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 $A \in \mathbb{R}^{n\times m}$:

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 $A^T A$ (i.e: via spectral decomposition)
2. Let $V = [v_1, \dots, v_r]$, $r = rk(A)$ 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 $\forall i \in [1, r].v_i = \frac{1}{\sigma_i}A^T u_i$
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 $\forall j \in [1, n].\ u_j = v_j - \text{proj}_{u_1}(v_j) - \dots - \text{proj}_{u_{j-1}}(v_j) = v_j - (e_1 \cdot v_j)e_1 - \dots - (e_{j-1} \cdot v_j)e_{j-1}$
4. Let $\forall j \in [1, n], e_j = \frac{u_j}{\|u_j\|}$

Computing the Cross Product of two vectors a and b

Set up a 3x3 matrix, with the unit vectors I, j, k in the first row, the elements of a in the second row, and the elements of b in the third row. The cross product = det of this matrix.

Jacobi Method to Solve $Ax = b$:

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

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

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...)

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

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

1. Let $L = \begin{bmatrix} l_{11} & 0 & 0 & \cdots\\ l_{21} & l_{22} & 0 & \cdots\\ \vdots & \vdots & \ddots & \\ \vdots & \vdots & \vdots & \ddots\end{bmatrix} \in \mathbb{R}^{n\times n}$
2. Compute $LL^T = \begin{bmatrix} l_{11}^2 & 0 & 0 & \cdots\\ l_{11}l_{21} & l_{21}^2 + l_{22}^2 & 0 & \cdots\\ l_{31}l_{11} & l_{31}l_{21} + l_{32}l_{22} & l_{31}^2 + l_{32}^2 + l_{33}^2 & \cdots\\ l_{41}l_{11} & l_{41}l_{21} + l_{42}l_{22} & l_{41}l_{31} + l_{42}l_{32} + l_{43}l_{33} & \cdots\end{bmatrix}$
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 $LL^T x = b$ 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 vector we have)

Method 6.1 (Finding the JNF of $A \in \mathbb{R}^{n\times n}$) (J = B^{-1}AB):

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

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 u_1. We do this by replacing v_2 with $v_2 - (v_2.u)u$, and then normalise to get 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 = $\begin{bmatrix}1 & -1\\ 1 & 1\\ 0 & 1\end{bmatrix}$

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. If we encounter a problem with more variables than our data (A) has columns (e.g, we have a constant field), we pad out the data with a column of constants (usually 1).

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. **Turn it into a maximization problem, if not already, by multiplying the Objective Function by -1 or by getting the Dual 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 non 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 3 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_s), x+2y+s = 900 (b_s), 2x+3y+t = 1600 (c_s), 2x + t = 1200 (d_s)

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 its **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 maximization problems as they're doable with the Simplex.

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

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, ..., s_n] ∈ $\mathbb{R}^{n+1}$.

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

Example: The QR Decomposition using Gram Schmidt Process

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

$$a_1 = \begin{bmatrix}1\\1\\0\end{bmatrix}, a_2 = \begin{bmatrix}1\\0\\1\end{bmatrix}, a_3 = \begin{bmatrix}0\\1\\1\end{bmatrix}$$

1. $u_1 = \begin{bmatrix}1\\1\\0\end{bmatrix}, e_1 = \frac{\sqrt{2}}{2}\begin{bmatrix}1\\1\\0\end{bmatrix}$

2. $u_2 = a_2 - (e_1 \cdot a_2)e_1 = \begin{bmatrix}1/2\\ -1/2\\ 1\end{bmatrix}, e_2 = \frac{\sqrt{6}}{6}\begin{bmatrix}1\\-1\\2\end{bmatrix}$

3. $u_3 = a_3 - (e_1 \cdot a_3)e_1 - (e_2 \cdot a_3)e_2 = \begin{bmatrix}-2/3\\ 2/3\\ 2/3\end{bmatrix}, e_3 = \frac{\sqrt{3}}{3}\begin{bmatrix}-1\\1\\1\end{bmatrix}$

4. Let $Q = [e_1, \dots, e_n] = \frac{\sqrt{6}}{6}\begin{bmatrix}\sqrt{3} & 1 & -\sqrt{2}\\ \sqrt{3} & -1 & \sqrt{2}\\ 0 & 2 & \sqrt{2}\end{bmatrix}$

Check: Q should be semi-orthogonal, so $Q^T Q = I$

5. Let $R = \begin{bmatrix}(e_1 \cdot a_1) & (e_1 \cdot a_2) & \cdots & (e_1 \cdot a_m)\\ 0 & (e_2 \cdot a_2) & \cdots & (e_2 \cdot a_m)\\ \vdots & 0 & \ddots & \vdots\\ 0 & \cdots & 0 & (e_m \cdot a_m)\end{bmatrix}$

$$\begin{bmatrix}e_1 \cdot a_1 & e_1 \cdot a_2 & e_1 \cdot a_3\\ 0 & e_2 \cdot a_2 & e_2 \cdot a_3\\ 0 & 0 & e_3 \cdot a_3\end{bmatrix} = \begin{bmatrix}\sqrt{2} & \sqrt{2}/2 & \sqrt{2}/2\\ 0 & \sqrt{6}/2 & \sqrt{6}/6\\ 0 & 0 & \sqrt{3}/3\end{bmatrix}$$

6. So, decomposition is $A = QR = \frac{\sqrt{6}}{6}\begin{bmatrix}\sqrt{3} & 1 & -\sqrt{2}\\ \sqrt{3} & -1 & \sqrt{2}\\ 0 & 2 & \sqrt{2}\end{bmatrix}\begin{bmatrix}\sqrt{2} & \sqrt{2}/2 & \sqrt{2}/2\\ 0 & \sqrt{6}/2 & \sqrt{6}/6\\ 0 & 0 & \sqrt{6}/6\end{bmatrix}$

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 $X_{n+1} = \frac{AX_n}{\|AX_n\|}$

Take arbitrary $x_0 = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}$. We keep iterating and computing X_{n+1}

and eventually $x_{15} = \begin{pmatrix}-0.97\\-1\\0.97\\1\end{pmatrix}$ 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 $C = \frac{1}{2}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$. 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 = $\begin{bmatrix}5 & 2\\ 3 & 4\end{bmatrix}$ ∈ $\mathbb{R}^{2\times 2}$. We work in $\mathbb{R}^2$ 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||$_1$ 3/4 < 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

$(A - \lambda_2 I)v_{2,1}^2 = v_{2,1}^1$, $\begin{bmatrix}-1 & 2 & 1\\ -7 & 3 & 2\\ 5 & 0 & -1\end{bmatrix}v_{2,1}^2 = \begin{bmatrix}1\\-1\\5\end{bmatrix}$, $v_{2,1}^2 = \begin{bmatrix}0\\3\\-5\end{bmatrix}$

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:

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

$$A^T A = \begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -1 & 1 & -1\\ -1 & 1 & -1 & 1\end{bmatrix}\begin{bmatrix}1 & 1 & -1\\ 1 & -1 & 1\\ 1 & 1 & -1\\ 1 & -1 & 1\end{bmatrix} = \begin{bmatrix}4 & 0 & 0\\ 0 & 4 & -4\\ 0 & -4 & 4\end{bmatrix}$$

$$\det(A^T A - \lambda I) = 0$$

$$\det\begin{bmatrix}4-\lambda & 0 & 0\\ 0 & 4-\lambda & -4\\ 0 & -4 & 4-\lambda\end{bmatrix} = (4-\lambda)((4-\lambda)(4-\lambda) - 4^2)$$

$$= (4-\lambda)(\lambda^2 - 8\lambda) = -\lambda^3 + 12\lambda^2 - 32\lambda = -\lambda(\lambda - 4)(\lambda - 8)$$

$$\lambda_1 = 8, \lambda_2 = 4, \lambda_3 = 0$$

$$S = \begin{bmatrix}2\sqrt{2} & 0 & 0\\ 0 & 2 & 0\\ 0 & 0 & 0\end{bmatrix}$$

$$(A^T A - \lambda_1 I)v_1 = 0$$

$$\begin{bmatrix}-4 & 0 & 0\\ 0 & -4 & -4\\ 0 & -4 & -4\end{bmatrix}v_1, v_1 = \begin{bmatrix}0\\ \sqrt{2}/2\\ -\sqrt{2}/2\end{bmatrix}$$

$$(A^T A - \lambda_2 I)v_2 = 0$$

$$\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & -4\\ 0 & -4 & 0\end{bmatrix}v_2 = 0, v_2 = \begin{bmatrix}1\\0\\0\end{bmatrix}$$

$$(A^T A - \lambda_3 I)v_3 = 0$$

$$\begin{bmatrix}4 & 0 & 0\\ 0 & 4 & -4\\ 0 & -4 & 4\end{bmatrix}v_3 = 0, v_3 = \begin{bmatrix}0\\ \sqrt{2}/2\\ \sqrt{2}/2\end{bmatrix}$$

$$V = [v_1, v_2, v_3] = \begin{bmatrix}0 & 1 & 0\\ \sqrt{2}/2 & 0 & \sqrt{2}/2\\ -\sqrt{2}/2 & 0 & \sqrt{2}/2\end{bmatrix}$$

$$u_1 = \frac{1}{\sigma_1}Av_1 = \frac{1}{2\sqrt{2}}\begin{bmatrix}1 & 1 & -1\\ 1 & -1 & 1\\ 1 & 1 & -1\\ 1 & -1 & 1\end{bmatrix}\begin{bmatrix}0\\ \sqrt{2}/2\\ -\sqrt{2}/2\end{bmatrix} = \frac{1}{2\sqrt{2}}\begin{bmatrix}\sqrt{2}\\ -\sqrt{2}\\ \sqrt{2}\\ -\sqrt{2}\end{bmatrix} = \frac{1}{2}\begin{bmatrix}1\\-1\\1\\-1\end{bmatrix}$$

$$u_2 = \frac{1}{\sigma_2}Av_2 = \frac{1}{2}\begin{bmatrix}1 & 1 & -1\\ 1 & -1 & 1\\ 1 & 1 & -1\\ 1 & -1 & 1\end{bmatrix}\begin{bmatrix}1\\0\\0\end{bmatrix} = \frac{1}{2}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}$$

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

$$u_4 = \text{normalise}(y - \text{proj}_{u_1}(y) - \text{proj}_{u_2}(y) - \text{proj}_{u_3}(y))$$

$$= \text{normalise}\left(\begin{bmatrix}0\\0\\0\\1\end{bmatrix} - \left(\frac{1}{2}\begin{bmatrix}1\\-1\\1\\-1\end{bmatrix}\cdot\begin{bmatrix}0\\0\\0\\1\end{bmatrix}\right)\left(\frac{1}{2}\begin{bmatrix}1\\-1\\1\\-1\end{bmatrix}\right) - \left(\frac{1}{2}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}\cdot\begin{bmatrix}0\\0\\0\\1\end{bmatrix}\right)\left(\frac{1}{2}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}\right) - \left(\begin{bmatrix}\sqrt{2}/2\\0\\-\sqrt{2}/2\\0\end{bmatrix}\cdot\begin{bmatrix}0\\0\\0\\1\end{bmatrix}\right)\left(\begin{bmatrix}\sqrt{2}/2\\0\\-\sqrt{2}/2\\0\end{bmatrix}\right)\right)$$

$$= \text{normalise}\left(\begin{bmatrix}0\\0\\0\\1\end{bmatrix} + \frac{1}{4}\begin{bmatrix}1\\-1\\1\\-1\end{bmatrix} - \frac{1}{4}\begin{bmatrix}1\\1\\1\\1\end{bmatrix}\right) = \text{normalise}\left(\begin{bmatrix}0\\-1/2\\0\\1/2\end{bmatrix}\right) = \frac{\sqrt{2}}{2}\begin{bmatrix}0\\-1\\0\\1\end{bmatrix}$$

$$U = \begin{bmatrix}1/2 & 1/2 & \sqrt{2}/2 & 0\\ -1/2 & 1/2 & 0 & -\frac{\sqrt{2}}{2}\\ 1/2 & 1/2 & -\sqrt{2}/2 & 0\\ -1/2 & 1/2 & 0 & \frac{\sqrt{2}}{2}\end{bmatrix}$$

$$A = \begin{bmatrix}1/2 & 1/2 & \sqrt{2}/2 & 0\\ -1/2 & 1/2 & 0 & -\sqrt{2}/2\\ 1/2 & 1/2 & -\sqrt{2}/2 & 0\\ -1/2 & 1/2 & 0 & \sqrt{2}/2\end{bmatrix}\begin{bmatrix}2\sqrt{2} & 0 & 0\\ 0 & 2 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}0 & \sqrt{2}/2 & -\sqrt{2}/2\\ 1 & 0 & 0\\ 0 & \sqrt{2}/2 & \sqrt{2}/2\end{bmatrix}^T$$

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 $\begin{bmatrix}f_{xx}(a, b) & f_{xy}(a, b)\\ f_{yx}(a, b) & f_{yy}(a, b)\end{bmatrix}$ 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 $\lambda_2 \neq \lambda_1$. An eigenvector x_2 of A corresponding to eigenvalue λ_2 is: $x_2 = H^{-1}\begin{bmatrix}\beta\\ z_2\end{bmatrix}$, $\beta = \frac{b^T z_2}{\lambda_2 - \lambda_1}$ 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 - represents** the reflection through a hyperplane with normal vector u.

First, we need to construct it:

$$H = I - \frac{2uu^T}{u^T u}$$

We take $x_1 = \begin{bmatrix}1\\1\\-1\\-1\end{bmatrix}$, which is a vector in the eigenspace of our biggest eigenvalue

u = x_1 + $\|x_1\|_2 e_1$

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, β = $b^T z_2$ / λ_2-λ_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: $\mathbf{D_u f(x, y) = f_x(x, y)cos\theta + f_y(x, y)sin\theta}$
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

$D_u f(x, y)$ at (1,2) where u is the unit vector at an angle $\theta = \pi/6$:

1) D_uf(x, y) = f_x(x, y)cosθ+f_y(x, y)sinθ
D_uf(x, y) = f_x(x, y)cosπ/6 +
f_y(x, y)sinπ/6 =
($3x^2$ - 3y) √3/2 + (-3x + 8y) 1/2 =
= 1/2[3√3x^2 - 3x + (8 - 3√3)y]
Subbing (1, 2) we get (13 - 3√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 ∇f(2,-1)=-4i+8j

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:

Max($D_u f$) = Max(∇f·u) = Max(|∇f||u|cosθ) = Max(|∇f| cos θ) = |∇f|.

The max value of cos is 1, at θ = 0. So the max value of $D_u f$ is |∇f| when θ = 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 Quadratics

1) Compute the gradient function, g^k = ∇f(x), and H which is computed in (Determining whether a Function has a Critical point) by taking second derivatives wrt specific variables in specific orders. The steepest descent vector is - ∇f(x), ie: if we have variables x_0 and x_1, compute the derivatives wrt both. Each derivative with the initial value subbed in corresponds to an element of our steepest descent vector.

2) Sub in the x^0 into the gradient function, then compute a_k = $(g^k)^T(g^k)$ / $((g^k)^T)^T$Hg^k.

3) Then generate x_1 as x_0 - a_0 ∇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= ∇f(x) = [$2x_1$, $2x_2$+2]^T

H = ∇^2f(x) = [2 0, 0 2]

The gradient at x^0=[2(2), 2(1)+2]=[4, 2]^T

The step length a = $(g^0)^T(g^0)$ / $((g^0)^T)^T$Hg^0 = [4 4][4 4]^T / [4 4][2 0, 0 2][4 4]^T = 32/64.

X^1 = [2 1]^T - ½[2 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) |∇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

To find the number of iterations to reach the minimum, we just keep computing the next iteration, until we have satisfied our conditions for a minimum.

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 ∇F(x) = 0, ie: solving Ax-b = 0.

A-conjugate direction: (d_0, ... 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, ∀i ≠ j.

d^T_iAd_i is also written as <d_i, d_j>$_A$ = <d_iAd_j> = <Ad_i, d_j> = <d_i, $A^T d_j$>

These directions d_0 ... $d_{n-1}$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 $r_0^T r_0$ / d_1^TAd_1
3) Now x_1 can be computed with the formula: x_1 = x_0 + $a_1 d_1$
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 β_2 = -$r_1^T r_1$ / $r_0^T r_0$
6) Use β_2 to compute the next search direction d_2 = r_1 - $\beta_2 d_1$
7) Calculate a_2 using d_2 = $r_1^T r_1$ / d_2^TAd_2
8) Find x_2 = x_1 + $a_2 d_2$
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 = $r_0^T r_0$ / 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 - $B_2 d_1$ = $\begin{bmatrix}-1.6\\ 1.6\end{bmatrix}$ + 0.04[-8 -8] = [-1.92 1.28]

7) a_2 =$r_1^T r_1$ / 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 + $a_2 d_2$ = $\begin{bmatrix}0.9333\\ 0.0667\end{bmatrix}$ + 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-0.