Alcune nozioni di base e introduzione vettoriale alla retta

Lorenzo Roi

Presentazione

In questo notebook si intendono dapprima presentare alcune nozioni di base della geometria analitica piana per poi introdurre le principali rappresentazioni piane della retta. Le nozioni di geometria analitica che si affrontano vengono sviluppate a partire da quelle del calcolo vettoriale: pertanto per comprendere questa prima parte sono necessari i seguenti requisiti

- componenti cartesiane di un vettore,
- somma vettoriale,
- moltiplicazione scalare-vettore,
- collinearità tra vettori,
- prodotto scalare e vettoriale.

Nelle lezioni successive a questa, le nozioni necessarie si limitano invece alle usuali conoscenze relative alle rappresentazioni implicita ed esplicita della retta: pertanto i notebook Lezione2-RettaFasci.nb e Lezione3-EserciziRette.nb possono essere utilizzati a prescindere da questa introduzione. Il notebook Help.nb definisce dapprima la terminologia seguita negli help delle singole funzioni e quindi presenta in ordine alfabetico tutte le funzioni contenute nel pacchetto **`RettaFasci.m`** e usate nell'intero lavoro.

Istruzioni

Per l'utilizzo in *Mathematica*, modificare il percorso contenuto nell'istruzione **`SetDirectory`** sottostante sostituendolo con quello effettivo contenente il pacchetto **`RettaFasci.m`** e i file estratti dall'archivio zip (ponendo attenzione alle maiuscole-minuscole del percorso e all'unità). Avviare quindi il calcolo delle tre celle successive cliccando sul bottone che segue.

Cliccare qui per caricare il pacchetto

```
SetDirectory["C:/Documents and Settings/Lorenzo/Documenti/rette"];
<< RettaFasci.m
```

```
Off[General::spell];
Off[General::spell1];
Needs["Miscellaneous`RealOnly`"];
Needs["Graphics`Arrow`"];
```

Le seguenti funzioni sono utilizzate per costruire le figure presenti solo in questo notebook.

```
$ContextPath = Join[$ContextPath, {"lez1`"}];
lez1`vettore[componenti_, puntoAppl_] := Graphics[
  {colore[rosso], Thickness[0.004], Arrow[puntoAppl, puntoAppl + componenti, HeadWidth → 0.4]}];
lez1`vettore[componenti_] := Graphics[
  {colore[rosso], Thickness[0.004], Arrow[{0, 0}, componenti, HeadWidth → 0.4]}];
lez1`labelPunto[punto_, stringa_] := Graphics[Text[stringa, punto + {0, .15}]];
```

Coordinate del punto medio di un segmento

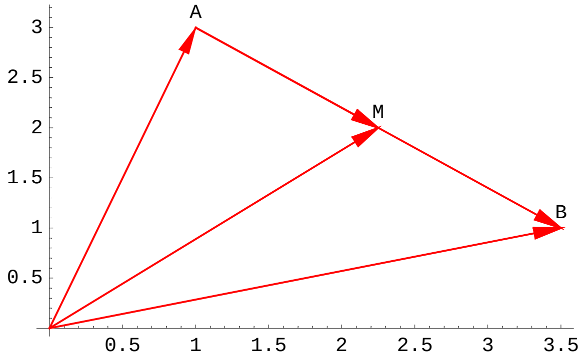
Intendiamo determinare le coordinate del punto medio di un segmento,

```
puntoM = {xM, yM};
```

assegnate le coordinate degli estremi *A* e *B*

```
puntoA = {xA, yA};      puntoB = {xB, yB};
```

In riferimento alla figura seguente, possiamo individuare i vettori (e relative componenti) sottraendo le coordinate dell'estremo con quelle del punto iniziale del vettore.




```
{vetOA = puntoA - {0, 0},
 vetOB = puntoB - {0, 0},
 vetAB = vetOB - vetOA}
```

```
{{xA, yA}, {xB, yB}, {-xA + xB, -yA + yB}}
```

Individuiamo invece il vettore incognito con le componenti

```
vetOM = puntoM - {0, 0}
```

```
{xM, yM}
```

Ricordando la definizione di somma vettoriale e di moltiplicazione di uno scalare con un vettore e, in riferimento alla figura sopra, il vettore *OM* si può esprimere come la somma del vettore OA con $\frac{1}{2}$ AB.

$$\text{vetOM} = \text{vetOA} + \frac{1}{2}\,\text{vetAB}$$

$$\left\{xA + \frac{1}{2}(-xA + xB),\ yA + \frac{1}{2}(-yA + yB)\right\}$$

Dopo una semplice riduzione di termini simili, si ottiene

```
vetOM = Simplify[vetOM = vetOA + 1/2 vetAB]
```

$$\left\{\frac{xA + xB}{2}, \frac{yA + yB}{2}\right\}$$

Questo vettore, avendo come origine l'origine del sistema, fornisce pure le coordinate del punto medio *M* che quindi sono

```
puntoM = vetOM
```

$$\left\{\frac{xA + xB}{2}, \frac{yA + yB}{2}\right\}$$

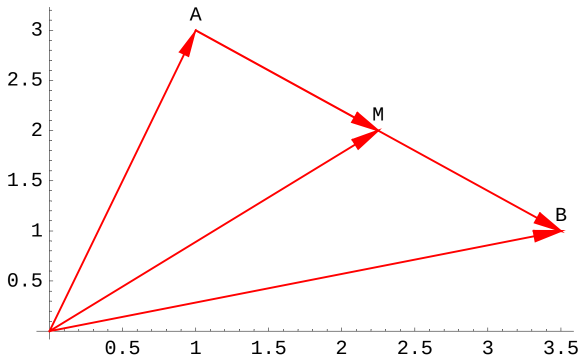
Istruzioni per la figura

Definite le coordinate dei punti *A* e *B*

```
{xA, yA} = {1, 3};
{xB, yB} = {3.5, 1};
```

sfruttiamo nelle istruzioni seguenti le funzioni definite nella parte di inizializzazione

```
Show[vettore /@ {puntoA, puntoB, vetOM}, vettore[vetAB, puntoA], vettore[1 / 2 vetAB, puntoA],
  labelPunto[puntoA, "A"], labelPunto[puntoB, "B"], labelPunto[puntoM, "M"], Axes → True];
```



Per utilizzare ancora le grandezze simboliche attivare l'istruzione seguente

```
Clear["Global`*"]
```

Distanza di due punti

```
Clear["Global`*"]
```

Dati due punti *A* e *B*,

```
puntoA = {xA, yA};
puntoB = {xB, yB};
```

la loro distanza si ottiene immediatamente determinando il modulo del vettore *AB* le cui componenti cartesiane si ottengono facilmente come

```
vetAB = puntoB - puntoA
```

```
{-xA + xB, -yA + yB}
```

Ricordando il significato di prodotto scalare di un vettore, date le sue componenti cartesiane, il modulo del vettore si ottiene estraendo la radice quadrata del prodotto scalare del vettore per sé stesso.

```
moduloVettore[vet_] := √vet.vet
```

Con i dati simbolici introdotti all'inizio abbiamo

```
moduloVettore[vetAB]
```

$$\sqrt{(-xA + xB)^2 + (-yA + yB)^2}$$

che coincide con quanto si ottiene applicando il teorema di Pitagora al triangolo rettangolo formato dalle componenti cartesiane del vettore *AB*.

Area del triangolo

```
Clear["Global`*"]
```

Lo scopo è di determinare l'area del triangolo di vertici *ABC* (figura 2). Siano quindi dati i tre vertici rappresentati dalle coppie

```
puntoA = {xA, yA};
puntoB = {xB, yB};
puntoC = {xC, yC};
```

Poiché il *modulo del prodotto vettoriale di due vettori è pari all'area del parallelogramma individuato dai vettori fattori*, l'area cercata sarà la metà di tale modulo. A tal fine definiamo i vettori *AB*, *AC*, le cui componenti si ottengono per differenza delle coordinate del punto terminale con quello iniziale (mentre la terza componente, essendo in un piano, è nulla).

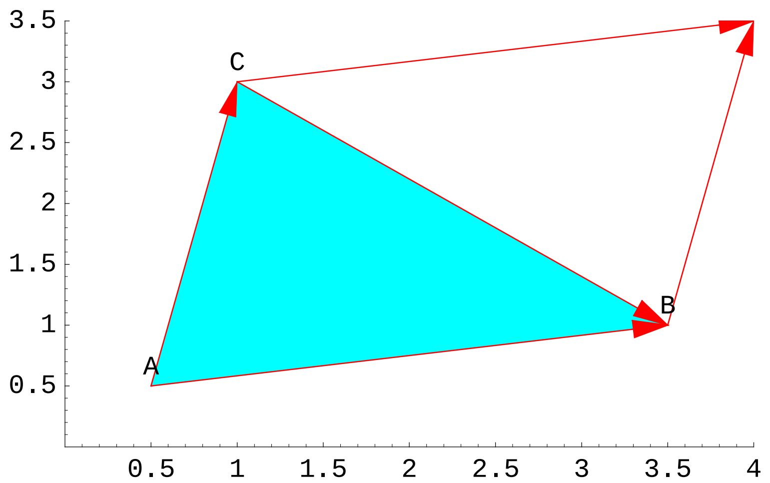


Figura 1

```
vetAB = puntoB - puntoA;
vetAC = puntoC - puntoA;
vetAB = Append[vetAB, 0]
vetAC = Append[vetAC, 0]
```

```
{-xA + xB, -yA + yB, 0}
```

```
{-xA + xC, -yA + yC, 0}
```

Disposte le componenti dei vettori appena definite in forma matriciale assieme ai versori del sistema

```
{{i, j, k}, vetAB, vetAC} // MatrixForm
```

$$\begin{pmatrix} i & j & k \\ -xA + xB & -yA + yB & 0 \\ -xA + xC & -yA + yC & 0 \end{pmatrix}$$

determiniamo il vettore prodotto calcolando il determinante di tale disposizione: otteniamo il vettore che, per meglio evidenziarne le componenti, rappresentiamo come una matrice colonna

```
vetAB × vetAC // MatrixForm
```

$$\begin{pmatrix} 0 \\ 0 \\ -xB\,yA + xC\,yA + xA\,yB - xC\,yB - xA\,yC + xB\,yC \end{pmatrix}$$

Come aspettato, l'unica componente non nulla è la componente z

```
componenteZ = (vetAB × vetAC)[[3]]
```

```
-xB yA + xC yA + xA yB - xC yB - xA yC + xB yC
```

per cui, considerando il suo valore assoluto e dividendo tale valore per 2

```
areaABC = 1/2 Abs[Collect[componenteZ, {xA, xB, xC}]]
```

$$\frac{1}{2}\,\mathrm{Abs}[xC\,(yA - yB) + xA\,(yB - yC) + xB\,(-yA + yC)]$$

otteniamo l'espressione cercata per l'area del triangolo *ABC*.

Onde poter più facilmente memorizzare tale espressione costruiamo direttamente la matrice 3x3 con, nelle prime due colonne, le coordinate dei vertici del triangolo e l'ultima che invece presenta valori unitari.

$$\text{matrice} = \begin{pmatrix} xA & yA & 1 \\ xB & yB & 1 \\ xC & yC & 1 \end{pmatrix};$$

Se calcoliamo, con le solite regole, il suo determinante e ne prendiamo il valore assoluto diviso per 2,

```
formulaMnemonica = 1/2 Abs[Simplify[Det[matrice]]]
```

$$\frac{1}{2}\text{Abs}[xC\,(yA - yB) + xA\,(yB - yC) + xB\,(-yA + yC)]$$

quanto otteniamo appare uguale all'area appena trovata.

```
areaABC == formulaMnemonica
```

```
True
```

Potremo quindi esprimere l'area di un triangolo per mezzo dell'espressione

$$\frac{1}{2}\text{Abs}\left[\text{Det}\left[\begin{pmatrix} xA & yA & 1 \\ xB & yB & 1 \\ xC & yC & 1 \end{pmatrix}\right]\right]$$

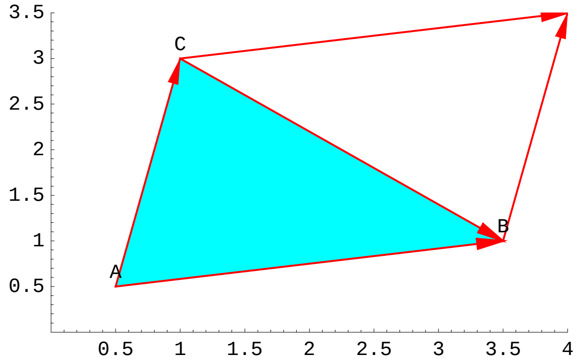
$$\frac{1}{2}\text{Abs}[-xB\,yA + xC\,yA + xA\,yB - xC\,yB - xA\,yC + xB\,yC]$$

Istruzioni per la figura

```
xC = 1; yC = 3; xB = 3.5; yB = 1; xA = 0.5; yA = 0.5;
```

```
vetAB = Drop[vetAB, -1]; vetAC = Drop[vetAC, -1];
```

```
Show[Graphics[{colore[cyan], Polygon[{puntoA, puntoB, puntoC}]}], vettore[vetAB, puntoA],
  vettore[vetAC, puntoA], vettore[ vetAB - vetAC, puntoC], vettore[vetAC, puntoB],
  vettore[vetAB, puntoC], labelPunto[puntoA, "A"], labelPunto[puntoB, "B"],
  labelPunto[puntoC, "C"], Axes → True, PlotRange -> {{0, 4}, {0, 3.5}}];
```




```
Clear["Global`*"]
```

Rappresentazione parametrica della retta

```
Clear["Global`*"]
```

L'idea per ottenere una rappresentazione algebrica della retta parte dalla **nozione di collinearità** di due vettori. Due vettori sono collineari se e solo se l'uno si può esprimere come la moltiplicazione di un opportuno scalare per l'altro vettore.

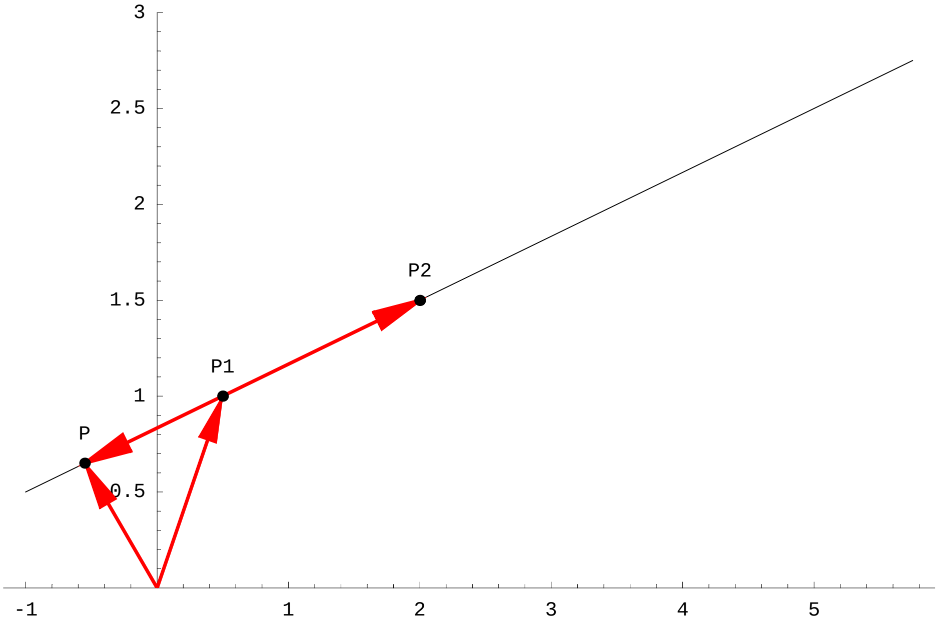
Siano quindi $P1$ e $P2$ due punti del piano che considereremo assegnati

```
puntoP1 = {x1, y1}; puntoP2 = {x2, y2};
```

mentre P verrà inteso come un punto qualsiasi della retta r (figura sottostante).

```
puntoP = {x, y};
```

Vogliamo determinare innanzitutto la dipendenza di x e y da qualche parametro reale e quindi esplicitare il legame esistente tra queste due variabili (rispettivamente ascissa e ordinata del punto P).



In riferimento alla figura, possiamo definire i vettori ***OP*** e ***OP1***, il primo variabile mentre il secondo è fisso

```
vetOP = puntoP - {0, 0};
vetOP1 = puntoP1 - {0, 0};
```

Analogamente i punti dati $P1$ e $P2$ determinano un vettore fisso ***P1P2***

```
vetP1P2 = puntoP2 - puntoP1
```

```
{-x1 + x2, -y1 + y2}
```

Possiamo infine introdurre un quarto vettore ***P1P*** dato dalla differenza ***P1P*** = ***OP*** – ***OP1*** e le cui componenti sono

```
vetP1P = puntoP - puntoP1
```

```
{x - x1, y - y1}
```

La condizione di collinearità tra i vettori ***P1P*** e ***P1P2*** si scrive come ***P1P*** == t * ***P1P2***

```
collinearita = vetP1P == t * vetP1P2
```

```
{x - x1, y - y1} == {t (-x1 + x2), t (-y1 + y2)}
```

oppure esplicitando ***OP*** in quanto è pure ***P1P*** = ***OP*** – ***OP1***,

```
vetOP == vetOP1 + t * vetP1P2
```

```
{x, y} == {x1 + t (-x1 + x2), y1 + t (-y1 + y2)}
```

Il punto P avrà le coordinate

```
puntoP = Collect[puntoP /. Flatten[Solve[vetOP == vetOP1 + t vetP1P2, {x, y}]], t]
```

```
{x1 + t (-x1 + x2), y1 + t (-y1 + y2)}
```

Abbiamo in tal modo ottenuto le coordinate del punto generico $P \in r$ come funzioni del parametro t: queste due funzioni (lineari) di t costituiscono assieme la rappresentazione parametrica della retta passante per i due punti dati.

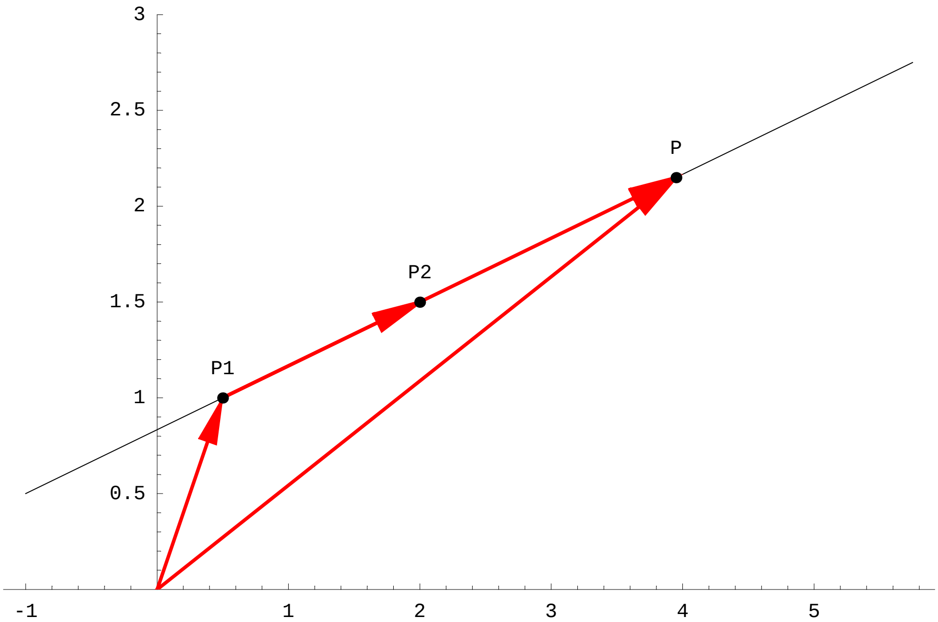
$$\begin{cases} x = x1 + t(-x1 + x2) \\ y = y1 + t(-y1 + y2) \end{cases}$$ rappresentazione parametrica della retta per due punti

Istruzioni per la figura

```
x1 = 0.5; y1 = 1; x2 = 2; y2 = 1.5;
```

```
r1 = ParametricPlot[puntoP, {t, -1, 3.5}, PlotRange → {0, 3}, DisplayFunction → Identity];
```

```
Table[Show[r1, vettore /@ {puntoP, vetOP1},
   vettore[vetP1P2, puntoP1], vettore[puntoP - puntoP1, puntoP1],
   Apply[labelPunto, {{puntoP1, "P1"}, {puntoP2, "P2"}, {puntoP, "P"}}, {1}],
   Graphics[{PointSize[0.012], Point /@ {puntoP, puntoP1, puntoP2}}], Axes → True,
   DisplayFunction → $DisplayFunction, ImageSize → 500], {t, -.7, 3, 0.1}];
```




```
Clear["x1", "y1", "x2", "y2", "r1"]
```

Rappresentazioni implicita ed esplicita

Partendo dal sistema di due equazioni ottenuto uguagliando le componenti dei vettori a primo e secondo membro (riportato appena sopra)

```
{x - x1, y - y1} == {t (-x1 + x2), t (-y1 + y2)}
```

e scritto, per comodità di lettura nella forma,

```
{eq1 = collinearita[[1, 1]] == collinearita[[2, 1]],
  eq2 = collinearita[[1, 2]] == collinearita[[2, 2]]} // TableForm
```

```
x - x1 == t (-x1 + x2)
y - y1 == t (-y1 + y2)
```

si può ricavare da ciascuna espressione il parametro t ottenendo, nel primo caso,

```
sol1 = Flatten[Solve[eq1, t]]
```

$$\left\{t \to \frac{-x + x1}{x1 - x2}\right\}$$

e nel secondo

```
sol2 = Flatten[Solve[eq2, t]]
```

$$\left\{t \to \frac{-y + y1}{y1 - y2}\right\}$$

Uguagliando per la proprietà transitiva le due espressioni trovate

```
eqRetta2Punti = (t /. sol1) == (t /. sol2)
```

$$\frac{-x + x1}{x1 - x2} == \frac{-y + y1}{y1 - y2}$$

e portando a primo membro il secondo, otteniamo

$$\text{eqRetta2Punti} = \left(\# - \frac{-y + y1}{y1 - y2}\right) \& /@ \text{eqRetta2Punti}$$

$$\frac{-x + x1}{x1 - x2} - \frac{-y + y1}{y1 - y2} == 0$$

equazione che costituisce l'**equazione rappresentativa di una retta passante per due punti** (evidentemente con diverse ascisse e ordinate in quanto $x1 \neq x2 \wedge y1 \neq y2$). Ora, riportando tutto ad un unico denominatore,

```
eq = Together[eqRetta2Punti[[1]]] == 0
```

$$\frac{x1\, y - x2\, y - x\, y1 + x2\, y1 + x\, y2 - x1\, y2}{(x1 - x2)\,(y1 - y2)} == 0$$

ne segue che dev'essere nullo il numeratore

```
eqNumeratore = Thread[eq (x1 - x2) (y1 - y2), Equal]
```

$x1\, y - x2\, y - x\, y1 + x2\, y1 + x\, y2 - x1\, y2 == 0$

Fattorizzando le variabili x e y a primo membro, quest'ultima assume la forma

```
equazioneImplicita[eqNumeratore]
```

$(x1 - x2)\, y + x2\, y1 - x1\, y2 + x\, (-y1 + y2) == 0$

Posto quindi

```
a = -y1 + y2; b = x1 - x2; c = x2 y1 - x1 y2;
```

si ottiene infine l'equazione che lega le variabili x e y espressa nella sua forma più semplice

$a\,x + b\,y + c = 0$	rappresentazione implicita della retta

Questa costituisce la **rappresentazione implicita** della retta r: le coordinate del punto P appaiono in questo contesto come le soluzioni di un'equazione lineare di primo grado mentre, per quanto detto inizialmente, i punti corrispondenti sono quelli della retta r.

Per ottenere una terza rappresentazione, esplicitiamo a primo membro la variabile dipendente y dividendo per $x1 - x2 \neq 0$ (ciò esclude le rette parallele all'asse y in quanto i punti P1 e P2 non potranno avere la medesima ascissa)

```
equazioneEsplicita[eqNumeratore]
```

$$y == \frac{x\,(y1 - y2)}{x1 - x2} + \frac{-x2\, y1 + x1\, y2}{x1 - x2}$$

Posto quindi

$$m = \frac{y2 - y1}{x2 - x1};\ q = \frac{-x2\, y1 + x1\, y2}{x1 - x2};$$

o anche (a partire dalla forma implicita con $b \neq 0$)

$$m = -\frac{a}{b};\ q = -\frac{c}{b}$$

si ha in definitiva

$y = m\,x + q$	rappresentazione esplicita della retta

che costituisce la **rappresentazione esplicita** di r.

Un'ultima, importante considerazione: quanto svolto procede da un ente geometrico, la retta r, e deduce una forma algebrica che la rappresenta in un piano cartesiano. Ci si può quindi chiedere se vale pure il viceversa ossia, se a partire da sistemi di equazioni del tipo

$$\begin{cases} x = \alpha t + \beta \\ y = \gamma t + \delta \end{cases}$$

per la forma parametrica, oppure equazioni di primo grado della forma

$$a x + b y + c = 0$$

per quella implicita ed, infine per quella esplicita,

$$y = m x + q$$

si possa associare a ciascuna equazione (o coppia, per la forma parametrica) e al variare nel campo dei numeri reali dei rispettivi coefficienti, una retta del piano. La risposta è affermativa con un unico limite per la forma esplicita: questa, per come è stata dedotta, non può rappresentare rette della forma implicita $x = k$ (con k costante reale) ossia non può esprimere rette parallele all'asse delle ordinate.

```
Clear["Global`*"]
```

Grafici vari di rette

Rette in forma parametrica

```
Clear["Global`*"]
```

Presentiamo in questa e nelle successive due sezioni alcuni grafici esemplificativi di rette nel piano.

Equazione di una retta in forma parametrica passante per due punti. Definiti i punti

```
puntoA = {1, 2}; puntoB = {3, 5}; puntoC = {-3, 2};
```

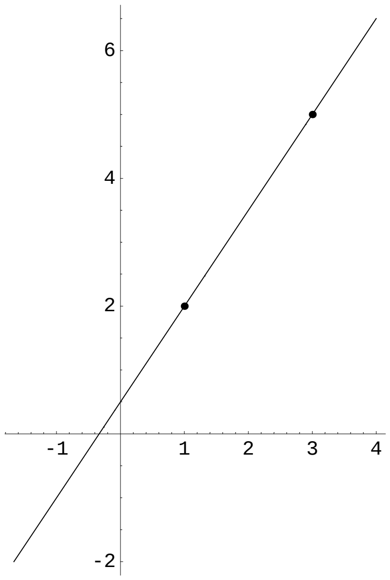
la retta per A e B è descritta dalle equazioni parametriche

```
rettaAB = retta2PuntiParametrica[puntoA, puntoB, t]
```

```
{x == 1 + 2 t, y == 2 + 3 t}
```

e ha il grafico (notare le parentesi graffe attorno a **rettaAB**)

```
Show[Graphics[{PointSize[0.02], Point[puntoA], Point[puntoB]}],
  graficoRette[{rettaAB}, t, {x, -2, 4}, {y, -2, 8}, DisplayFunction → Identity],
  DisplayFunction -> $DisplayFunction, Axes → True, AspectRatio → Automatic];
```

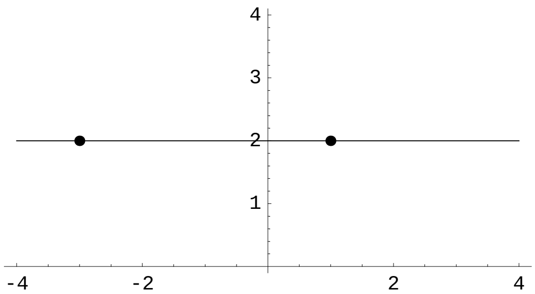


Retta per *A* e *C*

```
rettaAC = retta2PuntiParametrica[puntoA, puntoC, t]
```

```
{x == 1 - 4 t, y == 2}
```

```
Show[Graphics[{PointSize[0.02], Point[puntoA], Point[puntoC]}],
  graficoRette[{rettaAC}, t, {x, -4, 4}, {y, 0, 4}, DisplayFunction → Identity],
  DisplayFunction -> $DisplayFunction, Axes → True, AspectRatio → Automatic];
```

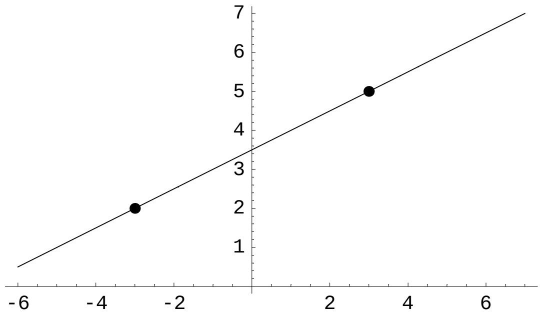


Retta per *B* e *C*

```
rettaBC = retta2PuntiParametrica[puntoB, puntoC, t]
```

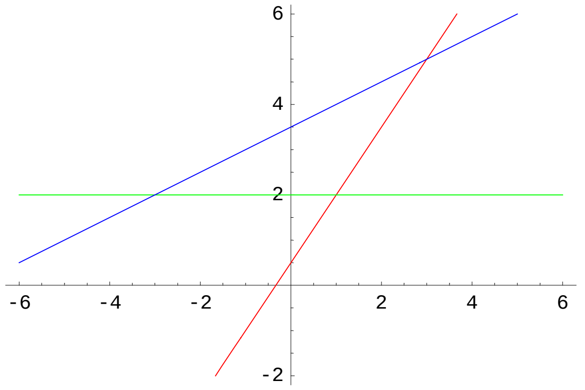
```
{x == 3 - 6 t, y == 5 - 3 t}
```

```
Show[Graphics[{PointSize[0.02], Point[puntoB], Point[puntoC]}],
  graficoRette[{rettaBC}, t, {x, -6, 7}, {y, -1, 8}, DisplayFunction → Identity],
  DisplayFunction -> $DisplayFunction, Axes → True, AspectRatio → Automatic];
```



Il grafico delle tre rette nello stesso piano cartesiano

```
graficoRette[{rettaAB, rettaAC, rettaBC}, t,
  {x, -6, 6}, {y, -2, 6}, PlotStyle → colore /@ {rosso, verde, blu}];
```

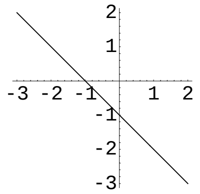


Retta parametrica generica

```
rettaParamQualsiasi = {x == t - 2, y == - t + 1}
```

```
{x == -2 + t, y == 1 - t}
```

```
graficoRette[{rettaParamQualsiasi}, t, {x, -3, 3}, {y, -3, 3}];
```




```
Clear["Global`*"]
```

■ Rette in forma implicita

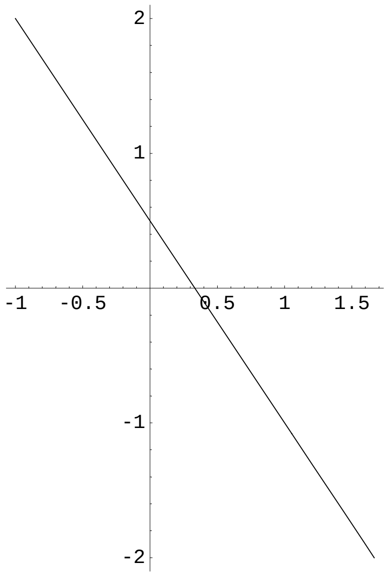
```
Clear["Global`*"]
```

Retta di equazione $3x + 2y - 1 = 0$ e suo grafico

```
r1 = 3 x + 2 y - 1 == 0
```

```
-1 + 3 x + 2 y == 0
```

```
graficoRette[{r1}, {x, -2, 2}, {y, -2, 2}];
```

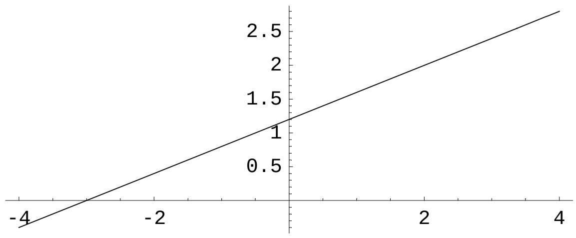


Retta di equazione $-2\,x + 5\,y - 6 = 0$ e suo grafico

```
r2 = -2 x + 5 y - 6 == 0
```

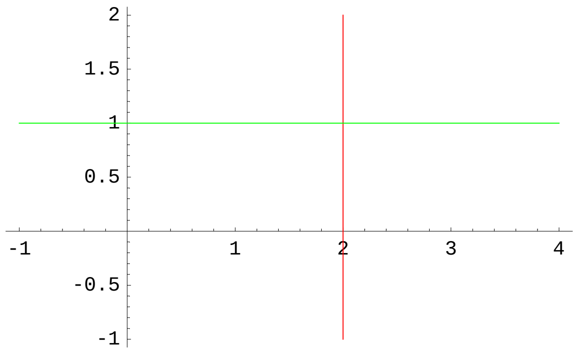
```
-6 - 2 x + 5 y == 0
```

```
graficoRette[{r2}, {x, -4, 4}, {y, -4, 4}];
```



Rette di equazione $x = 2$ e $y = 1$ e rispettivi grafici

```
graficoRette[{x == 2, y == 1}, {x, -1, 4}, {y, -1, 2}, PlotStyle → colore /@ {rosso, verde}];
```



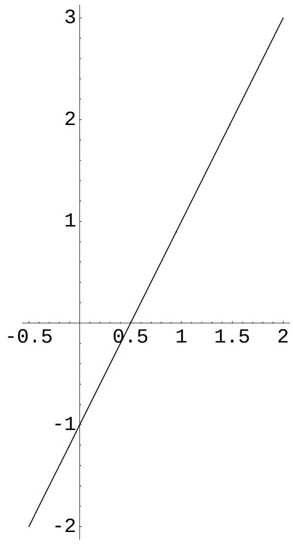
▪ Rette in forma esplicita

```
Clear["Global`*"]
```

Retta di equazione $y = 2x - 1$ e grafico

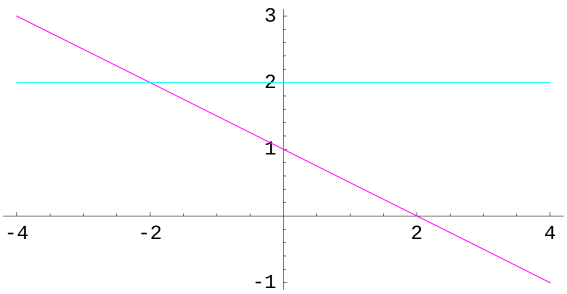
```
r3 = y == 2 x - 1;
```

```
graficoRette[{r3}, {x, -2, 2}, {y, -2, 3}];
```



Rette di equazione $y = -\frac{1}{2}x + 2$ e $y = 2$ e rispettivi grafici

```
graficoRette[{y == -1/2 x + 1, y == 2}, {x, -4, 4}, {y, -4, 4}, PlotStyle → colore /@ {magenta, cyan}];
```




```
Clear["Global`*"]
```