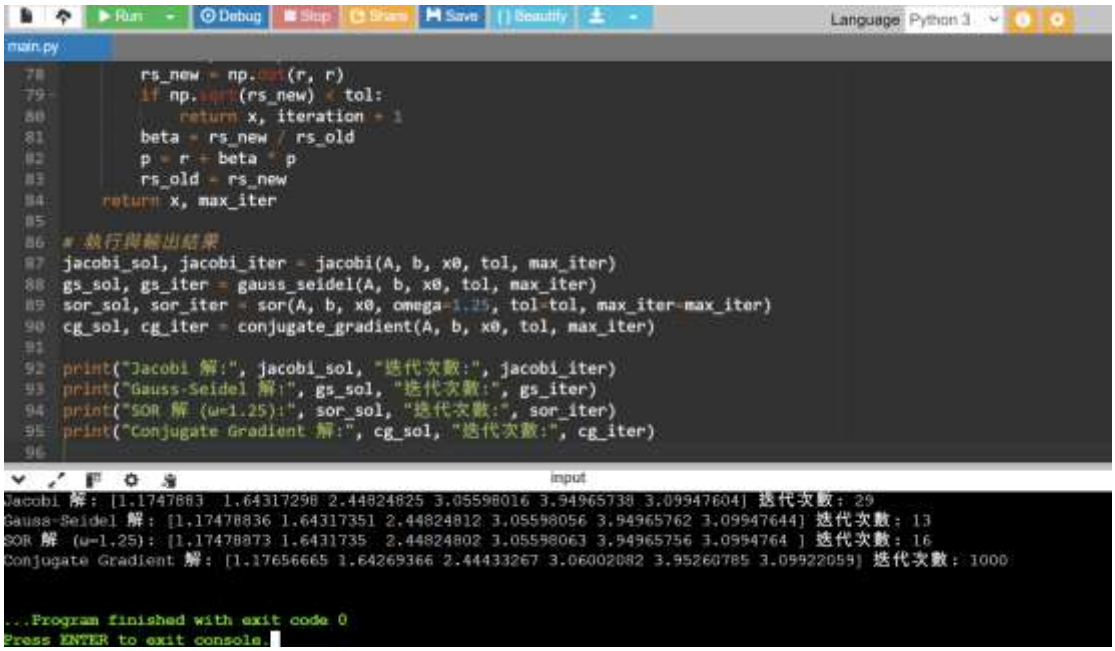
Run
Debug
Stop
Share
Save
{ } Beautify
Language
Python 3
main.py
rs_new = np.dot(r, r)
if np.sqrt(rs_new) < tol:
return x, iteration + 1
beta = rs_new / rs_old
p = r + beta * p
rs_old = rs_new
return x, max_iter
執行與輸出結果
jacobi_sol, jacobi_iter = jacobi(A, b, x0, tol, max_iter)
gs_sol, gs_iter = gauss_seidel(A, b, x0, tol, max_iter)
sor_sol, sor_iter = sor(A, b, x0, omega=1.25, tol=tol, max_iter=max_iter)
cg_sol, cg_iter = conjugate_gradient(A, b, x0, tol, max_iter)
print("Jacobi 解:", jacobi_sol, "迭代次數:", jacobi_iter)
print("Gauss-Seidel 解:", gs_sol, "迭代次數:", gs_iter)
print("SOR 解 (ω=1.25):", sor_sol, "迭代次數:", sor_iter)
print("Conjugate Gradient 解:", cg_sol, "迭代次數:", cg_iter)
input
Jacobi 解: [1.1747883 1.64317298 2.44824825 3.05598016 3.94965738 3.09947604] 迭代次數: 29
Gauss-Seidel 解: [1.17478836 1.64317351 2.44824812 3.05598056 3.94965762 3.09947644] 迭代次數: 13
SOR 解 (ω=1.25): [1.17478873 1.6431735 2.44824802 3.05598063 3.94965756 3.0994764] 迭代次數: 16
Conjugate Gradient 解: [1.17656665 1.64269366 2.44433267 3.06002082 3.95260785 3.09922059] 迭代次數: 1000
...Program finished with exit code 0
Press ENTER to exit console.