SIMPLE ALGORITHMS FOR LINEAR SYSTEMS AND LEAST SQUARES

David F. Gleich

August 21, 2023

We are going to look at a number of algorithms for solving linear systems of equations and least squares problems. These are all going to be "simple" algorithms in that we are going to derive them by using a few simple ideas that result from studying the equations that define a linear system.

The algorithms we are going to study right now are all of the flavor:

$$\text{start} \to \text{improve} \to \text{improve} \to \cdots$$

or as I like to think of them

$$\text{guess} \to \text{check} \to \text{correct} \to \text{check} \to \text{correct} \to \cdots.$$

That is to say, these are going to be "iterative" algorithms. We will construct a sequence of vectors that *hopefully* converges to the solution of the linear system of equations of least squares problem.

Learning objectives

1. A recap of what it means to solve a linear system.
2. Why we only talk about solving full rank systems.
3. The Neumann series algorithm for solving some linear systems.
4. How to evaluate an approximate solution.

Note that this section could also come after discussing vector and matrix norms, as we will use those in our discussion of approximate solutions.

We assume that you might already be familiar with the Euclidean vector norm: $\|\mathbf{x}\| = \sqrt{\sum_i x_i^2}$ from past experiences.

1 REVIEW OF LINEAR SYSTEMS OF EQUATIONS

Let's start with some basic properties of linear systems of equations.[1] A linear system

$$\boldsymbol{A}\mathbf{x} = \mathbf{b}$$

represents a set of equations

$$\begin{aligned} A_{1,1}x_1 + A_{1,2}x_2 + \ldots + A_{1,n}x_n &= b_1 \\ A_{2,1}x_1 + A_{2,2}x_2 + \ldots + A_{2,n}x_n &= b_2 \\ \cdots A_{m,1}x_1 + A_{m,2}x_2 + \ldots + A_{m,n}x_n &= b_m. \end{aligned}$$

This is a relationship described by m equations and n unknowns. These come from an enormous diversity of scenarios as detailed in previous lectures and notes.

If there are fewer equations than unknowns ($m < n$), then the system is called underdetermined and it may have 0, 1, or an infinite set of solutions. If $m = n$, the system is called *square* and the system can have 0, 1, or an infinite set of solutions. And if $m > n$, the system is called *over determined* and it can have 0, 1, or an infinite set of solutions.

The above expressions are all 0, 1, or an infinite number. As a small consideration, why can't we have *two* solutions but not an infinite number? This is a property of a linear set of equations that is part of what makes them special and *easy to solve*. Suppose we have two solutions $\mathbf{x}$ and $\mathbf{y}$

$$\boldsymbol{A}\mathbf{x} = \mathbf{b} \qquad \boldsymbol{A}\mathbf{y} = \mathbf{b} \qquad \mathbf{x} \neq \mathbf{y}.$$

Then *any combination of those solutions* is also a solution, such as

$$\boldsymbol{A}(\gamma\mathbf{x} + (1-\gamma)\mathbf{y}) = \gamma\boldsymbol{A}\mathbf{x} + (1-\gamma)\boldsymbol{A}\mathbf{y}) = \gamma\mathbf{b} + (1-\gamma)\mathbf{b} = \mathbf{b}$$

and we have this relationship for all γ. This is an infinite set of solutions.

How can we have zero solutions to an underdetermined system? This is because the above characterization did not prescribe anything about the *dependencies* among solutions. For instance, here are two equations

$$\begin{aligned} -x + y - z &= 2 \\ -x + y - z &= 3. \end{aligned}$$

[1] This section should be a review.

Note that these are the same equation with a different value. There is no solution. As a matrix $\boldsymbol{A}$, this scenario is a 2×3 matrix with rank 1.

Here is a fun case to consider. Let $\boldsymbol{A} = \mathbf{y}\mathbf{y}^T$. When does $\boldsymbol{A}\mathbf{x} = \mathbf{b}$ have a solution? When does it have no solution? Describe a procedure to find the solution.

For this reason, typically people have chosen to discuss equations in terms of *full rank* matrices. An $m \times n$ matrix is full rank if the rank is $\min(m, n)$. In this case, underdetermined problems $(m < n)$ always have an infinite number of solutions. Overdetermined problems $(m > n)$ have either 1 or 0 solutions. Square systems have only one *unique* solution.

Unfortunately, all this flexibility in terms of the number of solutions makes it hard to discuss algorithms. Consequently,

When we consider solving *linear systems, we always focus on the* square, full-rank *case.*

There are a large number of known ways to characterize when a square system of linear equations is full rank.

- rank($\boldsymbol{A}$) is n (or m since $m = n$)
- $\boldsymbol{A}$ is invertible
- the columns of $\boldsymbol{A}$ are linearly independent
- the rows of $\boldsymbol{A}$ are linearly independent
- the determinant of $\boldsymbol{A}$ is one
- the eigenvalues of $\boldsymbol{A}$ are all non-zero
- the singular values of $\boldsymbol{A}$ are all non-zero.

When $\boldsymbol{A}$ is square and full rank, then there exists a matrix $\boldsymbol{Y}$ such that $\boldsymbol{A}\boldsymbol{Y} = \boldsymbol{I}$ and $\boldsymbol{Y}\boldsymbol{A} = \boldsymbol{I}$. This matrix $\boldsymbol{Y}$ is called the *inverse* and is usually written $\boldsymbol{A}^{-1}$.

Given a linear system $\boldsymbol{A}\mathbf{x} = \mathbf{b}$, then we can multiply both sides by $\boldsymbol{Y}$ and get $\boldsymbol{Y}\boldsymbol{A}\mathbf{x} = \boldsymbol{Y}\mathbf{b}$ where $(\boldsymbol{Y}\boldsymbol{A}) = \boldsymbol{I}$, so we get $\mathbf{x} = \boldsymbol{Y}\mathbf{b}$ or $\mathbf{x} = \boldsymbol{A}^{-1}\mathbf{b}$. There are many, many interpretations of this statement.

2 A FIRST METHOD

This isn't the order I'm hoping to do these in eventually, but because of the homework, I want to go over this method.

Most people learn the following result somewhere in the educational background for this class. Let x be a scalar, then

$$1 + x + x^2 + x^3 + \ldots = \sum_{k=0}^{\infty} x^k = \frac{1}{1 - x}$$

when $|x| < 1$. That is, if $x^k \to 0$, then the infinite sequences converges to the value $1/(1-x)$, which we are going to write as $(1 - x)^{-1}$.

It turns out that this same result holds for matrices as well, with a few additional conditions.

THEOREM 1 (The Neumann Series) *If that* $\boldsymbol{A}^k \to 0$, *then*

$$\sum_{k=0}^{\infty} \boldsymbol{A}^k = (\boldsymbol{I} - \boldsymbol{A})^{-1}$$

Proof Our proof proceeds just by showing that a partial infinite sum becomes a better approximation to the inverse. Let $\boldsymbol{S}_\ell = \sum_{k=0}^{\ell} \boldsymbol{A}^k$ and consider

$$\boldsymbol{S}_\ell(\boldsymbol{I} - \boldsymbol{A}) = \sum_{k=0}^{\ell} \boldsymbol{A}^k = (\boldsymbol{I} - \boldsymbol{A}) + (\boldsymbol{A} - \boldsymbol{A}^2) + (\boldsymbol{A}^2 - \boldsymbol{A}^3) + \ldots = \boldsymbol{I} - \boldsymbol{A}^{\ell+1}.$$

Consequently,

$$\lim_{\ell\to\infty} \boldsymbol{S}_\ell(\boldsymbol{I} - \boldsymbol{A}) = \lim_{\ell\to\infty} \boldsymbol{I} - \boldsymbol{A}^{\ell+1} = \boldsymbol{I}$$

and we have finished the proof as this is an explicit form for the inverse. ■

3 OVERVIEW

Over the next few classes, we are going to see a bunch of different persepctives on this same algorithm.

4 CHECKING A POSSIBLE SOLUTION

One great aspect about solving linear equations is that "guestimates" are easy to check.

Let $\boldsymbol{A}\mathbf{x} = \mathbf{b}$ be the system we are trying to solve and let $\mathbf{y}$ be a potential solution. Then the following quantities all deal with how good $\mathbf{y}$ is:

$$\begin{aligned} \text{error} &= \mathbf{y} - \mathbf{x} \\ \text{error} &= \|\mathbf{y} - \mathbf{x}\| \\ \text{relative error} &= \|\mathbf{y} - \mathbf{x}\| / \|\mathbf{x}\| \end{aligned}$$

$$\begin{aligned} \text{residual} &= \mathbf{b} - \boldsymbol{A}\mathbf{y} \\ \text{residual} &= \boldsymbol{A}\mathbf{y} - \mathbf{b} \\ \text{residual} &= \|\boldsymbol{A}\mathbf{y} - \mathbf{b}\| \\ \text{relative residual} &= \|\boldsymbol{A}\mathbf{y} - \mathbf{b}\| / \|\mathbf{b}\| \end{aligned}$$

Note that there are terms that may refer to multiple quantities. These are often used interchangably where the definition is clear from context.

The *error* measures are the most useful quantities, however, they are not easily computable as they require *knowing* the solution $\mathbf{x}$. However, we can bound the error in terms of the residual.

THEOREM 2 *Let* $\mathbf{y}$ *be any vector, then the* error $\mathbf{e} = \mathbf{y} - \mathbf{x}$ *and* residual $\mathbf{r} = \boldsymbol{A}\mathbf{x} - \mathbf{b}$ *are related as follows:*

$$\boldsymbol{A}\mathbf{e} = \mathbf{r}.$$

Proof By definition:

$$\boldsymbol{A}\mathbf{e} = \boldsymbol{A}\mathbf{y} - \boldsymbol{A}\mathbf{x} = \boldsymbol{A}\mathbf{y} - \mathbf{b}$$

because $\boldsymbol{A}\mathbf{x} = \mathbf{b}$. ■

This results in the following bound.

COROLLARY 3 *Using the notation from Theorem 2, let* $\|\cdot\|$ *be a sub-multiplicative norm.*[2] *Then*

$$\|\mathbf{e}\| \le \|\boldsymbol{A}^{-1}\| \, \|\mathbf{r}\|.$$

What this means is that if we want the error to be small, then we want the residual to be small. And the residual is easy to compute!

[2] Not all matrix norms are sub-multiplicative, see the discussion of Matrix and Vector Norms.

The proof follows from $\mathbf{e} = \boldsymbol{A}^{-1}\mathbf{r}$ and using $\|\boldsymbol{A}^{-1}\mathbf{r}\| \le \|\boldsymbol{A}\|^{-1}\|\mathbf{r}\|$ for a sub-multiplicative norm.

5 OUR FIRST METHOD REVISITED

On reflection, there is a better way to introduce the algorithm involving the Neumann series of a matrix. This has to do with how we might check the solution of a linear system of equation.

Given some initial guess at a solution $\mathbf{x}_0$, then we are going to compute the residual: $\mathbf{r}_0 = \mathbf{b} - \boldsymbol{A}\mathbf{x}_0$. If we are close to a solution, this will be small. So let's just correct by the amount we need:

$$\mathbf{x}_1 = \mathbf{x}_0 + \mathbf{r}_0.$$

Now, if we just repeatedly do this, then

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{r}_k = \mathbf{x}_k + \mathbf{b} - \boldsymbol{A}\mathbf{x}_k = (\boldsymbol{I} - \boldsymbol{A})\mathbf{x}_k + \mathbf{b}.$$

Quiz. Let $\mathbf{x}_0 = \mathbf{b}$. Show that $\mathbf{x}_k$ will converge to the solution $\mathbf{x}$ as $k \to \infty$. State conditions if necessary for this to converge.

Solution. By definition, $\mathbf{x}_1 = (\boldsymbol{I} - \boldsymbol{A})\mathbf{b} + \mathbf{b}$ and $\mathbf{x}_2 = (\boldsymbol{I} - \boldsymbol{A})^2\mathbf{b} + (\boldsymbol{I} - \boldsymbol{A})\mathbf{b} + \mathbf{b}$. By induction, we have: $\mathbf{x}_k = \sum_{\ell=0}^{k}(\boldsymbol{I} - \boldsymbol{A})^{\ell}\mathbf{b}$ and as $k \to \infty$, then $\mathbf{x}_k \to (\boldsymbol{I} - \boldsymbol{H})^{-1}\mathbf{b}$ where $\boldsymbol{H} = \boldsymbol{I} - \boldsymbol{A}$. But using that definition gives $(\boldsymbol{I} - \boldsymbol{H})^{-1} = \boldsymbol{A}^{-1}$. Hence this algorithm will converge if $\rho(\boldsymbol{I} - \boldsymbol{A}) < 1$ and it just corresponds to using the Neumann series itself.

5.1 AN EXAMPLE WITH OUR SIMPLE RANDOM WALK BETWEEN -4 AND 6.

Recall our linear system that modeled how long it took a random walk to exit through -4 and $+6$. This was the linear system of equations

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{\{-4\}} \\ x_{\{-3\}} \\ x_{\{-2\}} \\ x_{\{-1\}} \\ x_{\{0\}} \\ x_{\{+1\}} \\ x_{\{+2\}} \\ x_{\{+3\}} \\ x_{\{+4\}} \\ x_{\{+5\}} \\ x_{\{+6\}} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}.$$

When we apply this method here starting from $\mathbf{x}^{(1)} = 0$ (the all zeros vector), we get a sequence of iterates $\mathbf{x}^{(k)}$ along with residuals $\mathbf{r}^{(k)}$. After a few hundred iterations, these have largely converged.

Value of iterate vector $\mathbf{x}^{(k)}$ *when* $k =$																						
1	2	3	4	5	6	7	8	9	10	20	30	40	50	60	70	80	90	100	200	300	400	500
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	1.0	1.5	2.0	2.4	2.8	3.1	3.4	3.6	3.9	5.9	7.1	7.9	8.3	8.6	8.8	8.8	8.9	8.9	9.0	9.0	9.0	9.0
0.0	1.0	2.0	2.8	3.5	4.1	4.8	5.3	5.8	6.3	10.0	12.0	14.0	15.0	15.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0
0.0	1.0	2.0	3.0	3.9	4.8	5.5	6.3	7.0	7.7	13.0	16.0	18.0	19.0	20.0	20.0	21.0	21.0	21.0	21.0	21.0	21.0	21.0
0.0	1.0	2.0	3.0	4.0	4.9	5.9	6.7	7.6	8.4	15.0	18.0	21.0	22.0	23.0	23.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0
0.0	1.0	2.0	3.0	4.0	5.0	5.9	6.9	7.7	8.6	15.0	19.0	21.0	23.0	24.0	24.0	25.0	25.0	25.0	25.0	25.0	25.0	25.0
0.0	1.0	2.0	3.0	4.0	4.9	5.9	6.7	7.6	8.4	15.0	18.0	21.0	22.0	23.0	23.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0
0.0	1.0	2.0	3.0	3.9	4.8	5.5	6.3	7.0	7.7	13.0	16.0	18.0	19.0	20.0	20.0	21.0	21.0	21.0	21.0	21.0	21.0	21.0
0.0	1.0	2.0	2.8	3.5	4.1	4.8	5.3	5.8	6.3	10.0	12.0	14.0	15.0	15.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0	16.0
0.0	1.0	1.5	2.0	2.4	2.8	3.1	3.4	3.6	3.9	5.9	7.1	7.9	8.3	8.6	8.8	8.8	8.9	8.9	9.0	9.0	9.0	9.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Value of residual vector $\mathbf{r}^{(k)}$ *when* k =

1	2	3	4	5	6	7	8	9	10	20	30	40	50	60	70	80	90	100	200	300	400	500
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$1e^{0}$	$5e^{-1}$	$5e^{-1}$	$4e^{-1}$	$4e^{-1}$	$3e^{-1}$	$3e^{-1}$	$3e^{-1}$	$3e^{-1}$	$2e^{-1}$	$1e^{-1}$	$9e^{-2}$	$5e^{-2}$	$3e^{-2}$	$2e^{-2}$	$1e^{-2}$	$7e^{-3}$	$4e^{-3}$	$3e^{-3}$	$2e^{-5}$	$1e^{-7}$	$8e^{-10}$	$5e^{-12}$
$1e^{0}$	$1e^{0}$	$8e^{-1}$	$8e^{-1}$	$6e^{-1}$	$6e^{-1}$	$5e^{-1}$	$5e^{-1}$	$5e^{-1}$	$5e^{-1}$	$3e^{-1}$	$2e^{-1}$	$1e^{-1}$	$7e^{-2}$	$4e^{-2}$	$2e^{-2}$	$1e^{-2}$	$9e^{-3}$	$5e^{-3}$	$4e^{-5}$	$2e^{-7}$	$2e^{-9}$	$1e^{-11}$
$1e^{0}$	$1e^{0}$	$1e^{0}$	$9e^{-1}$	$9e^{-1}$	$8e^{-1}$	$8e^{-1}$	$7e^{-1}$	$7e^{-1}$	$6e^{-1}$	$4e^{-1}$	$2e^{-1}$	$1e^{-1}$	$9e^{-2}$	$5e^{-2}$	$3e^{-2}$	$2e^{-2}$	$1e^{-2}$	$7e^{-3}$	$5e^{-5}$	$3e^{-7}$	$2e^{-9}$	$1e^{-11}$
$1e^{0}$	$1e^{0}$	$1e^{0}$	$1e^{0}$	$9e^{-1}$	$9e^{-1}$	$9e^{-1}$	$9e^{-1}$	$8e^{-1}$	$8e^{-1}$	$5e^{-1}$	$3e^{-1}$	$2e^{-1}$	$1e^{-1}$	$6e^{-2}$	$4e^{-2}$	$2e^{-2}$	$1e^{-2}$	$9e^{-3}$	$6e^{-5}$	$4e^{-7}$	$2e^{-9}$	$2e^{-11}$
$1e^{0}$	$1e^{0}$	$1e^{0}$	$1e^{0}$	$1e^{0}$	$9e^{-1}$	$9e^{-1}$	$9e^{-1}$	$9e^{-1}$	$8e^{-1}$	$5e^{-1}$	$3e^{-1}$	$2e^{-1}$	$1e^{-1}$	$6e^{-2}$	$4e^{-2}$	$2e^{-2}$	$1e^{-2}$	$9e^{-3}$	$6e^{-5}$	$4e^{-7}$	$2e^{-9}$	$2e^{-11}$
$1e^{0}$	$1e^{0}$	$1e^{0}$	$1e^{0}$	$9e^{-1}$	$9e^{-1}$	$9e^{-1}$	$9e^{-1}$	$8e^{-1}$	$8e^{-1}$	$5e^{-1}$	$3e^{-1}$	$2e^{-1}$	$1e^{-1}$	$6e^{-2}$	$4e^{-2}$	$2e^{-2}$	$1e^{-2}$	$9e^{-3}$	$6e^{-5}$	$4e^{-7}$	$2e^{-9}$	$2e^{-11}$
$1e^{0}$	$1e^{0}$	$1e^{0}$	$9e^{-1}$	$9e^{-1}$	$8e^{-1}$	$8e^{-1}$	$7e^{-1}$	$7e^{-1}$	$6e^{-1}$	$4e^{-1}$	$2e^{-1}$	$1e^{-1}$	$9e^{-2}$	$5e^{-2}$	$3e^{-2}$	$2e^{-2}$	$1e^{-2}$	$7e^{-3}$	$5e^{-5}$	$3e^{-7}$	$2e^{-9}$	$1e^{-11}$
$1e^{0}$	$1e^{0}$	$8e^{-1}$	$8e^{-1}$	$6e^{-1}$	$6e^{-1}$	$5e^{-1}$	$5e^{-1}$	$5e^{-1}$	$5e^{-1}$	$3e^{-1}$	$2e^{-1}$	$1e^{-1}$	$7e^{-2}$	$4e^{-2}$	$2e^{-2}$	$1e^{-2}$	$9e^{-3}$	$5e^{-3}$	$4e^{-5}$	$2e^{-7}$	$2e^{-9}$	$1e^{-11}$
$1e^{0}$	$5e^{-1}$	$5e^{-1}$	$4e^{-1}$	$4e^{-1}$	$3e^{-1}$	$3e^{-1}$	$3e^{-1}$	$3e^{-1}$	$2e^{-1}$	$1e^{-1}$	$9e^{-2}$	$5e^{-2}$	$3e^{-2}$	$2e^{-2}$	$1e^{-2}$	$7e^{-3}$	$4e^{-3}$	$3e^{-3}$	$2e^{-5}$	$1e^{-7}$	$8e^{-10}$	$5e^{-12}$
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.2 AN EXAMPLE WITH OUR LEAST-SQUARES PROBLEM

Recall the quadratic fitting problem from “What is a matrix.”[3] This is reproduced at right.

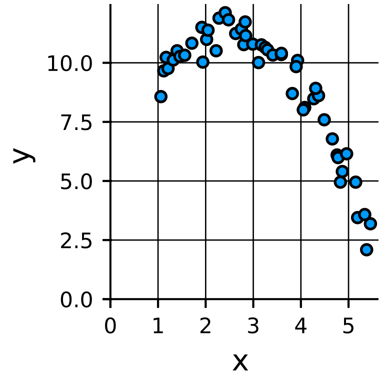

FIGURE 1 – We get a least squares problem to find a quadratic fit to this data $c_3x^2 + c_2x + c_1$. This gives a 3 × 3 linear system via the normal equations.

Consider the normal equations method of solving the least squares problem for the quadratic fit:

$$\begin{bmatrix} 50.0 & 162.63 & 616.468 \\ 162.63 & 616.468 & 2574.99 \\ 616.468 & 2574.99 & 11432.9 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 450.45 \\ 1329.6726 \\ 4549.258906 \end{bmatrix}$$

When we apply this method here starting from $\mathbf{c}^{(1)} = 0$ (the all zeros vector), we get a sequence of iterates $\mathbf{c}^{(k)}$ along with residuals $\mathbf{r}^{(k)}$. After less than one hundred iterations, this has produced `NaN` on the computer – a hallmark that the algorithm cannot converge on the problem.

Value of solution vector $\mathbf{c}^{(k)} = [c_1\ c_2\ c_3]$ *when* k =

1	2	3	4	5	6	7	8	9	10	20	30	40	50	60	70	80
0	$4e^{2}$	$-3e^{6}$	$4e^{10}$	$-4e^{14}$	$5e^{18}$	$-6e^{22}$	$8e^{26}$	$-9e^{30}$	$1e^{35}$	$7e^{75}$	$5e^{116}$	$3e^{157}$	$2e^{198}$	$1e^{239}$	$8e^{279}$	NaN
0	$1e^{3}$	$-1e^{7}$	$2e^{11}$	$-2e^{15}$	$2e^{19}$	$-3e^{23}$	$3e^{27}$	$-4e^{31}$	$5e^{35}$	$3e^{76}$	$2e^{117}$	$1e^{158}$	$8e^{198}$	$5e^{239}$	$3e^{280}$	NaN
0	$4e^{3}$	$-6e^{7}$	$7e^{11}$	$-8e^{15}$	$1e^{20}$	$-1e^{24}$	$1e^{28}$	$-2e^{32}$	$2e^{36}$	$1e^{77}$	$8e^{117}$	$5e^{158}$	$4e^{199}$	$2e^{240}$	$2e^{281}$	NaN

The residuals show the same behavior and quickly grow to ∞.

Value of residual vector $\mathbf{r}^{(k)}$ *when* k =

1	2	3	4	5	6	7	8	9	10	20	30	40	50	60	70	80
$5e^{2}$	$-3e^{6}$	$4e^{10}$	$-4e^{14}$	$5e^{18}$	$-6e^{22}$	$8e^{26}$	$-9e^{30}$	$1e^{35}$	$-1e^{39}$	$-9e^{79}$	$-6e^{120}$	$-4e^{161}$	$-2e^{202}$	$-1e^{243}$	$-1e^{284}$	NaN
$1e^{3}$	$-1e^{7}$	$2e^{11}$	$-2e^{15}$	$2e^{19}$	$-3e^{23}$	$3e^{27}$	$-4e^{31}$	$5e^{35}$	$-6e^{39}$	$-4e^{80}$	$-2e^{121}$	$-1e^{162}$	$-1e^{203}$	$-6e^{243}$	$-4e^{284}$	NaN
$5e^{3}$	$-6e^{7}$	$7e^{11}$	$-8e^{15}$	$1e^{20}$	$-1e^{24}$	$1e^{28}$	$-2e^{32}$	$2e^{36}$	$-2e^{40}$	$-2e^{81}$	$-1e^{122}$	$-7e^{162}$	$-4e^{203}$	$-3e^{244}$	$-2e^{285}$	NaN

In this case, we can compute $\rho(\boldsymbol{I} - \boldsymbol{A}) = 12047.41494842964$, so the spectral radius is much larger than 1 and we would not expect the method to work.

6 WHAT IF THIS ALGORITHM DOESN’T WORK?

Suppose that $\rho(\boldsymbol{I} - \boldsymbol{A}) > 1$. Are we out of luck with using this method? Not entirely! Consider that we can *transform* the linear system into an equivalent system of equations:

$$\boldsymbol{A}\mathbf{x} = \mathbf{b} \quad \Leftrightarrow \quad \alpha\boldsymbol{A}\mathbf{x} = \alpha\mathbf{b}$$

Then the iteration is:

$$\mathbf{x}_{k+1} = (\boldsymbol{I} - \alpha\boldsymbol{A})\mathbf{x}_k + \alpha\mathbf{b} = \sum_{\ell=0}^{k} (\boldsymbol{I} - \alpha\boldsymbol{A})^{\ell}(\alpha\mathbf{b}) \to (\alpha\boldsymbol{A})^{-1}\alpha\mathbf{b}.$$

[3] TODO, Add real reference.

This method is called the Richardson method for solving a linear system of equations. It is credited to Lewis Fry Richardson. Among other things, Richardson decided to spend his time in the trenches during World War I dreaming up better uses for the people fighting the war. His solution was to have them forecast the weather and he came up with this method.

When will this method converge?

Based on our analysis of the Neumann series, this will converge if $\rho(\boldsymbol{I} - \alpha \boldsymbol{A}) < 1$. Let λ be an eigenvalue of $\boldsymbol{A}$. This means we need that $|1 - \alpha\lambda| < 1$ for all eigenvalues of $\boldsymbol{A}$.

This means we can always make this algorithm work for a symmetric positive definite matrix $\boldsymbol{A}$ because all of the eigenvalues are positive.

7 ANOTHER DERIVATION OF THE SAME ALGORITHM

7.1 NOTES 1

Let's see *yet* another way to get at the same algorithm. This will involve some analysis of convex function.

Recall that a scalar quadratic function can be written:

$$f(x) = ax^2 + bx + c.$$

These look like bowls or lines (when $a = 0$).

Consider the problem

$$\underset{x}{\text{minimize}}\, ax^2 + bx + c$$

The solution is undefined is $a < 0$ (or just ∞). Otherwise, $x = -b/(2a)$ is the point that achieves the minimum. This can be found by looking for a point where the derivative is 0:

$$f'(x) = 2ax + b = 0 \Rightarrow x = -b/(2a).$$

A multivariate quadratic *looks* very similar.

7.2 NOTES 2

Gradient Descent for $\boldsymbol{A}\mathbf{x} = \mathbf{b}$.

It turns out that for any positive definite matrix $\boldsymbol{A}$, that we can view it as the solution of an optimization problem

$$\underset{\mathbf{x}}{\text{minimize}} \quad \tfrac{1}{2}\mathbf{x}^T \boldsymbol{A}\mathbf{x} - \mathbf{x}^T\mathbf{b}.$$

This is because if $\boldsymbol{A}$ is positive semi-definite, then this problem is convex with a unique global minimizer. A convex function is just one that always lies below any line connecting two points. Formally, this is $f(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) \le \alpha f(\mathbf{x}) + (1-\alpha) f(\mathbf{y})$. A global minimizer is any point $\mathbf{x}^*$ where $f(\mathbf{x}^*) \le f(\mathbf{x})$ for any other point $\mathbf{x}$. Note that if $f(\mathbf{x})$ is convex and if we have two global minimizers, then any point on the line connecting them must be a minimizer by the property of convexity.

There is a stronger result to prove here too.

THEOREM 4 *Let* $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \boldsymbol{A}\mathbf{x} - \mathbf{x}^T\mathbf{x}$. *Then* $f(\mathbf{x})$ *is convex if* $\boldsymbol{A}$ *is symmetric positive definite.*

Proof From the definition

$$\begin{aligned} f(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) &= (\alpha\mathbf{x} + (1-\alpha)\mathbf{y})^T \boldsymbol{A}(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) - (\alpha\mathbf{x} + (1-\alpha)\mathbf{y})^T\mathbf{b} \\ &= \alpha(\alpha\mathbf{x}^T\boldsymbol{A}\mathbf{x} - \mathbf{x}^T\mathbf{b} + (1-\alpha)((1-\alpha)\mathbf{y}^T\boldsymbol{A}\mathbf{y} - \mathbf{y}^T\mathbf{y}) + 2\alpha(1-\alpha)\mathbf{x}^T\boldsymbol{A}\mathbf{y} = \ldots \end{aligned}$$

■