

MSLIB Fortran 90

CS

Nomenclature : **M-MU-0-116-CIS**

Edition : 05 Date: 27/05/2004

Révision: 01 Date: 15/02/2005

Volume U

Utilitaires mathématiques

Rédigé par : Guylaine PRAT	le : CS (SI/Espace/FDS)	
Validé par : Guylaine PRAT Anne MAZZIETTI-ERSA (ingénieur qualité)	le : CS (SI/Espace/FDS) CS (SI/Espace)	
Pour application : Franck REINQUIN Hervé MADIEU	le : CNES (DCT/SB/OI)	

DIFFUSION INTERNE CNES**Observations**

Voir la note nomenclaturée M-NT-0-18-CN:
"Liste de diffusion de la documentation utilisateur MSLIB".

DIFFUSION EXTERNE CNES**Observations**

Voir la note nomenclaturée M-NT-0-18-CN:
"Liste de diffusion de la documentation utilisateur MSLIB".

BORDEREAU D'INDEXATION**CONFIDENTIALITE :** NC**MOTS-CLES :****TITRE :** Volume U - Utilitaires mathématiques**AUTEUR :** Guylaine PRAT**RESUME :**

Ce document rassemble les notices d'utilisation des routines du thème "Utilitaires mathématiques".

SITUATION DU DOCUMENT : Création**VOLUME :****PAGES :** 45**PLANCHES :****FIGURES :****LANGUES :** F**CONTRAT :** Marché 779/Cnes/2001/8929 BC4500009860**SYSTEME HOTE :** Frame6/MSLIB

MODIFICATION

ETAT DOCUMENT				PAGES REVISEES	
ED.	REV.	DATE	REFERENCE ORIGINE (pour chaque édition)	ETAT PAGE *	NUMERO DES PAGES
01	00	22/04/98	M-MU-0-116-CIS	M	Création
02	00	26/06/98	M-MU-0-116-CIS Rédacteur: V. Lépine avec la participation de G. Prat		Modification de toutes les pages.
03	00	19/10/00	M-MU-0-116-CIS Rédacteur: V. Lépine avec la participation de G. Prat	I	Ajout des pages liées aux nouvelles routines pour la MSLIB90 V3.0: mu_axe_angle_quat, mu_prod_quat, mu_quat_axe_angle, mu_quat_conjug, mu_quat_norme, mu_quat_rep
04	00	05/12/03	M-MU-0-116-CIS Rédacteur: V. Lépine avec la participation de G. Prat	I	Ajout des pages liées aux nouvelles routines pour la MSLIB90 V5.0
05	00	27/05/04	M-MU-0-116-CIS Rédacteur: V. Lépine avec la participation de G. Prat	I	Création de l'introduction du thème
05	01	15/02/05	M-MU-0-116-CIS Rédacteur: G. Prat	I	Ajout des pages liées à la nouvelle routine pour la MSLIB90 V6.0
				M	Correction exemple de mu_mat_quat

* I = Inséré

S = Supprimé

M = Modifié

Sommaire

Présentation du thème U :	<i>page 1</i>
Notations	<i>page 2</i>
Index.	<i>page 3</i>
1 Rotations	<i>page 4</i>
1.1 Définition du quaternion pour la MSLIB	<i>page 4</i>
1.2 Angles de Cardan et d'Euler	<i>page 4</i>
1.2.1 Angles de Cardan	<i>page 4</i>
1.2.2 Angles d'Euler	<i>page 5</i>
2 Documents de référence.	<i>page 6</i>

Liste des routines du thème U : *voir pages suivantes du sommaire.*

Liste des routines du thème U:

mu_3rot_quat :	<i>page 7</i>
“Calcul du quaternion associé à une rotation définie par 3 angles de Cardan ou d’Euler.”	
mu_angle2 :	<i>page 10</i>
“Dans le plan IR2, calcul de l’angle entre un vecteur et l’axe des abscisses.”	
mu_angle3 :	<i>page 12</i>
“Dans l’espace IR3, calcul de l’angle non orienté de deux vecteurs.”	
mu_axe_angle_quat :	<i>page 14</i>
“Conversion d’une rotation définie par son axe et son angle en un quaternion.”	
mu_compar_rot_quat :	<i>page 16</i>
“Comparaison de rotations définies à l’aide de quaternions.”	
mu_mat_quat :	<i>page 18</i>
“Calcul du quaternion associé à une matrice de rotation.”	
mu_norme :	<i>page 20</i>
“Calcul de la norme euclidienne d’un vecteur dans IR3 .”	
mu_prod_quat:	<i>page 22</i>
“Calcul du produit de deux quaternions.”	
mu_prod_vect :	<i>page 24</i>
“Calcul du produit vectoriel de deux vecteurs dans IR3 .”	
mu_quat_3rot :	<i>page 26</i>
“Calcul des 3 angles de Cardan ou d’Euler associés à une rotation définie par un quaternion.”	
mu_quat_axe_angle :	<i>page 29</i>
“Conversion d’un quaternion en une rotation définie par son axe et son angle.”	
mu_quat_conjug :	<i>page 31</i>
“Calcul du quaternion conjugué d’un quaternion donné.”	

mu_quat_mat :	<i>page 33</i>
“Calcul de la matrice de rotation associée à un quaternion.”	
mu_quat_norme :	<i>page 35</i>
“Normalisation d’un quaternion après calcul de sa norme.”	
mu_quat_rep :	<i>page 37</i>
“A l’aide d’un quaternion, calcul de changement de repère.”	

Présentation du thème U

Le thème “*Utilitaires mathématiques*” regroupe un ensemble de routines mathématiques.

Les routines de ce thème n’ont pas la prétention de constituer une bibliothèque mathématique générale. Leur but est de rendre la MSLIB autonome en terme de routines de type mathématiques par rapport à d’autres bibliothèques existantes (de type commerciales ou freewares). Leur présence dans la bibliothèque ne se justifie que par la notion d’utilsitaires pour les routines appartenant aux autres thèmes.

Par convention:

les angles utilisés dans les routines sont toujours exprimés en *radians*.

Notations

(O, \vec{i}, \vec{j})	repère orthonormé du plan IR^2 d'origine O
$\tilde{q} = (q_0, \vec{q})$	quaternion
$(\alpha_1, \alpha_2, \alpha_3)$	angles de Cardan ou d'Euler

Index

A

angles d'Euler **5**
angles de Cardan **4**

Q

quaternion **4**

1 Rotations

Les rotations en dimension 3 peuvent se représenter de différentes manières, par exemple: soit par des quaternions, soit par des angles de Cardan ou d'Euler, soit par des matrices de rotation. Il est donc nécessaire de préciser certaines de ces définitions et certaines conventions retenues par la MSLIB.

1.1 Définition du quaternion pour la MSLIB

Pour des définitions mathématiques liées aux quaternions se reporter à la note algorithmique: "*Algorithmes des routines du thème "Utilitaires mathématiques" de la MSLIB; G. Prat, avec la participation de L. Maisonobe (CS SI); référence MSLIB: M-NT-0-96-CIS*".

Plusieurs conventions existent pour la définition d'un **quaternion** d'axe \vec{u} et d'angle θ . La MSLIB a retenu la définition suivante:

$$\begin{cases} q_0 = \cos \frac{\theta}{2} \\ \vec{q} = \sin \frac{\theta}{2} \cdot \vec{u} \end{cases}$$

Cette définition peut différer selon les auteurs: voir par exemple [DR1] ou [DR2] où les formules font intervenir l'angle " $-\theta$ ".

1.2 Angles de Cardan et d'Euler

Une rotation quelconque en dimension 3 peut être définie par la composition de trois rotations successives autour des axes canoniques (pour plus de détails se reporter à [DR1] ou [DR2]).

Notation:

nous noterons dans les paragraphes suivant la rotation YXZ telle que la première rotation est réalisée autour de Y, la deuxième rotation autour de X', et la troisième rotation autour de Z".

1.2.1 Angles de Cardan

Lorsque les trois rotations s'effectuent autour des 3 axes canoniques, les angles associés s'appellent **angles de Cardan**.

Il existe alors 6 possibilités pour les angles de Cardan: XYZ, XZY, YXZ, YZX, ZXY, ZYX.

Il est à noter que si les angles $(\alpha_1, \alpha_2, \alpha_3)$ sont associés à une rotation définie par des angles de Cardan, alors on peut lui associer également les angles $(\pi + \alpha_1, \pi - \alpha_2, \pi + \alpha_3)$.

Cette propriété permet de restreindre le domaine de définition de α_2 à l'intervalle $[-\frac{\pi}{2}, \frac{\pi}{2}]$, si besoin.

1.2.2 Angles d'Euler

Lorsque la première et la dernière rotation s'effectuent autour d'un même axe canonique, et que la deuxième rotation s'effectue autour d'un autre axe canonique, les angles associés s'appellent ***angles d'Euler***.

Il existe alors 6 possibilités pour les angles d'Euler: XYX , XZX , YXY , YZY , ZXZ , ZYZ .

Il est à noter que si les angles $(\alpha_1, \alpha_2, \alpha_3)$ sont associés à une rotation définie par des angles d'Euler, alors on peut lui associer également les angles $(\pi + \alpha_1, 2\pi - \alpha_2, \pi + \alpha_3)$.

Cette propriété permet de restreindre le domaine de définition de α_2 à l'intervalle $[0, \pi]$, si besoin.

2 Documents de référence

DR1

L. MAISONOBE & G. FILAIRE: « MARMOTTES - documentation mathématiques »,
DTS/MPI/MS/AM/99-118; éd. 03 rév. 07

DR2

L. MAISONOBE : « Principes de simulation générique d'attitude »,
Note technique N°3 du Centre de Compétence Technique "Mécanique Orbitale".
Note technique CNES N°142 - Février 2001.

Routine mu_3rot_quat

Identification

“Calcul du **quat**ernion associé à une **rot**ation définie par **3** angles de Cardan ou d’Euler.”

Rôle

Calcul du quaternion associé à une rotation définie par 3 angles de Cardan ou d’Euler

Séquence d’appel

(voir explications dans le volume 3)

call mu_3rot_quat (def_3rot, angle1, angle2, angle3, quat, code_retour)

Description des arguments

(voir explications dans le volume 3)

• Entrées obligatoires

entier	def_3rot	Définition et ordre d’enchaînement des trois rotations
pm_reel	angle1	Valeur de l’angle α_1 associé à la première rotation (rad)
pm_reel	angle2	Valeur de l’angle α_2 associé à la deuxième rotation (rad)
pm_reel	angle3	Valeur de l’angle α_3 associé à la troisième rotation (rad)

• Sorties obligatoires

tm_quat	quat	quaternion
tm_code_retour	code_retour	

Conditions sur les arguments

• **def_3rot** correspond à un entier qui est défini par un paramètre associé à une rotation. Par exemple, la rotation XYZ (1^{ère} rotation autour de X, 2^{ième} rotation autour de Y, 3^{ième} rotation autour de Z) est associée au paramètre pm_1x_2y_3z donc **def_3rot** sera initialisée avec le paramètre pm_1x_2y_3z.

• Il existe 6 possibilités pour les angles de Cardan, et 6 possibilités pour les angles d’Euler:

Valeurs possibles de def_3rot	
Angles de Cardan	Angles d'Euler
pm_1x_2y_3z	pm_1x_2y_3x
pm_1x_2z_3y	pm_1x_2z_3x
pm_1y_2x_3z	pm_1y_2x_3y
pm_1y_2z_3x	pm_1y_2z_3y
pm_1z_2x_3y	pm_1z_2x_3z
pm_1z_2y_3x	pm_1z_2y_3z

Notes d'utilisation

- La transformation inverse peut s'effectuer à l'aide de la routine **mu_quat_3rot**.

Références documentaires

- Algorithmes des routines du thème "Utilitaires mathématiques" de la MSLIB; G. Prat, avec la participation de L. Maisonobe (CS SI); référence MSLIB: M-NT-0-96-CIS.

Code retour (voir explications dans le volume 3)

pm_OK (0) : Retour normal.

pm_err_clef_rot (-1810) : La clef de rotation ne correspond pas à une rotation de Cardan ou d'Euler.

Exemple en Fortran 90 portable

(voir explications dans le volume 3)

```
program MATH

  use mslib

  integer                :: DEF_3ROT
  real(pm_reel)          :: ANGLE1, ANGLE2, ANGLE3
  type(tm_quat)          :: QUAT
  type(tm_code_retour)   :: CODE_RETOUR

  DEF_3ROT = pm_1z_2y_3x
  ANGLE1   = 21._pm_reel*pm_deg_rad
  ANGLE2   = 79._pm_reel*pm_deg_rad
  ANGLE3   =  7._pm_reel*pm_deg_rad

  call mu_3rot_quat ( DEF_3ROT, ANGLE1, ANGLE2, ANGLE3, QUAT,
                     CODE_RETOUR )

  ! appel a la routine utilisateur d'ecriture des resultats
  call WRITE_RESULTATS (QUAT, CODE_RETOUR)

end program MATH
```

Résultats attendus:

```
QUAT%q0      = 0.764
QUAT%q123(1) = -0.0694
QUAT%q123(2) = 0.633
QUAT%q123(3) = 0.102
```

```
CODE_RETOUR%valeur  = 0
CODE_RETOUR%routine = 1145
```


Routine mu_angle2

Identification

“Dans le plan IR^2 , calcul de l’**angle** entre un vecteur et l’axe des abscisses.”

Rôle

Dans un repère orthonormé (O, \vec{i}, \vec{j}) , pour un vecteur $\vec{a}(x, y)$, calcul de l’angle $\theta = (\vec{i}, \vec{a}) \in [0, 2\pi[$.

Nota: x et y représentent les cosinus et sinus de l’angle.

Séquence d’appel

(voir explications dans le volume 3)

call mu_angle2 (x , y , angle , code_retour)

Description des arguments

(voir explications dans le volume 3)

• Entrées obligatoires

pm_reel	x	coordonnée x du vecteur \vec{a} .
pm_reel	y	coordonnée y du vecteur \vec{a} .

• Sorties obligatoires

pm_reel	angle	angle $\theta = (\vec{i}, \vec{a}) \in [0, 2\pi[$
tm_code_retour	code_retour	

Conditions sur les arguments

Sans objet.

Notes d’utilisation

Sans objet.

Références documentaires

- Algorithmes des routines du thème "Utilitaires mathématiques" de la MSLIB; G. Prat, avec la participation de L. Maisonobe (CS SI); référence MSLIB: M-NT-0-96-CIS.

Code retour *(voir explications dans le volume 3)*

pm_OK (0) : Retour normal.

pm_err_vect_nul (-2001) : Vecteur nul .

Exemple en Fortran 90 portable *(voir explications dans le volume 3)*

```
program MATH

  use mslib

  real(pm_reel)          :: X, Y, ANGLE
  type(tm_code_retour)   :: CODE_RETOUR

  X = 0._pm_reel
  Y = 5._pm_reel

  call mu_angle2 ( X, Y, ANGLE, CODE_RETOUR )

  ! appel a la routine utilisateur d'ecriture des resultats
  call WRITE_RESULTATS (ANGLE, CODE_RETOUR)

end program MATH
```

Résultats attendus:

ANGLE = .157 10⁺¹

CODE_RETOUR%valeur = 0

CODE_RETOUR%routine = 1004

Routine *mu_angle3*

Identification

“Dans l’espace IR^3 , calcul de l’**angle** non orienté de deux vecteurs.”

Rôle

Calcul de l’angle non orienté θ de deux vecteurs \vec{a} et \vec{b} de IR^3 . L’angle calculé est exprimé en radians, et $\theta \in [0, \pi]$.

Séquence d’appel

(voir explications dans le volume 3)

call *mu_angle3* (vect_a, vect_b, angle, code_retour)

Description des arguments

(voir explications dans le volume 3)

• Entrées obligatoires

pm_reel(3)	vect_a	vecteur \vec{a}
pm_reel(3)	vect_b	vecteur \vec{b}

• Sorties obligatoires

pm_reel	angle	angle θ non orienté entre les vecteurs \vec{a} et \vec{b}
tm_code_retour	code_retour	

Conditions sur les arguments

Sans objet.

Notes d’utilisation

Sans objet.

Références documentaires

- Algorithmes des routines du thème "Utilitaires mathématiques" de la MSLIB; G. Prat, avec la participation de L. Maisonobe (CS SI); référence MSLIB: M-NT-0-96-CIS.

Code retour (voir explications dans le volume 3)

pm_OK (0) : Retour normal.

pm_err_vect_nul (-2001) : Vecteur nul .

Exemple en Fortran 90 portable (voir explications dans le volume 3)

```
program MATH

  use mslib

  real(pm_reel), dimension(3)      :: VECT_A, VECT_B
  real(pm_reel)                    :: ANGLE
  type(tm_code_retour)              :: CODE_RETOUR

  VECT_A(:) = 1._pm_reel
  VECT_B(:) = -1._pm_reel

  call mu_angle3 ( VECT_A, VECT_B, ANGLE, CODE_RETOUR)

  ! appel a la routine utilisateur d'ecriture des resultats
  call WRITE_RESULTATS (ANGLE, CODE_RETOUR)

end program MATH
```

Résultats attendus:

ANGLE = 3.14

CODE_RETOUR%valeur = 0

CODE_RETOUR%routine = 1009

Routine mu_axe_angle_quat

Identification

“Conversion d’une rotation définie par son **axe** et son **angle** en un **quaternion**.”

Rôle

Calcul du quaternion normé associé à une rotation définie par son axe et son angle de rotation.

Séquence d’appel

(voir explications dans le volume 3)

call mu_axe_angle_quat (axe, angle, quat, code_retour)

Description des arguments

(voir explications dans le volume 3)

• Entrées obligatoires

pm_reel(3)	axe	axe de rotation.
pm_reel	angle	angle de rotation (rad).

• Sorties obligatoires

tm_quat	quat	quaternion (normé) associé à la rotation.
tm_code_retour	code_retour	

Conditions sur les arguments

- La norme de l’axe de rotation doit être strictement positive.

Notes d’utilisation

- Si l’angle n’appartient pas à $[0, 2\pi[$, alors il est ramené dans cet intervalle à l’aide d’un modulo 2π .
- La transformation inverse (quaternion \rightarrow rotation) peut être effectuée par la routine **mu_quat_axe_angle**

Références documentaires

- Algorithmes des routines du thème "Utilitaires mathématiques" de la MSLIB; G. Prat, avec la participation de L. Maisonobe (CS SI); référence MSLIB: M-NT-0-96-CIS.

Code retour

(voir explications dans le volume 3)

pm_OK (0) : Retour normal.

pm_err_axe_rot_nul (-2004) : La norme de l'axe de rotation est nulle.

Exemple en Fortran 90 portable

(voir explications dans le volume 3)

```
program MATH

  use mslib

  real(pm_reel), dimension(3) :: AXE
  real(pm_reel)               :: ANGLE
  type(tm_quat)               :: QUAT
  type(tm_code_retour)        :: CODE_RETOUTOUR

  AXE(1) = 1._pm_reel
  AXE(2) = 0._pm_reel
  AXE(3) = 2._pm_reel
  ANGLE  = 1._pm_reel

  call mu_axe_angle_quat ( AXE, ANGLE, QUAT, CODE_RETOUTOUR )

  ! appel a la routine utilisateur d'ecriture des resultats
  call WRITE_RESULTATS (QUAT, CODE_RETOUTOUR)

end program MATH
```

Résultats attendus:

```
QUAT%q0      = .878
QUAT%q123(1)  = .214
QUAT%q123(2)  = 0.
QUAT%q123(3)  = .429
```

```
CODE_RETOUTOUR%valeur = 0
CODE_RETOUTOUR%routine = 1123
```

Routine mu_compar_rot_quat

Identification

“**Compar**aison de **rot**ations définies à l’aide de **quat**ernions.”

Rôle

Calcul de l’écart angulaire entre deux rotations, où les rotations sont définies par un quaternion.

Séquence d’appel

(voir explications dans le volume 3)

call mu_compar_rot_quat (quat1, quat2, angle, code_retour)

Description des arguments

(voir explications dans le volume 3)

• Entrées obligatoires

tm_quat	quat1	quaternion \tilde{q}_1 associé à la 1 ^{ère} rotation
tm_quat	quat2	quaternion \tilde{q}_2 associé à la 2 ^{ième} rotation

• Sorties obligatoires

pm_reel	angle	écart angulaire $\in [0, \pi]$ entre les rotations définies par \tilde{q}_1 et \tilde{q}_2 (rad)
tm_code_retour	code_retour	

Conditions sur les arguments

- La norme des quaternions doit être non nulle.

Notes d’utilisation

- Si l’angle appartient à $[\pi, 2\pi[$, il est ramené à son complément à 2π .

Références documentaires

- Algorithmes des routines du thème "Utilitaires mathématiques" de la MSLIB; G. Prat, avec la participation de L. Maisonobe (CS SI); référence MSLIB: M-NT-0-96-CIS.

Code retour (voir explications dans le volume 3)

pm_OK (0) : Retour normal.
pm_err_quat_nul (-2005) : La norme du quaternion est nulle.

Exemple de résultats

Pour la description des types dérivés, se reporter à la la documentation utilisateur
MSLIB Fortran 90 :

- *Structuration des données pour la MSLIB Fortran 90* du volume 3 “Caractéristiques principales et conventions d’utilisation de la MSLIB Fortran 90; M-MU-0-103-CIS“

Entrées:

```
quat1%q0      = 2._pm_reel  
quat1%q123(1) = 1._pm_reel  
quat1%q123(2) = 0._pm_reel  
quat1%q123(3) = -1._pm_reel
```

```
quat2%q0      = 3._pm_reel  
quat2%q123(1) = 0._pm_reel  
quat2%q123(2) = -2._pm_reel  
quat2%q123(3) = -1._pm_reel
```

Résultats attendus:

angle = 1.403

code_retour%valeur = 0

Routine mu_mat_quat

Identification

“Calcul du **quat**ernion associé à une **mat**rice de rotation.”

Rôle

A partir d’une matrice de rotation, calcul du quaternion (normé) associé.

Séquence d’appel

(voir explications dans le volume 3)

call mu_mat_quat (mat, quat, code_retour)

Description des arguments

(voir explications dans le volume 3)

- Entrées obligatoires

pm_reel(3,3) **mat** matrice de rotation M

- Sorties obligatoires

tm_quat **quat** quaternion \tilde{q} correspondant à la matrice de rotation M
tm_code_retour **code_retour**

Conditions sur les arguments

- La matrice M est une matrice de rotation, et doit donc à ce titre être orthonormale et de déterminant = +1.

Notes d’utilisation

- La transformation inverse peut s’effectuer à l’aide de la routine **mu_quat_mat..**

Références documentaires

- Algorithmes des routines du thème "Utilitaires mathématiques" de la MSLIB; G. Prat, avec la participation de L. Maisonobe (CS SI); référence MSLIB: M-NT-0-96-CIS.

Code retour (voir explications dans le volume 3)

pm_OK (0) : Retour normal.
pm_err_mat_non_rot (-2007) : La matrice n'est pas une matrice de rotation.

Exemple en Fortran 90 portable (voir explications dans le volume 3)

```
program MATH

  use mslib

  real(pm_reel)                :: ANGLE
  real(pm_reel), dimension(3,3) :: MAT
  type(tm_quat)                :: QUAT
  type(tm_code_retour)         :: CODE_RETOUR

  intrinsic cos, sin

  ANGLE      = pm_pi*0.33_pm_reel
  MAT(:, :)  = 0._pm_reel
  MAT(1,1)   = cos(ANGLE)
  MAT(1,2)   = sin(ANGLE)
  MAT(2,1)   = -sin(ANGLE)
  MAT(2,2)   = cos(ANGLE)
  MAT(3,3)   = 1._pm_reel

  call mu_mat_quat ( MAT, QUAT, CODE_RETOUR )

  ! appel a la routine utilisateur d'ecriture des resultats
  call WRITE_RESULTATS (QUAT, CODE_RETOUR)

end program MATH
```

Résultats attendus:

```
QUAT%q0      = 0.869
QUAT%q123(1) = 0.
QUAT%q123(2) = 0.
QUAT%q123(3) = 0.495

CODE_RETOUR%valeur = 0
CODE_RETOUR%routine = 1149
```

Routine mu_norme

Identification

“Calcul de la **norme** euclidienne d’un vecteur dans IR^3 .”

Rôle

Calcul de la norme euclidienne d’un vecteur \vec{a} de IR^3 et, sur option, du vecteur normé associé.

Séquence d’appel

(voir explications dans le volume 3)

call mu_norme (vect, norme, code_retour [, vect_norme])

Description des arguments

(voir explications dans le volume 3)

- Entrées obligatoires

pm_reel(3) **vect** vecteur \vec{a}

- Sorties obligatoires

pm_reel **norme** norme de \vec{a} : $\|\vec{a}\|$

tm_code_retour **code_retour**

- Sorties facultatives

pm_reel(3) [**vect_norme**] vecteur normé : $\frac{\vec{a}}{\|\vec{a}\|}$

Conditions sur les arguments

Sans objet.

Notes d’utilisation

Sans objet.

Références documentaires

- Algorithmes des routines du thème "Utilitaires mathématiques" de la MSLIB; G. Prat, avec la participation de L. Maisonobe (CS SI); référence MSLIB: M-NT-0-96-CIS.

Code retour (voir explications dans le volume 3)

pm_OK (0) : Retour normal.
pm_err_vect_nul (-2001) : Vecteur nul .

Exemple en Fortran 90 portable (voir explications dans le volume 3)

```
program MATH

  use mslib
  real(pm_reel), dimension(3)      :: VECT      &
    =(/-2.30770288_pm_reel,      &
      -1.723990551_pm_reel,      &
      -.83824649_pm_reel/)      &
  real(pm_reel), dimension(3)      :: VECT_NORME
  real(pm_reel)                    :: NORME
  type(tm_code_retour)              :: CODE_RETOUR

  call mu_norme ( VECT, NORME, CODE_RETOUR,      &
    vect_norme = VECT_NORME )

  ! appel a la routine utilisateur d'ecriture des resultats
  call WRITE_RESULTATS (NORME, VECT_NORME, CODE_RETOUR)

end program MATH
```

Résultats attendus:

NORME = 3.

VECT_NORME= $\begin{bmatrix} -0.769 \\ -0.575 \\ -0.279 \end{bmatrix}$

CODE_RETOUR%valeur = 0
CODE_RETOUR%routine = 1011

Routine mu_prod_quat

Identification

“Calcul du **produit** de deux **quaternions**.”

Rôle

Calcul du produit de deux quaternions.

Séquence d'appel

(voir explications dans le volume 3)

call mu_prod_quat (quat1, quat2, quat_prod, code_retour)

Description des arguments

(voir explications dans le volume 3)

• Entrées obligatoires

tm_quat	quat1	quaternion \tilde{q}_1
---------	--------------	--------------------------

tm_quat	quat2	quaternion \tilde{q}_2
---------	--------------	--------------------------

• Sorties obligatoires

tm_quat	quat_prod	quaternion produit: $\tilde{q}_3 = \tilde{q}_1 * \tilde{q}_2$
---------	------------------	---

tm_code_retour	code_retour	
----------------	--------------------	--

Conditions sur les arguments

Sans objet.

Notes d'utilisation

Sans objet.

Références documentaires

- Algorithmes des routines du thème "Utilitaires mathématiques" de la MSLIB; G. Prat, avec la participation de L. Maisonobe (CS SI); référence MSLIB: M-NT-0-96-CIS.

Code retour (voir explications dans le volume 3)

pm_OK (0) : Retour normal.

Exemple en Fortran 90 portable (voir explications dans le volume 3)

```
program MATH

  use mslib

  type(tm_quat)                :: QUAT1, QUAT2
  type(tm_quat)                :: QUAT_PROD
  type(tm_code_retour)         :: CODE_RETOUR

  QUAT1%q0      =  2._pm_reel
  QUAT1%q123(1) =  1._pm_reel
  QUAT1%q123(2) =  0._pm_reel
  QUAT1%q123(3) = -1._pm_reel

  QUAT2%q0      =  3._pm_reel
  QUAT2%q123(1) =  0._pm_reel
  QUAT2%q123(2) = -2._pm_reel
  QUAT2%q123(3) = -1._pm_reel

  call mu_prod_quat (QUAT1, QUAT2, QUAT_PROD, CODE_RETOUR)

  ! appel a la routine utilisateur d'ecriture des resultats
  call WRITE_RESULTATS (QUAT_PROD, CODE_RETOUR)

end program MATH
```

Résultats attendus:

```
QUAT_PROD%q0      = 5.
QUAT_PROD%q123(1) = 1.
QUAT_PROD%q123(2) = -3.
QUAT_PROD%q123(3) = -7.
```

```
CODE_RETOUR%valeur = 0
CODE_RETOUR%routine = 1127
```

Routine *mu_prod_vect*

Identification

“Calcul du **produit vectoriel** de deux vecteurs dans IR^3 .”

Rôle

Dans un repère orthonormé, calcul du produit vectoriel \vec{c} de deux vecteurs \vec{a} et \vec{b} .

Séquence d'appel

(voir explications dans le volume 3)

call *mu_prod_vect* (vect_a, vect_b, vect_c, code_retour)

Description des arguments

(voir explications dans le volume 3)

• Entrées obligatoires

pm_reel(3) **vect_a** vecteur \vec{a}

pm_reel(3) **vect_b** vecteur \vec{b}

• Sorties obligatoires

pm_reel(3) **vect_c** produit vectoriel $\vec{c} = \vec{a} \wedge \vec{b}$

tm_code_retour **code_retour**

Conditions sur les arguments

Sans objet.

Notes d'utilisation

Sans objet.

Références documentaires

- Algorithmes des routines du thème "Utilitaires mathématiques" de la MSLIB; G. Prat, avec la participation de L. Maisonobe (CS SI); référence MSLIB: M-NT-0-96-CIS.

Code retour (voir explications dans le volume 3)

pm_OK (0) : Retour normal.

Exemple en Fortran 90 portable (voir explications dans le volume 3)

```
program MATH

  use mslib
  real(pm_reel), dimension(3)      :: VECT_A           &
      =( /1._pm_reel, 0._pm_reel, 0._pm_reel/ )
  real(pm_reel), dimension(3)      :: VECT_B           &
      =( /0._pm_reel, 0._pm_reel, 1._pm_reel/ )
  type(tm_code_retour)             :: CODE_RETOUR

  call mu_prod_vect ( VECT_A, VECT_B, VECT_C, CODE_RETOUR )

  ! appel a la routine utilisateur d'ecriture des resultats
  call WRITE_RESULTATS (VECT_C, CODE_RETOUR)

end program MATH
```

Résultats attendus:

$$\text{VECT_C} = \begin{bmatrix} 0. \\ -1. \\ 0. \end{bmatrix}$$

CODE_RETOUR%valeur = 0
CODE_RETOUR%routine = 1010

Routine *mu_quat_3rot*

Identification

“Calcul des **3** angles de Cardan ou d’Euler associés à une **rotation** définie par un **quaternion**.”

Rôle

Calcul des 3 angles de Cardan ou d’Euler associés à une rotation définie par un quaternion.

Séquence d’appel

(voir explications dans le volume 3)

call *mu_quat_3rot* (*def_3rot*, *quat*, *angle1*, *angle2*, *angle3*, *code_retour*)

Description des arguments

(voir explications dans le volume 3)

• Entrées obligatoires

entier	def_3rot	Définition et ordre d’enchaînement des trois rotations
tm_quat	quat	quaternion

• Sorties obligatoires

pm_reel	angle1	Valeur de l’angle α_1 associé à la première rotation (rad)
pm_reel	angle2	Valeur de l’angle α_2 associé à la deuxième rotation (rad)
pm_reel	angle3	Valeur de l’angle α_3 associé à la troisième rotation (rad)
tm_code_retour	code_retour	

Conditions sur les arguments

• **def_3rot** correspond à un entier qui est défini par un paramètre associé à une rotation. Par exemple, la rotation XYZ (1^{ère} rotation autour de X, 2^{ième} rotation autour de Y, 3^{ième} rotation autour de Z) est associée au paramètre *pm_1x_2y_3z* donc **def_3rot** sera initialisée avec le paramètre *pm_1x_2y_3z*.

• Il existe 6 possibilités pour les angles de Cardan, et 6 possibilités pour les angles d’Euler :

Valeurs possibles de def_3rot	
Angles de Cardan	Angles d'Euler
pm_1x_2y_3z	pm_1x_2y_3x
pm_1x_2z_3y	pm_1x_2z_3x
pm_1y_2x_3z	pm_1y_2x_3y
pm_1y_2z_3x	pm_1y_2z_3y
pm_1z_2x_3y	pm_1z_2x_3z
pm_1z_2y_3x	pm_1z_2y_3z

Notes d'utilisation

- Pour les angles de Cardan et d'Euler, les angles α_1 et α_3 sont donnés dans l'intervalle $[0, 2\pi[$.

Pour les angles de Cardan, l'angle α_2 est donné dans l'intervalle $[-\frac{\pi}{2}, \frac{\pi}{2}]$.

Pour les angles d'Euler, l'angle α_2 est donné dans l'intervalle $[0, \pi]$.

Pour plus d'explications: se reporter à l'introduction du thème, au paragraphe 1. *Rotations*

- La transformation inverse peut s'effectuer à l'aide de la routine **mu_3rot_quat**

Références documentaires

- Algorithmes des routines du thème "Utilitaires mathématiques" de la MSLIB; G. Prat, avec la participation de L. Maisonobe (CS SI); référence MSLIB: M-NT-0-96-CIS.

Code retour (voir explications dans le volume 3)

pm_OK (0) : Retour normal.

pm_warn_angle1_ou_3_indef (+2001) : Infinité de solutions pour le premier et le troisième angle. Arbitrairement nous donnons la valeur 0 au premier angle pour le cas de Cardan ou au troisième angle pour le cas d'Euler.

pm_err_clef_rot (-1810) : La clef de rotation ne correspond pas à une rotation de Cardan ou d'Euler.

pm_err_quat_nul (-2005) : La norme du quaternion est nulle.

Exemple en Fortran 90 portable

(voir explications dans le volume 3)

```
program MATH

  use mslib

  integer          :: DEF_3ROT
  type(tm_quat)    :: QUAT
  real(pm_reel)    :: ANGLE_SUR_2, ANGLE1, ANGLE2, ANGLE3
  type(tm_code_retour) :: CODE_RETOUR

  intrinsic cos, sin

  DEF_3ROT      = pm_1x_2y_3z
  ANGLE_SUR_2   = 20._pm_reel*pm_deg_rad
  QUAT%q0       = cos(ANGLE_SUR_2)
  QUAT%q123(1)  = sin(ANGLE_SUR_2)*0.3_pm_reel
  QUAT%q123(2)  = sin(ANGLE_SUR_2)*0.3_pm_reel
  QUAT%q123(3)  = sin(ANGLE_SUR_2)*0.3_pm_reel

  call mu_quat_3rot ( DEF_3ROT, QUAT, ANGLE1, ANGLE2, ANGLE3,
                     CODE_RETOUR )

  ! appel a la routine utilisateur d'ecriture des resultats
  call WRITE_RESULTATS (ANGLE1, ANGLE2, ANGLE3, CODE_RETOUR)

end program MATH
```

Résultats attendus:

ANGLE1 = 0.194
ANGLE2 = 0.236
ANGLE3 = 0.194

CODE_RETOUR%valeur = 0
CODE_RETOUR%routine = 1147

Routine mu_quat_axe_angle

Identification

“Conversion d’un **quat**ernion en une rotation définie par son **axe** et son **angle**.”

Rôle

Conversion d’un quaternion en une rotation définie par son axe et son angle (dans $[0, 2\pi[$).

Séquence d’appel

(voir explications dans le volume 3)

call mu_quat_axe_angle (quat, axe, angle, code_retour)

Description des arguments

(voir explications dans le volume 3)

- Entrées obligatoires

tm_quat	quat	quaternion.
---------	-------------	-------------

- Sorties obligatoires

pm_reel(3)	axe	axe de rotation normé.
------------	------------	------------------------

pm_reel	angle	angle de rotation (rad).
---------	--------------	--------------------------

tm_code_retour	code_retour	
----------------	--------------------	--

Conditions sur les arguments

- La norme du quaternion doit être non nulle.
- La première composante du quaternion normé doit être différente de +/- 1 pour pouvoir déterminer l’axe de rotation.

Notes d’utilisation

- La transformation inverse (rotation → quaternion) peut être effectuée par la routine **mu_axe_angle_quat**

Références documentaires

- Algorithmes des routines du thème "Utilitaires mathématiques" de la MSLIB; G. Prat, avec la participation de L. Maisonobe (CS SI); référence MSLIB: M-NT-0-96-CIS.

Code retour (voir explications dans le volume 3)

pm_OK (0) : Retour normal.

pm_err_quat_nul (-2005) : La norme du quaternion est nulle.

pm_err_axe_rot_indef (-2006) : La première composante du quaternion normé vaut 1 ou -1:
l'axe de rotation est indéfini.

Exemple en Fortran 90 portable (voir explications dans le volume 3)

```
program MATH

  use mslib

  type(tm_quat)                :: QUAT
  real(pm_reel), dimension(3) :: AXE
  real(pm_reel)                :: ANGLE
  type(tm_code_retour)         :: CODE_RETOUR

  QUAT%q0      = 2._pm_reel
  QUAT%q123(1) = 1._pm_reel
  QUAT%q123(2) = 0._pm_reel
  QUAT%q123(3) = -1._pm_reel

  call mu_quat_axe_angle ( QUAT, AXE, ANGLE, CODE_RETOUR )

  ! appel a la routine utilisateur d'ecriture des resultats
  call WRITE_RESULTATS (AXE, ANGLE, CODE_RETOUR)

end program MATH
```

Résultats attendus:

AXE(1) = .707
AXE(2) = 0.
AXE(3) = -.707
ANGLE = .123 10⁺¹

CODE_RETOUR%valeur = 0
CODE_RETOUR%routine = 1124

Routine mu_quat_conjug

Identification

“Calcul du quaternion conjugué d’un quaternion donné.”

Rôle

Calcul du quaternion conjugué d’un quaternion donné.

Séquence d’appel

(voir explications dans le volume 3)

call mu_quat_conjug (quat, quat_conjug, code_retour)

Description des arguments

(voir explications dans le volume 3)

- Entrées obligatoires

tm_quat	quat	quaternion.
---------	-------------	-------------

- Sorties obligatoires

tm_quat	quat_conjug	quaternion conjugué.
---------	--------------------	----------------------

tm_code_retour	code_retour	
----------------	--------------------	--

Conditions sur les arguments

Sans objet.

Notes d’utilisation

Sans objet.

Références documentaires

- Algorithmes des routines du thème "Utilitaires mathématiques" de la MSLIB; G. Prat, avec la participation de L. Maisonobe (CS SI); référence MSLIB: M-NT-0-96-CIS.

Code retour (voir explications dans le volume 3)

pm_OK (0) : Retour normal.

Exemple en Fortran 90 portable (voir explications dans le volume 3)

```
program MATH

  use mslib

  type(tm_quat)           :: QUAT
  type(tm_quat)           :: QUAT_CONJUG
  type(tm_code_retour)    :: CODE_RETOUR

  QUAT%q0      = 2._pm_reel
  QUAT%q123(1) = 1._pm_reel
  QUAT%q123(2) = 0._pm_reel
  QUAT%q123(3) = -1._pm_reel

  call mu_quat_conjug (QUAT, QUAT_CONJUG, CODE_RETOUR)
  ! Le code retour de mu_quat_conjug est = 0

  ! appel a la routine utilisateur d'ecriture des resultats
  call WRITE_RESULTATS (QUAT_CONJUG, CODE_RETOUR)

end program MATH
```

Résultats attendus:

```
QUAT_CONJUG%q0      = 2.
QUAT_CONJUG%q123(1) = -1.
QUAT_CONJUG%q123(2) = 0.
QUAT_CONJUG%q123(3) = 1.

CODE_RETOUR%valeur  = 0
CODE_RETOUR%routine = 1126
```

Routine *mu_quat_mat*

Identification

“Calcul de la **mat**rice de rotation associée à un **quat**ernion.”

Rôle

Calcul, à partir d'un quaternion (normé ou non), de la matrice de rotation associée.

Séquence d'appel

(voir explications dans le volume 3)

call *mu_quat_mat* (quat, mat, code_retour)

Description des arguments

(voir explications dans le volume 3)

- Entrées obligatoires

tm_quat	quat	quaternion \tilde{q} correspondant à la matrice de rotation M
---------	-------------	---

- Sorties obligatoires

pm_reel(3,3)	mat	matrice de rotation M
tm_code_retour	code_retour	

Conditions sur les arguments

- Le quaternion \tilde{q} en entrée doit être non nul.

Notes d'utilisation

- La transformation inverse peut s'effectuer à l'aide de la routine **mu_mat_quat..**

Références documentaires

- Algorithmes des routines du thème "Utilitaires mathématiques" de la MSLIB; G. Prat, avec la participation de L. Maisonobe (CS SI); référence MSLIB: M-NT-0-96-CIS.

Code retour (voir explications dans le volume 3)

pm_OK (0) : Retour normal.
pm_err_quat_nul (-2005) : La norme du quaternion est nulle.

Exemple en Fortran 90 portable (voir explications dans le volume 3)

```
program MATH

  use mslib

  type(tm_quat)                :: QUAT
  real(pm_reel), dimension(3,3):: MAT
  type(tm_code_retour)         :: CODE_RETOUR

  QUAT%q0      = 0.8686315144381912_pm_reel
  QUAT%q123(1) = 0._pm_reel
  QUAT%q123(2) = 0._pm_reel
  QUAT%q123(3) = 0.49545866843240755_pm_reel

  call mu_quat_mat ( QUAT, MAT, CODE_RETOUR )

  ! appel a la routine utilisateur d'ecriture des resultats
  call WRITE_RESULTATS (MAT, CODE_RETOUR)

end program MATH
```

Résultats attendus:

MAT(1 , 1) = 0.509
MAT(2 , 1) = -0.861
MAT(3 , 1) = 0.
MAT(1 , 2) = 0.861
MAT(2 , 2) = 0.509
MAT(3 , 2) = 0.
MAT(1 , 3) = 0.
MAT(2 , 3) = 0.
MAT(3 , 3) = 1.

CODE_RETOUR%valeur = 0
CODE_RETOUR%routine = 1148

Routine mu_quat_norme

Identification

“Normalisation d’un **quat**ernion après calcul de sa **norme**.”

Rôle

Calcul de la norme et normalisation d’un quaternion.

Séquence d’appel

(voir explications dans le volume 3)

call mu_quat_norme (quat, quat_norme, norme, code_retour)

Description des arguments

(voir explications dans le volume 3)

- Entrées obligatoires

tm_quat	quat	quaternion.
---------	-------------	-------------

- Sorties obligatoires

tm_quat	quat_norme	quaternion normé.
pm_reel	norme	norme du quaternion.
tm_code_retour	code_retour	

Conditions sur les arguments

- La norme du quaternion doit être strictement positive.

Notes d’utilisation

Sans objet.

Références documentaires

- Algorithmes des routines du thème "Utilitaires mathématiques" de la MSLIB; G. Prat, avec la participation de L. Maisonobe (CS SI); référence MSLIB: M-NT-0-96-CIS.

Code retour (voir explications dans le volume 3)

pm_OK (0) : Retour normal.
pm_err_quat_nul (-2005) : La norme du quaternion est nulle.

Exemple en Fortran 90 portable (voir explications dans le volume 3)

```
program MATH

  use mslib

  type(tm_quat)          :: QUAT
  type(tm_quat)          :: QUAT_NORME
  real(pm_reel)          :: NORME
  type(tm_code_retour)   :: CODE_RETOUR

  QUAT%q0      = 2._pm_reel
  QUAT%q123(1) = 1._pm_reel
  QUAT%q123(2) = 0._pm_reel
  QUAT%q123(3) = -1._pm_reel

  call mu_quat_norme (QUAT, QUAT_NORME, NORME, CODE_RETOUR)

  ! appel a la routine utilisateur d'ecriture des resultats
  call WRITE_RESULTATS (QUAT_NORME, NORME, CODE_RETOUR)

end program MATH
```

Résultats attendus:

```
QUAT_NORME%q0      = .816
QUAT_NORME%q123(1) = .408
QUAT_NORME%q123(2) = 0.
QUAT_NORME%q123(3) = -.408
NORME              = .245 10+1

CODE_RETOUR%valeur = 0
CODE_RETOUR%routine = 1125
```

Routine mu_quat_rep

Identification

“A l’aide d’un **quat**ernion, calcul de changement de **rep**ère.”

Rôle

A l’aide du quaternion de passage d’un repère R_1 vers un repère R_2 , on exprime dans R_2 un vecteur initialement exprimé dans R_1 .

Séquence d’appel

(voir explications dans le volume 3)

call mu_quat_rep (vect1, quat, vect2, code_retour)

Description des arguments

(voir explications dans le volume 3)

• Entrées obligatoires

pm_reel(3)	vect1	coordonnées du vecteur dans le repère R_1
tm_quat	quat	quaternion de passage du repère R_1 au repère R_2

• Sorties obligatoires

pm_reel(3)	vect2	coordonnées du vecteur dans le repère R_2
tm_code_retour	code_retour	

Conditions sur les arguments

- Le quaternion de passage ne doit pas être nul.

Notes d’utilisation

- Les deux vecteurs sont identiques, ils sont juste exprimés dans deux repères différents.

Références documentaires

- Algorithmes des routines du thème "Utilitaires mathématiques" de la MSLIB; G. Prat, avec la participation de L. Maisonobe (CS SI); référence MSLIB: M-NT-0-96-CIS.

Code retour (voir explications dans le volume 3)

pm_OK (0) : Retour normal.
pm_err_quat_nul (-2005) : La norme du quaternion est nulle.

Exemple en Fortran 90 portable (voir explications dans le volume 3)

```
program MATH

  use mslib

  real(pm_reel), dimension(3) :: VECT1
  type(tm_quat) :: QUAT
  real(pm_reel), dimension(3) :: VECT2
  type(tm_code_retour) :: CODE_RETOUTOUR

  VECT1(1) = 1._pm_reel
  VECT1(2) = 0._pm_reel
  VECT1(3) = 0._pm_reel

  QUAT%q0 = 2._pm_reel
  QUAT%q123(1) = 1._pm_reel
  QUAT%q123(2) = 0._pm_reel
  QUAT%q123(3) = -1._pm_reel

  ! Passage du vecteur vect1 de R1 a R2
  call mu_quat_rep (VECT1, QUAT, VECT2, CODE_RETOUTOUR)

  ! appel a la routine utilisateur d'ecriture des resultats
  call WRITE_RESULTATS (VECT2, CODE_RETOUTOUR)

end program MATH
```

Résultats attendus:

VECT2(1) = .667
VECT2(2) = .667
VECT2(3) = -.333

CODE_RETOUTOUR%valeur = 0
CODE_RETOUTOUR%routine = 1128