

pst-eucl

## Geometria euclidea con **PSTricks**

Luciano Battaia

Guida alle funzionalità di base, basata sul manuale originale di Dominique Rodriguez.  
Applicazioni didattiche per la Scuola Media Superiore.

Versione del 24 febbraio 2007

## Sommario

In questa miniguia sono raccolti i materiali e le indicazioni forniti durante i corsi di introduzione a `pst-eucl`, tenuti, come attività curricolare, agli studenti del Liceo Scientifico Grigoletti di Pordenone. L'impostazione è dunque quella di un manuale scolastico senza, in particolare, la pretesa di costruire una guida sistematica, per la quale si rimanda al manuale fornito con il pacchetto.

Le motivazioni didattiche che mi hanno condotto ad inserire questo software nei curricula scolastici, in particolar modo nei corsi ad indirizzo scientifico, si possono reperire all'interno della guida. Motivazioni di carattere più generale, e riferite all'intero sistema `LATEX`, si possono trovare nella mia miniguia di introduzione al più diffuso sistema di videoscrittura a livello scientifico.

La guida è pensata per gli utenti `MikTeX`, ma, con pochissime varianti, è adatta a tutte le installazioni `LATEX`. Nella parte tecnica è sostanzialmente un riassunto del Manuale distribuito con il pacchetto; per tutte le informazioni qui non riportate fate riferimento a quel manuale oltre che ai manuali di `PSTricks`. La versione utilizzata del pacchetto è la 1.3.2 del 28 marzo 2005.

# Indice

Premessa . . . . .	2
<b>1 Guida alle funzionalità di base</b>	<b>3</b>
1.1 Installazione . . . . .	3
1.2 Primi passi . . . . .	3
1.3 Punti . . . . .	4
1.4 Segmenti e linee . . . . .	5
1.5 Angoli . . . . .	6
1.6 Triangoli e poligoni . . . . .	8
1.7 Cerchi e archi di cerchio . . . . .	9
1.8 Punto medio di un segmento, baricentro di un triangolo . . . . .	11
1.9 Assi, circonferenza per tre punti . . . . .	11
1.10 Bisettrici . . . . .	13
1.11 Premessa sulle trasformazioni geometriche . . . . .	13
1.12 Simmetrie centrale e assiale . . . . .	14
1.13 Rotazioni . . . . .	14
1.14 Traslazioni . . . . .	15
1.15 Omotetie . . . . .	15
1.16 Proiezioni ortogonali . . . . .	16
1.17 Intersezioni . . . . .	16
<b>2 Applicazioni didattiche</b>	<b>19</b>
2.1 Costruzioni geometriche . . . . .	19
Inscrivere un quadrato in un triangolo . . . . .	19
Divisione di un segmento . . . . .	20
Circonferenze per due punti e tangenti a una retta . . . . .	21
Costruzione di un triangolo date le mediane . . . . .	22
Tangenti comuni a due circonferenze . . . . .	23
Sezione aurea di un segmento . . . . .	24
2.2 Esercizi proposti . . . . .	25

## Premessa

Il pacchetto `pst-eucl` consente di tracciare figure geometriche sostanzialmente con il metodo “riga e compasso”, in maniera non molto dissimile da quanto si fa con Cabri o altri software di geometria dinamica. Naturalmente non si hanno a disposizione “pulsanti” o altre scorciatoie che prevedono l’uso del mouse: tutto deve essere “programmato”, esattamente come si fa con un normale linguaggio in informatica, e questo rende, a mio avviso, il pacchetto estremamente interessante dal punto di vista delle sue ricadute didattiche, in particolare nella scuola media superiore dove l’abitudine allo studio sistematico della geometria euclidea va sempre più scemando.

Dal punto di vista strettamente “geometrico” la caratteristica molto interessante del pacchetto è che, una volta definiti i punti di base di una costruzione geometrica, tutto il resto si ottiene mediante i tradizionali sistemi della geometria: trasformazioni ed intersezioni. La ricerca del metodo adatto alle varie costruzioni è un ottimo esercizio, di gran lunga più efficace dei normali “problemi” che si trovano sui manuali scolastici. Se si aggiunge il fatto che la qualità tipografica dei risultati ottenuti è di livello assolutamente professionale (come per tutto quello che riguarda il “sistema  $\text{\LaTeX}$ !”) si capisce facilmente l’importanza che può assumere l’introduzione di un software come questo nei curricula degli studi di Scuola Media Superiore.

Lo studio di pacchetti come questo è inoltre, a mio avviso, di grande utilità ai fini dell’alfabetizzazione informatica in particolare in quei corsi dove non è previsto lo studio dell’informatica come materia autonoma, ma come appendice allo studio della matematica.

Il pacchetto è frutto del lavoro di Dominique Rodriguez e rimando alla documentazione originale in particolare per tutte le informazioni riguardanti la licenza d’uso, che comunque rientra nei termini del « $\text{\LaTeX}$  Project Public License».

# Capitolo 1

## Guida alle funzionalità di base

### 1.1 Installazione

Se avete MiKTeX 2.5 il pacchetto dovrebbe essere già installato. Potete comunque installarlo come tutti gli altri pacchetti L<sup>A</sup>T<sub>E</sub>X in MiKTeX, oppure seguire la procedura che segue.

Reperite l'indirizzo del pacchetto sul sito ufficiale di PStricks, <http://www.pstricks.de>, nella pagina "Packages". I file indispensabili sono `pst-eucl.sty`, `pst-eucl.pro`, `pst-eucl.tex`. Conviene mettere questi file nelle cartelle seguenti, già presenti sul vostro hard disk se avete installato Miktex versione "Total":

- `pst-eucl.sty` in `... \tex\latex\pstricks`;
- `pst-eucl.pro` in `... \dvips\pstricks`;
- `pst-eucl.tex` in `... \tex\generic\pstricks`.

In queste cartelle troverete altri file con le stesse estensioni. Fate, come sempre, un refresh del database di Miktex, tramite Miktex Options.

Potete poi scaricare il file `euclide.pdf` e stamparlo: si tratta del manuale completo (una quarantina di pagine) pronto per la stampa. Gli altri file sono di supporto al manuale, ma non servono per far funzionare il pacchetto, né per stampare il manuale.

Ricordatevi di leggere la licenza d'uso e di soddisfare le limitatissime richieste in essa contenute.

### 1.2 Primi passi

Per usare il pacchetto basterà caricarlo, dopo PStricks, con il solito `\usepackage{pst-eucl}` nel preambolo del documento. Dopodichè tutto funziona come in PStricks, con l'aggiunta dei comandi che vedremo. In sostanza le varie figure vengono costruite in un sistema di coordinate cartesiane, che rimarrà comunque, di default, invisibile. In alcuni degli esempi che proporremo mostreremo anche una griglia coordinata di riferimento, che ha solo scopo dimostrativo.

Tenete conto che il pacchetto carica di default una speciale opzione che consente di usare sia coordinate cartesiane che coordinate polari. La scrittura  $(x, y)$  indica le coordinate cartesiane di un punto, la scrittura  $(\rho; \theta)$  (con il punto e virgola al posto della virgola) indica le coordinate polari. Le coordinate cartesiane di default sono ortonormali (ma potrebbero essere cambiate).

Occorre anche ricordare che, come in PStricks, gli angoli sono misurati in gradi e in senso antiorario, con la solita regola valida nei sistemi cartesiani ortogonali.

Per partire occorre fissare alcuni punti base, dai quali dipenderà tutta la figura da costruire. Questi punti sono dei "nodi", nel senso speciale usato in PStricks per questa parola. Per gli scopi di questa guida non ci preoccuperemo comunque di questo fatto. La sola cosa che ci interesserà è che tutti i punti che verranno via via costruiti avranno un nome col quale potremo farvi riferimento, senza dover specificare più le loro coordinate cartesiane che eventualmente ci sono servite per crearli. È da segnalare che la scelta dei punti base della costruzione è una delle operazioni preliminari che richiedono un attento studio "a tavolino" della figura che si vuole costruire: anche questo è un esercizio di grande utilità per affinare le proprie conoscenze della geometria piana.

Negli esempi che saranno proposti utilizzeremo anche alcuni comandi propri di PStricks dei quali di solito daremo anche una breve spiegazione.

## 1.3 Punti

Occorre decidere le coordinate cartesiane (o polari) dei punti e i nomi dei punti stessi. Attenzione che, come vedremo, i nomi dei punti possono anche essere diversi dalle etichette che compariranno nel disegno a fianco dei punti stessi (anzi i punti potrebbero anche non avere etichette, come spesso succede nelle figure geometriche, ma devono comunque avere un nome). In ogni caso le etichette dei punti sono trattate di default in modo matematico (cioè vengono scritte con lo speciale stile *slanted* previsto per il modo matematico, anche se non devono essere racchiusi tra una coppia di segni di dollaro).

Supponendo di voler definire una serie di punti  $A_1, A_2, \dots$ , con coordinate  $(x_1, y_1), (x_2, y_2), \dots$ , il comando di base è:

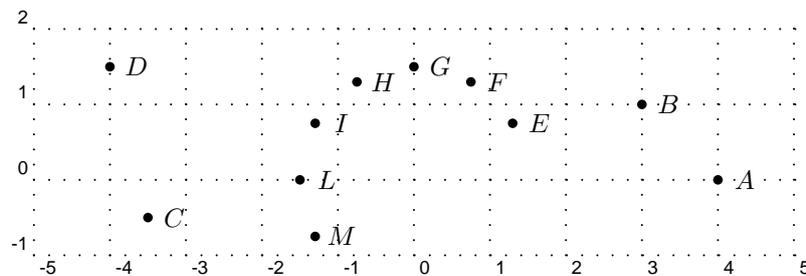
```
\pstGeonode[opzioni]( $x_1, y_1$ ){ $A_1$ }( $x_2, y_2$ ){ $A_2$ }...
```

Senza l'indicazione di alcuna opzione i punti vengono piazzati nelle posizioni indicate, con il simbolo di default (un pallino nero) e con l'etichetta piazzata sulla destra del punto, centrata e ad 1em di distanza.

**Esempio 1.1.** Tracciamento di punti.

```
\begin{pspicture}(-5.5,-1.5)(5.5,2.5)
\pstGeonode(4,0){A}(3,1){B}(-3.5,-0.5){C}(-4,1.5){D}(1.5;30){E}(1.5;60){F}(1.5;90)%
  {G}(1.5;120){H}(1.5;150){I}(1.5;180){L}(1.5;210){M}
\end{pspicture}
```

Si noti che la riga che contiene il comando `\pstGeonode` è troppo lunga ed è stata interrotta con un `%`: è una regola generale per le righe di `LATEX`, che non devono contenere spazi.



Veniamo ora a parlare delle opzioni. Se esse sono le stesse per tutti i punti della lista sono semplicemente del tipo `opzione=valore`. Se invece sono diverse per i vari punti, sono del tipo `opzione={valore1,valore2,...}`. Se i valori tra parentesi graffe sono meno dei punti della lista l'ultimo valore è usato per i punti successivi. Se le opzioni sono più di una vanno separate da una virgola.

Le opzioni comuni sono le seguenti:

- `PointNameSep`: indica quanto è distante l'etichetta dal punto (1em per default);
- `PosAngle`: indica l'angolo a cui viene posizionata l'etichetta del punto (0 per default);
- `PointName`: indica l'etichetta del punto, se si vuole che sia diversa dal nome del punto stesso; se il punto non deve avere etichetta, basta porre questo parametro uguale a `none`.
- `PtNameMath`: se posta uguale a `false`, fa scrivere l'etichetta in testo normale (cioè diritto) e non *slanted*.
- `PointSymbol`: indica il tipo di simbolo usato per il punto. Di default si tratta di un cerchietto pieno, ma sono previste altre possibilità (le stesse contemplate da `PSTricks`) e cioè le seguenti:

- |                |                 |                 |
|----------------|-----------------|-----------------|
| - * : ●        | - otimes : ⊗    | - diamond* : ◆  |
| - o : ○        | - triangle : △  | - pentagon : ⬠  |
| - + : +        | - triangle* : ▲ | - pentagon* : ⬡ |
| - x : ×        | - square : □    | -   :           |
| - asterisk : * | - square* : ■   |                 |
| - oplus : ⊕    | - diamond : ◇   |                 |

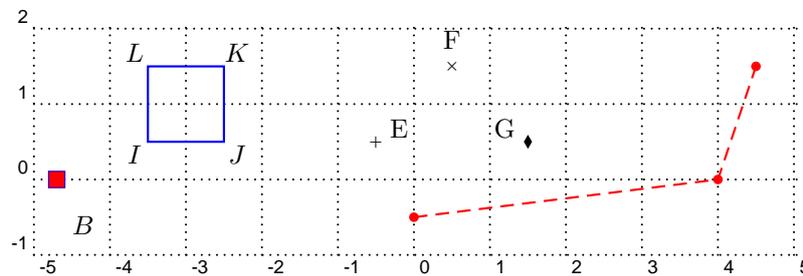
Tutti questi simboli accettano naturalmente le opzioni previste in `PSTricks`, di cui citiamo:

- `dotscale` : un numero che rappresenta la scalatura rispetto a quanto previsto di default, oppure due numeri tra parentesi graffe e separati da un virgola che rappresentano la scalatura orizzontale e verticale; per esempio `dotscale=2` per raddoppiare le dimensioni, `dotscale={1,2}` per mantenere la dimensione orizzontale di default e raddoppiare quella verticale;
- `dotangle` : un numero per l'angolo di rotazione del simbolo;

- `fillstyle` : un valore, per esempio `solid`, per indicare il tipo di riempimento;
  - `fillcolor` : un valore per indicare il colore del riempimento;
  - `linecolor` : un valore per indicare il colore del bordo del simbolo che rappresenta il punto.
- `CurveType` : nel caso di una lista di punti, i punti possono venire collegati da una spezzata (`polyline`), una poligonale chiusa (`polygon`), una curva aperta (`curve`).

**Esempio 1.2.** Tracciamento di punti e linee.

```
\begin{pspicture}(-5.5,-0.5)(5.5,2.5)
\pstGeonode[PointSymbol=square,dotscale=2,fillstyle=solid,fillcolor=red,%
linecolor=blue,PointName=B,PosAngle=-60,PointNameSep=2em](-4.7,0){A_2}
\pstGeonode[PtNameMath=false,PointSymbol={+,x,diamond*},PosAngle={30,90,150}]%
(-0.5,0.5){E}(0.5,1.5){F}(1.5,0.5){G}
\pstGeonode[PointName=none,CurveType=polyline,linestyle=dashed,linecolor=red]%
(0,-0.5){M}(4,0){N}(4.5,1.5){P}
\pstGeonode[PointSymbol=none,CurveType=polygon,linecolor=blue,PosAngle=%
{-135,-45,45,135},PointNameSep=0.7em](-3.5,0.5){I}(-2.5,0.5){J}(-2.5,1.5){K}(-3.5,1.5){L}
\end{pspicture}
```



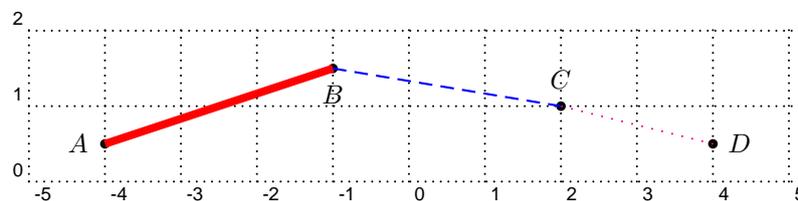
Si noti come i parametri introdotti consentano un controllo molto fine dell'aspetto della figura.

## 1.4 Segmenti e linee

Per tracciare segmenti si può usare il comando standard di PSTricks, `\ncline` (`nodeconnectionline`)

**Esempio 1.3.** Tracciamento di segmenti.

```
\begin{pspicture}(-5.5,-0.5)(5.5,2.5)
\pstGeonode[PosAngle={180,-90,90,0}](-4,0.5){A}(-1,1.5){B}(2,1){C}(4,0.5){D}
\ncline[linecolor=red,linewidth=3pt]{A}{B}
\ncline[linecolor=blue,linestyle=dashed]{B}{C}
\ncline[linecolor=magenta,linestyle=dotted]{C}{D}
\end{pspicture}
```



In geometria è spesso utile segnare i segmenti. Il comando

```
\pstSegmentMark[{opzioni}]{A}{B}
```

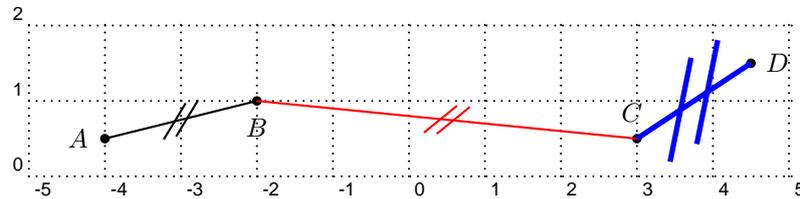
traccia il segmento che congiunge i nodi *A* e *B* precedentemente definiti e lo segna con un segno di default (che naturalmente può essere cambiato).

**Esempio 1.4.** Marcatura di segmenti.

```

\begin{pspicture}(-5.5,-0.5)(5.5,2.5)
\pstGeonode[PosAngle={180,-90,100,0}](-4,0.5){A}(-2,1){B}(3,0.5){C}(4.5,1.5){D}
\pstSegmentMark{A}{B}
\pstSegmentMark[linecolor=red]{B}{C}
\pstSegmentMark[linecolor=blue,linewidth=2pt]{C}{D}
\end{pspicture}

```



I segni predefiniti nel pacchetto sono i seguenti:

- pstslash :		;	- MarkHash :		;	- MarkCros :		;
- pstslashh :		;	- MarkHashh :		;	- MarkCross :		;
- pstslashhh :		;	- MarkHashhh :		;			

Nel caso dei simboli del tipo `MarkHash` si può addirittura regolare l'inclinazione con il parametro `MarkAngle`.

Per tracciare linee rette passanti per due punti è previsto il comando

```
\pstLineAB[opzioni]{A}{B}
```

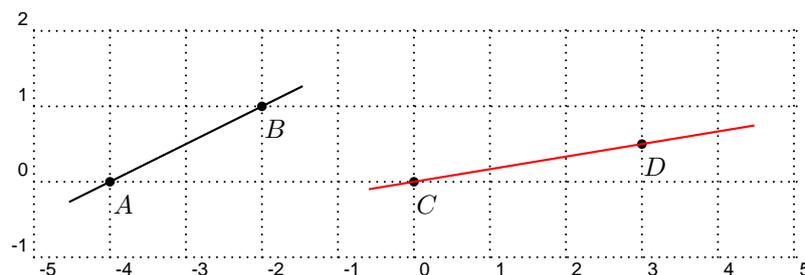
che, di default, traccia esattamente il segmento che congiunge  $A$  e  $B$ , ma che possiede il parametro `nodesep` (oppure i due parametri `nodesepA` e `nodesepB`) che consentono di “prolungare” questo segmento oltre gli estremi. Si può vedere l'esempio che segue.

**Esempio 1.5.** Linee.

```

\begin{pspicture}(-5.5,-1.5)(5.5,2.5)
\pstGeonode[PosAngle=-60](-4,0){A}(-2,1){B}(0,0){C}(3,0.5){D}
\pstLineAB[nodesep=-.6]{A}{B}
\pstLineAB[nodesepA=-.6,nodesepB=-1.5,linecolor=red]{C}{D}
\end{pspicture}

```



Per tracciare poligonal si può usare il comando standard di PSTricks `\psline`, seguito dai punti che fanno da vertici della poligonale. Se i punti sono nodi già definiti, basta scriverne il nome tra parentesi tonde: `\psline(A)(B)(C)...`

## 1.5 Angoli

I comandi previsti permettono di marcare gli angoli con simboli e di piazzare etichette. Ogni angolo deve essere identificato da tre punti, di cui quello centrale è il vertice. L'ordine dei tre punti è importante, perchè gli angoli sono orientati.

Per gli angoli retti il comando è:

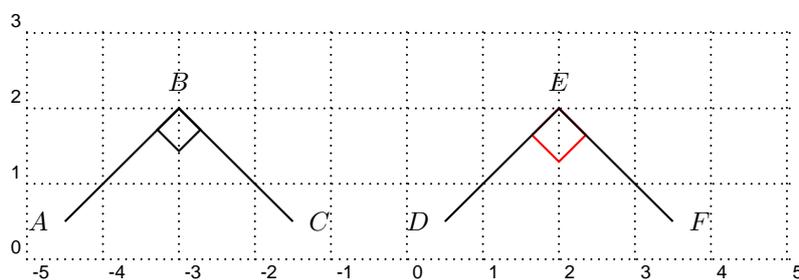
```
\pstRightAngle[⟨opzioni⟩]{⟨A⟩}{⟨B⟩}{⟨C⟩}
```

Il simbolo utilizzato è quello tradizionale per gli angoli retti (ci sono altre possibilità che non esaminiamo qui). Il parametro `RightAngleSize` determina le dimensioni del quadratino usato come simbolo.

**Esempio 1.6.** Angoli retti.

```
\begin{pspicture}(-5.5,-0.5)(5.5,3.5)
\pstGeonode[PosAngle={180,90,0}](-4.5,0.5){A}(-3,2){B}(-1.5,0.5){C}
\psline(A)(B)(C)
\pstRightAngle{A}{B}{C}
\pstGeonode[PosAngle={180,90,0}](0.5,0.5){D}(2,2){E}(3.5,0.5){F}
\pstRightAngle[RightAngleSize=.5,linecolor=red]{D}{E}{F}
\psline(D)(E)(F)
\end{pspicture}
```

Si noti che per il secondo angolo la marcatura è fatta prima di tracciare i lati: il simbolo utilizzato è infatti un quadratino che, essendo rosso, si sarebbe parzialmente sovrapposto al colore dei lati dell'angolo.



Per gli altri angoli il comando è:

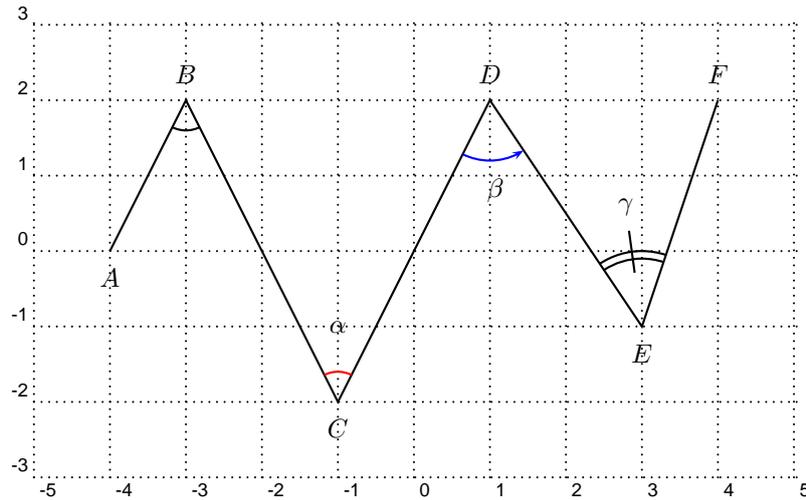
```
\pstMarkAngle[⟨opzioni⟩]{⟨A⟩}{⟨B⟩}{⟨C⟩}{⟨etichetta⟩}
```

L'etichetta può essere un qualunque testo accettabile da  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ; è piazzata ad una distanza di 1 unità dal vertice dell'angolo, sulla bisettrice dello stesso; la distanza può essere variata con il parametro `LabelSep`, mentre la sua posizione rispetto alla bisettrice può essere variata mediante `LabelAngleOffset`. Se non si vogliono etichette bisogna comunque mettere la coppia vuota di parentesi graffe. Per marcare l'angolo viene usato un arco di cerchio, il cui raggio di 0.4 unità può essere variato mediante il parametro `MarkAngleRadius`; a questo arco si possono aggiungere frecce mediante il parametro `arrows` (che assume i valori `->`, oppure `<-` oppure `<->`); infine si può aggiungere un segno dello stesso tipo di quello usato per i segmenti, mediante il parametro `Mark`. Si noti ancora una volta la flessibilità dei comandi, che rende possibile un controllo estremamente preciso della figura.

**Esempio 1.7.** Angoli qualunque.

```
\begin{pspicture}(-5.5,-3.5)(5.5,3.5)
\psset{PointSymbol=none}
\pstGeonode[CurveType=polyline,PosAngle={-90,90,-90,90,-90,90}]%
(-4,0){A}(-3,2){B}(-1,-2){C}(1,2){D}(3,-1){E}(4,2){F}
\pstMarkAngle{A}{B}{C}{ }
\pstMarkAngle[linecolor=red]{D}{C}{B}{\alpha}
\pstMarkAngle[linecolor=blue,MarkAngleRadius=0.8,LabelSep=1.2,arrows=->]{C}{D}{E}{\beta}
\pstMarkAngle[MarkAngleRadius=1,LabelSep=1.6,Mark=MarkHash]{F}{E}{D}{\gamma}
\pstMarkAngle[MarkAngleRadius=0.9]{F}{E}{D}{ }
\end{pspicture}
```

Da notare che abbiamo considerato gli angoli  $\widehat{DCB}$  e  $\widehat{FED}$  per avere quelli acuti. Se avessimo considerato  $\widehat{BCD}$  o  $\widehat{DEF}$  avremmo marcato gli angoli esplementari. L'istruzione `\psset`, propria di `PSTricks`, serve a stabilire un valore per un parametro che rimane valido fino a nuovo ordine: in questo caso non volevamo avere alcun simbolo per i punti. L'angolo  $\widehat{FED}$  è stato marcato due volte con raggi leggermente diversi per ottenere il doppio arco.



## 1.6 Triangoli e poligoni

Per i poligoni si può usare il comando standard di PSTricks, `\pspolygon`, seguito dalle coppie di coordinate dei vertici. Se i vertici sono nodi già definiti basta mettere il loro nome tra parentesi tonde. Se si vuole una poligonale aperta basta usare `\psline`, al posto di `\pspolygon`.

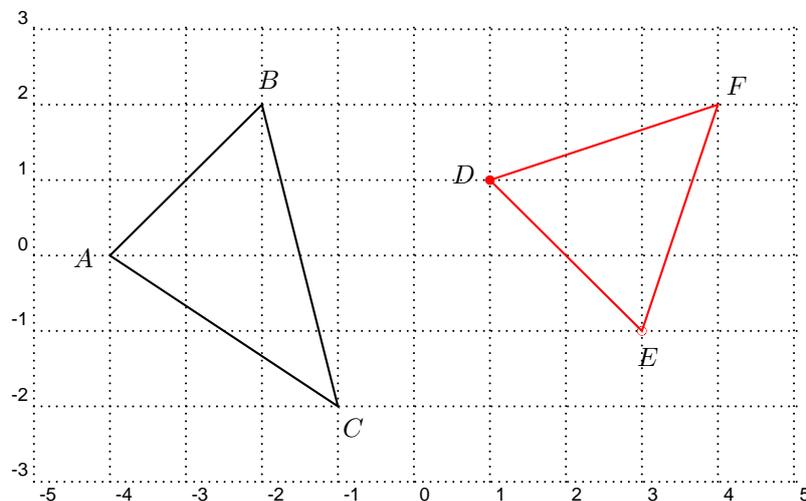
I triangoli sono figure così importanti e frequenti nella geometria piana che l'autore di `pst-eucl` ha implementato un comando speciale per costruirli:

```
\pstTriangle[⟨opzioni⟩](xA,yA⟩){⟨A⟩}(xB,yB⟩){⟨B⟩}(xC,yC⟩){⟨C⟩}
```

La particolarità di questo comando non sta nel tracciamento della figura, quanto nel posizionamento delle etichette dei vertici che è fatto, di default, sulla bisettrice dei tre angoli. La posizione può comunque essere modificata con il parametro `PosAngle`, o con i parametri `PosAngleA`, `PosAngleB`, `PosAngleC` se si vuole modificare la posizione di una sola delle etichette. Per i simboli e i nomi dei punti, oltre ai già noti `PointSymbol` e `PointName`, si possono usare anche `PointSymbolA`, `PointSymbolB`, `PointSymbolC` e `PointNameA`, `PointNameB`, `PointNameC`, con ovvio significato.

**Esempio 1.8.** Triangoli.

```
\begin{pspicture}(-5.5,-3.5)(5.5,3.5)
\psset{PointSymbol=none}
\pstTriangle(-4,0){A}(-2,2){B}(-1,-2){C}
\pstTriangle[linecolor=red,PointSymbolA=*,PointSymbolB=o](1,1){D}(3,-1){E}(4,2){F}
\end{pspicture}
```



## 1.7 Cerchi e archi di cerchio

Considerata l'importanza del cerchio nella geometria, ci sono, in questo pacchetto, diversi modi per tracciarli. I comandi fondamentali prevedono la specificazione del centro e di un punto, oppure di due punti diametralmente opposti:

```
\pstCircleOA[⟨opzioni⟩]{⟨O⟩}{⟨A⟩}
```

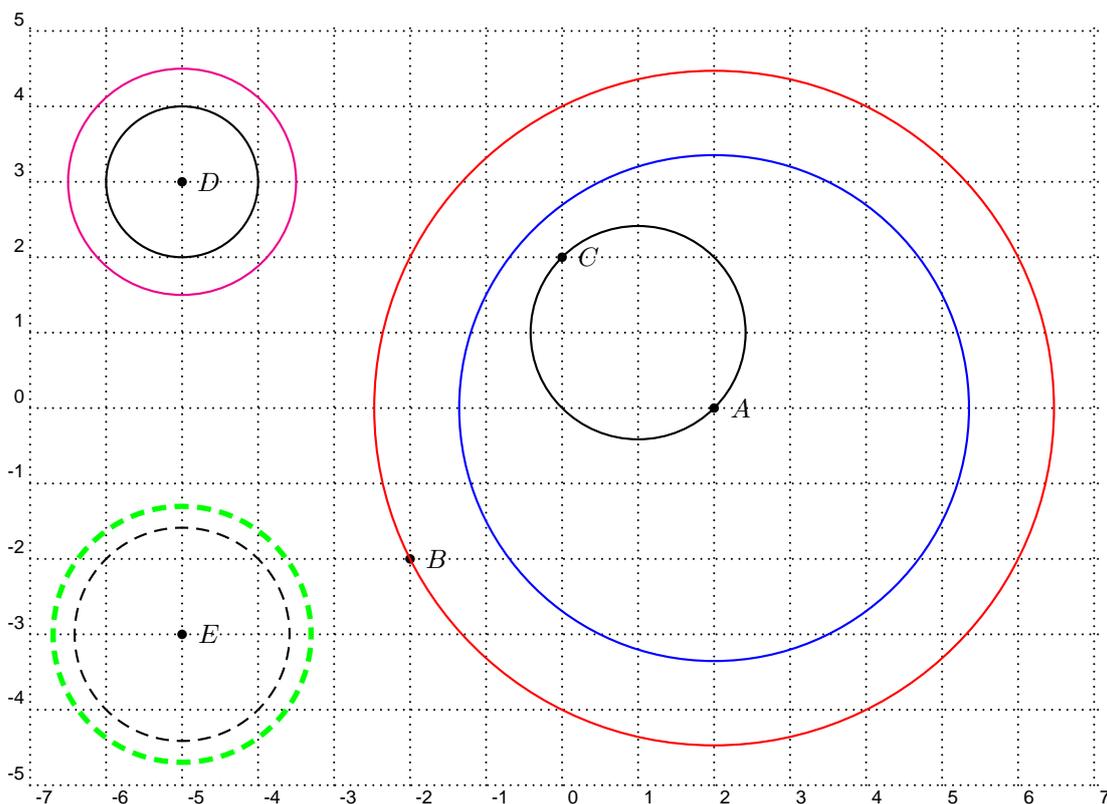
```
\pstCircleAB[⟨opzioni⟩]{⟨A⟩}{⟨B⟩}
```

Nel primo caso si può anche omettere il secondo punto (lasciando però la coppia vuota di parentesi graffe), e specificare invece il raggio (con il parametro `Radius`) o il diametro (con il parametro `Diameter`), per mezzo di una delle due funzioni:

- `\pstDistAB{⟨A⟩}{⟨B⟩}`; in questo modo si può tracciare un cerchio di dato centro e avente raggio uguale alla distanza tra due punti qualunque della figura  $A$  e  $B$ . È anche possibile introdurre un coefficiente moltiplicativo per questa distanza, mediante il parametro `DistCoef`, che può anche essere un codice di calcolo in PostScript e che deve precedere il parametro `\pstDistAB`.
- `\pstDistVal{x}`. In questo caso si può introdurre direttamente un valore numerico per  $x$ , che può anche essere un calcolo Postscript.

*Esempio 1.9.* Cerchi.

```
\begin{pspicture}(-7.5,-5.5)(7.5,5.5)
\pstGeonode(2,0){A}(-2,-2){B}(0,2){C}(-5,3){D}(-5,-3){E}
\pstCircleOA[linecolor=red]{A}{B}
\pstCircleOA[linecolor=blue,DistCoef=3 4 div,Radius=\pstDistAB{A}{B}]{A}{}
\pstCircleAB{A}{C}
\pstCircleOA[Diameter=\pstDistVal{2}]{D}{}
\pstCircleOA[Radius=\pstDistVal{1.5},linecolor=magenta]{D}{}
\pstCircleOA[DistCoef=.5,Radius=\pstDistAB{A}{C},linestyle=dashed]{E}{}
\pstCircleOA[DistCoef=.6,Radius=\pstDistAB{A}{C},%
linestyle=dashed,linewidth=2pt,linecolor=green]{E}{}
\end{pspicture}
```



Per tracciare archi di cerchio si deve conoscere il centro del cerchio e le due semirette di origine e fine dell'arco, individuate dal centro e da due punti  $A$  e  $B$ . Il raggio del cerchio è dato dalla distanza tra il centro e il primo dei due punti  $A$  e  $B$  (che non occorre siano alla stessa distanza da  $O$ ). Si possono tracciare archi in senso antiorario (positivo) od orario (negativo) e i comandi rispettivi sono:

```
\pstArcOAB[⟨opzioni⟩]{⟨O⟩}{⟨A⟩}{⟨A⟩}
```

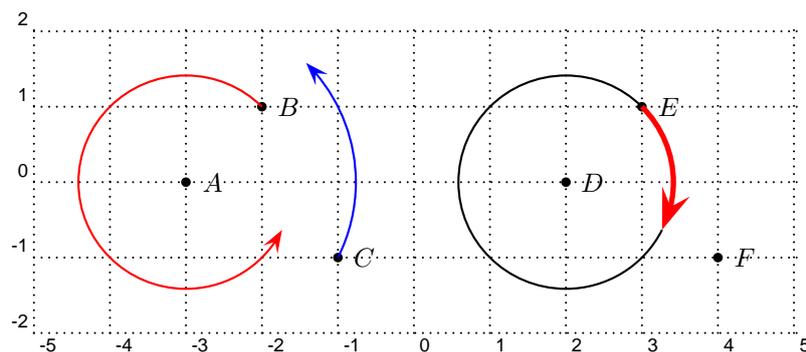
```
\pstArcnOAB[⟨opzioni⟩]{⟨O⟩}{⟨A⟩}{⟨A⟩}
```

Le opzioni sono quelle ormai solite; segnaliamo la possibilità di mettere delle frecce (e magari di dimensionarle con il parametro `arrowscale`).

**Esempio 1.10.** Archi di cerchio.

```
\begin{pspicture}(-5.5,-2.5)(5.5,2.5)
\pstGeonode(-3,0){A}(-2,1){B}(-1,-1){C}(2,0){D}(3,1){E}(4,-1){F}
\pstArcOAB[arrows=>,arrowscale=2,linecolor=red]{A}{B}{C}
\pstArcOAB[arrows=>,arrowscale=2,linecolor=blue]{A}{C}{B}
\pstArcOAB{D}{E}{F}
\pstArcnOAB[arrows=>,arrowscale=2,linecolor=red,linewidth=2pt]{D}{E}{F}
\end{pspicture}
```

Esistono in PSTricks dei comandi per tracciare frecce di qualunque tipo sia sui segmenti che sugli archi, con la possibilità di controllare nei minimi dettagli l'aspetto delle frecce stesse. Chi è interessato può consultare il manuale di PSTricks: nella maggior parte dei casi, comunque, l'introduzione di un semplice fattore di scala è più che sufficiente.



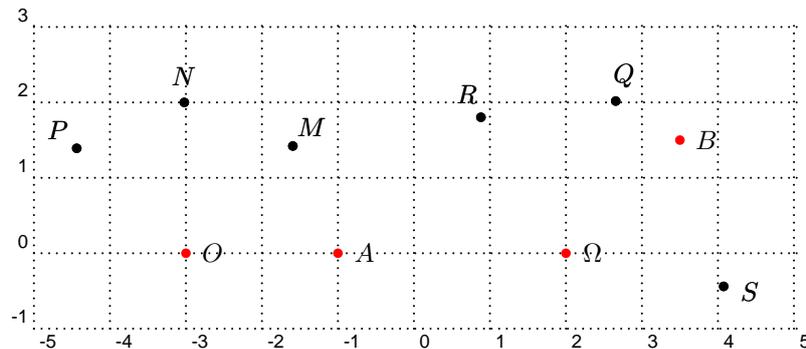
Infine l'ultimo comando utile per trattare i cerchi è il posizionamento di punti mediante ascissa curvilinea:

```
\pstCurvAbsNode[⟨opzioni⟩]{⟨O⟩}{⟨A⟩}{⟨B⟩}{⟨ascissa⟩}
```

Il punto  $B$  è piazzato sul cerchio di centro  $O$  e passante per  $A$ , mediante l'ascissa curvilinea  $ascissa$  che comincia su  $A$  ed è misurata in senso antiorario. Per usare il senso orario usare il parametro `CurvAbsNeg` ponendolo uguale a `true`. L'etichetta del punto  $B$  si piazza all'esterno del cerchio nella direzione del raggio, a meno che non sia variata con il solito parametro `PosAngle`. Il valore di  $ascissa$  può essere un numero, che rappresenta un'ascissa espressa in gradi, oppure può essere calcolata mediante i già visti parametri `\pstDistVal{x}` o `\pstDistAB{⟨A⟩}{⟨B⟩}`

**Esempio 1.11.** Ascisse curvilinee.

```
\begin{pspicture}(-5.5,-1.5)(5.5,3.5)
\pstGeonode[linecolor=red](-3,0){O}(-1,0){A}
\pstCurvAbsNode{O}{A}{M}{45}
\pstCurvAbsNode{O}{A}{N}{90}
\pstCurvAbsNode{O}{A}{P}{135}
\pstGeonode[PointName={\Omega,default},linecolor=red](2,0){O1}(3.5,1.5){B}
\pstCurvAbsNode{O1}{B}{Q}{\pstDistVal{1}}
\pstCurvAbsNode{O1}{B}{R}{\pstDistAB{A}{N}}
\pstCurvAbsNode[CurvAbsNeg=true]{O1}{B}{S}{60}
\end{pspicture}
```



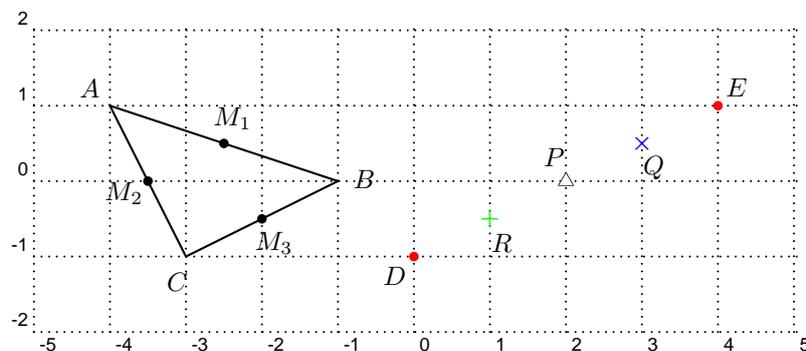
## 1.8 Punto medio di un segmento, baricentro di un triangolo

`\pstMiddleAB[opzioni]{A}{B}{M}`

Costruisce il punto medio  $M$  del segmento di estremi  $A$  e  $B$ , piazzando l'etichetta sopra o sotto il segmento. Si possono usare tutti i parametri ormai noti.

**Esempio 1.12.** Punto medio di un segmento.

```
\begin{pspicture}(-5.5,-2.5)(5.5,2.5)
\pstTriangle[PointSymbol=none](-4,1){A}(-1,0){B}(-3,-1){C}
\pstMiddleAB{A}{B}{M_1}
\pstMiddleAB{C}{A}{M_2}
\pstMiddleAB{B}{C}{M_3}
\pstGeonode[PosAngle={-135,45},linecolor=red](0,-1){D}(4,1){E}
\pstMiddleAB[PointSymbol=triangle,dotscale=1.5]{D}{E}{P}
\pstMiddleAB[PointSymbol=x,dotscale=1.5,linecolor=blue]{E}{P}{Q}
\pstMiddleAB[PointSymbol=+,dotscale=1.5,linecolor=green]{P}{D}{R}
\end{pspicture}
```



Il comando

`\pstCGravABC[opzioni]{A}{B}{C}{G}`

costruisce il baricentro  $G$  di un triangolo  $ABC$ . Tutti soliti parametri per controllare la figura possono essere utilizzati.

## 1.9 Assi, circonferenza per tre punti

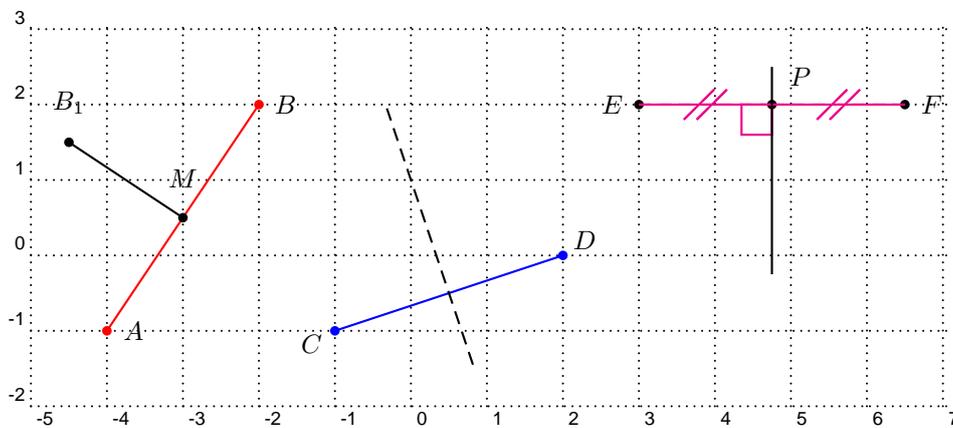
Il comando che segue traccia l'asse di un segmento dato mediante i suoi estremi  $A$  e  $B$ . Il tracciamento avviene nel seguente modo: prima viene tracciato il punto medio  $M$  di  $AB$ , successivamente si esegue una rotazione di  $90^\circ$  di  $B$  (cioè del secondo estremo del segmento dato) attorno ad  $M$ , ottenendo un punto  $B_1$  (rotazione in senso antiorario, come sempre), successivamente si traccia la retta  $B_1M$ . Naturalmente di default la retta coincide con il segmento  $AB$ : se la si vuole più lunga occorre usare il parametro `nodesep`. È dunque importante ai fini della figura scegliere l'ordine giusto degli estremi.

```
\pstMediatorAB[⟨opzioni⟩]{⟨A⟩}{⟨B⟩}{⟨M⟩}{⟨B_1⟩}
```

Il parametro `CodeFig`, se posto uguale a `true`, traccia anche il segmento  $AB$ , marca con il simbolo di default le due metà di  $AB$  e, infine, marca anche l'angolo retto: tutto questo viene fatto nel colore di default `cyan`. Se si vuole cambiare colore si può usare il parametro `CodeFigColor`. Se si vuole cambiare il simbolo di marcatura delle due metà di  $AB$ , si può usare il già noto parametro `SegmentSymbol`. L'aspetto dei due nuovi punti può essere controllato con i soliti parametri `PointSymbol`, `PointName`, `PointNameSep`, `PosAngle`, ecc.

**Esempio 1.13.** Asse di un segmento.

```
\begin{pspicture}(-5.5,-2.5)(7.5,3.5)
\pstGeonode[CurveType=polyline,linecolor=red](-4,-1){A}(-2,2){B}
\pstMediatorAB[PosAngle=90,PointNameSep=1.5em]{A}{B}{M}{B_1}
\pstGeonode[CurveType=polyline,linecolor=blue,PosAngle={-150,35}](-1,-1){C}(2,0){D}
\pstMediatorAB[PointSymbol=none,PointName=none,nodesep=-1,linestyle=dashed]{C}{D}{N}{D_1}
\pstGeonode[PosAngle={180,0}](3,2){E}(6.5,2){F}
\pstMediatorAB[PosAngle=45,PointNameSep=1.5em,PointNameB=none,PointSymbolB=none,%
nodesep=-0.5,CodeFig=true,CodeFigColor=magenta]{F}{E}{P}{E_1}
\end{pspicture}
```

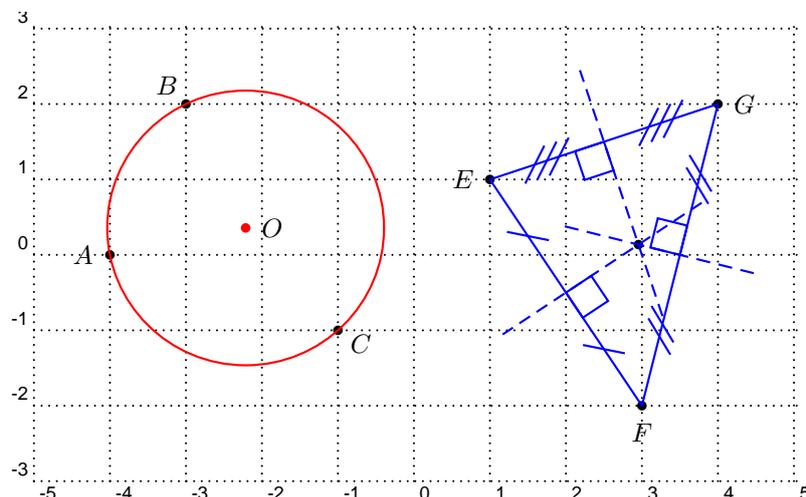


Una applicazione naturale del concetto di asse di un segmento è quella della costruzione del cerchio passante per tre punti, ovvero circoscritto ad un triangolo. Per questo è previsto il comando

```
\pstCircleABC[⟨opzioni⟩]{⟨A⟩}{⟨B⟩}{⟨C⟩}{⟨O⟩}
```

che, dati i tre punti  $A$ ,  $B$ ,  $C$ , traccia il punto  $O$  di intersezione degli assi dei segmenti individuati dai tre punti. Di default viene anche tracciato il cerchio per i tre punti: per evitare questo tracciamento basta mettere a `false` il parametro `DrawCirABC`. Mettendo a `true` il parametro `CodeFig` si ha anche il tracciamento delle linee di costruzione. Si possono usare tutti i parametri già visti.

**Esempio 1.14.** Cerchio per tre punti.



```

\begin{pspicture}(-5.5,-3.2)(5.5,3.2)
\pstGeonode[PosAngle={180,135,-30}](-4,0){A}(-3,2){B}(-1,-1){C}
\pstCircleABC[linecolor=red]{A}{B}{C}{O}
\pstGeonode[PosAngle={180,-90,0}](1,1){E}(3,-2){F}(4,2){G}
\pstCircleABC[DrawCirABC=false,CodeFig=true, CodeFigColor=blue,PointName=none]{E}{F}{G}{M}
\end{pspicture}

```

## 1.10 Bisettrici

Dato un angolo  $\widehat{ABC}$ , orientato sempre in senso antiorario, il comando

```
\pstBissectBAC[opzioni]{A}{B}{C}{M}
```

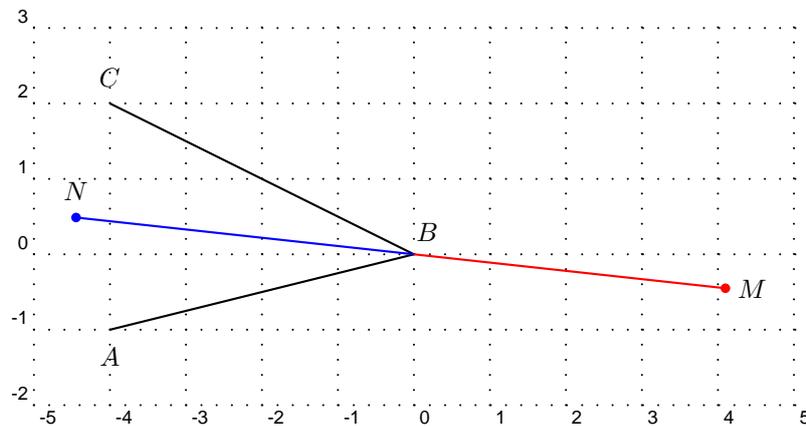
traccia un punto  $M$  sulla bisettrice e il segmento  $MA$  di bisettrice. Si possono usare tutti i parametri già noti.

**Esempio 1.15.** Bisettrici.

```

\begin{pspicture}(-5.5,-2.5)(5.5,3.5)
\pstGeonode[CurveType=polyline,PointSymbol=none,PosAngle={-90,60,90}](-4,-1){A}(0,0){B}(-4,2){C}
\pstBissectBAC[linecolor=red]{A}{B}{C}{M}
\pstBissectBAC[linecolor=blue,PosAngle=90]{C}{B}{A}{N}
\end{pspicture}

```



## 1.11 Premessa sulle trasformazioni geometriche

Le trasformazioni piane permettono facili costruzioni di figure anche complesse. Il pacchetto `pst-eucl` implementa un comando per tutte le trasformazioni elementari e, secondo me, questo costituisce uno dei punti cruciali di questo pacchetto dal punto di vista delle sue implicazioni didattiche.

Tutte le costruzioni hanno in comune il parametro `CodeFig` che permette di far apparire, se posizionato uguale a `true`, le linee di costruzione della figura. Il parametro è, per default, posto uguale a `false`. Si può anche scegliere uno stile per le linee di costruzione (`dashed` per default) e un colore mediante `CodeFigColor`, cyan per default, come già nel caso dell'asse di un segmento.

Le trasformazioni trattate sono:

- simmetria centrale;
- simmetria assiale (ovvero simmetria ortogonale);
- rotazione;
- traslazione;
- omotetia;
- proiezione ortogonale.

In tutte le costruzioni che seguono i punti ottenuti per simmetria, rotazione, ecc., hanno, di default, lo stesso nome dei punti originari, con l'aggiunta di un apice. Se volete specificare un nome diverso per tutti o alcuni dei punti originari aggiungete alla fine del comando, tra parentesi *quadre* (in quanto si tratta di di un parametro opzionale), i nuovi nomi che avete scelto: essi devono essere in numero minore od uguale ai nomi originari. Prestate attenzione: se costruite due diversi simmetrici di uno stesso punto, senza specificare un nuovo nome per i punti costruiti, essi avranno lo stesso nome (il nome originale più un apice) e questo potrebbe creare confusione.

## 1.12 Simmetrie centrale e assiale

Il comando base per la simmetria centrale è

```
\pstSymO[⟨opzioni⟩]{⟨O⟩}{⟨A1, A2, ...⟩}
```

dove  $O$  è il centro di simmetria e  $A_1, A_2, \dots$  sono punti dati. Il comando costruisce i simmetrici dei punti dati con lo stesso nome (per default) con l'aggiunta dell'apice.

Il comando base per la simmetria assiale è

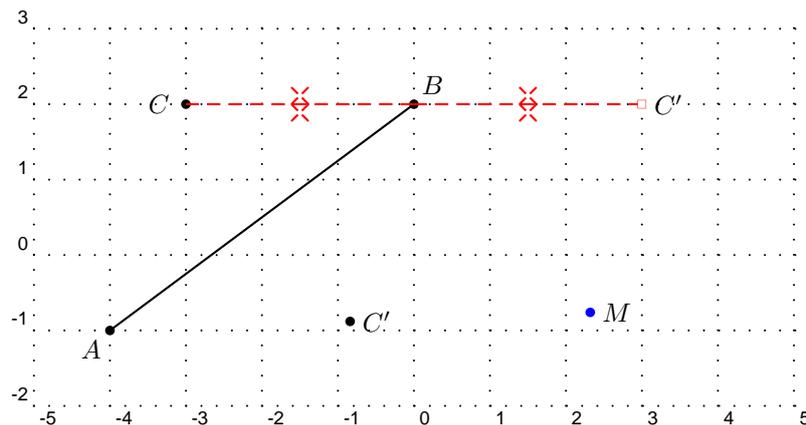
```
\pstOrtSym[⟨opzioni⟩]{⟨A⟩}{⟨B⟩}{⟨A1, A2, ...⟩}
```

dove  $A$  e  $B$  individuano l'asse di simmetria e  $A_1, A_2, \dots$  sono punti dati. Il comando costruisce i simmetrici dei punti dati.

Possono naturalmente essere usati tutti i parametri già noti per i nuovi punti costruiti.

**Esempio 1.16.** Simmetrie centrale ed assiale.

```
\begin{pspicture}(-5.5,-2.5)(5.5,3.5)
\pstGeonode[CurveType=polyline,PosAngle={-135,45}](-4,-1){A}(0,2){B}
\pstGeonode[PosAngle=180](-3,2){C}
\pstSymO[CodeFig=true,CodeFigColor=red,PointSymbol=square,%
linecolor=red,SegmentSymbol=MarkCross]{B}{C}
\pstOrtSym{A}{B}{C}
\pstSymO{C'}{A}
\end{pspicture}
```



## 1.13 Rotazioni

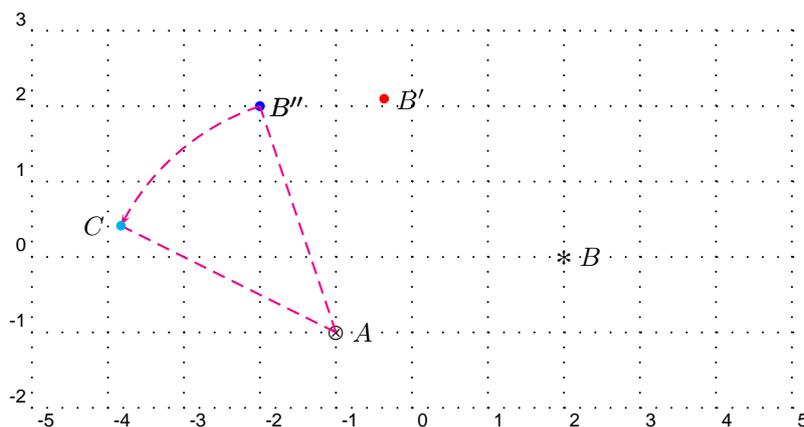
Il comando di base per le rotazioni

```
\pstRotation[⟨opzioni⟩]{⟨O⟩}{⟨A1, A2, ...⟩}
```

costruisce i punti ruotati dei punti dati  $A_1, A_2, \dots$  attorno al centro  $O$ ; l'angolo di default di rotazione è di  $60^\circ$ , ma si può assegnare un angolo qualunque. Si può anche usare un angolo orientato definito da tre punti e in questo caso occorre utilizzare la funzione  $\text{\pstAngleAOB}\{\langle A \rangle\}\{\langle O \rangle\}\{\langle B \rangle\}$ ; il parametro  $\text{AngleCoef}$  permette di applicare un coefficiente moltiplicativo all'angolo  $\widehat{AOB}$ .

**Esempio 1.17.** Rotazioni di punti.

```
\begin{pspicture}(-5.5,-2.5)(5.5,3.5)
\pstGeonode[PointSymbol={otimes,asterisk},dotsscale=1.5](-1,-1){A}(2,0){B}
\pstRotation[linecolor=red]{A}{B}
\pstRotation[linecolor=blue,RotAngle=30]{A}{B'}
\pstRotation[linecolor=blue,RotAngle=30]{A}{B''}
\pstRotation[PosAngle=180,AngleCoef=1.5,RotAngle=\pstAngleAOB{B'}{A}{B''},%
linecolor=cyan,CodeFig=true,CodeFigColor=magenta]{A}{B''}{C}
\end{pspicture}
\end{center}
```



## 1.14 Traslazioni

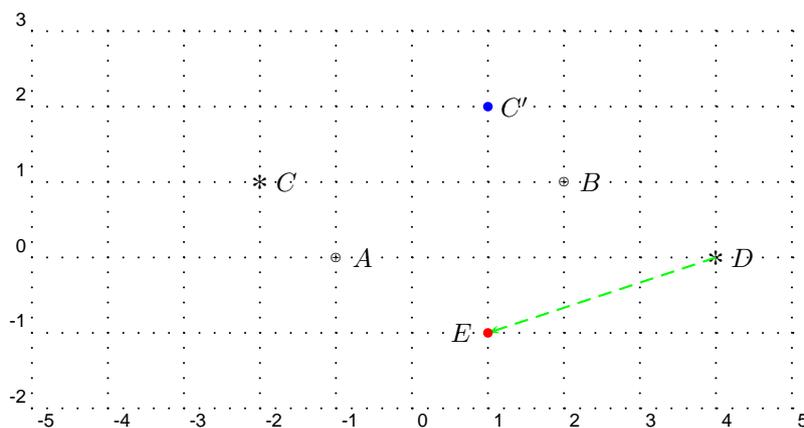
Il comando di base per le traslazioni

```
\pstTranslation[⟨opzioni⟩]{⟨A⟩}{⟨B⟩}{⟨P1, P2, ...⟩}
```

costruisce i traslati dei punti  $A_1, A_2, \dots$  con vettore di traslazione  $\overrightarrow{AB}$ .

**Esempio 1.18.** Traslazioni di punti.

```
\begin{pspicture}(-5.5,-2.5)(5.5,3.5)
\pstGeonode[PointSymbol=oplus](-1,0){A}(2,1){B}
\pstGeonode[PointSymbol=asterisk,dotscale=1.5](-2,1){C}
\pstTranslation[linecolor=blue]{A}{B}{C}
\pstGeonode[PointSymbol=asterisk,dotscale=1.5](4,0){D}
\pstTranslation[linecolor=red,CodeFig=true,CodeFigColor=green,PosAngle=180]{B}{A}{D}[E]
\end{pspicture}
```



## 1.15 Omotetie

Per costruire un'omotetia bisogna assegnare il coefficiente di omotetia, che si specifica utilizzando il parametro `HomCoef`. Il comando base è poi

```
\pstHomO[⟨opzioni⟩]{⟨O⟩}{⟨A1, A2, ...⟩}
```

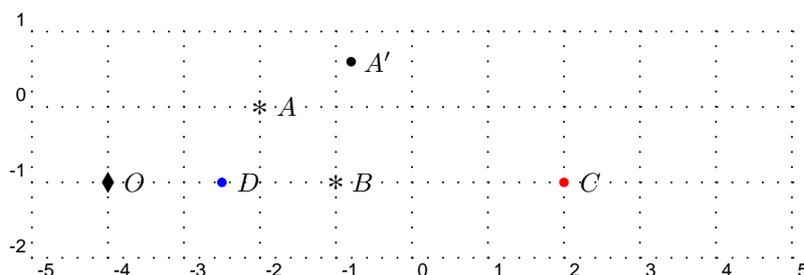
**Esempio 1.19.** Omotetie.

```
\begin{pspicture}(-5.5,-2.5)(5.5,1.5)
\pstGeonode[dotscale=1.5,PointSymbol={diamond*,asterisk,asterisk,}]%
```

```

(-4,-1){O}(-2,0){A}(-1,-1){B}
\pstHomO[HomCoef=1.6]{O}{A}
\pstHomO[HomCoef=2,linecolor=red]{O}{B}{C}
\pstHomO[HomCoef=0.5,linecolor=blue]{O}{B}{D}
\end{pspicture}

```

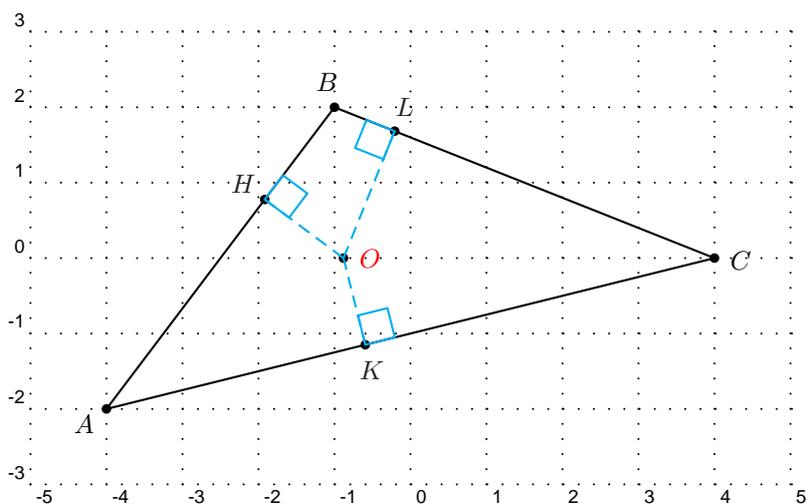


## 1.16 Proiezioni ortogonali

Questo comando, utile per esempio per tracciare le altezze di un triangolo, proietta ortogonalmente una serie di punti  $P_1, P_2, \dots$  sulla retta individuata da due punti  $A$  e  $B$ . Il comando base è

```
\pstProjection[⟨opzioni⟩]{⟨A⟩}{⟨B⟩}{⟨P1, P2, ...⟩}
```

**Esempio 1.20.** Proiezioni ortogonali.



```

\begin{pspicture}(-5.5,-3.5)(5.5,3.5)
\pstTriangle(-4,-2){A}(-1,2){B}(4,0){C}
{\color{red} \pstGeonode(-1,0){O}}
\pstProjection[CodeFig=true]{A}{B}{O}[H]
\pstProjection[CodeFig=true]{A}{C}{O}[K]
\pstProjection[CodeFig=true]{C}{B}{O}[L]
\end{pspicture}

```

In questo esempio abbiamo anche voluto mostrare come si possano colorare i nomi dei punti: basta racchiudere completamente il comando che li crea tra parentesi graffe e inserire il comando `\color{colorescelto}`; si tratta di una delle opzioni standard di PSTricks e non specifica di pst-eucl.

## 1.17 Intersezioni

Un sistema fondamentale per definire punti nella geometria è quello di considerare intersezioni: nella geometria euclidea coinvolgendo solo rette e circonferenze, più in generale coinvolgendo anche curve più complesse. Il

pacchetto che stiamo considerando considera sei tipi di intersezioni, per cui sono implementati i comandi qui di seguito descritti. Tutti i comandi prevedono che le intersezioni esistano. Se una o più di esse non ci sono, i rispettivi punti dichiarati vengono piazzati in una posizione di default (l'origine degli assi, il centro del cerchio, ...). È evidente che, per tracciare funzioni, bisogna caricare il pacchetto relativo (`pst-plot`).

#### 1. Retta-retta

$$\boxed{\backslash\text{pstInterLL}[\langle\text{opzioni}\rangle]\{\langle A\rangle\}\{\langle B\rangle\}\{\langle C\rangle\}\{\langle D\rangle\}\{\langle I\rangle\}}$$

$AB$  e  $CD$  sono le rette,  $I$  il punto di intersezione.

#### 2. Retta-cerchio

$$\boxed{\backslash\text{pstInterLC}[\langle\text{opzioni}\rangle]\{\langle A\rangle\}\{\langle B\rangle\}\{\langle O\rangle\}\{\langle C\rangle\}\{\langle I_1\rangle\}\{\langle I_2\rangle\}}$$

$AB$  è la retta,  $O$  è il centro del cerchio,  $C$  un punto del cerchio,  $I_1$  e  $I_2$  le intersezioni. Si può evitare di specificare il punto  $C$ , lasciando vuota la coppia di parentesi relativa; in questo caso bisogna precisare il Raggio o il Diametro, come spiegato nella costruzione di un cerchio.

#### 3. Cerchio-cerchio

$$\boxed{\backslash\text{pstInterCC}[\langle\text{opzioni}\rangle]\{\langle O_1\rangle\}\{\langle B\rangle\}\{\langle O_2\rangle\}\{\langle C\rangle\}\{\langle I_1\rangle\}\{\langle I_2\rangle\}}$$

I due cerchi sono definiti come nell'esempio precedente (anche con le stesse possibili varianti, con l'aggiunta dei parametri `RadiusA`, `RadiusB`, `DiameterA`, `DiameterB`, con ovvio significato dei simboli). Si possono usare anche i parametri `CodeFig`, `CodeFigA`, `CodeFigB` per far comparire gli archi a livello delle intersezioni. Sono poi previsti due parametri booleani per scegliere il senso orario o antiorario degli archi a livello delle intersezioni: `CodeFigArc` e `CodeFigBarc`.

#### 4. Funzione-funzione

$$\boxed{\backslash\text{pstInterFF}[\langle\text{opzioni}\rangle]\{\langle f\rangle\}\{\langle g\rangle\}\{\langle x_0\rangle\}\{\langle I\rangle\}}$$

Le due funzioni vanno scritte, al solito, con il codice Postscript. Il metodo utilizza l'algoritmo di Newton e  $x_0$  è il punto iniziale.  $I$  è il punto di intersezione. Naturalmente il metodo può non funzionare, nel qual caso l'intersezione è piazzata nell'origine.

#### 5. Funzione-retta

$$\boxed{\backslash\text{pstInterFL}[\langle\text{opzioni}\rangle]\{\langle f\rangle\}\{\langle A\rangle\}\{\langle B\rangle\}\{\langle x_0\rangle\}\{\langle I\rangle\}}$$

$f$  è la funzione,  $AB$  è la retta,  $x_0$  e  $I$  sono come nel caso precedente.

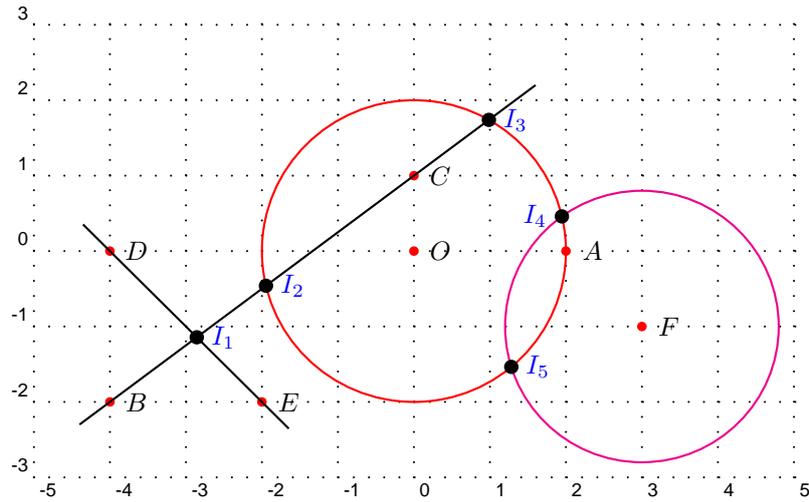
#### 6. Funzione-cerchio

$$\boxed{\backslash\text{pstInterFC}[\langle\text{opzioni}\rangle]\{\langle f\rangle\}\{\langle O\rangle\}\{\langle A\rangle\}\{\langle x_0\rangle\}\{\langle I\rangle\}}$$

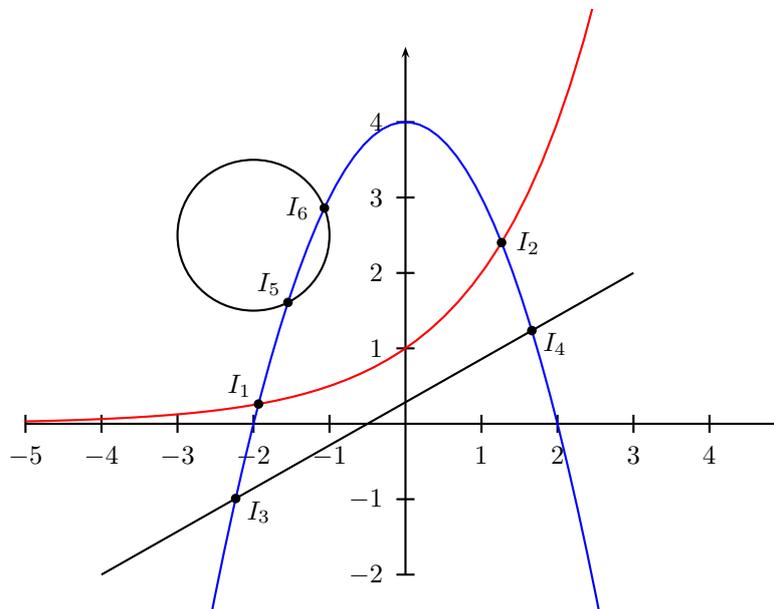
$f$  è la funzione,  $O$  è il centro e  $A$  un punto del cerchio,  $x_0$  e  $I$  sono come nel caso precedente.

**Esempio 1.21.** Intersezioni con rette e cerchi.

```
\begin{pspicture}(-5.5,-3.5)(5.5,3.5)
\pstGeonode[linecolor=red](0,0){O}(2,0){A}(-4,-2){B}(0,1){C}(-4,0){D}(-2,-2){E}(3,-1){F}
\pstCircleOA[linecolor=red]{O}{A}
\pstCircleOA[Radius=\pstDistVal{1.8},linecolor=magenta]{F}{O}
\pstLineAB[nodesepA=-0.5,nodesepB=-2]{B}{C}
\pstLineAB[nodesep=-0.5]{D}{E}
\color{blue} \pstInterLL[dotsscale=1.5]{B}{C}{D}{E}{I_1}
\pstInterLC[dotsscale=1.5]{B}{C}{O}{A}{I_2}{I_3}
\pstInterCC[dotsscale=1.5,RadiusB=\pstDistVal{1.8},PosAngleA=180]{O}{A}{F}{O}{I_4}{I_5}
\end{pspicture}
```



**Esempio 1.22.** Intersezioni con funzioni reali.



```

\begin{pspicture*}(-5.5,-2.5)(5.5,5.5)
\psaxes{->}(0,0)(-5,-2)(5,5)
\pstGeonode[PointSymbol=none,PointName=none](-4,-2){A}(3,2){B}(-2,2.5){C}(-2,3.5){D}
\pstLineAB{A}{B}
\pstCircleOA{C}{D}
\def\F{2 x exp}
\def\G{x 2 exp -1 mul 4 add}
\psplot[linecolor=red]{-5}{4}{\F}
\psplot[linecolor=blue]{-3}{3}{\G}
\pstInterFF[PosAngle=135]{\F}{\G}{-2}{I_1}
\pstInterFF{\F}{\G}{2}{I_2}
\pstInterFL[PosAngle=-30]{\G}{A}{B}{-2}{I_3}
\pstInterFL[PosAngle=-30]{\G}{A}{B}{2}{I_4}
\pstInterFC[PosAngle=135]{\G}{C}{D}{-1.5}{I_5}
\pstInterFC[PosAngle=180]{\G}{C}{D}{-1}{I_6}
\end{pspicture*}

```

## Capitolo 2

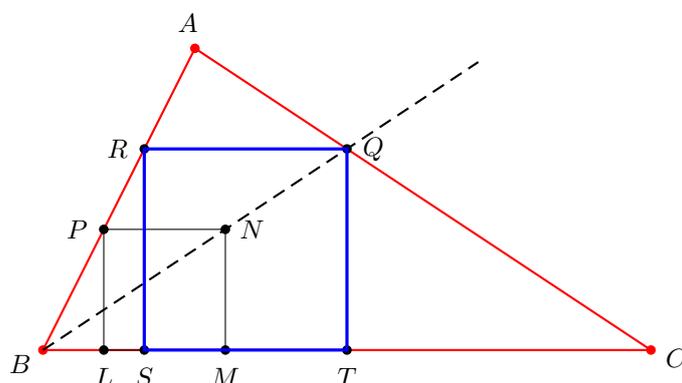
# Applicazioni didattiche

In questa seconda parte della miniguia voglio mostrare su alcuni semplici esempi come si possa usare il pacchetto `pst-eucl` per costruzioni geometriche con riga e compasso o per illustrare esercizi di geometria. Lo sforzo necessario a costruire le figure geometriche con questo sistema è spesso notevole e costituisce un ottimo esercizio, non tradizionale, sia di geometria vera e propria che di informatica.

Naturalmente tra gli scopi di questi esempi c'è anche quello di mostrare le potenzialità del pacchetto in esame: in ogni caso per alcuni esempi interessanti invito a consultare la guida originale di Dominique Rodriguez, distribuita con il pacchetto.

### 2.1 Costruzioni geometriche

**Esercizio 2.1.** Dato un triangolo qualunque  $ABC$ , inscrivere in esso un quadrato avente un lato sul lato  $BC$ , e i restanti due vertici uno sul lato  $AB$  e uno sul lato  $AC$ .



La costruzione può procedere con i passi qui di seguito descritti.

- Naturalmente si comincia col costruire il triangolo  $ABC$ .
- Si prende poi un punto  $P$  sul lato  $AB$ : si può usare convenientemente una omotetia di centro  $B$  e coefficiente minore di 1, per esempio 0.4.
- Si proietta il punto  $P$  su  $BC$ , ottenendo il punto  $L$ .
- Si costruisce poi il quadrato  $PLMN$ , determinando i punti  $M$  ed  $N$  mediante rotazioni attorno a  $P$  e  $L$ .
- Si può a questo punto tracciare la linea  $BN$  e ottenere la sua intersezione,  $Q$ , con il lato  $AC$ .
- Non resta altro da fare che proiettare  $Q$  in  $T$  su  $BC$  per ottenere il lato del quadrato voluto. Gli altri due vertici  $S$  ed  $R$  si ottengono per rotazione attorno a  $T$  e  $S$ .
- Qualche aggiustamento sulle posizioni dei nomi dei punti e sul colore delle varie parti completa il disegno.

Il codice completo della figura rispetta i passi indicati.

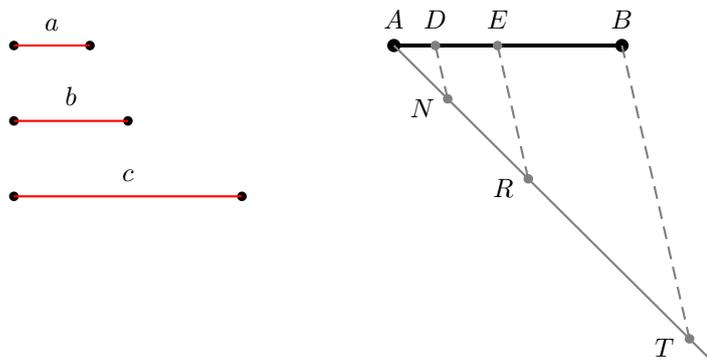
```
\begin{pspicture}(-5,-3)(5,3)
\pstTriangle[linecolor=red](-4,-2){B}(4,-2){C}(-2,2){A}
\pstHomO[HomCoef=.4,PosAngle=180]{B}{A}[P]
\pstProjection{B}{C}{P}[L]
\pstRotation[RotAngle=90]{P}{L}[N]
```

```

\pstRotation[RotAngle=-90,PosAngle=-90]{L}{P}[M]
\pspolygon[linewidth=.5\pslinewidth](P)(L)(M)(N)
\pstLineAB[nodesepB=-4,linestyle=dashed]{B}{N}
\pstInterLL{A}{C}{B}{N}{Q}
\pstProjection{B}{C}{Q}[T]
\pstRotation[RotAngle=90,PosAngle=-90]{T}{Q}[S]
\pstRotation[RotAngle=90,PosAngle=180]{S}{T}[R]
\pspolygon[linewidth=1.5\pslinewidth,linecolor=blue](Q)(R)(S)(T)
\end{pspicture}

```

**Esercizio 2.2.** Dividere un segmento in parti proporzionali a più segmenti dati.



La costruzione può procedere con i passi qui di seguito descritti.

- Si comincia col costruire i tre segmenti dati e il segmento che si vuole dividere. Come esempio interessante ho usato, per costruire i tre segmenti dati, il comando di PSTricks `\pcline` (pointconnectionline) seguito immediatamente dal comando `\Aput`, che piazza il contenuto tra parentesi graffe sopra la linea appena tracciata (il comando `\Bput` lo metterebbe sotto, ma ci sono molte possibilità di controllo preciso della posizione, per cui vi rimando al manuale del pacchetto PSTricks). Il segmento da dividere l'ho costruito invece usando il parametro di `pst-eucl CurveType`.
- Devo ora costruire una semiretta di origine  $A$  e direzione qualunque. Usando una rotazione oraria attorno ad  $A$  di un angolo di  $45^\circ$  ho costruito il punto  $C$ , che non ho tracciato, e poi la linea  $AC$ .
- Successivamente si deve riportare il segmento  $a$ , a partire da  $A$ , sulla semiretta  $AC$ . Per questo ho usato l'intersezione tra il cerchio di raggio uguale al segmento  $a$  e di centro  $A$  con la semiretta  $AC$ . Poiché ci sono due intersezioni, ho evidenziato solo quella che interessa, chiamandola  $N$ . Nello stesso modo, a partire da  $N$  ho trovato  $R$  e poi a partire da  $R$  ho trovato  $T$ .
- Il passo successivo consiste nel congiungere  $T$  con  $B$  e nel trovare le intersezioni tra  $AB$  e le parallele a  $TB$  condotte per  $R$  e  $N$ . Per costruire le parallele il metodo migliore è quello di traslare il punto  $R$  (e poi  $N$ ) del vettore  $\overrightarrow{TB}$ , ottenendo i punti  $R1$  ed  $N1$  (che non sono stati rappresentati e che in ogni caso uscirebbero dallo spazio riservato alla figura). Le intersezioni richieste sono ora immediate con il comando `\pstInterLL`.
- L'ultimo passo consiste, come al solito, nell'aggiustamento delle posizioni dei punti e nell'uso del colore per facilitare la lettura della figura. Consiglio sempre di lasciare questa parte al termine, preoccupandosi prima della costruzione vera e propria.

Il codice completo della figura rispetta i passi indicati.

```

\begin{pspicture*}(-5.5,-3.5)(5.5,2.5)
\pstGeonode[PointName=none](-4,2){P1}(-3,2){P2}(-4,1){P3}%
(-2.5,1){P4}(-4,0){P5}(-1,0){P6}
\pstGeonode[CurveType=polyline,linewidth=1.5pt,PosAngle=90](1,2){A}(4,2){B}
\psset{linecolor=red}
\pcline(P1)(P2)
\Aput{a}
\pcline(P3)(P4)
\Aput{b}
\pcline(P5)(P6)

```



- Successivamente si trova il punto medio  $M$  di  $AS$ .
- Per tracciare la perpendicolare per  $B$  ad  $AB$  ho ruotato  $S$  in  $S_1$  di  $90^\circ$ , attorno a  $B$ .
- Una delle due intersezioni,  $C$  e  $D$ , tra la retta perpendicolare  $BS_1$  e la circonferenza di centro  $M$  e passante per  $S$  consente di costruire il triangolo rettangolo  $ASC$  (oppure  $ASD$ ); il cateto  $SC$  è il medio proporzionale che stiamo cercando.
- Basta intersecare la circonferenza di centro  $S$  e passante per  $C$  con la retta  $r$  data, per trovare i punti  $T_1$  e  $T_2$ . Le circonferenze richieste sono quelle che passano per  $A, B$  e  $T_1$ , oppure per  $A, B$  e  $T_2$ .
- Al solito, per concludere dobbiamo solo fare degli aggiustamenti estetici.

Il codice completo della figura rispetta i passi indicati.

```
\begin{pspicture}(-5.5,-3.5)(5.5,3.5)
{\color{red} \pstGeonode[PosAngle=55](0,-1){B}(-0.3,-0.2){A}}
\pstGeonode[PointName=none,PointSymbol=none](-4,-3){X}(4,-2){Y}
\pstLineAB[nodesep=-1,linecolor=red]{X}{Y}
\pstInterLL[PosAngle=-80]{A}{B}{X}{Y}{S}
{\color{green} \pstMiddleAB[PointNameSep=12pt,linecolor=green]{A}{S}{M}}
\pstRotation[RotAngle=90,PointSymbol=none,PointName=none]{B}{S}{S1}
\pstInterLC[PosAngle=-20]{B}{S1}{M}{S}{D}{C}
\pspolygon[linecolor=magenta,linestyle=dashed,dash=3pt 3pt](A)(C)(S)
\pstLineAB[linestyle=dotted,nodesep=-1,linecolor=gray]{D}{C}
\pstCircleOA[linecolor=gray,linestyle=dashed]{M}{C}
\pstInterLC[PosAngle=-80]{X}{Y}{S}{C}{T1}{T2}
\pstCircleABC[linecolor=blue,PointName=none]{A}{B}{T2}{O2}
\pstCircleABC[linecolor=blue,PointName=none]{A}{B}{T1}{O1}
\end{pspicture}
```

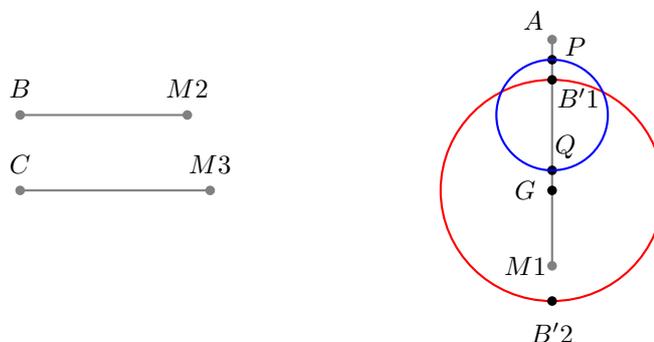
**Esercizio 2.4. Un problema non proprio elementare:** costruire un triangolo date le tre mediane.

Questa costruzione è decisamente più complessa delle precedenti e richiede anche alcuni passaggi teorici che discuterò brevemente.

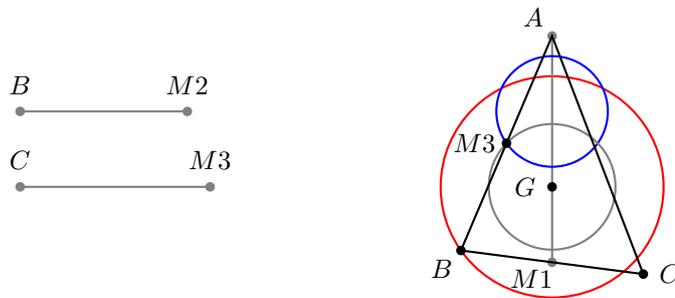
Se sono date le tre mediane  $AM_1$ ,  $BM_2$  e  $CM_3$ , il baricentro si trova, per ciascuna di esse, con la nota proprietà: il baricentro divide ciascuna mediana in due parti di cui quella che contiene il vertice è doppia dell'altra. Tracciamo allora una delle tre mediane (per esempio  $AM_1$  per fissare le idee) e posizioniamo su di essa il baricentro  $G$ .

Osserviamo poi che il punto  $B$  si deve trovare a distanza da  $G$  uguale a  $\frac{2}{3}\overline{BM_2}$ , ovvero sulla circonferenza, diciamola  $\mathcal{C}_1$ , di centro  $G$  e raggio  $\frac{2}{3}\overline{BM_2}$ .

Se ora congiungiamo  $A$  con un punto qualunque,  $B'$  di  $\mathcal{C}_1$ , possiamo osservare che il punto  $M_3$ , corrispondente a questa scelta di  $B$ , sarà il punto medio tra  $A$  e  $B'$ . Ebbene si può provare che al variare di  $B'$  su  $\mathcal{C}_1$ , il punto medio di  $AB'$  descrive una circonferenza (e questa è la parte teoricamente *difficile* della costruzione), il cui diametro è il segmento  $PQ$  indicato nella figura di seguito ( $B'_1$  e  $B'_2$  sono due possibili posizioni di  $B'$ ).



D'altro canto il punto  $M_3$  deve stare anche sulla circonferenza, diciamola  $\mathcal{C}_2$ , di centro  $G$  e raggio  $GM_3$ . A questo punto la posizione di  $M_3$  è determinata (a meno di una simmetria evidente). Il gioco è fatto:  $B$  è il simmetrico di  $A$  rispetto ad  $M_3$  e  $C$  è il simmetrico di  $B$  rispetto ad  $M_1$ .

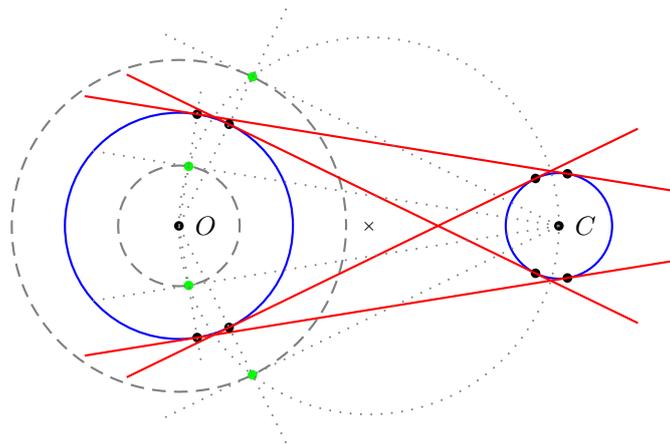


Segue il codice completo della figura.

```
\begin{pspicture}(-7.5,-2)(5.5,2.5)
\pstGeonode[CurveType=polyline,linecolor=gray,PosAngle=90](-7,1){B}(-4.8,1){M2}
\pstGeonode[CurveType=polyline,linecolor=gray,PosAngle=90](-7,0){C}(-4.5,0){M3}
\pstCircleOA[DistCoef=2 3 div,Radius=\pstDistAB{B}{M2},linecolor=red]{G}{}
\pstGeonode[CurveType=polyline,linecolor=gray,PosAngle={135,-140}](0,2){A}(0,-1){M1}
\pstHomO[HomCoef=2 3 div,PosAngle=180]{A}{M1}[G]
\pstInterLC[DistCoef=2 3 div,Radius=\pstDistAB{B}{M2},%
PointName=none,PointSymbol=none]{A}{G}{G}{}{B'1}{B'2}
\pstMiddleAB[PosAngle=30,PointName=none,PointSymbol=none]{A}{B'1}{P}
\pstMiddleAB[PosAngle=60,PointName=none,PointSymbol=none]{A}{B'2}{Q}
\pstMiddleAB[PointName=none,PointSymbol=none]{P}{Q}{O}
\pstCircleOA[linecolor=blue]{O}{Q}
\pstCircleOA[DistCoef=1 3 div,Radius=\pstDistAB{C}{M3},linecolor=gray]{G}{}
\pstInterCC[DistCoef=1 3 div,RadiusA=\pstDistAB{C}{M3},PointNameA=M3,PointNameB=none,%
PointSymbolB=none,PosAngleA=180,PointNameSep=1.2em]{G}{}{O}{Q}{R}{S}
\pstSymO[PointName=B,PosAngle=-135]{R}{A}[B1]
\pstSymO[PointName=C]{M1}{B1}[C1]
\pspolygon(A)(B1)(C1)
\end{pspicture}
```

**Esercizio 2.5.** Costruire le tangenti comuni a due circonferenze date.

Si tratta di un problema classico. Proponiamo solo la costruzione e il codice, senza commenti.



Il codice completo della figura.

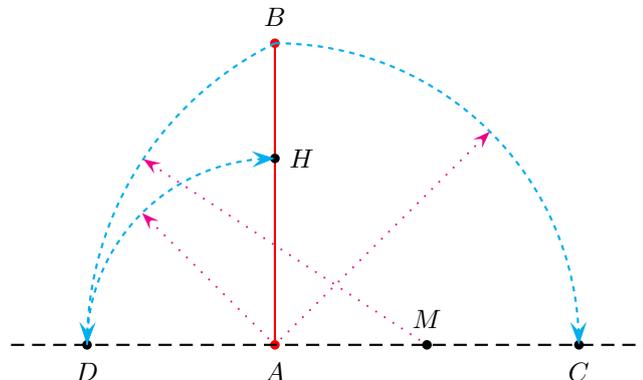
```
\begin{pspicture}(-5.5,-3)(5.5,3)
\pstGeonode[PointName={default,none,default,none},%
PointSymbol={default,none,default,none}](-2,0){O}(-0.5,0){A}(3,0){C}(2.3,0){B}
\psset{PointName=none}
\pstCircleOA[linecolor=blue]{O}{A}
\pstCircleOA[linecolor=blue]{C}{B}
\pstInterLC[Radius=\pstDistAB{C}{B},PointSymbol=none]{A}{B}{A}{}{Z}{W}
```

```

\pstCircleOA[linestyle=dashed,linecolor=gray]{O}{Z}
\pstCircleOA[linestyle=dashed,linecolor=gray]{O}{W}
\pstMiddleAB[PointSymbol=x]{O}{C}{M}
\pstInterCC[linecolor=green]{M}{O}{O}{Z}{R}{S}
\pstInterLC[PointSymbolA=none]{O}{S}{O}{A}{S2}{S1}
\pstTranslation{S}{S1}{C}[C1]
\pstLineAB[nodesep=-1.5,linecolor=red]{C1}{S1}
\pstInterLC[PointSymbolA=none]{O}{R}{O}{A}{R2}{R1}
\pstTranslation{R}{R1}{C}[C2]
\pstLineAB[nodesep=-1.5,linecolor=red]{C2}{R1}
\pstInterCC[linecolor=green]{M}{O}{O}{W}{T}{U}
\pstInterLC[PointSymbolA=none]{O}{U}{O}{A}{U2}{U1}
\pstTranslation{U}{U1}{C}[C3]
\pstLineAB[nodesep=-1.5,linecolor=red]{C3}{U1}
\pstInterLC[PointSymbolA=none]{O}{T}{O}{A}{T2}{T1}
\pstTranslation{T}{T1}{C}[C4]
\pstLineAB[nodesep=-1.5,linecolor=red]{C4}{T1}
%Le seguenti sono solo le linee di costruzione usate
\pstLineAB[nodesepB=-1.4,linecolor=gray,linestyle=dotted]{C}{R}
\pstLineAB[nodesepB=-1.4,linecolor=gray,linestyle=dotted]{C}{S}
\pstLineAB[nodesepB=-1.4,linecolor=gray,linestyle=dotted]{C}{T}
\pstLineAB[nodesepB=-1.4,linecolor=gray,linestyle=dotted]{C}{U}
\pstLineAB[nodesepB=-1,linecolor=gray,linestyle=dotted]{O}{R}
\pstLineAB[nodesepB=-1,linecolor=gray,linestyle=dotted]{O}{S}
\pstLineAB[nodesepB=-1,linecolor=gray,linestyle=dotted]{O}{T}
\pstLineAB[nodesepB=-1,linecolor=gray,linestyle=dotted]{O}{U}
\pstCircleOA[linecolor=gray,linestyle=dotted]{M}{O}
\end{pspicture}

```

**Esercizio 2.6.** Costruire la sezione aurea di un segmento.



Il codice completo della figura.

```

\begin{pspicture}(-5.5,-2.5)(5.5,2.5)
\pstGeonode[CurveType=polyline,linecolor=red,PosAngle={-90,90}](0,-2){A}(0,2){B}
\pstRotation[RotAngle=-90,PosAngle=-90]{A}{B}[C]
\pstMiddleAB{A}{C}{M}
\pstInterLC[PointNameB=none,PointSymbolB=none,PosAngle=-90]{A}{C}{M}{B}{D}{E}
\pstInterLC[PointNameA=none,PointSymbolA=none]{A}{B}{A}{D}{K}{H}
%Segue la visualizzazione di alcune linee di costruzione
\pstLineAB[nodesep=-1,linestyle=dashed]{D}{C}
\psset{arrows=->,arrowscale=2,dash=2pt 2pt,linestyle=dashed,linecolor=cyan}
\pstArcnOAB{A}{B}{C}
\pstArcnOAB{M}{B}{D}
\pstArcnOAB{A}{D}{H}
\psset{arrows=->,arrowscale=1.5,linestyle=dotted,linecolor=magenta}
\pstRotation[RotAngle=-45,PointName=none,PointSymbol=none]{A}{B}[Q]
\pstLineAB{A}{Q}
\pstRotation[RotAngle=30,PointName=none,PointSymbol=none]{M}{B}[T]

```

```

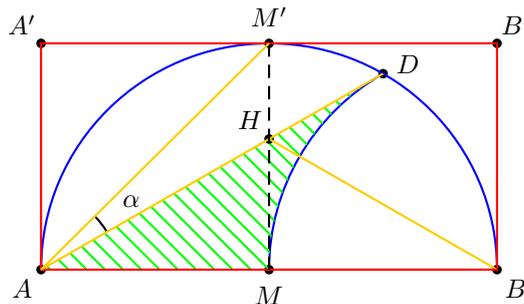
\pstLineAB{M}{T}
\pstRotation[RotAngle=45,PointName=none,PointSymbol=none]{A}{H}[U]
\pstLineAB{A}{U}
\end{pspicture}

```

## 2.2 Esercizi proposti

**Esercizio 2.7.** Della figura seguente si sa che:

1. il quadrilatero  $(ABB'A')$  è un rettangolo;
2.  $M$  ed  $M'$  sono i punti medi di  $AB$  e  $A'B'$ ;
3.  $\overline{MB} = \overline{BD} = 1$ .



Ti viene richiesto:

1. Se  $H = MM' \cap AD$ ,  $K = BH \cap \widehat{MD}$  è il punto medio dell'arco circolare  $\widehat{MD}$ ?
2. Qual è la misura in gradi dell'angolo  $\alpha = \widehat{DAM}$ ?
3. Qual è la misura dell'area del triangolo mistilineo di lati  $AD$ ,  $AM$ ,  $\widehat{MD}$ , evidenziato?

Segue il codice della figura inserita nell'esercizio, che contiene anche parecchie istruzioni proprie di PSTricks, per le quali rimando al relativo manuale.

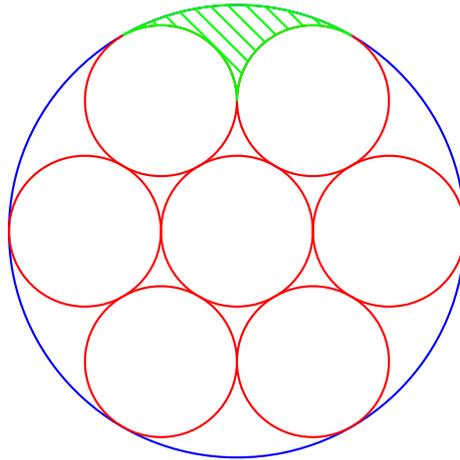
```

\begin{pspicture}(-3.5,-0.5)(3.5,3.5)
\definecolor{Arancio}{rgb}{1,0.8,0}
\pstGeonode[PosAngle={-135,-90,-45,45,90,135,20}]{(-3,0){A}%
(0,0){M}(3,0){B}(3,3){B'}(0,3){M'}(-3,3){A'}(3,60){D}
\psarc[linecolor=blue](M){3}{0}{180}
\psarc[linecolor=blue](B){3}{120}{180}
\pscustom[fillstyle=vlines,linestyle=none,hatchcolor=green]{%
\psline(M)(A)(D)
\psarc(B){3}{120}{180}
}
\pstInterLL[PosAngle=135]{A}{D}{M'}{M}{H}
\psline[linecolor=Arancio](H)(B)
\pstMarkAngle[LabelSep=1.5,MarkAngleRadius=1]{D}{A}{M'}{\alpha}
\pspolygon[linecolor=red](A)(B)(B')(A')
\psline[linecolor=Arancio](A)(D)
\psline[linestyle=dashed](M')(M)
\psline[linecolor=Arancio](A)(M')
\end{pspicture}

```

**Esercizio 2.8.** Se nella figura il raggio dei circoli piccoli è 1, quanto vale l'area tratteggiata?

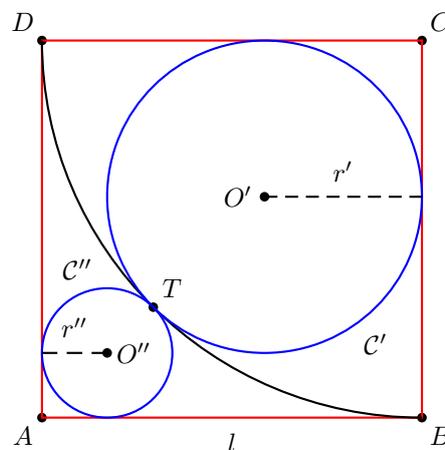
1.  $\frac{\pi}{4}$
2.  $\frac{\sqrt{3}}{2}$
3.  $\frac{5\pi}{6} - \sqrt{3}$
4.  $\frac{\pi}{3}$
5. Nessuna delle risposte precedenti.



Di seguito il codice completo della figura, che contiene solo istruzioni di **PSTricks**, senza alcun comando specifico di **pst-eucl**: in molti casi le macro previste nel codice standard del pacchetto **PSTricks** sono più che sufficienti per disegnare le figure richieste. Il problema è che, per ottenere la figura in questione senza usare **pst-eucl**, si è dovuti ricorrere al calcolo manuale delle posizioni dei punti. Un utile esercizio che propongo al lettore è di rifare la stessa costruzione, con **pst-eucl**, usando come punto di partenza solo il centro e il raggio del cerchio grande, e chiedendo al codice di determinare tutti gli altri punti richiesti.

```
\begin{pspicture}(-.5,-.5)(6.5,6.5)
\pscircle[linecolor=blue](3,3){3}
\pscircle[linecolor=red](3,3){1}
\pscircle[linecolor=red](1,3){1}
\pscircle[linecolor=red](5,3){1}
\pscircle[linecolor=red](2,1.27){1}
\pscircle[linecolor=red](4,1.27){1}
\pscircle[linecolor=red](2,4.73){1}
\pscircle[linecolor=red](4,4.73){1}
\pscustom[linecolor=green,fillstyle=vlines,hatchcolor=green]{%
  \psarcn(2,4.73){1}{120}{0}
  \psarcn(4,4.73){1}{180}{60}
  \psarc(3,3){3}{60}{120}}
\end{pspicture}
```

**Esercizio 2.9.** È dato un quadrato  $Q = (ABCD)$ , di lato  $l$ .



Tracciato l'arco  $\widehat{BTD}$  del cerchio  $C$  di centro  $C$  e raggio  $l$ , esprima in funzione di  $l$  le misure dei raggi

$$r' \text{ di } C' \text{ ed } r'' \text{ di } C''.$$

Costruisci i due cerchi  $C'$  e  $C''$  con riga e compasso.

Segue il codice completo della figura del problema.

```

\begin{pspicture}(-0.5,-0.5)(5.5,5.5)
\pstGeonode[PosAngle={-135,-45,45,135}](0,0){A}(5,0){B}(5,5){C}(0,5){D}
\pspolygon[linecolor=red](A)(B)(C)(D)
\pstArcOAB{C}{D}{B}
\pstInterLC[PointSymbolB=none,PointNameB=none,PosAngleA=45]{A}{C}{C}{D}{T}{T1}
\pstSymO[PosAngle=180]{T}{A}{O'}
\pstCircleOA[linecolor=blue]{O'}{T}
\pstRotation[RotAngle=90,PointName=none,PointSymbol=none]{T}{O'}{E}
\pstBisectBAC[PointName=none,PointSymbol=none,linestyle=none]{A}{E}{T}{L}
\pstInterLL{E}{L}{A}{C}{O''}
\pstCircleOA[linecolor=blue]{O''}{T}
\pstProjection[PointSymbol=none,PointName=none]{B}{C}{O'}{M}
\pstProjection[PointSymbol=none,PointName=none]{A}{D}{O''}{N}
\pcline[linestyle=none](A)(B)
\Bput{1}
\pcline[linestyle=dashed](O')(M)
\Aput{r}
\pcline[linestyle=dashed](O'')(N)
\Bput{r''}
\pstRotation[RotAngle=80,PointName=none,PointSymbol=none]{O'}{T}{T1}
\uput[dr](T1){\mathcal{C}' }
\pstRotation[RotAngle=50,PointName=none,PointSymbol=none]{O''}{T}{T2}
\uput[ul](T2){\mathcal{C}'' }
\end{pspicture}

```