These are the lecture notes for CSC349A Numerical Analysis taught by Rich Little in the Spring of 2018. They roughly correspond to the material covered in each lecture in the classroom but the actual classroom presentation might deviate significantly from them depending on the flow of the course delivery. They are provide as a reference to the instructor as well as supporting material for students who miss the lectures. They are simply notes to support the lecture so the text is not detailed and they are not thoroughly checked. Use at your own risk.

1 Stability and Condition

In analyzing the effects of errors in a computation to solve a problem, the concepts of "stability" and "condition" distinguish between whether the algorithm (the procedure for computing a solution to the problem) is satisfactory, or if the problem is such that no algorithm can be expected to reasonably solve the problem. The concepts involved are: **stable/unstable algorithm** and **well-conditioned/ill-conditioned problem**.

Definition: A problem whose (exact) solution can change greatly with small changes in the data defining the problem is called **ill-conditioned**.

Suppose that for the original problem we have using exact arithmetic :

$$\text{data} \quad \{d_i\} \rightarrow \quad \text{exact solution}\{r_i\}$$

and we create a perturbed problem also using exact arithmetic:

$$\text{data} \quad \{\hat{d}_i\} = \{d_i + \varepsilon_i\} \rightarrow \quad \text{exact solution}\{\hat{r}_i\}$$

where $|\frac{\varepsilon_i}{d_i}|$ is small.

If there exist small ε_i such that $\{\hat{r}_i\}$ are not close to $\{r_i\}$, then the problem is **ill-conditioned**.

If $\{\hat{r}_i\} \approx \{r_i\}$ for **all** small ε_i, then the problem is **well-conditioned**.

Note: The condition of a problem has nothing to do with floating-point arithmetic or round-off error; it is defined in terms of exact computation. However, if a problem ill-conditioned, it will be difficult (or impossible) to solve accurately using floating-point arithmetic.

1.1 Example 1

Suppose we want to solve $y = \frac{x}{1-x}$ for values of x near 1. For example, if $x = 0.93$ then $y = \frac{0.93}{1-0.93} = 13.2857\cdots$. Now, let the perturbed data be $\hat{x} = 0.94$. Here, we get $\hat{y} = \frac{0.94}{1-0.94} = 15.666\cdots$.

The relative error from the perturbation is given by $\frac{|x-\hat{x}|}{|x|} = 0.01075\cdots$ which is roughly 1%. On the other hand, the relative error in the results is $\frac{|13.2857-15.6666|}{|13.2857|} = 0.1792\cdots$ or about 18%. So, this problem is considered to be ill-conditioned.

Now, do the same for $x = -0.93$ and $\hat{x} = -0.94$. Here, we get $y = -0.4818\cdots$ and $\hat{y} = -0.4845\cdots$, which is a relative difference of $0.00560\cdots$. For this data the problem does not prove to be ill-contitioned although this is only one value of ε it is in fact well-conditioned for this x. We will see this in a little bit.

1.2 Condition Number

The **condition number** is another approach to analyzing the condition of a problem if the first derivative of the quantity $f(x)$ being computed can be determined. By the Taylor polynomial approximation of order $n = 1$ for $f(x)$ expanded around $\tilde{x}$ we have:

$$f(x) \approx f(\tilde{x}) + f'(\tilde{x})(x - \tilde{x})$$

which implies that:

$$\frac{f(x) - f(\tilde{x})}{f(\tilde{x})} \approx \frac{\tilde{x}f'(\tilde{x})}{f(\tilde{x})} \left(\frac{x - \tilde{x}}{\tilde{x}}\right)$$

If $\tilde{x}$ is some small pertubation os x, then the left hand side above is the *relative change* in $f(x)$ as x is perturbed to $\tilde{x}$. Thus,

$$\text{relative change in } f(x) \approx \left(\frac{\tilde{x}f'(\tilde{x})}{f(\tilde{x})}\right) \times \text{ relative change in } x$$

The quantity $\frac{\tilde{x}f'(\tilde{x})}{f(\tilde{x})}$ is called a **condition number** for the computation of $f(x)$. If this number is "large", then $f(x)$ is ill-conditioned; if this number is "small", then $f(x)$ is well-conditioned.

1.3 Example 1 revisited

Recall, $f(x) = \frac{x}{1-x}$ and so $f'(x) = \frac{1}{(1-x)^2}$. Thus,

$$\frac{xf'(x)}{f(x)} = \frac{x/(1-x)^2}{x/(1-x)} = \frac{1}{1-x}$$

Therefore, when $x = 0.93$, the condition number is $\frac{1}{1-0.93} = 14.2857\cdots$ which greater than 1 and thus ill-conditioned. On the other hand, when $x = -0.93$, then the condtion number is $\frac{1}{1+0.93} = 0.5181\cdots$ which less than 1 and thus well-conditioned. These results are consistent with our analysis above.

2 Stability of an algorithm

A computation is **numerically unstable** if the uncertainty of the input values is greatly magnified by the numerical method.
Definition: An algorithm is said to be **stable** (for a class of problems) if it determines a computed solution (using floating-point arithmetic) that is close to the exact solution of some (small) perturbation of the given problem.

Suppose that for the original problem we have using floating-point computation:

$$\text{data} \quad \{d_i\} \rightarrow \quad \text{computed solution}\{r_i\}$$

and we create a perturbed problem using exact computation:

$$\text{data} \quad \{\hat{d}_i\} = \{d_i + \varepsilon_i\} \rightarrow \quad \text{exact solution}\{\hat{r}_i\}$$

where $|\frac{\varepsilon_i}{d_i}|$ is small.

If there exist data $\hat{d}_i \approx d_i$ (small ε_i for all i) such that $\hat{r}_i \approx r_i$ for all i, then the algorithm is said to be **stable**.

If there exists **no set** of data $\{\hat{d}_i\}$ close to $\{d_i\}$ such that $\hat{r}_i \approx r_i$ for all i, then the algorithm is said to be **unstable**

Meaning of numerical stability: the effect of uncertainty in the input data or of the floating-point artihmetic (the round-off error) is no worse that the effect of slightly perturbing the given problem, and solving the perturbed problem exactly.

See Handout 7 for examples on stability analysis.