1) Definitions & Properties

Linear Dependence - If we have pivot vars, we are linearly independent, else we're not.
For a matrix R^{nm} n = rows, m = cols. Rows are across the matrix, cols are down the matrix.

1.1) Determinants: For A, B $\in R^{n\times n}$
1) $\det(A^T) = \det(A)$
2) $\det(AB) = \det(BA) = \det(A)\det(B)$. $\det(kA) = k^n\det(A)$
$\det(A) = 0 \Leftrightarrow$ No Inverse.
Else, $\det(A^{-1}) = 1/\det(A)$.
Dets only exist for Sq Matrices, we get them by getting RREF and multiplying by lead diag. RREF ops have effects on the det:
1) Swapping row multiplies it by -1
2) Adding/subbing rows does nothing
3) If any two rows are equal, or lead diag = 0, then det = 0.
4) Multiplying by scalar also increases det by a scalar.

Trace - sum of diag elems.
Rank of invertible matrices - nxn matrix is invertible ⇔ rank is n ⇔ cols are lin indep, rows too
Singular - A square matrix is non-singular if the columns are linearly indep, ie: if rk(A) = n, or det(A) =/= 0. Else its singular.
Vector Space - a set U =/= ∅ is a vector space if U is closed under addition and scalar multiplication: 1) For all u, v ∈ U, u + v ∈ U
2) For all u ∈ U and c ∈ R, cu ∈ U
Vector Subspace: subset of a Vector Space
Generating set: Our vector subspace X is a generating set of U it could express every vector in U in the Vector Space as a linear combo of its vectors.
A **basis** is a minimal generating set. A **simple basis** is one with as many 0s, as possible, gotten by getting RREF and taking the pivot columns span as our simple basis.
Dimension = num basis vectors.
M1) Finding Basis: get the REF, and take original vector of each pivot column. Span of this = basis.
M2) Change of Basis Matrix:
We just represent each basis vector in terms of the other basis, and each representation is one of our columns.
Euclidean norm: magnitude of vector: root of sum of squares.
Parallelogram Law: $\forall u, v \in R^n$, $||u+v||^2+||u-v||^2=2||u||^2+2||v||^2$
Angle between vectors
cos(x) = u.v / ||u|| ||v||

2) Orthogonality
Take vectors u,v **Orthogonal**: u.v = 0.
Orthonormal: orthogonal ∧ u &v's magnitudes are 1.
Orthogonal Matrix: $A \in R^{n\times n}$ is invertible with $A^{-1} = A^T$
Orthogonal Subspaces: (aka: U⊥V)
For ∀u ∈ U, ∀v ∈ V, u·v = 0.
Cauchy Schwarz inequality:
$u, v \in R^n$, $|u \cdot v| \leq ||u||\,||v||$
Triangle inequality:
$\forall u, v \in R^n$, $||u + v|| \leq ||u|| + ||k||$
$u \perp v \Leftrightarrow ||u + v||^2 = ||u||^2 + ||v||^2$

3) Linear Maps
A map from one vector subspace to another is linear if:
For all u, v ∈ V , f(u + v) = f(u) + f(v)
For all u ∈ V and c ∈ R, f(cu) = cf(u)
A basis change matrix from u to v is a linear map. Example: (map from 2 dimensions to 3)
$f(x)=\begin{pmatrix} y\\ x\\ y+x\end{pmatrix}=\begin{bmatrix}0&1\\1&0\\1&1\end{bmatrix}\begin{pmatrix}x\\y\end{pmatrix}$
We can compose basis change and linear maps.
For an f: $R^n \to R^m$ linear map, the **Image Space**: refers to the set of points mapped to in R^m.
The **Kernel/Null** space is the set of points in the N space that is 0.
Rank Nullity Theorem: For linear map f:$R^n \to R^m$ with Mat $A \in R^{mn}$: dim(im(f)) + dim(ker(f)) = rk(A) + dim(ker(A)) = n

4) Eigenvalues and Eigenvectors
Eigenvalues = solutions to the Char. Polynomial (CP) (A-λI).
Spectrum: set of EigVals of a Mat. Tr(A) = sum of eigenvalues.
det(A) = product of eigenvalues. Algebraic Multiplicity (AM) of an eigenvalue = num times solves CP. Geometric Multiplicity (GM) = dim of the eigenspace of eigenvalue.
1 <= AM <= GM <= Dim of Matrix.
Eigenspace of an eigenvalue = span of its eigenvectors.
Similar Matrices: Two matrices A and B are similar if there is an matrix P such that $A = PBP^{-1}$. **Cayley Hamilton Theorem**: Take CP of A; take A as lambda, we get the 0 matrix. We find inverses this way: $A =\begin{pmatrix}1 & -1\\ 2 & 1\end{pmatrix}$CP: $x^2-2x+3=0$. Then: $-A^2+2A=3I \Leftrightarrow A*1/3(-A+2I) = I$
$A^{-1}=1/3(-A+2I) = I$. So $A^{-1}=1/3(2I-A)$

Transpose - Write the rows as columns. $(AB)^T = B^TA^T$

Computational Techniques

1) Vector Norms
Computing size on a vector.
Properties For vector x, y in Rn, scalar k
1) ||x|| > 0. (for non-zero x)
2) ||kx|| = |k| ||x||
3) ||x+y|| <= ||x|| + ||y||
l_p norm is $||v||_p =(\sum_{i=1}^n |v_i|^p)^{1/p}$
L_1 norm = Sum of Vector Elems.
L_2 = Euclidean Norm
L_∞ = max vector entry.
Any two vector norms $||.||_a$ and $||.||_b$ in R^n are equivalent. Formally:
We have r > 0 and s > 0 s.t
$\forall x \in R^n$, $r||x||_a \leq ||x||_b \leq s||x||_a$
L1, L2, L∞ are **equivalent** as below: 1) $||x||_\infty \leq ||x||_2 \leq ||x||_1$
2) $||x||_2 \leq \sqrt{n}||x||_\infty$
3) $||x||_1 \leq \sqrt{n}||x||_2$

2) Matrix Norms (MN):
1) ||A|| > 0   2) ||kA|| = ||k|| ||A||
3) ||A + B|| <= ||A|| + ||B||,
In this course we only use Sub-multiplicative MNs: which have
4) ||AB|| <= ||A|| ||B||
Computing the Norms:
L_1= max absolute column sum (add up abs values of each entry) L_2=Largest singular value of A. L3= Maximum absolute row sum.
Consistent: A Matrix Norm is consistent with the vector norms $||.||_a$ and $||.||_b$ if for all A^{mn} and x^n $||Ax||_b \leq ||A||\,||x||_a$. If a = b, they're compatible.
Subordinate Matrix Norm: For R^n vector norm ||.||, the subordinate matrix norm ||A|| R^{mn} is: $\forall A\in R^{mn}$ $||A||=\max\{||Ax||:x\in R^n,||x||=1\}$
A vector norm is compatible with its subordinate matrix norm AKA: for all A^{mn} and x^n, $||Ax|| \leq ||A||\,||x||$. For p = 1, 2 and infinity the MN is subordinate to the vector norm. Thus the matrix norm is also compatible with the vector norm.

3) Complex Linear Maps
Complex Norm * Conjugate > 0. **Complex Conjugate of a Matrix**: Compute the conjugate of each element (if complex).
If a vector, scalar or matrix = its conjugate, then its real.
Standard Inner Product <u, v>: Of two vectors u, v $\in C^n = \bar{u}^T v$.
The inner product is **conjugate** symmetric: $\langle u, v\rangle = \overline{\langle u, v\rangle}$
Standard Norm on u: $\sqrt{\bar{u}^T u}$
SIP > 0; if it is 0 then u = 0. **Complex Eigenvalues and Eigenvectors**: The eigenvalues of a real matrix can be complex. This means our eigenvectors are complex too.

4) Least Squares Method
Endomorphism: A linear map where the domain and codomain are the same f: $R^n \to R^n$
Automorphism: bijective endomorphism.
Automorphism
f: $R^n \to R^n \Leftrightarrow$ f injective (ker(f)={0}) ⟺ f surjective (im(f) = R^n)
Projection of a Subspace:
Let $U \subseteq R^n$ be an n dimensional subspace generated by an ordered basis $(u_1,...,u_n)$. Let U = $[U_1,...,U_n]$
The **orthogonal projection** Π_U on U is the following endomorphism:
$\Pi_U: R^m \to R^m$
$v \to \Pi_U(v) = U(U^TU)^{-1}U^Tv$
im(A) ⊥ ker(A^T)
Unique Vector Decomposition: Let $A \in R^{mn}$. For all vectors $b \in R^m$, there exists unique $b_i \in$ im(A), and unique $b_k \in$ ker(A^T) such that: $b = b_i + b_k$.
Let $A \in R^{mn}$ and $b \in R^n$. Suppose Ax = b has no solution for $x \in R^n$, i.e., b ∉ im(A).
LSM finds $x \in R^n$ such that $||Ax - b||_2$, or equivalently $||Ax - b||_2^2$ is minimised. With $||Ax-b_i||_2$ minimised when $||Ax-b_i||_2 = 0 \Leftrightarrow Ax = b_i$.

5) Linear Regression
We have a set of points (y_i, a_i), y a real number, and a is a real vector of dimension n.
We want to find the model of best fit with parameters $s_0 \in R$ and $s \in R^n$, so the sum of the errors squared is minimized:
$$\sum_{i=1}^m (s_0 + s*a_i - y_i)^2$$
We require $s_0+s \cdot a_i \approx y_i$
The sum of the squared errors is this:
$\sum_{i=1}^m(s_0 + s*a_i - y_i)^2 = ||Az - b||_2^2$

6) Spectral Decomposition of Symmetric Matrices
Properties of Orthogonal Matrices:
1) *Orthogonal Matrix transformations preserve Euclidean length of vectors:*
$\forall u \in R^n$, $||Qu||_2 = ||u||_2$
2) *Orthogonal transformations preserve the magnitude of the angle between vectors:*
$\forall u, v \in R^n$, $\overline{Qu}\cdot\overline{Qv} = \overline{u}\cdot\overline{v}$
3) The transformations performed by an orthogonal matrix can be interpreted as a change of basis or a series of rotations and reflections.
4) det = 1 or -1.
5) All eigenvalues have modulus 1

Properties of Symmetric Matrices:
-Defined by $A^T = A$.
1) If A is a real symmetric matrix, then all its Eigenvalues are real.
2) If A is a real symmetric matrix, then for each Eigenvalue the algebraic multiplicity and geometric multiplicity are equal.
3) If A is an n by n real symmetric matrix, eigenvectors for distinct eigenvalues are orthogonal.

6.1) Spectral Theorem
If A is a real, symmetric matrix then it can be diagonalised like so:
$A = QDQ^T = QDQ^{-1}$
where: Q is an **orthogonal** matrix
D is the **diagonal** eigenvalue matrix.

7) Singular Value Decomposition
A is real symmetric matrix.
Positive Definite: A is positive definite iff: $\forall x \in R^n - \{0\}$, $x^TAx > 0$.
Positive Semi-Definite iff:
$\forall x \in R^n - \{0\}$, $x^TAx \geq 0$.
Theorem: Positive definiteness in terms of Eigenvalues:
1) Positive definite ⇔ all eigenvalues are strictly positive
2) Positive semi definite ⇔ eigenvalues are non negative
Properties of A^TA and AA^T
Take arbitrary $A \in R^{mn}$
1) $A^TA \in R^{nn}$ and $AA^T \in R^{mm}$ are both **symmetric** and **positive-semi definite.**
This gives rise to SVD - a more general decomposition than spectral, which takes more work to get into but still exhibits useful properties.

8) SVD Definition:
Take arbitrary $A \in R^{mn}$. SVD of A is any decomposition of the form:
$A = USV^T$
$U \in R^{mm}$ and $V \in R^{nn}$ are orthogonal.
$S \in R^{mn}$ is a diagonal matrix, S = diag($\sigma_1, \sigma_2, \sigma_3, ... \sigma_p$) where p = min(m, n).
$\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_p \geq 0$ **(We write largest first)**
The values $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_p$, are known as the singular values of A.
Useful Properties of SVD:
1) SVD: $A = USV^T$.
If U = $[u_1,...,u_m]$ and V = $[v_1, ... , v_n]$, then
$A = \sigma_1u_1v_1^T + \sigma_2u_2v_2^T + \cdots + \sigma_ru_rv_r^T$.
2) **rank of A = num singular positive values in S.**
3) For $A \in R^{mn}$ $||A||_2$ = largest singular value of A.
The positive singular values of A = positive square roots of the eigenvalues of AA^T or A^TA.
5) The span of the first r columns of U = im(A). The span of the last m-r columns is ker(A), where r is the rank.
6) **The Principal axes and first principal axis of a collection of samples**: Assume $A \in R^{mn}$ represents m samples of n dimensional data. The principal axes of A are the columns of V, i.e., v_i for 1 ≤ i ≤ r. The first (principal) axis of the collection of m samples is defined as: $w(1) = \underset{||w||=1}{\text{argmax}}\, w^TA^TAw$
7) **The principal axes of a Singular Value Decomposition** Assume $A \in R^{mn}$ represents m samples of n dimensional data. Assume we have the following singular value decomposition for A: $A = USV^T$, where V = $[v_1, ... v_n]$ Then, v_1 is the first (principal) axis of A. It corresponds to the column vector of V associated with the largest singular value of A.
8) **The principal components (scores of a selection of samples)**
The columns of US, i.e., σ_iu_i for 1 ≤ i ≤ r, are the principal components or scores of A.

9) Generalised Eigenvectors
Take a square matrix $A \in R^{nn}$.
Generalised Eigenvector: A nonzero vector $V \in C^n$ **of rank m** associated with eigenvalue λ when $(A - \lambda I)^m v = 0$ and $(A - \lambda I)^{m-1} v$ =/= 0
Any eigenvector associated with its eigenvalue is a generalised eigenvector of size 1.
An eigenvector of A associated with λ and multiplied by the matrix A - λI, results in the zero vector.
If A has a generalized eigenvector of rank 1 associated with λ, then when you multiply it by A - λI, the result will be another eigenvector associated with λ.
If A has a generalized eigenvector of rank 2 associated with λ, then when you multiply it by A - λI, the result will be a linear combination of a rank 1 generalized eigenvector and another eigenvector associated with λ. This pattern continues for higher-rank generalized eigenvectors. Example:
Take $A = \begin{bmatrix}1&1&1\\0&1&1\\0&0&1\end{bmatrix}$ CP: $(1-\lambda)^3$,
λ_1 = 1 (with AM = 3). We end up with two linearly indep EVecs: $(0, 1, -1)^T$, $(1, 0, 0)^T$. There is one more EVec $v_3 = (0, 0, 1)^T$. We compute

10) Jordan Normal Form
A matrix is in Jordan Normal Form if it is of this form:
$$\begin{bmatrix}J_{k_1}(\lambda_1) & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & J_{k_n}(\lambda_n)\end{bmatrix}$$
Each $J_k(\lambda_i)$ is a Jordan block of size k_i with a diagonal (not always unique) coefficient λ_i:
$$\begin{bmatrix}\lambda_i & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \lambda_i\end{bmatrix}$$
Even if $A \in R^{nn}$, a JNF might not be in R^{nn} but rather in C^{nn}.
Blocs in JNF and Multiplicities of Eigenvalues: The algebraic multiplicity of an eigenvalue λ is the sum of the sizes of blocks with λ on the diagonal. The geometric multiplicity of λ is the number of blocks with λ on the diagonal.

11) Cholesky Decomposition
A matrix A^{nn} is **lower triangular** ∀i < j, A_{ij} = 0
A matrix A^{nn} is **upper triangular** ∀i > j, A_{ij} = 0
These matrices exhibit useful properties. The equation Ax = b can easily be solved on them, by first getting x_1, and then x_2 with direct substitution and so on, on lower triangular matrices. For upper triangular we get x_n first, and go backwards.
Additional Properties of Symmetric Matrices:
Let $A \in R^{nn}$ be a symmetric matrix.
1) If A is **positive definite**, all its diagonal elements are **strictly positive.**
2) If A is **positive semi-definite**, all its diagonal elements are **non-negative.**
3) If A is positive definite then max (A_{ii}, A_{jj}) > $|A_{ij}|$. If A is positive semi definite then max (A_{ii}, A_{jj}) >= $|A_{ij}|$. Thus, the largest coefficient of A is on its diagonal.
4) If A is positive definite then the 1x1, 2x2, ... mxm... matrices in the upper left corner of A are also positive definite. Same holds for semi-definite.
We can quickly notice non-positive-**semi definite** matrices - if we have a symmetric matrix with a negative element, it cannot be PSD. Also: if we see a matrix element $|A_{ij}|$ > max (A_{ii}, A_{jj}) then it's not PSD (e.g if the 3^{rd} element in the first row = 3^{rd} in 1^{st} row = 2, 3^{rd} in 3^{rd} row = 1, then we violate rule 3 as above).
Cholesky Decomposition: any decomposition of a real square matrix A^{nn} of the form $A = LL^T$, where L is a lower triangular matrix.
If A is not positive semi-definite then it doesn't have a Cholesky Decomposition. If A is positive semi definite then it always has a Cholesky Decomposition. Also there exists a version of L with strictly positive diagonal elements.

12) QR Decomposition
QR decomposition is a matrix factorisation method, decomposing our matrix A into an orthogonal matrix Q and an upper triangular matrix R such that A = QR.
Using the Gram-Schmidt Process:
$a_1, ..., a_n$ are the vector columns of A: $[a_1...a_n]$.
Assume they're linearly independent.
1) We apply GS process constructing an orthonormal basis $(e_1...e_n)$ s.t span$(e_1,...,e_i)$ = span$(a_1,..., a_i)$.
2) Selecting Q and R: Set Q = $[e_1, ..., e_n]$. It is semi orthogonal (e.g Q^TQ = I). We then set R like so:
$$\begin{bmatrix}e_1a_1 & e_1a_2 & \cdots & e_1a_n\\ 0 & e_2a_2 & \cdots & e_2a_n\\ 0 & 0 & \ddots & \vdots\\ 0 & \cdots & 0 & e_na_n\end{bmatrix}$$

13) QR Algorithm
Efficiently finds the eigenvalues of a matrix.
1) We start with our matrix A
2) Use GS method to get the QR decomposition: A = QR.
3) Compute the product RQ, set A = RQ.
4) Repeat steps 2-3, until A converges to a diagonal matrix D, whose diagonal entries are the eigenvalues of A.
Properties of the QR Decomposition:
1) For k ∈ N, A_k is similar to A.
2) For k ∈ N, we have $A_k = Q_k^TAQ_k$ from above. So A_k and A have the same eigenvalues and v is an eigenvector of A_k if and only if Q_kv is an eigenvector of A
3) The sequence (A_k) converges to an upper triangular matrix under certain conditions. Useful because of the following property
4) (if A is symmetric) The eigenvalues of an upper triangular matrix are simply its diagonal elements. So, the QR decomposition is easily findable, and from this, we can converge it to an upper triangular matrix, from which eigenvalues are easily findable!!
5) 2) $A^n = Q_1 ... Q_nR_1 ... R_n$

14) Application of the QR Decomposition to Symmetric Matrices
If A is symmetric, so are all the A_k.
If A is symmetric, the algorithm converges, under certain conditions, to a diagonal matrix, hence the eigenvectors of A are in effect the columns of Q^{-1}_k for large enough k

14.1) Convergence Under Specific Conditions:
The LU Decomposition:
A non singular matrix $A \in R^{nn}$ can be factorised as A = LU
L is a lower triangular matrix
U is an upper triangular matrix
We can do this *if and only if* A can be reduced to its row echelon form *without* swapping any two rows.
Properties of the LU Decomposition:
1) Uniqueness of the LU Deomposition when A is non singular: If A = LU and the diagonal elements of L are all one, the decomposition is unique.
2) Let $A \in R^{nn}$ be a symmetric positive definite matrix with distinct eigenvalues $\lambda_1 > \lambda_2 > ... > \lambda_n > 0$ with eigendecomposition $A = Q\Lambda Q^T$. Suppose Q^T = LU with unit lower triangular L and the diagonal elements of U are positive. Then $A_k \to \Lambda$

15) Fixed Points
1) Convergence of a Sequence of Real Numbers: Let $(a_n)_{n\in N} \in R^N$ be a sequence of real numbers and l ∈ R. The sequence (a_n) is said to converge to its limit l, $\lim_{n\to\infty} a_n$ = l if and only if: ∀e > 0, ∃N ∈ N such that ∀n > N, $|a_n - l| < e$
The steps we take to show a sequence of real numbers converges:
1) Find the limit l
2) Take e > 0
3) Find N ∈ N such that $|a_n - l| < e$ for n > N, the value of N will usually depend on and decrease with e
2) Cauchy Sequence: Let $(a_n)_{n\in N} \in R^N$ a sequence of real numbers. Then (a_n) is said to be a Cauchy sequence if and only if: ∀e > 0, ∃N ∈ N such that ∀n, m > N, $|a_n - a_m| < e$
This gives rise to the Cauchy Test: Let $(a_n)_{n\in N} \in R^N$ a sequence of real numbers. Then (a_n) is convergent if and only if it is a Cauchy sequence
3) Metric Spaces: A metric space is a tuple (S, d) where S is non-empty set and d is a metric over S. d : S × S → R is such that:
∀x, y ∈ S, d(x, y) ≥ 0 (positivity)
∀x, y ∈ S, d(x, y) = 0 ⟺ x = y (reflexivity)
∀x, y ∈ S, d(x, y) = d(y, x) (symmetry)
∀x, y, z ∈ S, d(x, z) ≤ d(x, y) + d(y, z) (tri. ineq.)
4) Metric in a normed Vector Space: V is a vector space equipped with the norm ||.||.
Let d be the function $d: \begin{cases} V \times V \to R \\ (x, y) \mapsto ||x - y|| \end{cases}$
d is a metric space.
From here we consider the concepts seen in R but generalised to a metric space (S, d).
5) Convergence in a Metric Space:
Convergence in a metric space. Let (S, d) be a metric space and (a_n) a sequence in S. Then a_n is said to converge to a limit l ∈ S iff : ∀e > 0, ∃N ∈ N such that ∀n > N, $d(a_n, l) < e$
6) Cauchy Sequence in a Metric Space: Let (S, d) be a metric space and (a_n) a sequence in S. (a_n) is a Cauchy sequence iff: ∀e > 0, ∃N ∈ N such that ∀n, m > N, $d(a_n, a_m) < e$
7) Cauchy Test on a Metric Space: Let (S, d) be a metric space and (a_n) a sequence in S: if (a_n) is convergent, then it is a Cauchy sequence.
8) Complete Space: Let (S, d) be a metric space. Then it is said to be a complete space if and only if every Cauchy sequence in S is also converging in S.
9) Completeness of L1, L2 and L∞ Norms: For any k > 0, R^k equipped with any of the three metrics induced by L1, L2 or L∞ is complete. The space (C[a, b], k · k∞) is complete.
10) Only Cauchy Sequences Converge in Complete Metric Spaces: Let (S, d) be a complete metric space and (a_n) a sequence in S. (a_n) is converges iff it's a Cauchy seq.
11) Fixed Point: Let S be a non-empty set and f: S → S a function from S to itself. p ∈ S is a fixed point if: f(p) = p
12) Contraction: Let (S, d) be a metric space and f: S → S. f is called a contraction of S (or a contracting map) **if there exists 0 ≤ α < 1** called the **contraction constant** s.t:
∀x, y ∈ S, d(f(x), f(y)) ≤ αd(x, y).
13) The Fixed Point Theorem: Let (S, d) a complete metric space and f a contraction of S. Then f has a unique fixed point. This leads to the theorem:
14) Uniqueness of Solutions of Differential Equations: Differential Equations have unique solutions: there exists a unique function x: (t_0 - d, t_0 + d) satisfying: dx/dt = f(x(t), t)) with $x(t_0) = x_0$.
d: $\begin{cases}\frac{dx}{dt} = \dot{x}(t) = f(x,t)\\ x(t_0) = x_0\end{cases}$
Where d = min(a,b/(B+bc)) with B = sup(|f(x, t)| : $|x-x_0|$<b, $|t-t_0|$<a)>0

16) Condition Numbers
Condition numbers are the measure of sensitivity of a system to small fluctuations in input.
1) Calculating Condition Numbers:
$$\kappa(P) = \max_{\epsilon} \frac{||s(d) - s(d+\epsilon)||}{||\epsilon||}$$
2) Relative Condition Number:
$$\max \frac{||d||}{||s(d)||}\frac{||s(d) - s(d+\epsilon)||}{||\epsilon||}$$
3) Condition Numbers on Square Matrices:
Let A be a non-singular matrix. The Condition Number of A, κ(A) = ||A-1|| ||A|| = *Relative Condition Number.*
4) Condition Numbers of a Linear Equation:
Ax = b = Condition Number of Coeff Mat A.
Big norm != ill conditioned, we might just be working with big numbers!!
5.1) The Pseudoinverse of a Matrix:
$Ax = b \Leftrightarrow A^TAx = A^Tb \Leftrightarrow x = (A^TA)^{-1}A^Tb$.
The matrix $(A^TA)^{-1}A^T$ is known as the **pseudo inverse** of A, and denoted $A^\dagger$.
5.2) Cond Numbers of non square matrices:
Compute the Pseudoinverse of A. Then to compute the condition number of a non square matrix A: k(A) = $||A^\dagger||\,||A||$
6) Conditioning of a Problem:
To decide if a problem is well or ill conditioned, we can use this Rule of Thumb to guide us: For a condition number κ(A), you lose about **log10(κ(A)) significant figures** in accuracy. If we have an accurate implementation to 12 dp, if the condition number is 1,000,000 then we'll lose about 6 dp in precision.

17) Iterative Solutions to Linear Equations
$A \in R^{nn}$ and $b \in R^n$. We want to solve Ax = b.
Gaussian Elimination $O(n^3)$ is too expensive.
We can approx. the solution with iteration.
1) Theorem of Convergence of Sequences: Let $M \in R^{nn}$ and $c \in R^n$. Let ||.|| be a consistent norm on R^{nn}, if ||M|| < 1 then the sequence (x_k) defined by $x_{k+1} = Mx_k + c$ converges for any starting point x_0.
So, to solve Ax+b we solve the equation
x = Mx+c. Where M = $-G^{-1}R$, c = $G^{-1}b$.
This is in the form required for our Theorem. A solution to the equation, thanks to fixed point theory would be when $x_{k-1} = x_k$
2) Choosing Efficient Choices of Splitting:
1. We want $G^{-1}Rx$ and $G^{-1}b$ to be easy to compute
2. We want ||M|| is small (for fast convergence to a solution).
We'll assume A has no 0s on it's diagonal (if it does we can do a change of basis to achieve it):
3) The ADLU Split: We can write A = D + L + U, where D is the diagonal of A, L, U are the lower and upper triangle parts of A respectively.
4) The Jacobi Method: 1) Set R = L+U
2) We want to solve, for some vector $b \in R^n$
Ax = b. Ax = b ⇔ (D+R)x = b ⇔ x = $-D^{-1}Rx + D^{-1}b$
⇔ x = Mx + c. (M = $-D^{-1}R$, c = $D^{-1}b$).
This splitting is good as x^{k-1} is easily computed as:
D^{-1} = reciprocal of each diag element as diags, R = the combination of the upper and lower triangle matrices, so: Mx = $-D^{-1}Rx$ =
$$-D^{-1}\begin{bmatrix}\sum_{j\neq1}a_{1j}x_j\\ \sum_{j\neq2}a_{2j}x_j\\ \vdots\\ \sum_{j\neq n}a_{nj}x_j\end{bmatrix} = \begin{bmatrix}-\frac{1}{a_{11}}\sum_{j\neq1}a_{1j}x_j\\ -\frac{1}{a_{22}}\sum_{j\neq2}a_{2j}x_j\\ \vdots\\ -\frac{1}{a_{nn}}\sum_{j\neq n}a_{nj}x_j\end{bmatrix}$$
(This makes sense as $-D^{-1}$ is the negative of the reciprocal of each diagonal element, and we multiply this by L+U, which is a matrix of elements except the diagonal).
Since $x^{(k-1)} = Mx^{(k)} + c$, to solve $Ax = b$ we have
$$x_i^{(k+1)} = \frac{1}{a_{i,i}}\left(b_i - \sum_{j\neq i}a_{i,j}x_j^{(k)}\right)$$
Using the formula to compute it shown on the other page.
5) Gauss-Seidel Method : use the ADLU splitting, but do A = (D+L) + U.
Then Ax = b ⇔ (D+L)x = b - Ux ⇔ x = Mx + c,
Where M = $-(D+L)^{-1}U$, c = $(D+L)^{-1}b$.
D+L is a lower triangle matrix with a diagonal D, U is an upper triangle matrix.
And then $x^{k+1} = Mx^k + c \Leftrightarrow (D+L)x^{k+1} = -Ux^k+b$
$$\forall i \in [1,n]\ \sum_{j\leq i}a_{i,j}x_j^{(k+1)} = -\sum_{j>i}a_{i,j}x_j^{(k)} + b_i$$
So the i^{th} element of $x^{(n-1)}$ can be computed thanks to A, $x^{(n)}$, b and the k^{th} elements of $x^{(n-1)}$ for k < i. As the update is computed with more recent quantities, the convergence is faster with the GS method.
These methods are only guaranteed to converge is A is strictly row diagonally dominant - AKA:$\forall i$, $|a_{ii}| > \sum_{j\neq i}|a_{ij}|$
1) Influence of the Condition Number on Convergence: These techniques converge faster for low condition numbers. If big, it might even diverge
2) Irreducible: A matrix is irreducible if it can't take the form: $\begin{bmatrix}A_{11} & A_{12}\\ 0 & A_{22}\end{bmatrix}$
where A_{11} and A_{22} are block matrices.
3) If A is weakly row diagonally dominant and irreducible, Jacobi and G-S still converge.

18) Iterative Techniques for computing Eigenvectors and Eigenvalues
It's $O(n^3)$ to get eigenvectors, (Solve CP; GE for EVecs). Need numerical methods:
The Dominant Eigenvalue of A is the eigenvalue with the largest modulus, a dominant eigenvector is an eigenvector of a dominant eigenvalue.
Power Iteration Theorem:
Let $A \in R^{nn}$ be a diagonalisable matrix with eigenvalues of distinct modulus. Let λ ∈ R be the dominant eigenvalue.
We consider the sequence (x_k) defined by
$x_{k+1} = Ax_k / ||Ax_k||$ and $x_0 \in R^n \setminus (0)$.
Then, as k→infinity, x_k → v, and $||Ax_k||$ → |λ|. This convergence to v is not rigorous - if λ is negative then we oscillate between v and -v.
Power Iteration has limitations:
1. x_0 is chosen at random, it's possible for it to be such that α1 = ... = αp = 0. In this case, the iteration yields second dominant eigenvalue and eigenvector. We need to make sure there's a non zero component in the corresponding eigenspace - usually happens in computers due to floating point error.
2. Eigenvalues might not have distinct modulus, in which case the power iteration will converge to a linear combination of the corresponding eigenvectors.
3. Convergence might be slow if dominant evec isn't too dominant.

18.1) Inverse Power Iteration
Finds the smallest eigenvalue and its eigenvector. We do this by taking the inverse of A, and then performing power method. This yields the eigenvalue 1/λ, where λ = smallest eigenvalue of A.
Inverse Power Iteration Theorem:
Let $A \in R^{nn}$ be a diagonalisable non-singular matrix with eigenvalues of distinct modulus. Let λ ∈ R be the eigenvalue with the smallest modulus. We consider the sequence (x_k) defined by $x_{k+1} = A^{-1}x_k / ||A^{-1}x_k||$ and $x_0 \in R^n \setminus (0)$. Then:
$$x_k \xrightarrow[k\to\infty]{} v \text{ and } ||A^{-1}x_k|| \xrightarrow[k\to\infty]{} \left|\frac{1}{\lambda}\right|$$

18.2) Shifts
Let $A \in R^{nn}$ and s ∈ R. The matrix A-sI is the **shifted matrix** and with the following property:
Let $A \in R^{nn}$ and s ∈ R. λ ∈ R is an eigenvalue of A if and only if λ-s is an eigenvalue of A-sI with the same eigenvectors. Thanks to the shifted matrix, it is possible to focus on a particular eigenvalue of the matrix A.
Theorem of Shifts: $A \in R^{nn}$ is a diagonalisable matrix with eigenvalues of distinct modulus and s ∈ R. Suppose that the shifted matrix A - sI is non-singular (it is non-singular as long as s is not an eigenvalue of A), then by performing inverse power iterations, we can find the eigenvalue of A that it is the closest to s.
Thus we can use the Inverse Power Method and Shifts to discover all eigenvalues.

18.3) Rayleigh Quotient
Let $A \in R^{nn}$ and $x \in R^n \setminus (0)$. Rayleigh quotient R(A, x) is given by: x^TAx / x^Tx, where we apply it on each x in our sequence of power iteration. When using an iterative technique to find the EVal/EVec of a matrix the Rayleigh quotient monitors the convergence to the eigenvalue as it gives access to an approximation of the eigenvalue (rather than the modulus).

18.4) Deflation
We can use the power method to get the max eigenvalue. But it's a problem to get the next eigenvalue, as we have to perform a shift - but we don't actually know what to shift by.
Instead we can **deflate our matrix into an $A^{(n-1)(n-1)}$** matrix, removing the dominant eigenvector. Let $\lambda_1, ..., \lambda_n$ be the eigenvalues of A, ordered according to their magnitude with λ_1 being the dominant one. Define H as a non-singular matrix such that:
$Hx_1 = \alpha e_1$ where α ∈ R \ {0} and $e1=[1,0,..., 0]^T$ is the first vector in the standard basis. We then have:
$HAH^{-1}e_1 = HA\,x_1/\alpha = H\,\lambda_1/\alpha\,x_1 = \lambda_1e_1$
The first column of $HAH^{-1}$is$[\lambda_1\ 0, ... 0]^T$ and so $HAH^{-1} = [\lambda_1\ b^T, 0\ B]$, where
B is an (n-1) by (n-1) matrix with eigenvalues $\lambda_2, ... , \lambda_n$.
They also have the property:
Let λ_2 be the second dominant eigenvector of A with λ_2 =/= λ_1.
An eigenvector x_2 of A corresponding to eigenvalue λ_2 is: $x_2 = H^{-1}[\beta\ z_2]$
With $\beta = b^Tz_2 / \lambda_2 - \lambda_1$ and z_2 is a dominant eigenvector of B

19) Functions of Several Variables
Functions of N Variables are rules which assign to each ordered tuple of real numbers ($x_1, x_2, ... x_n$) a unique real number denoted by f($x_1, x_2, ... x_n$).
Range = set of values f takes, **domain** = set of values taken by params. The graph of f is the set of all points (x, y, z) in R^{n-1}.

19.1) Level Contours
The level curves of a function on two variables are the curves with equation
f(x, y) = k, where k is a constant (in the range of f) A level curve f(x, y) = k is the set of all points in the domain of f at which f takes on a given value k.
These look like rings at a height k, around the parts of the graph which assume that value.
Like contour maps, the surface is steep where level curves are close.

19.2) Partial Derivatives
For a function of two variables, the partial derivatives are denoted by
$f_x(x, y) = \lim_{h\to0} f(x+h, y) - f(x,y)/h$
$f_y(x, y) = \lim_{h\to0} f(x, y + h) - f(x,y)/h$
(AKA just differentiate with respect to 1 and treat the other as a constant).
Geometrically, the partial derivatives of a 2 valued functions are tangent lines that are constant to the dimension which they "ignore" (as seen by T_1 and T_2 which are tangents to C_1 and C_2 (both curves are in the range) at P.
They represent the rate of change of z, with respect to x or y, when the other is fixed. The higher derivatives are computed as imagined
Clairaut's theorem: Suppose f is defined on a disk D that contains the point (a, b). If the functions f_{xy} and f_{yx} are both continuous on D then $f_{xy}(a, b) = f_{yx}(a, b)$

19.3) Directional Derivatives
For a function z = f(x, y), the partial derivatives fx and fy, represent the rates of change in z in the x - and y - directions, i.e., in the directions of unit vectors i and j. (calculated in the same way as before).
The rate of change of an arbitrary unit vector can be found with basic trig.
Our tangent line represents the rate of change of z at a specific point, the direction of which depends on the variable we diff with respect to. (e.g, how much do we move in x and y as we progress?)
Theorem: If f is a differentiable function of x and y, then f has a directional derivative in the direction of a unit vector u = a, b:
$D_uf(x, y) = f_x(x, y)a + f_y(x, y)b$
If the unit vector u makes an angle θ with the positive x-axis, then we can write u = <cos θ, sin θ> and the formula becomes $D_uf(x, y) = f_x(x, y)\cos\theta + f_y(x, y)\sin\theta$

20) The Gradient Vector
$D_uf(x, y) = \langle f_x(x, y), f_y(x, y)\rangle \cdot u$
The first vector in the dot product is the gradient vector, denoted as grad f or ∇f.
It is given by this formula:
$\nabla f(x, y) = \langle f_x(x, y) , f_y(x, y)\rangle$
$= \partial f/\partial x\, i + \partial f/\partial y\, j$
Tangent Plane: The tangent plane to a surface at a given point is a flat plane that locally approximates the surface at that point, and is parallel to the plane spanned by the partial derivatives of the surface at that point. We have these for 3 valued functions.
The gradient vector $\nabla F(x_0, y_0, z_0)$ gives the direction of fastest increase of f.
$\nabla F(x_0, y_0, z_0)$ is orthogonal to the level surface S of f through P.
If we consider a topographical map of a hill and let f(x, y) represent the height above sea level at a point with coordinates (x, y) then a curve of steepest ascent can be drawn as shown in the figure by making it perpendicular to all of the contour lines.
Maxima and Minima:
Local Maximum & Local Min; Absolute Max and Absolute Min - remember these terms.

21) Gradient Based Optimization
A quadratic n variable function looks like:
$Q(x_1,..., x_n) = b_{11}x_1^2 + b_{12}x_1x_2 + \cdots + b_{ij}x_ix_j + \cdots + b_{nn}x_n^2$
We can write this more conveniently with matrices: $Q(x) = x^TAx$, where x is a column vector with elements x_i, 1 <= i <= n, and A is a symm. matrix with $a_{ij} = b_{ij} + b_{ji} / 2$
Note $x_ix_j = x_jx_i$. Eg:
$Q(x_1,x_2) = x_1^2 -ax_1x_2+bx_2x_1 - cx_1x_3 + 5x_2^2$ is expressed as
$x = [x_1, x_2, x_3]$ $A = \begin{bmatrix}1 & \frac{a+b}{2} & -\frac{c}{2}\\ \frac{a+b}{2} & 5 & 0\\ -\frac{c}{2} & 0 & 0\end{bmatrix}$
And then $x^TAx = Q(x_1, x_2)$

23.1) Gradient Descent:
$\nabla f(x^{(k)})$ is a gradient of an n-dimensional function f(x) at iteration k.
The Gradient Descent Formula:
Gradient descent* at iteration k + 1:
$x^{(k+1)} = x^{(k)} - \alpha\nabla f(x^{(k)})$, α > 0 is a fixed step size. (AKA, to compute the next point, we take the current point and subtract the gradient and some step size).
Steepest Descent is a faster version - as we minimise what we have left.
$x^{(k+1)} = x^{(k)} - \alpha_k\nabla f(x^{(k)})$
$a_k = \text{argmin}\, f(x^{(k)} - \alpha\nabla f(x^{(k)}))$
Graphically, we compute the gradient at our point, then go the opposite direction by step size a, and repeat.
For quadratic functions of the form
$f(x) = 1/2x^TQx - b^Tx$ where
Q^n is a real symmetric positive definite matrix, b^n is a real vector, and x^n is a real vector the gradient of the function is:
$\nabla f(x) = Qx - b$, Hessian = $\nabla^2f(x) = Q$.
With this, the optimal step size at iteration k is: $a_k = (g^k)^T(g^k) / ((g^k)^T)Hg^k$
$g^k = \nabla f(x)$ and H = $\nabla^2f(x)$.

Methods:

Method 4.1 Spectral Decomposition On Symmetric Matrix $A \in \mathbb{R}^{n\times n}$:

1. Obtain eigenvalues $\lambda_1,\dots\lambda_k$ of A.
2. Obtain eigenspaces $E_{\lambda_1},\dots,E_{\lambda_k}$ of eigenvalues.
3. For each eigenspace E_{λ_i}, find an orthonormal basis $(v_{(\lambda_i,1)},\dots,v_{(\lambda_i,\dim E_{\lambda_i})})$
4. Combine all these bases (i.e: concatenate) to form an orthogonal matrix $Q = [v_{(\lambda_1,1)},\dots,v_{(\lambda_1,\dim E_{\lambda_1})},\dots,v_{(\lambda_k,1)},\dots,v_{(\lambda_k,\dim E_{\lambda_k})}]$
5. Spectral decomposition of A can now be written as $A = Q\begin{bmatrix}\lambda_1 & 0 & \cdots & 0\\ 0 & \ddots & & \vdots\\ \vdots & & \ddots & 0\\ 0 & \cdots & 0 & \lambda_k\end{bmatrix}Q^T$

Singular Value Decomposition of Matrix $A\in\mathbb{R}^{m\times n}$:

Method 5.1 (More Rows Than Columns, $m > n$):

1. Obtain eigenvalues $\sigma_1^2 \geq \sigma_2^2 \geq \dots \geq \sigma_n^2 \geq 0$ and eigenvectors $v_1,\dots v_n$ of A^TA (i.e: via spectral decomposition)
2. Let $V = [v_1,\dots,v_n]$, $r = rk(A)$ and $\forall i\in[1,r].\ u_i = \frac{1}{\sigma_i}Av_i$
3. Use method 5.3 to extend $u_1,\dots,u_r$ to an orthonormal basis $u_1,\dots,u_m$ of $\mathbb{R}^m$. Orthogonal matrix $U = [u_1,\dots,u_m]$

Method 5.2 (More Columns Than Rows):

1. Obtain eigenvalues $\sigma_1^2\geq\sigma_2^2\geq\dots\geq\sigma_m^2\geq 0$ and eigenvectors $u_1,\dots u_m$ of AA^T (i.e: via spectral decomposition)
2. Let $U = [u_1,\dots,u_m]$, $r = rk(A)$ and $\forall i\in[1,r].\ v_i = \frac{1}{\sigma_i}A^Tu_i$
3. Use method 5.3 to extend $v_1,\dots,v_r$ to an orthonormal basis $v_1,\dots,v_n$ of $\mathbb{R}^n$. Orthogonal matrix $U = [v_1,\dots,v_n]$

Method 5.3 (Extending to a Basis):
Add vectors to the basis iteratively, ensuring they are perpendicular to all other vectors. In the $\mathbb{R}^3$ case, can simply do a cross product. Otherwise, can pick an arbitrary vector not in the plane and use GS.

Gram-Schmidt (GS) Algorithm to Build an Orthonormal Basis $e_1,\dots e_n$ for the n-dimensional Subspace Generated By $v_1,\dots v_j\in\mathbb{R}^n$:

1. Let $u_1 = v_1$
2. Let $u_2 = v_2 - \text{proj}_{u_1}(v_2) = v_2 - (e_1\cdot v_2)e_1$
3. Let $\forall j\in[1,n].\ u_j = v_j - \text{proj}_{u_1}(v_2) - \dots - \text{proj}_{u_{j-1}}(v_j)$
$= v_j - (e_1\cdot v_j)e_1 - \dots - (e_{j-1}\cdot v_j)e_{j-1}$
4. Let $\forall j\in[1,n].\ e_j = \frac{u_j}{\|u_j\|}$

Jacobi Method to Solve $Ax = b$:

1. Pick some $x^{(0)}\in\mathbb{R}^n$
2. Iterate using the below formula:

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\Big(b_i - \sum_{j\neq i}a_{i,j}x_j^{(k)}\Big)$$

Gauss-Seidel Method to Solve $Ax = b$:

1. Pick some $x^{(0)}\in\mathbb{R}^n$
2. Iterate using the below formula (LHS is a summation, so first pick $i = 1$, to find $x_1^{(k+1)}$, then pick $i = 2$ and subtract the previous result from both sides to find $x_2^{(k+1)}$, etc…)

$$\forall i\in[1,n]\sum_{j\leq i}a_{i,j}x_j^{(k+1)} = -\sum_{j>i}a_{i,j}x_j^{(k)} + b_i$$

Method 7.2 (Finding a Cholesky Decomposition of Positive Semi-definite Matrix $A\in\mathbb{R}^{n\times n}$):

1. Let $L = \begin{bmatrix}l_{11} & 0 & 0 & \cdots\\ l_{21} & l_{22} & 0 & \cdots\\ \vdots & \vdots & \ddots & \ddots\\ \vdots & \vdots & & \ddots\end{bmatrix}\in\mathbb{R}^{n\times n}$
2. Compute $LL^T = \begin{bmatrix}l_{11}^2 & 0 & 0 & \cdots\\ l_{11}l_{21} & l_{21}^2 + l_{22}^2 & 0 & \cdots\\ \vdots & \vdots & \ddots & \ddots\\ \vdots & \vdots & & \ddots\end{bmatrix}$
3. Let $A = LL^T$ and solve (i.e: can go from top-left towards the bottom-right, picking the element of LL^T that contains exactly one unknown)

Method 7.3 Using Cholesky Decomposition to solve an equation
Given an equation Ax = b, where A has a Cholesky Decomposition $A = LL^T$, we can solve the equation $LL^Tx = b$ in 2 steps:
1. Let $y = L^Tx$. Solve $Ly = b$ by forward substitution for y. (as y is a vector)
2. Solve $L^Tx = y$ by backward substitution to find x (taking the y vector we have)

Method 6.1 (Finding the JNF of $A\in\mathbb{R}^{n\times n}$):

1. Obtain eigenvalues $\lambda_1,\dots,\lambda_m$ with algebraic multiplicities $a_1,\dots,a_m$.
2. Obtain associated eigenspaces $E_{\lambda_1},\dots,E_{\lambda_m}$ and geometric multiplicities $g_1,\dots,g_m$. Let $\forall i\in[1,m], j\in[1,g_i].\ v_{ij}^1$ be the associated eigenvectors.
3. $\forall i\in[1,m]$. if $g_i < a_i$, find this missing $a_i - g_i$ generalised eigenvectors. $\forall j\in[1,g_i], k\in[1,o_{ij}]$. find all $v_{ij}^k\in\mathbb{C}^n$ such that $(A-\lambda_iI)v_{ij}^k = v_{ij}^{k-1}$ via Gaussian elimination (i.e: find v_{ij}^2 in terms of v_{ij}^1 and so on)
4. Let $B = [v_{11}^1,\dots,v_{11}^{o_{11}},\dots,v_{1g_1}^1,\dots,v_{1g_1}^{o_{1g_1}},\dots,v_{m1}^1,\dots,v_{m1}^{o_{m1}},\dots,v_{mg_m}^1,\dots,v_{mg_m}^{k_{mg_m}}]$
5. Let $J = \begin{bmatrix}J_{k_{11}}(\lambda_1) & & 0\\ & \ddots & \\ 0 & & J_{k_{mg_m}}(\lambda_m)\end{bmatrix}$

where $J_{k_{ij}}(\lambda_i)\in\mathbb{R}^{k_{ij}\times k_{ij}} = \begin{bmatrix}\lambda_i & 1 & 0 & 0 & 0\\ 0 & \lambda_i & \ddots & 0 & 0\\ \vdots & \vdots & \ddots & 1 & 0\\ 0 & 0 & 0 & \lambda_i & 1\\ 0 & 0 & \cdots & 0 & \lambda_i\end{bmatrix}$

Gram-Schmidt (GS) Process:
If we have a linearly independent set of vectors that are a basis for V, we can use the GS Process to convert this set into an orthonormal basis for V (unit vectors that are a basis for V; all vecs orthogonal to each other)
1) From left to right, considering 1 to n vectors at a time. v_1 is orthogonal to everything so far as we haven't considered any other vectors yet. Divide the vector by its magnitude to normalise the vector.
2) For the 2nd vector v_2, we need to find an orthogonal version to $\mathbf{u_1}$. We do this by replacing v_2 with $v_2 - (v_2.u)u$, and then normalise to get $\mathbf{u_2}$. (aka we replace v_2 with v_2 without its projection on the u plane).
3) We do the same for all vectors subtracting its projection onto all the planes from before:
Formula: $u_j = v_j - \text{proj}_{u_1}(v_j) - \text{proj}_{u_2}(v_j) - \dots - \text{proj}_{u_{j-1}}(v_j)$

Finding Basis: get the REF, and take original vector of each pivot column. Span of this = basis.

Finding a Change of Basis Matrix:
We just represent each basis vector in terms of the other basis, and each representation is one of our columns. Here, $B_1 = 4B'_1 + 6B'_2$, $B_2 = -B'_2$.
$B = \begin{bmatrix}2 & 1\\ -2 & 1\end{bmatrix}$, $B' = \begin{bmatrix}2 & -1\\ 1 & -1\end{bmatrix}$, $I_{B'B} = \begin{bmatrix}4 & 0\\ 6 & 1\end{bmatrix}$

Intersection of Subspaces:
$U = \begin{bmatrix}1 & 0\\ 0 & 1\\ 0 & 0\end{bmatrix}$, $V = \begin{bmatrix}1 & -1\\ 1 & 1\\ 1 & 1\end{bmatrix}$
Then we solve x = V1 + V2, x = U1 + U2, and get the line.

Diagonalization of a Matrix A:
1) Obtain the eigenvalues by solving CP, get their eigenspaces.
2) Write the matrix in the form $A = PDP^{-1}$, by writing
D = matrix of eigenvalues, P = matrix of eigenvectors, preserving order.

Least Squares Method (LSM)
Solving $A^TAx = A^Tb$ gives the solution to LS problem.
Take $A = \begin{bmatrix}2 & 2\\ 1 & 2\\ 2 & 0\end{bmatrix}$ $b = \begin{bmatrix}0\\ 5\\ -1\end{bmatrix}$ Ax = b has **no solution**. So, compute $A^TA = \begin{bmatrix}9 & 6\\ 6 & 8\end{bmatrix}$, $A^Tb = [3, 10]$. We proceed with Gaussian Elim and get $x_1 = -1$, $x_2 = 2$.
$(v_j) - \dots - \text{proj}_{u_{j-1}}(v_j)$

The Simplex Method and an Example
1) Write out the list of constraints from our problem
2) Use slack variables; tighten inequality constraints to equality constraints
3) Put the problem into tableau form
4) Choose the most negative entry in the z row and mark that column, evaluate the ratios off the solution column and the positive entries in the our chosen column. Choose the smallest of these and mark that row.
5) Divide the row by our chosen value (denoted by our row and column). Then use Gaussian Elimination to clear every other entry in the y column.
6) Repeat step 4 and 5, until all the entries in the z row are none negative. Then our optimum is achieved.

Example:
A manufacturing company has two circuit boards, R1 and R2, with different components. R1 has 3 resistors, 1 capacitor, 2 transistors, and 2 inductors, while R2 has 4 resistors, 2 capacitors, and 3 transistors. The company has 2400 resistors, 900 capacitors, 1600 transistors, and 1200 inductors for a day's production. We make a profit of 5p on R1 and 9p on R2. Calculate how many of each circuit board the company should produce daily to maximize its overall profits.
1) Constraints: We know x >=0, y >= 0.
Maximise z(profit) = 5x + 9y
3x+4y<=2400 (a), x+2y<=900 (b),
2x+3y<=1600 (c), 2x<=1200 (d)
2) Constraints are now:
3x+4y+r = 2400(a), x+2y+s = 900 (b), 2x+3y+t = 1600 (c), 2x + t = 1200 (d)
3)

		Non-basic variables		Basic variables				
		x	y	r	s	t	u	Solution
Objective function	z	−5	−9	0	0	0	0	0
Basic variables	r	3	4	1	0	0	0	2400
	s	1	2	0	1	0	0	900
	t	2	3	0	0	1	0	1600
	u	2	0	0	0	0	1	1200

4) We choose column y, and value 2 (row s), because the ratios are 2400/4, 900/2, 1600/3, of which 900/2 is the smallest.
5) (z row)+9*(y row)•(r row)-4*(y row)•(t row)- 3 * (y row).

	x	y	r	s	t	u	Solution
z	−1/2	0	0	9/2	0	0	4050
r	1	0	1	−2	0	0	600
y	1/2	1	0	1/2	0	0	450
t	1/2	0	0	−3/2	1	0	250
u	2	0	0	0	0	1	1200

6) elided, but we get $z = 4300$, $r = 100$, $y = 200$, $x = 500$ and $u = 200$

The Dual LP Problem
Every linear programming problem is associated with another problem, known as it's **Dual Problem**.
The Duality Principle: The objective function of the minimisation problem reaches its minimum if and only if the objective function of its dual reaches its maximum. And when they do, they are equal.
We can find the dual maximization problem by taking the transpose, or a maximization or minimization problem. It might be easier to solve, and ideally we want to solve minimization problems as they're doable with the above.

Primal Problem: ➔ Dual Problem:
Minimise: $z = 12x_1 + 16x_2$ | Maximise: $z = 40y_1 + 30y_2$
Subject to the constraints: | Subject to the constraints:
$x_1 + 2x_2 \geq 40$ | $y_1 + y_2 \leq 12$
$x_1 + x_2 \geq 30$ | $2y_1 + y_2 \leq 16$
$x_1 \geq 0; x_2 \geq 0$ | $y_1 \geq 0; y_2 \geq 0$

	x	y	r	s	t	u	Solution
z	−1/2	0	0	9/2	0	0	4050
r	1	0	1	−2	0	0	600
y	1/2	1	0	1/2	0	0	450
t	1/2	0	0	−3/2	1	0	250
u	2	0	0	0	0	1	1200

Transpose matrix (maximisation)

1	1	12
2	1	16
40	30	0

Original matrix (minimisation)

1	2	40
1	1	30
12	16	0

Linear Regression Method:
We construct a Matrix A, with 1 column, and record each of our data, a_i row by row, corresponding to our y_i (which had that data).

$$A = \begin{bmatrix}1 & a_{11} & a_{12} & \dots & a_{1n}\\ 1 & a_{21} & a_{22} & \dots & a_{2n}\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ 1 & a_{i1} & a_{i2} & \dots & a_{in}\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ 1 & a_{m1} & a_{m2} & \dots & a_{mn}\end{bmatrix}$$

We take $z = [s_0, s_1, \dots, s_n] \in R^{n+1}$.
Az-b = $\begin{bmatrix}s_0 + s_1a_{11} + \dots + s_na_{1n} - y_1\\ s_0 + s_1a_{m1} + \dots + s_na_{mn} - y_m\end{bmatrix}$
Minimizing Az-b gives us our solution. We do this by solving the normal equation $A^TAz = A^ty$
To be clear:
z = vector of parameters we want to estimate.
A = matrix of data point estimates we're minimizing
b = y = vector of measurements

Condition numbers on the context of linear equations
1) x + y = 1.
2) x + ay = 0, where a is some unknown constant. When a =/= 1, our eqs have a solution. When a is close to one, its value drastically changes our solutions: Take a = 0.9999, x = -9999 and y = 10000.
In this example, d = 0.9999, e = -0.0009, s(d) = (-9999, 10000), s(d+e) ≈ (-999, 1000).
So, k(P) = max||s(d) – s(d+e)|| / e = ||(-9999, 10000) - (-999, 1000)||/-0.0009, = ||(-9000, 9000)||/-0.0009.
We can choose different norms to evaluate the top part, with the l_1 norm we end with $2*10^7$, with the l2, we get root $2 * 10^7$.

The QR Decomposition using Gram Schmidt Process
Let A = $\begin{bmatrix}1 & 1 & 0\\ 1 & 0 & 1\\ 0 & 1 & 1\end{bmatrix}$
$a_1 = \begin{pmatrix}1\\1\\0\end{pmatrix}$, $a_2 = \begin{pmatrix}1\\0\\1\end{pmatrix}$, $a_3 = = \begin{pmatrix}0\\1\\1\end{pmatrix}$.
$e_1 = \begin{pmatrix}1\\1\\0\end{pmatrix}/\sqrt{2}$ (i.e. a_1 but normalized). e_1 as in the Gram Schmidt Process we take the first basis vector unmodified.
And $e_1\cdot a_1 = \sqrt{2}$, $e_1\cdot a_2 = 1/\sqrt{2}$, $e_1\cdot a_3 = 1/\sqrt{2}$
$a_2 - (e_1\cdot a_2)e_1 = \begin{pmatrix}1/2\\-1/2\\1\end{pmatrix}$, $e_2 = \begin{pmatrix}1\\-1\\2\end{pmatrix}/\sqrt{6}$.
Also $e_2\cdot a_2 = 3/\sqrt{6}$, $e_2\cdot a_3 = 1/\sqrt{6}$
$u_3 = a_3 - (e_1\cdot a_3)e_1 - (e_2\cdot a_3)e_2 = \begin{pmatrix}-1\\1\\1\end{pmatrix}/\sqrt{3}$ with $e_3 = \begin{pmatrix}-1\\1\\1\end{pmatrix}/\sqrt{3}$. $e_3\cdot a_3 = 1/\sqrt{6}$.
2) Selecting Q and R: Set $Q = [e_1, \dots, e_n]$. It is semi orthogonal (e.g $Q^TQ = I$). We set R like so

The QR Algorithm:
1) We start with our matrix A
2) Use GS method to get the QR decomposition: A = QR.
3) Compute the product RQ, set A = RQ.
4) Repeat steps 2-3, until A converges to a diagonal matrix D, whose diagonal entries are the eigenvalues of A.

Power Method:
A = $\begin{bmatrix}1 & -3 & 2 & 4\\ -3 & 1 & 4 & 2\\ 2 & 4 & 1 & -3\\ 4 & 2 & -3 & 1\end{bmatrix}$
Incidentally (not related to the method) our eigenvalues are 2, 4, 6, -8, with eigenspaces:
$E_2 = \begin{pmatrix}-1\\1\\1\\1\end{pmatrix}$ $E_4 = \begin{pmatrix}1\\1\\1\\1\end{pmatrix}$ $E_6 = \begin{pmatrix}-1\\1\\1\\-1\end{pmatrix}$ $E_8 = \begin{pmatrix}-1\\-1\\1\\1\end{pmatrix}$
Now consider the sequence $X_{n+1} = \frac{AX_n}{\|AX_n\|_\infty}$.
Take arbitrary $x_0 = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}$. We keep iterating and computing X_{n+1} and eventually $x_{15} = \begin{pmatrix}-0.97\\-1\\0.97\\1\end{pmatrix}$ Indeed, we are converging to the eigenvector that corresponds to the eigenvalue of largest modulus.

Change of Basis to solve $Ax = b$: We want to solve $Ax = b$ with $A = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$. It has 0s on its diagonal.
Consider $C = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$. Then $C^{-1} = C$ and

$$Ax = b \Leftrightarrow C^{-1}Ax = C^{-1}b \Leftrightarrow C^{-1}Ax = C^{-1}b \Leftrightarrow (C^{-1}AC)C^{-1}x = C^{-1}b$$
$$\Leftrightarrow (C^{-1}AC)C^{-1}x = C^{-1}b$$

We select C such that $C^{-1}AC$ is diagonal: here $C^{-1}AC = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} = B$.
By denoting $C^{-1}x = y$ and $C^{-1}b =$ we can solve **By** = **c** where B can be split according to the given method. Solving this equation will give y and x can be retrieved as x

Finding the Nullspace (Kernel) and the Imagespace (Rangespace)
To compute the nullspace of our matrix A, we solve Ax = 0 using Gaussian Elimination.
To compute the imagespace, we move A into RREF and then take the span vectors of the columns.

Proving that we can iteratively solve a system of Linear Equations
$A = \begin{bmatrix}5 & 2\\ 3 & 4\end{bmatrix} \in R^{2x2}$. We work in R^2 with the L1-norm.
Take $G = \begin{bmatrix}5 & 0\\ 0 & 4\end{bmatrix}$ $R = \begin{bmatrix}0 & 2\\ 3 & 0\end{bmatrix}$ Since $M = -G^{-1}R = -[0\ 2/5, 3/4\ 0]$. $\|M\|_1$ ¾ < 1, we can find an iterative solution.

Example of finding the JNF

$$A = \begin{bmatrix}-2 & 2 & 1\\ -7 & 4 & 2\\ 5 & 0 & 0\end{bmatrix}$$
$$\det(A - \lambda I) = -\lambda(1-\lambda)^2$$

We have $\lambda_1 = 0, \lambda_2 = 1$, with AM 1 and 2 respectively.
$E_{\lambda_1} = \text{span}\left\{\begin{bmatrix}0\\-1\\2\end{bmatrix}\right\}$, $AM = GM$ so done
$E_{\lambda_2} = \text{span}\left\{\begin{bmatrix}1\\-1\\5\end{bmatrix}\right\}$, $AM < GM$ so must find generalised eigenvector
$(A-\lambda_2I)v_{2,1}^2 = v_{2,1}^1$, $\begin{bmatrix}-1 & 2 & 1\\ -7 & 3 & 2\\ 5 & 0 & -\end{bmatrix}v_{2,1}^2 = \begin{bmatrix}1\\-1\\5\end{bmatrix}$, $v_{2,1}^2 = \begin{bmatrix}0\\3\\-5\end{bmatrix}$
So, $B = \begin{bmatrix}0 & 1 & 0\\ -1 & -1 & 3\\ 2 & 5 & -5\end{bmatrix}$

5) So, we have $B^{-1}AB = J = \begin{bmatrix}0 & 0 & 0\\ 0 & 1 & 1\\ 0 & 0 & 1\end{bmatrix}$

SVD Example:

$$A = \begin{bmatrix}1 & 1 & -1\\ 1 & -1 & 1\\ 1 & 1 & -1\\ 1 & -1 & 1\end{bmatrix}$$
$$A^TA = \begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ -1 & 1 & -1 & 1\end{bmatrix}\begin{bmatrix}1 & 1 & -1\\ 1 & -1 & 1\\ 1 & 1 & -1\\ 1 & -1 & 1\end{bmatrix} = \begin{bmatrix}4 & 0 & 0\\ 0 & 4 & -4\\ 0 & -4 & 4\end{bmatrix}$$
$$\det(A^TA - \lambda I) = 0$$
$$\det\begin{bmatrix}4-\lambda & 0 & 0\\ 0 & 4-\lambda & -4\\ 0 & -4 & 4-\lambda\end{bmatrix} = (4-\lambda)((4-\lambda)(4-\lambda) - 4^2)$$
$$= (4-\lambda)(\lambda^2 - 8\lambda) = -\lambda^3 + 12\lambda^2 - 32\lambda = -\lambda(\lambda-4)(\lambda-8)$$
$$\lambda_1 = 8, \lambda_2 = 4, \lambda_3 = 0$$
$$S = \begin{bmatrix}2\sqrt{2} & 0 & 0\\ 0 & 2 & 0\\ 0 & 0 & 0\end{bmatrix}$$
$$(A^TA - \lambda_1I)v_1 = 0$$
$$\begin{bmatrix}-4 & 0 & 0\\ 0 & -4 & -4\\ 0 & -4 & -4\end{bmatrix}v_1, v_1 = \begin{bmatrix}0\\ \sqrt{2}/2\\ -\sqrt{2}/2\end{bmatrix}$$
$$(A^TA - \lambda_2I)v_2 = 0$$
$$\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & -4\\ 0 & -4 & 0\end{bmatrix}v_2 = 0, v_2 = \begin{bmatrix}1\\0\\0\end{bmatrix}$$
$$(A^TA - \lambda_3I)v_3 = 0$$
$$\begin{bmatrix}4 & 0 & 0\\ 0 & 4 & -4\\ 0 & -4 & 4\end{bmatrix}v_3 = 0, v_3 = \begin{bmatrix}0\\ \sqrt{2}/2\\ \sqrt{2}/2\end{bmatrix}$$
$$V = [v_1, v_2, v_3] = \begin{bmatrix}0 & 1 & 0\\ \sqrt{2}/2 & 0 & \sqrt{2}/2\\ -\sqrt{2}/2 & 0 & \sqrt{2}/2\end{bmatrix}$$
$$u_1 = \frac{1}{\sigma_1}Av_1 = \frac{1}{2\sqrt{2}}\begin{bmatrix}1 & 1 & -1\\ 1 & -1 & 1\\ 1 & 1 & -1\\ 1 & -1 & 1\end{bmatrix}\begin{bmatrix}0\\ \sqrt{2}/2\\ -\sqrt{2}/2\end{bmatrix} = \frac{1}{2\sqrt{2}}\begin{bmatrix}\sqrt{2}\\ -\sqrt{2}\\ \sqrt{2}\\ -\sqrt{2}\end{bmatrix} = \frac{1}{2}\begin{bmatrix}1\\-1\\1\\-1\end{bmatrix}$$
$$u_2 = \frac{1}{\sigma_2}Av_2 = \frac{1}{2}\begin{bmatrix}1 & 1 & -1\\ 1 & -1 & 1\\ 1 & 1 & -1\\ 1 & -1 & 1\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix} = \frac{1}{2}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}$$

Now need to extend to orthonormal basis: find u_3 and u_4

Pick arbitrary vector $x\in\mathbb{R}^4$, $x = \begin{bmatrix}1\\-1\\-1\\-1\end{bmatrix}$

$$u_3 = \text{normalise}(x - \text{proj}_{u_1}(x) - \text{proj}_{u_2}(x))$$
$$= \text{normalise}\left(\begin{bmatrix}1\\-1\\-1\\-1\end{bmatrix} - \left(\frac{1}{2}\begin{bmatrix}1\\-1\\1\\-1\end{bmatrix}\cdot\begin{bmatrix}1\\-1\\-1\\-1\end{bmatrix}\right)\left(\frac{1}{2}\begin{bmatrix}1\\-1\\1\\-1\end{bmatrix}\right) - \left(\frac{1}{2}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}\cdot\begin{bmatrix}1\\-1\\-1\\-1\end{bmatrix}\right)\left(\frac{1}{2}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}\right)\right)$$
$$= \text{normalise}\left(\begin{bmatrix}1\\-1\\-1\\-1\end{bmatrix} - \frac{1}{2}\begin{bmatrix}1\\-1\\1\\-1\end{bmatrix} + \frac{1}{2}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}\right) = \text{normalise}\left(\begin{bmatrix}1\\0\\-1\\0\end{bmatrix}\right) = \begin{bmatrix}\sqrt{2}/2\\ 0\\ -\sqrt{2}/2\\ 0\end{bmatrix}$$

Pick arbitrary vector $y\in\mathbb{R}^4$, $y = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$

$$u_4 = \text{normalise}(y - \text{proj}_{u_1}(y) - \text{proj}_{u_2}(y) - \text{proj}_{u_3}(y))$$
$$= \text{normalise}\left(\begin{bmatrix}0\\0\\0\\1\end{bmatrix} - \left(\frac{1}{2}\begin{bmatrix}1\\-1\\1\\-1\end{bmatrix}\cdot\begin{bmatrix}0\\0\\0\\1\end{bmatrix}\right)\left(\frac{1}{2}\begin{bmatrix}1\\-1\\1\\-1\end{bmatrix}\right) - \left(\frac{1}{2}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}\cdot\begin{bmatrix}0\\0\\0\\1\end{bmatrix}\right)\left(\frac{1}{2}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}\right)\right.$$
$$\left. - \left(\frac{\sqrt{2}}{-2}\begin{bmatrix}1\\0\\-1\\-\end{bmatrix}\cdot\begin{bmatrix}0\\0\\0\\1\end{bmatrix}\right)\left(\frac{\sqrt{2}}{-2}\begin{bmatrix}1\\0\\-1\\0\end{bmatrix}\right)\right)$$
$$= \text{normalise}\left(\begin{bmatrix}0\\0\\0\\1\end{bmatrix} + \frac{1}{4}\begin{bmatrix}1\\-1\\1\\-1\end{bmatrix} - \frac{1}{4}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}\right) = \text{normalise}\left(\begin{bmatrix}0\\-1/2\\0\\1/2\end{bmatrix}\right) = \frac{\sqrt{2}}{2}\begin{bmatrix}0\\-1\\0\\1\end{bmatrix}$$
$$U = \begin{bmatrix}1/2 & 1/2 & \sqrt{2}/2 & 0\\ -1/2 & 1/2 & 0 & -\frac{\sqrt{2}}{2}\\ 1/2 & 1/2 & -\sqrt{2}/2 & 0\\ -1/2 & 1/2 & 0 & \frac{\sqrt{2}}{2}\end{bmatrix}$$
$$A = \begin{bmatrix}1/2 & 1/2 & \sqrt{2}/2 & 0\\ -1/2 & 1/2 & 0 & -\sqrt{2}/2\\ 1/2 & 1/2 & -\sqrt{2}/2 & 0\\ -1/2 & 1/2 & 0 & \sqrt{2}/2\end{bmatrix}\begin{bmatrix}2\sqrt{2} & 0 & 0\\ 0 & 2 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}0 & 1 & 0\\ \sqrt{2}/2 & 0 & \sqrt{2}/2\\ -\sqrt{2}/2 & 0 & \sqrt{2}/2\end{bmatrix}^T$$

Determining Whether a Function $f(x,y)$ has Extreme Value at a Critical Point (a,b):
1. First verify it is a critical (turning) point. i.e: $f_x(a,b) = f_y(a,b) = 0$
2. Let $D = \det\begin{bmatrix}f_{xx}(a,b) & f_{xy}(a,b)\\ f_{xy}(a,b) & f_{yy}(a,b)\end{bmatrix}$
(Note that $f_{xy} = f_{yx}$ and $\begin{bmatrix}f_{xx}(a,b) & f_{xy}(a,b)\\ f_{xy}(a,b) & f_{yy}(a,b)\end{bmatrix}$ is termed the Hessian Matrix)
- If $D > 0$ and $f_{xx}(a,b) > 0$, then $f(a,b)$ is a local minimum
- If $D > 0$ and $f_{xx}(a,b) < 0$, then $f(a,b)$ is a local maximum
- If $D < 0$, then $f(a,b)$ is a saddle point, not an extremum
- If $D = 0$, then the test is inconclusive and $f(a,b)$ could be a local maximum, minimum or a saddle point.

Determining Whether a Function $f(x,y,z,\dots)$ has Extreme Value at a Critical Point P:
1. Compute the function's Hessian matrix at P
2. Obtain the eigenvalues $\lambda_1,\dots,\lambda_k$ of this matrix
- If $\forall i\in[1,k].\lambda_i > 0$ (matrix is positive-definite), then P is a local minimum
- If $\forall i\in[1,k].\lambda_i < 0$ (matrix is negative-definite), then P is a local maximum
- If $\exists i\in[1,k].\lambda_i = 0$ (matrix is singular), then the test is inconclusive
- Otherwise (some positive and some negative eigenvalues, but all are non-zero), point is a saddle point

Deflation Method
Property: Let λ_2 be the second dominant eigenvector of A with $\lambda_2 =/= \lambda_1$. An eigenvector x_2 of A corresponding to eigenvalue λ_2 is: $x_2 = H^{-1}\begin{bmatrix}\beta\\ z_2\end{bmatrix}$. $\beta = b^Tz_2 / \lambda_2 - \lambda_1$ and z_2 is a dominant eigenvector of B
Let A = $\begin{bmatrix}1 & -3 & 2 & 4\\ -3 & 1 & 4 & 2\\ 2 & 4 & 1 & -3\\ 4 & 2 & 3 & -1\end{bmatrix}$.
1) We'll apply deflation to this using the **Householder matrix**. First we need to construct it:
$H = I = 2uu^T / u^Tu$
We take $x_1 = [1\ 1\ -1\ -1]^T$, which is a vector in the eigenspace of our biggest eigenvalue.
$u = x_1 + \|x_1\|_2e_1$
So u = $\begin{bmatrix}3\\1\\-1\\-1\end{bmatrix}$ & $x_1 = \begin{bmatrix}1\\1\\-1\\-1\end{bmatrix}$ and so H = 1/6 $\begin{bmatrix}-3 & -3 & 3 & 3\\ -3 & 5 & 1 & 1\\ 3 & 1 & 5 & -1\\ 3 & 1 & -1 & 5\end{bmatrix}$.
Since H is a householder matrix,

$$HAH = HAH^{-1} = \begin{bmatrix}\lambda_1 & b^T\\ 0 & B\end{bmatrix} = \begin{bmatrix}-8 & 0 & 0 & 0\\ 0 & 4 & 4/3 & 4/3\\ 0 & 4/3 & 10/3 & 0\\ 0 & -4/3 & 0 & 14/3\end{bmatrix}$$

And we receive a matrix B (the 3x3 matrix from the bottom right corner), and -8 is our largest eigenvalue (as we knew). $b^T = [0, 0, 0]$. Performing power iterations on B, the algorithm converges to $z_2 = \begin{bmatrix}1\\1/2\\-1\end{bmatrix}$, with corresponding eigenvalue 6. In this case, $\beta = b^Tz_2 / \lambda_2 - \lambda_1 = 0$ and $X_2 = H^{-1}\begin{bmatrix}0\\1\\1/2\\-1\end{bmatrix} = H\begin{bmatrix}0\\1\\1/2\\-1\end{bmatrix} = 3/4\begin{bmatrix}-1\\1\\1\\-1\end{bmatrix}$
which is an eigenvector of A.

Calculating the Directional Derivative
1. The formula is: $\mathbf{D_uf(x, y) = f_x(x, y)\cos\theta + f_y(x, y)\sin\theta}$
2. So we partial differentiate wrt x, and y and then sub in, along with our theta.
3. We will get a value which represents how much we change in the direction u, per unit u.

Example:
For a function $f_{(x,y)} = x^3 - 3xy + 4y^2$, find the directional derivative
$D_uf(x, y)$ at (1,2) where **u** is the unit vector at an angle $\theta = \pi/6$:
1) $D_uf(x, y) = f_x(x, y)\cos\theta + f_y(x, y)\sin\theta$
$D_uf(x, y) = f_x(x, y)\cos\pi/6 +$
$f_y(x, y)\sin\pi/6 =$
$(3x^2 - 3y)\sqrt{3}/2 + (-3x + 8y)\ 1/2 =$
$= ½[3\sqrt{3}x^2 - 3x + (8 - 3\sqrt{3})y]$
Subbing (1, 2) we get $(13 - 3\sqrt{3})1 / 2$

Calculating Gradient Vector
1. To calculate, just differentiate wrt x – this is our i component, and differentiate wrt y, this is our j component.

Example:
$f(x, y) = x^2y^3 - 4y$. Find the directional derivative at (2, -1) in the direction of v = 2i + 5j. Taking partial derivatives and subbing in (2,-1), we get $\nabla f(2,-1) = -4\mathbf{i} + 8\mathbf{j}$
This is our gradient vector, and we can continue to get the directional derivative by multiplying it by the directional unit vector, and multiplying u and grad f, then finding the magnitude

Maximising the Directional Vector
We have a function f of two/three variables. To maximise it the directional vector (out of all possible directions), we can use this theorem:
$\text{Max}(D_uf) = \text{Max}(\nabla f\cdot\mathbf{u}) = \text{Max}(|\nabla f||\mathbf{u}|\cos\theta) = \text{Max}(|\nabla f|\cos\theta) = |\nabla f|$.
The max value of cos is 1, at $\theta = 0$. So the max value of D_uf is $|\nabla f|$ when $\theta = 0$. AKA it's just the magnitude of the gradient vector – this makes sense as the gradient vector and directional unit vector are going in the same path – the angle between them is 0.

Steepest Descent on Quadratic
1) Compute the gradient function, $g^k = \nabla f(x)$, and H which is computed in (Determining if we have a Critical point) by taking second derivatives wrt specific variables in specific orders.
2) Sub in the x^0 into the gradient function, then compute $a_k = (g^k)^T(g^k) / ((g^k)^T)Hg^k$.
3) Then generate x_1 as $x_0 - a_0\nabla f(x^0)$
3) Repeat until a constraint is met.
Example: One step
We apply Steepest Descent to **minimize** $f(x) = x_1^2 + x_2^2 + 2x_2 + 4$.
Choose $x^0 = [2\ 1]^T$
The gradient $g^k = \nabla f(x) = [2x_1, 2x_2 + 2]^T$
$H = \nabla^2f(x) = [2\ 0, 0\ 2]$
The gradient at $x^0 = [2(2), 2(1)+2] = [4, 2]^T$
The step length $a = (g^0)^T(g^0) / ((g^0)^THg^0 = [4\ 4][4\ 4]^T / [4\ 4][2\ 0, 0\ 2][4\ 4]^T = 32/64$.
$X^1 = [2\ 1]^T - ½[4\ 4]^T = [0\ -1]$.
We continue and find the next gradient is 0, so stop as we found a minimum point.
On a computer, because of flop, we wouldn't actually have 0.
Common stopping criteria include:
1) $\|\nabla f\| < e$ – gradient becomes close to 0
2) $\|f(x_{k+1}) - f(k)\| < e$ – terms don't change much
3) $\|f(x_{k+1}) - f(k)\| / \max(1, |f(x_k)|)$ – a relative measure, or we could take the norm of x_k instead of function val.
Rather than computing best step size, we could use a constant, a diminishing one, a small minimization.

24) Conjugate Gradient Method
Minimizes a quadratic function.
$F(x) = ½\ x^TAx - bx$, where A^{nn} is a real symmetric positive definite matrix, and b^n is a real vector. The solution to the minimization problem is equivalent to finding x in $\nabla F(x) = 0$, ie: solving Ax-b = 0.
A-conjugate direction: $\{d_0, \dots d_{n-1}\}$, $d_i\in\mathbb{R}^n$ are search direction vectors orthogonal to each other wrt A if $d^T_iAd_j = 0, \forall i\neq j$.
$d^T_iAd_j$ is also written as $\langle d_i, d_j\rangle_A = \langle d_i, Ad_j\rangle = \langle Ad_i, d_j\rangle = \langle d_j, A^Td_j\rangle$
These directions $d_0 \dots d_n$-1 are A-orthogonal or A-conjugate directions. If A = I these vectors are orthogonal in the normal way.

Using the Conjugate Gradient Method
1) Use a residual vector r_0 as the initial search direction d_1. The method of selecting d_k changes afterwards.
$r_0 = b - Ax_0$
2) Calculate the scalar a_1 with $r_0^Tr_0 / d_1^TAd_1$
3) Now x_1 can be computed with the formula: $x_1 = x_0 + a_1d_1$
4) Compute the residual r_1 for the next iteration: $r_0 - a_1Ad_1$
5) We are now at iteration 2. Calculate the Scalar $\beta_2 = -r_1^Tr_1 / r_0^Tr_0$
6) Use β_2 to compute the next search direction $d_2 = r_1 - \beta_2d_1$
7) Calculate a_2 using $d_2 = r_1^Tr_1 / d_2^TAd_2$
8) Find $x_2 = x_1 + a_2d_2$
9) Compute the residual $r_2 = r_1 - a_2Ad_2$ and use a norm (usually L_2) to see if we have small enough value to terminate. If not, repeat steps 5-9.

Example:
1) Use residual vector r_0 as the initial search direction d_1.
$A = \begin{bmatrix}5 & 1\\ 1 & 8\end{bmatrix}$, $x = \begin{bmatrix}x_1\\ x_2\end{bmatrix}$, $b = \begin{bmatrix}3\\2\end{bmatrix}$. We'll take x_0 as $\begin{bmatrix}2\\1\end{bmatrix}$
$r_0 = b - Ax_0 = \begin{bmatrix}3\\2\end{bmatrix} - \begin{bmatrix}5 & 1\\ 1 & 8\end{bmatrix}\begin{bmatrix}2\\1\end{bmatrix} = \begin{bmatrix}-8\\-8\end{bmatrix}$. So $d_1 = r_0 = \begin{bmatrix}-8\\-8\end{bmatrix}$
2) Calculate scalar $a_1 = r_0^Tr_0 / d_1^TAd_1 = [-8, -8]\begin{bmatrix}-8\\-8\end{bmatrix}$ /
$[-8, -8]\begin{bmatrix}5 & 1\\ 1 & 8\end{bmatrix}\begin{bmatrix}-8\\-8\end{bmatrix} = 2/15$
3) Now x_1 (improved approximation) can be reached
$x_1 = \begin{bmatrix}2\\1\end{bmatrix} + 2/15\begin{bmatrix}-8\\-8\end{bmatrix} = [0.933, -0.0667]$
4) Compute the residual for the next iteration $r_1 = r_0 - a_1Ad_1$
$= [\begin{bmatrix}-8\\-8\end{bmatrix} - 2/15\begin{bmatrix}5 & 1\\ 1 & 8\end{bmatrix}\begin{bmatrix}-8\\-8\end{bmatrix} = \begin{bmatrix}-1.6\\-1.6\end{bmatrix}$
5) Calculate the scalar B_2:
$[-1.6, 1.6]\begin{bmatrix}-1.6\\-1.6\end{bmatrix} / [-8, -8]\begin{bmatrix}-8\\-8\end{bmatrix} = -0.04$
6) $d_2 = r_1 - B_2d_1 = \begin{bmatrix}-1.6\\-1.6\end{bmatrix} + 0.04[-8\ -8] = [-1.92\ 1.28]$
7) $a_2 = r_1^Tr_1 / d_2^TAd_2 =$
$[-1.6, -1.6]\begin{bmatrix}-1.6\\-1.6\end{bmatrix}$ / $[-1.92, 1.28]\begin{bmatrix}5 & 1\\ 1 & 8\end{bmatrix}[-1.92\ 1.28] = 0.1923$
8) Find x_2 and see if we have reached our solution:

9) $r_2 = r_1 - a_2Ad_2 = \begin{bmatrix}-1.6\\-1.6\end{bmatrix} - 0.1923\begin{bmatrix}5 & 1\\ 1 & 8\end{bmatrix}\begin{bmatrix}-1.92\\ 1.28\end{bmatrix}$
$x_2 = x_1 + a_2d_2 = \begin{bmatrix}0.9333\\ 0.0667\end{bmatrix} + 0.1923\begin{bmatrix}-1.92\\ 1.28\end{bmatrix} = \begin{bmatrix}0.5641\\ 0.1794\end{bmatrix}$
$= 1.0e - 0.4[-0.64, 0.64]$. $\|r_2\| = 9.051e\text{-}0.5$