Comparison of (direct) least squares algorithms

	Normal eqns	QR	SVD
$m \approx n$	$\frac{4}{3}n^3$	$\frac{4}{3}n^3$	$\approx 13n^3$
$m \gg n$	mn^2	$2mn^2$	$2mn^2$

TL;DR: Normal equations faster than QR faster than SVD.

Is that all there is to say?

```
>> A = [1 1 2; 1 2 3; 3 1 4; 1 2 3+1e-8];
>> y = A*[3;4;5];
>> [Q1, R1] = qr(A, 0); x1 = R1 \ (Q1'*y)
x1 =
    2.999999625816564e+00
    3.999999625816566e+00
    5.000000374183434e+00
>> [U, S, V] = svd(A, 0); x2 = V*pinv(S)*U'*y
x2 =
    2.999999523162842e+00
    4.000000000000000e+00
    5.000000000000000e+00
>> x3 = (A'*A) \ (A'*y)
Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate. RCOND = 2.619349e-18.
x3 =
   -3.356626253322333e+01
   -3.256626267687653e+01
    4.156626257240148e+01
```

Also, residuals don't tell us much about the accuracy of these solutions:

```
>> norm(A*x2 - y)
ans =
    1.651812369889142e-06
>> norm(A*x3 - y)
ans =
    2.676376351036689e-07
```

And parentheses in x_2 matter:

```
>> [U, S, V] = svd(A, 0); x4 = V*(pinv(S)*(U'*y))
x4 =
    3.000000371542074e+00
    4.000000371542088e+00
    4.999999628457917e+00
>> norm(A*x4 - y)
ans =
    2.190039795013723e-14
```

Sensitivity issues

```
>> A = [1 1 2; 1 2 3; 3 1 4; 1 2 3+1e-8];
```

A is at distance 10^{-8} from a non-full-rank matrix:

```
>> svd(A)
ans =
   7.553509024056715
   1.715954977117343
   0.000000004225771
```

This will be a common trend: problems close to unsolvable are numerically troublesome.

Big questions

Why are normal equations much less accurate than QR/SVD?
How can we assess the accuracy of a computed solution?

To understand more, we need to study sensitivity and stability.

Sensitivity of a problem

Computational problems map an input to an output.
Example: solving a linear system: input: $A, \mathbf{y}$; output: $\mathbf{x} = A^{-1}\mathbf{y}$.
Example: training a neural network: input: training data x_i, y_i; output: weights w.

Basic question: how does the output of a problem change when we change its input.

Example: if I turn the shower tap by 10 degrees, how much does the water temperature change?

Example: compute $f(x) = x^2$. If I change x to $\tilde{x} = x + \delta$, the output becomes $f(\tilde{x}) = x^2 + 2\delta x + \delta^2$.

Change in input: $|\tilde{x} - x| = |\delta|$.
Change in output: $|\tilde{x}^2 - x^2| = |2\delta x + \delta^2|$.

Definition: (absolute) condition number

Definition The (absolute) condition number of a function $f : \mathbb{R} \to \mathbb{R}$ is the best bound K of the form

$$|f(x+\delta) - f(x)| \leq K|\delta| + \underbrace{o(\delta)}_{\text{higher-order terms: } \delta^2,\ \delta^3,\ \ldots}$$

Or, more formally,

$$\kappa_{abs}(f, x) = \lim_{\delta \to 0} \frac{|f(x+\delta) - f(x)|}{|\delta|}.$$

For a scalar-valued function, this is essentially the norm of the derivative (when it exists):

$$\kappa_{abs}(f, x) = \left| \frac{df}{dx} \right|.$$

Example: absolute condition number

We can generalize the definition to problems with multiple inputs.

Example: computing $f(x, y) = x^2 y$, input x. If I change x to $\tilde{x} = x + \delta$, the output becomes $(x + \delta)^2 y = x^2 y + 2xy\delta + \delta^2 y$.

Change in output:

$$|f(\tilde{x}, y) - f(x, y)| = |(x+\delta)^2 y - x^2 y| = |2xy\delta + y\delta^2| = \underbrace{2|xy|}_{=\kappa_{abs}(f,x)} |\delta| + O(\delta^2).$$

Or:

$$\lim_{|\delta| \to 0} \frac{|2xy\delta + y\delta^2|}{|\delta|} = 2|xy|.$$

Analogously, one can define a condition number with respect to y, and it is $\frac{\partial f}{\partial y}(x, y)$.

Functions of vectors/matrices

For vector and matrix arguments, we make two changes:

- use norms rather than absolute values;
- take the largest change over all possible directions $\mathbf{d} \in \mathbb{R}^n$.

Indeed, functions of several variables can change faster in some directions than in others (cfr. tomography).

$$\|f(\mathbf{x}+\mathbf{d}) - f(\mathbf{x})\| \leq K\|\mathbf{d}\| + \underbrace{o(\|\mathbf{d}\|)}_{\text{higher-order terms: } \|\mathbf{d}\|^2,\ \|\mathbf{d}\|^3,\ \dots}$$

The formal definition is slightly more involved:

$$\kappa_{abs}(f, \mathbf{x}) = \lim_{\delta \to 0} \sup_{\|\mathbf{d}\| \leq \delta} \frac{\|f(\mathbf{x}+\mathbf{d}) - f(\mathbf{x})\|}{\|\mathbf{d}\|}.$$

For differentiable real-valued functions, $\kappa_{abs}(f, \mathbf{x}) = \|\nabla f_{\mathbf{x}}\|$ (for the norm-2, at least).
For a general norm and $f : \mathbb{R}^m \to \mathbb{R}^n$, $\kappa_{abs}(f, \mathbf{x})$ is the norm of the Jacobian matrix.

Relative condition number

Example Sensitivity of $f(x) = x^2$ around $x = 1000$: perturbing the input to $\tilde{x} = 1000.01$ changes the output from $f(x) = 1\,000\,000$ to $f(\tilde{x}) = 1\,000\,020.000\,1$.

Change in input: 0.01. Change in output: $20.000\,1$.

This does not fit our intuition that $f(x)$ and $f(\tilde{x})$ are close. It is better to measure input/output changes as relative changes:

Relative change in input: $\frac{\|\tilde{x}-x\|}{\|x\|} = 10^{-5}$.
Relative change in output: $\frac{\|f(\tilde{x})-f(x)\|}{\|f(x)\|} \approx 2 \times 10^{-5}$.

Definition The relative condition number of a function f is

$$\kappa_{rel}(f, \mathbf{x}) = \lim_{\delta \to 0} \sup_{\|\mathbf{d}\| \leq \delta} \frac{\frac{\|f(\mathbf{x}+\mathbf{d})-f(\mathbf{x})\|}{\|f(\mathbf{x})\|}}{\frac{\|\mathbf{d}\|}{\|\mathbf{x}\|}} = \kappa_{abs}(f, \mathbf{x}) \frac{\|\mathbf{x}\|}{\|f(\mathbf{x})\|}.$$

Why relative errors?

Absolute errors are useless without a reference point:

Example We have built a neural network to estimate an optimal price x. In our experiments, it computes a price $\tilde{x}$ with $|\tilde{x} - x| = 0.823\$$.

- If x is the salary of an NBA player, e.g., $x = 10^7\$$, it's a great estimate;
- If x is the optimal price of a nail, e.g., $x = 0.001\$$, it's a terrible one.

Relative errors:

- $\frac{|\tilde{x}-x|}{|x|} \approx 1$: very bad accuracy; it's just a number with the same order of magnitude.
- $\frac{|\tilde{x}-x|}{|x|} \approx 10^{-3}$: about 3 correct significant digits.
- $\frac{|\tilde{x}-x|}{|x|} \approx 10^{-16}$: about 16 correct digits; we can't do better typically (with `double` precision arithmetic).

Use relative errors

I cannot stress it enough: use relative errors whenever you have to measure if something is small or large: thresholds in algorithms, error measures, stability checks, etc.

```
>> norm(A - Q*R)
ans =
   7.2625e+00 % is this good or bad?
>> norm(A - Q*R) / norm(A)
ans =
   2.1162e-16 % good!
```

This “whenever” includes, in particular, your project.

Remarks

TL;DR: some problems are bad (highly sensitive); errors in the input will be amplified.

Condition number is a theoretical property, related to the derivative.

It does not depend on floating point computations, or on the choice of algorithms. . .

Good metaphor: a minimal turn of the knob in a shower can turn the water from freezing to scalding.

It is a warning sign: if your model calls for computing an ill-conditioned quantity, the solution is probably going to be useless.

Exercises

1. What is the absolute condition number of $f(x, y) = x + 2y$ with respect to its input $y \in \mathbb{R}$?
2. Show that the absolute condition number of solving a linear system $A\mathbf{x} = \mathbf{y}$ w.r.t the input $\mathbf{y}$ is $\|A^{-1}\|$.
3. What is the relative condition number of $f(x, y) = x - y$ (with $x, y \in \mathbb{R}$)?
4. Let f be a function with Lipschitz constant L. What can we say about its absolute condition number?

Book references: Trefethen-Bau, Lecture 12.