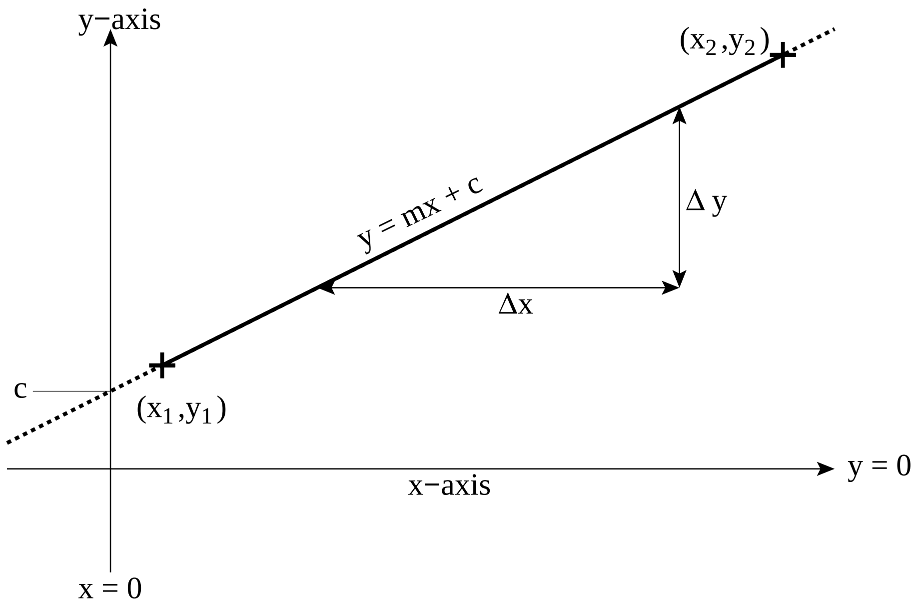
Session Objectives

- Define vector equations for straight lines and circles
- Compute the intersection of two lines
- Define matrices
- Compute standard matrix operations
- Define and use matrices in Ada

Vector Geometry

- Vectors are a convenient way of defining geometric shapes.
- The simplest place to start is a straight line. This is extremely useful in graphics to define straight edges of objects and paths of motion.
- We saw eariler that the equation of a straight line could be written as $y = mx + c$, where c is the intercept and m is the gradient.



- The gradient of the line, $\Delta y/\Delta x$ gives the ratio of the increase in y to the increase in x.
- A line is straight if and only if the gradient is constant. Given two points on the line, the gradient of the line joining them is $(y_2 - y_1)/(x_2 - x_1)$.
- For a general point (x, y) to be on the line, the gradient between (x_1, y_1) and (x, y) must equal this

  $$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1} \Leftrightarrow \frac{y - y_1}{y_2 - y_1} = \frac{x - x_1}{x_2 - x_1}.$$

  The second form is more convenient because the two variables x and y are on different sides of the equation.

Exercise

Find the equation of the straight line through the points $(2, 4)$ and $(0, 6)$; rewrite it in the form $y = mx + c$.

Solution

Let $(x_1, y_1) = (2, 4)$ and $(x_2, y_2) = (0, 6)$. The the equation of the straight line is

$$\frac{y-4}{6-4} = \frac{x-2}{0-2}$$

$$\Leftrightarrow \frac{y-4}{2} = \frac{x-2}{-2}$$

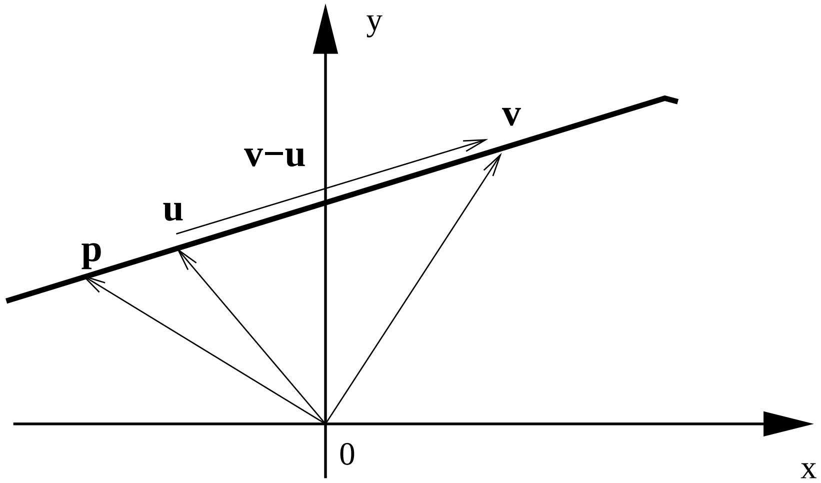
We can rewrite this as follows:

$$\frac{y-4}{2} = \frac{x-2}{-2}$$

$$\Rightarrow y - 4 = -x + 2$$

$$\Rightarrow y = -x + 6$$

Vector Equation of Straight Line



$\mathbf{u} = (x_1, y_1)$, $\mathbf{v} = (x_2, y_2)$, and $\mathbf{p}$ denotes the general point on the line joining $\mathbf{u}$ and $\mathbf{v}$.

- The direction of the line is given by the vector $\mathbf{v} - \mathbf{u}$.
- The general point on the line, $\mathbf{p}$, is given by $\mathbf{u}$ plus some scalar multiple, t say, of $\mathbf{v} - \mathbf{u}$.
- The vector equation of the line can be written as

$$\mathbf{p} = \mathbf{u} + t(\mathbf{v} - \mathbf{u}) = (1 - t)\mathbf{u} + t\mathbf{v}.$$

- In two dimensions, this can be written as

$$\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} (1-t)u_1 + tv_1 \\ (1-t)u_2 + tv_2 \end{bmatrix}$$

Parametric Equations

This type of equation is called a parametric form, since it defines how to generate points on the line by a parameter (in this case, t). The parametric form has several advantages:

- defining line segments (i.e. pieces of line) by constraining the value of t (e.g. between 0 and 1);
- equations of motion, when t represents time, and $\mathbf{u}$ is the initial position of the point.
- efficiency of representation. Every point on a line segment can be generated from knowledge of the two vectors $\mathbf{u}$ and $\mathbf{v}$ and the range of possible values of t. This is a total of six numbers. Representing the same line segment as a set of discrete points (as in a bitmap) requires two numbers per point, which will generally be many more values.

Exercise

Work out the vector equation of the line through the points $(2, 4)$ and $(0, 6)$. Show that this reduces to the same equation as in the first Exercise.

Solution

$$\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} (1-t)u_1 + tv_1 \\ (1-t)u_2 + tv_2 \end{bmatrix}$$

$$= \begin{bmatrix} (1-t) \times 2 + t \times 0 \\ (1-t) \times 4 + t \times 6 \end{bmatrix}$$

$$= \begin{bmatrix} 2 - 2t \\ 4 + 2t \end{bmatrix}$$

As a check, putting $t = 0$ gives $(2, 4)$ and $t = 1$ gives $(0, 6)$.

Putting $y = 4 + 2t$ and $x = 2 - 2t$, we see that $-x = 2t - 2$ so $y = -x + 6$.

Vector Equation of a Circle

- Suppose that the circle has centre $\mathbf{c}$ and radius r.
- Then the set of points on the circle all satisfy the property that they are at distance r from $\mathbf{c}$, which can be written as

$$\|\mathbf{p} - \mathbf{c}\| = r.$$

- This is a quadratic function.

Finding Points of Intersection

- Vector equations can also be used to find the point of intersection of lines: this is useful to find the corners of geometric objects.
- If l_1 is defined by $(1-t)\mathbf{u}+t\mathbf{v}$ and l_2 is defined by $(1-s)\mathbf{w}+s\mathbf{x}$, then they meet when these two vectors are the same, that is, when

$$\begin{bmatrix}(1-t)u_1+tv_1\\(1-t)u_2+tv_2\end{bmatrix}=\begin{bmatrix}(1-s)w_1+sx_1\\(1-s)w_2+sx_2\end{bmatrix}$$

  This gives us two equations in two unknowns (s and t), which generally has a unique solution.

- It can only fail if the two lines are parallel, when there are either no solutions (the lines never meet) or an infinite number (the two lines are the same).

Example

Find the point of intersection of l_1, which passes through $(2,4)$ and $(0,6)$, and l_2, which passes through $(5,4)$ and $(1,2)$.

1. The parametric form of l_1 is $(2-2t, 4+2t)$. The parametric form of l_2 is $(5(1-s)+s, 4(1-s)+2s) = (5-4s, 4-2s)$.
2. At the point of intersection, the coordinates of l_1 and l_2 are equal:

$$2-2t = 5-4s$$

$$4+2t = 4-2s$$

3. Adding both equations gives

$$6 = 9-6s \Leftrightarrow 6s = 3 \Leftrightarrow s = \frac{1}{2}.$$

   Substituting this back into l_2, we find that the point of intersection is $(5-4\times(1/2), 4-2\times(1/2)) = (3,3)$.

Checking the Answer

As a check, we can substitute the value $s = 1/2$ back into one of the original equations to find t:

$$\begin{aligned} 2 - 2t &= 5 - 4 \times \frac{1}{2} \\ \Leftrightarrow 2 - 2t &= 3 \\ \Leftrightarrow 2t &= -1 \\ \Leftrightarrow t &= \frac{-1}{2}. \end{aligned}$$

This gives a point on l_1 which is $(3, 3)$, the same as on l_2.

Summary

1. A vector is an n-tuple of numbers representing a point in an n-dimensional space.
2. The following vector operations are defined: addition, subtraction, scalar multiplication, dot product, length.
3. A unit vector is a vector of length 1. The unit vector in the direction of $\mathbf{v}$ is given by $\mathbf{v}/\|\mathbf{v}\|$.
4. We can also use the dot product to calculate the angle between two vectors. This gives us a geometric interpretation of the dot product in terms of projecting one vector $\mathbf{v}$ onto a unit vector $\mathbf{u}$.
5. The most appropriate Ada data structure to represent a vector is the unconstrained array.
6. The vector equation of a line is $\mathbf{p} = \mathbf{u} + t(\mathbf{v} - \mathbf{u}) = (1 - t)\mathbf{u} + t\mathbf{v}$. This is a parametric equation.
7. The vector equation of a circle is $\|\mathbf{p} - \mathbf{c}\| = r$.

Matrices

- Matrices occur in many practical applications in computing: graphics, error correcting codes, data encryption, and simulation.
- An $m \times n$ matrix is a rectangular array with m rows and n columns; m and n are positive integers. The matrix is enclosed by square (or round) brackets.
- In general we shall use an upper case bold letter to denote a matrix, and the corresponding lower case letter to denote its entries. Here $\mathbf{A}$ is a 3×2 matrix:

$$\mathbf{A} = \begin{bmatrix} 2 & 3 \\ 1 & 7 \\ 0 & 42 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}.$$

  The element in row i and column j of $\mathbf{A}$ is denoted by a_{ij}. So, in the example, $a_{21} = 1$.

Special Matrices

- If $\mathbf{A}$ is an $n \times n$ matrix, we say that $\mathbf{A}$ is a square matrix of order n.
- A $1 \times n$ matrix is called a row vector of order n, while a $m \times 1$ matrix is called a column vector of order m.
- When referring to the elements of a column vector, the column number is always 1 and is often omitted. So the element in row i of a column vector $\mathbf{U}$ is usually denoted by u_i rather than u_{i1}.
- The elements of a matrix can be drawn from any of the number systems we have seen so far. For example, if the elements of the matrix are real numbers, we talk of a matrix over $\mathbb{R}$.

Matrix Operations

Equality Two matrices $\mathbf{A}$ and $\mathbf{B}$ are equal if and only if they are the same size and contain the same elements. So $\mathbf{A} = \mathbf{B}$ if and only if the number of rows in A equals the number of rows in $\mathbf{B}$, the number of columns in A equals the number of columns in $\mathbf{B}$, and $a_{ij} = b_{ij}$ for $1 \leq i \leq m$ and $1 \leq j \leq n$.

Addition and Subtraction We can add or subtract two matrices $\mathbf{A}$ and $\mathbf{B}$ only if they are the same size. The operations are carried out element-by-element. If $\mathbf{C} = \mathbf{A} + \mathbf{B}$, then $c_{ij} = a_{ij} + b_{ij}$. Similarly, if $\mathbf{D} = \mathbf{A} - \mathbf{B}$, then $d_{ij} = a_{ij} - b_{ij}$. For example, if

$$\mathbf{A} = \begin{bmatrix} 11 & -4 & -7 \\ 7 & -2 & -5 \\ 10 & -4 & -6 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} -3 & -7 & -5 \\ 2 & 4 & 3 \\ 1 & 2 & 2 \end{bmatrix}$$

then

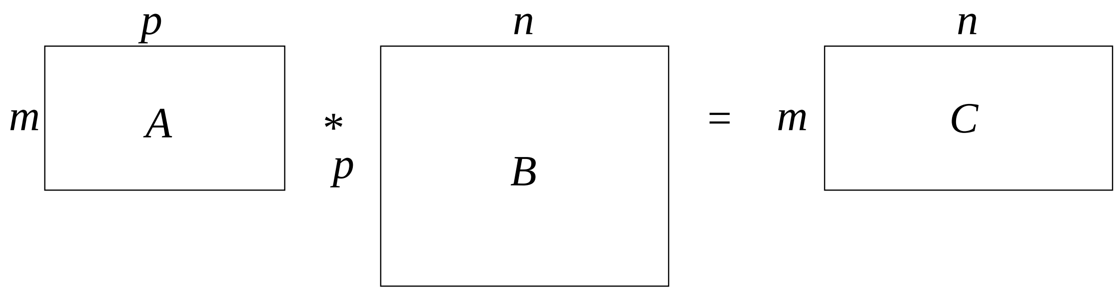
$$\mathbf{C} = \mathbf{A} + \mathbf{B} = \begin{bmatrix} 8 & -11 & -12 \\ 9 & 2 & -2 \\ 11 & -2 & -4 \end{bmatrix} \quad \text{and} \quad \mathbf{D} = \mathbf{A} - \mathbf{B} = \begin{bmatrix} 14 & 3 & -2 \\ 5 & -6 & -8 \\ 9 & -6 & -8 \end{bmatrix}$$

Scalar Multiplication

- Multiplication of a matrix $\mathbf{A}$ by a scalar t produces a matrix of the same size whose elements are each multiplied by t.
- Thus if $\mathbf{A}$ is an $m \times n$ matrix, then $\mathbf{B} = t\mathbf{A}$ is also $m \times n$ and $b_{ij} = ta_{ij}$, for $1 \leq i \leq m$ and $1 \leq j \leq n$.
- For example,

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 6 & 5 & -4 \end{bmatrix} \Longrightarrow 3\mathbf{A} = \begin{bmatrix} 3 & 6 & 9 \\ 18 & 15 & -12 \end{bmatrix}.$$

Matrix Multiplication



- Suppose that $\mathbf{A}$ and $\mathbf{B}$ are $m \times p$ and $q \times n$ matrices respectively.
- The matrix product $\mathbf{AB}$ is only defined if $p = q$: that is, if the number of columns in $\mathbf{A}$ is equal to the number of rows in $\mathbf{B}$. We say that $\mathbf{A}$ and $\mathbf{B}$ are conformant.
- In this case, the matrix $\mathbf{C}$ has m rows and n columns and element c_{ij} is defined as

$$c_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj} = a_{i1} b_{ij} + a_{i2} b_{2j} + \cdots + a_{ip} b_{pj}.$$

Example

- One way to remember this is as

  Along the landing and down the stairs.

- If $\mathbf{A} = \begin{bmatrix} 2 & 3 & 4 \\ -1 & -2 & 5 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} -3 & -7 & -5 \\ 2 & 4 & 3 \\ 1 & 2 & 2 \end{bmatrix}$ then

$$\mathbf{C} = \mathbf{AB} = \begin{bmatrix} 4 & 6 & 7 \\ 4 & 9 & 9 \end{bmatrix}.$$

- However, $\mathbf{BA}$ is not defined, since the number of columns in $\mathbf{B}$ is three and this is not equal to the number of rows in $\mathbf{A}$, which is 2.

Order Matters

- Note that even if both $\mathbf{A}$ and $\mathbf{B}$ are square matrices of the same size (so that both $\mathbf{AB}$ and $\mathbf{BA}$ are defined and have the same size) it is not usually true that $\mathbf{AB} = \mathbf{BA}$; we say that matrix multiplication is not commutative.
- This is quite different from multiplication of numbers.
- For example, if

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

then

$$\mathbf{C} = \mathbf{AB} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix} \quad \text{and} \quad \mathbf{D} = \mathbf{BA} = \begin{bmatrix} 23 & 34 \\ 31 & 46 \end{bmatrix}$$

Special Matrices

Zero Matrix The zero matrix $\mathbf{0}_{mn}$ is the $m \times n$ matrix with all its elements equal to zero. The 2×2 zero matrix is $\mathbf{0}_{22} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$. If $\mathbf{A}$ is any $m \times n$ matrix, then the following properties of the zero matrix can be proved very easily.

$$\mathbf{A} + \mathbf{0}_{mn} = \mathbf{0}_{mn} + \mathbf{A} = \mathbf{A}, \quad \mathbf{A}\mathbf{0}_{np} = \mathbf{0}_{mp}, \quad \mathbf{0}_{qm}\mathbf{A} = \mathbf{0}_{qn}.$$

Identity Matrix The identity matrix $\boldsymbol{I}_n$ is a square matrix or order n with ones on the main diagonal (from top left to bottom right) and zeros elsewhere. The 2×2 identity matrix is

$$\boldsymbol{I}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Multiplying a matrix by an identity matrix of the appropriate size leaves the matrix unchanged. If $\mathbf{A}$ is an $m \times n$ matrix, then

$$\mathbf{A}\boldsymbol{I}_n = \mathbf{A} \quad \text{and} \quad \boldsymbol{I}_m\mathbf{A} = \mathbf{A}.$$

Matrix Operations

Matrix Transpose The transpose of a matrix $\mathbf{A}$, which is denoted by either $\mathbf{A}'$ or $\mathbf{A}^T$, is obtained by swapping the rows and columns of the matrix. For example,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \quad \Longrightarrow \quad \mathbf{A}' = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ a_{13} & a_{23} \end{bmatrix}$$

Matrix Division Instead of defining matrix division, we define a (multiplicative) inverse matrix $\mathbf{A}^{-1}$ with the property that $\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$. Multiplying by $\mathbf{A}^{-1}$ is equivalent to dividing by $\mathbf{A}$. Note that $\mathbf{A}$ must be square for this definition to make sense; $\mathbf{A}$ must also have some other properties that we won't go into here. The formula for computing $\mathbf{A}^{-1}$ is quite complicated, and we won't cover it here.

Matrices in Ada

There is no direct support for matrix algebra in Ada. However, it is relatively easy to define a suitable unconstrained two-dimensional array type `MATRIX`:

```
SUBTYPE Elem_Type IS ....; -- Float, Integer, or a Modular type
TYPE Matrix IS
   ARRAY Matrix(Positive RANGE <>, Positive RANGE <>) OF Elem_Type;

-- As necessary, define subtypes for matrices of various sizes, e.g.
SUBTYPE ThreeByTwo IS Matrix(1..3, 1..2);
SUBTYPE Square3 IS Matrix(1..3, 1..3);
SUBTYPE RowVector3 IS Matrix(1..1, 1..3);
SUBTYPE ColumnVector3 IS Matrix(1..3, 1..1);

A : Square3 := ((1, 2, 3),
                (4, 5, 6),
                (7, 8, 9));    -- 2-D (3x3) array aggregate
B : RowVector3 := ((9, 8, 7)); -- 2-D (1x3) array aggregate
```

Note the use of array aggregates to initialise matrices as they are declared.

Matrix Operations in Ada

```
FUNCTION Add(A, B: Matrix) RETURN Matrix IS
   C : Matrix(A'Range(1), A'Range(2));  -- same size as A
BEGIN
   -- First check that sizes of A and B are the same
   IF A'Length(1) /= B'Length(1) OR
       A'Length(2) /= B'Length(2) THEN
       RAISE Size_Error;    -- where Size_Error : EXCEPTION
   END IF;

   -- Now add corresponding elements and store in C
   FOR Row in C'Range(1) LOOP
      FOR Col in C'Range(2) LOOP
         C(Row, Col) := A(Row, Col) + B(Row, Col);
      END LOOP;
   END LOOP;

   RETURN C;
END Add;
```

Best to place all these definitions together in an Ada package with other functions and operators implementing scalar and matrix multiplication, matrix subtraction etc., plus declarations of exceptions such as `Size_Error`.

Summary: Matrix Algebra

1. Matrices occur in many practical applications in computing: graphics, error correcting codes, data encryption, and simulation.
2. An $m \times n$ matrix is a rectangular array with m rows and n columns; clearly m and n are positive integers.
3. The following matrix operations are defined: equality; addition and subtraction; scalar multiplication; matrix multiplication (along the landing and down the stairs); transpose.
4. The zero matrix has all elements zero. The identity matrix $\boldsymbol{I}_n$ is a square matrix or order n with ones on the main diagonal (from top left to bottom right) and zeros elsewhere.
5. Matrix division (for square matrices only) is defined in terms of multiplying by a matrix inverse $\mathbf{A}^{-1}$. This is complicated and has been left out.
6. An unconstrained two-dimensional array is the best way to represent matrices in Ada.

Session Objectives

- Define vector equations for straight lines and circles
- Compute the intersection of two lines
- Define matrices
- Compute standard matrix operations
- Define and use matrices in Ada