Equations élémentaires pour la méthode de Paul

	Équation (s)	inconnue (s)	fonctions matlab fichiers: *solve_type_i.m*
arctan2	$\begin{cases}[1]\sin(A_i)-\mathrm{Im}=0\\ [2]\cos(A_i)-\mathrm{Re}=0\end{cases}$	Ai	Ai=atan2(Im,Re) (Ai est l’argument de z= Re+j.Im)
Type 1	$X.L_i-Y=0$	L_i	S=*solve_type_1*(X,Y)
Type 2	$X.\sin(A_i)+Y.\cos(A_i)-Z=0$	A_i	S=*solve_type_2*(X,Y,Z)
Type 3	$\begin{cases}[1]X_1.\sin(A_i)+Y_1.\cos(A_i)-Z_1=0\\ [2]X_2.\sin(A_i)+Y_2.\cos(A_i)-Z_2=0\end{cases}$	A_i	S=*solve_type_3*(X1,Y1,Z1,X2,Y2,Z2)
Type 4	$\begin{cases}[1]X_1.R_j.\sin(A_i)-Y_1=0\\ [2]X_2.R_j.\cos(A_i)-Y_2=0\end{cases}$	$[R_j,A_i]$	S=*solve_type_4*(X1,Y1,X2,Y2)
Type 5	$\begin{Bmatrix}[1]\ \ X_1.\sin(A_i)-Y_1-Z_1.R_j=0\\ [2]\ \ X_2.\cos(A_i)-Y_2-Z_2.R_j=0\end{Bmatrix}$	$[R_j,A_i]$	S=*solve_type_5*(X1,Y1,Z1,X2,Y2,Z2)
Type 6	$\begin{cases}[1]\ \ W.\sin(A_k)-X.\cos(A_i)-Y.\sin(A_i)-Z1=0\\ [2]\ \ W.\cos(A_k)-X.\sin(A_i)+Y.\cos(A_i)-Z2=0\end{cases}$	$[A_k,A_i]$	S=*solve_type_6*(W,X,Y,Z1,Z2)
Type 7 modifié	$\begin{cases}W2.\sin(A_k)+W1.\cos(A_k)-X.\cos(A_i)-Y.\sin(A_i)-Z1\\ W2.\cos(A_k)-W1.\sin(A_k)-X.\sin(A_i)+Y.\cos(A_i)-Z2\end{cases}$	$[A_k,A_i]$	S=*solve_type_7*(W1,W2,X,Y,Z1,Z2)
Type 8 modifié	$\begin{cases}X.\cos(A_i)+Y.\cos(A_k)-Z1=0\\ X.\sin(A_i)+Y.\sin(A_k)-Z2=0\end{cases}$	$[A_k,A_i]$	S=*solve_type_8*(X,Y,Z1,Z2)
Type générique 2 équs 2 angles	$\begin{cases}V1.\cos(A_i)+W1.\sin(A_i)+X1.\cos(A_k)+Y1.\sin(A_k)-Z1=0\\ V2.\cos(A_i)+W2.\sin(A_i)+X2.\cos(A_k)+Y2.\sin(A_k)-Z2=0\end{cases}$	$[A_k,A_i]$	S=*solve_type_g*(V1,W1,X1,Y1,V2,W2,X2,Y2,Z2)

Le tableau ci-avant liste les différentes équations de Paul, ainsi que les fonctions de résolution correspondantes, en langage matlab, dans les fichiers : *solve_type_1.m,…,solve_type_8.m* .

Ces fonctions renvoient les résultats dans une structure *S*, qui contient systématiquement les champs suivants :

S.well_posed	*%1 si l'équation est bien posée, 0 (faux) sinon*
	%par exemple une équation de type 1 avec X=Y=0 est mal posée)
S.nb_sols	*%nombre de solutions de l'équation, n'est valable que pour une équation bien posée*
S.name	*%chaîne de caractères rappelant le type, les inconnues et les paramètres de l'équation*

Les solutions sont stockées dans des vecteurs matlab dont les noms correspondent aux inconnues indiquées dans le tableau.
Par exemple, pour une équation de *type 5*, la structure *S* contiendra 2 tableaux, *S.Ai*(1..S.nb_sols) et *S.Rj*(1..nb_sols), correspondant aux couples de solution $[R_j,A_i]$

% exemple : programme de résolution et affichage des solutions pour une équation de type 5
% vous pouvez également visualiser toutes les infos contenues dans S5, en affichant S5 sous matlab :>> S5

```
X1=1;Y1=1;Z1=1;
X2=2;Y2=-1;Z2=1;
S5=solve_type_5(X1,Y1,Z1,X2,Y2,Z2);
if ( S5.well_posed~=1 )
  error(’ equation mal posee !..’);
end
msg=sprintf(’les %d solutions de l equation %s : sont :‘,S5.nb_sols,S5.name);
disp(msg) ;
for i=1:S5.nb_sols,
  Ai=S5.Ai(i);
  Rj=S5.Rj(i);
  msg=sprintf(’solution %d [ Ai=%.15e, Rj=%.15e ]‘ A,Ai,Rj) ;
  disp(msg) ;
end
```

Sortie matlab correspondante :
>>les 2 solutions de l equation : equation de type 5 :[1]X1.sin(Ai)=Y1 +Z1 .Rj ;[2] X2. cos(Ai)=Y2 + Z2 . Rj sont
>>solution 1 [Ai = 3.1415927, Rj =-1]
>>solution 2 [Ai = 2.2142974, Rj =-0.2]

ANNEXE : DETAIL DES SOLUTIONS :

Dans cette partie, on utilisera la fonction **atan2**, qui est une fonction de 2 variables *Im* et *Re.*
Soit un nombre complexe z=Re+ j.Im , alors atan2(Im,Re) est la fonction qui renvoie l'argument de z, dans l'intervalle $]-\pi,+\pi]$.

lorsque z est à partie réelle strictement positive *: atan2(Im,Re)=* $\arctan\left(\frac{\mathrm{Im}}{\mathrm{Re}}\right)$

lorsque z est à partie réelle strictement négative *:atan2(Im,Re)=* $\left[\pi+\arctan\left(\frac{\mathrm{Im}}{\mathrm{Re}}\right)\right]\mathrm{modulo}\ \pi$

dans tous les cas :*atan2(Im,Re)=* $Arg(\mathrm{Re}+j.\mathrm{Im})$

Type 1 : on cherche L_i telle que : [1] $X.L_i=Y$

Solution : [1.1] $L_i=\frac{X}{Y}$

Type 2 : on cherche $\{A_i\}$ tels que : [2] $X.\sin(A_i)+Y.\cos(A_i)=Z$

Une personne habituée voit une équation du type $\sin(\alpha+A_i)=cte$ *, et la met en évidence en procédant comme suit :*

étape 1- elle multiplie l'équation [2] par $\frac{1}{\sqrt{X^2+Y^2}}$, ce qui donne :

[2.1] $\underbrace{\frac{X}{\sqrt{X^2+Y^2}}}_{C}.\sin(A_i)+\underbrace{\frac{Y}{\sqrt{X^2+Y^2}}}_{S}.\cos(A_i)=\underbrace{\frac{Z}{\sqrt{X^2+Y^2}}}_{cte}$

dans cette équation, on peut poser $\begin{cases}\cos(\alpha)=C=\frac{X}{\sqrt{(X^2+Y^2)}}\\ \sin(\alpha)=S=\frac{Y}{\sqrt{(X^2+Y^2)}}\end{cases}$, car $C^2+S^2=1$, on obtient donc

[2.2] $\underbrace{\cos(\alpha).\sin(A_i)+\sin(\alpha).\cos(A_i)}_{\sin(\alpha+A_i)}=\frac{Z}{\sqrt{X^2+Y^2}}$

cette équation n'admet de solution que si $Z^2\leqslant X^2+Y^2$, qui sont alors :

$$\begin{cases}\text{solution 1 }: A_{i1}=\arcsin\left(\frac{Z}{\sqrt{(X^2+Y^2)}}\right)-\alpha\\ \text{solution 2 }: A_{i2}=\pi-\arcsin\left(\frac{Z}{\sqrt{(X^2+Y^2)}}\right)-\alpha\end{cases}, \text{ où } \alpha=atan2(Y,X)=argument(X+j.Y)$$

Type 3 : On cherche $\{A_i\}$ tels que $\begin{cases}[1]X_1.\sin(A_i)+Y_1.\cos(A_i)=Z_1\\ [2]X_2.\sin(A_i)+Y_2.\cos(A_i)=Z_2\end{cases}$

On reconnaît 2 équations de type 2, il suffit donc de retenir l'intersection des solutions de chacune des équations(avec une tolérance donnée lors de la résolution numérique).

Type 4 : On cherche l'ensemble des couples $\{R_j, A_i\}$ tels que $\begin{cases}[1] X_1 . R_j . \sin(A_i)=Y_1 \\ [2] X_{2.} R_j . \cos(A_i)=Y_2\end{cases}$

On ne considère pas ici le cas où l'une des variables X_1 , X_2 est égale à 0, auquel cas on a une infinité de solutions en $\{R_j, A_i\}$, évidentes à trouver...
En multipliant l'équation 1 par X_2 , et l'équation 2 par X_1 , puis en sommant les carrés des 2 équations, on voit que l'on obtient 2 solutions en Rj

$$R_j = \pm\sqrt{\left(\frac{X_2^2 . Y_1^2 + X_1^2 . Y_2^2}{X_1^2 . X_2^2}\right)}$$

connaissant R_j , la solution en A_i est unique (le sinus et le cosinus étant simultanément imposés) :

$$A_i = arg\left(\frac{Y2}{X2 . R_j} + j . \frac{Y1}{X1 . R_j}\right) = atan2\left(Y1 . sign(X1 . R_j), Y2 . sign(X2 . R_j)\right)$$

remarque : ces équations sont "dangereuses" dans le cas où l'une des variables X_1 , X_2 est proche de zéro.

Type 5 : On cherche $\{R_j, A_i\}$ tels que $\begin{Bmatrix}[1]\ X_1 . \sin(A_i)=Y_1+Z_1 . R_j \\ [2]\ X_{2.} \cos(A_i)=Y_2+Z_2 . R_j\end{Bmatrix}$

On ne considère pas ici le cas où l'une des variables X_1 , X_2 est égale à 0, auquel cas on a une infinité de solutions en A_i, R_j, évidentes à trouver...
En multipliant l'équation 1 par X_2 , et l'équation 2 par X_1 , puis en sommant les carrés des 2 équations, on voit que l'on obtient une équation du second degré à une inconnue en R_j

$$a_0 + a_1 . R_j + a_2 . R_j^2 = 0 \quad \text{avec} \quad \begin{Bmatrix} a_0 = X2^2 . Y1^2 + X1^2 . Y2^2 - X1^2 . X2^2 \\ a_1 = 2.\left(X2^2 . Y1.Z1 + X1^2 . Y2.Z2\right) \\ a_2 = X2^2 . Z1^2 + X1^2 . Z2^2 \end{Bmatrix}$$

si le discriminant $\Delta = a_1^2 - 4 . a_0 . a_2$ de cette équation est < 0 on n'a pas de solution,

sinon on obtient 2 solutions possibles en R_j : $\begin{Bmatrix} R_{j1} = \dfrac{-a_1 - \sqrt{(\Delta)}}{2. a_2} \\ R_{j2} = \dfrac{-a_1 + \sqrt{(\Delta)}}{2. a_2} \end{Bmatrix}$

Pour chacune de ces solutions , on obtient une solution correspondante en A_i , puisque connaissant Rj, la solution en Ai est unique (le sinus et le cos sont simultanément imposés) :

$$A_{i1,2} = arg\left(\frac{Y2 + Z2 . R_{j1,2}}{X2} + j . \frac{Y1 + Z1 . R_{j1,2}}{X1}\right) = atan2\left(\frac{Y1 + Z1 . R_{j1,2}}{X1}, \frac{Y2 + Z2 . R_{j1,2}}{X2}\right)$$

remarque : cette équation est "dangereuse" dans le cas où l'une des variables X_1 , X_2 est proche de zéro.

Type 6 : On cherche $\{A_k, A_i\}$ tels que $\begin{cases}[1]\ W . \sin(A_k) = X . \cos(A_i) + Y . \sin(A_i) + Z1 \\ [2]\ W . \cos(A_k) = X . \sin(A_i) - Y . \cos(A_i) + Z2\end{cases}$

on somme les carrés des 2 équations, ce qui donne

$$\underbrace{W^2-Z1^2-Z2^2-X^2-Y^2}_{Z'}=\underbrace{[2.X.Z2+2.Y.Z1]}_{X'}.\sin(A_i)+\underbrace{[2.X.Z1-2.Y.Z2]}_{Y'}.\cos(A_i)$$

On est alors ramené à une équation de **type 2** en A_i : $X'.\sin(A_i)+Y'.\cos(A_i)=Z'$ qui au plus admet 2 solutions A_{i1}, A_{i2}

en posant $\begin{cases} I(A_i)=X.\cos(A_i)+Y.\sin(A_i)+Z1 \\ R(A_i)=X.\sin(A_i)-Y.\cos(A_i)+Z2 \end{cases}$

Pour chacune de ces solutions , il existe une seule solution en A_k , donnée par

$$A_{k1,2}=arg\left(\frac{R(A_{i1,2})}{W}+j.\frac{I(A_{i1,2})}{W}\right)=atan2\left(I(A_{i1,2}).sign(W),R(A_{i1,2}).sign(W)\right)$$

Type 7 (un peu modifié) : On cherche $\{A_k, A_i\}$ tels que

$$\begin{cases} W2.\sin(A_k)+W1.\cos(A_k)=X.\cos(A_i)+Y.\sin(A_i)+Z1 \\ W2.\cos(A_k)-W1.\sin(A_k)=X.\sin(A_i)-Y.\cos(A_i)+Z2 \end{cases}$$

En posant $\begin{Bmatrix} W=\sqrt{(W1^2+W2^2)} \\ \sin(\alpha)=\dfrac{W1}{\sqrt{(W1^2+W2^2)}} \\ \cos(\alpha)=\dfrac{W2}{\sqrt{(W1^2+W2^2)}} \end{Bmatrix}$ soit encore , $\alpha=arg(W2+j.W1)=atan2(W1,W2)$

l'équation se réécrit alors

$\begin{Bmatrix} W.\sin(\underbrace{\alpha+A_k}_{A_k'})=X.\cos(A_i)+Y.\sin(A_i)+Z1 \\ W.\cos(\underbrace{\alpha+A_k}_{A_k'})=X.\sin(A_i)-Y.\cos(A_i)+Z2 \end{Bmatrix}$, ce qui est une équation de *type 6* en A_k' et A_i

Type 8 (un peu modifié) : On cherche $\{A_k, A_i\}$ tels que

$$\begin{cases} X.\cos(A_i)+Y.\cos(A_k)=Z1 \\ X.\sin(A_i)+Y.\sin(A_k)=Z2 \end{cases}$$

on passe les termes en Ai à droite dans les équations,

$$\begin{cases} Y.\cos(A_k)=Z1-X.\cos(A_i) \\ Y.\sin(A_k)=Z2-X.\sin(A_i) \end{cases}$$

on somme les carrés des 2 équations, ce qui donne

$$\underbrace{Y^2-Z1^2-Z2^2-X^2}_{Z'}=\underbrace{[-2.X.Z2]}_{X'}.\sin(A_i)+\underbrace{[-2.X.Z1]}_{Y'}.\cos(A_i) \quad ,$$

On est alors ramené à une équation de **type 2** en A_i : $X'.\sin(A_i)+Y'.\cos(A_i)=Z'$ qui au plus admet 2 solutions A_{i1}, A_{i2}

en posant $\begin{cases} I(A_i)=Z2-X.\sin(A_i) \\ R(A_i)=Z1-X.\cos(A_i) \end{cases}$ pour chacune de ces solutions, on trouve la solution A_k , correspondante :

$$A_{k1,2}=arg\left(\frac{R(A_{i1,2})}{Y}+j.\frac{I(A_{i1,2})}{Y}\right)=atan2\left(I(A_{i1,2}).sign(Y),R(A_{i1,2}).sign(Y)\right)$$

Type générique : On cherche $\{A_k, A_i\}$ tels que

$$\begin{cases} V1.\cos(A_i)+W1.\sin(A_i)+X1.\cos(A_k)+Y1.\sin(A_k)-Z1=0 \\ V2.\cos(A_i)+W2.\sin(A_i)+X2.\cos(A_k)+Y2.\sin(A_k)-Z2=0 \end{cases}$$

AVERTISSEMENT : la méthode de résolution dyalitique que l'on va employer ici est beaucoup plus générique que les 'bidouillages' précédents.

Par contre elle fait appel à des mathématiques beaucoup plus avancées, et on ne présentera ici que la trame de résolution.

Pour ceux qui sont intéressés par la traduction de ces concepts en langage matlab :

-Les calculs détaillés des expressions sont disponibles dans le programme matlab *dyalitic_2angles_symbolic.m*

-L'utilisation de ces expressions pour trouver les solutions est disponible dans la fonction matlab *solve_type_g.m*

méthode de résolution

étape 1 : On remplace les *cos* et *sin* des angles par leurs expressions en fonction des tangentes des angles moitié: *t1=tan(Ai/2) , t2=tan(Ak/2), en posant :*

cos(Ai)= (1-t1^2)/(1+t1^2) , *sin(Ai)*= 2.t1/(1+t1^2) , *cos(Ak)*= (1-t2^2)/(1+t2^2) , *sin(Ak)*= 2.t2/(1+t2^2)

étape 2 : on fait apparaître des équations polynomiales en t1, t2 en multipliant par (1+t1^2).(1+t2^2)

Le problème se ramène à trouver tous les couples de solutions réelles (t1,t2) des 2 équations

étape 3 : On se débrouille à mettre les équations sous la forme d'un problème linéaire

$$\underbrace{M(t1)}_{\text{matrice 2X3 dépendant de t1}} \quad . \quad \underbrace{\begin{bmatrix} 1 \\ t2 \\ t2^2 \end{bmatrix}}_{X=3 \text{ inconnues à déterminer}} = \underbrace{\begin{bmatrix} 0 \\ 0 \end{bmatrix}}_{\text{équations [1] et [2]}}$$

il y a plus d'inconnues que d'équations, on aura donc une <u>infinité de solutions</u> à ce problème

étape 4 : En multipliant les équations par *t2*, on fait apparaître 2 équations et une inconnue (t2^3) supplémentaire.

On fait donc apparaître un problème à 4 équations, et 4 inconnues :

$$\underbrace{H(t1)}_{\text{matrice 4X4 dépendant de t1}} \quad . \quad \underbrace{\begin{bmatrix} 1 \\ t2 \\ t2^2 \\ t2^3 \end{bmatrix}}_{X=4 \text{ inconnues à déterminer}} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{4 \text{ équations}}$$

étape5: Il y a des <u>solutions *X* non nulles</u> à ces équations uniquement si le <u>déterminant de *H* est égal à 0</u>.

Or le déterminant de *H* est un polynôme de degré 8 en *t1*, le problème se ramène donc à <u>trouver les 8 racines *t1n* du polynôme *det(H)*</u>

Le programme matlab symbolique <u>dyalitic_2angles_symbolic.m</u> crée les fonctions calculant M,H,det(H).

La fonction numérique solve_type_g.m résout les équations de la façon suivante

étape1: On détermine les racines *t1n* de l'équation *det(H)* =0, et on ne retient que les racines réelles

étape2: pour chacune des racines trouvées, on calcule le vecteur propre *Xn* de *H(t1n)* associé à la valeur propre 0, et on <u>normalise sa première composante à 1</u>.(on emploie pour cela la décomposition en valeurs singulières de *H(t1n)*, mieux conditionnée)

étape3: On vérifie que le vecteur *Xn* peut bien s'écrire sous la forme $X_n=\begin{bmatrix} 1 \\ t2n \\ t2n^2 \\ t2n^3 \end{bmatrix}$, en trouvant la meilleure solution *t2n* au sens des moindres carrés de l'équation $X_n(2:4)=X_n(1:3).t2n$.

Si l'erreur $X_n(2:4)-X_n(1:3).t2n$ est assez petite ,on retient le couple de solutions (t1n,t2n), dont on déduit le couple de solutions angulaires (Ai,Ak) correspondant