Polygonal Approximation of Boundary Integrals for Area, Centroid and Area Moment of Inertia

QUANTITY	AREA INTEGRAL $\iint \left(\frac{\partial B}{\partial y} - \frac{\partial C}{\partial x} \right) dx\, dy$	CLOCKWISE BOUNDARY INTEGRAL $\oint$ (B dx + C dy)	CLOCKWISE SUMMATIONS FOR CLOSED POLYGON n vertices $x_i\ y_i$ $\Delta x = x_{i+1} - x_i \quad \Delta y = y_{i+1} - y_i \quad x_{n+1} = x_1 \quad y_{n+1} = y_1$
Area A	$\iint dx\, dy$	$\oint (y\, dx - x\, dy) / 2$	$\sum_{i=1}^{n} (y_i \Delta x - x_i \Delta y) / 2$
First Moment A x_c about y axis	$\iint x\, dx\, dy$	$\oint (2xy\, dx - x^2\, dy) / 4$	$\sum_{i=1}^{n} (6x_i y_i \Delta x - 3x_i^2 \Delta y + 3y_i \Delta x^2 + \Delta x^2 \Delta y) / 12$
First Moment A y_c about x axis	$\iint y\, dx\, dy$	$\oint (y^2\, dx - 2xy\, dy) / 4$	$\sum_{i=1}^{n} (3y_i^2 \Delta x - 6x_i y_i \Delta y - 3x_i \Delta y^2 - \Delta x \Delta y^2) / 12$
Second Moment I_{xx} about x axis	$\iint y^2\, dx\, dy$	$\oint (y^3\, dx - 3xy^2\, dy) / 6$	$\sum_{i=1}^{n} (2y_i^3 \Delta x - 6x_i y_i^2 \Delta y - 6x_i y_i \Delta y^2 - 2x_i \Delta y^3 - 2y_i \Delta x \Delta y^2 - \Delta x \Delta y^3) / 12$
Second Moment I_{yy} about y axis	$\iint x^2\, dx\, dy$	$\oint (3x^2 y\, dx - x^3\, dy) / 6$	$\sum_{i=1}^{n} (6x_i^2 y_i \Delta x - 2x_i^3 \Delta y + 6x_i y_i \Delta x^2 + 2y_i \Delta x^3 + 2x_i \Delta x^2 \Delta y + \Delta x^3 \Delta y) / 12$
Cross Moment I_{xy}	$\iint xy\, dx\, dy$	$\oint (xy^2\, dx - x^2 y\, dy) / 4$	$\sum_{i=1}^{n} (6x_i y_i^2 \Delta x - 6x_i^2 y_i \Delta y + 3y_i^2 \Delta x^2 - 3x_i^2 \Delta y^2 + 2y_i \Delta x^2 \Delta y - 2x_i \Delta x \Delta y^2) / 24$
Perimeter P		$\oint \mathrm{sqrt}(dx^2 + dy^2)$	$\sum_{i=1}^{n} \mathrm{sqrt}(\Delta x^2 + \Delta y^2)$

Centroidal moments $\quad I_{uu} = I_{xx} - A\, y_c^2 \quad I_{vv} = I_{yy} - A\, x_c^2 \quad I_{uv} = I_{xy} - A\, x_c\, y_c \quad J' = I_{uu} + I_{vv}$

Principal moments $\quad I_1, I_2 = (I_{uu} + I_{vv}) / 2 \pm \mathrm{sqrt}[\, (I_{uu} - I_{vv})^2 / 4 + I_{uv}^2\,] \qquad \tan 2\theta = 2\, I_{uv} / (I_{vv} - I_{uu})$

EXAMPLE SUMMATION

$$\oint (2xy\,dx - x^2\,dy)/4 = \sum_{i=1}^{n} \left[\int_{x_i}^{x_{i+1}} 2xy\,dx - \int_{y_i}^{y_{i+1}} x^2\,dy \right]/4$$

for polygonal sides use $x = x_i + t\,\Delta x$ and $y = y_i + t\,\Delta y$ over $t = 0$ to 1 where $\Delta x = x_{i+1} - x_i$ and $\Delta y = y_{i+1} - y_i$

$$= \sum_{i=1}^{n} \left[2\Delta x \int_0^1 (x_i + t\,\Delta x)(y_i + t\,\Delta y)\,dt - \Delta y \int_0^1 (x_i + t\,\Delta x)^2\,dt \right]/4$$

$$= \sum_{i=1}^{n} \left[2\Delta x \int_0^1 (x_i y_i + t(\Delta x\,y_i + \Delta y\,x_i) + t^2 \Delta x \Delta y)\,dt - \Delta y \int_0^1 (x_i^2 + 2t\,x_i \Delta x + t^2 \Delta x^2)\,dt \right]/4$$

$$= \sum_{i=1}^{n} \left[2\Delta x (x_i y_i t + \frac{1}{2}(\Delta x\,y_i + \Delta y\,x_i)t^2 + \frac{1}{3}\Delta x \Delta y\,t^3)\Big|_{t=0}^{1} - \Delta y (x_i^2 t + x_i \Delta x\,t^2 + \frac{1}{3}\Delta x^2 t^3)\Big|_{t=0}^{1} \right]/4$$

$$= \sum_{i=1}^{n} \left[2\Delta x\,x_i y_i + \Delta x^2 y_i + \Delta x \Delta y\,x_i + \frac{2}{3}\Delta x^2 \Delta y - \Delta y\,x_i^2 - \Delta x \Delta y\,x_i - \frac{1}{3}\Delta x^2 \Delta y \right]/4$$

$$= \sum_{i=1}^{n} [6\Delta x\,x_i y_i + 3\Delta x^2\,y_i + \Delta x^2 \Delta y - 3\Delta y\,x_i^2]/12$$

Using boundary integrals for an object with holes

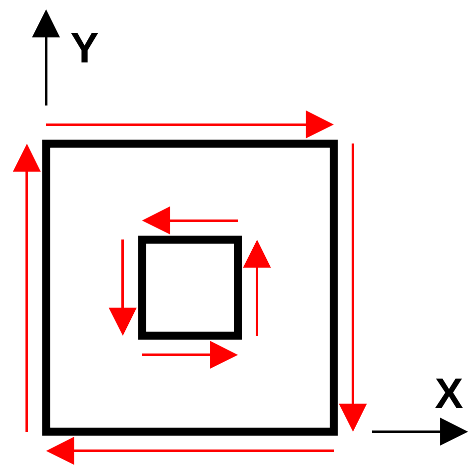
1) Digitize the outline of the object in the CW direction. Be certain to close the outline (i.e. the first and last points must be the same).
2) Digitize outlines of holes in the CCW direction. Be certain to close the outlines (i.e. the first and last points must be the same).
3) Append the data strings
4) Area, centroid and moment computations will be correct. Perimeter will NOT be correct.

Sample data for figure at right

```
x_outline  =  [  0  0  3  3  0  ];
y_outline  =  [  0  3  3  0  0  ];

x_hole  =  [  1  2  2  1  1  ];
y_hole  =  [  1  1  2  2  1  ];

x  =  [  x_outline  x_hole  ];
y  =  [  y_outline  y_hole  ];
```



MATLAB Code

```
% t_polygeom.m - test polygeom
%   area, centroid, perimeter and area moments of polygonal outline
% H.J. Sommer III - 02.05.14 - tested under MATLAB v5.2

clear

% constants
d2r = pi / 180;

% 3x5 test rectangle with long axis at 30 degrees
% area=15, x_cen=3.415, y_cen=6.549, perimeter=16
% I1=11.249, I2=31.247, J=42.496
x = [ 2.000  0.500  4.830  6.330 ]';
y = [ 4.000  6.598  9.098  6.500 ]';

% get geometry
[ geom, iner, cpmo ] = polygeom( x, y );

% show results
area = geom(1);
x_cen = geom(2);
y_cen = geom(3);
perimeter = geom(4);
disp( [ ' ' ] )
disp( [ '3x5 test rectangle with long axis at 30 degrees' ] )
disp( [ ' ' ] )
disp( [ '      area     x_cen      y_cen      perim' ] )
disp( [ area  x_cen  y_cen  perimeter ] )

I1 = cpmo(1);
angle1 = cpmo(2);
I2 = cpmo(3);
angle2 = cpmo(4);
disp( [ ' ' ] )
disp( [ '        I1         I2' ] )
disp( [ I1 I2 ] )
disp( [ '    angle1    angle2' ] )
disp( [ angle1/d2r angle2/d2r ] )

% plot outline
xplot = [ x ; x(1) ];
yplot = [ y ; y(1) ];
rad = 10;
x1 = [ x_cen-rad*cos(angle1)  x_cen+rad*cos(angle1) ];
y1 = [ y_cen-rad*sin(angle1)  y_cen+rad*sin(angle1) ];
x2 = [ x_cen-rad*cos(angle2)  x_cen+rad*cos(angle2) ];
y2 = [ y_cen-rad*sin(angle2)  y_cen+rad*sin(angle2) ];
plot( xplot,yplot,'b', x_cen,y_cen,'ro', ...
      x1,y1,'g:', x2,y2,'g:'  )
axis( [ 0  rad  0  rad ] )
axis square


% bottom of t_polygeom
```

```
function [ geom, iner, cpmo ] = polygeom( x, y )
%POLYGEOM Geometry of a planar polygon
%
%   POLYGEOM( X, Y ) returns area, X centroid,
%   Y centroid and perimeter for the planar polygon
%   specified by vertices in vectors X and Y.
%
%   [ GEOM, INER, CPMO ] = POLYGEOM( X, Y ) returns
%   area, centroid, perimeter and area moments of
%   inertia for the polygon.
%   GEOM = [ area   X_cen  Y_cen  perimeter ]
%   INER = [ Ixx    Iyy    Ixy    Iuu    Ivv    Iuv ]
%     u,v are centroidal axes parallel to x,y axes.
%   CPMO = [ I1     ang1   I2     ang2   J ]
%     I1,I2 are centroidal principal moments about axes
%         at angles ang1,ang2.
%     ang1 and ang2 are in radians.
%     J is centroidal polar moment.  J = I1 + I2 = Iuu + Ivv

% H.J. Sommer III - 02.05.14 - tested under MATLAB v5.2
%
% sample data
% x = [ 2.000  0.500  4.830  6.330 ]';
% y = [ 4.000  6.598  9.098  6.500 ]';
% 3x5 test rectangle with long axis at 30 degrees
% area=15, x_cen=3.415, y_cen=6.549, perimeter=16
% Ixx=659.561, Iyy=201.173, Ixy=344.117
% Iuu=16.249, Ivv=26.247, Iuv=8.660
% I1=11.249, ang1=30deg, I2=31.247, ang2=120deg, J=42.496
%
% H.J. Sommer III, Ph.D., Professor of Mechanical Engineering, 337 Leonhard Bldg
% The Pennsylvania State University, University Park, PA  16802
% (814)863-8997  FAX (814)865-9693  hjs1@psu.edu  www.me.psu.edu/sommer/

% begin function POLYGEOM

% check if inputs are same size
if ~isequal( size(x), size(y) ),
  error( 'X and Y must be the same size');
end

% number of vertices
[ x, ns ] = shiftdim( x );
[ y, ns ] = shiftdim( y );
[ n, c ] = size( x );

% temporarily shift data to mean of vertices for improved accuracy
xm = mean(x);
ym = mean(y);
x = x - xm*ones(n,1);
y = y - ym*ones(n,1);

% delta x and delta y
dx = x( [ 2:n 1 ] ) - x;
dy = y( [ 2:n 1 ] ) - y;

% summations for CW boundary integrals
A = sum( y.*dx - x.*dy )/2;
Axc = sum( 6*x.*y.*dx -3*x.*x.*dy +3*y.*dx.*dx +dx.*dx.*dy )/12;
Ayc = sum( 3*y.*y.*dx -6*x.*y.*dy -3*x.*dy.*dy -dx.*dy.*dy )/12;
Ixx = sum( 2*y.*y.*y.*dx -6*x.*y.*y.*dy -6*x.*y.*dy.*dy ...
          -2*x.*dy.*dy.*dy -2*y.*dx.*dy.*dy -dx.*dy.*dy.*dy )/12;
Iyy = sum( 6*x.*x.*y.*dx -2*x.*x.*x.*dy +6*x.*y.*dx.*dx ...
          +2*y.*dx.*dx.*dx +2*x.*dx.*dx.*dy +dx.*dx.*dx.*dy )/12;
Ixy = sum( 6*x.*y.*y.*dx -6*x.*x.*y.*dy +3*y.*y.*dx.*dx ...
          -3*x.*x.*dy.*dy +2*y.*dx.*dx.*dy -2*x.*dx.*dy.*dy )/24;
P = sum( sqrt( dx.*dx +dy.*dy ) );

% check for CCW versus CW boundary
if A < 0,
  A = -A;
```

```
  Axc = -Axc;
  Ayc = -Ayc;
  Ixx = -Ixx;
  Iyy = -Iyy;
  Ixy = -Ixy;
end

% centroidal moments
xc = Axc / A;
yc = Ayc / A;
Iuu = Ixx - A*yc*yc;
Ivv = Iyy - A*xc*xc;
Iuv = Ixy - A*xc*yc;
J = Iuu + Ivv;

% replace mean of vertices
x_cen = xc + xm;
y_cen = yc + ym;
Ixx = Iuu + A*y_cen*y_cen;
Iyy = Ivv + A*x_cen*x_cen;
Ixy = Iuv + A*x_cen*y_cen;

% principal moments and orientation
I = [ Iuu  -Iuv ;
     -Iuv   Ivv ];
[ eig_vec, eig_val ] = eig(I);
I1 = eig_val(1,1);
I2 = eig_val(2,2);
ang1 = atan2( eig_vec(2,1), eig_vec(1,1) );
ang2 = atan2( eig_vec(2,2), eig_vec(1,2) );

% return values
geom = [ A  x_cen  y_cen  P ];
iner = [ Ixx  Iyy  Ixy  Iuu  Ivv  Iuv ];
cpmo = [ I1  ang1  I2  ang2  J ];

% bottom of polygeom
```