Math 170A Lecture Notes (Professor: Lei Huang)

Isabelle Mills

January 13, 2024

Week 1 Notes (1/8 - 1/12/2024)

For this class, we shall define the number of flops an algoirthm takes as the number of individual $+$, $-$, $\times$, $/$, and $\sqrt{\ }$ operations on real numbers used in the algorithm.

> For example: taking the inner product of two vectors $\overrightarrow{u} = (u_1, u_2, \ldots, u_n)$ and $\overrightarrow{v} = (v_1, v_2, \ldots, v_n)$ requires n multiplications and $(n-1)$ additions. So we'll say it has a flops count of $2n-1$.

Technically, the word "flop" stands for floating (point) operation. Based on that knowledge, hopefully it is easier to guess what is and is not a flop. For instance, observe the code written below for taking an inner product of two n-vectors.

```
P = 0
for i = 1:n
   P = P + v(i) * w(i)
end
```

Neither incrementing `i` nor initializing any other variables are counted towards the flop number. Because the code does `n` additions and multiplications between floating point numbers, we say this function has $2n$ flops.

Here is how we formally define Big-O Notation:

For a sequence a_n, we define $a_n = O(b_n)$ if there exists real constants $C, N \geq 0$ such that for $n \geq N$, $\;a_n \leq Cb_n$.

Example problem:

```
1   function x = lowertriangsolve(L, b)
2      N = size(L);
3      n=size(b,1);
4      x=b;
5
6      for i=1:N(1):
7         for j=1:i-1
8            x(i) = x(i) - L(i,j)*x(j);
9         end
10
11        if L(i, j) == 0
12           error('matrix is singular')
13        end
14        x(i) = x(i)/L(i,i);
15     end
16  end
```

Inside the main for loop (lines 6-15):

Line 8 has $2(\mathrm{i}-1)$ flops.

Line 14 has an additional flop.

Thus, the total number of flops is:

$$\sum_{\mathrm{i}=1}^{n}(2\mathrm{i}-1) = 2\left(\sum_{\mathrm{i}=1}^{n}\mathrm{i}\right) - n$$

This then simplifies to:

$$2\left(\frac{n(n+1)}{2}\right) - n = 1n^2$$

So `x` has $O(n^2)$ flops as `n` goes to infinity.

test for Wednesday