Week 1 - Maths refresher 1

Sets, functions, and optimisation

the university for the real world

My housekeeping

When you'll be seeing me:

1. First two weeks: maths refresher
2. Week 7 & 8: Evolutionary game theory & iterated games

A word about the 'maths refresher' portion ...

Feedback from last semester

Student with coding background:

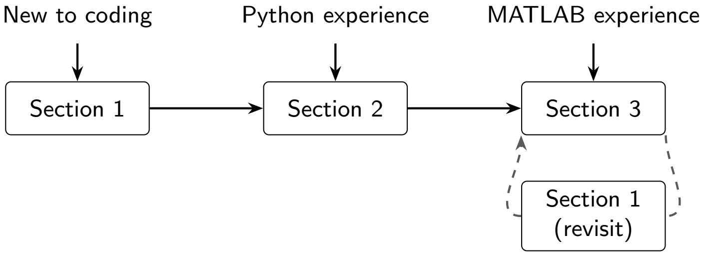
> *I think this class should be taught in python and because of matlabs, I wasn't a fan of the practicals.... Practicals fell short for me*

Student with maths/other background:

> *I found it difficult to pick up Matlab concepts, and felt the jump in difficulty was a bit extreme. More hands-on or guided experience with Matlab would be beneficial.*

Tutorials

Week 1: MATLAB



Week 2: Maths refresher

Be selective about which questions you answer.

Today's topics

1. Sets: notation and operations
2. Sets and probability
3. Stationary points and local maxima and minima
4. Critical points and global optima
5. Preview of MALTAB and some linear algebra

Set notation

$$A \cup B$$

$$E = \left\{ 2x \;\middle|\; x \in \mathbb{Z}^+, x < 5 \right\}$$

$$A \subsetneq B$$

$$a \in S$$

$$A \cap B$$

$$A \setminus B$$

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

the university for the real world

Set notation

A **set** is a collection of objects viewed as a single entity, e.g., here is the set of even positive numbers less than 10.

$$E = \{2, 4, 6, 8\}$$

A set can be empty, e.g.,

$$F = \{\ \} = \emptyset$$

The objects in the set are called **elements** or **members**

$$2 \in E, \quad 12 \notin E$$

Mnemonic: $\in$lement

Set builder notation: a recipe

$$E = \{2, 4, 6, 8\}$$

$$E = \left\{ 2x \;\mid\; x \in \mathbb{Z}^+, x < 5 \right\}$$

$$E = \{\text{recipe}\} = \{\text{output} \;\mid\; \text{rules}\}$$

Set builder notation

Filter form:

$$E = \left\{ \overset{\text{dummy variable}}{x} \in \overset{\text{generator}}{\mathbb{Z}^+} \;\underset{\text{such that}}{\mid}\; \underset{\text{guard, condition}}{x \text{ is even and } x < 10} \right\}$$

Example:

$$\mathbb{Z}^+ = \{\cancel{1}, 2, \cancel{3}, 4, \cancel{5}, 6, \cancel{7}, 8, \cancel{9}, \cancel{10, 11, 12}, \ldots\}$$

$$E = \{2, 4, 6, 8\}$$

Set builder notation

Map form:

$$E = \left\{ \overbrace{2x}^{\text{output expression}} \; \underbrace{\mid}_{\text{such that}} \; x \in \overbrace{\mathbb{Z}^+}^{\text{generator}}, \; \underbrace{x < 5}_{\text{guard, condition}} \right\}$$

Example:

$$x \in \mathbb{Z}^+, x < 5 \to \{1, 2, 3, 4\}$$

$$2x \to E = \{2, 4, 6, 8\}.$$

Subset

A is a **subset** of B if every element of A is also an element of B

implies

$$A \subseteq B \quad \text{means} \quad x \in A \implies x \in B$$

Example:

$$\{2,4\} \subseteq \{2,4,6,8\}$$

Every set is a subset of itself, and the empty set is a subset of every set:

$$\{2,4\} \subseteq \{2,4\}, \quad A \subseteq A, \quad \emptyset \subseteq A$$

$\{\}$

A is a **proper subset** of B if $A \subseteq B$ but $A \neq B$, i.e., B has at least one element not in A

$$\{2,4\} \subsetneq \{2,4,6,8\}, \quad \text{but} \quad \{2,4\} \not\subsetneq \{2,4\}$$

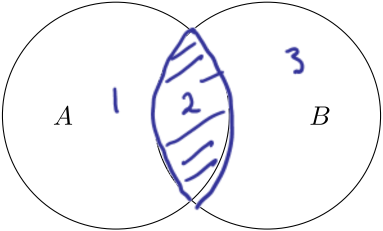
Mnemonics: $\subseteq$ is like $\leq$

$\subset$ $\quad$ $\subsetneq$ is like $<$

Intersection

The **intersection** of two sets A and B is the set of elements that belong to both A and B

$$A \cap B = \{x \mid x \in A \text{ and } x \in B\}$$



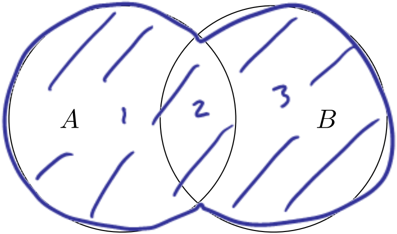
e.g., $A = \{1, 2\}, \quad B = \{2, 3\}, \quad A \cap B = \{2\}$

Mnemonic: i∩tersection, a∩d

Union

The **union** of two sets A and B is the set of elements that belong to at least one of A or B

$$A \cup B = \{x \mid x \in A \text{ or } x \in B\}$$



only 1 of each.

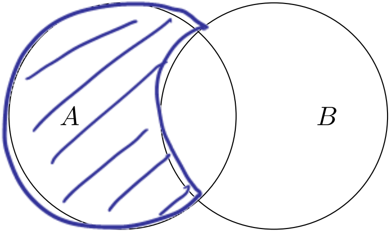
e.g., $A = \{1, 2\}, \quad B = \{2, 3\}, \quad A \cup B = \{1, 2, 3\}$

Mnemonic: $\cup$nion, $\cup$r

Set difference

The **difference** between two sets A and B is the set of all elements in A that are not in B

$$A \setminus B = \{x \mid x \in A \text{ and } x \notin B\}$$



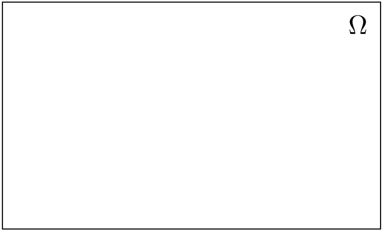
e.g., $A = \{1, 2\}, \quad B = \{2, 3\}, \quad A \setminus B = \{1\}$

Universal set

The **universal set** is the set of all possible outcomes from an experiment or scenario.

Example: Discussing the World Cup, the universal set might be the set of all countries playing

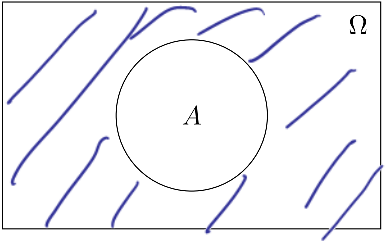
$$\Omega = \{\text{Algeria}, \text{Argentina}, \text{Australia}, \text{Austria}, \ldots\}$$



Set complement

The **complement** of a set A includes everything in the universal set that is **not** in A

$$A^c = \{x \mid x \in \Omega \text{ and } x \notin A\}$$



e.g., If A is countries who've won before, then A^c is countries who've never won before.

Sets summary

Summary:

- Set builder notation: $\{\text{output} \mid \text{rules}\}$
- Subset: $A \subseteq B \quad \text{means} \quad x \in A \implies x \in B$
- Proper subset: $A \subsetneq B \quad \text{means} \quad A \subseteq B$ but $A \neq B$
- Intersection: $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$
- Union: $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$
- Set difference: $A \setminus B = \{x \mid x \in A \text{ and } x \notin B\}$
- Set complement: $A^c = \{x \mid x \in \Omega \text{ and } x \notin A\}$

f

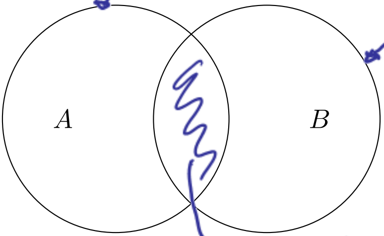
Uses in this course:

- Describing sets of players, strategies, outcomes, payoffs
- Calculating probabilities (up next)

Sets and probability: Union

Ur

Example: The probability that a team loses at home **or** it loses by at least two goals.



× double count

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

Sets and probability: Intersection

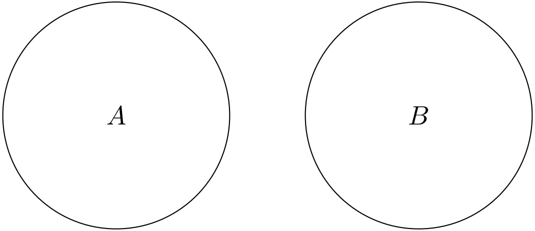
$$P(A) \cap P(B)$$

Three rules, depending on the scenario:

1. Disjoint outcomes
2. Independent outcomes
3. Dependent outcomes

Sets and probability: Intersection.
1. Disjoint

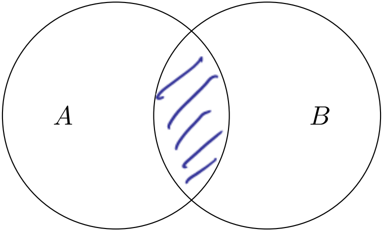
Example: The probability that a person flies to Iceland **and** they attend the World Cup.



$$A \cap B = \emptyset \quad \rightsquigarrow \quad P(A \cap B) = 0$$

Sets and probability: Intersection. 2. Independent

Example: The probability that a person's flight is delayed **and** Spain wins the world cup.



A and B independent:

$$P(A \cap B) = P(A) \cdot P(B)$$

Sets and probability: Intersection. 3. Dependent

Example: The probability that a team wins at home **and** wins by at least two goals.

Events are not independent. 'Home team advantage': Teams are advantaged at home due to familiarity, more support from the crowd, etc.

A and B dependent:

P(A given B)

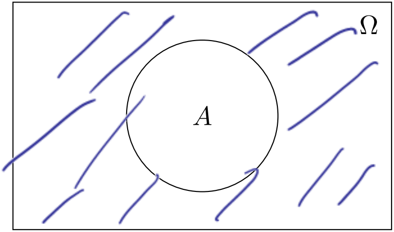
$$P(A \cap B) = P(A \mid B) \cdot P(B)$$

Rearranging:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

Sets and probability: Complement

$$P(A^c) = 1 - P(A)$$



e.g., $P(\text{non-CAF team wins}) = 1 - P(\text{CAF team wins})$

Sets and probability: Summary

- Union: $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
- Disjoint intersection: $A \cap B = \emptyset \quad \rightarrow \quad P(A \cap B) = 0$
- Independent intersection (special case): $P(A \cap B) = P(A) \cdot P(B)$
- Dependent intersection (general case): $P(A \cap B) = P(A \mid B) \cdot P(B)$
- Complement: $P(A^c) = 1 - P(A)$

f

Finding optima (maxima and minima)

A lot of game theory is about finding optima:

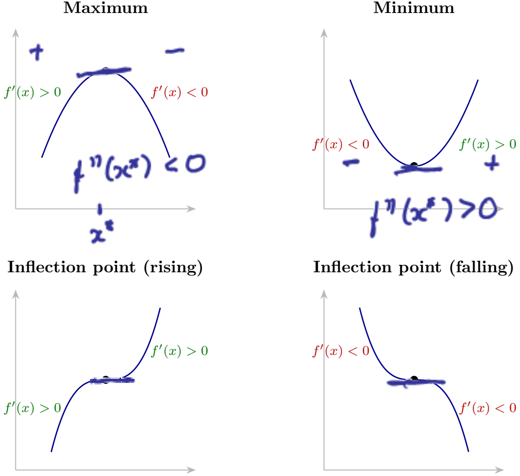
- Maximum payoff
- Maximum fitness

For example

Find the best response ...

Stationary points

Assuming f is twice differentiable.



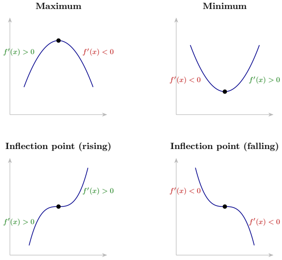
Stationary point:

slope where $f'(x) = 0$.

Type: $f''(x^*)$

Stationary points

1. Find stationary points (x^*, y^*) by solving:
$$f'(x) = 0$$
2. Classify stationary points:
   - $f''(x^*) > 0$ minimum
   - $f''(x^*) < 0$ maximum
   - $f''(x^*) = 0$ inconclusive
3. If second-derivative test inconclusive, check value of $f'(x)$ nearby $f'(x^*)$.



Step 1. Find stationary points

$$y = f(x) = x^4 - 4x^3 + 5$$

$f'(x) = 4x^3 - 12x^2 = \underbrace{4x^2}_{=0}\underbrace{(x-3)}_{=0}$

$f'(x) = 0 \implies x^* = 0 + 3$

When $x = 0$, $y = 5$

When $x = 3$, $y = -22$.

Stationary points (x^*, y^*): $(0,5) + (3,-22)$

Step 2. Second-derivative test

$$y = f(x) = x^4 - 4x^3 + 5$$

$$f'(x) = 4x^3 - 12x^2$$

(x^*, y^*): $(0, 5)$ and $(3, -22)$

$f''(x) = 12x^2 - 24x = 12x(x-2)$

$f''(0) = 0 \implies$ inconclusive

$f''(3) = 36 > 0 \implies$ minimum

Step 3 (inconclusive). Check nearby

$$y = f(x) = x^4 - 4x^3 + 5$$

$$f'(x) = 4x^3 - 12x^2$$

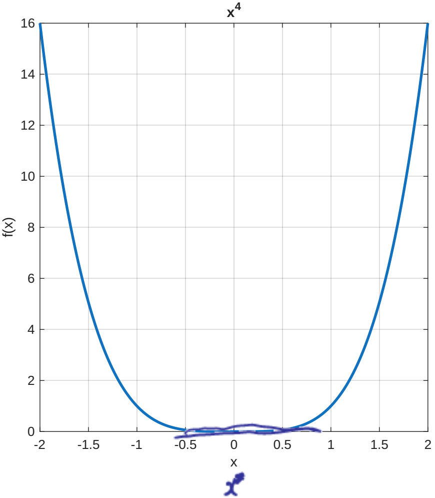
(x^*, y^*): $(0, 5)$ inconclusive; $\quad (3, -22)$ minimum

Choose test points $a < x^* < b$ close enough so no stationary points lie between a and x^* nor between x^* and b.

Chose $-1 < x^* = 0 < 1$

$f'(-1) < 0$ and $f'(1) < 0$; inflection point (falling)

Caution: $f''(x^*) = 0$ does not mean inflection



Example:

$$f(x) = x^4$$

$$f'(x) = 4x^3$$

$$f'(x) = 0 \implies x^* = 0.$$

$f''(0) = 0$ inconclusive

Stationary points: Summary

Summary:

1. Find stationary points (x^*, y^*) by solving:

$$f'(x) = 0$$

2. Classify stationary points:
   - $f''(x^*) > 0$ minimum
   - $f''(x^*) < 0$ maximum
   - $f''(x^*) = 0$ inconclusive
3. If second-derivative test inconclusive, check value of $f'(x)$ nearby $f'(x^*)$.

The above finds *local* maxima and minima. But we're often interested in the *global* optima...

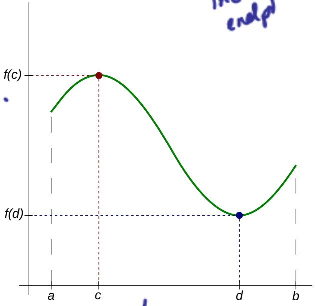
Global optima and the Extreme Value Theorem

If f is *continuous* on a domain that is *closed* and *bounded*, then f attains a global maximum and a global minimum — both are reached at actual points.

incl endpt

no ∞

range.



domain

	cld	bnd
$[a, b]$	✓	✓
$[a, b)$	✗	✓
$[a, \infty)$	✓	✗
(a, ∞)	✗	✗

A continuous function $f(x)$ on the closed interval $[a, b]$ showing the absolute max (red) and the absolute min (blue) (pbroks13, Public Domain).

Global optima: check the endpoints

To find global optima, we must check the endpoints as well as any interior stationary points.

$$y = f(x) = x^4 - 8x^2 \quad \text{on } [-1,\ 3]$$

$$f'(x) = 4x^3 - 16x = 4x(x-2)(x+2) = 0 \implies x^* = -2, 0, 2$$

x		$f(x)$
-1	endpoint	-7
0	interior stat. pt.	0
2	interior stat. pt.	-16
3	endpoint	9

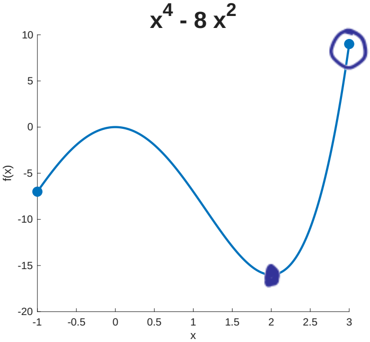
Global optima: check the endpoints

$$y = f(x) = x^4 - 8x^2 \quad \text{on } [-1,\ 3]$$

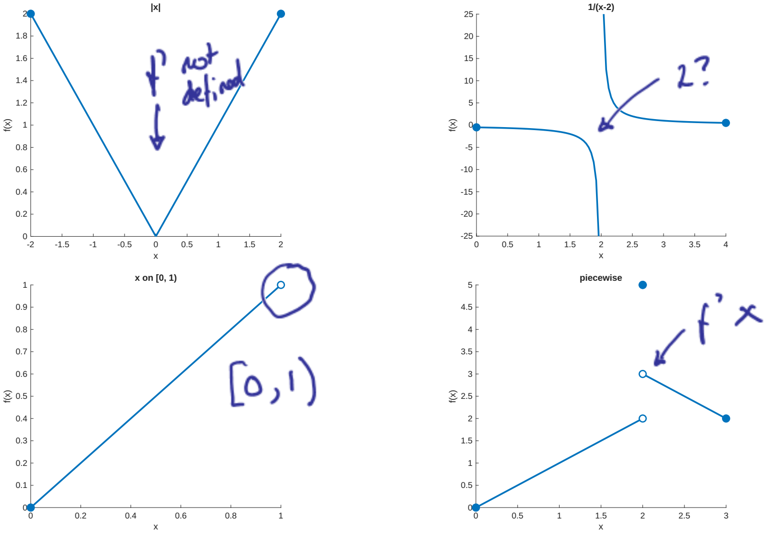
x		$f(x)$
-1	endpoint	-7
0	interior stat. pt.	0
2	interior stat. pt.	-16
3	endpoint	9

$\Longrightarrow$ Global min: (2, -16)

$\Longrightarrow$ Global max: (3, 9)



Complications

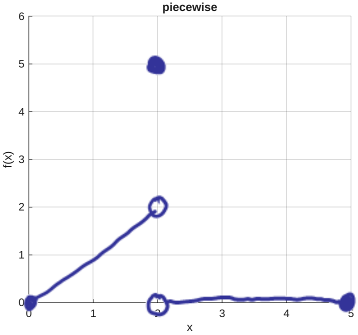


Piecewise functions

A **piecewise** function gives a different rule on each part of its domain.

Example

$$f(x) = \begin{cases} x & 0 \leq x < 2 \\ 5 & x = 2 \\ 0 & 2 < x \leq 5. \end{cases}$$



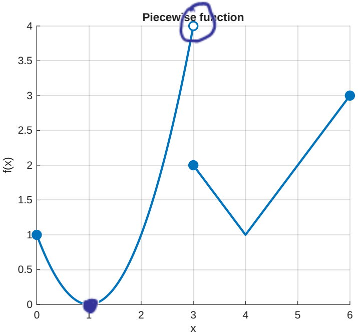
Global optima example

$$f(x) = \begin{cases} (x-1)^2 & 0 \le x < 3 \\ |x-4|+1 & 3 \le x \le 6 \end{cases} \quad \text{on } [0,6].$$

Global minimum at $(1,0)$

$\sup f = 4$ approaching $x = 3$ from left, but not attained

Therefore: No global maximum



Global optima recipe

1. Find suspect points c:
   - stationary points, endpoints, interior points where f is undefined or jumps, and continuous points where f' doesn't exist.
2. Tabulate them and calculate:
   - limit from left, limit from right, value at c
3. Use table to identify infimum and supremum
   - infimum obtained $\implies$ global minimum
   - supremum obtained $\implies$ global maximum

Global optima example

$$f(x) = \begin{cases} (x-1)^2 & 0 \le x < 3 \\ |x-4|+1 & 3 \le x \le 6 \end{cases} \qquad \text{on } [0,6].$$

Suspect list:

1. Stationary points:
   - $f' = 2(x-1)$ on piece A $\implies x^* = 1$;
     $f''(1) = 2 > 0 \implies$ local minimum
   - on piece B the slope is ± 1, never zero.
2. Endpoints: $x = 0, 6$
3. Seam at $x = 3$
4. Corner from $|x-4|$ at $x = 4$.

Global optima example

$$f(x) = \begin{cases} (x-1)^2 & 0 \leq x < 3 \\ |x-4|+1 & 3 \leq x \leq 6 \end{cases} \quad \text{on } [0,6]$$

$$f(c^-) := \lim_{x \to c^-} f(x), \quad f(c^+) := \lim_{x \to c^+} f(x)$$

Point c	Type	$f(c^-)$ left	$f(c)$	$f(c^+)$ right
0	endpt	—	1	1
1	stat'y	0	0	0
3	seam	4	2	2
4	corner	1	1	1
6	endpt	3	3	—

Global optima example

Are supremum and infimum obtained?

attained?

Point c	Type	$f(c^-)$	$f(c)$	$f(c^+)$
0	endpoint	—	1	1
1	stat. local min	0	0	0
3	jump	4	2	2
4	corner	1	1	1
6	endpoint	3	3	—

Scan the whole table for the biggest and smallest numbers anywhere, then look up whether that number lives in the middle column.

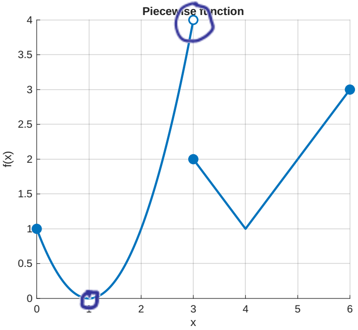
- $\inf f =$ 0 Attained => global min
- $\sup f =$ 4. Not attained => no global max

Global optima example

$$f(x) = \begin{cases} (x-1)^2 & 0 \le x < 3 \\ |x-4|+1 & 3 \le x \le 6 \end{cases} \quad \text{on } [0,6]$$

c	Type	$f(c^-)$	$f(c)$	$f(c^+)$
0	endpt	—	1	1
1	stat'y	0	**0**	0
3	seam	**4**	2	2
4	corner	1	1	1
6	endpt	3	3	—

Global min: (1, 0)
Global max: none



Step 1: Find suspect points

1. Stationary points $f'(x) = 0$
2. Endpoints of the domain
   - even when excluded, e.g. $(0, 3)$.
   - if the domain is unbounded, the ends $x \to \pm\infty$
3. Interior points where f is undefined or jumps
   3.1 denominator hits zero or f otherwise undefined
      - e.g., $\sqrt{x}$ at $x < 0$, $\log(x)$ at $x \leq 0$

   3.2 removed point / jump discontinuity
4. Continuous points where f' does not exist:
   4.1 inside of an absolute value hits zero: solve $g(x) = 0$ in $|g(x)|$
   4.2 piecewise seam
   4.3 min/max crossover: solve $f(x) = g(x)$ in $\min(f, g)$ or $\max(f, g)$

Step 2: Are infimum and supremum obtained?

1. For each suspect $x = c$, tabulate:

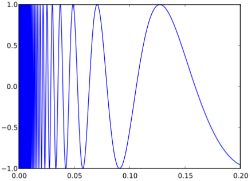
$$f(c^-) := \lim_{x \to c^-} f(x), \quad f(c), \quad f(c^+) := \lim_{x \to c^+} f(x)$$

   Any of these may be $\pm\infty$, or undefined if $c \notin \operatorname{dom} f$.

2. Find the infimum and supremum
   - smallest and largest values in the table
3. The supremum is attained only if it appears in the $f(c)$ column. Same for the infimum.
   - supremum obtained $\implies$ global maximum
   - infimum obtained $\implies$ global minimum

Why it works

Between two consecutive suspects, f is continuous and differentiable with $f' \neq 0$. By Darboux, f' can't change sign without vanishing, so f is strictly monotone there (entirely non-decreasing or non-increasing). A monotone function has no interior extremum; its extremes sit at the ends of the piece, which are suspects. So the suspect list plus one-sided limits genuinely captures everything.

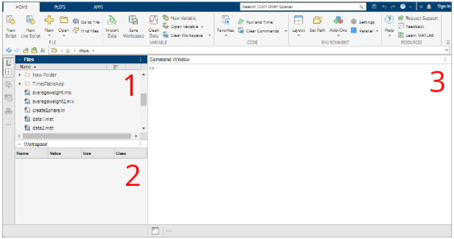


Note: this needs *finitely many* suspects, e.g., $\sin(1/x)$ breaks it

f

MATLAB stands for Matrix Laboratory

Workshops are in MATLAB.



MATLAB was originally designed as a matrix calculator to teach linear algebra.

$$A = \begin{bmatrix} 1 & 3 & 2 \\ 2 & 2 & 2 \\ 3 & 5 & 6 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -3 \\ -2 \\ -5 \end{bmatrix}$$

Example. Solving systems of linear equations

Solve a system of equations

$$\begin{aligned} x + 3y + 2z &= -3 \\ 2x + 2y + 2z &= -2 \\ 3x + 5y + 6z &= -5 \end{aligned}$$

System can be written in matrix form

$$\underbrace{\begin{bmatrix} 1 & 3 & 2 \\ 2 & 2 & 2 \\ 3 & 5 & 6 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x \\ y \\ z \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} -3 \\ -2 \\ -5 \end{bmatrix}}_{\mathbf{b}}$$

Example. Solving systems of linear equations

Solve by simultaneous equations or Gaussian elimination.

$$[A \mid b] = \left[\begin{array}{ccc|c} 1 & 3 & 2 & -3 \\ 2 & 2 & 2 & -2 \\ 3 & 5 & 6 & -5 \end{array}\right]$$

$$\left[\begin{array}{ccc|c} 1 & 3 & 2 & -3 \\ 0 & -4 & -2 & 4 \\ 0 & -4 & 0 & 4 \end{array}\right] \begin{array}{l} \\ R_2 \leftarrow R_2 - 2R_1 \\ R_3 \leftarrow R_3 - 3R_1 \end{array}$$

$$\left[\begin{array}{ccc|c} 1 & 3 & 2 & -3 \\ 0 & -4 & -2 & 4 \\ 0 & 0 & 2 & 0 \end{array}\right] \begin{array}{l} \\ \\ R_3 \leftarrow R_3 - R_2 \end{array}$$

f

Example. Solving systems of linear equations

$$
\begin{array}{c}
\begin{array}{ccc|c} & \text{LHS} & & \text{RHS} \end{array} \\
\left[\begin{array}{ccc|c} 1 & 3 & 2 & -3 \\ 0 & -4 & -2 & 4 \\ 0 & 0 & 2 & 0 \end{array}\right]
\end{array}
\longrightarrow
\begin{array}{rcrcrcr}
x & + & 3y & + & 2z & = & -3 \\
 & - & 4y & - & 2z & = & 4 \\
 & & & & 2z & = & 0
\end{array}
$$

Solve by back substitution:

$$2z = 0 \implies z = 0;$$

$$-4y - 2(0) = 4 \implies y = -1;$$

$$x + 3(-1) + 2(0) = -3 \implies x = 0$$

$$\boxed{(x, y, z) = (0, \,-1, \,0)}$$

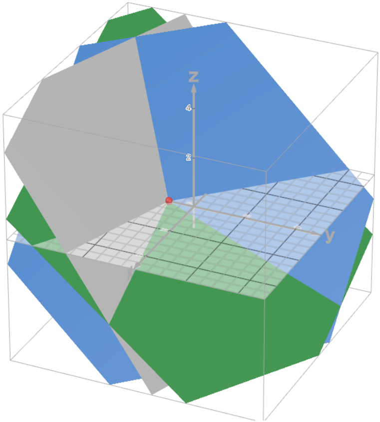
Example. Solving systems of linear equations

System of equations:

$$
\begin{aligned}
x + 3y + 2z &= -3 \\
2x + 2y + 2z &= -2 \\
3x + 5y + 6z &= -5
\end{aligned}
$$

Solution:

$$(x, y, z) = (0,\ -1,\ 0)$$



https://www.desmos.com/3d/mh1sl6ipq6

MATLAB stands for ‘Matrix Laboratory’

System of equations:

$$
\begin{aligned}
x + 3y + 2z &= -3\\
2x + 2y + 2z &= -2\\
3x + 5y + 6z &= -5
\end{aligned}
$$

In matrix form:

$$
\underbrace{\begin{bmatrix}1 & 3 & 2\\ 2 & 2 & 2\\ 3 & 5 & 6\end{bmatrix}}_{A}
\underbrace{\begin{bmatrix}x\\ y\\ z\end{bmatrix}}_{\mathbf{x}}
=
\underbrace{\begin{bmatrix}-3\\ -2\\ -5\end{bmatrix}}_{\mathbf{b}}
$$

1st 2nd 3rd .

```
>> A = [1 3 2; 2 2 2; 3 5 6];
>> b = [-3; -2; -5];
>> x = A \ b                   % or use: x = inv(A) * b
x =
     0
    -1
     0
```

Matrix operations

1. Addition

```
>> [1 2; 3 4] + [5 6; 7 8]
ans =
     6     8
    10    12
```

2. Transpose flips rows to columns `A'`

A^T

```
>> [1 2; 3 4]'
ans =
     1     3
     2     4
```

3. Inverse `inv(A)`

$$A\mathbf{x} = \mathbf{b} \implies \mathbf{x} = A^{-1}\mathbf{b}$$

Matrix operations

1. $A + B = B + A$ addition commutative
2. $A(BC) = (AB)C$ multiplication associative
3. $A(B + C) = AB + AC$ multiplication distributive
4. $AB \neq BA$ multiplication not commutative

Example: Matrix multiplication

$$\begin{array}{c c} & \\ 1 \rightarrow & \\ 2 \rightarrow & \end{array} A = \begin{bmatrix} 1 & 0 & 2 \\ 2 & 1 & -1 \end{bmatrix}_{2\times 3} \qquad B = \begin{array}{c} \begin{array}{cc} 1 & 2 \\ \downarrow & \downarrow \end{array} \\ \begin{bmatrix} 1 & 2 \\ 0 & 1 \\ 3 & -1 \end{bmatrix}_{3\times 2} \end{array}$$

$$AB = \begin{bmatrix} \overset{(1,1)}{\overset{1}{(1)(1)} + \overset{0}{(0)(0)} + \overset{6}{(2)(3)}} & , & \overset{1,2}{\overset{2}{(1)(2)} + \overset{0}{(0)(1)} + \overset{-2}{(2)(-1)}} \\ \overset{2}{(2)(1)} + \overset{0}{(1)(0)} + \overset{-3}{(-1)(3)} & , & \overset{4}{(2)(2)} + \overset{1}{(1)(1)} + \overset{1}{(-1)(-1)} \end{bmatrix}_{2,2}$$

$$AB = \begin{bmatrix} 7 & 0 \\ -1 & 6 \end{bmatrix}$$

Example: Matrix multiplication

$$A = \begin{bmatrix} 1 & 0 & 2 \\ 2 & 1 & -1 \end{bmatrix}_{2\times 3} \qquad B = \begin{bmatrix} 1 & 2 \\ 0 & 1 \\ 3 & -1 \end{bmatrix}_{3\times 2} \qquad AB = \begin{bmatrix} 7 & 0 \\ -1 & 6 \end{bmatrix}$$

$$BA = \begin{bmatrix} 5 & 2 & 0 \\ 2 & 1 & -1 \\ 1 & -1 & 7 \end{bmatrix}$$

Example: Matrix multiplication

$$A = \begin{bmatrix} 1 & 0 & 2 \\ 2 & 1 & -1 \end{bmatrix}_{2\times 3} \qquad B = \begin{bmatrix} 1 & 2 \\ 0 & 1 \\ 3 & -1 \end{bmatrix}_{3\times 2} \qquad AB = \begin{bmatrix} 7 & 0 \\ -1 & 6 \end{bmatrix}$$

$$\begin{aligned} BA &= \begin{bmatrix} (1)(1)+(2)(2) & (1)(0)+(2)(1) & (1)(2)+(2)(-1) \\ (0)(1)+(1)(2) & (0)(0)+(1)(1) & (0)(2)+(1)(-1) \\ (3)(1)+(-1)(2) & (3)(0)+(-1)(1) & (3)(2)+(-1)(-1) \end{bmatrix} \\ &= \begin{bmatrix} 5 & 2 & 0 \\ 2 & 1 & -1 \\ 1 & -1 & 7 \end{bmatrix} \end{aligned}$$

Matrix multiplication summary

$$\mathbf{A}_{m\times\mathbf{n}}\,\mathbf{B}_{\mathbf{n}\times p} = \begin{bmatrix} \text{——— row } \mathbf{1} \text{ ———} \\ \text{——— row } \mathbf{2} \text{ ———} \\ \text{——— row } \mathbf{3} \text{ ———} \\ \vdots \end{bmatrix}_{m\times\mathbf{n}} \begin{bmatrix} | & | & \\ \text{col } \mathbf{1} & \text{col } \mathbf{2} & \cdots \\ | & | & \end{bmatrix}_{\mathbf{n}\times p}$$

match.

$$= \begin{bmatrix} \text{row } \mathbf{1} \cdot \text{col } \mathbf{1} & \text{row } \mathbf{1} \cdot \text{col } \mathbf{2} & \cdots \\ \text{row } \mathbf{2} \cdot \text{col } \mathbf{1} & \text{row } \mathbf{2} \cdot \text{col } \mathbf{2} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}_{m\times p}$$

dot prod. $\rightarrow$ $$\text{row } i \cdot \text{col } j = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj}$$

Linear programming example

A factory makes widgets and cogs. For each product, they earn the following profits:

$$\$30/\text{widget and } \$40/\text{cog}$$

Three machines are needed to make widgets and cogs

Hours per machine:

	machine hrs		
	A	B	C
widgets	1	1	3
cogs	2	1	2

Machine availability:

machine	hrs avail. / wk
A	24
B	16
C	44

Linear programming example

Profits: \$30/widget and \$40/cog

	machine hrs A	B	C
widgets	1	1	3
cogs	2	1	2
	24	16	44

machine	hrs avail. / wk
A	24
B	16
C	44

$$
\begin{array}{lrcrcr}
\text{maximize:} & 30w & + & 40c & \leftarrow \text{profit.} & \\
\text{subject to:} & w & + & 2c & \leq & 24 \\
 & w & + & c & \leq & 16 \\
 & 3w & + & 2c & \leq & 44
\end{array}
$$

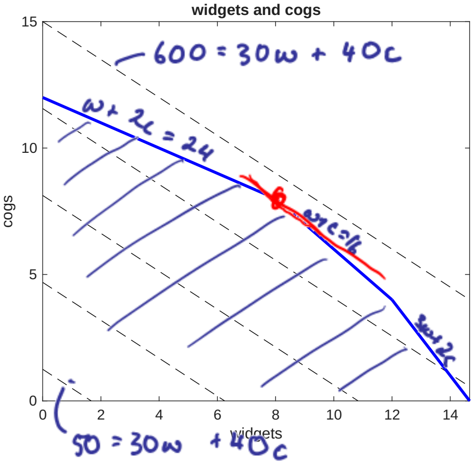
Linear programming example

maximise:

profit $$30w \quad + \quad 40c$$

subject to:

$$\begin{array}{rcrcr} w & + & 2c & \leq & 24 \\ w & + & c & \leq & 16 \\ 3w & + & 2c & \leq & 44 \end{array}$$



Linear programming example

MATLAB can solve linear programming problems:

```
>> help linprog
 linprog - Solve linear programming problems
    Linear programming solver

    Syntax
      x = linprog(f,A,b)
```

However, the input needs to be given in matrix form

$$\begin{aligned} \text{minimise:} \quad & \mathbf{f}^T\mathbf{x} \\ \text{subject to:} \quad & A\mathbf{x} \leq \mathbf{b} \end{aligned}$$

Linear programming example

$$\begin{array}{rl} \text{maximize:} & 30w + 40c \\ \text{subject to:} & 1w + 2c \le 24 \\ & 1w + 1c \le 16 \\ & 3w + 2c \le 44 \end{array} \quad \rightarrow \quad \begin{array}{rl} \text{minimise:} & \mathbf{f}^T\mathbf{x} \\ \text{subject to:} & A\mathbf{x} \le \mathbf{b} \end{array}$$

Turn maximisation into minimisation problem:

$$\text{maximize: } 30w + 40c \longrightarrow \text{minimise: } \underbrace{-30}_{f_1} w \underbrace{-40}_{f_2} c$$

minimise:

$$\underbrace{\begin{array}{cc} f_1 & f_2 \\ \left[-30\right. & \left.-40\right] \end{array}}_{\mathbf{f}^T} \underbrace{\begin{bmatrix} w \\ c \end{bmatrix}}_{\mathbf{x}}$$

$$f_1 w + f_2 c$$

subject to:

$$\underbrace{\begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 3 & 2 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} w \\ c \end{bmatrix}}_{\mathbf{x}} \le \underbrace{\begin{bmatrix} 24 \\ 16 \\ 44 \end{bmatrix}}_{b}$$

Linear programming example

minimise:

$$\underbrace{\begin{bmatrix} -30 & -40 \end{bmatrix}}_{\mathbf{f}^T} \underbrace{\begin{bmatrix} w \\ c \end{bmatrix}}_{\mathbf{x}}$$

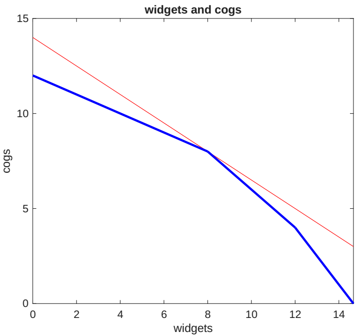
subject to:

$$\underbrace{\begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 3 & 2 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} w \\ c \end{bmatrix}}_{\mathbf{x}} \leq \underbrace{\begin{bmatrix} 24 \\ 16 \\ 44 \end{bmatrix}}_{b}$$

```
>> f = [-30; -40];
>> A = [1 2; 1 1; 3 2];
>> b = [24; 16; 44];
>> linprog(f, A, b)
Optimal solution found.
ans =
     8
     8
```

Linear programming example

```
>> f = [-30; -40];
>> A = [1 2; 1 1; 3 2];
>> b = [24; 16; 44];
>> linprog(f, A, b)
Optimal solution found.
ans =
     8
     8
```



f

Linear programming

```
>> help linprog

    Syntax
      x = linprog(f,A,b)
      x = linprog(f,A,b,Aeq,beq)
      x = linprog(f,A,b,Aeq,beq,lb,ub)

    Input Arguments
      f - Coefficient vector
      A - Linear inequality constraints
      Aeq - Linear equality constraints
      b - Linear inequality constraints
      beq - Linear equality constraints
      lb - Lower bounds
      ub - Upper bounds
```

Questions and feedback

Suggestion: STOP, START, CONTINUE

Break

Turn to the person next to you (in person or on Zoom) and ask these questions:

1. What's your name and what course are you doing?
2. What's your favourite movie?
3. If you had to teach a 20 min class on something you're weirdly good at, what would it be?

(There will be an exam on your friends' answers in Week 7)