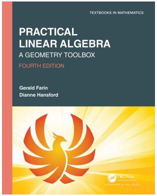
Practical Linear Algebra: A Geometry Toolbox
Fourth Edition

Chapter 13: Alternative System Solvers

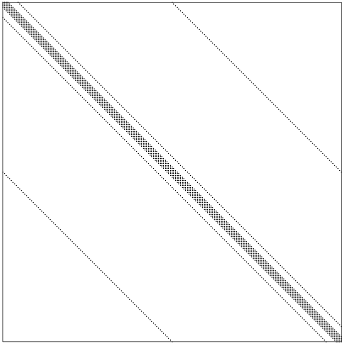
Gerald Farin & Dianne Hansford

A K Peters/CRC Press
www.farinhansford.com/books/pla



Outline

A sparse matrix: few nonzero entries (marked)



Gauss elimination methods
work well for
— Moderately-sized linear systems
(up to a few thousand equations)
— Systems absent of numerical problems

Ill-conditioned problems:
— More efficiently attacked using *the Householder method*
Huge systems (≤ 1 million equations)
— More successfully solved with *iterative methods*

The Householder Method

Problem: solve the linear system $A\mathbf{u} = \mathbf{b}$
$n \times n$ matrix A comprised of n column vectors – each with n elements

$$[\mathbf{a}_1 \dots \mathbf{a}_n]\mathbf{u} = \mathbf{b}$$

Gauss elimination: apply shears G_i to achieve upper triangular form

$$G_{n-1} \dots G_1 A\mathbf{u} = G_{n-1} \dots G_1 \mathbf{b}$$

Solve for $\mathbf{u}$ with back substitution
Each G_i transforms i^{th} column vector $G_{i-1} \dots G_1 \mathbf{a}_i$
to a vector with zeroes below the diagonal element $a_{i,i}$

Gauss elimination is not the most robust method
More numerically stable method: replace shears with *reflections*
This is the Householder method

The Householder Method

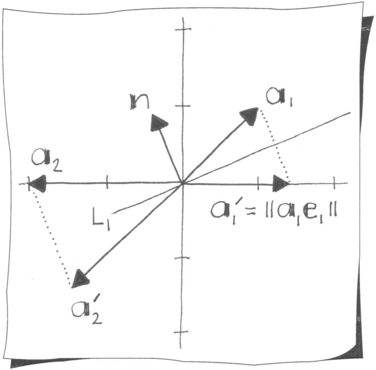
The Householder method:

Series of reflections H_i applied

$$H_{n-1} \dots H_1 A\mathbf{u} = H_{n-1} \dots H_1 \mathbf{b}$$

Each H_i transforms column vector $H_{i-1} \dots H_1 \mathbf{a}_i$
to a vector with zeroes below the diagonal element

H_i called a Householder transformation



Example: 2×2 matrix

$$A = \begin{bmatrix} 1 & -2 \\ 1 & 0 \end{bmatrix}$$

First transformation: $H_1 A$
Reflect $\mathbf{a}_1$ onto the $\mathbf{e}_1$ axis
to the vector $\mathbf{a}_1' = ||\mathbf{a}_1||\mathbf{e}_1$

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \rightarrow \quad \begin{bmatrix} \sqrt{2} \\ 0 \end{bmatrix}$$

Reflect about the line L_1
Construct a normal $\mathbf{n}_1$ to this line:

$$\mathbf{n}_1 = \frac{\mathbf{a}_1 - ||\mathbf{a}_1||\mathbf{e}_1}{||\mathbf{a}_1 - ||\mathbf{a}_1||\mathbf{e}_1||}$$

The Householder Method

Implicit equation of the line L_1

$$\mathbf{n}_1^{\mathrm{T}}\mathbf{x} = 0$$

$\mathbf{n}_1^{\mathrm{T}}\mathbf{a}_1$ is distance of the point $\mathbf{o} + \mathbf{a}_1$ to L_1
Reflection equivalent to moving twice $\mathbf{n}_1^{\mathrm{T}}\mathbf{a}_1$ in normal direction:

$$\mathbf{a}_1' = \mathbf{a}_1 - (2\mathbf{n}_1^{\mathrm{T}}\mathbf{a}_1)\mathbf{n}_1 \qquad (2\mathbf{n}_1^{\mathrm{T}}\mathbf{a}_1 \text{ is a scalar})$$

Reflection in matrix form:

$$\begin{aligned}\mathbf{a}_1' &= \mathbf{a}_1 - 2\mathbf{n}_1(\mathbf{n}_1^{\mathrm{T}}\mathbf{a}_1)\\ &= \left[I - 2\mathbf{n}_1\mathbf{n}_1^{\mathrm{T}}\right]\mathbf{a}_1 \qquad (2\mathbf{n}_1\mathbf{n}_1^{\mathrm{T}} \text{ is a dyadic matrix})\end{aligned}$$

Householder transformation:

$$H_1 = I - 2\mathbf{n}_1\mathbf{n}_1^{\mathrm{T}}$$

(Precisely the reflection constructed in Chapter 11)

Example:

$$A = \begin{bmatrix} 1 & -2 \\ 1 & 0 \end{bmatrix}$$

$$\mathbf{a}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \rightarrow \quad ||\mathbf{a}_1||\mathbf{e}_1 = \begin{bmatrix} \sqrt{2} \\ 0 \end{bmatrix}$$

Construct Householder matrix H_1:

$$\mathbf{n}_1 = \begin{bmatrix} -0.382 \\ 0.923 \end{bmatrix}$$

$$H_1 = I - 2 \begin{bmatrix} 0.146 & -0.353 \\ -0.353 & 0.853 \end{bmatrix} = \begin{bmatrix} 0.707 & 0.707 \\ 0.707 & -0.707 \end{bmatrix}$$

Transformed matrix is formed from the column vectors

$$H_1\mathbf{a}_1 = \begin{bmatrix} \sqrt{2} \\ 0 \end{bmatrix} \quad \text{and} \quad H_1\mathbf{a}_2 = \begin{bmatrix} -\sqrt{2} \\ -\sqrt{2} \end{bmatrix}$$

The Householder Method

2×2 example illustrates underlying geometry of a reflection matrix
General Householder transformation H_i construction more complicated

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ 0 & a_{2,2} & a_{2,3} & a_{2,4} \\ 0 & 0 & a_{3,3} & a_{3,4} \\ 0 & 0 & a_{4,3} & a_{4,4} \end{bmatrix}$$

Construct H_3 to zero the element $a_{4,3}$ and preserve upper triangular structure

$$\text{Let } \bar{\mathbf{a}}_3 = \begin{bmatrix} 0 \\ 0 \\ a_{3,3} \\ a_{4,3} \end{bmatrix} \qquad H_3\bar{\mathbf{a}}_3 = \gamma\mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ \gamma \\ 0 \end{bmatrix} \quad \text{where } \gamma = \pm\|\bar{\mathbf{a}}_3\|$$

— $H_3\mathbf{a}_3$ will only modify elements $a_{3,3}$ and $a_{4,3}$
— Length of $\mathbf{a}_3$ preserved

The Householder Method

Develop idea for $n \times n$ matrices:
Start with

$$\bar{\mathbf{a}}_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ a_{i,i} \\ \vdots \\ a_{n,i} \end{bmatrix}$$

Want Householder matrix H_i for transformation

$$H_i \bar{\mathbf{a}}_i = \gamma \mathbf{e}_i = \begin{bmatrix} 0 \\ \vdots \\ \gamma \\ \vdots \\ 0 \end{bmatrix} \qquad \text{where } \gamma = \pm \|\bar{\mathbf{a}}_i\|$$

Like the 2×2 example

$$H_i = I - 2\mathbf{n}_i\mathbf{n}_i^{\mathrm{T}} \qquad \text{where } \mathbf{n}_i = \frac{\bar{\mathbf{a}}_i - \gamma\mathbf{e}_i}{\|\cdot\|} \quad \text{and } \gamma = \pm\|\bar{\mathbf{a}}_i\|$$

$\pm\|\bar{\mathbf{a}}_i\|$ used to combat numerical problems:
— If $\bar{\mathbf{a}}_i$ nearly parallel to $\mathbf{e}_i$
then loss of *significant digits* will occur
from subtraction of nearly equal numbers
— Better to reflect onto direction of $\mathbf{e}_i$-axis representing largest reflection

The Householder Method

Householder matrix

$$H_i = I - 2\mathbf{n}_i\mathbf{n}_i^{\mathrm{T}}$$

Built from *symmetric* and *idempotent* matrix $N_i = \mathbf{n}_i\mathbf{n}_i^{\mathrm{T}}$

Properties of H_i:

- symmetric: $H_i = H_i^{\mathrm{T}}$ — since N_i is symmetric
- involutory: $H_i H_i = I \quad \Rightarrow H_i = H_i^{-1}$
- unitary (orthogonal): $H_i^{\mathrm{T}} H_i = I \quad \Rightarrow \|H_i\mathbf{v}\| = \|\mathbf{v}\|$

The Householder Method

Implementation of Householder transformations:
— Householder matrix not explicitly constructed
— Numerically and computationally more efficient algorithm implemented using knowledge of how H_i acts on column vectors

Variables to aid optimization:

$$\mathbf{v}_i = \bar{\mathbf{a}}_i - \gamma \mathbf{e}_i \quad \text{where} \quad \gamma = \begin{cases} -\text{sign } a_{i,i} \|\bar{\mathbf{a}}_i\| & \text{if } a_{i,i} \neq 0 \\ -\|\bar{\mathbf{a}}_i\| & \text{otherwise} \end{cases}$$

Leads to modification of $\mathbf{n}$

$$2\mathbf{n}\mathbf{n}^{\mathrm{T}} = \frac{\mathbf{v}\mathbf{v}^{\mathrm{T}}}{\frac{1}{2}\mathbf{v}^{\mathrm{T}}\mathbf{v}} = \frac{\mathbf{v}\mathbf{v}^{\mathrm{T}}}{\alpha} \qquad \alpha = \gamma^2 - a_{i,i}\gamma$$

When H_i applied to column vector $\mathbf{c}$

$$H_i\mathbf{c} = \left[I - \frac{\mathbf{v}\mathbf{v}^{\mathrm{T}}}{\alpha} \right] \mathbf{c} = \mathbf{c} - s\mathbf{v}$$

The Householder Method

In the Householder algorithm
As we work on the j^{th} column vector

$$\hat{\mathbf{a}}_k = \begin{bmatrix} a_{j,k} \\ \vdots \\ a_{n,k} \end{bmatrix}$$

only elements $j, \ldots, n$ of the k^{th} column vector $\mathbf{a}_k$ $(k \geq j)$ are involved in a calculation
$\Rightarrow$ application of H_j results in changes in the sub-block of A with $a_{j,j}$ at the upper-left corner

Vector $\mathbf{a}_j$ and $H_j\mathbf{a}_j$ coincide in the first $j - 1$ components

The Householder Method

Algorithm:

Input:

$n \times m$ matrix A, where $n \geq m$ and rank of A is m

n vector $\mathbf{b}$, augmented to A as the $(m+1)^{st}$ column

Output:

Upper triangular matrix HA written over A

$H\mathbf{b}$ written over $\mathbf{b}$ in the augmented $(m+1)^{st}$ column of A

$(H = H_{n-1} \ldots H_1)$

Algorithm continued:

If $n = m$ then $p = n - 1$; Else $p = m$ (p is last column to transform)
For $j = 1, 2, \ldots, p$
$a = \hat{\mathbf{a}}_j \cdot \hat{\mathbf{a}}_j$
$\gamma = -\text{sign}(a_{j,j})\sqrt{a}$
$\alpha = a - a_{j,j}\gamma$
Temporarily set $a_{j,j} = a_{j,j} - \gamma$
For $k = j + 1, \ldots, m + 1$
$s = \frac{1}{\alpha}(\hat{\mathbf{a}}_j \cdot \hat{\mathbf{a}}_k)$
$\hat{\mathbf{a}}_k = \hat{\mathbf{a}}_k - s\hat{\mathbf{a}}_j$
Set $\hat{\mathbf{a}}_j = \begin{bmatrix} \gamma & 0 & \ldots & 0 \end{bmatrix}^{\mathrm{T}}$

The Householder Method

Example:

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{u} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

$$j = 1: \quad \gamma = -\sqrt{2} \quad \alpha = 2 + \sqrt{2} \quad \text{(temporarily set) } \hat{\mathbf{a}}_1 = \begin{bmatrix} 1+\sqrt{2} \\ 1 \\ 0 \end{bmatrix}$$

$$k = 2: \quad s = \sqrt{2}/(2+\sqrt{2}) \quad \hat{\mathbf{a}}_2 = \begin{bmatrix} 0 \\ -\sqrt{2} \\ 0 \end{bmatrix}$$

$k = 3 : s = 0$ and $\hat{\mathbf{a}}_3$ remains unchanged

$$k = 4: \quad s = -\sqrt{2}/2 \quad \hat{\mathbf{a}}_4 = \begin{bmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \\ 0 \end{bmatrix}$$

The Householder Method

Set $\hat{\mathbf{a}}_1$ and reflection H_1 results in

$$\begin{bmatrix} -\sqrt{2} & 0 & 0 \\ 0 & -\sqrt{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{u} = \begin{bmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \\ 1 \end{bmatrix}$$

Not explicitly computed

$$\mathbf{n}_1 = \begin{bmatrix} 1+\sqrt{2} \\ 1 \\ 0 \end{bmatrix} /\|\cdot\| \qquad H_1 = \begin{bmatrix} -\sqrt{2}/2 & -\sqrt{2}/2 & 0 \\ -\sqrt{2}/2 & \sqrt{2}/2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Note:

— $\mathbf{a}_3$ was not affected

- It is in the plane about reflecting
- Result of the involutory property of the Householder matrix

— Length of each column vector not changed

- Result of the orthogonal property

Matrix is upper triangular

$\Rightarrow$ Use back substitution to find the solution vector

$$\mathbf{u} = \begin{bmatrix} -1/2 \\ -1/2 \\ 1 \end{bmatrix}$$

The Householder Method

Householder's algorithm is method of choice for ill-conditioned systems

Example: least squares solution for some data sets
— Forming $A^{\mathrm{T}}A$ is the problem (more on this later)

Revisit linear least squares approximation to time/temperature data problem: find line $x_2 = u_1 x_1 + u_2$

$$\begin{bmatrix} 0 & 1 \\ 10 & 1 \\ 20 & 1 \\ 30 & 1 \\ 40 & 1 \\ 50 & 1 \\ 60 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 30 \\ 25 \\ 40 \\ 40 \\ 30 \\ 5 \\ 25 \end{bmatrix}$$

The Householder Method

First Householder reflection ($j = 1$) linear system becomes

$$\begin{bmatrix} -95.39 & -2.20 \\ 0 & 0.66 \\ 0 & 0.33 \\ 0 & -0.0068 \\ 0 & -0.34 \\ 0 & -0.68 \\ 0 & -1.01 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -54.51 \\ 16.14 \\ 22.28 \\ 13.45 \\ -5.44 \\ -39.29 \\ -28.15 \end{bmatrix}$$

The Householder Method

Second Householder reflection ($j = 2$) linear system becomes

$$\begin{bmatrix} -95.39 & -2.20 \\ 0 & -1.47 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -54.51 \\ -51.10 \\ 11.91 \\ 13.64 \\ 5.36 \\ -17.91 \\ 3.81 \end{bmatrix}$$

Solve system with back substitution — starting with first non-zero row

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -0.23 \\ 34.82 \end{bmatrix}$$

Excluding numerical round-off — same solution found using normal equations

The Householder Method

The Householder method will appear in subsequent chapters

It will help with the potentially ill-conditioned matrix product $A^{\mathrm{T}}A$ that arises in the steps for computing the singular value decomposition
— See Chapter 16

It will help with the QR decomposition introduced as a matrix approach to the Gram-Schmidt method that avoids potential rounding error
— See Chapter 12

Vector Norms

Vector norm measures *magnitude* or *length* of a vector

Fundamental to many geometric operations in 3D
Fundamental in n-dimensions – even if vectors have no geometric meaning
— Example: iterative methods for solving linear systems (later in chapter)
Vector length key for monitoring improvements in the solution

"Usual" way to measure length:

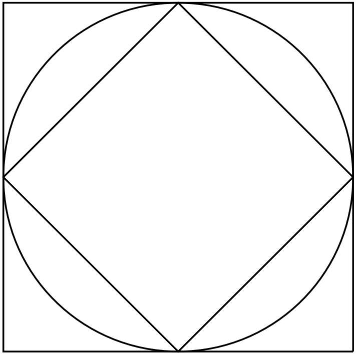
$$\|\mathbf{v}\|_2 = \sqrt{v_1^2 + \ldots + v_n^2}$$

— Non-negative scalar
— Referred to as the *Euclidean norm* because in $\mathbb{R}^3$ it is Euclidean length
— Subscript 2 is often omitted

Vector Norms

Outline of the *unit vectors*

2-norm $\Rightarrow$ circle, ∞-norm $\Rightarrow$ square, 1-norm $\Rightarrow$ diamond



1-norm (Manhattan or taxicab norm)

$$\|\mathbf{v}\|_1 = |v_1| + |v_2| + \ldots + |v_n|$$

∞-norm (max norm)

$$\|\mathbf{v}\|_\infty = \max_i |v_i|$$

Family of norms — p-norms

$$\|\mathbf{v}\|_p = (v_1^p + v_2^p + \ldots + v_n^p)^{1/p}$$

Vector Norms

Example:

$$\mathbf{v} = \begin{bmatrix} 1 \\ 0 \\ -2 \end{bmatrix} \qquad \|\mathbf{v}\|_1 = 3 \qquad \|\mathbf{v}\|_2 = \sqrt{5} \approx 2.24 \qquad \|\mathbf{v}\|_\infty = 2$$

Relationship between norms:

$$\|\mathbf{v}\|_1 \geq \|\mathbf{v}\|_2 \geq \|\mathbf{v}\|_\infty$$

Example application:

Given: 100K point pairs and a 2-norm tolerance t

Find: point pairs closer than t

— 2-norm takes more CPU clock cycles than other norms

— Max norm allows for trivial reject of some point pairs
If $\|\cdot\|_\infty \geq t$ then $\|\cdot\|_2 \geq t \quad \Rightarrow$ reject point pair

Vector Norms

Basic properties:

1. $\|\mathbf{v}\| \geq 0$
2. $\|\mathbf{v}\| = 0$ if and only if $\mathbf{v} = \mathbf{0}$
3. $\|c\mathbf{v}\| = |c|\|\mathbf{v}\|$ for $c \in \mathbb{R}$
4. $\|\mathbf{v} + \mathbf{w}\| \leq \|\mathbf{v}\| + \|\mathbf{w}\|$ *triangle inequality*

Vector Norms

Show: vector norm properties hold for the ∞-norm

Properties 1 and 2:
For each $\mathbf{v}$ in $\mathbb{R}^n$ by definition $\max_i |v_i| \geq 0$
$\max_i |v_i| = 0$ iff $v_i = 0$ for each $i = 1, \ldots n \Rightarrow \mathbf{v} = \mathbf{0}$

Property 3:

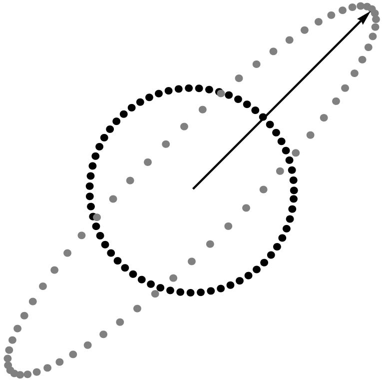
$$\|c\mathbf{v}\|_\infty = \max_i |cv_i| = |c| \max_i |v_i| = |c| \|\mathbf{v}\|_\infty$$

Property 4 (triangle inequality):

$$\begin{aligned} \|\mathbf{v} + \mathbf{w}\|_\infty &= \max_i |v_i + w_i| \\ &\leq \max_i \{|v_i| + |w_i|\} \\ &\leq \max_i |v_i| + \max_i |w_i| \\ &= \|\mathbf{v}\|_\infty + \|\mathbf{w}\|_\infty \end{aligned}$$

Matrix Norms

Magnitude of a matrix?



Insight from a 2×2 matrix:
— Maps unit circle to *action ellipse*

Consider $A^{\mathrm{T}}A$
— *symmetric* and *positive definite*
$\Rightarrow$ real and positive eigenvalues λ'_i
Singular values of A:

$$\sigma_i = \sqrt{\lambda'_i}$$

σ_1: length of semi-major axis
σ_2: length of semi-minor axis

If A symmetric and positive definite
$\Rightarrow \sigma_i = \lambda_i$

How much does A distort the unit circle?

Measured by its 2-norm $\|A\|_2$

If we find the largest $\|A\mathbf{v}_i\|_2$
then have an indication of how much A distorts

With k unit vectors $\mathbf{v}_i$ compute

$$\|A\|_2 \approx \max_i \|A\mathbf{v}_i\|_2$$

Increase k: $\Rightarrow$ better and better approximation to $\|A\|_2$

$$\|A\|_2 = \max_{\|\mathbf{v}\|_2=1} \|A\mathbf{v}\|_2$$

Matrix Norms

Matrix norms not restricted to 2×2 matrices

For $n \times n$

$$\|A\|_2 = \sigma_1 \qquad (A\text{'s largest singular value})$$

Inverse matrix A^{-1} "undoes" the action of A

Let singular values of A^{-1} be called $\hat{\sigma}_i$

$$\hat{\sigma}_1 = \frac{1}{\sigma_n}, \quad \dots, \quad \hat{\sigma}_n = \frac{1}{\sigma_1}$$

$$\|A^{-1}\|_2 = \frac{1}{\sigma_n}$$

Singular values typically computed using a method called *Singular Value Decomposition* or *SVD*

— Focus of Chapter 16

Matrix Norms

Analogous to vector norms — there are several matrix norms

$$\begin{aligned} \|A\|_1 &: \text{ maximum absolute column sum} \\ \|A\|_\infty &: \text{ maximum absolute row sum} \end{aligned}$$

Careful: notation for matrix and vector norms identical

Frobenius norm: gives the total distortion caused by A

$$\|A\|_F = \sqrt{\sigma_1^2 + \ldots + \sigma_n^2}$$

Euclidean norm:

$$\|A\|_E = \sqrt{a_{1,1}^2 + a_{1,2}^2 + \ldots + a_{n,n}^2}$$

Not obvious: $\|A\|_F = \|A\|_E$

Matrix Norms

Example:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 5 \\ 5 & 6 & -7 \end{bmatrix}$$

Singular values: 10.5, 7.97, 0.334

$$\begin{aligned} \|A\|_2 &= \max\{10.5, 7.97, 0.334\} = 10.5 \\ \|A\|_1 &= \max\{9, 12, 15\} = 15 \\ \|A\|_\infty &= \max\{6, 12, 18\} = 18 \\ \|A\|_F &= \sqrt{1^2 + 3^2 + \ldots (-7)^2} = \sqrt{10.5^2 + 7.97^2 + 0.334} = 13.2 \end{aligned}$$

Matrix Norms

Matrix norms are real-valued functions
of the linear space defined over all $n \times n$ matrices

Matrix norms satisfy conditions very similar to the vector norm conditions

$$\begin{aligned} &\|A\| > 0 \quad \text{for} \quad A \neq Z \\ &\|A\| = 0 \quad \text{for} \quad A = Z \\ &\|cA\| = |c|\|A| \qquad c \in \mathbb{R} \\ &\|A + B\| \leq \|A\| + \|B\| \\ &\|AB\| \leq \|A\|\|B\| \end{aligned}$$

Z being the zero matrix

How to choose a matrix norm?
Computational expense and properties of the norm are the deciders
Example: the Frobenius and 2-norms are invariant with respect to orthogonal transformations

The Condition Number

How sensitive is the solution to $A\mathbf{u} = \mathbf{b}$ is to changes in A and $\mathbf{b}$?

Action ellipse/ellipsoid describes geometry of map
— Semi-major length $= \sigma_1$ (singular value of A)
Semi-minor axis length $= \sigma_n$
— 2×2: if σ_1 very large and σ_2 very small $\Rightarrow$ elongated ellipse

Condition number

$$\kappa(A) = \|A\|_2 \|A^{-1}\|_2 = \sigma_1/\sigma_n$$

Figure: symmetric, positive definite $A = \begin{bmatrix} 1.5 & 0 \\ 0 & 0.05 \end{bmatrix}$

$A^{\mathrm{T}}A$ is symmetric and positive definite $\Rightarrow \kappa(A) \geq 1$
— Well-conditioned matrix: $\kappa(A)$ close to one
No distortion: $\kappa(A) = 1$ Example: the identity matrix
— Ill-conditioned matrix: $\kappa(A)$ "large"

Example: Rotation matrix — no distortion

$$A = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}$$

$$A^{\mathrm{T}}A = I \quad \Rightarrow \quad \sigma_1 = \sigma_2 = 1 \quad \Rightarrow \quad \kappa(A) = 1$$

Example: Non-uniform scaling — severely distorting

$$A = \begin{bmatrix} 100 & 0 \\ 0 & 0.01 \end{bmatrix}$$

$$\sigma_1 = 100 \text{ and } \sigma_2 = 0.01 \quad \Rightarrow \quad \kappa(A) = 100/0.01 = 10,000$$

The Condition Number

Back to solving $A\mathbf{u} = \mathbf{b}$

Avoid creating a poorly designed linear system with ill-conditioned A

— Definition of large $\kappa(A)$ subjective and problem-specific

— Guideline: $\kappa(A) \approx 10^k$ can result in a loss of k digits of accuracy

— If $\kappa(A)$ large then solution cannot be depended upon (irrespective of round-off)

Ill-conditioned matrix
$\Rightarrow$ solution is numerically very sensitive to small changes in A or $\mathbf{b}$

Well-conditioned matrix
$\Rightarrow$ can confidently calculate the inverse

The Condition Number

Condition number is a better measure of singularity than the determinant

— Scale and size n invariant measure: $\kappa(sA) = \kappa(A)$

Example: Let $n = 100$

Form the identity matrix I and $J = 0.1I$

$$\det I = 1 \qquad \kappa(I) = 1 \qquad \det J = 10^{-100} \qquad \kappa(J) = 1$$

$\det J$ small $\Rightarrow$ problem with this matrix

But scale of J poses no problem in solving a linear system

The Condition Number

Overdetermined linear systems $A\mathbf{u} = \mathbf{b}$

Least squares approximation:

— Solved the system $A^{\mathrm{T}}A\mathbf{u} = A^{\mathrm{T}}\mathbf{b}$

— Condition number $\kappa(A^{\mathrm{T}}A) = \kappa(A)^2$

— If A has a high condition number $\Rightarrow$ ill-posed problem

— The Householder method is preferred

Vector Sequences

Sequences of real numbers:

$$1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots \qquad \text{limit } 0$$

$$1, 2, 4, 8, \ldots \qquad \text{no limit}$$

Limit: A sequence of real numbers a_i has a limit a
if beyond some index i all a_i differ from the limit by an arbitrarily small ϵ

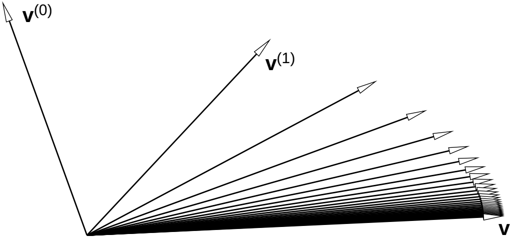
Vector sequences in $\mathbb{R}^n$: $\mathbf{v}^{(0)}, \mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots,$
A vector sequence has a limit if each component has a limit

Example: vector sequences

$$\mathbf{v}^{(i)} = \begin{bmatrix} 1/i \\ 1/i^2 \\ 1/i^3 \end{bmatrix} \quad \text{limit } \mathbf{v} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \mathbf{v}^{(i)} = \begin{bmatrix} i \\ 1/i^2 \\ 1/i^3 \end{bmatrix} \quad \text{no limit}$$

Vector sequence *converges* to $\mathbf{v}$ with respect to a norm
if for any tolerance $\epsilon > 0$ there exists an integer m such that

$$\|\mathbf{v}^{(i)} - \mathbf{v}\| < \epsilon \qquad \text{for all} \quad i > m$$



Vector Sequences

If sequence converges with respect to one norm
it will converge with respect to all norms

In practical applications: limit vector $\mathbf{v}$ not known

For some problems: know limit exists but do not know it *a priori*
$\Rightarrow$ Modify the theoretical convergence measure to
distance between iterations:

$$\|\mathbf{v}^{(i)} - \mathbf{v}^{(i+1)}\| < \epsilon$$

Iterative System Solvers: Gauss-Jacobi and Gauss-Seidel

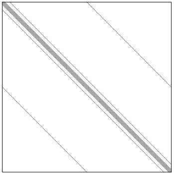
- Some applications generate linear systems with many thousands of equations
- Example: *Finite Element Methods* (*FEM*) and fluid flow problems
- Gauss elimination too slow
- Typically huge linear systems have a *sparse* coefficient matrix

— Only a few nonzero entries per row

— Example: $100{,}000 \times 100{,}000$ system

$\Rightarrow$ 10,000,000,000 matrix elements and 1,000,000 nonzero entries

- Solution to large sparse systems typically obtained by *iterative methods*



An iterative method starts from a *guess* for the solution

Then refines it until it *is* the solution

Gauss-Jacobi iteration:

Example: $$\begin{bmatrix} 4 & 1 & 0 \\ 2 & 5 & 1 \\ -1 & 2 & 4 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix}$$

Guess: $$\mathbf{u}^{(1)} = \begin{bmatrix} u_1^{(1)} \\ u_2^{(1)} \\ u_3^{(1)} \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad A\mathbf{u}^{(1)} \neq \mathbf{b}$$

Better guess: use $u_i^{(1)}$ and solve i^{th} equation for a new $u_i^{(2)}$

$$\begin{aligned} 4u_1^{(2)} + 1 &= 1 \\ 2 + 5u_2^{(2)} + 1 &= 0 \\ -1 + 2 + 4u_3^{(2)} &= 3 \end{aligned} \qquad \Rightarrow \quad \mathbf{u}^{(2)} = \begin{bmatrix} 0 \\ -0.6 \\ 0.5 \end{bmatrix}$$

Iterative System Solvers: Gauss-Jacobi and Gauss-Seidel

Next iteration:

$$
\begin{aligned}
4u_1^{(3)} - 0.6 &= 1 \\
5u_2^{(3)} + 0.5 &= 0 \\
-1.2 + 4u_3^{(3)} &= 3
\end{aligned}
\qquad \Rightarrow \qquad \mathbf{u}^{(3)} = \begin{bmatrix} 0.4 \\ -0.1 \\ 1.05 \end{bmatrix}
$$

After a few more iterations — close enough to the true solution

$$
\mathbf{u} = \begin{bmatrix} 0.333 \\ -0.333 \\ 1.0 \end{bmatrix}
$$

Iterative System Solvers: Gauss-Jacobi and Gauss-Seidel

Gauss-Jacobi iteration for $A\mathbf{u} = \mathbf{b}$ with n equations and n unknowns

D: diagonal matrix with A's diagonal elements
R: A with all diagonal elements set to zero

$$A = D + R \quad \Rightarrow \quad D\mathbf{u} + R\mathbf{u} = \mathbf{b}$$

$$\mathbf{u} = D^{-1}[\mathbf{b} - R\mathbf{u}]$$

$$\mathbf{u}^{(k+1)} = D^{-1}\left[\mathbf{b} - R\mathbf{u}^{(k)}\right]$$

Example:

$$A = \begin{bmatrix} 4 & 1 & 0 \\ 2 & 5 & 1 \\ -1 & 2 & 4 \end{bmatrix} \quad R = \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 1 \\ -1 & 2 & 0 \end{bmatrix} \quad D^{-1} = \begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.25 \end{bmatrix}$$

$$\mathbf{u}^{(2)} = \begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.25 \end{bmatrix} \left(\begin{bmatrix} 1 \\ 0 \\ 3 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 1 \\ -1 & 2 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ -0.6 \\ 0.5 \end{bmatrix}$$

Iterative System Solvers: Gauss-Jacobi and Gauss-Seidel

Will the Gauss-Jacobi method succeed?
$\Rightarrow$ Will sequence of vectors $\mathbf{u}^{(k)}$ converge?

Answer: sometimes yes and sometimes no

It will *always* succeed if A is diagonally dominant
— for every row:
$|\text{diagonal element}| > \sum |\text{remaining elements}|$
— Result of many practical problems — e.g., FEM

How to determine if convergence is taking place?
Length of the residual vector

$$\|A\mathbf{u}^{(k)} - \mathbf{b}\| < \text{tolerance}$$

Iterative System Solvers: Gauss-Jacobi and Gauss-Seidel

Gauss-Seidel iteration

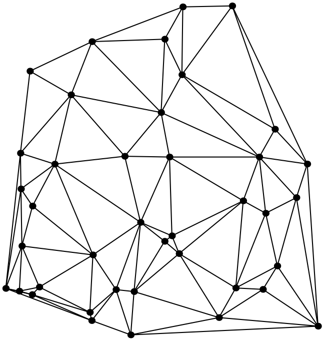
Modification of Gauss-Jacobi
— In computation of $\mathbf{u}^{(k+1)}$: $u_2^{(k+1)}$ computed using $u_1^{(k)}, u_3^{(k)}, \ldots, u_n^{(k)}$
— Instead: could use newly computed $u_1^{(k+1)}$
$\Rightarrow$ Idea of Gauss-Seidel iteration

Summary:
Gauss-Jacobi updates the new guess vector once all elements computed
Gauss-Seidel updates as soon as a new element is computed

Typically Gauss-Seidel converges faster than Gauss-Jacobi

Application: Mesh Smoothing

Triangulation smoothing application
Left: "rough" triangulation
Right: smoother triangulation after application of Laplacian smoothing
— triangles are closer to being equilateral
— achieve desired shape properties
via partial differential equations/minimize an energy functional

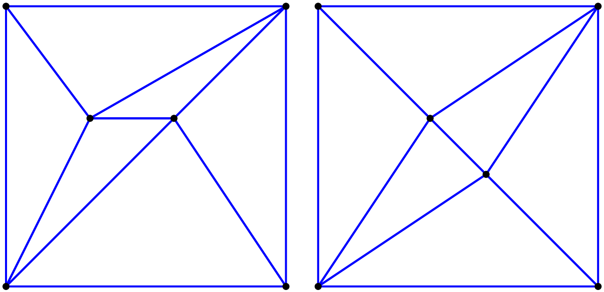


Application: Mesh Smoothing

Simple example of Laplacian smoothing
Boundary points fixed
Move interior points — average of their neighbors

$$\mathbf{p}_5 = 0.25(\mathbf{p}_1 + \mathbf{p}_3 + \mathbf{p}_4 + \mathbf{p}_6) \qquad \mathbf{p}_6 = 0.25(\mathbf{p}_1 + \mathbf{p}_2 + \mathbf{p}_3 + \mathbf{p}_5)$$

$$\begin{bmatrix} 1 & -0.25 \\ -0.25 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_5 \\ \mathbf{p}_6 \end{bmatrix} = \begin{bmatrix} 0.25(\mathbf{p}_1 + \mathbf{p}_3 + \mathbf{p}_4) \\ 0.25(\mathbf{p}_1 + \mathbf{p}_2 + \mathbf{p}_3) \end{bmatrix}$$



WYSK

- reflection matrix
- Householder method
- overdetermined system
- symmetric matrix
- involutory matrix
- orthogonal matrix
- unitary matrix
- vector norm
- vector norm properties
- Euclidean norm
- L^2 norm
- Manhattan norm
- matrix norm
- matrix norm properties
- Frobenius norm
- action ellipse axes
- singular values
- condition number
- well-conditioned matrix
- ill-conditioned matrix
- vector sequence
- convergence
- iterative method
- sparse matrix
- Gauss-Jacobi method
- Gauss-Seidel method
- residual vector