02.02 systems of equations : iterative

why stop at one? part two.

gauss elimination is a finite sequence of $\mathcal{O}(n^3)$ operations that result in a solution. ie, it is a direct method. iterative methods solve systems of linear equations by refining an initial guess.

1 jacobi method

jacobi is fixed-point iteration for a system of equations and FPI rewrites equations then solves for the unknown.

example 19

apply jacobi to system $3u + v = 5, u + 2v = 5$ starting with $(u_0, v_0) = (0, 0)$. solve first equation for u first.

$$\begin{aligned} u &= \frac{5-v}{3} \\ v &= \frac{5-u}{2} \end{aligned}$$

$$\begin{aligned}
\begin{bmatrix} u_0 \\ v_0 \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} &= \begin{bmatrix} \frac{5-v_0}{3} \\ \frac{5-u_0}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-0}{3} \\ \frac{5-0}{2} \end{bmatrix} = \begin{bmatrix} \frac{5}{3} \\ \frac{5}{2} \end{bmatrix} \\
\begin{bmatrix} u_2 \\ v_2 \end{bmatrix} &= \begin{bmatrix} \frac{5-v_1}{3} \\ \frac{5-u_1}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-\frac{5}{2}}{3} \\ \frac{5-\frac{5}{3}}{2} \end{bmatrix} = \begin{bmatrix} \frac{5}{6} \\ \frac{5}{3} \end{bmatrix} \\
\begin{bmatrix} u_3 \\ v_3 \end{bmatrix} &= \begin{bmatrix} \frac{5-v_2}{3} \\ \frac{5-u_2}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-\frac{5}{3}}{3} \\ \frac{5-\frac{5}{6}}{2} \end{bmatrix} = \begin{bmatrix} \frac{10}{9} \\ \frac{25}{12} \end{bmatrix}.
\end{aligned}$$

further steps show convergence to solution, $[1, 2]^T$.

example 20

apply jacobi to system $u + 2v = 5, 3u + v = 5$ starting with $(u_0, v_0) = (0, 0)$. same equations, flip the order before solving for u.

$$\begin{aligned} u &= 5 - 2v \\ v &= 5 - 3u \end{aligned}$$

$$\begin{aligned}
\begin{bmatrix} u_0 \\ v_0 \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} &= \begin{bmatrix} 5-2v_0 \\ 5-3u_0 \end{bmatrix} = \begin{bmatrix} 5-2(0) \\ 5-3(0) \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \end{bmatrix} \\
\begin{bmatrix} u_2 \\ v_2 \end{bmatrix} &= \begin{bmatrix} 5-2v_1 \\ 5-3u_1 \end{bmatrix} = \begin{bmatrix} 5-2(5) \\ 5-3(5) \end{bmatrix} = \begin{bmatrix} -5 \\ -10 \end{bmatrix} \\
\begin{bmatrix} u_3 \\ v_3 \end{bmatrix} &= \begin{bmatrix} 5-2v_2 \\ 5-3u_2 \end{bmatrix} = \begin{bmatrix} 5-2(-10) \\ 5-3(-5) \end{bmatrix} = \begin{bmatrix} 25 \\ 20 \end{bmatrix}.
\end{aligned}$$

obviously, this one is not your bff. you need some rules.

definition 09. $n \times n$ matrix $A = (a_{ii})$ is **strictly diagonally dominant** if for each $1 \leq i \leq n$, $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$.

theorem 10. if $n \times n$ matrix A is strictly diagonally dominant, then (1) A is nonsingular and (2) for every vector b and every starting guess, the jacobi method applied to $Ax = b$ converges to the (unique) solution.

example 21

$$A = \begin{bmatrix} 3 & 1 & -1 \\ 2 & -5 & 2 \\ 1 & 6 & 8 \end{bmatrix} \text{ and } B = \begin{bmatrix} 3 & 2 & 6 \\ 1 & 8 & 1 \\ 9 & 2 & -2 \end{bmatrix}$$

where A is and B isnt.

usw

let D denote main diagonal of A, L denote lower triangle of A (below the diagonal) and U denote upper triangle of A (above the triangle) such that $A = L + D + U$. then

$$\begin{aligned} Ax &= b \\ (D + L + U)x &= b \\ Dx &= b - (L + U)x \\ x &= D^{-1}(b - (L + U)x). \end{aligned}$$

jacobi method

$$\begin{aligned} x_0 &= \text{initial vector} \\ x_{k+1} &= D^{-1}(b - (L + U)x_k) \quad k = 0, 1, 2, \ldots \end{aligned}$$

example 19, continued

$$\begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \end{bmatrix}$$
$$\Downarrow$$
$$x_k = \begin{bmatrix} u_k \\ v_k \end{bmatrix}$$
$$\begin{aligned} x_{k+1} &= D^{-1}(b-(L+U)x_k) \\ &= \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}\left(\begin{bmatrix} 5 \\ 5 \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} u_k \\ v_k \end{bmatrix}\right) \\ &= \begin{bmatrix} \frac{5-v_k}{3} \\ \frac{5-u_k}{2} \end{bmatrix}. \end{aligned}$$

2 gauss-seidel and SOR

with gauss-seidel, values are used as they are available.

example 19, continued

$$\begin{aligned} \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} u_1 \\ v_1 \end{bmatrix} &= \begin{bmatrix} \frac{5-v_0}{3} \\ \frac{5-u_1}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-0}{3} \\ \frac{5-\frac{5}{3}}{2} \end{bmatrix} = \begin{bmatrix} \frac{5}{3} \\ \frac{5}{3} \end{bmatrix} \\ \begin{bmatrix} u_2 \\ v_2 \end{bmatrix} &= \begin{bmatrix} \frac{5-v_1}{3} \\ \frac{5-u_2}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-\frac{5}{3}}{3} \\ \frac{5-\frac{10}{9}}{2} \end{bmatrix} = \begin{bmatrix} \frac{10}{9} \\ \frac{35}{18} \end{bmatrix} \\ \begin{bmatrix} u_3 \\ v_3 \end{bmatrix} &= \begin{bmatrix} \frac{5-v_2}{3} \\ \frac{5-u_3}{2} \end{bmatrix} = \begin{bmatrix} \frac{5-\frac{35}{18}}{3} \\ \frac{5-\frac{55}{54}}{2} \end{bmatrix} = \begin{bmatrix} \frac{55}{54} \\ \frac{215}{108} \end{bmatrix}. \end{aligned}$$

usw

gauss-seidel

$$\begin{aligned} x_0 &= \text{initial vector} \\ x_{k+1} &= D^{-1}(b - Ux_k - Lx_{k+1}) \quad k = 0, 1, 2, \ldots \end{aligned}$$

example 22

apply gauss-seidel to system

$$\begin{bmatrix} 3 & 1 & -1 \\ 2 & 4 & 1 \\ -1 & 2 & 5 \end{bmatrix}\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 1 \end{bmatrix}.$$

iteration:

$$\begin{aligned} u_{k+1} &= \frac{4 - v_k + w_k}{3} \\ v_{k+1} &= \frac{1 - 2u_{k+1} - w_k}{4} \\ w_{k+1} &= \frac{1 + u_{k+1} - 2v_{k+1}}{5} \end{aligned}$$

$$\begin{bmatrix} u_0 \\ v_0 \\ w_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} u_1 \\ v_1 \\ w_1 \end{bmatrix} = \begin{bmatrix} \frac{4-0-0}{3} = \frac{4}{3} \\ \frac{1-\frac{8}{3}-0}{4} = -\frac{5}{12} \\ \frac{1+\frac{4}{3}+\frac{5}{6}}{5} = \frac{19}{30} \end{bmatrix} \approx \begin{bmatrix} 1.3333 \\ -0.4167 \\ 0.6333 \end{bmatrix}$$

$$\begin{bmatrix} u_2 \\ v_2 \\ w_2 \end{bmatrix} = \begin{bmatrix} \frac{101}{60} \\ -\frac{3}{4} \\ \frac{251}{300} \end{bmatrix} \approx \begin{bmatrix} 1.6833 \\ -0.7500 \\ 0.8367 \end{bmatrix}$$

this system is strictly diagonally dominant and will converge to $[2, -1, 1]^T$.

further, if weights are applied to gauss-seidel to speed convergence, thats **successive over-relaxation (SOR)**.

let $\omega \in \mathbb{R}$ and define each component of new guess x_{k+1} as a weighted average of ω times gauss-seidel formula and $1-\omega$ times the current guess x_k. ω is the **relaxation parameter** and $\omega \in [0, 2]$. $\omega > 1$ is referred to as **over-relaxation**; and $\omega < 1$ is under-relaxation; and $\omega = 1$ is gauss-seidel, :).

example 23

apply SOR with $\omega = 1.25$ to example 22.

$$
\begin{aligned}
u_{k+1} &= (1-\omega)u_k + \omega\frac{4 - v_k + w_k}{3} \\
v_{k+1} &= (1-\omega)v_k + \omega\frac{1 - 2u_{k+1} - w_k}{4} \\
w_{k+1} &= (1-\omega)w_k + \omega\frac{1 + u_{k+1} - 2v_{k+1}}{5}
\end{aligned}
$$

$$
\begin{bmatrix} u_0 \\ v_0 \\ w_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}
$$

$$
\begin{bmatrix} u_1 \\ v_1 \\ w_1 \end{bmatrix} \approx \begin{bmatrix} 1.6667 \\ -0.7292 \\ 1.0312 \end{bmatrix}
$$

$$
\begin{bmatrix} u_2 \\ v_2 \\ w_2 \end{bmatrix} \approx \begin{bmatrix} 1.9835 \\ -1.0672 \\ 1.0216 \end{bmatrix}
$$

usw

and

$$
\begin{aligned}
(\omega L + \omega D + \omega U)x &= \omega b \\
(\omega L + D)x &= \omega b - \omega U x + (1-\omega)Dx \\
x &= (\omega L + D)^{-1}[(1-\omega)Dx - \omega U x] + \omega(D + \omega L)^{-1}b
\end{aligned}
$$

$$
\Downarrow
$$

$$
\text{no no no!} \quad x_{k+1} = (\omega L + D)^{-1}[(1-\omega)Dx_k - \omega U x_k] + \omega(D + \omega L)^{-1}b
$$

this logic bc $(\omega L + D)x$ was previous LHS and x_{k+1} relates to L; however, this distorts the impact of $\omega, (1-\omega)$ and does not reflect $x_{k+1} = (1-\omega)x_k + \omega x_{k,\text{ gauss seidel}}$. happy birthday!

successive over-relaxation (SOR)

$$
\begin{aligned}
x_0 &= \text{initial vector} \\
x_{k+1} &= \underbrace{(1-\omega)x_k}_{x_k\text{, weighted}} + \underbrace{\omega(D^{-1}(b - Ux_k - Lx_{k+1}))}_{x_{k+1}\text{ (gauss-seidel), weighted}} \quad k = 0, 1, 2, \ldots
\end{aligned}
$$

also

- relaxation parameter with gauss-seidel @themathguy. (its more complicated than above. but bc i was verifying previous algorithm - ie, preceding red text.)

example 24

compare jacobi, gauss-seidel, SOR with system

$$
\begin{bmatrix}
3 & -1 & 0 & 0 & 0 & \frac{1}{2} \\
-1 & 3 & -1 & 0 & \frac{1}{2} & 0 \\
0 & -1 & 3 & -1 & 0 & 0 \\
0 & 0 & -1 & 3 & -1 & 0 \\
0 & \frac{1}{2} & 0 & -1 & 3 & -1 \\
\frac{1}{2} & 0 & 0 & 0 & -1 & 3
\end{bmatrix}
\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix}
=
\begin{bmatrix} \frac{5}{2} \\ \frac{3}{2} \\ 1 \\ 1 \\ \frac{3}{2} \\ \frac{5}{2} \end{bmatrix}.
$$

solution is $x = [1, 1, 1, 1, 1, 1]^T$. x at six iterations below.

code

```
# lol, no. not this problem but its code is homework.

```

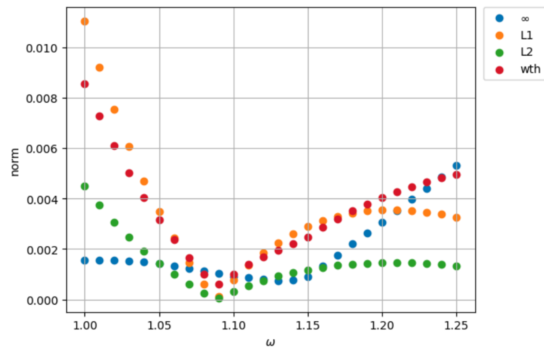
results

```
jac step 1  [0.83333333 0.5        0.33333333 0.33333333 0.5        0.83333333]
jac step 2  [0.86111111 0.80555556 0.61111111 0.61111111 0.80555556 0.86111111]
jac step 3  [0.95833333 0.85648148 0.80555556 0.80555556 0.85648148 0.95833333]
jac step 4  [0.95910494 0.94521605 0.88734568 0.88734568 0.94521605 0.95910494]
jac step 5  [0.98855453 0.95794753 0.94418724 0.94418724 0.95794753 0.98855453]
jac step 6  [0.98789009 0.98458933 0.96737826 0.96737826 0.98458933 0.98789009]

gas step 1  [0.83333333 0.77777778 0.59259259 0.5308642  0.5473251  0.87688615]
gas step 2  [0.9464449  0.92179165 0.81755195 0.78829235 0.90142756 0.97606837]
gas step 3  [0.97791915 0.94825244 0.9121816  0.93786972 0.97993729 0.99699257]
gas step 4  [0.98325205 0.96848833 0.96878602 0.98290777 0.99855206 1.00230868]
gas step 5  [0.98911133 0.98620711 0.98970496 0.99608567 1.0017636  1.00240264]
gas step 6  [0.99500193 0.99460836 0.99689801 0.99955387 1.00155078 1.00134994]

sor step 1  [0.91666667 0.88611111 0.69157407 0.62024383 0.61496903 0.97409976]
sor step 2  [0.97132245 0.95837329 0.87633554 0.85145396 0.98217102 1.00131028]
sor step 3  [0.98736441 0.95745463 0.9422996  0.9871605  1.0053555  1.00414918]
sor step 4  [0.98490291 0.97658028 0.99247499 1.00048846 1.00545854 1.00435435]
sor step 5  [0.99212418 0.99569427 0.99935284 1.00171532 1.00246908 1.00191379]
sor step 6  [0.99885795 0.99932187 1.00044502 1.00089697 1.00090803 1.00035094]
```

i	ω	Δ inf-norm	Δ L1-norm	Δ L2-norm
1	1	0.00155078	0.0110371	0.00449611
2	1.01	0.00156893	0.00920453	0.00374942
3	1.02	0.00156105	0.00754652	0.00307386
4	1.03	0.00153027	0.00605072	0.0024644
5	1.04	0.00147958	0.00470545	0.00191626
6	1.05	0.00141185	0.00349963	0.00142493
7	1.06	0.0013298	0.00242285	0.00098616
8	1.07	0.00123601	0.00146531	0.000595961
9	1.08	0.00113293	0.000617815	0.000250581
10	1.09	0.00102288	0.000128212	5.34801e-05
11	1.1	0.00090803	0.000780776	0.000319488
12	1.11	0.00086417	0.00134731	0.000550474
13	1.12	0.00081327	0.00183471	0.000749247
14	1.13	0.000746841	0.00224932	0.0009184
15	1.14	0.000770714	0.00259701	0.00106032
16	1.15	0.000910315	0.00288314	0.0011772
17	1.16	0.0013337	0.00311264	0.00127104
18	1.17	0.00176139	0.00328998	0.00134368
19	1.18	0.00219329	0.00341927	0.00139678
20	1.19	0.00262933	0.00350422	0.00143186
21	1.2	0.0030694	0.00354822	0.00145031
22	1.21	0.00351337	0.00355434	0.00145339
23	1.22	0.00396111	0.00352542	0.00144225
24	1.23	0.00441248	0.00346402	0.00141797
25	1.24	0.00486735	0.00337254	0.00138153
26	1.25	0.00532558	0.00325322	0.00133386



3 sparse matrices

why approximate with iterative methods vs solve directly with gaussian elimination methods? bc its operationally cheaper. its even more so if starting with a good guess.

iterative methods also avoid the bleed over that happens with sparse matrices that implement gauss elimination. (there are also structures for sparse matrices.)