

## Contents

---

- [Using for loop code](#)
- [Using built-in sum function code](#)
- [Summary](#)

## Using for loop code

---

```
tic
%set accumulator s
s = 0;

for i = 1:1000000
    %norm() computes ell-2 norm of columns of X
    s = s + norm(X(:,i));
end
toc

disp("final output for for loop: "+s);
```

Elapsed time is 1.113525 seconds.  
final output for for loop: 1380518.3753

## Using built-in sum function code

---

```
tic
%X.^2 squares the matrix X element-wise

%the inner sum adds elements in each column of X.^2 together and
%store result of the sums into 1x1000000 row vector

%then .^0.5 applies element-wise square root on row vector
%Finally the outer sum adds the rows of row vector into s2, which
%holds the desired output.
s2 = sum(sum(X.^2).^0.5);
toc

disp("final output for built-in sum function: "+s2);
```

Elapsed time is 0.081590 seconds.  
final output for built-in sum function: 1380518.3753

## Summary

---

On average, the sum function version performed around 10-14 times faster than the for loop, using the tic method of measuring time.

We see from this lab that it is always better to use built-in functions whenever applicable. This is because sum is a function so commonly used so that matlab provides an optimized version of it implemented in C, and C is a compiled language much faster than an interpreted language such as matlab.

