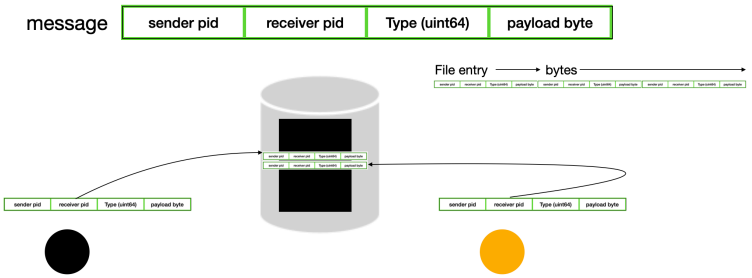
lecture 14

- *mmap (improving the shared file approach - segue to shared memory) for > 2 processes*
  - *issues and fixes*

- *linear algebra flash session*
  - *machine learning connections*
  - *span, linear independence, row & column & null space, rank and nullity*
  - *linear systems - Gauss elimination method*
    - *elementary matrices, LU decomposition*
  - *orthogonal projection*
  - *best fit (least squares) - the normal equations*

Kenneth J. Roche
University of Washington
AMATH 483 / 583 High Performance Scientific Computing
Spring Quarter 2025

- *(Continue from Lecture 13) ipc (using mmap in a file to pass messages as shared communication) for > 2 processes*




```c
#include <sys/mman.h>

int mlock(const void *addr, size_t len);
```

Parameters:

- `addr` : Starting address of the memory region
- `len` : Length (in bytes) of memory to lock

Return value:

- Returns `0` on success
- Returns `-1` on failure and sets `errno`

Purpose:

- Ensures that the memory remains in RAM (not swapped), which is useful for:
  - Low-latency messaging
  - Realtime or cryptographic applications
  - Avoiding page faults

```c
// Write message to file using mmap + mlock
void send_message(pid_t sender, pid_t receiver, const std::string &message)
{
    int fd = open(COMM_FILENAME.c_str(), O_RDWR | O_CREAT, 0666);
    if (fd == -1)
    {
        perror("open");
        return;
    }

    // Extend file size
    struct stat st;
    fstat(fd, &st);
    size_t old_size = st.st_size;
    size_t new_size = old_size + sizeof(MessageHeader) + message.size();
    ftruncate(fd, new_size);

    void *addr = mmap(nullptr, new_size, PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0);
    if (addr == MAP_FAILED)
    {
        perror("mmap");
        close(fd);
        return;
    }

    // Lock mapped memory
    mlock(addr, new_size);

    char *write_ptr = static_cast<char *>(addr) + old_size;
    MessageHeader header{sender, receiver, message.size()};
    std::memcpy(write_ptr, &header, sizeof(header));
    std::memcpy(write_ptr + sizeof(header), message.data(), message.size());

    munmap(addr, new_size);
    close(fd);
}
```

Kenneth J. Roche
University of Washington

- *(Continue from Lecture 13) ipc (using mmap in a file to pass messages as shared communication) for > 2 processes*

```cpp
int main(int argc, char *argv[])
{
    if (argc != 2)
    {
        std::cerr << "Usage: " << argv[0] << " <num_children>\n";
        return 1;
    }

    int num_children = std::stoi(argv[1]);
    pid_t parent_pid = getpid();

    // Clear file
    std::ofstream ofs(COMM_FILENAME, std::ios::binary | std::ios::trunc);
    ofs.close();

    std::vector<pid_t> child_pids;

    for (int i = 0; i < num_children; ++i)
    {
        pid_t pid = fork();
        if (pid == 0)
        {
            // Child process
            pid_t my_pid = getpid();

            // Let parent send first
            sleep(1);

            send_message(my_pid, parent_pid, "Hello from child " + std::to_string(i));

            auto inbox = receive_messages(my_pid);
            for (const auto &[sender, msg] : inbox)
            {
                std::cout << "Child " << i << " received from " << sender << ": " << msg << "\n";
            }

            _exit(0);
        }
        else
        {
            child_pids.push_back(pid);
        }
    }

    // Parent sends messages to each child
    for (int i = 0; i < num_children; ++i)
    {
        send_message(parent_pid, child_pids[i], "Hello to child " + std::to_string(i));
    }

    // Wait for all children
    for (pid_t pid : child_pids)
    {
        waitpid(pid, nullptr, 0);
    }

    std::cout << "All child processes finished.\n";

    auto inbox = receive_messages(parent_pid);
    for (const auto &[sender, msg] : inbox)
    {
        std::cout << "Parent received from " << sender << ": " << msg << "\n";
    }

    unlink(COMM_FILENAME.c_str());
    return 0;
}
```

Kenneth J. Roche
University of Washington
AMATH 483 / 583 High Performance Scientific Computing
Spring Quarter 2025

- *(Continue from Lecture 13) ipc (using mmap in a file to pass messages as shared communication) for > 2 processes*

```
bash-3.2$ g++ -std=c++17 kr-mmap-messaging-np.cpp -o xmmap-messaging-np
bash-3.2$ ./xmmap-messaging-np 5
Child 4 received from 4763: Hello to child 4
Child 1 received from 4763: Hello to child 1
Child 2 received from 4763: Hello to child 2
Child 0 received from 4763: Hello to child 0
Child 3 received from 4763: Hello to child 3
All child processes finished.
Parent received from 4768: Hello from child 3
Parent received from 4765: Hello from child 0
Parent received from 4769: Hello from child 4
bash-3.2$ ./xmmap-messaging-np 5
Child 0 received from 4771: Hello to child 0
Child 4 received from 4771: Hello to child 4
Child 1 received from 4771: Hello to child 1
Child 2 received from 4771: Hello to child 2
Child 3 received from 4771: Hello to child 3
All child processes finished.
Parent received from 4772: Hello from child 0
Parent received from 4773: Hello from child 1
Parent received from 4774: Hello from child 2
Parent received from 4776: Hello from child 4
Parent received from 4775: Hello from child 3
bash-3.2$ ./xmmap-messaging-np 5
Child 0 received from 4777: Hello to child 0
Child 1 received from 4777: Hello to child 1
Child 2 received from 4777: Hello to child 2
Child 4 received from 4777: Hello to child 4
Child 3 received from 4777: Hello to child 3
All child processes finished.
Parent received from 4778: Hello from child 0
Parent received from 4782: Hello from child 4
Parent received from 4781: Hello from child 3
bash-3.2$
```

Execution behavior is unpredictable:

two (or more) children might:

- compute the current file size at the same time
- `ftruncate()` and `mmap()` with the same `old_size`
- and write to **overlapping regions**, clobbering each other's messages

Kenneth J. Roche
University of Washington

- *(Continue from Lecture 13) ipc (using mmap in a file to pass messages as shared communication) for > 2 processes : lock / serialize writing to common resource*

```
void send_message(pid_t sender, pid_t receiver, const std::string &message)
{ //kr: fix the unprotected access to file mmap for n > 1 senders
    int fd = open(COMM_FILENAME.c_str(), O_RDWR | O_CREAT, 0666);
    if (fd == -1)
    {
        perror("open");
        return;
    }

    // Apply exclusive lock using fcntl()
    struct flock lock;
    lock.l_type = F_WRLCK;
    lock.l_whence = SEEK_SET;
    lock.l_start = 0;
    lock.l_len = 0; // Lock the whole file
    lock.l_pid = getpid();

    if (fcntl(fd, F_SETLKW, &lock) == -1)
    {
        perror("fcntl lock");
        close(fd);
        return;
    }

    // Extend file size and mmap
    struct stat st;
    fstat(fd, &st);
    size_t old_size = st.st_size;
    size_t new_size = old_size + sizeof(MessageHeader) + message.size();
    ftruncate(fd, new_size);

    void *addr = mmap(nullptr, new_size, PROT_READ | PROT_WRITE, MAP_SHARED, fd, 0);
    if (addr == MAP_FAILED)
    {
        perror("mmap");
        close(fd);
        return;
    }

    mlock(addr, new_size); // optional: pin pages in memory

    char *write_ptr = static_cast<char *>(addr) + old_size;
    MessageHeader header{sender, receiver, message.size()};
    std::memcpy(write_ptr, &header, sizeof(header));
    std::memcpy(write_ptr + sizeof(header), message.data(), message.size());

    munmap(addr, new_size);

    // Release file lock
    lock.l_type = F_UNLCK;
    fcntl(fd, F_SETLK, &lock);

    close(fd);
}
```

Kenneth J. Roche
University of Washington

- *(Continue from Lecture 13) ipc (using mmap in a file to pass messages as shared communication) for > 2 processes : lock / serialize writing to common resource*

```
void send_message(pid_t sender, pid_t receive
{ //kr: fix the unprotected access to file mm
    int fd = open(COMM_FILENAME.c_str(), O_RD
    if (fd == -1)
    {
        perror("open");
        return;
    }

    // Apply exclusive lock using fcntl()
    struct flock lock;
    lock.l_type = F_WRLCK;
    lock.l_whence = SEEK_SET;
    lock.l_start = 0;
    lock.l_len = 0; // Lock the whole file
    lock.l_pid = getpid();

    if (fcntl(fd, F_SETLKW, &lock) == -1)
    {
        perror("fcntl lock");
        close(fd);
        return;
    }

    // Extend file size and mmap
    struct stat st;
    fstat(fd, &st);
    size_t old_size = st.st_size;
    size_t new_size = old_size + sizeof(MessageHeader) + message.size();
    ftruncate(fd, new_size);
```

```
struct flock {
    short l_type;    // Type of lock: F_RDLCK, F_WRLCK, or F_UNLCK
    short l_whence;  // How to interpret l_start: SEEK_SET, SEEK_CUR, SEEK_END
    off_t l_start;   // Starting offset for the lock
    off_t l_len;     // Number of bytes to lock (0 means "to EOF")
    pid_t l_pid;     // PID of the process holding the lock (set by kernel on GETLK)
};
```

```
struct flock lock;
lock.l_type = F_WRLCK;        // Exclusive lock
lock.l_whence = SEEK_SET;     // Relative to beginning of file
lock.l_start = 0;             // Offset from start
lock.l_len = 0;               // Lock to EOF
lock.l_pid = getpid();        // Your process ID
```

```
RED, fd, 0);

    char *write_ptr = sta
    MessageHeader header{
    std::memcpy(write_ptr
    std::memcpy(write_ptr

    munmap(addr, new_size

    // Release file lock
    lock.l_type = F_UNLCK;
    fcntl(fd, F_SETLK, &lock);

    close(fd);
}
```

`F_SETLK` : Set the lock *as specified by the structure* (non-blocking)

`lock.l_type = F_RDLCK` : acquire a shared (read) lock

`lock.l_type = F_WRLCK` : acquire an exclusive (write) lock

`lock.l_type = F_UNLCK` : **release** the lock

Kenneth J. Roche
University of Washington
AMATH 483 / 583 High Performance Scientific Computing
Spring Quarter 2025

- *(Continue from Lecture 13) ipc (using mmap in a file to pass messages as shared communication) for > 2 processes : lock / serialize writing to common resource*

```
bash-3.2$ g++ -std=c++17 kr-mmap-messaging-np-safe.cpp -o xmmap-messaging-np-safe
bash-3.2$ ./xmmap-messaging-np-safe 5
Child 2 received from 5016: Hello to child 2
Child 4 received from 5016: Hello to child 4
Child 3 received from 5016: Hello to child 3
Child 0 received from 5016: Hello to child 0
Child 1 received from 5016: Hello to child 1
All child processes finished.
Parent received from 5020: Hello from child 2
Parent received from 5022: Hello from child 4
Parent received from 5018: Hello from child 0
Parent received from 5021: Hello from child 3
Parent received from 5019: Hello from child 1
bash-3.2$ ./xmmap-messaging-np-safe 10
Child 0 received from 5023: Hello to child 0
Child 1 received from 5023: Hello to child 1
Child 4 received from 5023: Hello to child 4
Child 3 received from 5023: Hello to child 3
Child 8 received from 5023: Hello to child 8
Child 9 received from 5023: Hello to child 9
Child 5 received from 5023: Hello to child 5
Child 7 received from 5023: Hello to child 7
Child 2 received from 5023: Hello to child 2
Child 6 received from 5023: Hello to child 6
All child processes finished.
Parent received from 5024: Hello from child 0
Parent received from 5025: Hello from child 1
Parent received from 5028: Hello from child 4
Parent received from 5027: Hello from child 3
Parent received from 5032: Hello from child 8
Parent received from 5033: Hello from child 9
Parent received from 5029: Hello from child 5
Parent received from 5031: Hello from child 7
Parent received from 5026: Hello from child 2
Parent received from 5030: Hello from child 6
bash-3.2$
```

`F_WRLCK` **(write lock) is exclusive.**

- When a sender uses `F_WRLCK`, it *prevents* any other process from acquiring either a read (`F_RDLCK`) or write (`F_WRLCK`) lock on the file region.
- This ensures that only **one writer at a time** can access the memory-mapped file, preventing race conditions during appends.

linear algebra concepts (continued)

- machine learning connections

In the context of neural networks, a "forward" function is responsible for propagating input data through the network to produce an output. This process involves a series of matrix multiplications and nonlinear activations.

1. **Matrix Multiplication**: In a neural network, the weights between neurons are typically represented as a matrix. During the forward pass, input data (which can also be represented as a matrix) is multiplied by these weight matrices to produce activations for the next layer.

2. **Solving Systems of Linear Equations**: In a neural network, each layer can be seen as solving a set of equations where the unknowns are the activations of the neurons. The weights of the network serve as coefficients of these equations. During the forward pass, the network combines these equations through matrix multiplications and activations to produce the final output.

3. **Vector Space Transformations**: Each layer in a neural network can be thought of as transforming the input data from one vector space to another. The weights of the network determine the nature of this transformation. During the forward pass, the input vector is transformed through a series of linear transformations (matrix multiplications) and nonlinear transformations (activation functions) to map it to the output space.

Thus the "forward" function in a neural network can be related to linear algebra concepts such as matrix multiplication, solving systems of linear equations (albeit in a more abstract sense), and vector space transformations. These connections highlight the mathematical underpinnings of neural networks and their relationship to fundamental linear algebra concepts.

machine learning connections

- forward pass in a basic feed forward neural network with one hidden layer
- want to demonstrate how matrix multiplication and activation functions are used in the forward pass

1. Matrix-Vector Multiplication:
   - Given:
     - X : Input vector of size $n \times 1$
     - W_1 : Weight matrix of the first layer of size $m \times n$
     - b_1 : Bias vector of the first layer of size $m \times 1$
   - Calculate:
     - $Z_1 = W_1 \cdot X + b_1$
   - Result:
     - Z_1 : Intermediate vector of size $m \times 1$
2. Activation Function:
   - Apply the sigmoid activation function element-wise to each element of Z_1 to get A_1:
     - $A_1[i] = \text{sigmoid}(Z_1[i] + b_1[i])$ for $i = 0, 1, ..., m-1$
3. Matrix-Vector Multiplication:
   - Given:
     - A_1 : Output vector from the first layer of size $m \times 1$
     - W_2 : Weight matrix of the second layer of size $p \times m$
     - b_2 : Bias vector of the second layer of size $p \times 1$
   - Calculate:
     - $Z_2 = W_2 \cdot A_1 + b_2$
   - Result:
     - Z_2 : Output vector of size $p \times 1$
4. Activation Function:
   - Apply the sigmoid activation function element-wise to each element of Z_2 to get the final output vector:
     - $\text{output}[i] = \text{sigmoid}(Z_2[i] + b_2[i])$ for $i = 0, 1, ..., p-1$

Kenneth J. Roche
University of Washington
AMATH 483 / 583 High Performance Scientific Computing
Spring Quarter 2025

- span, linear independence, row & column & null space, rank and nullity

another aside on linear algebra. introduction to

K.J. Roche.

linearly independent vectors - span.

if $S = \{\vec{v}_i\}$ $i = 1, m$ and $v_i \in \mathbb{R}^n$ or $\mathbb{F}^n$

it may be that S spans $\mathbb{R}^n$ or $\mathbb{C}^n$.

In this case any vector can be represented as a linear combination of members of S.

let $\vec{w} \in \mathbb{R}^n$, then $\vec{w} = a_1\vec{v}_1 + a_2\vec{v}_2 + \dots + a_m\vec{v}_m$.

for some $a_i \in \mathbb{R}$, $\mathbb{C}$. $a_i \in \mathbb{F}$ ← some field, $\mathbb{R}$ or $\mathbb{C}$.

linear dependence.

ie, trivial when $a_i = 0$ $\forall i$.

if $\exists$ any non-trivial set $\{a_i\} \ni \sum_{i=1}^{m} a_i\vec{v}_i = \vec{0}$,

then S is a linearly dependent set.

linear independence.

when $\nexists \{a_i\} \ni \sum_{i=1}^{m} a_i\vec{v}_i = \vec{0}$ other than trivial case.

If S linearly independent,

$\{\vec{v}_1, \vec{v}_2\}$. $\vec{v}_1 = \begin{pmatrix}1\\0\\0\end{pmatrix}$ $\vec{v}_2 = \begin{pmatrix}0\\0\\1\end{pmatrix}$.

clearly $a_1\vec{v}_1 + a_2\vec{v}_2 = \begin{pmatrix}a_1\\0\\0\end{pmatrix} + \begin{pmatrix}0\\0\\a_2\end{pmatrix} = \begin{pmatrix}a_1\\0\\a_2\end{pmatrix}$

this only $= \begin{pmatrix}0\\0\\0\end{pmatrix}$ when $a_1 = a_2 = 0$. ← so v_1 & v_2 are independent. but clearly don't span $\mathbb{R}^3$.

since. $\vec{w} = \begin{pmatrix}w_1\\w_2\\w_3\end{pmatrix} = a_1\begin{pmatrix}1\\0\\0\end{pmatrix} + a_2\begin{pmatrix}0\\0\\1\end{pmatrix} = \begin{pmatrix}a_1\\0\\a_2\end{pmatrix}$

if $w_1 = a_1$, $w_3 = a_2$ - that's part of it.

but $w_2 = 0$ isn't generally true. //

any set that contains the $\vec{0}$ vector is linearly dependent by definition.

if $m > n$, then S is linearly dependent set.

let $n = 2$, $m = 3$,

x, y, z. ∋ $x = \begin{pmatrix}x_1\\x_2\end{pmatrix}$, $y = \begin{pmatrix}y_1\\y_2\end{pmatrix}$, $z = \begin{pmatrix}z_1\\z_2\end{pmatrix}$.

Then $a_1\vec{x} + a_2\vec{y} + a_3\vec{z} = \vec{0} \Leftrightarrow \vec{z} = \dfrac{-a_1\vec{x} - a_2\vec{y}}{a_3}$ //

let $a_3 = -\dfrac{(a_1x_2 + a_2y_2)}{z_2}$. $z_2 \neq 0$.

clearly for this a_3 ii) is true and

$a_1, a_2, a_3 \neq 0$.

S a basis for a linear vector space $\vec{V}$ when:

i) S spans $\vec{V}$ and 2) S lin. independent.

S bases Then $\forall \vec{w} \in \vec{V}$, $\exists$ a unique expansion in S.

Kenneth J. Roche
University of Washington
AMATH 483 / 583 High Performance Scientific Computing
Spring Quarter 2025

some bases:

$\{\vec{e}_i\}_{\text{basis}}$ $\{1, x, x^2, \cdots, x^n\}$, $\hat{M} = \{M_i\} \ni M_i \in \mathbb{R}^2 \overset{2}{\times} \mathbb{R}^2$

$M_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$, $M_2 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$

$M_3 = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$, $M_4 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$

etc.

---

other.

* any 2 basis sets for vector space $\hat{V}$ have the same number of vectors.

* dimension of $\hat{V}$, $\dim(V)$, is the number of vectors in the basis set.

Row & Column space.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{m1} & - & - & a_{mn} \end{pmatrix} \rightarrow r_i = a_{i1}\ a_{i2}\ a_{i3} \cdots a_{in} \quad \forall i = 1, m.$$ } Rows

also, $c_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{pmatrix} \forall j = 1, n.$ } columns.

note $A\vec{x} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{matrix} x_1 a_{11} + x_2 a_{12} \\ x_1 a_{21} + x_2 a_{22} \end{matrix} = x_1\begin{pmatrix} a_{11} \\ a_{21} \end{pmatrix} + x_2\begin{pmatrix} a_{12} \\ a_{22} \end{pmatrix}$

$= x_1 \vec{c}_1 + x_2 \vec{c}_2$ //

$A \in \mathbb{R}^{m,n}$.

row space, subspace of $\mathbb{R}^n$ spanned by $R = \{\vec{r}_i\}$ $i = 1, m$

$\vec{r}_i = (r_{i1}\ r_{i2} \cdots r_{in})$

column space., subspace of $\mathbb{R}^m$ spanned by $C = \{\vec{c}_j\}$ $j = 1, n$.

$\vec{c}_j = \begin{pmatrix} c_{1j} \\ c_{2j} \\ \vdots \\ c_{mj} \end{pmatrix}$.

null space, subspace of $\mathbb{R}^n$, is solution space of homogeneous system $A\vec{x} = \vec{0}$

* take A & reduce to row echelon form.

↳ leading columns contains leading 1's and zeroes only.

Kenneth J. Roche
University of Washington
AMATH 483 / 583 High Performance Scientific Computing
Spring Quarter 2025

Ex.

$$A = \begin{pmatrix} 2 & 2 & -1 & 0 & 1 \\ -1 & -1 & 2 & -3 & 1 \\ 1 & 1 & -2 & 0 & -1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

find null space.

affiliated w/ rows!!

ie subspace of $\mathbb{R}^n$.

here, $n = 5$, $m = 4$.

add $\vec{r}_3$ to $\vec{r}_2$.

add $-2\vec{r}_3$ to $\vec{r}_1$.

$$\begin{pmatrix} 0 & 0 & 3 & 0 & 3 \\ 0 & 0 & 0 & -3 & 0 \\ 1 & 1 & -2 & 0 & -1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

Kenneth J. Roche
University of Washington
AMATH 483 / 583 High Performance Scientific Computing
Spring Quarter 2025

$$\begin{pmatrix} 1 & 1 & -2 & 0 & -1 \\ 0 & 0 & 3 & 0 & 3 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & -3 & 0 \end{pmatrix} \begin{matrix} \\ \frac{1}{3}\cdot(-1)\vec{r}_2 + \vec{r}_3 \\ (2)\vec{r}_2 + \vec{r}_1 \\ -\frac{1}{3}\cdot(-1)\cdot\vec{r}_4 + \vec{r}_3 \end{matrix} \Rightarrow \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

1. $x_1 + x_2 + x_5 = 0$
2. $x_3 + x_5 = 0$
3. $x_4 = 0$

$\Rightarrow$ ☆

$x_1 = -x_2 - x_5$.

$x_3 = -x_5$.

let $x_2 = s$

$x_5 = t$.

☆ see

$(-s-t) + s + t = 0.$

$-t + t = 0.$

$\Rightarrow$ ok.

Go. solution:

$$\vec{w} = \begin{pmatrix} -s-t \\ s \\ -t \\ 0 \\ t \end{pmatrix} = \begin{pmatrix} -s \\ s \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -t \\ 0 \\ -t \\ 0 \\ t \end{pmatrix}$$

$$= s\begin{pmatrix} -1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + t\begin{pmatrix} -1 \\ 0 \\ -1 \\ 0 \\ 1 \end{pmatrix}$$

so that.

$\hat{r}_1 = -1\ 1\ 0\ 0\ 0$

$\hat{r}_2 = -1\ 0\ -1\ 0\ 1$

← This is the set of vectors that spans the null space.

note. The nonzero row vectors in any row echelon form of a matrix A form a basis for the row-space of A, and thus the original row vectors of A.

also, if A and R are row equivalent (one is row-echelon form of other), a set of column vectors of A is a basis for the column space of A. iff the corresponding columns of R form a basis for column space of R.

✱ For ANY matrix A, the row space and column space have the same dimension — same number of basis vectors. This dimension is called the rank of A, rank(A). The dimension of the null space — ie $A\vec{x} = \vec{0}$ $\leftrightarrow$ is called the nullity of A, nullity(A).

✱ for any matrix A w/ n columns

$$\text{rank}(A) + \text{nullity}(A) = n.$$

↑ big result!

Thm $A\vec{x} = \vec{b}$ consistent iff b in column space of A.

but, $x_1\vec{a}_1 + x_2\vec{a}_2 + \cdots + x_n\vec{a}_n = \vec{b}$ aug$(A) = (A \,|\, b)$ ↑

implies we solve for $\vec{x}$. We need rank(A) = rank(aug(A)) // (row-reduce.)

non-homogeneous, $A\vec{x} = \vec{b}$ } same A.

homogeneous, $A\vec{x} = \vec{0}$

$Ax = b$. suppose $\vec{x}_0 \ni A\vec{x}_0 - \vec{b} = \vec{0}$ } $\vec{x}_0$, particular soln.

also $\{\vec{v}_i\} \ni A\{c_i\vec{v}_i\} = \vec{0}$

where $\{\vec{v}_i\}$ are the basis vectors for the homogeneous system.

then $\hat{x} = \vec{x}_0 + \sum_i c_i\vec{v}_i$ is } $\hat{x}$, general soln.

consistent solution. $A\hat{x} - b - v$.

Kenneth J. Roche
University of Washington
AMATH 483 / 583 High Performance Scientific Computing
Spring Quarter 2025

- linear systems first pass - Gauss elimination

Kenneth J. Roche
University of Washington
AMATH 483 / 583 High Performance Scientific Computing
Spring Quarter 2025

- # linear systems, elementary matrices, Gauss elimination

given an n-vector $\boldsymbol{a}$, we can annihilate *all* of its entries below the kth position, provided that $a_k \neq 0$, by the following transformation:

$$\boldsymbol{M}_k \boldsymbol{a} = \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & -m_{k+1} & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -m_n & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_k \\ a_{k+1} \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} a_1 \\ \vdots \\ a_k \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

where $m_i = a_i/a_k$, $i = k+1, \ldots, n$. The divisor a_k is called the *pivot*. A matrix of this form is sometimes called an *elementary elimination matrix* or *Gauss transformation*, and its effect on a vector is to add a multiple of row k to each subsequent row, with the multipliers m_i chosen so that the result in each case is zero.

Kenneth J. Roche
University of Washington
AMATH 483 / 583 High Performance Scientific Computing
Spring Quarter 2025

1. $\boldsymbol{M}_k$ is a lower triangular matrix with unit main diagonal, and hence it must be nonsingular.
2. $\boldsymbol{M}_k = \boldsymbol{I} - \boldsymbol{m}\boldsymbol{e}_k^T$, where $\boldsymbol{m} = [0, \ldots, 0, m_{k+1}, \ldots, m_n]^T$ and $\boldsymbol{e}_k$ is the kth column of the identity matrix.
3. $\boldsymbol{M}_k^{-1} = \boldsymbol{I} + \boldsymbol{m}\boldsymbol{e}_k^T$, which means that $\boldsymbol{M}_k^{-1}$, which we will denote by $\boldsymbol{L}_k$, is the same as $\boldsymbol{M}_k$ except that the signs of the multipliers are reversed.
4. If $\boldsymbol{M}_j$, $j > k$, is another elementary elimination matrix, with vector of multipliers $\boldsymbol{t}$, then

$$\boldsymbol{M}_k \boldsymbol{M}_j = \boldsymbol{I} - \boldsymbol{m}\boldsymbol{e}_k^T - \boldsymbol{t}\boldsymbol{e}_j^T + \boldsymbol{m}\boldsymbol{e}_k^T \boldsymbol{t}\boldsymbol{e}_j^T = \boldsymbol{I} - \boldsymbol{m}\boldsymbol{e}_k^T - \boldsymbol{t}\boldsymbol{e}_j^T,$$

since $\boldsymbol{e}_k^T \boldsymbol{t} = 0$. Thus, their product is essentially their "union." Because they have the same form, a similar result holds for the product of their inverses, $\boldsymbol{L}_k \boldsymbol{L}_j$. Note that the order of multiplication is significant; these results do not hold for the reverse product.

Elementary Elimination Matrices. If $\boldsymbol{a} = [2 \quad 4 \quad -2]^T$, then

$$\boldsymbol{M}_1 \boldsymbol{a} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \\ -2 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{M}_2 \boldsymbol{a} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0.5 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \\ -2 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 0 \end{bmatrix}.$$

We also note that

$$\boldsymbol{L}_1 = \boldsymbol{M}_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{L}_2 = \boldsymbol{M}_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -0.5 & 1 \end{bmatrix},$$

and

$$\boldsymbol{M}_1 \boldsymbol{M}_2 = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0.5 & 1 \end{bmatrix}, \quad \boldsymbol{L}_1 \boldsymbol{L}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -0.5 & 1 \end{bmatrix}.$$

Solve: linear system $\boldsymbol{Ax} = \boldsymbol{b}$

$$\boldsymbol{A} = \boldsymbol{LU} \qquad \boldsymbol{LUx} = \boldsymbol{b} \qquad \boldsymbol{Ly} = \boldsymbol{b} \qquad \boldsymbol{Ux} = \boldsymbol{y}$$

define the matrix $\boldsymbol{M} = \boldsymbol{M}_{n-1} \cdots \boldsymbol{M}_1$

$$\boldsymbol{MAx} = \boldsymbol{M}_{n-1} \cdots \boldsymbol{M}_1 \boldsymbol{Ax} = \boldsymbol{M}_{n-1} \cdots \boldsymbol{M}_1 \boldsymbol{b} = \boldsymbol{Mb}$$

$$\boldsymbol{U} = \boldsymbol{MA}$$

$$\boldsymbol{L} = \boldsymbol{M}^{-1} = (\boldsymbol{M}_{n-1} \cdots \boldsymbol{M}_1)^{-1} = \boldsymbol{M}_1^{-1} \cdots \boldsymbol{M}_{n-1}^{-1} = \boldsymbol{L}_1 \cdots \boldsymbol{L}_{n-1}$$

Note that $\boldsymbol{L}_k \boldsymbol{L}_j$ is unit lower triangular $k < j$

LU direct decomposition of A

```
for k = 1 to n - 1                          { loop over columns }
    if a_kk = 0 then stop                   { stop if pivot is zero }
    for i = k + 1 to n                      { compute multipliers
        m_ik = a_ik / a_kk                      for current column }
    end
    for j = k + 1 to n
        for i = k + 1 to n                  { apply transformation to
            a_ij = a_ij - m_ik a_kj             remaining submatrix }
        end
    end
end
```

$$
\begin{aligned}
x_1 + 2x_2 + 2x_3 &= 3, \\
4x_1 + 4x_2 + 2x_3 &= 6, \\
4x_1 + 6x_2 + 4x_3 &= 10,
\end{aligned}
\qquad
\boldsymbol{A}\boldsymbol{x} = \begin{bmatrix} 1 & 2 & 2 \\ 4 & 4 & 2 \\ 4 & 6 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \\ 10 \end{bmatrix} = \boldsymbol{b}
$$

$$
\boldsymbol{M}_1\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 \\ -4 & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 2 \\ 4 & 4 & 2 \\ 4 & 6 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 2 \\ 0 & -4 & -6 \\ 0 & -2 & -4 \end{bmatrix}
\qquad
\boldsymbol{M}_1\boldsymbol{b} = \begin{bmatrix} 1 & 0 & 0 \\ -4 & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 6 \\ 10 \end{bmatrix} = \begin{bmatrix} 3 \\ -6 \\ -2 \end{bmatrix}
$$

$$
\boldsymbol{M}_2\boldsymbol{M}_1\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -0.5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 2 \\ 0 & -4 & -6 \\ 0 & -2 & -4 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 2 \\ 0 & -4 & -6 \\ 0 & 0 & -1 \end{bmatrix}
\qquad
\boldsymbol{M}_2\boldsymbol{M}_1\boldsymbol{b} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -0.5 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ -6 \\ -2 \end{bmatrix} = \begin{bmatrix} 3 \\ -6 \\ 1 \end{bmatrix}
$$

$$
\boldsymbol{U}\boldsymbol{x} = \begin{bmatrix} 1 & 2 & 2 \\ 0 & -4 & -6 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ -6 \\ 1 \end{bmatrix} = \boldsymbol{M}\boldsymbol{b} = \boldsymbol{y}
$$

$$
\mathsf{x} = \begin{bmatrix} -1 & 3 & -1 \end{bmatrix}^T
$$

solve by back substitution

Kenneth J. Roche
University of Washington

$$\boldsymbol{L}_1\boldsymbol{L}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0.5 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 4 & 0.5 & 1 \end{bmatrix} = \boldsymbol{L}$$

$$\boldsymbol{A} = \begin{bmatrix} 1 & 2 & 2 \\ 4 & 4 & 2 \\ 4 & 6 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 4 & 0.5 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 2 \\ 0 & -4 & -6 \\ 0 & 0 & -1 \end{bmatrix} = \boldsymbol{LU}$$

- generating a discrete orthonormal basis on a digital lattice -concepts

$\mathbb{X}$ inner product space. If $x \in \mathbb{X}$ and $x \in \text{span}\{e_1, e_2, \ldots, e_n\}$ n fixed, $x = \alpha_i e_i$.

$(x, e_j) = (\alpha_i e_i, e_j) = \sum_i \alpha_i (e_i, e_j) = \sum_i \alpha_i \delta_{ij} = \alpha_j \Rightarrow x = \sum_{i=1}^{n} (x, e_i) e_i$. Now,

$\forall\, x \in \mathbb{X}$ and x not necessarily in $Y_n = \text{span}\{e_1, e_2, \ldots, e_n\}$, we can write $x = y + z$

where $y \in Y_n$. Here $z = x - y$ and $z \perp y$. This will be true for a particular choice of y. let $y = \sum_i \alpha_i e_i \in Y_n$. and choose $\alpha_i = (x, e_i)$ — The projection of x onto Y_n.

then $\|y\|^2 = (y, y) = \left(\sum_i \alpha_i e_i, \sum_j \alpha_j e_j\right) = \sum_i \sum_j \alpha_i \bar{\alpha}_j (e_i, e_j) = \sum_i \sum_j \alpha_i \bar{\alpha}_j \delta_{ij}$ ...

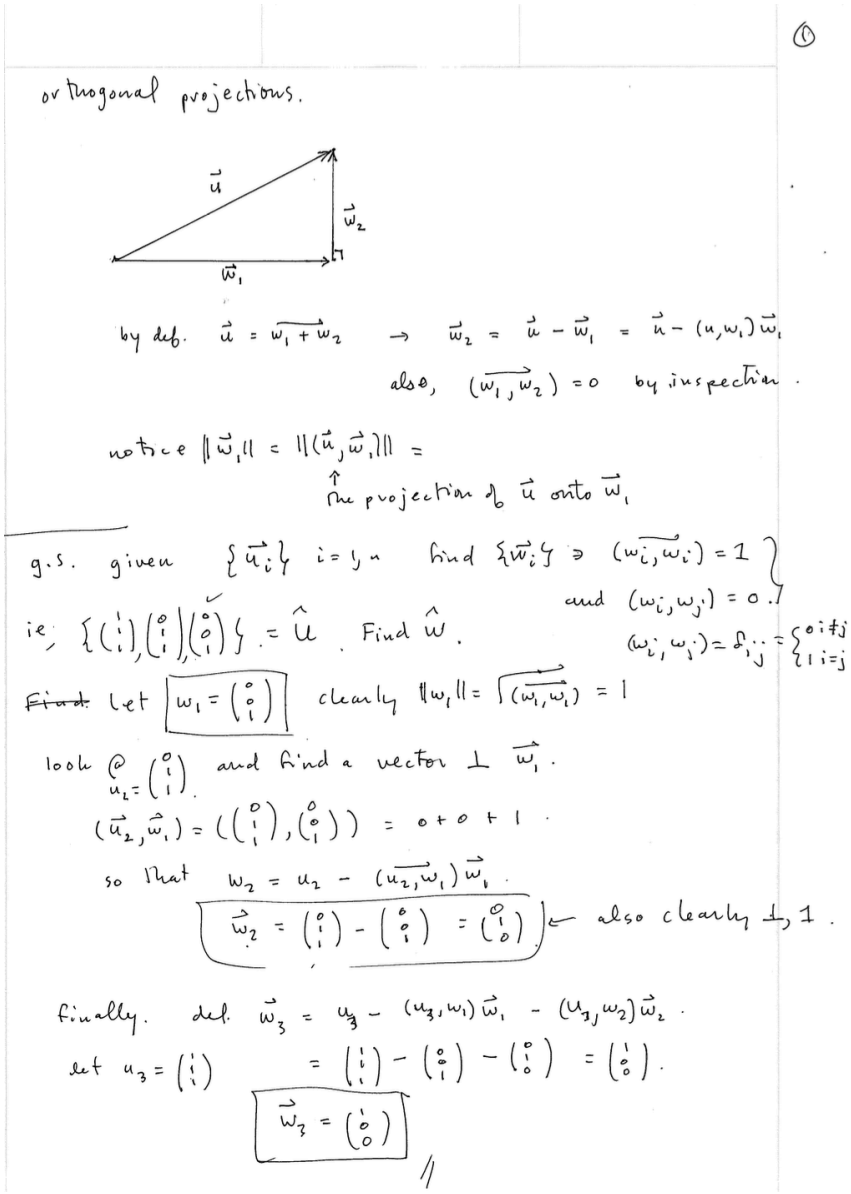
$= \sum_i |\alpha_i|^2 = \sum_i |(x, e_i)|^2$. AND if $z \perp y$ then $(z, y) = 0$. Check. $(x - y, y) = (x, y) - (y, y)$.

$= \left(x, \sum_i (x, e_i) e_i\right) - \|y\|^2 = \sum_i \overline{(x, e_i)} (x, e_i) - \sum_i |(x, e_i)|^2 = 0$. So, for this choice of y the statement holds, $z = x - y$ is $\perp$ y.

Kenneth J. Roche
University of Washington
AMATH 483 / 583 High Performance Scientific Computing
Spring Quarter 2025

orthogonal projections and construction of orthonormal basis set

Kenneth J. Roche
University of Washington
AMATH 483 / 583 High Performance Scientific Computing
Spring Quarter 2025



① orthogonal projections.

by def. $\vec{u} = \vec{w_1} + \vec{w_2} \rightarrow \vec{w}_2 = \vec{u} - \vec{w}_1 = \vec{u} - (u, w_1)\vec{w}_1$

also, $(\vec{w_1}, \vec{w_2}) = 0$ by inspection.

notice $\|\vec{w}_1\| = \|(\vec{u}, \vec{w}_1)\| =$ ↑ the projection of $\vec{u}$ onto $\vec{w}_1$

g.s. given $\{\vec{u}_i\}$ $i = 1, n$ find $\{\vec{w}_i\} \ni (\vec{w_i}, \vec{w_i}) = 1$ and $(w_i, w_j) = 0$.

ie; $\{\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}\} = \hat{u}$. Find $\hat{w}$.

$(w_i, w_j) = \delta_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$

~~Find~~ Let $\boxed{w_1 = \begin{pmatrix}0\\1\end{pmatrix}}$ clearly $\|w_1\| = \sqrt{(\vec{w_1}, \vec{w_1})} = 1$

look @ $u_2 = \begin{pmatrix}0\\1\end{pmatrix}$ and find a vector $\perp \vec{w}_1$.

$(\vec{u}_2, \vec{w}_1) = \left(\begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}\right) = 0 + 0 + 1$

so that $w_2 = u_2 - (\vec{u_2}, \vec{w_1})\vec{w}_1$

$\boxed{\vec{w}_2 = \begin{pmatrix}0\\1\end{pmatrix} - \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix}}$ ← also clearly $\perp$, 1.

finally. def $\vec{w}_3 = u_3 - (u_3, w_1)\vec{w}_1 - (u_3, w_2)\vec{w}_2$.

let $u_3 = \begin{pmatrix}1\\1\end{pmatrix}$ $= \begin{pmatrix}1\\1\end{pmatrix} - \begin{pmatrix}0\\1\end{pmatrix} - \begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix}$.

$\boxed{\vec{w}_3 = \begin{pmatrix}0\\0\end{pmatrix}}$ //

one more example.

$$\{\vec{u_1}, \vec{u_2}, \vec{u_3}\} = \left\{\begin{pmatrix}1\\1\\1\end{pmatrix}, \begin{pmatrix}-1\\1\\0\end{pmatrix}, \begin{pmatrix}1\\2\\1\end{pmatrix}\right\}.$$

First. def. $\vec{w}_1 = \dfrac{u_1}{\|u_1\|}$ $\qquad \|u_1\| = (u_1, u_1)^{1/2} = \sqrt{1^2+1^2+1^2} = \sqrt{3}.$

so. $\boxed{\vec{w}_1 = \frac{1}{\sqrt{3}}\begin{pmatrix}1\\1\\1\end{pmatrix}}$

now. form $\vec{w}_2 = u_2 - (u_2, \vec{w}_1)\vec{w}_1$ $\qquad (\vec{u_2}, \vec{w_1}) = \underbrace{(-1\ 1\ 0)}\ \frac{1}{\sqrt{3}}\begin{pmatrix}1\\1\\1\end{pmatrix}.$

$= \vec{u}_2.$ $\qquad = \frac{-1}{\sqrt{3}} + \frac{1}{\sqrt{3}} + 0 = 0.$

$\|\vec{u}_2\| = (u_2, u_2)^{1/2} = \sqrt{(-1)^2 + (1)^2} = \sqrt{2}.$

$\Rightarrow \boxed{\vec{w}_2 = \frac{1}{\sqrt{2}}\begin{pmatrix}-1\\1\\0\end{pmatrix}}$

Finally. $\vec{w}_3 = u_3 - (u_3, w_2)\vec{w}_2 - (u_3, w_1)\vec{w}_1$

$(u_3, w_2) = \underbrace{(1\ 2\ 1)}\ \frac{1}{\sqrt{2}}\begin{pmatrix}-1\\1\\0\end{pmatrix} = \frac{1}{\sqrt{2}}(-1+2+0) = \frac{1}{\sqrt{2}}.$

$(u_3, w_1) = \underbrace{(1\ 2\ 1)}\ \frac{1}{\sqrt{3}}\begin{pmatrix}1\\1\\1\end{pmatrix} = \frac{1}{\sqrt{3}}(1+2+1) = \frac{4}{\sqrt{3}}.$

$$= \begin{pmatrix}1\\2\\1\end{pmatrix} - \frac{1}{\sqrt{2}}\cdot\frac{1}{\sqrt{2}}\begin{pmatrix}-1\\1\\0\end{pmatrix} - \frac{4}{\sqrt{3}}\cdot\frac{1}{\sqrt{3}}\begin{pmatrix}1\\1\\1\end{pmatrix}$$

$$= \begin{pmatrix}1\\2\\1\end{pmatrix} - \begin{pmatrix}-\frac{1}{2}\\\frac{1}{2}\\0\end{pmatrix} - \begin{pmatrix}\frac{4}{3}\\\frac{4}{3}\\\frac{4}{3}\end{pmatrix}$$

$$\vec{w}_3 = \begin{pmatrix}1+\frac{1}{2}-\frac{4}{3}\\2-\frac{1}{2}-\frac{4}{3}\\1-0-\frac{4}{3}\end{pmatrix} = \begin{pmatrix}\frac{6}{6}+\frac{3}{6}-\frac{8}{6}\\\frac{12}{6}-\frac{3}{6}-\frac{8}{6}\\\frac{3}{3}-\frac{4}{3}\end{pmatrix} = \begin{matrix}\frac{1}{6}\\\frac{1}{6}\\-\frac{1}{3}=-\frac{2}{6}\end{matrix} = \frac{1}{6}\begin{pmatrix}1\\1\\-2\end{pmatrix}$$

$\vec{w}_3 = \frac{1}{6}\begin{pmatrix}1\\1\\-2\end{pmatrix}.$ $\qquad \|\vec{w}_3\| = (w_3, w_3)^{1/2} = \sqrt{\frac{1}{36}\cdot\sqrt{1^2+1^2+2^2}}$

$= \frac{1}{6}\cdot\sqrt{6} = \frac{\sqrt{6}}{6}.$ ↓

$\vec{w}_3 \to \dfrac{1}{\|\vec{w}_3\|}\cdot\vec{w}_3 = \dfrac{1}{\left(\frac{\sqrt{6}}{6}\right)}\cdot\frac{1}{6}\begin{pmatrix}1\\1\\-2\end{pmatrix} = \frac{6}{\sqrt{6}}\cdot\frac{1}{6}\begin{pmatrix}1\\1\\-2\end{pmatrix} = \frac{1}{\sqrt{6}}\begin{pmatrix}1\\1\\-2\end{pmatrix}$ //

check $\|\vec{w}_3\| = \left[\underbrace{\frac{1}{\sqrt{6}}(1\ 1\ -2)}\cdot\frac{1}{\sqrt{6}}\begin{pmatrix}1\\1\\-2\end{pmatrix}\right]^{1/2} = \sqrt{\frac{1}{6}(1^2+1^2+(-2)^2)}$

$= 1$ //

check $(\vec{w_3}, \vec{w_1}) = \underbrace{\frac{1}{\sqrt{6}}(1\ 1\ -2)}\ \frac{1}{\sqrt{3}}\begin{pmatrix}1\\1\\1\end{pmatrix} = \frac{1}{\sqrt{18}}\cdot(\underbrace{1\cdot1 + 1\cdot1 + (-2)\cdot1}_{2-2=0}) = 0.$ //

$(\vec{w_3}, \vec{w_2}) = \frac{1}{\sqrt{6}}\underbrace{(1\ 1\ -2)}\ \frac{1}{\sqrt{2}}\begin{pmatrix}-1\\1\\0\end{pmatrix} = \frac{1}{\sqrt{12}}(\underbrace{1\cdot(-1)}_{-1} + \underbrace{1\cdot1}_{+1} + (-2)\cdot0) = 0.$ //

$(\vec{w_1}, \vec{w_2}) = \frac{1}{\sqrt{3}}\underbrace{(1\ 1\ 1)}\ \frac{1}{\sqrt{2}}\begin{pmatrix}-1\\1\\0\end{pmatrix} = \frac{1}{\sqrt{6}}(-1+1+0) = 0.$ //

check now, $\vec{u}_1 \overset{?}{=} (\vec{u_1}, \vec{w}_1)\vec{w}_1 + (\vec{u_1}, \vec{w_2})\vec{w}_2 + (\vec{u_1}, \vec{w_3})\vec{w}_3$

$$\begin{pmatrix}1\\1\\1\end{pmatrix} = \left[\underbrace{(1\ 1\ 1)}\ \frac{1}{\sqrt{3}}\begin{pmatrix}1\\1\\1\end{pmatrix}\right]\frac{1}{\sqrt{3}}\begin{pmatrix}1\\1\\1\end{pmatrix} + \left[\underbrace{(1\ 1\ 1)}\ \frac{1}{\sqrt{2}}\begin{pmatrix}-1\\1\\0\end{pmatrix}\right]\frac{1}{\sqrt{2}}\begin{pmatrix}-1\\1\\0\end{pmatrix}$$

$$+ \left[\underbrace{(1\ 1\ 1)}\ \frac{1}{\sqrt{6}}\begin{pmatrix}1\\1\\-2\end{pmatrix}\right]\frac{1}{\sqrt{6}}\begin{pmatrix}1\\1\\-2\end{pmatrix}$$

$$= \frac{1}{\sqrt{3}}\cdot3\cdot\frac{1}{\sqrt{3}}\begin{pmatrix}1\\1\\1\end{pmatrix} + \frac{1}{\sqrt{2}}\cdot0\cdot\frac{1}{\sqrt{2}}\begin{pmatrix}-1\\1\\0\end{pmatrix} + \frac{1}{\sqrt{6}}\cdot0\cdot\frac{1}{\sqrt{6}}\begin{pmatrix}1\\1\\-2\end{pmatrix}$$

$= \frac{3}{3}\begin{pmatrix}1\\1\\1\end{pmatrix} = \begin{pmatrix}1\\1\\1\end{pmatrix}$ // ✓

$\vec{u}_2$? $\quad \vec{u}_2 = (u_2, w_1)w_1 + (u_2, w_2)w_2 + (u_2, w_3)w_3.$

$= \underbrace{(-1\ 1\ 0)}\ \frac{1}{\sqrt{3}}\begin{pmatrix}1\\1\\1\end{pmatrix} + \underbrace{(-1\ 1\ 0)}\ \frac{1}{\sqrt{2}}\begin{pmatrix}-1\\1\\0\end{pmatrix}\cdot\frac{1}{\sqrt{2}}\begin{pmatrix}-1\\1\\0\end{pmatrix} + \underbrace{(-1\ 1\ 0)}\ \frac{1}{\sqrt{6}}\begin{pmatrix}1\\1\\-2\end{pmatrix}$

$\begin{pmatrix}-1\\1\\0\end{pmatrix} = \frac{2}{2}\begin{pmatrix}-1\\1\\0\end{pmatrix} = \begin{pmatrix}-1\\1\\0\end{pmatrix}$ ✓ // etc.

Kenneth J. Roche
University of Washington
AMATH 483 / 583 High Performance Scientific Computing
Spring Quarter 2025

- Fourier sequences - useful orthonormal sets ??

Continuous Fnc. Example. $[0, 2\pi] \leftrightarrow [-\pi, \pi]$ and $(x,y) = \int_{-\pi}^{\pi} x(t)\, y(t)\, dt$ (Real Space)

The set $\{1, \cos nt, \sin nt\}$ is orthogonal. $\{u_n(t)\}$ is orthogonal sequence in X w/ $u_n(t) = \cos nt$, $n \geq 0$

$\{v_n(t)\}$ is orthogonal sequence in X with $v_n(t) = \sin nt$, $n \geq 1$. Seek (u_n, u_m), (v_n, v_m), (u_n, v_m).

$\rightarrow \int_{-\pi}^{\pi} \cos mt \cos nt\, dt = \begin{cases} 0 & m \neq n \\ \pi & m = n \neq 0 \\ 2\pi & m = n = 0 \end{cases}$. $\int_{-\pi}^{\pi} \sin mt \sin nt\, dt = \begin{cases} 0 & m \neq n \\ \pi & m = n \neq 0 \end{cases}$. $\int_{-\pi}^{\pi} \cos mt \sin nt\, dt = 0$.

normalize. we want that $(u_m, u_m) = 1$ but $(u_m, u_m) = \begin{cases} \pi & m \neq 0 \\ 2\pi & m = 0 \end{cases}$ so define $e_m = \frac{u_m}{\sqrt{(u_m, u_m)}}$.

We get $\{u_m\} \rightarrow \left\{\frac{1}{\sqrt{2\pi}}, \frac{1}{\sqrt{\pi}} \cos nt\right\}$ and $\{v_n\} \rightarrow \left\{\frac{1}{\sqrt{\pi}} \sin nt\right\}$. Fourier Sequences ...

- data analysis - best fit

Kenneth J. Roche
University of Washington
AMATH 483 / 583 High Performance Scientific Computing
Spring Quarter 2025

k. j. roche.

given set $\{(x_i, y_i)\}$ of experimental / observed values.

wish to find f ∋ $f(x) = y$ is a good fit to the data.

- data contains errors / isn't perfect.

plan: find a curve that "best" fits the data.

simple. try using a line so that

$$y = mx + b \quad \rightarrow \quad y_i = mx_i + b.$$

$$\left.\begin{aligned} y_1 &= mx_1 + b \\ y_2 &= mx_2 + b \\ &\vdots \\ y_n &= mx_n + b. \end{aligned}\right\} \quad \vec{y} = m\vec{x} + b = \begin{pmatrix} \vec{1} & \vec{x} \end{pmatrix} \begin{pmatrix} b \\ m \end{pmatrix}$$

$$\text{so, } \vec{y} = \begin{bmatrix} \vec{1} & \vec{x} \end{bmatrix} \begin{pmatrix} b \\ m \end{pmatrix}$$

$$\{\vec{y} = M\vec{v}\}$$

where $\vec{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$, $M = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}$, $\vec{v} = \begin{pmatrix} b \\ m \end{pmatrix}$

(linear dependence / independence / subspaces)

There are more equations than unknowns – overdetermined. ✱

add page 1b

Now, unless (x_i, y_i) are all colinear, then possible to satisfy the equality. It means $\|\overrightarrow{y - M\vec{v}}\| \neq 0$.

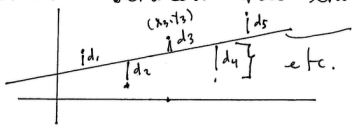
We seek $\hat{\vec{v}}$ ∋ $\|\overrightarrow{y - M\hat{\vec{v}}}\|$ minimizes the error, is as close to zero possible. $\sqrt{(y - Mw,\ y - Mw)} \equiv$ inner product.

$$\|y - Mw\|^2 = (y_1 - (b + mx_1))^2 + (y_2 - (b - mx_2))^2 + \cdots + (y_n - (b - mx_n))^2.$$

let $d_1 = |y_1 - b - mx_1|$, then $d_i^2 = (y_i - (b - mx_i))^2$ ∀ i.

and $\|y - Mw\|^2 = \sum_i d_i^2$. let d_i represent the

distance between the line and point (x_i, y_i).



look @ d_3 ⇒ $y_3 - d_3$ is on the line.

look @ d_4 ⇒ $y_4 + d_4$ is on the line.

that is $y_3 - d_3 = mx_3 + b$ ↳ etc.

and $y_4 + d_4 = mx_4 + b$.

✱ Thm. $\vec{w}$ minimizes $\|\vec{y} - M\vec{w}\|$ when $M\vec{w}$ is the ⊥ projection of $\vec{y}$ on the column space of M.

It means that $(y - M\vec{w}) \cdot M\vec{v} = 0 \quad \forall\ \vec{v} \in \mathbb{R}^2$

we have. $(y - Mw) \cdot M\vec{v} = (M\vec{v})^T \cdot (y - Mw)$

$= \vec{v}^T M^T (y - Mw)$

$= \vec{v}^T (M^T y - M^T M w)$

$= (M^T y - M^T M w) \cdot \vec{v} = 0$

since $\vec{v}$ arbitrary, $M^T y - M^T M w = 0$

$\rightarrow$ $M^T y = M^T M w$

here 2 eqns in 2 unknowns.

These are the normal eqns.

clearly $M^T y = M^T M \omega$ and we seek $\vec{\omega}$.

if $(M^T M)^{-1}$ exists, we're good.

$\vec{\omega} = (M^T M)^{-1} M^T y$ done. ☆ Normal eqns are solved!

Ex Find least squares fit to S.

$\{(0,1),(1,3),(2,4),(3,4)\} = S$ ← given.

$$\begin{aligned} 1 &= m\cdot 0 + b \\ 3 &= m\cdot 1 + b \\ 4 &= m\cdot 2 + b \\ 4 &= m\cdot 3 + b \end{aligned} \quad \Rightarrow \quad \begin{pmatrix} 1 \\ 3 \\ 4 \\ 4 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 1 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} m \\ b \end{pmatrix}.$$

so, $M = \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 1 \\ 3 & 1 \end{pmatrix}$ & $M^T = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{pmatrix}$.

$$M^T M = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 1 \\ 2 & 1 \\ 3 & 1 \end{pmatrix} = \begin{pmatrix} (0+1+2+3) & (1+1+1+1) \\ (0+1+4+9) & (0+1+2+3) \end{pmatrix}$$

$$N = \begin{pmatrix} 6 & 4 \\ 14 & 6 \end{pmatrix}$$

recall $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$

Then $A^{-1} = \frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$

↕

so long as $\det A \neq 0$.

$\det N = 36 - 56 = -20 \neq 0$.

$N^{-1} = \frac{1}{(-20)}\begin{pmatrix} 6 & -4 \\ -14 & 6 \end{pmatrix}$ ★

so that we have

$$\vec{\omega} = \underbrace{\frac{1}{(-20)}\begin{pmatrix} 6 & -4 \\ -14 & 6 \end{pmatrix}}_{-\frac{1}{10}\begin{pmatrix} 3 & -2 \\ -7 & 3 \end{pmatrix}} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \\ 4 \\ 4 \end{pmatrix}$$

$$= \underbrace{\begin{pmatrix} (1+3+4+4) \\ (0+3+8+12) \end{pmatrix}}_{= \begin{pmatrix} 12 \\ 23 \end{pmatrix}} = \begin{pmatrix} 12 \\ 23 \end{pmatrix}$$

$$= \left(-\frac{1}{10}\right)\begin{pmatrix} 3 & -2 \\ -7 & 3 \end{pmatrix}\begin{pmatrix} 12 \\ 23 \end{pmatrix}$$

$$= (\,)\begin{pmatrix} 3\cdot 12 - 2\cdot 23 \\ -7\cdot 12 + 3\cdot 23 \end{pmatrix} = -\frac{1}{10}\begin{pmatrix} 36-46 \\ -84+69 \end{pmatrix} = -\frac{1}{10}\begin{pmatrix} -10 \\ -15 \end{pmatrix} = \begin{pmatrix} \frac{-10}{-10} \\ \frac{-15}{-10} \end{pmatrix} = \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix}$$

$$\begin{array}{r} 84 \\ -69 \\ \hline 15 \end{array}$$

Thus, $\vec{\omega} = \begin{pmatrix} 1 \\ \frac{1}{2} \end{pmatrix} = \begin{pmatrix} m \\ b \end{pmatrix}$

and the best fit (in the least square sense)

is $\boxed{y = x + \frac{1}{2}}$ //

☆ Now, let's fit a polynomial of degree n to the data.

ie $y = a_0 + a_1 x^1 + \cdots + a_m x^m$

$S = \{(x_i, y_i)\}$ given.

so, we have.

$$y_1 = a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_m x_1^m$$
$$y_2 = a_0 + a_1 x_2 + a_2 x_2^2 + \cdots + a_m x_2^m$$
$$\vdots$$
$$y_n = a_0 + a_1 x_n + a_2 x_n^2 + \cdots + a_m x_n^m$$

clearly this reduces to.

$$\vec{y} = M\vec{w}$$

where $M = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^m \\ 1 & x_2 & x_2^2 & \cdots & x_2^m \\ \vdots & & & & \\ 1 & x_n & x_n^2 & \cdots & x_n^m \end{pmatrix}$ $\vec{w} = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{pmatrix}$

here $M = n \times m$ matrix.

Apply same method and find. $\vec{w} = (M^T M)^{-1} M^T \vec{y}$ done.

$E_0 = E_f$.
conservative force $f(x)$ (func. of position)

$E_0 = T_0 + U_0 = mgh$
$E_f = \frac{1}{2} m v^2$

$\Big\} \; mgh = \frac{1}{2} m v^2 \rightarrow v = \sqrt{2gh}$ at the ground.

also. recall.

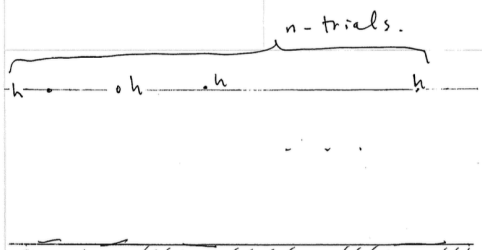
$g = a_g(t) = \frac{dv_y}{dt} \rightarrow v_y = gt + c$ (g: constant)

$dy = (gt + c)\,dt$.

but $v_y = \frac{dy}{dt}$.

$$y(t) = y_0 + v_0 t + \frac{1}{2} g t^2.$$

conduct experiment to estimate the acceleration due to gravity.



object of mass m is dropped n-times. The time & distance traveled are estimated by measure.

The following data is collected: $\{(.1, .18), (.2, .31), (.3, 1.03), (.4, 2.48), (.5, 3.73)\}$.

estimate g, the initial displacement y_0, and initial velocity v_0.

$$\begin{pmatrix} .18 \\ .31 \\ 1.03 \\ 2.48 \\ 3.73 \end{pmatrix} = \begin{pmatrix} y_0 + v_0(.1) + \frac{1}{2} g (.1)^2 \\ y_0 + v_0(.31) + \frac{1}{2} g (.31)^2 \\ y_0 + v_0(1.03) + \frac{1}{2} g (1.03)^2 \\ y_0 + v_0(2.48) + \frac{1}{2} g (2.48)^2 \\ y_0 + v_0(3.73) + \frac{1}{2} g (3.73)^2 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & (.1) & \frac{(.1)^2}{2} \\ 1 & (.31) & \frac{(.31)^2}{2} \\ 1 & (1.03) & \frac{(1.03)^2}{2} \\ 1 & (2.48) & \frac{(2.48)^2}{2} \\ 1 & (3.73) & \frac{(3.73)^2}{2} \end{pmatrix} \begin{pmatrix} y_0 \\ v_0 \\ g \end{pmatrix}$$

$\vec{y} = M\vec{w}$

find. $\vec{w} = (M^T M)^{-1} M^T \vec{y}$ // done.

how good is this estimate?

End Lecture 14

Kenneth J. Roche
University of Washington
AMATH 483 / 583 High Performance Scientific Computing
Spring Quarter 2025