Programming of Maple functions

```
> restart;
> A:=[1,0]; B:=[3,0]; C:=[4,sqrt(5)];
```

$$A := [1, 0]$$

$$B := [3, 0]$$

$$C := [4, \sqrt{5}\,]$$

IntAngle(A,B,C)

Size of the interior angle at the vertex B (middle parameter) of a triangle ABC

```
> IntAngle:=proc(A,B,C)
          local u,v,alpha;
          u:=A-B: v:=C-B:

  alpha:=arccos(abs(linalg[dotprod](u,v))/(linalg[norm](u,2)*lina
  g[norm](v,2))):
          alpha:=evalf(convert(alpha,degrees)):
          end proc:
```

EXAMPLE:

```
> IntAngle(B,A,C);
```

$$36.69922519 \, \mathit{degrees}$$

Volume_Tr(A,B,C)

Volume of a triangle which is given by its vertices A,B,C

```
> Volume_Tr:=proc(A,B,C)
          local M;
          M:=linalg[matrix]([B-A,C-A]):
          1/2*abs(linalg[det](M)):
       end proc:
```

EXAMPLE:

```
> Volume_Tr(A,B,C);
```

$$\sqrt{5}$$

Geq_Pu(P,u)

General equation of a line which is given by its point P and directional vector u

```
> GEq_Pu:=proc(P,u)
          local M;
          M:=linalg[matrix]([[x-P[1],y-P[2]],u]):
          sort(linalg[det](M))=0;
          end proc:
```

EXAMPLE:

```
> GEq_Pu(A,B-A);
```

$$-2\,y = 0$$

```
> GEq_Pu(A,C-A);
```

$$\sqrt{5}\,x - 3\,y - \sqrt{5} = 0$$

Dist_Pl(P,l)

Distance of a point P to a line l which given by general equation

```
> Dist_Pl:=proc(P,l)
                 local a,b,c,k;
                 k:=lhs(l):
                 a:=coeff(k,x): b:=coeff(k,y):
  c:=coeff(coeff(k,x,0),y,0):
                 abs(a*P[1]+b*P[2]+c)/sqrt(a^2+b^2);
                 end proc:
```

EXAMPLE:

```
> Dist_Pl([2,3],3*x-5*y+11=0);
```

$$\frac{\sqrt{34}}{17}$$

```
> Dist_Pl(A,GEq_Pu(B,C-B));
```

$$\frac{\sqrt{5}\,\sqrt{6}}{3}$$

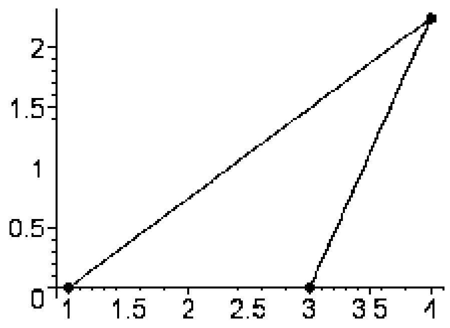
Plot_Tr(A,B,C)

Draw a triangle ABC

```
> Plot_Tr:=proc(A,B,C)
             local Ad,Bd,Cd,T,rd;
             T:=plot([A,B,C,A],thickness=3,color=black):
             rd:=linalg[norm](B-A,2)/50:
             Ad:=plottools[disk](A,rd,color=black):
             Bd:=plottools[disk](B,rd,color=black):
             Cd:=plottools[disk](C,rd,color=black):
             plots[display](T,Ad,Bd,Cd,scaling=constrained):
             end proc:
```

EXAMPLE:

```
> Plot_Tr(A,B,C);
```



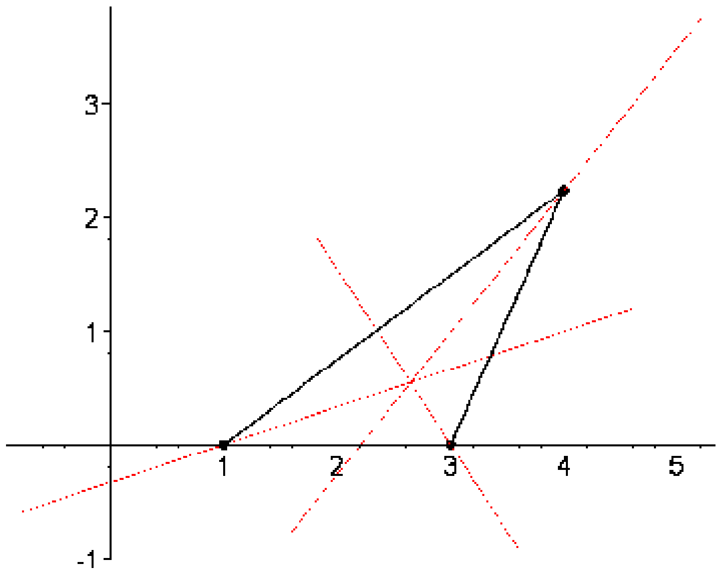
PEq_AngBis(A,B,C,t)

Parametric equations (vector-valued function) with parameter t of an angular bisector of the interior angle at the vertex B of a triangle ABC.

```
> PEq_AngBis:=proc(A,B,C,t)

  expand(B+t*((A-B)/linalg[norm](A-B,2)+(C-B)/linalg[norm](C-B,2))
  ):
```

```
            end proc:
```
EXAMPLE:
```
> PEq_AngBis(A,B,C,r);
```
$$\left[\frac{1}{6}\sqrt{6}\,r - r + 3, \frac{\sqrt{5}\,\sqrt{6}\,r}{6}\right]$$

EXAMPLE: Plotting of angular bisectors of the triangle ABC
```
> BisA:=plot([op(PEq_AngBis(C,A,B,t)),t=-1..2]):
> BisB:=plot([op(PEq_AngBis(A,B,C,t)),t=-1..2]):
> BisC:=plot([op(PEq_AngBis(B,C,A,t)),t=-1..2]):
> plots[display](Plot_Tr(A,B,C),BisA,BisB,BisC,scaling=constrained
  );
```


CentreInC(A,B,C)

Derive the centre of the circle which is inscribed to a triangle ABC
```
> CentreInC:=proc(A,B,C)
            local
  BisA, BisB, Sol;
  BisA:=expand(A+r*((B-A)/linalg[norm](B-A,2)+(C-A)/linalg[norm](C
  -A,2))):
  BisB:=expand(B+s*((A-B)/linalg[norm](A-B,2)+(C-B)/linalg[norm](C
  -B,2))):
  Sol:=solve({op(expand(BisA-BisB))},{r,s}):
  simplify(eval(BisA,Sol)):
           end proc:
```
EXAMPLE:
```
> CentreInC(A,B,C);
```
$$\left[-\frac{\sqrt{2}\,\sqrt{3}}{2} + \frac{\sqrt{2}\,\sqrt{7}}{2} + 2, \frac{\sqrt{5}\,\sqrt{2}\,\sqrt{7}\,(3\sqrt{3}\,\sqrt{7} - 7\sqrt{2}\,\sqrt{3} + 4\sqrt{2}\,\sqrt{7} - 7)}{70}\right]$$

Eq_InC(A,B,C)
Equation of the inscribed circel to a triangle ABC

```
> Eq_InC:=proc(A,B,C)
          local R,S,Sol;
          S:=CentreInC(A,B,C):
          R:=Dist_Pl(S,GEq_Pu(A,B-A)):
          (x-S[1])^2+(y-S[2])^2=R^2;
          end proc:
```

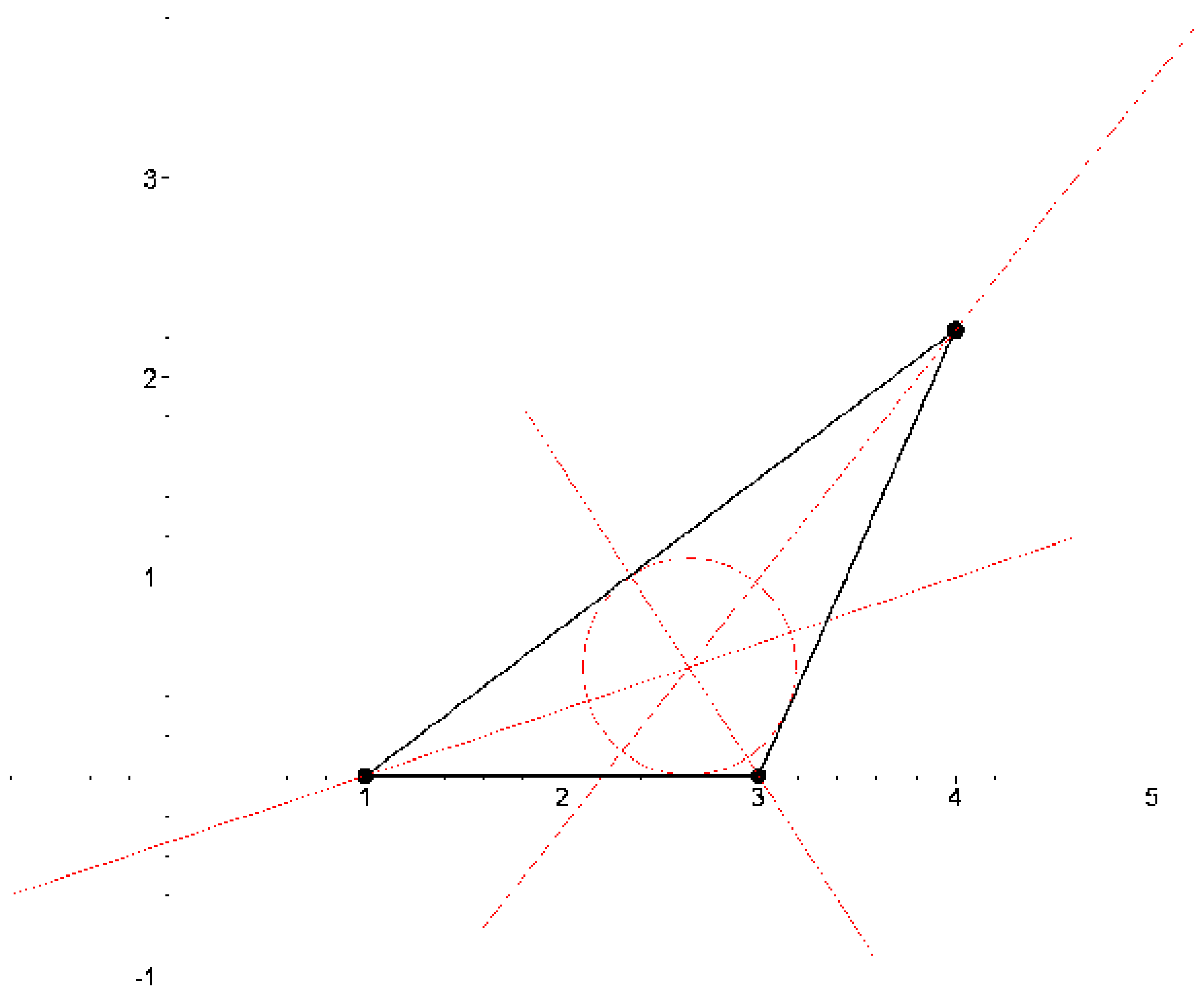
EXAMPLE:

```
> Eq_InC(A,B,C);
```

$$\left(x+\frac{\sqrt{2}\,\sqrt{3}}{2}-\frac{\sqrt{2}\,\sqrt{7}}{2}-2\right)^2+\left(y-\frac{\sqrt{5}\,\sqrt{2}\,\sqrt{7}\,(3\sqrt{3}\,\sqrt{7}-7\sqrt{2}\,\sqrt{3}+4\sqrt{2}\,\sqrt{7}-7)}{70}\right)^2$$
$$=\frac{(3\sqrt{3}\,\sqrt{7}-7\sqrt{2}\,\sqrt{3}+4\sqrt{2}\,\sqrt{7}-7)^2}{70}$$

EXAMPLE:

```
> ICg:=plots[implicitplot](Eq_InC(A,B,C),x=-10..10,y=-10..10,grid=
  [200,200]):
> plots[display](Plot_Tr(A,B,C),BisA,BisB,BisC,ICg,scaling=constra
  ined);
```



How to make an own package

Besides the using of the build-in packages we can easily make packages of our own procedures. Maple views a package as a type of table. Each key in the table is the name of a procedure, and the value corresponding to the key is the procedure. All we have to do is define the table, just as we would define any Maple table, then save it in a ***.m** file. Maple can easily recognize our package and treat it like any of the ones with which it is already familiar. One advantage of saving a group of related procedures as a table is that the user can load the package using the **with** command.

Let us illustrate this process by the building of the **GeometryPack** package of our foregoing procedures. For brief illustration we will use only one command - IntAngle.

```
> GeometryPack:=table();
```

GeometryPack := table([])

```
> GeometryPack[IntAngle]:=proc(A,B,C)
```

```
            local u,v,alpha;
            u:=A-B: v:=C-B:

  alpha:=arccos(abs(linalg[dotprod](u,v))/(linalg[norm](u,2)*linal
  g[norm](v,2))):
            alpha:=evalf(convert(alpha,degrees)):
            end proc:
```

Since your package is a table, you can save it as you would save any other Maple object. The command below saves the
table **GeometryPack** in Maple's binary format to the file **GeometryPack.m** in the directory **C:\Program Files\Maple 9.5\Packs\**

```
> save(GeometryPack,"C:\\Program Files\\Maple
  9.5\\Packs\\GeometryPack.m");
```

The **with** command can load our new package, provided we save it in a file with a name of the form **packagename.m** (we did), and provided we tell Maple in
which directories it should search for the new package file. The variable **libname** is a sequence of directory names which Maple searches in order from left to right. Thus we must append the name of the directory containing our package to the **libname**.

```
> libname:="C:\\Program Files\\Maple 9.5\\Packs",libname;
          libname := "C:\Program Files\Maple 9.5\Packs", "C:\Program Files\Maple 9.5/lib"
```

Now we can load our package using the **with** command and commence using it.

```
> with(GeometryPack);
                                  [IntAngle]
> IntAngle([1,1],[2,3],[-5,10]);
                              71.56505120 degrees
>
```