CS416: Optimisation Methods

1630022

March 2, 2020

1 Pure Newton

1.1 Implementation

Implementation of the pure newton method follows a direct application of the expression for the newton direction using `numpy.linalg.sol`
An additional clause was added to detect if the algorithm is going to fail to converge. This done simply by aborting the pure newton method if an iteration of x is more than 10^{10} away from $(0,0)$. This clause is sufficient for all the functions that will be considered in this document.

1.2 Function g

We are considering a function $g : \mathbb{R}^2 \to \mathbb{R}$

$$g(x_1, x_2) = \sqrt{x_1^2 + 1} + \sqrt{x_2^2 + 1}$$

1. The gradient and hessian of g can be calculated as follows.

$$\nabla g(x_1, x_2) = \begin{bmatrix} x_1/\sqrt{x_1^2+1} \\ x_2/\sqrt{x_2^2+1} \end{bmatrix}$$

$$\mathbf{H}_g = \begin{bmatrix} \frac{\sqrt{x_1^2+1}}{x_1^4+2x_1^2+1} & 0 \\ 0 & \frac{\sqrt{x_2^2+1}}{x_2^4+2x_2^2+1} \end{bmatrix}$$

2. Since $\mathbf{H}_g$ is a diagonal matrix the eigenvalues are the diagonal entries to confirm that $\mathbf{H}_g$ is positive definite we need to confirm that the diagonal entries are strictly positive, which follows immediately from the fact that the numerator is always positive and that the denominator can be factorised as $\left(x_i^2 + 1\right)^2 \;\; i \in \{1, 2\}$.

3. A minimum must satisfy $\nabla g(x_1, x_2) = (0, 0)$. From our derivation of $\nabla g(x_1, x_2)$ in 1. we see that this can only have once solution where $\mathbf{x} = (0, 0)$.

4. The function is not m-strongly convex for any $m > 0$. To be m-strongly convex the function must satisfy the following:

$$\exists m > 0 : \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^2 : g(\mathbf{y}) \geq g(\mathbf{x}) + \langle \nabla g(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle \rangle + \frac{m}{2} \|\mathbf{y} - \mathbf{x}\|_2^2$$

However, we see that this would require the diagonal elements of the hessian to be bounded from below by some strictly positive constant and $\mathbf{H}_g$ is only bounded below by 0.

1.3 Numerical Tests

1. Starting at $\mathbf{x}_0 = (1, 1)^T$ converges in 37 iterations and stops at $\left(-8.92748047 \cdot 10^{-13}, -8.92748047 \cdot 10^{-13}\right)^T$

2. Starting at $\mathbf{x}_0 = (10, 10)^T$ fails to converge.

2 Damped Newton

2.1 Implementation

The implementation of the damped newton method follows the same structure as the pure newton method discussed above.

2.2 Numerical Tests

1. Starting at $\mathbf{x}_0 = (1,1)^T$ converges in 1 iteration and stops at $\left(1.11022302 \cdot 10^{-16}, 1.11022302 \cdot 10^{-16}\right)^T$
2. Starting at $\mathbf{x}_0 = (10,10)^T$ converges in 18 iterations and stops at $\left(-1.2409507 \cdot 10^{-15}, -1.2409507 \cdot 10^{-15}\right)^T$

The performance of damped newtons method is much greater than the pure newton method allowing the first test case to converge in a single iteration instead of 37 and the second example converges instead of diverges.

3 Hybrid Gradient-Newton

3.1 Implementation

Again the implementation mostly follows the previous two methods however we now need a method of determining whether a matrix is positive definite. To do this I use `numpy.linalg.eigvals` to check that all the eigenvalues are strictly positive. Although this method is less efficient than computing the Cholesky decomposition and catching the `LinAlg` error it is simpler and easier to read and it will only be used for 2×2 matrices.

3.2 Numerical Tests

We will be attempting to minimize the following function:

$$h(\mathbf{x}) = (h_1(\mathbf{x}))^2 + (h_2(\mathbf{x}))^2$$

$$\begin{aligned} h_1(\mathbf{x}) &= -13 + x_1 + ((5 - x_2) \cdot x_2 - 2) \cdot x_2 \\ h_2(\mathbf{x}) &= -29 + x_1 + ((x_2 + 1) \cdot x_2 - 14) \cdot x_2 \end{aligned}$$

1. $\mathbf{x}_0 = (-50, 5)^T$
   (a) Hybrid Method converges in 8 iterations to $(5,4)^T$
   (b) Damped Method converges in 8 iterations to $(5,4)^T$
2. $\mathbf{x}_0 = (20, 7)^T$
   (a) Hybrid Method converges in 8 iterations to $(5,4)^T$
   (b) Damped Method converges in 8 iterations to $(5,4)^T$
3. $\mathbf{x}_0 = (20, -18)^T$
   (a) Hybrid Method converges in 16 iterations to $(11.41277899, -0.89680525)^T$
   (b) Damped Method converges in 16 iterations to $(11.41277899, -0.89680525)^T$
4. $\mathbf{x}_0 = (5, -10)^T$
   (a) Hybrid Method converges in 13 iterations to $(11.41277899, -0.89680525)^T$
   (b) Damped Method converges in 13 iterations to $(11.41277899, -0.89680525)^T$

As we can see there is no difference between the Hybrid Gradient-Newton Method and the Damped Newton Method upon closer inspection this is because the Hessian $\mathbf{H}_h$ is positive definite for all iterations of the Hybrid Gradient-Newton Method. This means that the iterations of the two methods are identical.