

[GeneticsOfPolygons.org](https://GeneticsOfPolygons.org)

# Tangent map algorithms

## The Tangent Map Algorithm



```

TangentMapRoutines = Module[ { },
  npoints = Length[Mc]; Mc=Append[Mc,Mc[[1]]]; (*add redundant edge for routines*)

  Axy=Range[npoints]; Mxy=Axy; Bxy=Axy;

  Edge=Table[{Mc[[i]],Mc[[i+1]]},{i,1,npoints}];

  Dxy=Table[{Edge[[i,2,1]]-Edge[[i,1,1]],Edge[[i,2,2]]-Edge[[i,1,2]]},{i,npoints}];
  (*Deltax and Deltay for each edge*)

  Do[If[Abs[Dxy[[i,2]]]<=Abs[Dxy[[i,1]]],Axy[[i]]="True",Axy[[i]]="False"],{i,1,npoints}];
  (* swap x and y when slope>1*)

  Do[If[Axy[[i]]=="True",Mxy[[i]]=Dxy[[i,2]]/Dxy[[i,1]];
    If[Abs[Mxy[[i]]]<10^(-10),Mxy[[i]]=0,Mxy[[i]]=Dxy[[i,1]]/Dxy[[i,2]]];
    If[Abs[Mxy[[i]]]<10^(-10),Mxy[[i]]=0,{i,1,npoints}]; (*the slopes of the edges*)

  Do[If[Axy[[i]]=="True",Bxy[[i]]=Mc[[i,2]]-Mxy[[i]]*Mc[[i,1]],Bxy[[i]]=Mc[[i,1]]-
    Mxy[[i]]*Mc[[i,2]],{i,1,npoints}]; (*intercepts*)

  Do[f[i_,x_]:=N[Mxy[[i]],dplaces]*x+N[Bxy[[i]],dplaces],{i,1,npoints}];(*one function for each edge*)

  Fc[p_]:=Module[{d=Range[npoints]},Do[d[[i]]=(p[[1]]-Mc[[i,1]])^2+(p[[2]]-
    Mc[[i,2]])^2,{i,1,npoints}]; Extract[Flatten[Position[d,Min[d]],1]];
  (*Use nearest vertex for first corner*)

  Cros[p_,i_]:=Dxy[[i,2]]*(Mc[[i,1]]-p[[1]])-Dxy[[i,1]]*(Mc[[i,2]]-p[[2]]); (*cross product to test for
    valid tangent. Cross is the Mathematica cross product but we just need a portion of it*)

  Nc[p_]:=Module[{}],While[Cros[p,cc]>0,cc=Mod[cc+1,npoints,1]];
  If[Cros[p,cc]<-10^(-14),Return[cc],Return[0]]; (*Use the 'right-hand' rule to determine if the tangent
    line is valid*)

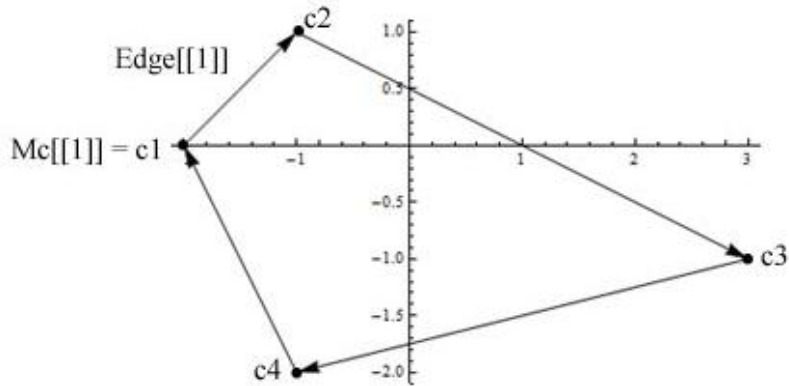
  Np[p_,i_]:=If[i>0,Flatten[{-p+2*Mc[[i]]}],p]; (*next point routine:pnew=-p+2c*)

  NextPoint[p_]:=N[Np[p,Nc[p]],dplaces]; (*apply NP to corner to get next point in orbit*)

  V[p_,k_]:=Module[{}],cc=Fc[p];NestList[NextPoint,p,k]; (*for generating orbits of length k*)

  Tau[p_]:= Last[V[p,1]]; (*End Module*)];
  
```

Example: All of the notebooks use the same TangenMap module. Below we will use the NonRegular.nb with the lattice polygon:  $\mathbf{Mc} = \{-2, 0\}, \{-1, 1\}, \{3, -1\}, \{-1, -2\}$   
**npoints = 4; dplaces = 5;** Using a lattice polygon simplifies the arithmetic without sacrificing generality. We will call these vertices  $\{c1, c2, c3, c4\}$ , so  $\mathbf{Mc}[[1]] = c1 = \{-2, 0\}$

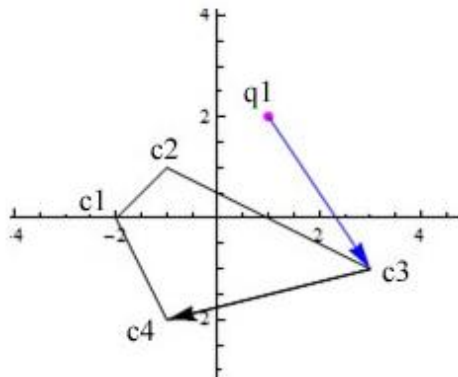


Edge[[1]] is the vector  $\{c1, c2\}$ . Each edge is also assigned a function:  $f[k, z]$  is the function for edge  $k$ : For the matrix above:  $f[1, z] = 2.0000 + 1.0000 z$  and this is the same as  $y = x + 2$ . If the slope exceeds 1, these variables are reversed. This guarantees that no slope exceeds 1. The matrix  $Axy$  monitors the slopes, and here  $Axy = \{\text{True}, \text{True}, \text{True}, \text{False}\}$ , so  $f[4, z] = -2.0000 - 0.50000 z$  has the form  $x = .5y - 2$

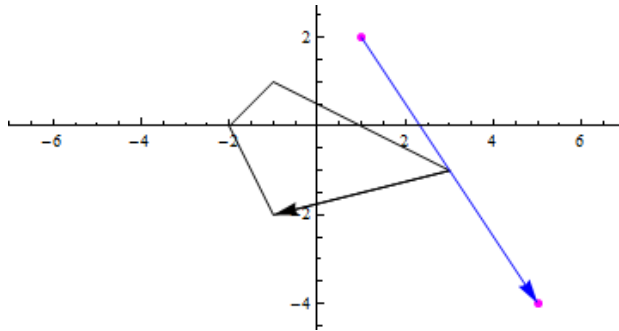
To track the orbit of  $q1 = \{1, 2\}$  shown below, the program first finds the nearest vertex:

FirstCorner =  $\mathbf{Fc}[q1] = 2$  (\*This will set the 'current corner' to 2, so  $cc = 2$ \*)

NextCorner =  $\mathbf{Nc}[q1] = 3$  (\* Starting with  $cc + 1$ , the first vertex to check is  $c3$ . It will be accepted or rejected depending on the sign of the Cross product of the (blue) vector joining  $q1$  to  $c3$  and the vector for Edge[[3]]. The 'right-hand' rule for the sign of the Cross product will yield a 'thumbs down' here, which corresponds to a negative value:  $\mathbf{Cros}[q1, 3] = -14$ . This says that  $c3$  is a valid tangent corner.

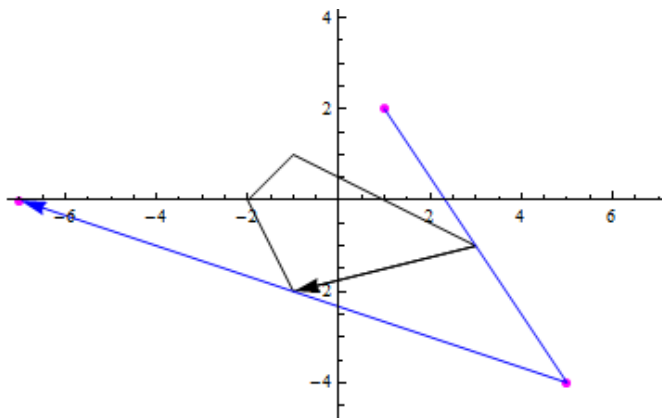


NextPoint =  $\mathbf{Np}[q1,3] = -q1 + 2*c3 = \{5,-4\}$  (\* call this new point q2\*)



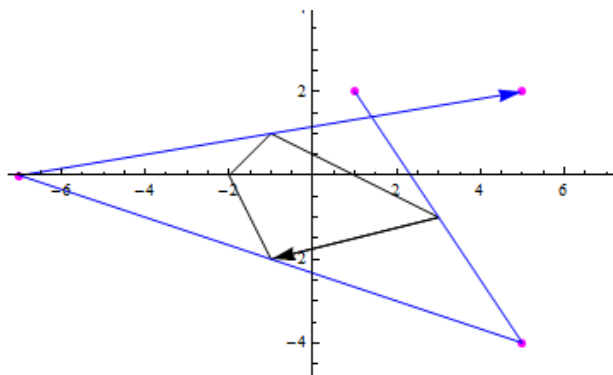
For the next iteration, cc is known so we do not need  $Fc[q2]$ . NextCorner starts searching at  $cc + 1 = 4$  and  $\mathbf{Cros}[q2,4] = -10$  so  $c4$  is accepted as the next valid corner:  $\mathbf{Nc}[q2] = 4$ ;

NextPoint =  $\mathbf{Np}[q2,4] = -q2 + 2*c4 = \{-7,0\}$  (\* this is q3 as shown below\*)

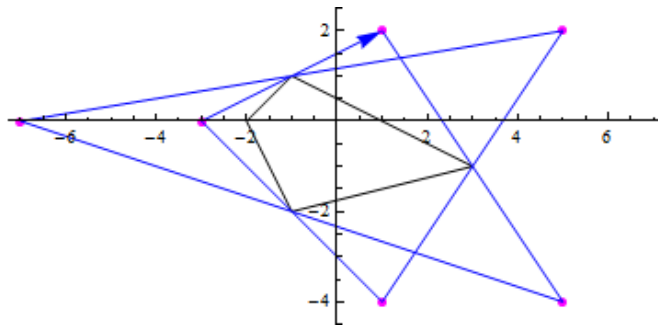


Now  $cc + 1 = 5 \pmod{4} = 1$  but it will be rejected by  $\mathbf{Np}[q3]$  because  $\mathbf{Cros}[q3,1] = 5$ . However vertex  $c2$  is accepted because  $\mathbf{Cros}[q3,2] = -8$ :  $\mathbf{Nc}[q3] = 2$

NextPoint =  $\mathbf{Np}[q3,2] = -q3 + 2*c2 = \{5,2\}$  (\* this is q4 shown below\*)



Two ways to find an orbit:  $\text{Orbit} = \text{NestList}[\text{Tau}, \{1, 2\}, 3] = \{\{1, 2\}, \{5, -4\}, \{-7, 0\}, \{5, 2\}\}$   
or  $\text{Orbit} = \text{V}[\{1, 2\}, 3]$ ; To find the period:  $\text{Period}[\text{V}[\{1, 2\}, 50]] = 6$ . The full orbit is shown below:  
 $\text{Orbit} = \text{Round}[\text{V}[\{1, 2\}, 6]] = \{\{1, 2\}, \{5, -4\}, \{-7, 0\}, \{5, 2\}, \{1, -4\}, \{-3, 0\}, \{1, 2\}\}$



The Tangent Map software provides three pieces of information about an orbit:

(i) The points in the orbit:  $\text{Orbit} = \text{V}[\{1, 2\}, 5] = \{\{1, 2\}, \{5, -4\}, \{-7, 0\}, \{5, 2\}, \{1, -4\}, \{-3, 0\}, \{1, 2\}\}$

(ii) The vertices visited in the orbit:  $\text{Indices} = \text{IND}[\{1, 2\}, 5] = \{3, 4, 2, 3, 4, 2\}$

This list has period 6, but the prime period is 3. This information is used primarily to generate projections, but it can reproduce an orbit if the initial point is known. It uses the step sequences from the  $\text{Vp}$  routine below.

(iii) The 'step sequence' is the number of vertices skipped on each iteration:

$\text{Vp}[\{1, 2\}, 6] = \{1, 2, 1, 1, 2, 1\}$  This is the first difference of the indices from (ii) above. Again the period is 6 but the prime period is 3. It is common for the indices and step sequences to have a shorter prime period than the orbit itself. The step sequences show underlying symmetry in orbits, but their main use is to classify invariant regions and find the winding number of orbits.

