**Objective:** We aim to optimize the given function to find the global minima.

**Description of function:** The function takes the sum of the term i\*xi over the first 100 natural numbers.

**Breaking down the function:** Since, the summation operator is distributive, the first function may be re-written as:

Breaking down the function allows us to compute the gradient and hessian more easily. The gradient of the function is a 100 \* 1 vector, where each element is the partial differentiation of f(x) with respect to the ith  value of x, where ‘i’ is the current row of the vector. This give us:

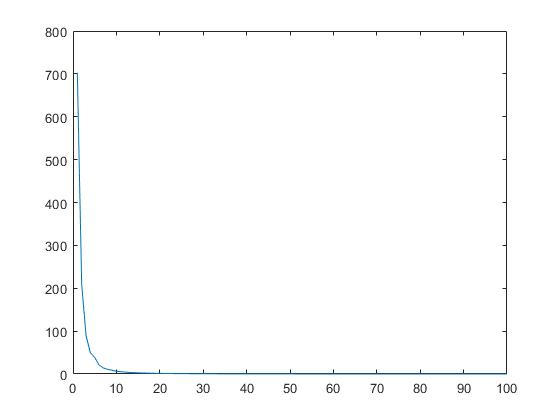
Till x100. So at every row the value of the gradient is 2\*i\*x(i).

Similarly, the hessian can be written as a 100\*100 matrix where   
  
H(x) = 2\*i ; if i == j

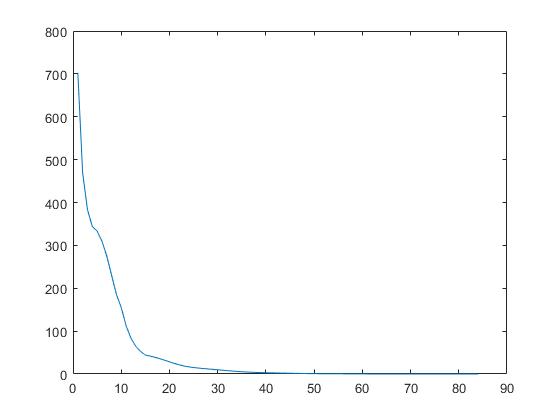
= 0; otherwise.

**Testing the three optimization algorithms on the function:**

1. Gradient Descent: The function reaches the global minimum of 0 before the maximum number of iterations are reached. The following graph, shows errors on the y axis, vs iterations on x.



1. Newton Method: The function converges extremely fast:
2. Quasi Newton Method: The function converges to the global minimum but takes a bit more time than that of gradient descent or the newton method. The following graph shows the errors on the y axis vs the iterations on x.



**Final thoughts:**

1. Function one converges for all three methods, described.
2. It is easy to evaluate since at every step it multiplies the square of the x vector.