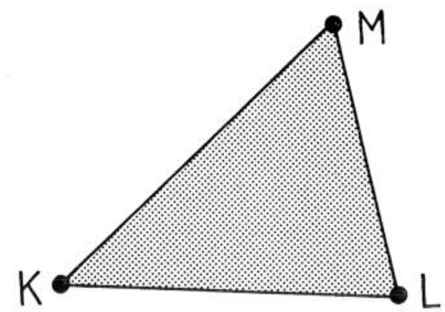
4

Areas

4.1 Area of a Triangle



The triangle is particularly useful in computer geometry because it is simply and unambiguously described by the coordinates of its three vertices. It also has the merit of always being planar, even when the vertices are points in space. It is therefore common to break down two-dimensional shapes and three-dimensional surfaces into triangles, so that programs working with these structures have only one element to handle.

The area of a triangle is found by evaluating the determinant (see Section 6.3)

$$\frac{1}{2}\begin{vmatrix} (x_L - x_K) & (x_M - x_K) \\ (y_L - y_K) & (y_M - y_K) \end{vmatrix}$$

```
XLK = XL - XK
XMK = XM - XK
YLK = YL - YK
YMK = YM - YK

AREA = 0.5*(XLK*YMK - XMK*YLK)
AREA = ABS(AREA)
```

If the absolute value of the area is not taken, its sign indicates the relationship between point K (in general, the subtracted point) and the line segment defined by the other two points. If the determinant is positive, K, L, and M are in anti-clockwise order; if it is negative they are clockwise. Thus without the call to the ABS function and without the factor of 0.5 this code gives a quick way to decide on which side of a line segment a given point lies.

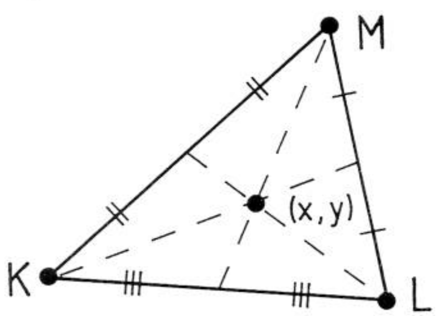
An alternative area formula is in terms of the side lengths

$$\sqrt{[s(s - r_{KL})(s - r_{LM})(s - r_{MK})]}$$

where r_{KL} and so on are the side lengths, and s is half the the length of the triangle perimeter.

To find the area of a triangle in three dimensions use the formula given at the end of Section 8.2.

4.2 Centre of Gravity of a Triangle

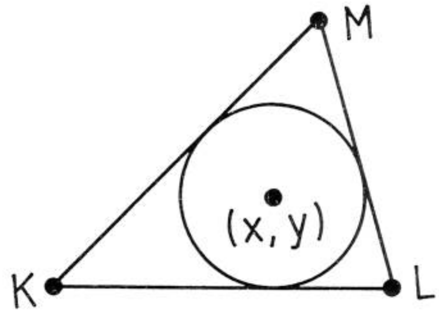


The centroid of a triangle lies at the meeting point of lines drawn from each vertex to the mid-point of the opposite side. It is, however, simply calculated as the average of the vertex coordinates:

$$x_{CG} = \frac{x_K + x_L + x_M}{3}$$

$$y_{CG} = \frac{y_K + y_L + y_M}{3}$$

4.3 Incentre of a Triangle



The inscribed circle is the largest that can be drawn inside a triangle. It is tangent to all three sides, and the lines from its centre to the vertices bisect the angle made by the triangle's sides at each vertex.

The position of the incentre is given by

$$x_{IN} = \frac{r_{LM}x_K + r_{MK}x_L + r_{KL}x_M}{t}$$

and

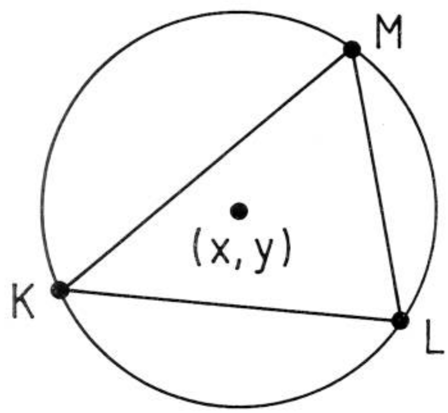
$$y_{IN} = \frac{r_{LM}y_K + r_{MK}y_L + r_{KL}y_M}{t}$$

where r_{KL} and so on are the lengths of the sides of the triangle and t is its perimeter. The radius of the inscribed circle is

$$R_{IN} = \sqrt{\left(\frac{(s - r_{LM})(s - r_{MK})(s - r_{KL})}{s}\right)}$$

where s = t/2

4.4 Circumcentre of a Triangle



The circumcentre of a triangle is the centre of the circle that passes through the vertices of the triangle. It is found by treating one vertex temporarily as the origin, when it becomes

$$x_{CC} = \frac{\begin{vmatrix} (x_{LK}^2 + y_{LK}^2) & y_{LK} \\ (x_{MK}^2 + y_{MK}^2) & y_{MK} \end{vmatrix}}{2\begin{vmatrix} x_{LK} & y_{LK} \\ x_{MK} & y_{MK} \end{vmatrix}}$$

$$y_{CC} = \frac{\begin{vmatrix} x_{LK} & (x_{LK}^2 + y_{LK}^2) \\ x_{MK} & (x_{MK}^2 + y_{MK}^2) \end{vmatrix}}{2\begin{vmatrix} x_{LK} & y_{LK} \\ x_{MK} & y_{MK} \end{vmatrix}}$$

(see Section 6.3 for a brief explanation of determinants) where

$$x_{LK} = x_L - x_K , \qquad x_{MK} = x_M - x_K$$

and $$y_{LK} = y_L - y_K , \qquad y_{MK} = y_M - y_K$$

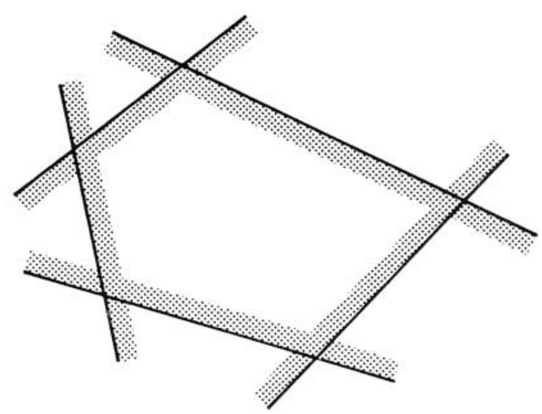
The centre given by (x_{CC}, y_{CC}) is relative to the position of K, so the coordinates of that point have to be added to get the absolute position of the centre. The relative position can be used to find the squared radius before this addition is performed:

$$r_{CC}^{2} = x_{CC}^{2} + y_{CC}^{2}$$

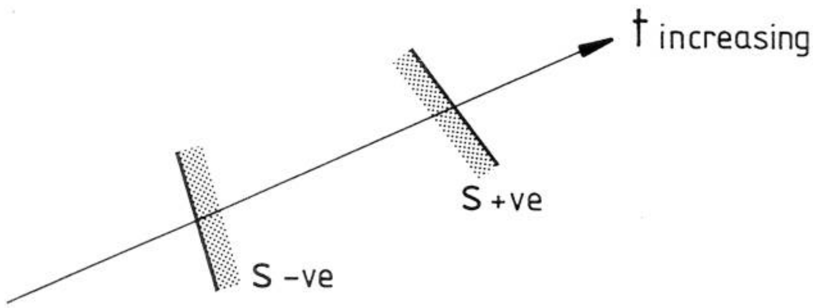
This is coded:

```
XLK = XL - XK
YLK = YL - YK
XMK = XM - XK
YMK = YM - YK
DET = XLK*YMK - XMK*YLK
IF (ABS(DET).LT.ACCY) THEN

        ..... At least two of the points are coincident

ELSE
        DETINV = 0.5/DET
        RLKSQ = XLK*XLK + YLK*YLK
        RMKSQ = XMK*XMK + YMK*YMK
        XCC = DETINV*(RLKSQ*YMK - RMKSQ*YLK)
        YCC = DETINV*(XLK*RMKSQ - XMK*RLKSQ)
        RCCSQ = XCC*XCC + YCC*YCC
        X = XCC + XK
        Y = YCC + YK
ENDIF
```

4.5 Representation of a Polygon



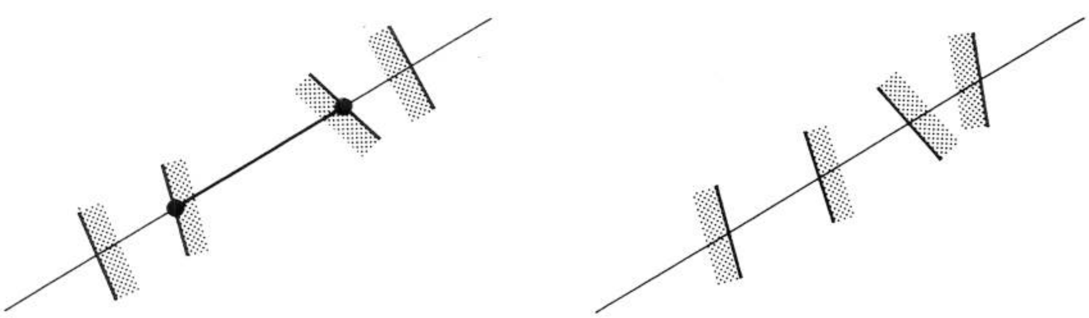
The simplest way to define a *convex* polygon (in other words a polygon where all the internal angles are less than 180°) is to consider each of the sides as a linear half-plane (see Section 1.2), with its true side pointing inwards. The polygon is the region on the true side of all the half-planes. As a consequence of this decision, it is easy to determine whether or not a given point is inside the polygon. The point coordinates are substituted into each half-plane equation in turn. If all these substitutions give a negative result (by the convention in Section 1.2), the point is inside the polygon. If any result is positive, it is outside.



The half-planes may be converted to an ordered list of vertices as follows. Each half-plane is converted to parametric form (see Section 1.2), and its intersections with the other half-planes is found (see Section 1.5). For each intersection the parameter value on the candidate line is generated. By taking the scalar product (Section 6.1) of the half-plane normal and the parametric line slope coefficients

$$s = fa + gb$$

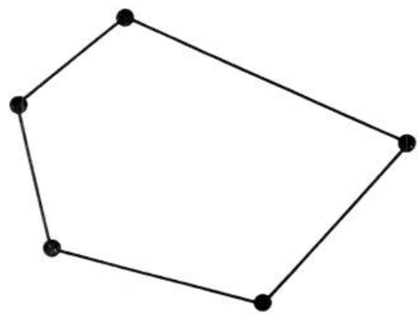
the intersections may be classified as including the *half-line* with parameter increasing or decreasing (we assume an outward pointing normal on the half-planes).



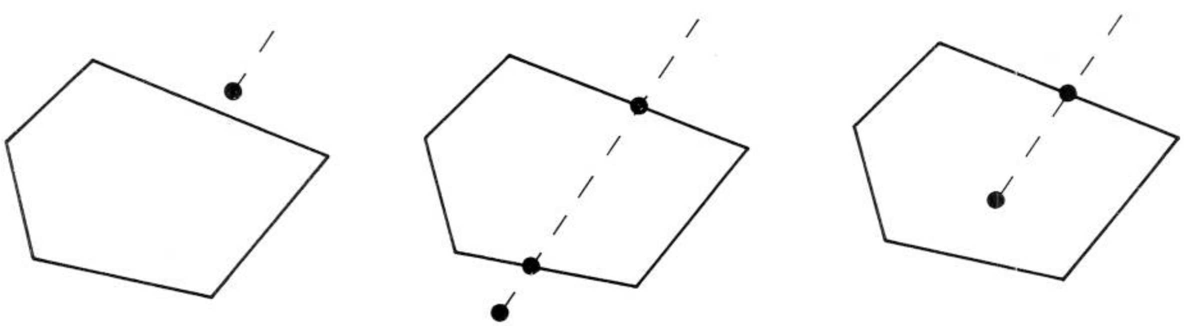
If the lowest intersection which includes the half-line with parameter decreasing is parametrically greater than the highest intersection which includes the half line with parameter decreasing, then the two intersections describe a side of the polygon. Otherwise the half-plane does not contribute to the polygon at all.

As each segment is generated, the identity of the half-plane which created each end is noted. When all the segments have been generated this information is used to trace round the segments in order, and hence to generate the list of vertex coordinates.

An ordered list of vertices is a more common and slightly more compact (two-thirds of the space) way of storing polygons, but it does not ensure that the polygon is convex. However, it is useful in many applications, the simplest being to draw the polygon.



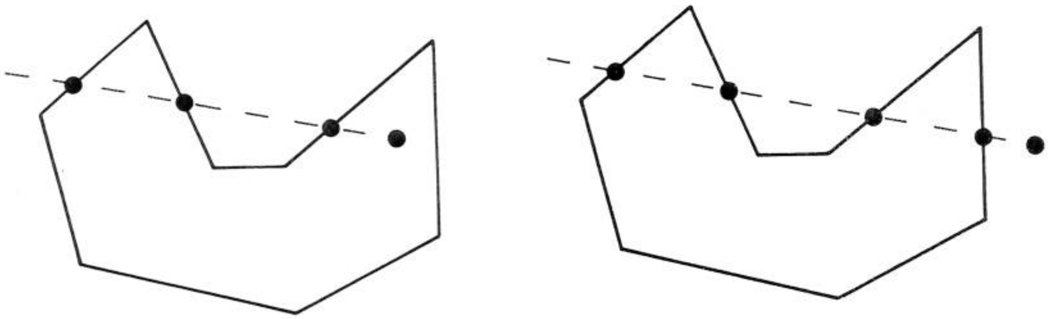
The problem of determining whether a point is inside the polygon is now more complex. It is necessary to construct a line or *ray* from the point to be tested to infinity and to determine whether this crosses the sides of the polygon. If it does not cross the sides, or crosses them twice, the point is outside. If it crosses them once, it is inside.



This test suffers accuracy problems when the ray passes near a vertex. This occurrence must be detected and a new ray chosen.

The half-plane representation of a convex polygon may be written as the set-theoretic *intersections* (∩) of the regions defined by the half-planes: each region being considered to be made up of an infinite set of points. By using the additional set-theoretic operators union (∪) and difference (⊥) any polygon can be created by adding and subtracting convex polygons.
The ordered list of vertices is easily extendible to non-convex polygons; it has already been said

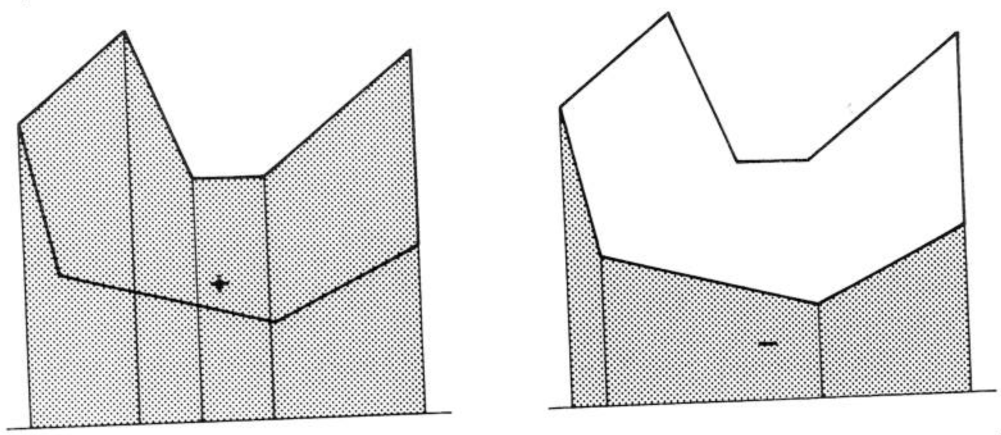
that it does not ensure convexity in any case. The ray test may be used to identify the inside and outside regions of such a polygon, odd numbers of intersections with edges indicating that the candidate point is inside the polygon, even numbers (including zero) indicating outside.



If the ordered list representation is used, care must be taken that the polygon sides do not cross each other. This condition invalidates the algorithms given in Sections 4.6 and 4.7.

4.6 Area of a Polygon

The area of any polygon represented as a vertex list may be calculated by summing the areas of the trapezia under each side, down to the axis. The direction of the sides must be taken into account, so that sides on the bottom of the polygon are subtracted from the total.



Suppose that the real arrays XVERT and YVERT hold the NVERT coordinates of a polygon's vertices, stored in anti-clockwise order. The area of the polygon may be calculated using the following code:

```
AREA = 0.0
XOLD = XVERT(NVERT)
YOLD = YVERT(NVERT)
```

```
      DO 10 N = 1, NVERT
               X = XVERT(N)
               Y = YVERT(N)
               AREA = AREA + (XOLD - X)*(YOLD + Y)
               XOLD = X
               YOLD = Y
   10 CONTINUE

      AREA = 0.5*AREA
```

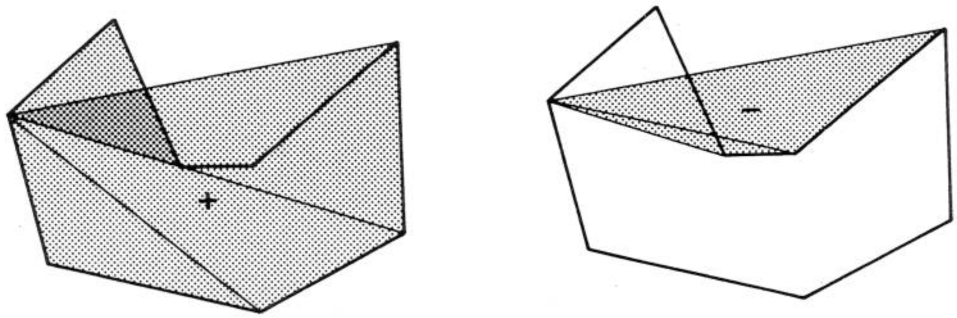
If it is not known whether the polygon was stored in anticlockwise or clockwise order then the absolute value of the area should be taken.

There is one major problem with this approach. If the polygon is a long way from the x axis then the area of the trapezia will be much larger than the area of the polygon and accuracy will be lost. Temporarily making one vertex the y origin will avoid this problem:

```
      AREA = 0.0
      XOLD = XVERT(NVERT)
      YORIG = YVERT(NVERT)
      YOLD = 0.0

      DO 10 N = 1,NVERT
               X = XVERT(N)
               Y = YVERT(N) - YORIG
               AREA = AREA + (XOLD - X)*(YOLD + Y)
               XOLD = X
               YOLD = Y
   10 CONTINUE

      AREA = 0.5*AREA
```

4.7 Centre of Gravity of a Polygon

An alternative approach to computing the area of a polygon is to take a point and construct triangles by joining the point to all the vertices of the polygon. The point need not lie in the



It so happens that this approach is computationally slightly less efficient than the trapezoidal method for area calculations, but more efficient when the centre of gravity of the polygon is required.

The formulae for the area and centre of gravity of a triangle have already been described (Sections 4.1 and 4.2). Using the same arrays, XVERT and YVERT, that were used to find areas in Section 4.6, and taking vertex NVERT as the common point, the centre of gravity of the polygon may be found as follows. In this case it is essential to know the cyclic direction of the polygon, as the centre of gravity may genuinely have negative coordinates. We again assume that the polygon vertices are stored anti-clockwise.

```
      XCG = 0.0
      YCG = 0.0
      ARESUM = 0.0
      XCOM = XVERT(NVERT)
      YCOM = YVERT(NVERT)
      XOLD = XVERT(1)
      YOLD = YVERT(1)

      NVT1 = NVERT - 1
      DO 10 N = 2, NVT1
              X = XVERT(N)
              Y = YVERT(N)
              ARETRI = (XCOM - X)*(YOLD - YCOM) +
     &                  (XOLD - XCOM)*(Y - YCOM)

              XCG = XCG + ARETRI*(X + XOLD)
              YCG = YCG + ARETRI*(Y + YOLD)
              ARESUM = ARESUM + ARETRI
              XOLD = X
              YOLD = Y
   10 CONTINUE
      IF(ARESUM.LT.ACCY) THEN
```

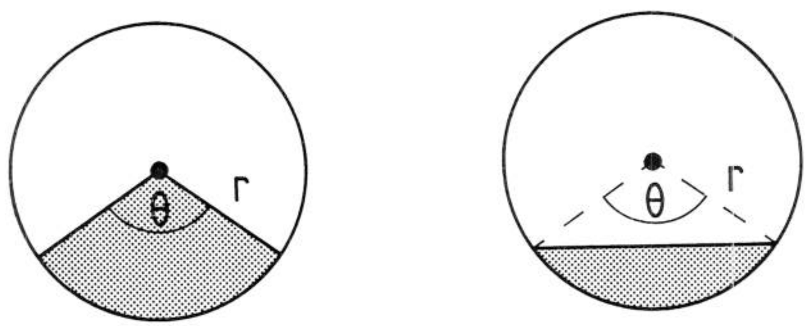
..... *The polygon covers no area or is defined clockwise*

```
ELSE
        AREINV = 1.0/ARESUM
        XCG = (XCG*AREINV + XCOM)*0.333333
        YCG = (YCG*AREINV + YCOM)*0.333333
ENDIF
```

Note that XCOM and YCOM are not added into the centre of gravity for every triangle and then divided by the area, but that this operation is performed only once at the end. ARESUM is *double* the true area of the polygon, so if both the centre of gravity and the area of a polygon are required a separate calculation for the latter is unnecessary.

4.8 Centre of Gravity of a Sector and a Segment

It is well known that the circumference of a circle is $2\pi r$, and that its area is πr^2. More interesting are the sector and segment:



The area of a segment is simply given as a fraction of the area of the circle:

$$\frac{\theta r^2}{2}$$

The area of a sector is obtained by subtracting a triangular piece from this:

$$\frac{r^2(\theta - \sin\theta)}{2}$$

The centres of gravity of both figures lie on the bisector of the central angle, by symmetry. The distance from the circle centre to the centroid is given by

$$\frac{4r\sin(\theta/2)}{3\theta}$$ for the sector

and

$$\frac{4r\sin^3(\theta/2)}{3(\theta - \sin\theta)}$$ for the segment.

Both of these formulae should be used with caution for small values of θ.