



TRIBELON

RIVISTA DI DISEGNO
UNIVERSITÀ DEGLI
STUDI DI FIRENZE

VOL. 2 | N. 3 | 2025
MODELLI, FORME E GEOMETRIE
MODELS, SHAPES AND GEOMETRIES

Citation: G. Anzani, *Come tenersi a debita distanza da Punti, Rette e Piani*, in *Codici grafici*, TRIBELON, II, 2024, 3, pp. 111-117.

ISSN (stampa): 3035-143X

ISSN (online): 3035-1421

doi: <https://doi.org/10.36253/tribelon-3519>

Published: June, 2025

Copyright: 2025 Anzani G., this is an open access article published by Firenze University Press (<http://www.riviste.fupress.net/index.php/tribelon>) and distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Competing Interests: The Author(s) declare(s) no conflict of interest.

Journal Website: riviste.fupress.net/tribelon

CODICI GRAFICI

COME TENERSI A DEBITA DISTANZA DA PUNTI, RETTE E PIANI

GIOVANNI ANZANI

University of Florence
giovanni.anzani@unifi.it

Prendendo spunto dal tema del terzo numero della rivista, si propone una rivisitazione in chiave algoritmica delle nozioni classiche dei fondamenti della geometria descrittiva: gli enti geometrici fondamentali (punto, retta e piano), le condizioni fondamentali (appartenenza, parallelismo e perpendicolarità) e ausiliarie (allineamento e complanarità), le operazioni grafiche (disegno e verifica).

Tali nozioni hanno trovato, ormai da tempo, modo di transitare dall'analogico cartaceo (2D) al digitale CAD (2D/3D) guadagnando una terza dimensione nello spazio vettoriale; se ne propone qua un'ulteriore transizione con una serie di algoritmi che possono automatizzare e ampliare le potenzialità applicative di tali fondamenti.

Il testo descrive gli algoritmi AutoLISP atti a ridefinire opportunamente in AutoCAD i fondamenti della geometria descrittiva citati.

Saranno illustrati i predicati in grado di svolgere le operazioni grafiche di verifica delle condizioni fondamentali e ausiliarie e le procedure in grado di svolgere le operazioni grafiche di disegno che consentono la determinazione univoca di enti geometrici fondamentali, poi disegnabili in AutoCAD, applicando in maniera combinata le condizioni fondamentali. Farà seguito un'applicazione della geometria descrittiva specificatamente dedicata al tema delle equidistanze tra enti geometrici.

L'insieme di questi argomenti è frutto di una ricerca pluriennale e che ha portato ad un listato di codice con oltre 400 definizioni di funzione; per agevolarne la lettura, si è cercato di nominare sempre le variabili contenenti punti rette e piani come (Pu, Pu1, Pi2, ...) (Re, Re1, Re2, ...) (Pi, Pi1, Pi2, ...) e le condizioni come (App Par Per All Com).

¹ L'origine del sistema di coordinate è un punto definito da una lista contenente una terna di valori nulli; possiamo definire una variabile globale a cui le procedure sviluppate potranno riferirsi quale punto in cui è collocata l'origine del sistema di coordinate (WCS o UCS):

```
◦ (setq *Pu000* (list 0.0 0.0 0.0))
```

² Definizione di un punto come lista contenente una terna di valori:

```
◦ (setq Pu1 (list Pu1_x Pu1_y Pu1_z)),
```

³ Conversione di coordinate di un punto da UCS (1) a WCS (0) e viceversa:

```
◦ (setq Pu1_w (trans Pu1_u 1 0))
```

```
◦ (setq Pu1_u (trans Pu1_w 0 1))
```

È possibile automatizzare queste operazioni definendo delle procedure dedicate:

```
◦ (defun Pu_trans (Pu1 fr to) (trans Pu1 fr to))
```

```
◦ (defun Pu_trans_01 (Pu1) (Pu_trans Pu1 0 1))
```

```
◦ (defun Pu_trans_10 (Pu1) (Pu_trans Pu1 1 0))
```

⁴ Definizione di una retta come lista contenente una terna di liste:

```
◦ (defun trasfo_2Pu_Re (Pu1 Pu2 / vers orig orig)
```

```
  (setq vers (vers_2Pu Pu1 Pu2)
```

```
        orig (flat_z_Pu (trans Pu1 0 vers))
```

```
        orig (trans orig vers 0))
```

```
  (list vers orig orig))
```

Tale procedura e altre a seguire si avvalgono di alcune procedure dedicate alla gestione di vettori e versori, al cambiamento delle coordinate di un punto, all'estrazione di elementi da una lista:

La definizione di Punti, Rette e Piani nello spazio vettoriale

In AutoCAD gli assi del sistema di coordinate cartesiane possono essere fissi WCS (World Coordinate System) o mobili UCS (User Coordinate System) definiti di volta in volta dall'utente rispetto a quelli WCS ridefinendone l'origine¹ e l'orientamento (degli assi x y e con essi dell'asse z). Per definire un punto in AutoCAD², sarà possibile attenersi alla modalità in uso per le coordinate cartesiane nello spazio euclideo tridimensionale restituendola in forma di lista contenente una terna ordinata di numeri reali; tale lista corrisponderà sempre allo stesso punto fisso nello spazio se in AutoCAD sarà attivo il WCS, mentre potrà corrispondere ad un qualsiasi altro punto nello spazio se in AutoCAD sarà attivo un UCS utente e sarà un punto mobile nello spazio al variare dell'UCS attivo.

Quando in AutoCAD è attivo uno specifico UCS è possibile effettuare la conversione di coordinate di un punto da mobili UCS a fisse WCS o viceversa³.

Per definire una retta in AutoCAD⁴, di cui ci interessa solo la direzione e non il verso, sarà possibile riferirsi alle equazioni parametriche della retta che, a partire da un qualsiasi punto appartenente alla retta e da un vettore non nullo parallelo alla retta, si esprimono in un sistema lineare. Per una serie di vantaggi d'uso il sistema lineare non verrà determinato, il punto appartenente alla retta sarà sempre quello risultante dalla proiezione ortogonale dell'origine del sistema di Coordinate globali WCS sulla retta (ed indicato come origine della retta), il vettore non nullo parallelo alla retta sarà sostituito dal versore parallelo alla retta orientato indifferentemente in uno dei due versi della direzione della retta.

- (defun modu_2Pu (Pu1 Pu2)
 - (distance Pu1 Pu2))
 - (defun vett_2Pu (Pu1 Pu2) (mapcar '- Pu2 Pu1))
 - (defun vers_2Pu (Pu1 Pu2)
 - (mapcar '(lambda (x) (/ x (modu_2Pu Pu1 Pu2)))
 - (vett_2Pu Pu1 Pu2)))
 - (defun flat_z_Pu (Pu1) (chan_z_Pu Pu1 0.0))
 - (defun chan_z_Pu (Pu1 z)
 - (list (1° Pu1) (2° Pu1) z))
 - (defun 1° (Li1) (nth 0 Li1))
 - (defun 2° (Li1) (nth 1 Li1))
 - (defun 3° (Li1) (nth 2 Li1))
- 5 **Predicato dedicato a verificare se il sistema di coordinate attivo è quello globale WCS:**
 - (defun P_WCS ()
 - (and (equal (getvar "ucsorg") '(0.0 0.0 0.0))
 - (equal (getvar "ucsdirdir") '(1.0 0.0 0.0))
 - (equal (getvar "ucsydir") '(0.0 1.0 0.0))))
- 6 **Definizione di un Pi** come lista contenente una terna di liste:
 - (defun trasfo_3Pu_Pi (Pu1 Pu2 Pu3 / vers orig orig)
 - (setq vers (vers_3Pu Pu1 Pu2 Pu3))
 - origt (flat_xy_Pu (trans Pu1 0 vers))
 - orig (trans origt vers 0))
 - (list vers orig origt))
 - Tale procedura e/o altre a seguire si avvalgono di alcune procedure per la gestione di vettori e versori e per la modifica delle coordinate di un punto:
 - (defun vers_3Pu (Pu1 Pu2 Pu3 / vett_Pi1)
 - (setq vett_Pi (vett_3Pu Pu1 Pu2 Pu3))
 - (mapcar '(lambda (x) (/ x (modu_2Pu '(0.0 0.0 0.0) vett_Pi))) vett_Pi))
 - (defun vett_3Pu (Pu1 Pu2 Pu3 / x1 x2 x3 y1 y2 y3 z1 z2 z3)
 - (mapcar 'set '(x1 y1 z1 x2 y2 z2 x3 y3 z3)
 - (append Pu1 Pu2 Pu3))
 - (list (- (* (- y2 y1)(- z3 z1)) (* (- y3 y1)(- z2 z1)))
 - (- (* (- x3 x1)(- z2 z1)) (* (- x2 x1)(- z3 z1)))
 - (- (* (- x2 x1)(- y3 y1)) (* (- x3 x1)(- y2 y1))))
 - (defun flat_xy_Pu (Pu1)
 - (chan_xy_Pu1 Pu1 0.0 0.0))
 - (defun chan_xy_Pu (Pu1 x y) (list x y (3° Pu1)))
 - (defun comb_xy_Pu1_z_Pu2 (Pu1 Pu2)
 - (chan_z_Pu Pu1 (3° Pu2)))
 - 7 **Procedure di selezione.** Procedura base per la selezione di un punto nel disegno corrente in coordinate UCS poi convertite in coordinate WCS, entrambe le coordinate vengono scritte al command e la procedura restituisce in uscita il punto in coordinate WCS:
 - (defun get_WCS_point (Pu1 txt / Pu2t Pu2)
 - (initget 9)
 - (setq Pu2t (if Pu1 (getpoint Pu1 txt) (getpoint txt)))
 - (setq Pu2 (trans Pu2t 1 0))
 - (princ "coordinate: UCS ") (princ Pu2t)
 - (princ " WCS ") (princ Pu2) (princ "\n") Pu2)
 - Procedure specifiche per la selezione di punti rette e piani nel disegno corrente:
 - (defun Get_Pu () (get_WCS_point nil "\nPu:\n1° punto: "))
 - (defun Get_Re (/ Pu1 Pu2)
 - (setq Pu1 *Pu000* Pu2 *Pu000*) (while (P_coinc_Pu_Pu Pu1 Pu2) (setq Pu1 (get_WCS_point nil "\nRe:\n1° punto: ")) (setq Pu2 (get_WCS_point (trans Pu1 0 1) "\n2° punto: ")))
 - (princ (if (P_coinc_Pu_Pu Pu1 Pu2) "\nSono stati forniti due punti coincidenti riprovare\n" "\n"))(trasfo_2Pu_Re Pu1 Pu2))
 - (defun Get_Pi (/ Pu1 Pu2 Pu3 vers_Pi1 orig_Pi1)
 - (setq Pu1 *Pu000* Pu2 *Pu000* Pu3 *Pu000*)
 - (while (or (P_allin_Pu_Pu_Pu Pu1 Pu2 Pu3) (P_coinc_Pu_Pu Pu1 Pu2) (P_coinc_Pu_Pu Pu1 Pu3) (P_coinc_Pu_Pu Pu2 Pu3)) (setq Pu1 (get_WCS_point nil "\nP_i:\n1° punto: ")) (setq Pu2 (get_WCS_point (trans Pu1 0 1) "\n2° punto: ")) (setq Pu3 (get_WCS_point (trans Pu2 0 1) "\n3° punto: ")))

Con tali specifiche due rette coincidenti di verso opposto avranno stessa origine e versore di segno opposto; rette passanti per l'origine del sistema di coordinate globali WCS avranno tale punto come origine. Sapendo che per due punti non coincidenti passa una sola retta sarà possibile definire, come vedremo più avanti nei predicati di verifica, un predicato finalizzato ad escludere l'eventuale coincidenza dei due punti (**Pu App Pu**) e disponendone, sarà possibile definire una funzione che, dati due punti non coincidenti determini la retta passante per essi (**Re App 2Pu**) restituendola in forma di lista contenente il versore della retta, le coordinate in forma WCS ed UCS orientato dell'origine della retta. In AutoCAD esistono tre variabili di sistema necessarie all'identificazione dell'UCS corrente: "ucsorg" che contiene le coordinate WCS del punto di origine del sistema di coordinate corrente; "ucsdirdir" e "ucsydir" che contengono i versori rispettivamente della direzione x e y del sistema di coordinate corrente. Da questi dati sarà possibile determinare il versore della direzione z che seguirà la nota regola della mano destra. Tenendo conto di ciò è possibile definire uno specifico predicato che possa verificare se in AutoCAD è attivo un UCS o se è attivo il WCS⁵. Definire in AutoCAD un UCS equivale a definire un piano orientato di cui si conosce non solo la giacitura, ma anche la direzione su esso di una coppia di rette ortogonali tra loro uscenti dall'origine degli assi.

Per definire un **piano in AutoCAD**⁶, di cui ci interessa la giacitura ma non l'orientamento, sarà possibile riferirsi alle equazioni parametriche del piano che, a partire da un qualsiasi punto appartenente al piano e da due vettori non nulli paralleli al piano, si esprimono in un sistema lineare. Per una serie di vantaggi d'uso il sistema lineare non verrà determinato, il punto appartenente al piano sarà sempre quello risultante dalla proiezione ortogonale dell'origine del sistema di Coordinate globali WCS sul piano (ed indicato come origine del piano), la coppia di vettori non nulli paralleli al piano saranno sostituiti dal versore ortogonale al piano orientato indifferentemente in uno dei due versi della direzione ortogonale al piano. Con tali specifiche due piani coincidenti di orientamento opposto avranno stessa origine e versore di segno opposto; piani passanti per l'origine del sistema di coordinate globali

WCS avranno tale punto come origine. Sapendo che per tre punti non allineati passa un solo piano sarà possibile definire, come vedremo più avanti nei predicati di verifica, un predicato finalizzato ad escludere l'eventuale allineamento dei tre punti (**Pu All 2Pu**) e disponendone, sarà possibile definire una funzione che, dati tre punti non coincidenti a due a due e non allineati determini il piano passante per essi (**Pi App 3Pu**) restituendolo in forma di lista contenente il versore ortogonale al piano, le coordinate in forma WCS ed UCS orientato dell'origine del piano.

Dai punti AutoCAD agli enti geometrici fondamentali e viceversa

Al fine di selezionare nel disegno punti rette e piani si è reso necessario scrivere specifiche procedure che agevolassero tali operazioni.

Una prima procedura base di selezione, si occupa di acquisire le coordinate tridimensionali di un punto nel disegno corrente che potrà poi essere utilizzato per definire: un **ente punto** (come unico punto necessario a definirlo), un **ente retta** (come uno dei due punti necessari a definirla), un **ente piano** (come uno dei tre punti necessari a definirlo).

Altre procedure specifiche di selezione, una per ciascun ente geometrico fondamentale: punto retta piano, si occupano, grazie alla procedura base di selezione, della raccolta dei punti necessari e, verificate eventuali coincidenze o allineamenti tra loro, della determinazione e scrittura, nel rispetto delle specifiche descritte in precedenza, di punti, rette e piani⁷.

Per alcune procedure che seguono, risulta necessario definire alcune procedure di trasformazione base⁸ in grado di effettuare una conversione di rette e piani in liste di punti: data una retta si determina una coppia di punti (2Pu App Re) posti ad una distanza assegnata dall'origine della retta (come definita in precedenza); dato un piano si determina una terna di punti (3Pu App Pi), il primo coincidente con l'origine del piano (come definita in precedenza) e gli altri due posti ad una distanza assegnata dall'origine del piano secondo due direzioni ortogonali tra loro al fine da risultare tre punti del piano non allineati.

```
(if (P_allin_Pu_Pu_Pu Pu1 Pu2 Pu3)
(princ "\nSono stati forniti tre punti allineati ri-
provare\n" ))
(if (P_coinc_Pu_Pu Pu1 Pu2)
(princ "\nSono stati forniti due punti coinci-
denti (1 e 2) riprovare\n" ))
(if (P_coinc_Pu_Pu Pu1 Pu3)
(princ "\nSono stati forniti due punti coinci-
denti (1 e 3) riprovare\n" ))
(if (P_coinc_Pu_Pu Pu2 Pu3)
(princ "\nSono stati forniti due punti coinci-
denti (2 e 3) riprovare\n" ))
(princ "\n")
(trasfo_3Pu_Pi Pu1 Pu2 Pu3))
```

8 Procedure di trasformazione base per la conversione di rette e piani in liste di punti:

```
(defun trasfo_Re_2Pu_dis (Re1 dis)
(list
(trans (chan_z_Pu (2° Re1) (+ dis)) (1° Re1) 0)
(trans (chan_z_Pu (2° Re1) (- dis)) (1° Re1) 0)))
(defun trasfo_Pi_3Pu_dis (Pi1 dis)
(list(3° Pi1)
(trans (chan_xy_Pu (2° Pi1) dis 0.0) (1° Pi1) 0)
(trans (chan_xy_Pu (2° Pi1) 0.0 dis) (1° Pi1) 0)))
```

9 Procedure di trasformazione specifiche per la conversione di rette e piani in liste di punti posti a distanze preassegnate.

Data una retta si determina una coppia di punti (2Pu App Re) posti ad alcune distanze preassegnate.

```
(defun trasfo_Re_2Pu_gra (Re1)
(trasfo_Re_2Pu_dis Re1 *+gra*))
(defun trasfo_Re_2Pu_gig (Re1)
(trasfo_Re_2Pu_dis Re1 *+gig*))
(defun trasfo_Re_2Pu_inf (Re1)
(trasfo_Re_2Pu_dis Re1 *+inf*))
```

Dato un piano si determina una terna di punti (3Pu App Re), il primo coincidente con l'origine del piano come definita in precedenza e gli altri due posti ad alcune delle distanze preassegnate.

```
(defun trasfo_Pi_3Pu_gra (Pi1)
(trasfo_Pi_3Pu_dis Pi1 *+gra*))
(defun trasfo_Pi_3Pu_gig (Pi1)
(trasfo_Pi_3Pu_dis Pi1 *+gig*))
(defun trasfo_Pi_3Pu_inf (Pi1)
(trasfo_Pi_3Pu_dis Pi1 *+inf*))
```

10 Distanze preimpostate Anche per altre procedure che seguiranno oltre definiamo dei valori numerici preimpostati come segue:

```
(setq *+0.0* 1e-8)
(setq *+pic* 1e-2)
(setq *+gra* 1e+2)
(setq *+gig* 1e+4)
(setq *+inf* 1e+8)
```

Nel codice realizzato il primo valore è assimilabile ad un valore nullo ed il quinto valore è assimilabile ad un valore infinito. Il settaggio approssimato del valore nullo *+0.0* e del valore infinito *+inf* si rende necessario stanti i limiti intrinseci della precisione di AutoCAD limitata ad un numero finito di cifre decimali. La trattazione esaustiva di questo aspetto esula dal contesto della trattazione prevista per questo numero della rubrica codici grafici, se ne darà un esempio: determinati con due procedure differenti due punti $Pu_1 (Pu_{1x}, Pu_{1y}, Pu_{1z})$ e $Pu_2 (Pu_{2x}, Pu_{2y}, Pu_{2z})$, i limiti di calcolo a otto cifre decimali ed i conseguenti arrotondamenti, potrebbero determinare differenze minime nei decimali finali delle loro coordinate anche se delle costruzioni geometriche sostitutive delle algoritmiche determinerebbero due punti coincidenti; in tal caso la loro distanza risulterebbe infinitesima e non nulla come dovrebbe.

11 Predicati sull'appartenenza per la verifica di tale condizione tra tra punti rette e piani:

```
(defun P_appar_Pu_Re (Pu1 Re1)
(P_coinc_Pu_Pu Pu1 (Pu_pro_Re Pu1 Re1)))
(defun P_appar_Pu_Pi (Pu1 Pi1)
(P_coinc_Pu_Pu Pu1 (Pu_pro_Pi Pu1 Pi1)))
```

```
(defun P_appar_Re_Pi (Re1 Pi1)
(P_coinc_Re_Re Re1 (Re_pro_Pi Re1 Pi1)))
12 Predicati sulla coincidenza per la verifica di tale condizione tra punti rette e piani:
(defun P_coinc_Pu_Pu (Pu1 Pu2)
(equal 0.0 (distance Pu1 Pu2) *+0.0*))
(defun P_coinc_Re_Re (Re1 Re2)
(and (P_paral_Re_Re Re1 Re2)
(P_coinc_Pu_Pu (3° Re1) (3° Re2))))
(defun P_coinc_Pi_Pi (Pi1 Pi2)
(and (P_paral_Pi_Pi Pi1 Pi2)
(P_coinc_Pu_Pu (3° Pi1) (3° Pi2))))
```

13 Si rimanda a quanto scritto nella nota 10.

14 Procedure sulle proiezioni ortogonali tra enti geometrici:

```
(defun Pu_pro_Re (Pu1 Re1)
(trans (comb_xy_Pu1_z_Pu2 (3° Re1) (trans Pu1 0 (1° Re1))) (1° Re1) 0))
(defun Pu_pro_Pi (Pu1 Pi1)
(trans (comb_xy_Pu1_z_Pu2 (trans Pu1 0 (1° Pi1)) (3° Pi1)) (1° Pi1) 0))
(defun Re_pro_Pi (Re1 Pi1 / IPu IPuP)
(apply 'trasfo_2Pu_Re (IPu_pro_Pi
(trasfo_Re_2Pu_gra Re1) Pi1)))
(defun IPu_pro_Pi (IPu Pi1)
(mapcar '(lambda (x) (Pu_pro_Pi x Pi1)) IPu))
```

15 Predicati sul parallelismo e sulla perpendicolarità per la verifica di tali condizioni tra due versori:

```
(defun P_paral_2Ver (Ver1 Ver2)
(or (equal Ver1 Ver2 *+0.0*)
(equal Ver1 (mapcar '(lambda (x) (* x -1)) Ver2) *+0.0*)))
(defun P_perpe_2Ver (Ver1 Ver2)
(equal 0.0 (apply '+ (mapcar '* Ver1 Ver2)) *+0.0*))
```

16 Predicati sul parallelismo e sulla perpendicolarità per la verifica di tali relazioni fondamentali tra rette e piani:

```
(defun P_paral_Re_Re (Re1 Re2)
(P_paral_2Ver (1° Re1) (1° Re2)))
(defun P_paral_Pi_Pi (Pi1 Pi2)
(P_paral_2Ver (1° Pi1) (1° Pi2)))
(defun P_paral_Re_Pi (Re1 Pi1)
(P_perpe_2Ver (1° Re1) (1° Pi1)))
(defun P_perpe_Re_Re (Re1 Re2)
(P_perpe_2Ver (1° Re1) (1° Re2)))
(defun P_perpe_Pi_Pi (Pi1 Pi2)
(P_perpe_2Ver (1° Pi1) (1° Pi2)))
(defun P_perpe_Re_Pi (Re1 Pi1)
(P_paral_2Ver (1° Re1) (1° Pi1)))
```

17 Predicati sull'allineamento e sulla complanarità per la verifica di tali condizioni tra tre punti rette e piani:

```
(defun P_allin_Pu_Pu_Pu (Pu1 Pu2 Pu3)
(or (P_coinc_Pu_Pu Pu1 Pu2)
(P_coinc_Pu_Pu Pu1 Pu3)
(P_coinc_Pu_Pu Pu2 Pu3)))
(P_coinc_Pu_Pu Pu3 (Pu_pro_2Pu Pu3 Pu1 Pu2))
(P_coinc_Pu_Pu Pu2 (Pu_pro_2Pu Pu2 Pu1 Pu3))
(P_coinc_Pu_Pu Pu1 (Pu_pro_2Pu Pu1 Pu2 Pu3)))
(defun P_compl_Pu_Pu_Pu (Pu1 Pu2 Pu3 Pu4)
(or (P_allin_Pu_Pu_Pu Pu1 Pu2 Pu3)
(P_allin_Pu_Pu_Pu Pu1 Pu2 Pu4)
(P_allin_Pu_Pu_Pu Pu1 Pu3 Pu4)
(P_allin_Pu_Pu_Pu Pu2 Pu3 Pu4)
(P_coinc_Pu_Pu Pu4 (Pu_pro_3Pu Pu4 Pu1 Pu2 Pu3))
(P_coinc_Pu_Pu Pu3 (Pu_pro_3Pu Pu3 Pu1 Pu2 Pu4))
(P_coinc_Pu_Pu Pu2 (Pu_pro_3Pu Pu2 Pu1 Pu3 Pu4))
(P_coinc_Pu_Pu Pu1 (Pu_pro_3Pu Pu1 Pu2 Pu3 Pu4))))
(defun P_compl_4Pu (Pu1 Pu2 Pu3 Pu4)
(P_compl_Pu_Pu_Pu_Pu Pu1 Pu2 Pu3 Pu4))
(defun P_compl_Pu_Pi (Pu1 Pi1) (P_coinc_Pu_Pu Pu1 (Pu_pro_Pi Pu1 Pi1)))
(defun P_compl_Re_Re (Re1 Re2 / Pu1 Pu2 Pu3 Pu4)
(mapcar 'set '(Pu1 Pu2) (trasfo_Re_2Pu_gra Re1))
(mapcar 'set '(Pu3 Pu4) (trasfo_Re_2Pu_gra Re2))
(P_compl_4Pu Pu1 Pu2 Pu3 Pu4))
```

È possibile predisporre per comodità delle procedure di trasformazione specifiche⁹ che restituiscano tali punti, a distanze preimpostate¹⁰ rispetto alle origini di rette e piani.

Predicati di verifica delle condizioni

I predicati che seguono si occupano della verifica delle condizioni fondamentali di appartenenza tra punti rette e piani, di parallelismo e perpendicolarità tra rette e piani e delle condizioni ausiliarie di allineamento e complanarità tra punti rette e piani.

La descrizione algoritmica di tali predicati è riportata caso per caso nel testo, la descrizione geometrica è invece riportata nella didascalia di Fig.1.

Le verifiche di appartenenza hanno interessato enti geometrici fondamentali di tipo differente¹¹:

- **(Pu App Re)** Un punto ed una retta si appartengono se il punto coincide con la sua proiezione ortogonale sulla retta;

- **(Pu App Pi)** Un punto ed un piano si appartengono se il punto coincide con la sua proiezione ortogonale sul piano;

- **(Re App Pi)** Una retta ed un piano si appartengono se la retta coincide con la sua proiezione ortogonale sul piano.

ma anche enti geometrici fondamentali dello stesso tipo, distinguendo però formalmente i predicati seguenti di verifica della coincidenza dai precedenti di verifica dell'appartenenza¹²:

- **(Pu App Pu)** Due punti sono coincidenti se la loro distanza è pari a 0 o inferiore a *+0.0*¹³;

- **(Re App Re)** Due rette sono coincidenti se sono parallele e le loro origini in coordinate WCS sono coincidenti;

- **(Pi App Pi)** Due piani sono coincidenti se sono paralleli e le loro origini in coordinate WCS sono coincidenti;

I predicati per le verifiche di appartenenza e coincidenza si avvalgono di procedure già descritte e di procedure dedicate alla proiezione ortogonale di punti e rette su piani¹⁴.

Le verifiche di parallelismo e di perpendicolarità si occupano solamente degli enti geometrici fondamentali dotati di un orientamento, ovvero di rette e piani: relativamente alle rette considerano

Entità	Appartenenza				Parallelismo				Perpendicolarità			
	ID	Relazione	Disegno	Verifica	ID	Relazione	Disegno	Verifica	ID	Relazione	Disegno	Verifica
Pu	A	$\circ \in /$	----	----								
	B	$\circ \in \square$	D A	(C D) (D C)								
Re	C	$/ \in \circ$	----	----	G	$/ // /$	----	----	O P	$/ \perp /$	R D F Q	(F R) (R F)
	D	$/ \in \square$	----	----	H I	$/ // \square$	N D D G	(N F) (D G)	Q	$/ \perp \square$	----	----
Pi	E	$\square \in \circ$	C F	(C D) (D C)	L M	$\square // /$	G F F N	(G D) (F N)	R	$\square \perp /$	----	----
	F	$\square \in /$	----	----	N	$\square // \square$	----	----	S T	$\square \perp \square$	Q F D R	(D Q) (Q D)

1 | Le relazioni fondamentali. Una schematizzazione delle condizioni fondamentali di appartenenza (da A ad F), di parallelismo (da G a N) di perpendicolarità (da O a T) tra punti rette e piani. Le condizioni che seguono possono essere applicate senza necessità di costruzioni ausiliarie:

- **A F** – Data una retta è possibile disegnare (Pu App Re) o Pi App Re, o dati punto e retta o retta e piano è possibile verificare la reciproca App.
 - **C D** – Dato un punto o un piano è possibile disegnare Re App Pu o Re App Pu; dati punto e retta o retta e piano è possibile verificare la reciproca App.
 - **G N** – Data una retta o un piano è possibile disegnare Re Par Re o Pi Par Pi, o dati due rette o due piani è possibile verificare il reciproco Par.
 - **Q R** – Data una retta o un piano è possibile disegnare Pi Per Re o Re Per Pi, o dati una retta e un piano è possibile verificare la reciproca Per.
- Per le condizioni che seguono l'assegnazione di una lettera alle varie condizioni permette la descrizione simbolica dell' algoritmo geometrico necessario alla loro applicazione.
- **B E** – Dato un piano per disegnare Pu App Pi è necessario avvalersi di D e A (Pu App Re App Pi), dato un punto per dis Pi App Pu è necessario avvalersi di C e F (Pi App Re App Pu); dati un punto e un piano per verificare la reciproca App è necessario avvalersi di C che ver D (Re App Pu Pi) o di D che ver C (Re App Pi Pu).
 - **H I** – Dato un piano per disegnare Re Par Pi è necessario avvalersi di N e D (Re App Pi Par Pi) o di D e G (Re Par Re App Pi) o dati una retta e un piano per verificare il reciproco Par è necessario avvalersi di N che ver F (Pi Par Pi e App Re) o di D che ver G (Re App Pi e Par Re).
 - **L M** – Data una retta per disegnare Pi Par Re è necessario avvalersi di G e F (Pi App Re Par Re) o di F e N (Pi Par Pi App Re) o dati una retta ed un piano per verificare il reciproco Par è necessario avvalersi di G che ver D (Re Par Re e App Pi) o di F che ver N (Pi App Re e Par Pi).
 - **O P** – Data una retta per disegnare Re Per Re è necessario avvalersi di R e D (Re App Pi Per Re) o di F e Q (Re Per Pi App Re); date due rette per verificare la reciproca Per è necessario avvalersi di F che ver R (Pi App Re e Per Re) o di R che ver F (Pi Per Re e App Re).
 - **S T** – Dato un piano per disegnare Pi Per Pi è necessario avvalersi di Q e F (Pi App Re Per Pi) o di D e R (Pi Per Re App Pi); dati due piani per verificare la reciproca Per è necessario avvalersi di D che ver Q (Re App Pi e Per Pi) o di Q che ver D (Re Per Pi e App Pi).

- 18 Procedure** sulle proiezioni ortogonali tra punti e per la determinazione di vettori tra punti:
- (defun Pu_pro_2Pu (Pu1 Pu2 Pu3 / vett_23) (setq vett_23 (vett_2Pu Pu2 Pu3)) (trans (comb_xy_Pu1_z_Pu2 (trans Pu2 0 vett_23) (trans Pu1 0 vett_23)) vett_23 0))
 - (defun vett_2Pu (Pu1 Pu2) (mapcar '- Pu2 Pu1))
 - (defun Pu_pro_3Pu (Pu1 Pu2 Pu3 Pu4 / vett_234) (setq vett_234 (vett_3Pu Pu2 Pu3 Pu4)) (trans (comb_xy_Pu1_z_Pu2 (trans Pu1 0 vett_234) (trans Pu2 0 vett_234)) vett_234 0))
 - (defun vett_3Pu (Pu1 Pu2 Pu3 / x1 x2 x3 y1 y2 y3 z1 z2 z3) (mapcar 'set '(x1 y1 z1 x2 y2 z2 x3 y3 z3) (append Pu1 Pu2 Pu3)) (list (- (* (- y2 y1)(- z3 z1)) (* (- y3 y1)(- z2 z1)) (- (* (- x3 x1)(- z2 z1)) (* (- x2 x1)(- z3 z1))) (- (* (- x2 x1)(- y3 y1)) (* (- x3 x1)(- y2 y1))))))
- 19 Il disegno delle condizioni** tramite procedure è limitato necessariamente a quelle che determinano univocamente un ente, per l'appartenenza: (Pu App 2Re), (Pu App 3Pi), (Re App 2Pu), (Re App 2Pi), (Pi App 2Re), (Pi App 3Pu); mentre, sempre per l'appartenenza: (Pu App Re), (Pu App Pi) (Pu App 2Pi), (Re App Pu), (Re App Pi), (Pi App Re), (Pi App Pu), (Pi App 2Pu) saranno condizioni ammissibili solamente per una verifica.
- 20 Le distanze minima e massima** rispetto ad un singolo ente geometrico saranno rispettivamente: nulla perché il punto da sé stesso, i punti dalla retta punteggiata e i punti del piano punteggiato si troveranno per coincidenza o per appartenenza a distanza nulla dall'ente geometrico; infinita perché dall'ente potremo allontanarci ponendoci ad una distanza via via crescente e tendente ad infinito.
- 21 I luoghi geometrici di distanza specifica** dei singoli enti geometrici saranno rispettivamente: per un punto una superficie sferica di raggio DSp ed avente il centro nel punto dato, per la retta una superficie cilindrica circolare retta di raggio DSp ed asse coincidente con la retta data e per un piano una coppia di superfici piane poste a distanza DSp dal piano dato, una per ciascun semispazio individuato dal piano stesso. Tali luoghi geometrici, a distanza nulla coincideranno con l'ente: la superficie sferica di raggio 0 si ridurrà ad un punto, il cilindro di raggio 0 si ridurrà ad una retta e i due piani a distanza 0 dal piano dato coincideranno con esso.

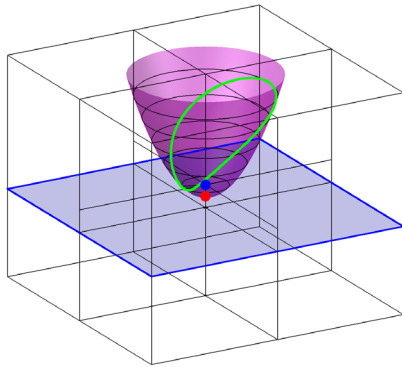
solamente la loro direzione e relativamente ai piani considerano solamente la loro giacitura. Dato che direzioni e giaciture sono parte delle loro definizioni come versore parallelo alla retta e come versore ortogonale al piano è molto agevole definire i predicati relativi a tali verifiche perché, a seconda dei casi, sarà sufficiente verificare il parallelismo o la perpendicolarità dei due versori coinvolti con degli specifici predicati¹⁵.

Si sono considerate le seguenti verifiche di parallelismo e perpendicolarità tra rette e piani¹⁶:

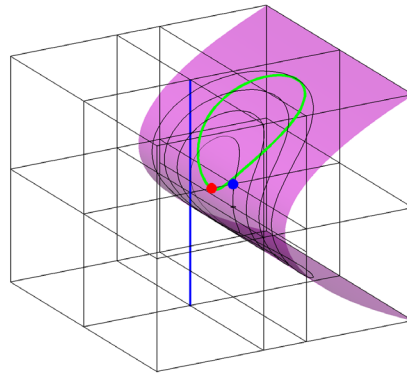
- **(Re Par Re)** Due rette sono tra loro parallele se lo sono i loro versori;
- **(Re Par Re)** Due piani sono tra loro paralleli se lo sono i loro versori;
- **(Re Par Pi)** Una retta ed un piano sono tra loro paralleli se i loro versori sono perpendicolari;
- **(Re Per Re)** Due rette sono tra loro perpendicolari se lo sono i loro versori;
- **(Pi Per Pi)** Due piani sono tra loro perpendicolari se lo sono i loro versori;
- **(Re Per Pi)** Una retta ed un piano sono tra loro perpendicolari se i loro versori sono paralleli.

Si sono considerate le seguenti verifiche di allineamento e complanarità tra punti rette e piani¹⁷:

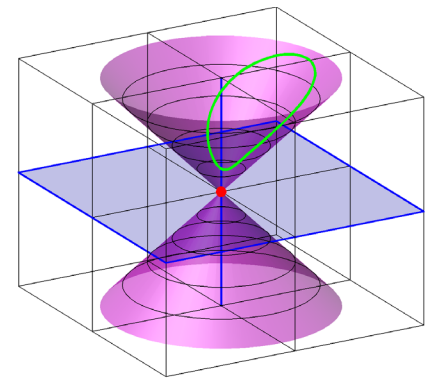
- **(3Pu All)** Tre punti sono allineati tra loro per l'eventuale presenza: di coincidenze tra due dei tre punti o di coincidenze tra uno dei punti e la sua proiezione sulla retta definita dagli altri due punti.
- **(4Pu Com)** Quattro punti sono complanari tra loro per l'eventuale presenza: di coincidenze tra due dei quattro punti, di allineamenti tra tre dei quattro punti o di coincidenze tra uno dei punti



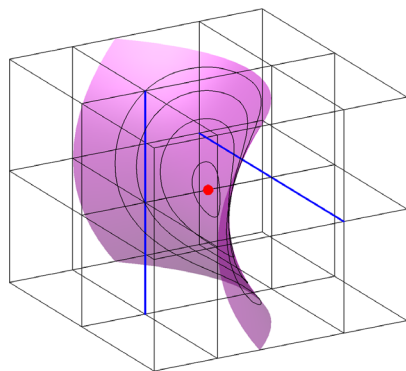
2 | L'equidistanza tra un punto ed un piano. Dati (in blu) un punto e un piano è possibile determinare geometricamente: il luogo geometrico (in magenta) a cui appartengono tutti i punti che si trovano alla stessa distanza dai due enti geometrici, LG generalmente rappresentato da un paraboloido circolare (ellittico di rotazione) di equidistanza complessiva EQCo (Pu Pi); la curva quartica (in verde) LG di equidistanza complessiva EQCo (Pu Re Pi) (si veda Fig.9) appartenente al LG EQCo (Pu Pi); circonferenze (in nero) LG di equidistanza specifica EQSp (Pu Pi) appartenenti al LG EQCo (Pu Pi); il punto (in rosso) di equidistanza minima EQMi (Pu Pi).



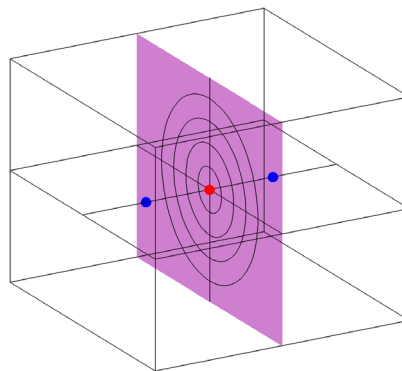
3 | L'equidistanza tra un punto ed una retta. Dati (in blu) un punto e una retta è possibile determinare geometricamente il luogo geometrico (in magenta) a cui appartengono tutti i punti che si trovano alla stessa distanza dai due enti geometrici, LG generalmente rappresentato da un cilindro parabolico di equidistanza complessiva EQCo (Pu Re); la curva quartica (in verde) LG di equidistanza complessiva EQCo (Pu Re Pi) (si veda Fig.9) appartenente al LG EQCo (Pu Re); curve (in nero) di equidistanza specifica EQSp (Pu Re) appartenenti al LG EQCo (Pu Re); il punto (in rosso) di equidistanza minima EQMi (Pu Re).



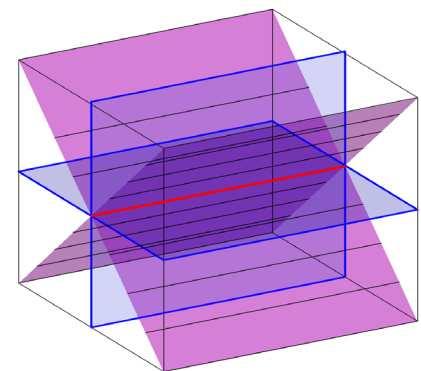
4 | L'equidistanza tra una retta ed un piano. Dati (in blu) retta e piano è possibile determinare geometricamente il luogo geometrico (in magenta) a cui appartengono tutti i punti che si trovano alla stessa distanza dai due enti geometrici, LG generalmente rappresentato da un cono ellittico retto a due falde (in figura, nel caso particolare di Re Per Pi, rappresentato da un cono circolare retto a due falde) di equidistanza complessiva EQCo (Re Pi); la curva quartica (in verde) LG di equidistanza complessiva EQCo (Pu Re Pi) (si veda Fig.9) appartenente al LG EQCo (Re Pi); circonferenze (in nero) di equidistanza specifica EQSp (Re Pi) appartenenti al LG EQCo (Re Pi); il punto (in rosso) di equidistanza minima EQMi (Re Pi).



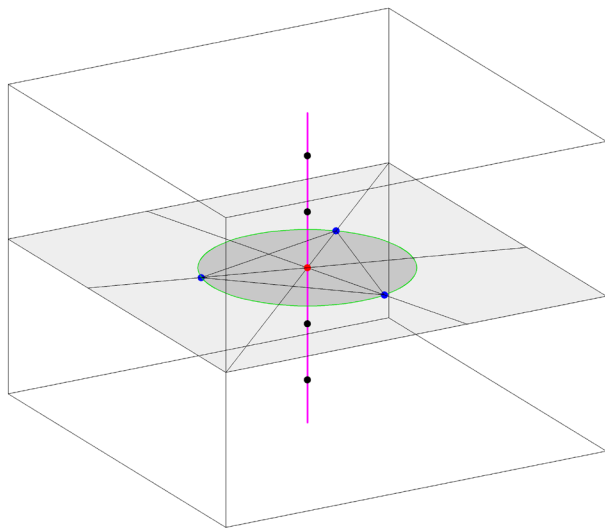
5 | L'equidistanza tra due rette. Date (in blu) due rette è possibile determinare geometricamente: il luogo geometrico (in magenta) a cui appartengono tutti i punti che si trovano alla stessa distanza dai due enti geometrici, LG generalmente rappresentato da un paraboloido iperbolico (o a sella) (in figura, nel caso particolare Re Per Re) di equidistanza complessiva EQCo (Re Re); curve (in nero) di equidistanza specifica EQSp (Re Re) appartenenti al LG EQCo (Re Re); il punto (in rosso) di equidistanza minima EQMi (Re Re).



6 | L'equidistanza tra due punti. Dati (in blu) due punti è possibile determinare geometricamente: il luogo geometrico (in magenta) a cui appartengono tutti i punti che si trovano alla stessa distanza dai due enti geometrici, LG rappresentato da una superficie piana (quadrica data da due piani coincidenti) di equidistanza complessiva EQCo (Pu Pu); circonferenze (in nero) LG di equidistanza specifica EQSp (Pu Pu) appartenenti al LG EQCo (Pu Pu); il punto (in rosso) di equidistanza minima EQMi (Pu Pu). Supponiamo la coppia di punti non coincidenti Pu1 Pu2 (in blu), posti a distanza 100, il loro punto medio Pu3 (in rosso) sarà equidistante 50 da entrambi (EQMi), tale punto apparterrà alla retta Re1 per i punti dati e al piano Pi1 (in viola) ortogonale a Re1 ed appartenente a Pu3; sul piano Pi1 luogo geometrico di equidistanza complessiva EQCo si troveranno infinite circonferenze (in nero) di centro Pu3 luogo geometrico di equidistanza specifica EQSp (ad esempio i punti di una circonferenza di raggio 50 saranno equidistanti 70.71 dai punti dati).



7 | L'equidistanza tra due piani. Dati (in blu) due piani è possibile determinare geometricamente il luogo geometrico (in magenta) a cui appartengono tutti i punti che si trovano alla stessa distanza dai due enti geometrici; LG generalmente rappresentato dalla coppia di piani bisettori dei due piani dati (quadrica data da due piani incidenti) di equidistanza complessiva EQCo (Pi Pi); quaterne di rette (in nero), due per piano bisettore, LG di equidistanza specifica EQSp (Pi Pi) appartenenti al LG EQCo (Pi Pi); la retta (in rosso) di equidistanza minima EQMi (Pi Pi).



e la sua proiezione sul piano definito dagli altri tre punti;

- **(Pu Com Pi)** Un punto è complanare ad un piano per l'eventuale coincidenza del punto con la sua proiezione sul piano;
- **(2Re Com)** Due rette sono complanari se è verificata la complanarità dei quattro punti ricavati a coppie dalle due rette.

I predicati per le verifiche di allineamento e complanarità si avvalgono di procedure geometriche dedicate alla proiezione ortogonale di punti su coppie o terne di punti e per la determinazione di vettori paralleli o ortogonali, rispettivamente a coppie o terne di punti¹⁸.

Predicati di disegno delle condizioni

La trattazione delle funzioni di disegno delle condizioni fondamentali di appartenenza, parallelismo e perpendicolarità tra punti rette e piani¹⁹ è limitata alla loro descrizione geometrica riportata nella didascalia della tabella di Fig.1.

L'equidistanza tra enti geometrici fondamentali

La trattazione delle funzioni dedicate alle equidistanze è limitata ai concetti introduttivi, alla nomenclatura e ad una serie di immagini indicate a seguire.

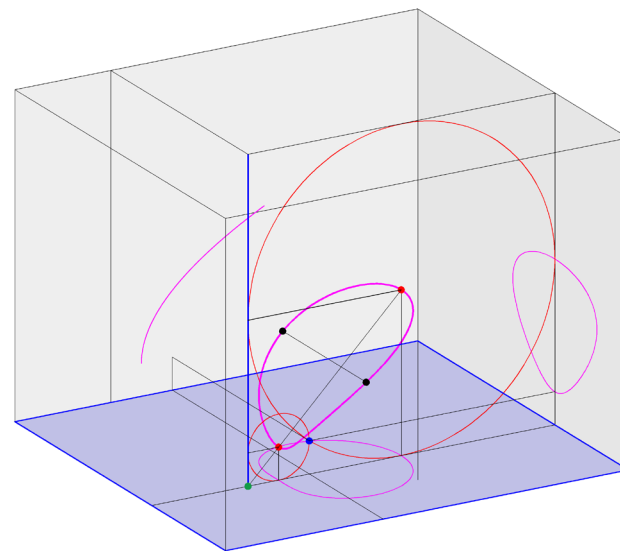
Dato un singolo ente geometrico fondamentale (**Pu**, **Re**, **Pi**), è possibile definire rispetto ad esso: • un valore nullo di distanza minima (**DMi**); • un valore infinito di distanza massima (**DMA**)²⁰; • il luogo geometrico, rappresentato da

una **superficie**²¹ di distanza specifica (**DSp**), contenente tutti i punti tra loro equidistanti dall'ente geometrico e posti a distanza compresa tra **DMi** e **DMA**;

- il luogo geometrico, rappresentato dall'intero spazio 3D di distanza complessiva (**DCo**) contenente tutte le superfici **DSp**.

Data una coppia di enti geometrici di tipo differente (**PuRe** Fig.3, **PuPi** Fig.2, **RePi** Fig.4) o dello stesso tipo (**2Pu** Fig.6, **2Re** Fig.5 **2Pi** Fig.7) è possibile definire rispetto ad essi: • un valore (nullo se incidenti) di equidistanza minima (**EQMi**); • un valore (infinito) di equidistanza massima (**EQMa**); • un luogo geometrico di equidistanza specifica (**EQSp**), rappresentato in genere da una curva contenente tutti i punti posti ad un'equidistanza specifica compresa tra **EQMi** e **EQMa**; • un luogo geometrico di equidistanza complessiva (**EQCo**), rappresentato in genere da una superficie quadrica contenente tutte le curve **EQSp**.

Data una terna di enti geometrici di tipo differente (**2PuRe**, **2PuPi**, **Pu2Re** Fig.10, **Pu2Pi**, **PuRePi** Fig.9 e 10, **2RePi** Fig.10, **Re2Pi**) o dello stesso tipo (**3Pu** Fig.8, **3Re**, **3Pi**) è possibile definire rispetto ad essi: • un valore (nullo se tutti incidenti) di equidistanza minima (**EQMi**); • un valore (in alcuni casi infinito) di equidistanza massima (**EQMa**); • una serie finita di punti posti ad un valore di equidistanza specifica (**EQSp**) compresa tra **EQMi** e **EQMa**; • un luogo geometrico di equidistanza complessiva (**EQCo**), rappresentato in genere da una curva quartica contenente tutte le serie di punti **EQSp**. sentato dall'intero spazio 3D di distanza complessiva (**DCo**) contenente tutte le superfici **DSp**.



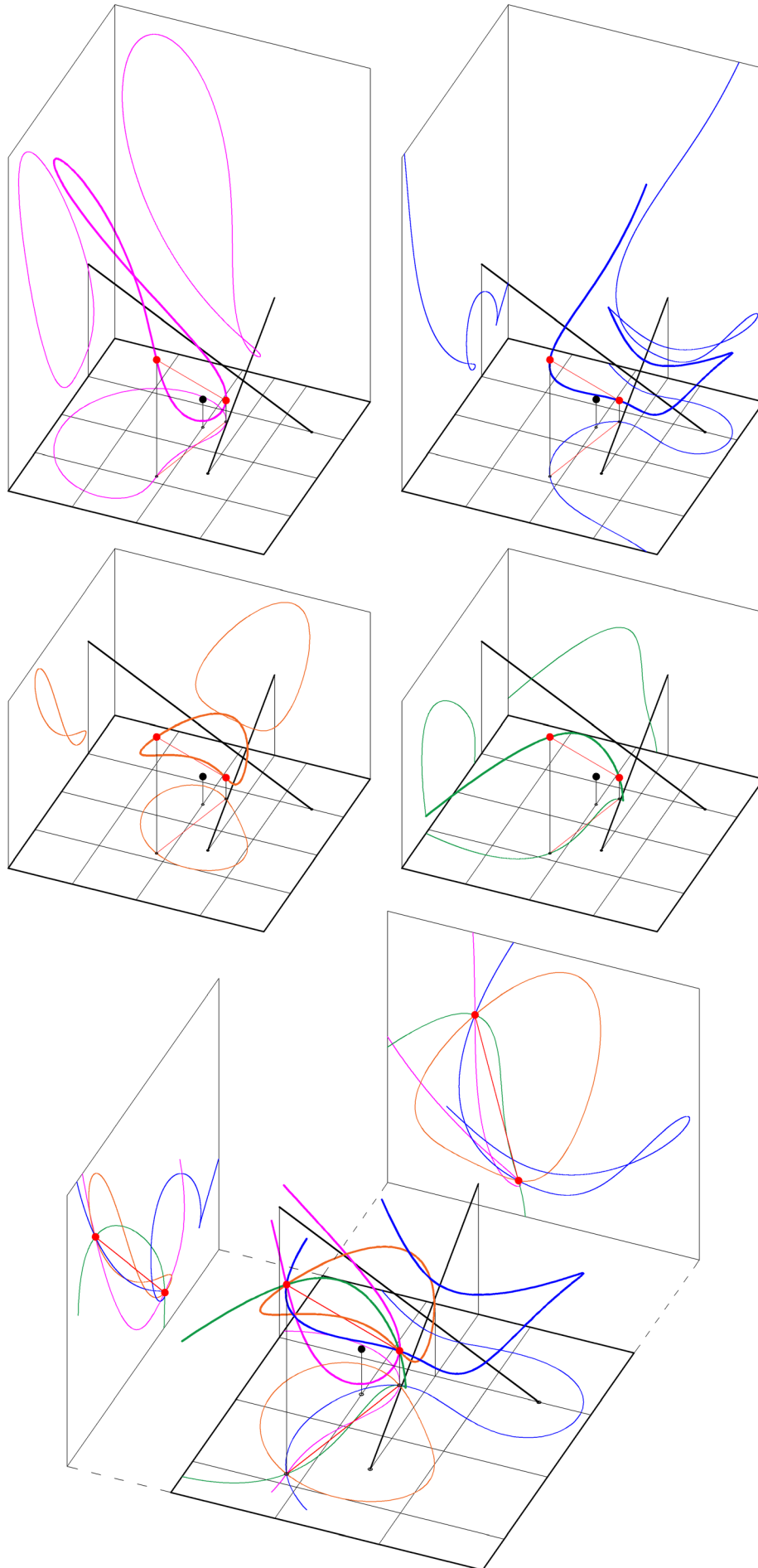
8 | L'equidistanza tra tre punti.

Dati (in blu) tre punti è possibile determinare geometricamente: il luogo geometrico (in magenta) a cui appartengono tutti i punti che si trovano alla stessa distanza dai tre enti geometrici, **LG** rappresentato da una retta di equidistanza complessiva **EQCo** (**Pu Pu Pu**); coppie di punti (in nero) di equidistanza specifica **EQSp** (**Pu Pu Pu**) appartenenti al **LG EQCo** (**Pu Pu Pu**); il punto (in rosso) di equidistanza minima **EQMi** (**Pu Pu Pu**).

Supponiamo la terna di punti non coincidenti e non allineati **Pu1 Pu2 Pu3** (in blu), posti a formare un triangolo equilatero di lato 100, il loro circocentro **Pu4** (in rosso) sarà equidistante 57,735 dai tre punti dati, valore di equidistanza minima **EQMi**, tale punto apparterrà al piano **Pi1** (in grigio) per i punti dati ed alla retta **Re1** (in viola) appartenente a **Pu4** e perpendicolare a **Pi1**; su **Re1** si troveranno infinite coppie di punti (in nero), simmetrici a **Pu4** e a **Pi1** posti ad una equidistanza specifica **EQSp** (**Pu Pu Pu**) dai punti dati (ad esempio i punti a distanza 50 da **Pu4** saranno equidistanti 76,376 dai punti dati); la retta **Re1**, contenente tutti i punti equidistanti dai punti dati definirà il luogo geometrico di equidistanza complessiva **EQCo**.

9 | L'equidistanza tra un punto una retta ed un piano.

Dati (in blu) un punto, una retta ed un piano è possibile determinare geometricamente: il luogo geometrico (in magenta) a cui appartengono tutti i punti che si trovano alla stessa distanza dai tre enti geometrici dati, **LG** rappresentato da una curva quartica (monogrammica e simmetrica rispetto al piano definito dal punto e dalla retta dati) di equidistanza complessiva **EQCo** (**Pu Re Pi**); due circonferenze complanari (in rosso) aventi per centri una coppia di punti (in rosso) di equidistanza minima **EQMi** (**Pu Re Pi**) e massima **EQMa** (**Pu Re Pi**) appartenenti al **LG EQCo** (circonferenze passanti per il punto dato e tangenti alla retta e al piano dato a dimostrazione dell'equidistanza dei due centri dai tre enti geometrici fondamentali assegnati); coppie di punti (in nero) di equidistanza specifica **EQSp** (**Pu Re Pi**) appartenenti, anch'essi al **LG EQCo**. La curva quartica di equidistanza complessiva dei tre enti geometrici assegnati deriva dall'intersezione a coppie delle tre superfici quadriche di equidistanza complessiva ottenibili dai tre enti geometrici assegnati presi a coppie (Fig. 2, 3, 4)



“ Le nozioni classiche dei fondamenti della geometria descrittiva possono essere reinterpretate con algoritmi informatici che ne automatizzano ed ampliano le potenzialità applicative.

10 | L'equidistanza tra un punto due rette ed un piano. Dati (in nero) un punto, due rette ed un piano è possibile determinare geometricamente (in rosso) i punti di equidistanza complessiva EQCo ($Pu\ 2Re\ Pi$); tali punti derivano dall'intersezione a coppie delle quattro curve quartiche equidistanza complessiva presenti nelle prime quattro immagini (in magenta, blu, arancione e verde) ed ottenibili dai quattro enti geometrici dati presi a terne; nell'immagine finale un ingrandimento di sintesi.

Bibliografia

Autodesk, *AutoCAD: manuale di personalizzazione release 13*, Autodesk development B.V., 1994.

Anzani Giovanni, *Algoritmi di geometria descrittiva in AutoLISP su punti rette e piani*, Lulu, 2018.

Berzolari Luigi, *Geometria analitica – II, Curve e superficie del secondo ordine*, Editore Ulrico Hoepli, Milano, 1916.

Campedelli Luigi, *Lezioni di geometria Vol. 1° e 2*, Cedam, Padova, 1967

Standiford Kevin, *AutoLISP to Visual LISP*. Thomson Learning (Autodesk Press), Canada, 2001.