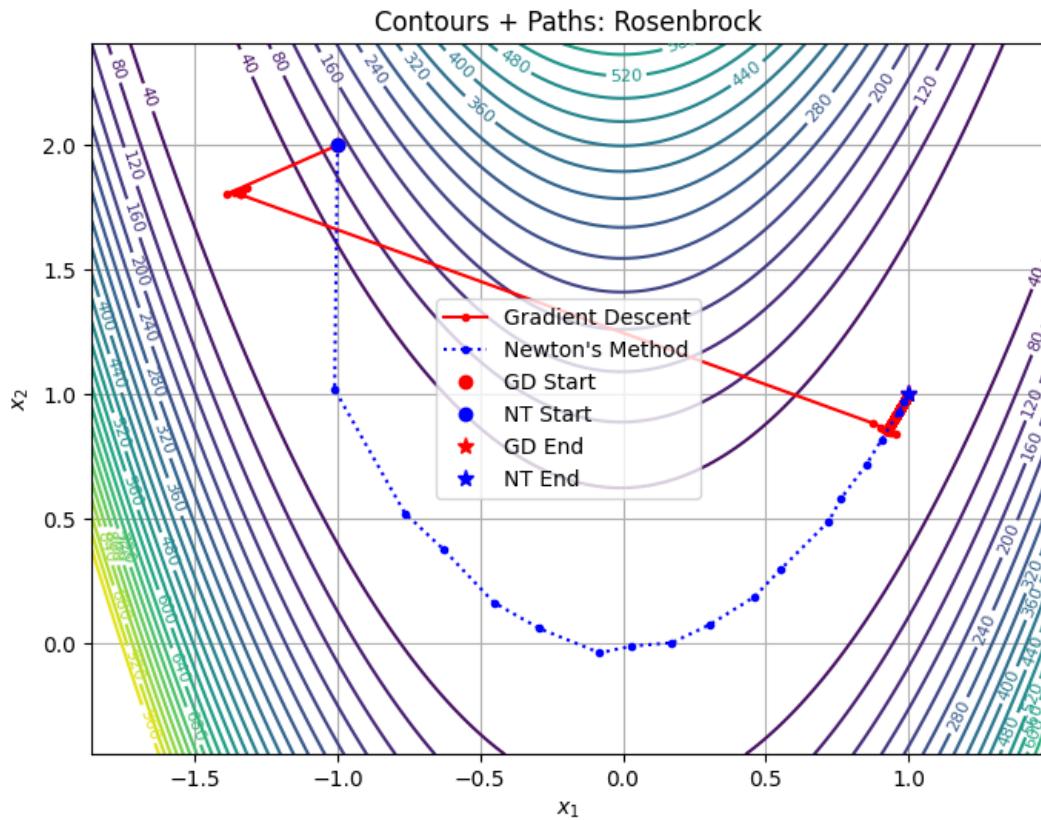
function: quadratic\_rotated, method: gradient\_descent, lter 100:  $x = [0.07682763 \ 0.13062022]$ ,  $f(x) = 0.023112584192700524$ , success = False

function: quadratic\_rotated, method: newton\_method, lter 1:  $x = [0. \ 0.]$ ,  $f(x) = 0.0$ , success = True



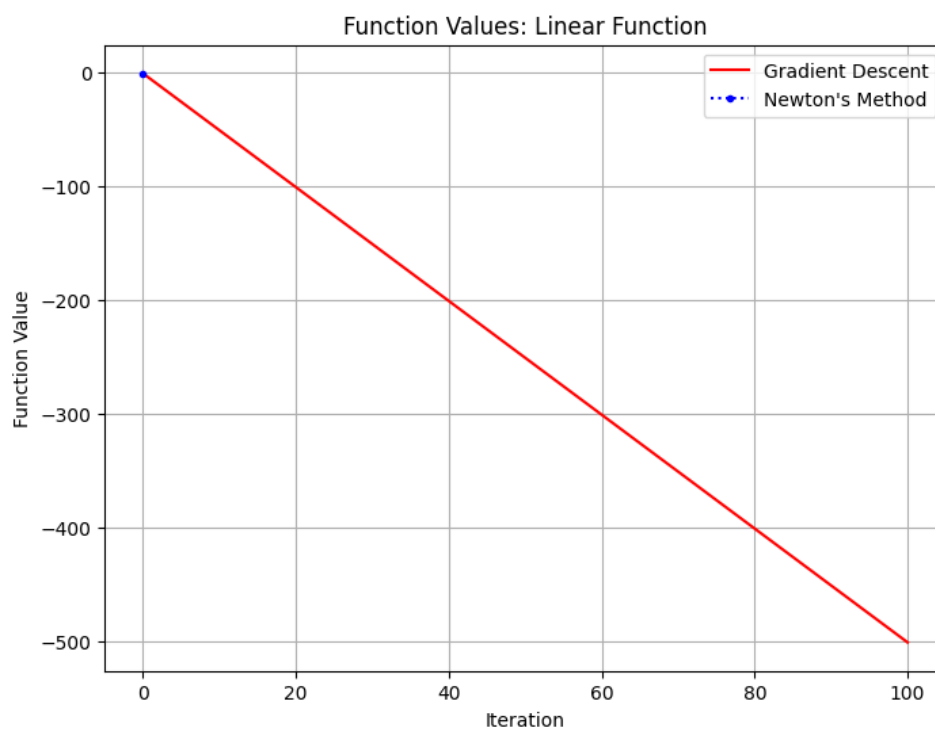
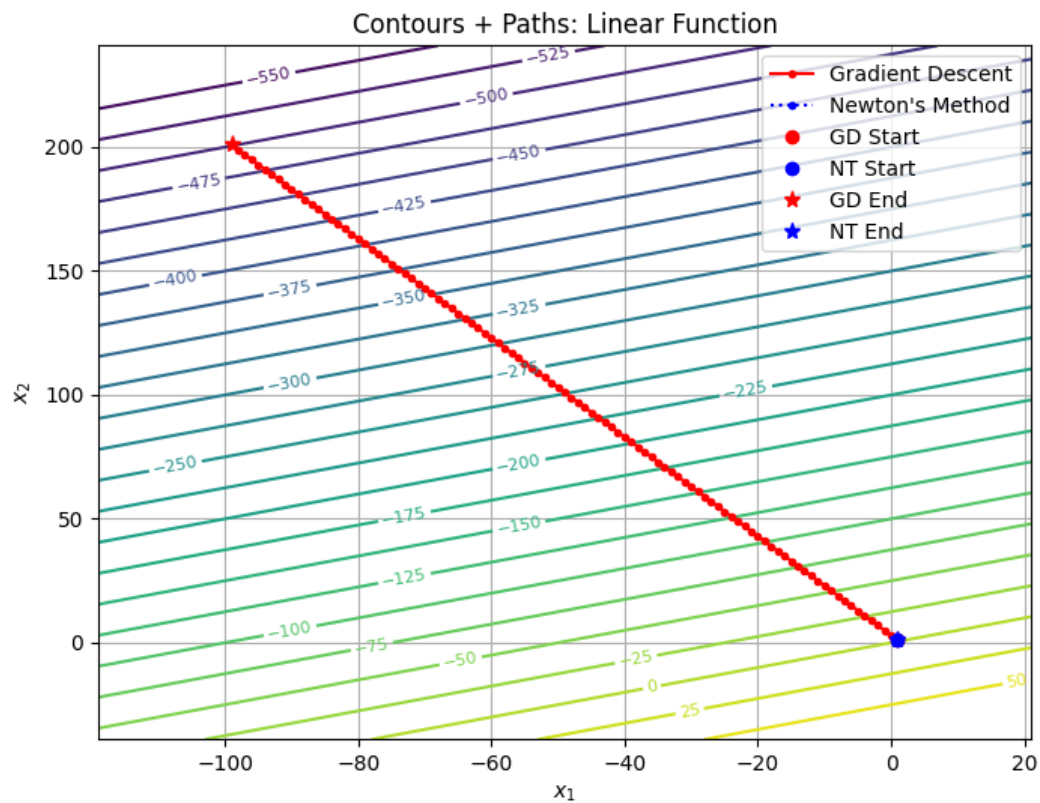
function: **rosenbrock**, method: **gradient\_descent**, Iter 7750:  $x = [0.99989437, 0.99978771]$ ,  $f(x) = 1.1267173492101773e-08$ , success = True

function: **rosenbrock**, method: **newton\_method**, Iter 20:  $x = [1. 1.]$ ,  $f(x) = 6.668339839446365e-30$ , success = True



function: **linear\_function**, method: **gradient\_descent**, Iter 100:  $x = [-99.201.]$ ,  $f(x) = -501.0$ , success = False

function: **linear\_function**, method: **newton\_method**, Iter 100:  $x = [1.1.]$ ,  $f(x) = -1.0$ , success = False



function: **exponential\_triangle**, method: **gradient\_descent**, Iter 28:  $x = [-3.46572886e-01 \ -9.76799665e-07]$ ,  $f(x) = 2.559266696664345$ , success = True

function: **exponential\_triangle**, method: **newton\_method**, Iter 8:  $x = [-3.46573590e-01 \ -2.17907897e-17]$ ,  $f(x) = 2.5592666966582156$ , success = True

