

MSLIB Fortran 90

CS

Nomenclature : **M-MU-0-118-CIS**

Edition : 04 Date: 15/02/2005

Révision 00 Date: 15/02/2005

Volume V

**changement de Variables pour les paramètres
orbitaux**

Rédigé par : Guylaine PRAT avec la participation de: Véronique LÉPINE	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 V - changement de Variables pour les paramètres orbitaux**AUTEUR :** Guylaine PRAT avec la participation de: Véronique LÉPINE**RESUME :**

Ce document rassemble les notices d'utilisation des routines du thème "changement de Variables pour les paramètres orbitaux".

SITUATION DU DOCUMENT : Création**VOLUME :****PAGES :** 76**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	14/08/98	M-MU-0-118-CIS Rédacteur: V. Lépine avec la participation de G. Prat.		Création
02	00	28/01/00	M-MU-0-118-CIS Rédacteur : S. Vresk avec la participation de G. Prat	M	Modifications de toutes les pages
02	01	30/09/03	M-MU-0-118-CIS Rédacteur : B. Revelin	M	Évolutions de mv_car_kep et mv_car_equa (limite sur e)
03	00	27/05/04	M-MU-0-118-CIS Rédacteur: V. Lépine avec la participation de G. Prat	I	Ajout des pages liées aux nouvelles routines pour la MSLIB90 V6.0
04	00	15/02/05	M-MU-0-118-CIS Rédacteur: G. Prat avec la participation de V. Lépine	I	Ajout des pages liées à la nouvelle routine pour la MSLIB90 V6.2

* I = Inséré

S = Supprimé

M = Modifié

Sommaire

Présentation du thème V :	<i>page 1</i>
Notations	<i>page 2</i>
Index.	<i>page 3</i>
1 Paramètres cartésiens dans un repère inertiel (pour un satellite).	<i>page 4</i>
2 Paramètres orbitaux dans un repère inertiel (pour un satellite).	<i>page 5</i>
2.1 Définition des paramètres orbitaux	<i>page 5</i>
2.2 Définition des jeux de paramètres orbitaux	<i>page 8</i>
2.3 Utilisation des différents jeux de paramètres orbitaux	<i>page 9</i>
3 Utilisation des routines du thème	<i>page 10</i>
3.1 Utilisation des routines du thème pour effectuer des changements de formes dans un repère inertiel (pour un satellite)	<i>page 10</i>
3.2 Utilisation des routines du thème pour les résolutions des équations de Kepler et de Barker	<i>page 13</i>
3.3 Utilisation des routines du thème pour effectuer des changements d'anomalies E, M et v	<i>page 13</i>
4 Documents de référence du thème	<i>page 14</i>

Liste des routines du thème V : voir pages suivantes du sommaire.

Liste des routines du thème V:

mv_car_cir :	page 15
“Passage des paramètres cartésiens aux paramètres orbitaux dits adaptés aux orbites circulaires non équatoriales”.		
mv_car_cir_equa :	page 18
“Passage des paramètres cartésiens aux paramètres orbitaux dits adaptés aux orbites circulaires équatoriales”.		
mv_car_equa :	page 21
“Passage des paramètres cartésiens aux paramètres orbitaux dits adaptés aux orbites équatoriales non circulaires”.		
mv_car_kep :	page 25
“Passage des paramètres cartésiens aux paramètres képlériens”.		
mv_cir_car :	page 29
“Passage des paramètres orbitaux dits adaptés aux orbites circulaires non équatoriales aux paramètres cartésiens ”.		
mv_cir_equa_car :	page 32
“Passage des paramètres orbitaux dits adaptés aux orbites circulaires équatoriales aux paramètres cartésiens”.		
mv_cir_equa_kep :	page 36
“Passage des paramètres orbitaux dits adaptés aux orbites circulaires équatoriales aux paramètres képlériens.”		
mv_cir_kep :	page 39
“Passage des paramètres orbitaux dits adaptés aux orbites circulaires non équatoriales aux paramètres képlériens.”		
mv_conv_anom :	page 42
“Conversion d’anomalies excentrique, moyenne et vraie dans les cas elliptique, hyperbolique et parabolique.”		
mv_equa_kep :	page 49
“Passage des paramètres orbitaux dits adaptés aux orbites équatoriales non circulaires aux paramètres képlériens.”		

mv_kep_car :	<i>page 52</i>
“Passage des paramètres képlériens aux paramètres cartésiens”.	
mv_kep_cir :	<i>page 56</i>
“Passage des paramètres képlériens aux paramètres orbitaux dits adaptés aux orbites circulaires non équatoriales.”	
mv_kep_cir_equa :	<i>page 58</i>
“Passage des paramètres képlériens aux paramètres orbitaux dits adaptés aux orbites circulaires équatoriales.”	
mv_kep_equa :	<i>page 60</i>
“Passage des paramètres képlériens aux paramètres orbitaux dits adaptés aux orbites équatoriales non circulaires.”	
mv_kepler_gene :	<i>page 65</i>
“Résolution de l’équation de Kepler généralisée”.	
mv_kepler_std :	<i>page 68</i>
“Résolution de l’équation de Kepler standard (orbite elliptique)”.	

Présentation du thème V

Le thème “*changement de Variables pour les paramètres orbitaux*” regroupe un ensemble de routines permettant d’effectuer des transformations, dans un même système de référence, entre différentes formes. Il existe également des routines permettant de résoudre les *équations de Kepler* (standard, généralisée) et de *Barker*.

Le but de cette présentation est, d’une part, de définir les termes et les notations employés par la MSLIB dans ce thème, et, d’autre part, de présenter les routines du thème.

Les formes décrites dans cette présentation sont définies dans un repère inertiel.

A noter:

on appelle *forme* un type de jeu de coordonnées (coordonnées cartésiennes, paramètres képlériens, ...)

Notations

Alphabet latin

a	demi-grand axe
D	anomalie “excentrique” pour la parabole = $\tan(v/2)$
E	anomalie excentrique (pour l’ellipse)
H	anomalie excentrique pour l’hyperbole
e	excentricité
(e_x, e_y)	vecteur excentricité (dit adapté aux orbites circulaires) avec: $e_x = e \cos \omega$ et $e_y = e \sin \omega$
$(\tilde{e}_x, \tilde{e}_y)$	vecteur excentricité (dit adapté aux orbites circulaires équatoriales) avec: $\tilde{e}_x = e \cos (\omega + \Omega)$ et $\tilde{e}_y = e \sin (\omega + \Omega)$
i	inclinaison
(i_x, i_y)	vecteur inclinaison (dit adapté aux orbites équatoriales) avec: $i_x = 2 \sin (i/2) \cos \Omega$ et $i_y = 2 \sin (i/2) \sin \Omega$
M	anomalie moyenne
n	moyen mouvement
p	paramètre de la parabole
v	anomalie vraie
$(\vec{X}, \dot{\vec{X}})$	paramètres cartésiens (d’un satellite) avec: $\vec{X} = (x, y, z)$ et $\dot{\vec{X}} = (\dot{x}, \dot{y}, \dot{z})$

Alphabet grec

Ω	longitude (ou ascension droite) du nœud ascendant
μ	constante gravitationnelle (pour la Terre $\approx 39\,860\,047\,10^7 \text{ m}^3\text{s}^{-2}$)
ω	argument du périégée

Index

A

anomalie excentrique 6
anomalie moyenne 6
anomalie vraie 6
argument du périée 5
ascension droite du nœud ascendant 5

C

constante gravitationnelle 6

D

demi-grand axe 5

E

équation de Barker 1, 13
équation de Kepler 1, 13
excentricité 5

F

forme 1

I

inclinaison 5

L

ligne des nœuds 5
longitude du nœud ascendant 5

M

moyen mouvement 6

N

nœud ascendant 5

O

orbite circulaire 5
orbite elliptique 5, 9
orbite équatoriale 5
orbite équatoriale rétrograde 5
orbite hyperbolique 5, 9
orbite parabolique 5, 9

P

paramètre de la parabole 5
paramètres adaptés 8
paramètres cartésiens 4
paramètres képlériens 8
paramètres orbitaux 5, 8

V

vecteur inclinaison 7
vecteurs excentricité 7

1 Paramètres cartésiens dans un repère inertiel (pour un satellite)

Les *paramètres cartésiens* $\left(\overrightarrow{X}, \dot{\overrightarrow{X}} \right)$ définissent naturellement la position-vitesse d'un satellite à un instant donné, dans un repère inertiel.

2 Paramètres orbitaux dans un repère inertiel (pour un satellite)

2.1 Définition des paramètres orbitaux

Dans un repère inertiel ($OXYZ$), les *paramètres orbitaux* permettent de définir sans ambiguïté la position d'un satellite à un instant donné. On définit pour cela:

- l'orientation du plan de l'orbite dans l'espace:

→ Ω (*longitude du nœud ascendant* ou *ascension droite du nœud ascendant*; $\Omega \in [0, 2\pi[$)

donne l'angle entre la ligne des nœuds et l'axe OX du repère; la *ligne des nœuds* étant la droite intersection du plan (OXY) du repère et du plan de l'orbite;

le point d'intersection tel que le satellite passe de Z^- à Z^+ à la traversée du plan (OXY) est nommé *nœud ascendant*, et c'est cette direction que l'angle Ω repère,

→ i (*inclinaison*; $i \in [0, \pi]$)

donne l'angle formé par le plan (OXY) du repère et le plan de l'orbite ; suivant l'inclinaison :

$i = 0$ ou π : orbite équatoriale (dans le cas où $i = \pi$, on parle alors d'orbite équatoriale rétrograde)

$i \neq 0$ et $\neq \pi$: orbite non équatoriale

- la position de l'orbite dans son plan:

→ ω (*argument du périégée*)

donne l'angle entre la direction du nœud ascendant et la direction du périégée de l'orbite;

- les dimensions et la forme de la conique (ellipse, parabole, hyperbole) :

→ e (*excentricité*)

donne la forme de la conique; suivant l'excentricité :

$e = 0$: orbite circulaire

$0 < e < 1$: orbite elliptique

$e = 1$: orbite parabolique

$e > 1$: orbite hyperbolique

→ a (*demi-grand axe*), associé à l'excentricité,

donne les dimensions de l'ellipse ou de l'hyperbole;

→ p (*paramètre de la parabole*)

donne la dimension de la parabole;

- la position du satellite sur son orbite (information de date):

→ M (*anomalie moyenne*)

est un angle qui permet de connaître la position du satellite sur l'orbite;

les angles v (*anomalie vraie*) ou E (*anomalie excentrique* pour l'ellipse, noté H pour l'hyperbole) permettent aussi de connaître la position du satellite sur son orbite;

pour v et E voir la Figure 2. en page 7; M est donné par:

$$M = n (t - t_p)$$

où

t = date à laquelle on connaît les paramètres orbitaux du satellite,

t_p = date de passage du satellite au périégée,

n (*moyen mouvement*) = $\sqrt{\mu/a^3}$ et μ = *constante gravitationnelle*,

A noter:

L'axe OZ est couramment l'axe des pôles de la planète.

Le plan (OXY) est quant à lui souvent confondu avec le plan équatorial de la planète.

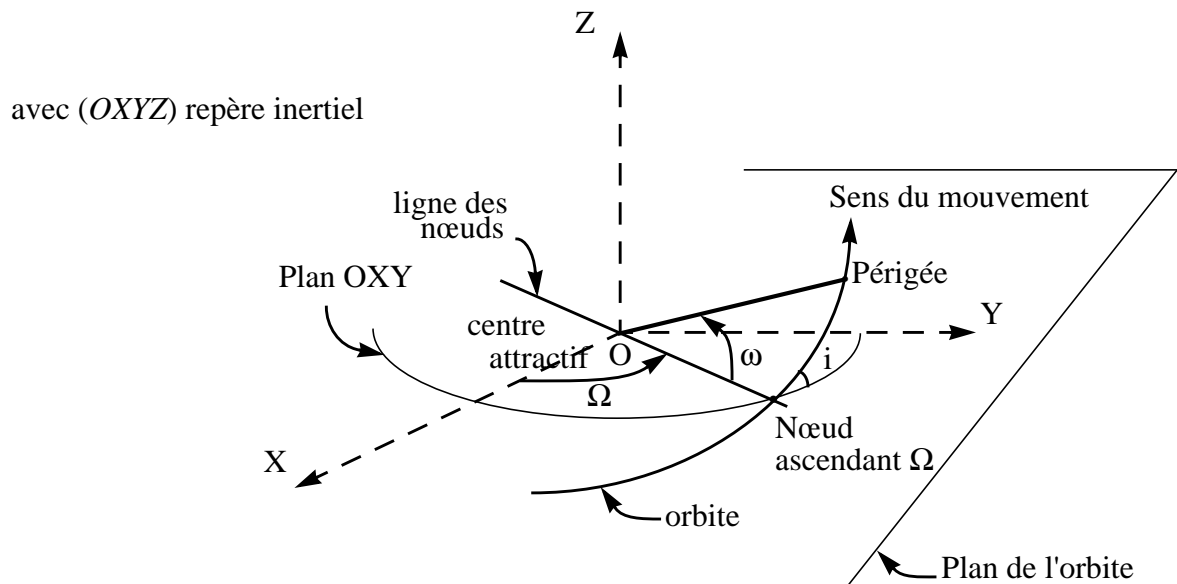


Figure 1.

A noter:

pour la définition des paramètres liés à une parabole ou une hyperbole: voir les documents de référence (§ 4 en page 14).

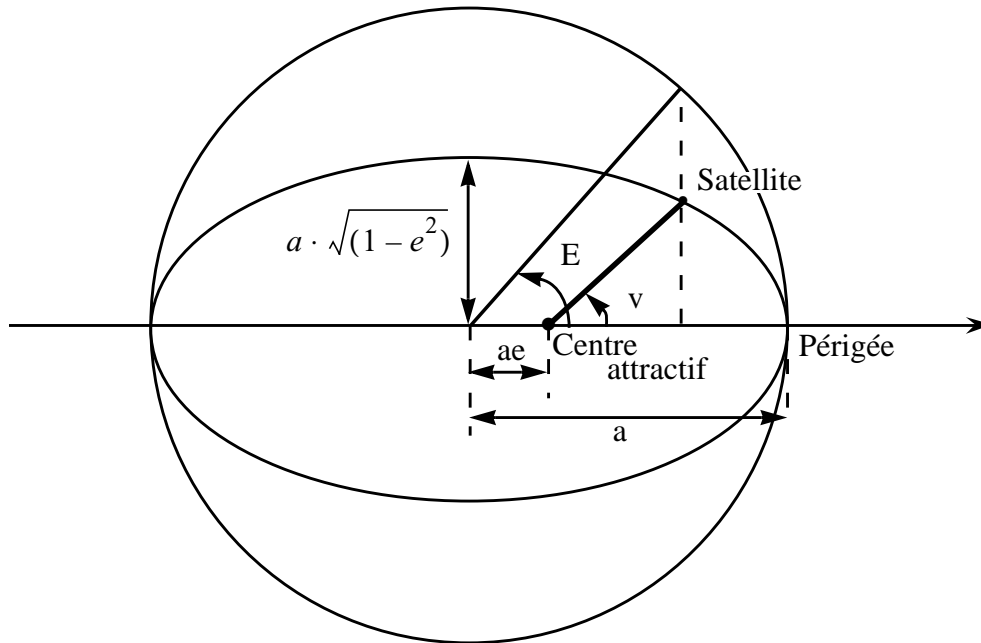


Figure 2.

On définit, par ailleurs, les *vecteurs excentricité*:

$$\begin{pmatrix} e_x \\ e_y \end{pmatrix} = \begin{pmatrix} e \cdot \cos \omega \\ e \cdot \sin \omega \end{pmatrix} \quad \text{et} \quad \begin{pmatrix} \tilde{e}_x \\ \tilde{e}_y \end{pmatrix} = \begin{pmatrix} e \cdot \cos(\omega + \Omega) \\ e \cdot \sin(\omega + \Omega) \end{pmatrix}$$

et le *vecteur inclinaison*:

$$\begin{pmatrix} i_x \\ i_y \end{pmatrix} = \begin{pmatrix} 2 \sin \frac{i}{2} \cos \Omega \\ 2 \sin \frac{i}{2} \sin \Omega \end{pmatrix}$$

Remarque:

pour le vecteur (i_x, i_y) , certaines notations correspondent plutôt à (mais nous ne les utiliserons pas ici): $(i_x, i_y) = (\sin i \cdot \cos \Omega, \sin i \cdot \sin \Omega)$ ou encore $(i \cdot \cos \Omega, i \cdot \sin \Omega)$

2.2 Définition des jeux de paramètres orbitaux

A l'aide des *paramètres orbitaux* décrits précédemment, on définit les jeux de paramètres suivants:

Paramètres orbitaux $(a, \tilde{e}_x, \tilde{e}_y, i_x, i_y, \omega + \Omega + M)$

Ce jeu de paramètres est couramment appelé *paramètres adaptés* aux orbites circulaires équatoriales non rétrogrades ($i = \pi$).

Ce jeu de paramètres est conseillé pour des orbites :

- elliptiques (circulaires ou non), non équatoriales rétrogrades.

Paramètres orbitaux $(a \text{ ou } p, e, \omega + \Omega, i_x, i_y, M)$

Ce jeu de paramètres est couramment appelé *paramètres adaptés* aux orbites équatoriales non rétrogrades et non circulaires.

Ce jeu de paramètres est conseillé pour des orbites :

- elliptiques (non circulaires), non équatoriales rétrogrades,
- paraboliques, non équatoriales rétrogrades,
- hyperboliques, non équatoriales rétrogrades.

Paramètres orbitaux $(a, e_x, e_y, i, \Omega, \omega + M)$

Ce jeu de paramètres est couramment appelé *paramètres adaptés* aux orbites circulaires non équatoriales.

Ce jeu de paramètres est conseillé pour des orbites :

- elliptiques (circulaires ou non) non équatoriales.

Paramètres orbitaux $(a \text{ ou } p, e, i, \omega, \Omega, M)$

Ce jeu de paramètres est couramment appelé *paramètres képlériens*.

Ce jeu de paramètres est conseillé pour des orbites :

- elliptiques (non circulaires) non équatoriales,
- paraboliques non équatoriales,
- hyperboliques non équatoriales.

2.3 Utilisation des différents jeux de paramètres orbitaux

On peut remarquer que pour des *orbites elliptiques*:

- non circulaires et non équatoriales, on peut utiliser les jeux de paramètres:

$$(a, \tilde{e}_x, \tilde{e}_y, i_x, i_y, \omega + \Omega + M)$$

$$(a, e, \omega + \Omega, i_x, i_y, M)$$

$$(a, e_x, e_y, i, \Omega, \omega + M)$$

$$(a, e, i, \omega, \Omega, M)$$

- circulaires ou non, et non équatoriales, on peut utiliser les jeux de paramètres:

$$(a, \tilde{e}_x, \tilde{e}_y, i_x, i_y, \omega + \Omega + M)$$

$$(a, e_x, e_y, i, \Omega, \omega + M)$$

- non circulaires et non équatoriales rétrogrades, on peut utiliser les jeux de paramètres:

$$(a, \tilde{e}_x, \tilde{e}_y, i_x, i_y, \omega + \Omega + M)$$

$$(a, e, \omega + \Omega, i_x, i_y, M)$$

- circulaires ou non et non équatoriales rétrogrades, on peut utiliser les jeux de paramètres:

$$(a, \tilde{e}_x, \tilde{e}_y, i_x, i_y, \omega + \Omega + M)$$

Par ailleurs, on peut remarquer que pour les *orbites paraboliques* ou *hyperboliques*:

- non équatoriales, on peut utiliser les jeux de paramètres:

$$(a \text{ ou } p, e, \omega + \Omega, i_x, i_y, M)$$

$$(a \text{ ou } p, e, i, \omega, \Omega, M)$$

- non équatoriales rétrogrades, on peut utiliser les jeux de paramètres:

$$(a \text{ ou } p, e, \omega + \Omega, i_x, i_y, M)$$

3 Utilisation des routines du thème

Nous vous proposons dans ce paragraphe une représentation schématique des transformations disponibles dans le thème. Les notations et les définitions utilisées dans ce paragraphe ont déjà été explicitées auparavant et ne seront donc pas répétées ici.

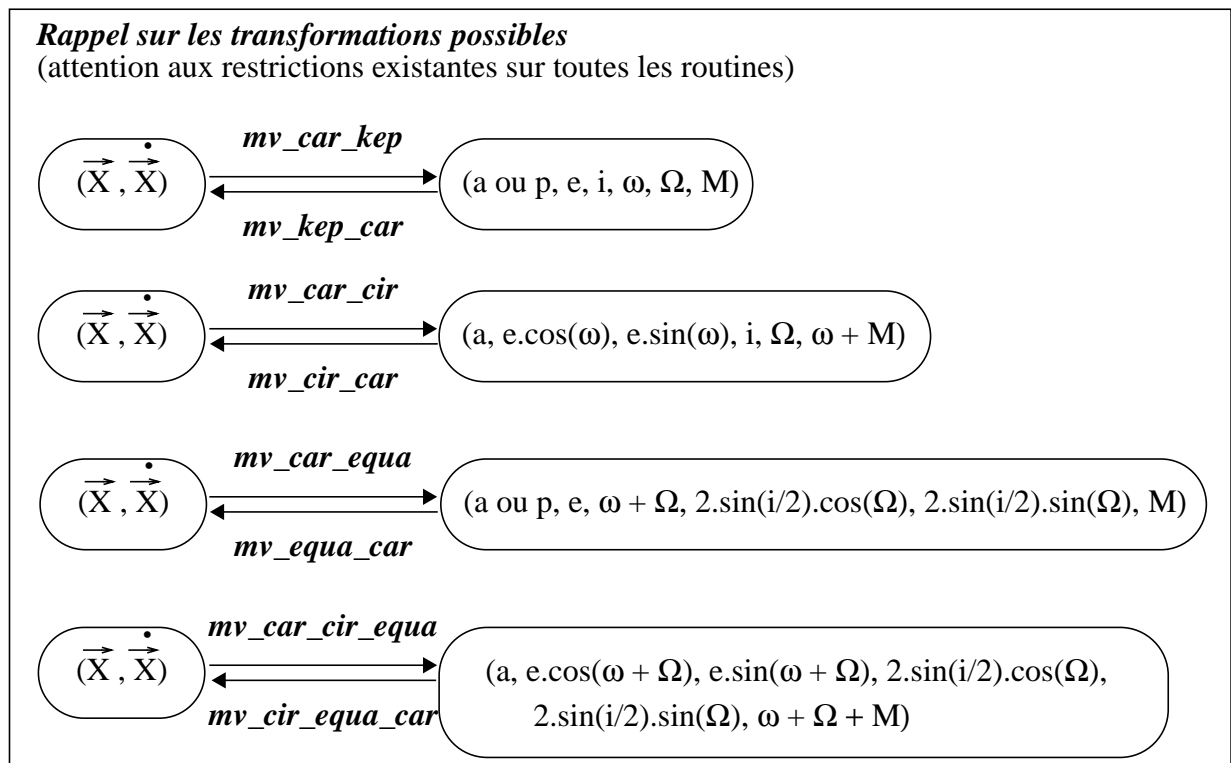
3.1 Utilisation des routines du thème pour effectuer des changements de formes dans un repère inertiel (pour un satellite)

routines concernées:

$mv_car_kep \Leftrightarrow mv_kep_car$
 $mv_car_cir \Leftrightarrow mv_cir_car$
 $mv_car_equa \Leftrightarrow mv_equa_car$
 $mv_car_cir_equa \Leftrightarrow mv_cir_equa_car$

Transformations pour les passages

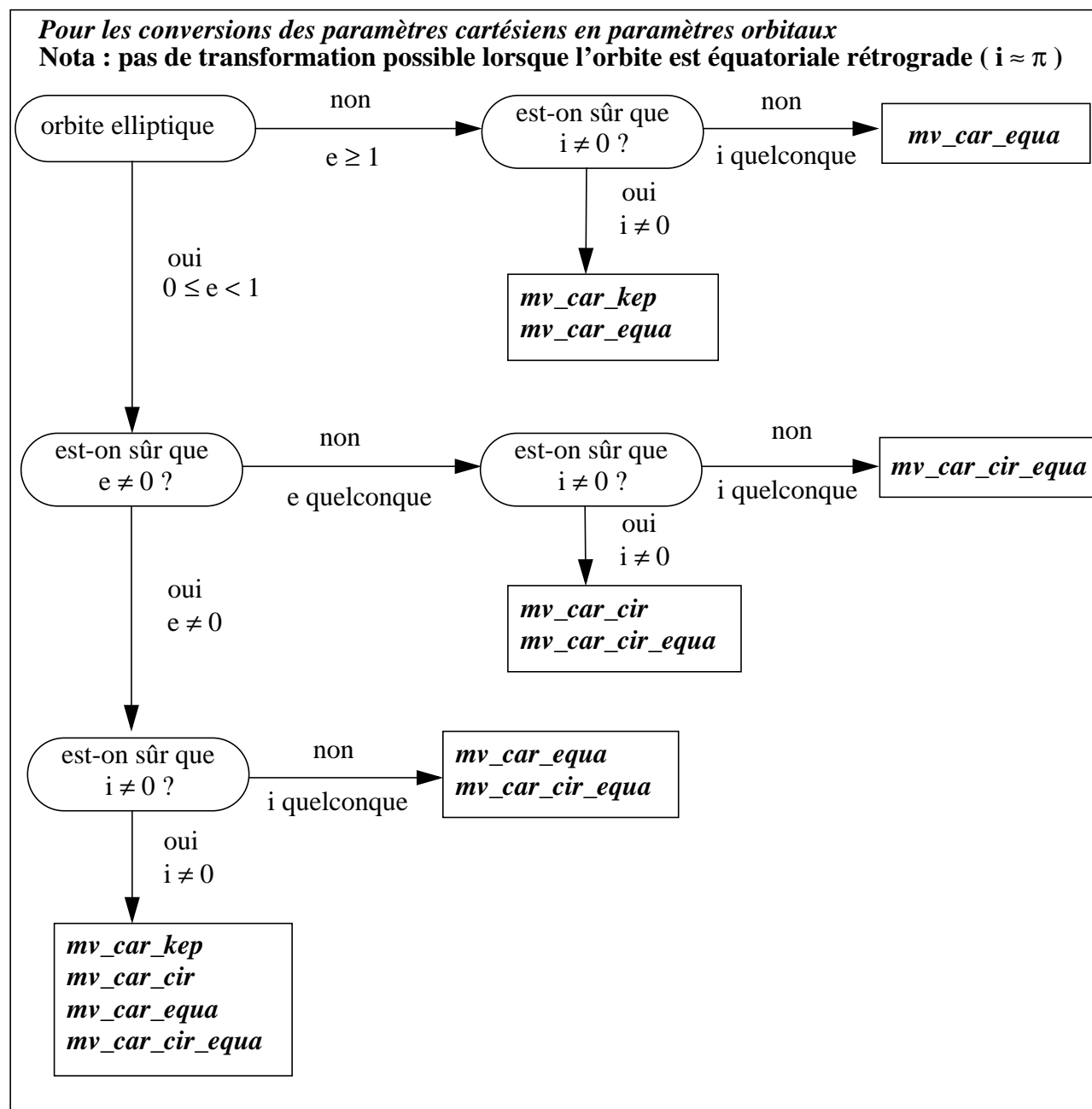
paramètres cartésiens \Leftrightarrow paramètres orbitaux:



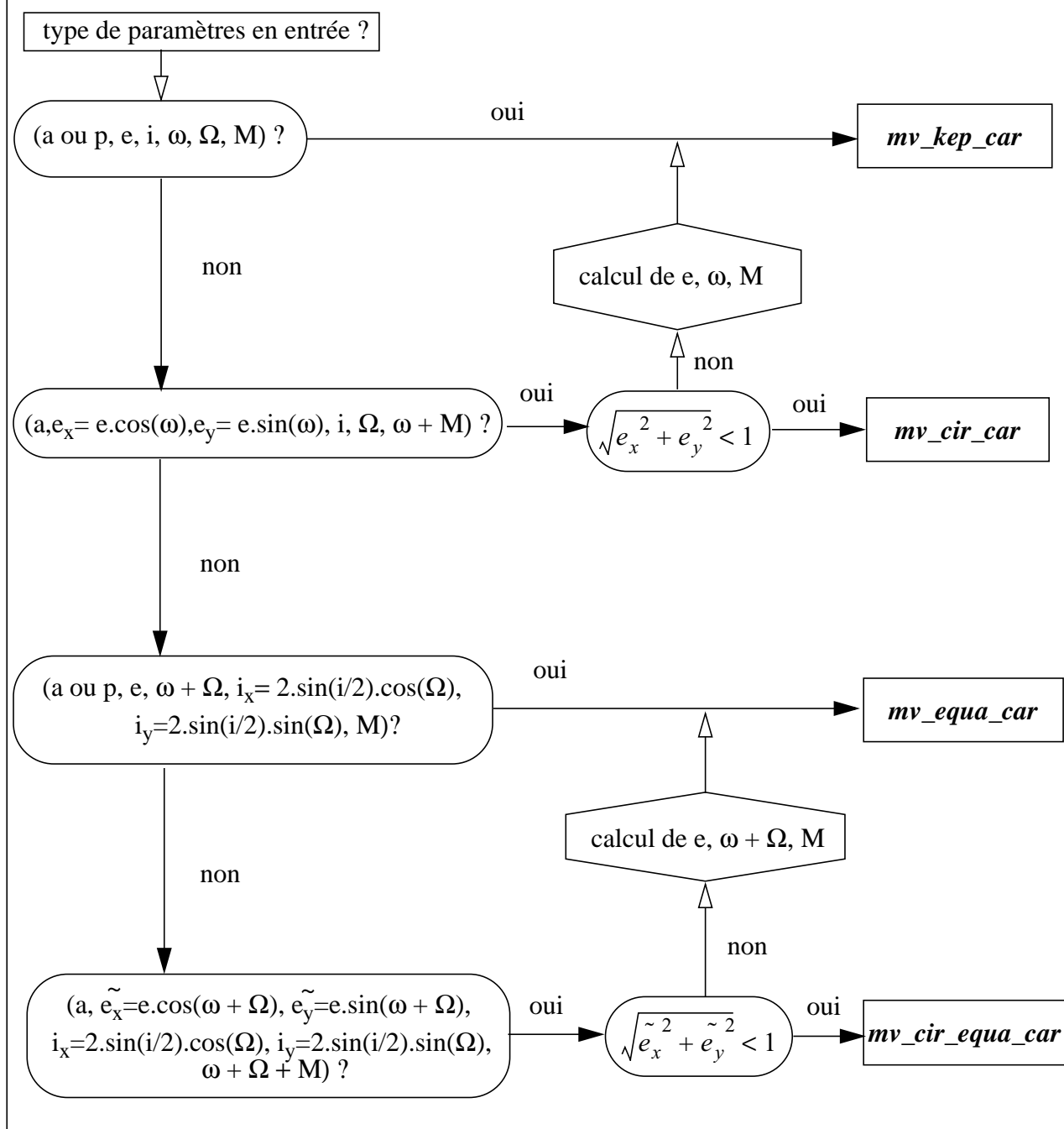
Conditions d'utilisation des routines: (pour les valeurs limites utilisées pour l'excentricité et l'inclinaison, se reporter au thème **Constantes**).

mv_car_kep	: $e \neq 0$ et $i \neq \{0, \pi\}$	mv_kep_car	: $e \geq 0$ et $i \in [0, \pi]$
mv_car_cir	: $0 \leq e < 1$ et $i \neq \{0, \pi\}$	mv_cir_car	: $0 \leq e < 1$ et $i \in [0, \pi]$
mv_car_equa	: $e \neq 0$ et $i \neq \pi$	mv_equa_car	: $e \geq 0$ et $i \in [0, \pi[$
$mv_car_cir_equa$: $0 \leq e < 1$ et $i \neq \pi$	$mv_cir_equa_car$: $0 \leq e < 1$ et $i \in [0, \pi[$

Compte tenu des restrictions existantes sur toutes les routines, nous vous proposons les graphes de décisions suivants:



Pour les conversions des paramètres orbitaux en paramètres cartésiens

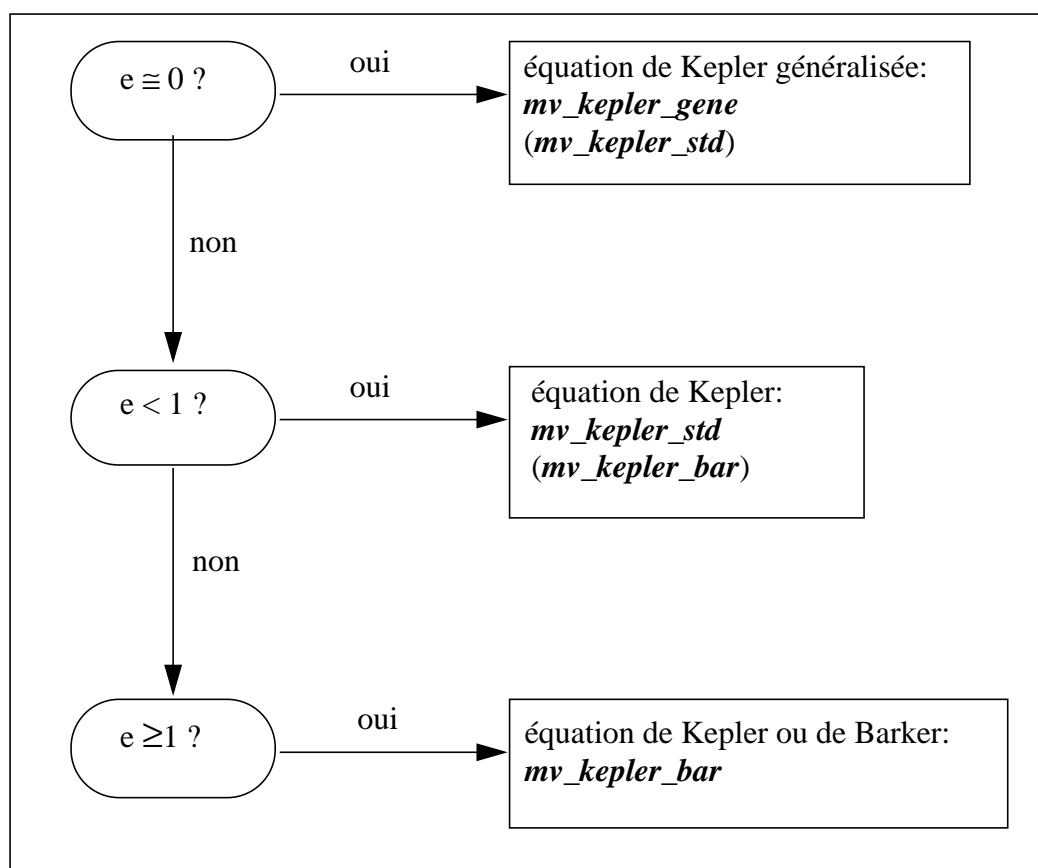


3.2 Utilisation des routines du thème pour les résolutions des équations de Kepler et de Barker

routines concernées:

`mv_kepler_std`
`mv_kepler_gene`
`mv_kepler_bar`

Pour résoudre les *équations de Kepler* et de *Barker*, nous vous proposons le graphe de décision suivant:



3.3 Utilisation des routines du thème pour effectuer des changements d'anomalies E, M et v

routines concernées:

`mv_conv_anom`
(cf également le § 3.2)

4 Documents de référence du thème

- Le mouvement du véhicule spatial en orbite, 1980, CNES, Cepadues.
- Le mouvement du satellite, 1983, CNES, Cepadues.
- Mathématiques Spatiales, 1984, CNES, Cepadues.
- Trajectoires spatiales, O. Zarrouati, 1987, CNES Cepadues.
- Mécanique Spatiale 1995, tomes I et II, CNES, Cepadues.

Routine mv_car_cir

Identification

“Passage des paramètres **cart**ésiens aux paramètres orbitaux dits adaptés aux orbites **cir**culaires non équatoriales”.

Rôle

Passage des paramètres cartésiens $(x, y, z, \dot{x}, \dot{y}, \dot{z})$ aux paramètres orbitaux $(a, e_x, e_y, i, \Omega, \omega + M)$ dans le cas d'une orbite circulaire ou elliptique, et NON équatoriale. Le jacobien de la transformation est calculé en option.

Séquence d'appel

(voir explications dans le volume 3)

call mv_car_cir (mu, pos_car, vit_car, cir, code_retour [, jacob])

Description des arguments

(voir explications dans le volume 3)

• Entrées obligatoires

pm_reel	mu	constante de la gravitation ($\text{m}^3.\text{s}^{-2}$).
pm_reel(3)	pos_car	position en coordonnées cartésiennes, $\vec{X} = (x, y, z)$ (m)
pm_reel(3)	vit_car	vitesse en coordonnées cartésiennes, $\vec{\dot{X}} = (\dot{x}, \dot{y}, \dot{z})$ (m.s^{-1})

• Sorties obligatoires

tm_orb_cir	cir	paramètres orbitaux $(a, e_x, e_y, i, \Omega, \omega + M)$ (m, rad)
tm_code_retour	code_retour	

• Sorties facultatives

pm_reel(6,6)	[jacob]	jacobien de la transformation $\frac{\partial(a, e_x, e_y, i, \Omega, \omega + M)}{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})}$
--------------	------------------	---

Conditions sur les arguments

- Le jacobien vaut:

$$jacob(i, j) = \frac{\partial(\text{parametre } a, e_x, e_y, i, \Omega, \omega + M \ n^\circ i)}{\partial(\text{parametre } x, y, z, \dot{x}, \dot{y}, \dot{z} \ n^\circ j)}$$

Exemple:

$$jacob(1, 4) = \frac{\partial a}{\partial \dot{x}}$$

Notes d'utilisation

- Non applicable aux orbites équatoriales.
- Non applicable aux orbites paraboliques ou hyperboliques.
- La transformation inverse peut se faire par la routine **mv_cir_car**.

Références documentaires

- Algorithmes des routines du thème "changement de Variables pour les paramètres orbitaux" de la MSLIB; B. Revelin (CS SI); référence MSLIB: M-NT-0-97-CIS.

Code retour (voir explications dans le volume 3)

pm_OK	(0) : Retour normal.
pm_err_mu_negatif	(-1001) : La constante gravitationnelle est négative.
pm_err_mu_nul	(-1002) : La constante gravitationnelle est proche de 0.
pm_err_e_non_ellip	(-1208) : L'excentricité (e) n'appartient pas à l'intervalle [0, 1 [; l'orbite n'est pas elliptique.
pm_err_i_equa	(-1302) : sin(i) est proche de 0 ; l'orbite est équatoriale (i=0 ou i=pi).
pm_err_pos_nul	(-1501) : La norme du vecteur position est proche de 0 .
pm_err_vit_nul	(-1502) : La norme du vecteur vitesse est proche de 0 .

pm_err_pos_vit_colineaire (-1503) : Le produit vectoriel position-vitesse est pratiquement nul (ce qui signifie: position nulle ou vitesse nulle ou vecteurs position et vitesse colinéaires).

pm_err_cni (-1999) : Problème numérique. Contacter l'assistance utilisateur MSLIB.

Exemple en Fortran 90 portable

(voir explications dans le volume 3)

```
program CHANGE_VAR

  use mslib

  real(pm_reel)                :: MU
  real(pm_reel), dimension(3) :: POS_CAR
  real(pm_reel), dimension(3) :: VIT_CAR
  type(tm_orb_cir)             :: CIR
  type(tm_code_retour)         :: CODE_RETOUR

  MU      = 39860047e+15_pm_reel
  POS_CAR(1) = -5910180._pm_reel
  POS_CAR(2) = 4077714._pm_reel
  POS_CAR(3) = -620640._pm_reel
  VIT_CAR(1) = 129._pm_reel
  VIT_CAR(2) = -1286._pm_reel
  VIT_CAR(3) = -7325._pm_reel

  call mv_car_cir ( MU, POS_CAR, VIT_CAR, CIR, CODE_RETOUR)

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

end program CHANGE_VAR
```

Résultats attendus:

```
CIR%a      = .721 10+7
CIR%ex      = .275 10-2
CIR%ey      = -.271 10-1
CIR%i       = .144 10+1
CIR%gom     = .567 10+1
CIR%pso_M   = .328 10+1
```

```
CODE_RETOUR%valeur = 0
CODE_RETOUR%routine = 1035
```


Routine mv_car_cir_equa

Identification

“Passage des paramètres **cart**ésiens aux paramètres orbitaux dits adaptés aux orbites **cir**culaires **équa**toriales”.

Rôle

Passage des paramètres cartésiens $(x, y, z, \dot{x}, \dot{y}, \dot{z})$ aux paramètres orbitaux

$(a, \tilde{e}_x, \tilde{e}_y, i_x, i_y, \omega + \Omega + M)$ dans le cas d’une orbite circulaire ou elliptique, et pour une inclinaison différente de π . Le jacobien de la transformation est calculé en option.

Nota : lorsque l’inclinaison est proche de π , l’orbite est dite équatoriale rétrograde.

Séquence d’appel

(voir explications dans le volume 3)

call mv_car_cir_equa (mu, pos_car, vit_car, cir_equa, code_retour [, jacob])

Description des arguments

(voir explications dans le volume 3)

• Entrées obligatoires

pm_reel	mu	constante de la gravitation ($\text{m}^3.\text{s}^{-2}$).
pm_reel(3)	pos_car	position en coordonnées cartésiennes, $\vec{X} = (x, y, z)$ (m)
pm_reel(3)	vit_car	vitesse en coordonnées cartésiennes, $\vec{\dot{X}} = (\dot{x}, \dot{y}, \dot{z})$ ($\text{m}.\text{s}^{-1}$)

• Sorties obligatoires

tm_orb_cir_equa	cir_equa	paramètres orbitaux $(a, \tilde{e}_x, \tilde{e}_y, i_x, i_y, \omega + \Omega + M)$ (m, rad)
tm_code_retour	code_retour	

• Sorties facultatives

pm_reel(6,6)	[jacob]	jacobien de la transformation $\frac{\partial(a, \tilde{e}_x, \tilde{e}_y, i_x, i_y, \omega + \Omega + M)}{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})}$
--------------	------------------	---

Conditions sur les arguments

- Le jacobien vaut:

$$jacob(i, j) = \frac{\partial(\text{parametre } (a, \tilde{e}_x, \tilde{e}_y, i_x, i_y, \omega + \Omega + M \text{ } n^\circ i))}{\partial(\text{parametre } x, y, z, \dot{x}, \dot{y}, \dot{z} \text{ } n^\circ j)}$$

Exemple:

$$jacob(1, 4) = \frac{\partial a}{\partial \dot{x}}$$

Notes d'utilisation

- Non applicable aux orbites paraboliques ou hyperboliques.
- Non applicable a une orbite équatoriale rétrograde ($i \approx \pi$), par contre fonctionne pour tous les autres types d'orbites, y compris pour les orbites équatoriales non rétrogrades ($i \approx 0$).
- La transformation inverse peut se faire par la routine **mv_cir_equa_car**.

Références documentaires

- Algorithmes des routines du thème "changement de Variables pour les paramètres orbitaux" de la MSLIB; B. Revelin (CS SI); référence MSLIB: M-NT-0-97-CIS.

Code retour

(voir explications dans le volume 3)

pm_OK	(0) : Retour normal.
pm_err_mu_negatif	(-1001) : La constante gravitationnelle est négative.
pm_err_mu_nul	(-1002) : La constante gravitationnelle est proche de 0.
pm_err_a_negatif	(-1101) : Le demi-grand axe (a) ou le paramètre (p) de la parabole est négatif.
pm_err_a_infini	(-1103) : L'inverse du demi-grand axe (1/a) est proche de 0 (le demi-grand axe est donc infini).
pm_err_e_non_ellip	(-1208) : L'excentricité (e) n'appartient pas à l'intervalle [0, 1 [; l'orbite n'est pas elliptique.
pm_err_i_equa_retro	(-1307) : L'inclinaison (i) est égale à pi.

pm_err_pos_nul (-1501) : La norme du vecteur position est proche de 0 .

pm_err_pos_vit_colineaire (-1503) : Le produit vectoriel position-vitesse est pratiquement nul (ce qui signifie: position nulle ou vitesse nulle ou vecteurs position et vitesse colinéaires).

pm_err_cni (-1999) : Problème numérique. Contacter l'assistance utilisateur MSLIB.

Exemple en Fortran 90 portable

(voir explications dans le volume 3)

```

program CHANGE_VAR

  use mslib

  real(pm_reel)                :: MU
  real(pm_reel), dimension(3)  :: POS_CAR
  real(pm_reel), dimension(3)  :: VIT_CAR
  type(tm_orb_cir_equa)         :: CIR_EQUA
  type(tm_code_retour)          :: CODE_RETOUR

  MU      = .39860047e+15_pm_reel
  POS_CAR(1) = -29536113._pm_reel
  POS_CAR(2) = 30329259._pm_reel
  POS_CAR(3) = -100125._pm_reel
  VIT_CAR(1) = -2194._pm_reel
  VIT_CAR(2) = -2141._pm_reel
  VIT_CAR(3) = -8._pm_reel

  call mv_car_cir_equa ( MU, POS_CAR, VIT_CAR, CIR_EQUA,      &
                        CODE_RETOUR)

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

end program CHANGE_VAR

```

Résultats attendus:

```

CIR_EQUA%a      = .423 10+8
CIR_EQUA%ex     = .593 10-3
CIR_EQUA%ey     = -.206 10-2
CIR_EQUA%ix     = .128 10-3
CIR_EQUA%iy     = -.352 10-2
CIR_EQUA%pso_M  = .234 10+1

CODE_RETOUR%valeur = 0
CODE_RETOUR%routine = 1045

```

Routine mv_car_equa

Identification

“Passage des paramètres **cart**ésiens aux paramètres orbitaux dits adaptés aux orbites **équ**atoriales non circulaires”.

Rôle

Passage des paramètres cartésiens $(x, y, z, \dot{x}, \dot{y}, \dot{z})$ aux paramètres orbitaux $(a \text{ ou } p, e, \omega + \Omega, i_x, i_y, M)$ dans le cas d’une orbite non circulaire et pour une inclinaison différente de π . Le jacobien de la transformation est calculé en option.

Nota : lorsque l’inclinaison est proche de π , l’orbite est dite équatoriale rétrograde.

Séquence d’appel

(voir explications dans le volume 3)

call mv_car_equa (mu, pos_car, vit_car, equa, code_retour [, jacob])

Description des arguments

(voir explications dans le volume 3)

• Entrées obligatoires

pm_reel	mu	constante de la gravitation ($\text{m}^3.\text{s}^{-2}$).
pm_reel(3)	pos_car	position en coordonnées cartésiennes, $\vec{X} = (x, y, z)$ (m)
pm_reel(3)	vit_car	vitesse en coordonnées cartésiennes, $\vec{\dot{X}} = (\dot{x}, \dot{y}, \dot{z})$ ($\text{m}.\text{s}^{-1}$)

• Sorties obligatoires

tm_orb_equa	equa	paramètres orbitaux $(a \text{ ou } p, e, \omega + \Omega, i_x, i_y, M)$ (m, rad.)
tm_code_retour	code_retour	

• Sorties facultatives

pm_reel(6,6)	[jacob]	jacobien de la transformation: $\frac{\partial(a, e, \omega + \Omega, i_x, i_y, M)}{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})} \text{ ou } \frac{\partial(p, e, \omega + \Omega, i_x, i_y, M)}{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})}$
--------------	------------------	---

Conditions sur les arguments

- Dans le cas elliptique ou hyperbolique, le jacobien vaut:

$$jacob(i, j) = \frac{\partial(\text{parametre } a, e, \omega + \Omega, i_x, i_y, M n^\circ i)}{\partial(\text{parametre } x, y, z, \dot{x}, \dot{y}, \dot{z} n^\circ j)}$$

Et dans le cas parabolique, il vaut:

$$jacob(i, j) = \frac{\partial(\text{parametre } p, e, \omega + \Omega, i_x, i_y, M n^\circ i)}{\partial(\text{parametre } x, y, z, \dot{x}, \dot{y}, \dot{z} n^\circ j)}$$

Exemple:

$$jacob(1, 4) = \frac{\partial a}{\partial \dot{x}} \text{ ou } \frac{\partial p}{\partial \dot{x}}$$

Notes d'utilisation

- Ne fonctionne pas pour une orbite circulaire ($e \approx 0$).
- Applicable aux trois types de conique : ellipse, hyperbole, parabole.
- Ne fonctionne pas pour une orbite équatoriale rétrograde ($i \approx \pi$) mais fonctionne dans le cas d'orbites équatoriales non-rétrogrades ($i \approx 0$).
- La transformation inverse peut se faire par **mv_equa_car**.

Références documentaires

- Algorithmes des routines du thème "changement de Variables pour les paramètres orbitaux" de la MSLIB; B. Revelin (CS SI); référence MSLIB: M-NT-0-97-CIS.

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

pm_OK	(0) : Retour normal.
pm_err_mu_negatif	(-1001) : La constante gravitationnelle est négative.
pm_err_mu_nul	(-1002) : La constante gravitationnelle est proche de 0.
pm_err_a_negatif	(-1101) : Le demi-grand axe (a) ou le paramètre (p) de la parabole est négatif.
pm_err_a_nul	(-1102) : Le demi-grand axe (a) ou le paramètre (p) de la parabole est proche de 0 .
pm_err_a_infini	(-1103) : L'inverse du demi-grand axe (1/a) est proche de 0 (le demi-grand axe est donc infini).
pm_err_e_faible	(-1212) : L'excentricité (e) est inférieure à 1e-7; les routines de changement de variables car -> kep et car -> equa n'autorisent pas ces valeurs de l'excentricité.
pm_err_i_equa_retro	(-1307) : L'inclinaison (i) est égale à pi.
pm_err_pos_nul	(-1501) : La norme du vecteur position est proche de 0 .
pm_err_vit_nul	(-1502) : La norme du vecteur vitesse est proche de 0 .
pm_err_pos_vit_colineaire	(-1503) : Le produit vectoriel position-vitesse est pratiquement nul (ce qui signifie: position nulle ou vitesse nulle ou vecteurs position et vitesse colinéaires).
pm_err_cni	(-1999) : Problème numérique. Contacter l'assistance utilisateur MSLIB.
pm_warn_e_parab	(+1206) : L'excentricité (e) est proche de 1 : les calculs ont été faits en considérant que l'orbite était parabolique.

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

```
program CHANGE_VAR
```

```
  use mslib
```

```
  real(pm_reel)                :: MU
  real(pm_reel), dimension(3)  :: POS_CAR
  real(pm_reel), dimension(3)  :: VIT_CAR
  type(tm_orb_equa)             :: EQUA
  type(tm_code_retour)          :: CODE_RETOUT
```

```
MU          =      .39860047e+15_pm_reel
POS_CAR(1)  =      .516812407223e+7_pm_reel
POS_CAR(2)  =     -.410313660482e+7_pm_reel
POS_CAR(3)  =     -.997907822992e+3_pm_reel
VIT_CAR(1)  =      .676026276781e+4_pm_reel
VIT_CAR(2)  =      .766758428731e+4_pm_reel
VIT_CAR(3)  =     -.178204046852e+2_pm_reel
```

```
call mv_car_equa ( MU, POS_CAR, VIT_CAR, EQUA, CODE_RETOUR )
```

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

```
end program CHANGE_VAR
```

Résultats attendus:

```
EQUA%a      = .244 10+8
EQUA%e      = .731
EQUA%pgom   = .549 10+1
EQUA%ix     = -.127 10-2
EQUA%iy     = .120 10-2
EQUA%M      = .130 10-1
```

```
CODE_RETOUR%valeur  = 0
CODE_RETOUR%routine = 1037
```

Routine mv_car_kep

Identification

“Passage des paramètres **cartésiens** aux paramètres **képlériens**”.

Rôle

Passage des paramètres cartésiens $(x, y, z, \dot{x}, \dot{y}, \dot{z})$ aux paramètres képlériens $(a$ ou $p, e, i, \omega, \Omega, M)$, dans le cas d’une orbite non circulaire et non équatoriale. Le jacobien de la transformation est calculé en option.

Séquence d’appel

(voir explications dans le volume 3)

call mv_car_kep (mu, pos_car, vit_car, kep, code_retour [, jacob])

Description des arguments

(voir explications dans le volume 3)

• Entrées obligatoires

pm_reel	mu	constante de la gravitation ($\text{m}^3.\text{s}^{-2}$).
pm_reel(3)	pos_car	position en coordonnées cartésiennes, $\vec{X} = (x, y, z)$ (m)
pm_reel(3)	vit_car	vitesse en coordonnées cartésiennes, $\vec{\dot{X}} = (\dot{x}, \dot{y}, \dot{z})$ (m.s^{-1})

• Sorties obligatoires

tm_orb_kep	kep	paramètres képlériens $(a$ ou $p, e, i, \omega, \Omega, M)$ (m, rad)
tm_code_retour	code_retour	

• Sorties facultatives

pm_reel(6,6)	[jacob]	jacobien de la transformation: $\frac{\partial(a, e, i, \omega, \Omega, M)}{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})} \text{ ou } \frac{\partial(p, e, i, \omega, \Omega, M)}{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})} .$
--------------	------------------	---

Conditions sur les arguments

- Dans le cas elliptique ou hyperbolique, le jacobien vaut:

$$jacob(i, j) = \frac{\partial(\text{parametre } (a, e, i, \omega, \Omega, M \text{ } n^{\circ}i))}{\partial(\text{parametre } x, y, z, \dot{x}, \dot{y}, \dot{z} \text{ } n^{\circ}j)}$$

Dans le cas parabolique, le jacobien vaut:

$$jacob(i, j) = \frac{\partial(\text{parametre } (p, e, i, \omega, \Omega, M \text{ } n^{\circ}i))}{\partial(\text{parametre } x, y, z, \dot{x}, \dot{y}, \dot{z} \text{ } n^{\circ}j)}$$

Exemple:

$$jacob(1, 4) = \frac{\partial a}{\partial \dot{x}} \text{ ou } \frac{\partial p}{\partial \dot{x}}$$

Notes d'utilisation

- Non applicables aux orbites circulaires et/ou équatoriales.
- Applicable aux trois types de conique : ellipse, hyperbole, parabole.
- La transformation inverse peut se faire par **mv_kep_car**.

Références documentaires

- Algorithmes des routines du thème "changement de Variables pour les paramètres orbitaux" de la MSLIB; B. Revelin (CS SI); référence MSLIB: M-NT-0-97-CIS.

Code retour

(voir explications dans le volume 3)

pm_OK

(0) : Retour normal.

pm_err_mu_negatif

(-1001) : La constante gravitationnelle est négative.

pm_err_mu_nul	(-1002) : La constante gravitationnelle est proche de 0.
pm_err_a_negatif	(-1101) : Le demi-grand axe (a) ou le paramètre (p) de la parabole est négatif.
pm_err_a_nul	(-1102) : Le demi-grand axe (a) ou le paramètre (p) de la parabole est proche de 0 .
pm_err_a_infini	(-1103) : L'inverse du demi-grand axe (1/a) est proche de 0 (le demi-grand axe est donc infini).
pm_err_e_faible	(-1212) : L'excentricité (e) est inférieure à 1e-7; les routines de changement de variables car -> kep et car -> equa n'autorisent pas ces valeurs de l'excentricité.
pm_err_i_equa	(-1302) : sin(i) est proche de 0 ; l'orbite est équatoriale (i=0 ou i=pi).
pm_err_pos_nul	(-1501) : La norme du vecteur position est proche de 0 .
pm_err_vit_nul	(-1502) : La norme du vecteur vitesse est proche de 0 .
pm_err_pos_vit_colineaire	(-1503) : Le produit vectoriel position-vitesse est pratiquement nul (ce qui signifie: position nulle ou vitesse nulle ou vecteurs position et vitesse colinéaires).
pm_err_cni	(-1999) : Problème numérique. Contacter l'assistance utilisateur MSLIB.
pm_warn_e_parab	(+1206) : L'excentricité (e) est proche de 1 : les calculs ont été faits en considérant que l'orbite était parabolique.

Exemple en Fortran 90 portable

(voir explications dans le volume 3)

```
program CHANGE_VAR
```

```
  use mslib
```

```
  real(pm_reel)                :: MU
  real(pm_reel), dimension(3)  :: POS_CAR
  real(pm_reel), dimension(3)  :: VIT_CAR
  type(tm_orb_kep)              :: KEP
  type(tm_code_retour)          :: CODE_RETOUR
```

```
  MU      = .39860047e+15_pm_reel
  POS_CAR(1) = -26655470._pm_reel
  POS_CAR(2) = 29881667._pm_reel
  POS_CAR(3) = -113657._pm_reel
  VIT_CAR(1) = -1125._pm_reel
  VIT_CAR(2) = -1122._pm_reel
  VIT_CAR(3) = 195._pm_reel
```

```
call mv_car_kep ( MU, POS_CAR, VIT_CAR, KEP, CODE_RETOUR)
```

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

```
end program CHANGE_VAR
```

Résultats attendus:

```
KEP%a      = .230 10+8  
KEP%e      = .744  
KEP%i      = .122  
KEP%pom    = .310 10+1  
KEP%gom    = .232 10+1  
KEP%M      = .323 10+1
```

```
CODE_RETOUR%valeur = 0  
CODE_RETOUR%routine = 1027
```

Routine mv_cir_car

Identification

“Passage des paramètres orbitaux dits adaptés aux orbites **circulaires** non équatoriales aux paramètres **cartésiens**”.

Rôle

Passage des paramètres orbitaux $(a, e_x, e_y, i, \Omega, \omega+M)$ dans le cas d'une orbite circulaire ou elliptique et équatoriale ou non, aux paramètres cartésiens $(x, y, z, \dot{x}, \dot{y}, \dot{z})$. Le jacobien de la transformation est calculé en option.

Séquence d'appel

(voir explications dans le volume 3)

call mv_cir_car (mu, cir, pos_car, vit_car, code_retour [, jacob])

Description des arguments

(voir explications dans le volume 3)

• Entrées obligatoires

pm_reel	mu	constante de la gravitation ($\text{m}^3.\text{s}^{-2}$).
tm_orb_cir	cir	paramètres orbitaux $(a, e_x, e_y, i, \Omega, \omega+M)$ (m, rad.).

• Sorties obligatoires

pm_reel(3)	pos_car	position en coordonnées cartésiennes, $\vec{X} = (x, y, z)$ (m)
pm_reel(3)	vit_car	vitesse en coordonnées cartésiennes, $\dot{\vec{X}} = (\dot{x}, \dot{y}, \dot{z})$ (m.s^{-1})
tm_code_retour	code_retour	

• Sorties facultatives

pm_reel(6,6)	[jacob]	jacobien de la transformation $\frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})}{\partial(a, e_x, e_y, i, \Omega, \omega + M)}$
--------------	------------------	---

Conditions sur les arguments

- L'excentricité doit appartenir à l'intervalle $[0., 1[$.

- Le jacobien vaut:

$$jacob(i, j) = \frac{\partial(\text{parametre } x, y, z, \dot{x}, \dot{y}, \dot{z} \text{ n}^{\circ} i)}{\partial(\text{parametre } a, e_x, e_y, i, \Omega, \omega + M \text{ n}^{\circ} j)}$$

Exemple:

$$jacob(4, 1) = \frac{\partial \dot{x}}{\partial a}$$

Notes d'utilisation

- Applicable aux orbites elliptiques ou circulaires et équatoriales ou non.
- Non applicable aux orbites paraboliques ou hyperboliques.
- La transformation inverse peut se faire par **mv_car_cir**.

Références documentaires

- Algorithmes des routines du thème "changement de Variables pour les paramètres orbitaux" de la MSLIB; B. Revelin (CS SI); référence MSLIB: M-NT-0-97-CIS.

Code retour (voir explications dans le volume 3)

pm_OK	(0) : Retour normal.
pm_err_mu_negatif	(-1001) : La constante gravitationnelle est négative.
pm_err_mu_nul	(-1002) : La constante gravitationnelle est proche de 0.
pm_err_a_negatif	(-1101) : Le demi-grand axe (a) ou le paramètre (p) de la parabole est négatif.
pm_err_a_nul	(-1102) : Le demi-grand axe (a) ou le paramètre (p) de la parabole est proche de 0 .
pm_err_e_non_ellip	(-1208) : L'excentricité (e) n'appartient pas à l'intervalle [0, 1 [; l'orbite n'est pas elliptique.
pm_err_conv_kepler_gene	(-1904) : L'algorithme itératif utilisé pour la résolution de l'équation de Kepler généralisée (équation avec le vecteur excentricité) n'a pas réussi à converger vers la bonne solution. Contacter l'assistance utilisateur MSLIB.

Exemple en Fortran 90 portable

(voir explications dans le volume 3)

```

program CHANGE_VAR
  use mslib

  real(pm_reel)                :: MU
  type(tm_orb_cir)              :: CIR
  real(pm_reel), dimension(3)   :: POS_CAR
  real(pm_reel), dimension(3)   :: VIT_CAR
  type(tm_code_retour)          :: CODE_RETOUR
  real(pm_reel), dimension(6,6) :: JACOB

  MU      = .39860047e+15_pm_reel
  CIR%a   = 7204649._pm_reel
  CIR%ex  = -2.9e-4_pm_reel
  CIR%ey  = 1.34e-3_pm_reel
  CIR%i   = 1.7233_pm_reel
  CIR%gom = 1.5745_pm_reel
  CIR%pso_M= 0.5726_pm_reel

  call mv_cir_car ( MU, CIR, POS_CAR, VIT_CAR, CODE_RETOUR &
                   jacob = JACOB)

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

end program CHANGE_VAR

```

Résultats attendus:

```

POS_CAR(1) = .568 10+6
POS_CAR(2) = .606 10+7
POS_CAR(3) = .384 10+7
VIT_CAR(1) = .966 10+3
VIT_CAR(2) = -.402 10+4
VIT_CAR(3) = .619 10+4

```

$$\text{JACOB} = \begin{bmatrix}
 0.079 & 0.842 & 0.533 & 0.000 & 0.028 \cdot 10^{-2} & -0.004 \cdot 10^{-1} \\
 0.533 \cdot 10^6 & -0.931 \cdot 10^7 & 0.325 \cdot 10^7 & 0.495 \cdot 10^3 & -0.677 \cdot 10^4 & 0.306 \cdot 10^4 \\
 -0.188 \cdot 10^7 & 0.326 \cdot 10^7 & -0.122 \cdot 10^8 & 0.102 \cdot 10^4 & 0.310 \cdot 10^4 & 0.669 \cdot 10^4 \\
 0.384 \cdot 10^7 & 0.142 \cdot 10^5 & -0.590 \cdot 10^6 & 0.619 \cdot 10^4 & 0.229 \cdot 10^2 & -0.951 \cdot 10^3 \\
 -0.606 \cdot 10^7 & 0.568 \cdot 10^6 & 0.000 & 0.402 \cdot 10^4 & 0.966 \cdot 10^3 & 0.000 \\
 0.936 \cdot 10^6 & -0.389 \cdot 10^7 & 0.599 \cdot 10^7 & -0.587 \cdot 10^3 & 0.627 \cdot 10^4 & -0.397 \cdot 10^4
 \end{bmatrix}$$

```

CODE_RETOUR%valeur = 0
CODE_RETOUR%routine = 1036

```

Routine mv_cir_equa_car

Identification

“Passage des paramètres orbitaux dits adaptés aux orbites **circulaires équatoriales** aux paramètres **cartésiens**”.

Rôle

Passage des paramètres orbitaux $(a, \tilde{e}_x, \tilde{e}_y, i_x, i_y, \omega + \Omega + M)$ aux paramètres cartésiens $(x, y, z, \dot{x}, \dot{y}, \dot{z})$ dans le cas d’une orbite circulaire ou elliptique et non équatoriale rétrograde. Le jacobien de la transformation est calculé en option.

Séquence d’appel

(voir explications dans le volume 3)

call mv_cir_equa_car (mu, cir_equa, pos_car, vit_car, code_retour [, jacob])

Description des arguments

(voir explications dans le volume 3)

• Entrées obligatoires

pm_reel	mu	constante de la gravitation ($\text{m}^3.\text{s}^{-2}$).
tm_orb_cir_equa	cir_equa	paramètres orbitaux $(a, \tilde{e}_x, \tilde{e}_y, i_x, i_y, \omega + \Omega + M)$ (m,rad.).

• Sorties obligatoires

pm_reel(3)	pos_car	position en coordonnées cartésiennes, $\vec{X} = (x, y, z)$ (m)
pm_reel(3)	vit_car	vitesse en coordonnées cartésiennes, $\dot{\vec{X}} = (\dot{x}, \dot{y}, \dot{z})$ (m.s^{-1})
tm_code_retour	code_retour	

• Sorties facultatives

pm_reel(6,6)	[jacob]	jacobien de la transformation $\frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})}{\partial(a, \tilde{e}_x, \tilde{e}_y, i_x, i_y, \omega + \Omega + M)}$
--------------	------------------	--

Conditions sur les arguments

- L'excentricité doit appartenir à l'intervalle [0.,1[.
- La norme du vecteur inclinaison (i_x, i_y) doit être strictement inférieure à 2.
- Le jacobien vaut:

$$jacob(i, j) = \frac{\partial(\text{parametre } x, y, z, \dot{x}, \dot{y}, \dot{z} \text{ } n^{\circ} i)}{\partial(\text{parametre } a, \tilde{e}_x, \tilde{e}_y, i_x, i_y, \omega + \Omega + M \text{ } n^{\circ} j)}$$

Exemple:

$$jacob(4, 1) = \frac{\partial \dot{x}}{\partial a}$$

Notes d'utilisation

- Applicable aux orbites elliptiques ou circulaires.
- Applicable aux orbites équatoriales non rétrogrades et aux orbites non équatoriales.
- Non applicable aux orbites hyperboliques ou paraboliques.
- La transformation inverse peut se faire par **mv_car_cir_equa**.

Références documentaires

- Algorithmes des routines du thème "changement de Variables pour les paramètres orbitaux" de la MSLIB; B. Revelin (CS SI); référence MSLIB: M-NT-0-97-CIS.

Code retour (voir explications dans le volume 3)

pm_OK	(0) : Retour normal.
pm_err_mu_negatif	(-1001) : La constante gravitationnelle est négative.
pm_err_mu_nul	(-1002) : La constante gravitationnelle est proche de 0.
pm_err_a_negatif	(-1101) : Le demi-grand axe (a) ou le paramètre (p) de la parabole est négatif.
pm_err_a_nul	(-1102) : Le demi-grand axe (a) ou le paramètre (p) de la parabole est proche de 0 .

pm_err_e_non_ellip (-1208) : L'excentricité (e) n'appartient pas à l'intervalle [0, 1 [; l'orbite n'est pas elliptique.

pm_err_ix_iy_sup2 (-1306) : La norme du vecteur inclinaison est trop grande (supérieure à 2).

pm_err_i_equa_retro (-1307) : L'inclinaison (i) est égale à pi.

pm_err_conv_kepler_gene (-1904) : L'algorithme itératif utilisé pour la résolution de l'équation de Kepler généralisée (équation avec le vecteur excentricité) n'a pas réussi à converger vers la bonne solution. Contacter l'assistance utilisateur MSLIB.

Exemple en Fortran 90 portable

(voir explications dans le volume 3)

```
program CHANGE_VAR

  use mslib

  real(pm_reel)                :: MU
  type(tm_orb_cir_equa)        :: CIR_EQUA
  real(pm_reel), dimension(3)  :: POS_CAR
  real(pm_reel), dimension(3)  :: VIT_CAR
  type(tm_code_retour)         :: CODE_RETOUR
  real(pm_reel), dimension(6,6) :: JACOB

  MU          = .39860047e15_pm_reel
  CIR_EQUA%a  = 42166.712_pm_reel
  CIR_EQUA%ex = -7.9e-6_pm_reel
  CIR_EQUA%ey = 1.1e-4_pm_reel
  CIR_EQUA%ix = 1.2e-4_pm_reel
  CIR_EQUA%iy = -1.16e-4_pm_reel
  CIR_EQUA%pso_M = 5.3_pm_reel

  call mv_cir_equa_car ( MU, CIR_EQUA, POS_CAR, VIT_CAR,
                        CODE_RETOUR, jacob = JACOB)

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

end program CHANGE_VAR
```

Résultats attendus:

POS_CAR(1) = .234 10⁺⁵
POS_CAR(2) = -.351 10⁺⁵
POS_CAR(3) = -.150 10⁺¹

VIT_CAR(1) = .809 10⁺⁵
VIT_CAR(2) = .539 10⁺⁵
VIT_CAR(3) = .159 10⁺²

JACOB =

$$\begin{bmatrix} 0.554 & -0.832 & -0.356 \cdot 10^{-4} & -0.959 & -0.639 & -0.188 \cdot 10^{-3} \\ -0.714 \cdot 10^5 & -0.195 \cdot 10^5 & -0.106 \cdot 10^2 & 0.897 \cdot 10^5 & -0.375 \cdot 10^5 & 0.591 \cdot 10^1 \\ -0.195 \cdot 10^5 & -0.551 \cdot 10^5 & -0.887 \cdot 10^1 & -0.375 \cdot 10^5 & -0.897 \cdot 10^5 & -0.151 \cdot 10^2 \\ 0.204 \cdot 10^1 & 0.286 \cdot 10^1 & -0.351 \cdot 10^5 & -0.313 \cdot 10^1 & -0.112 \cdot 10^2 & 0.539 \cdot 10^5 \\ 0.605 & 0.140 \cdot 10^1 & -0.234 \cdot 10^5 & 0.126 \cdot 10^2 & 0.485 \cdot 10^1 & -0.809 \cdot 10^5 \\ 0.351 \cdot 10^5 & 0.234 \cdot 10^5 & 0.688 \cdot 10^1 & -0.539 \cdot 10^5 & 0.809 \cdot 10^5 & 0.346 \cdot 10^1 \end{bmatrix}$$

CODE_RETOUR%valeur = 0
CODE_RETOUR%routine = 1046

Routine mv_cir_equa_kep

Identification

“Passage des paramètres orbitaux dits adaptés aux orbites **circulaires équatoriales** aux paramètres **képlériens**.”

Rôle

Passage des paramètres orbitaux dits adaptés aux orbites circulaires équatoriales

$(a, \tilde{e}_x, \tilde{e}_y, i_x, i_y, \omega + \Omega + M)$ aux paramètres képlériens $(a, e, i, \omega, \Omega, M)$.

Le calcul du jacobien de la transformation $\frac{\partial(a, e, i, \omega, \Omega, M)}{\partial(a, \tilde{e}_x, \tilde{e}_y, i_x, i_y, \omega + \Omega + M)}$ est optionnel.

Séquence d'appel

call mv_cir_equa_kep (cir_equa, kep, code_retour [, jacob])

Description des arguments

• Entrées obligatoires

tm_orb_cir_equa **cir_equa** paramètres orbitaux $(a, \tilde{e}_x, \tilde{e}_y, i_x, i_y, \omega + \Omega + M)$ (m, rad.)

• Sorties obligatoires

tm_orb_kep **kep** paramètres képlériens $(a, e, i, \omega, \Omega, M)$ (m, rad.)

tm_code_retour **code_retour**

• Sorties facultatives

pm_reel(6,6) [**jacob**] jacobienne de la transformation $\frac{\partial(a, e, i, \omega, \Omega, M)}{\partial(a, \tilde{e}_x, \tilde{e}_y, i_x, i_y, \omega + \Omega + M)}$

Conditions sur les arguments

Sans objet

Notes d'utilisation

- Applicable aux orbites circulaires ou elliptiques.
- Applicable aux orbites équatoriales ou non.
- Si l'orbite est équatoriale ($\sin(i) \approx 0$), la longitude du nœud ascendant Ω est indéfinie. Arbitrairement, nous lui donnons la valeur: $\Omega = 0$. De plus, la jacobienne n'est pas calculable.
- Si l'orbite est circulaire ($e \approx 0$), la somme de l'argument du périhélie ω et de la longitude du nœud ascendant Ω est indéfinie. Arbitrairement, nous lui donnons la valeur: $\omega + \Omega = 0$. De plus, la jacobienne n'est pas calculable.
- La transformation inverse peut être effectuée par **mv_kep_cir_equa**.

Références documentaires

- Algorithmes des routines du thème "changement de Variables pour les paramètres orbitaux" de la MSLIB; B. Revelin (CS SI); référence MSLIB: M-NT-0-97-CIS.

Code retour

pm_OK	(0) : Retour normal.
pm_err_ix_iy_sup2	(-1306) : La norme du vecteur inclinaison est trop grande (supérieure à 2).
pm_warn_i_equa	(+1302) : $\sin(i)$ est proche de 0; l'orbite est équatoriale ($i=0$ ou $i=\pi$).
pm_warn_e_circul	(+1204) : L'excentricité (e) est proche de 0; l'orbite est circulaire.
pm_warn_e_circul_i_equa	(+1201) : L'excentricité (e) est proche de 0 et $\sin(i)$ est proche de 0; l'orbite est circulaire et équatoriale ($i = 0$ ou $i = \pi$).
pm_err_jac_non_calc_i_equa	(-1308) : L'orbite est équatoriale ($i=0$ ou π). Il en résulte que la jacobienne n'est pas calculable.
pm_err_jac_non_calc_e_circul	(-1213) : L'orbite est circulaire ($e=0$). Il en résulte que la jacobienne n'est pas calculable.

Exemple de résultats

Pour la description des types dérivés, se reporter à 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

Entrées:

cir_equa%a	= 42166.712_pm_reel
cir_equa%ex	= -7.900e-06_pm_reel
cir_equa%ey	= 1.100e-04_pm_reel
cir_equa%ix	= 1.200e-04_pm_reel
cir_equa%iy	= -1.16e-04_pm_reel
cir_equa%pso_M	= 5.300_pm_reel

Résultats attendus:

kep%a	= 42166.712
kep%e	= 1.103 10 ⁻⁴
kep%i	= 1.669 10 ⁻⁴
kep%pom	= -3.872
kep%gom	= 5.515
kep%M	= 3.658

code_retour%valeur	= 0
--------------------	-----

Routine mv_cir_kep

Identification

“Passage des paramètres orbitaux dits adaptés aux orbites **circ**ulaires non équatoriales aux paramètres **képl**ériens.”

Rôle

Passage des paramètres orbitaux dits adaptés aux orbites circulaires non équatoriales $(a, e_x, e_y, i, \Omega, \omega+M)$ aux paramètres képlériens $(a, e, i, \omega, \Omega, M)$.

Le calcul du jacobien de la transformation $\frac{\partial(a, e, i, \omega, \Omega, M)}{\partial(a, e_x, e_y, i, \Omega, \omega + M)}$ est optionnel.

Séquence d'appel

call mv_cir_kep (cir, kep, code_retour [, jacob])

Description des arguments

• Entrées obligatoires

tm_orb_cir	cir	paramètres orbitaux $(a, e_x, e_y, i, \Omega, \omega+M)$ (m, rad.)
------------	------------	--

• Sorties obligatoires

tm_orb_kep	kep	paramètres képlériens $(a, e, i, \omega, \Omega, M)$ (m, rad.)
tm_code_retour	code_retour	

• Sorties facultatives

pm_reel(6,6)	[jacob]	jacobienne de la transformation $\frac{\partial(a, e, i, \omega, \Omega, M)}{\partial(a, e_x, e_y, i, \Omega, \omega + M)}$
--------------	------------------	---

Conditions sur les arguments

Sans objet

Notes d'utilisation

- Applicable aux orbites circulaires ou elliptiques.
- Applicable aux orbites équatoriales ou non.
- Si l'orbite est circulaire ($e \approx 0$), l'argument du périée ω est indéfinie. Arbitrairement, nous lui donnons la valeur: $\omega = 0$. De plus, la jacobienne n'est pas calculable.
- La transformation inverse peut être effectuée par **mv_kep_cir**.

Références documentaires

- Algorithmes des routines du thème "changement de Variables pour les paramètres orbitaux" de la MSLIB; B. Revelin (CS SI); référence MSLIB: M-NT-0-97-CIS.

Code retour

pm_OK	(0) : Retour normal.
pm_warn_e_circul	(+1204) : L'excentricité (e) est proche de 0; l'orbite est circulaire.
pm_err_jac_non_calc_e_circul	(-1213) : L'orbite est circulaire (e=0). Il en résulte que la jacobienne n'est pas calculable.

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.

Entrées:

cir%a	= 7204649.0_pm_reel
cir%ex	= -2.9000e-04_pm_reel
cir%ey	= 1.3400e-03_pm_reel
cir%i	= 1.7233_pm_reel
cir%gom	= 1.5745_pm_reel
cir%pso_M	= 0.5726_pm_reel

Résultats attendus:

kep%a = 7204649.
kep%e = 1.371 10⁻³
kep%i = 1.723
kep%pom = 1.784
kep%gom = 1.575
kep%M = -1.211

code_retour%valeur = 0

Routine mv_conv_anom

Identification

“**Conversion d’anomalies** excentrique, moyenne et vraie dans les cas elliptique, hyperbolique et parabolique.”

Rôle

Calcul de l’anomalie excentrique E_2 , moyenne M_2 ou vraie v_2 à partir de l’anomalie excentrique E_1 , moyenne M_1 ou vraie v_1 pour tout type d’excentricité e , avec:

E : anomalie excentrique

M : anomalie moyenne

v : anomalie vraie

e : excentricité

Notation:

nous regroupons sous le terme générique "anomalie excentrique E " le cas de :

- E pour l’ellipse,
- H pour l’hyperbole,
- $D = \tan(v/2)$ pour la parabole.

Séquence d’appel

call mv_conv_anom (e, type_anom1, anom1, type_anom2, anom2, code_retour)

Description des arguments

• Entrées obligatoires

pm_reel	e	excentricité e
integer	type_anom1	type de l’anomalie à convertir
pm_reel	anom1	anomalie à convertir E_1 , M_1 ou v_1 (rad)
integer	type_anom2	type de l’anomalie finale

• Sorties obligatoires

pm_reel	anom2	anomalie finale E_2 , M_2 ou v_2 (rad)
tm_code_retour	code_retour	

Conditions sur les arguments

- **type_anom1** et **type_anom2** correspondent chacun à un entier qui définit le type d'anomalie :

Valeurs possibles pour type_anom1 et type_anom2	
Anomalie excentrique E	pm_anom_E
Anomalie moyenne M	pm_anom_M
Anomalie vraie v	pm_anom_v

- Le domaine pour l'anomalie vraie v_1 est limité dans le cas de l'hyperbole ($e > 1$), il faut que:

$$\cos v_1 > \frac{-1}{e}$$

Notes d'utilisation

- Applicable aux orbites elliptiques, hyperboliques et paraboliques.
- Le domaine pour l'anomalie vraie v_2 calculée dépend du type d'orbite:

domaine pour l'anomalie vraie v_2 calculée	
ellipse ($e < 1$)	$v_2 \in [0, 2\pi]$
hyperbole ($e > 1$)	v_2 tel que $\left(\cos v_2 > \frac{-1}{e} \right)$, donc $v_2 \in]-\theta, \theta[\subset]-\pi, \pi[$ avec $\theta = \arccos\left(\frac{-1}{e}\right)$
parabole ($e = 1$)	$v_2 \in]-\pi, \pi[$

- Dans le cas où la transformation demandée correspond à l'identité (**type_anom1** = **type_anom2**), l'anomalie calculée **anom2** = **anom1** (sans aucune transformation, en particulier pour respecter les domaines de l'anomalie vraie).

Références documentaires

- Algorithmes des routines du thème "changement de Variables pour les paramètres orbitaux" de la MSLIB; B. Revelin (CS SI); référence MSLIB: M-NT-0-97-CIS.

Code retour

pm_OK	(0) : Retour normal.
pm_warn_conv_identite	(+1802): Conversion identité: les types d'anomalie en entrée et en sortie sont les mêmes.
pm_err_type_anom	(-1811) : La valeur donnée pour le type d'anomalie est incorrecte.
pm_err_anom_v_incompatible_e	(-1214) : L'anomalie vraie v est incompatible avec l'excentricité de l'hyperbole: $\cos(v) < -1/e$.

Pour tout autre code retour, se reporter à l'*annexe 2* du volume 3 "*Caractéristiques principales et conventions d'utilisation de la MSLIB Fortran 90*"; M-MU-0-103-CIS".

Exemple de résultats

Pour la description des types dérivés, se reporter à 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:

e	= 0.5_pm_reel
type_anom1	= pm_anom_E
anom1	= 1.4_pm_reel
type_anom2	= pm_anom_v

Résultats attendus:

anom2	= 1.94
code_retour%valeur	= 0

Routine mv_equa_car

Identification

“Passage des paramètres orbitaux dits adaptés aux orbites équatoriales non circulaires aux paramètres cartésiens”.

Rôle

Passage des paramètres orbitaux (a ou p , e , $\omega + \Omega$, i_x , i_y , M) aux paramètres cartésiens (x , y , z , \dot{x} , \dot{y} , \dot{z}) dans le cas d’une orbite circulaire, elliptique, parabolique ou hyperbolique, et non équatoriale rétro-grade. Le jacobien de la transformation est calculé en option.

Séquence d’appel

(voir explications dans le volume 3)

call mv_equa_car (mu, equa, pos_car, vit_car, code_retour [, jacob])

Description des arguments

(voir explications dans le volume 3)

• Entrées obligatoires

pm_reel	mu	constante de la gravitation ($\text{m}^3.\text{s}^{-2}$).
tm_orb_equa	equa	paramètres orbitaux (a ou p , e , $\omega + \Omega$, i_x , i_y , M) (m, rad.).

• Sorties obligatoires

pm_reel(3)	pos_car	position en coordonnées cartésiennes, $\vec{X} = (x, y, z)$ (m)
pm_reel(3)	vit_car	vitesse en coordonnées cartésiennes, $\dot{\vec{X}} = (\dot{x}, \dot{y}, \dot{z})$ (m.s^{-1})
tm_code_retour	code_retour	

• Sorties facultatives

pm_reel(6,6)	[jacob]	jacobien de la transformation
$\frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})}{\partial(a, e, \omega + \Omega, i_x, i_y, M)} \text{ ou } \frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})}{\partial(p, e, \omega + \Omega, i_x, i_y, M)}$		

Conditions sur les arguments

- La norme du vecteur inclinaison (i_x, i_y) doit être strictement inférieure à 2.
- Dans le cas elliptique ou hyperbolique, le jacobien vaut: .

$$jacob(i, j) = \frac{\partial(\text{parametre } x, y, z, \dot{x}, \dot{y}, \dot{z} \text{ } n^{\circ} i)}{\partial(\text{parametre } a, e, \omega + \Omega, i_x, i_y, M \text{ } n^{\circ} j)}$$

Dans le cas parabolique, le jacobien vaut:

$$jacob(i, j) = \frac{\partial(\text{parametre } x, y, z, \dot{x}, \dot{y}, \dot{z} \text{ } n^{\circ} i)}{\partial(\text{parametre } p, e, \omega + \Omega, i_x, i_y, M \text{ } n^{\circ} j)}$$

Exemple:

$$jacob(4, 1) = \frac{\partial \dot{x}}{\partial a} \text{ ou } \frac{\partial \dot{x}}{\partial p}$$

Notes d'utilisation

- Applicable aux orbites circulaires.
- Applicable aux trois types de coniques (ellipse, hyperbole, parabole).
- Applicable aux orbites équatoriales non rétrogrades et aux orbites non équatoriales.
- La transformation inverse peut se faire par **mv_car_equa**.

Références documentaires

- Algorithmes des routines du thème "changement de Variables pour les paramètres orbitaux" de la MSLIB; B. Revelin (CS SI); référence MSLIB: M-NT-0-97-CIS.

Code retour (voir explications dans le volume 3)

pm_OK

(0) : Retour normal.

pm_err_mu_negatif

(-1001) : La constante gravitationnelle est négative.

pm_err_mu_nul	(-1002) : La constante gravitationnelle est proche de 0.
pm_err_a_negatif	(-1101) : Le demi-grand axe (a) ou le paramètre (p) de la parabole est négatif.
pm_err_a_nul	(-1102) : Le demi-grand axe (a) ou le paramètre (p) de la parabole est proche de 0 .
pm_err_e_negatif	(-1201) : L'excentricité (e) est négative.
pm_err_ix_iy_sup2	(-1306) : La norme du vecteur inclinaison est trop grande (supérieure à 2).
pm_err_i_equa_retro	(-1307) : L'inclinaison (i) est égale à pi.
pm_err_conv_kepler_ellip	(-1902) : L'algorithme itératif utilisé pour la résolution de l'équation de Kepler (orbite elliptique) n'a pas réussi à converger vers la bonne solution. Contacter l'assistance utilisateur MSLIB.
pm_err_conv_kepler_hyperb	(-1903) : L'algorithme itératif utilisé pour la résolution de l'équation de Kepler (orbite hyperbolique) n'a pas réussi à converger vers la bonne solution. Contacter l'assistance utilisateur MSLIB.
pm_warn_e_parab	(+1206) : L'excentricité (e) est proche de 1 : les calculs ont été faits en considérant que l'orbite était parabolique.

Exemple en Fortran 90 portable

(voir explications dans le volume 3)

```

program CHANGE_VAR

  use mslib

  real(pm_reel)                :: MU
  type(tm_orb_equa)            :: EQUA
  real(pm_reel), dimension(3)  :: POS_CAR
  real(pm_reel), dimension(3)  :: VIT_CAR
  type(tm_code_retour)         :: CODE_RETOUR

  MU      = .39860047e15_pm_reel
  EQUA%a  = 24431271.47_pm_reel
  EQUA%e  = .73075_pm_reel
  EQUA%pgom = 5.49_pm_reel
  EQUA%ix  = -1.267e-3_pm_reel
  EQUA%iy  = 1.199e-3_pm_reel
  EQUA%M   = .013_pm_reel

  call mv_equa_car ( MU, EQUA, POS_CAR, VIT_CAR, CODE_RETOUR)

```

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

```
end program CHANGE_VAR
```

Résultats attendus:

```
POS_CAR(1)  = .517 10+7  
POS_CAR(2)  = -.410 10+7  
POS_CAR(3)  = -.998 10+3  
VIT_CAR(1)  = .676 10+4  
VIT_CAR(2)  = .767 10+4  
VIT_CAR(3)  = -.178 10+2
```

```
CODE_RETOUR%valeur  = 0  
CODE_RETOUR%routine = 1041
```

Routine mv_equa_kep

Identification

“Passage des paramètres orbitaux dits adaptés aux orbites équatoriales non circulaires aux paramètres képlériens.”

Rôle

Passage des paramètres orbitaux dits adaptés aux orbites équatoriales non circulaires (a ou p , e , $\omega + \Omega$, i_x , i_y , M) aux paramètres képlériens (a ou p , e , i , ω , Ω , M).

Le calcul du jacobien de la transformation $\frac{\partial(a \text{ ou } p, e, i, \omega, \Omega, M)}{\partial(a \text{ ou } p, e, \omega + \Omega, i_x, i_y, M)}$ est optionnel.

Séquence d'appel

call mv_equa_kep (equa, kep, code_retour [, jacob])

Description des arguments

• Entrées obligatoires

tm_orb_equa **equa** paramètres orbitaux (a ou p , e , $\omega + \Omega$, i_x , i_y , M) (m, rad.)

• Sorties obligatoires

tm_orb_kep **kep** paramètres képlériens (a ou p , e , i , ω , Ω , M) (m, rad.)

tm_code_retour **code_retour**

• Sorties facultatives

pm_reel(6,6) [**jacob**] jacobienne de la transformation $\frac{\partial(a \text{ ou } p, e, i, \omega, \Omega, M)}{\partial(a \text{ ou } p, e, \omega + \Omega, i_x, i_y, M)}$

Conditions sur les arguments

Sans objet

Notes d'utilisation

- Applicable aux orbites circulaires, elliptiques, paraboliques ou hyperboliques.
- Applicable aux orbites équatoriales ou non.
- Si l'orbite est équatoriale ($\sin(i) \approx 0$), la longitude du nœud ascendant Ω est indéfinie. Arbitrairement, nous lui donnons la valeur: $\Omega = 0$. De plus, la jacobienne n'est pas calculable.
- La transformation inverse peut être effectuée par **mv_kep_equa**.

Références documentaires

- Algorithmes des routines du thème "changement de Variables pour les paramètres orbitaux" de la MSLIB; B. Revelin (CS SI); référence MSLIB: M-NT-0-97-CIS.

Code retour

pm_OK	(0) : Retour normal.
pm_err_ix_iy_sup2	(-1306) : La norme du vecteur inclinaison est trop grande (supérieure à 2).
pm_warn_i_equa	(+1302) : $\sin(i)$ est proche de 0; l'orbite est équatoriale ($i=0$ ou $i=\pi$).
pm_err_jac_non_calc_i_equa	(-1308) : L'orbite est équatoriale ($i=0$ ou π). Il en résulte que la jacobienne n'est pas calculable.

Exemple de résultats

Pour la description des types dérivés, se reporter à 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.

Entrées:

equa%a	= 24431271.47_pm_reel
equa%e	= 0.73075_pm_reel
equa%pgom	= 5.490_pm_reel
equa%ix	= -1.267e-03_pm_reel
equa%iy	= 1.199e-03_pm_reel
equa%M	= 0.0130_pm_reel

Résultats attendus:

kep%a	= 24431271.47
kep%e	= 0.73075
kep%i	= 1.74438 10 ⁻³
kep%pom	= 3.1062374
kep%gom	= 2.383762
kep%M	= 1.30 10 ⁻²

code_retour% valeur	= 0
---------------------	-----

Routine mv_kep_car

Identification

“Passage des paramètres **képlériens** aux paramètres **cartésiens**”.

Rôle

Passage des paramètres képlériens (a ou p , e , i , ω , Ω , M) aux paramètres cartésiens (x , y , z , \dot{x} , \dot{y} , \dot{z}) pour une orbite circulaire, elliptique, parabolique ou hyperbolique et équatoriale ou non. Le jacobien de la transformation est calculé en option.

Séquence d'appel

(voir explications dans le volume 3)

call mv_kep_car (mu, kep, pos_car, vit_car, code_retour [, jacob])

Description des arguments

(voir explications dans le volume 3)

• Entrées obligatoires

pm_reel	mu	constante de la gravitation ($\text{m}^3.\text{s}^{-2}$).
tm_orb_kep	kep	paramètres képlériens (a ou p , e , i , ω , Ω , M) (m, rad.).

• Sorties obligatoires

pm_reel(3)	pos_car	position en coordonnées cartésiennes, $\vec{X} = (x, y, z)$ (m)
pm_reel(3)	vit_car	vitesse en coordonnées cartésiennes, $\vec{\dot{X}} = (\dot{x}, \dot{y}, \dot{z})$ (m.s^{-1})
tm_code_retour	code_retour	

• Sorties facultatives

pm_reel(6,6)	[jacob]	jacobien de la transformation : $\frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})}{\partial(a, e, i, \omega, \Omega, M)} \text{ ou } \frac{\partial(x, y, z, \dot{x}, \dot{y}, \dot{z})}{\partial(p, e, i, \omega, \Omega, M)}$
--------------	------------------	--

Conditions sur les arguments

- Dans le cas elliptique ou hyperbolique, le jacobien vaut: .

$$jacob(i, j) = \frac{\partial(\text{parametre } x, y, z, \dot{x}, \dot{y}, \dot{z} \text{ } n^{\circ} i)}{\partial(\text{parametre } a, e, i, \omega, \Omega, M \text{ } n^{\circ} j)}$$

Dans le cas parabolique, le jacobien vaut:

$$jacob(i, j) = \frac{\partial(\text{parametre } x, y, z, \dot{x}, \dot{y}, \dot{z} \text{ } n^{\circ} i)}{\partial(\text{parametre } p, e, i, \omega, \Omega, M \text{ } n^{\circ} j)}$$

Exemple:

$$jacob(4, 1) = \frac{\partial \dot{x}}{\partial a} \text{ ou } \frac{\partial \dot{x}}{\partial p}$$

Notes d'utilisation

- Applicable avec une orbite circulaire, elliptique, parabolique ou hyperbolique.
- Applicable avec une orbite équatoriale ou non.
- La transformation inverse peut se faire par **mv_car_kep**.

Références documentaires

- Algorithmes des routines du thème "changement de Variables pour les paramètres orbitaux" de la MSLIB; B. Revelin (CS SI); référence MSLIB: M-NT-0-97-CIS.

Code retour (voir explications dans le volume 3)

pm_OK

(0) : Retour normal.

pm_err_mu_negatif

(-1001) : La constante gravitationnelle est négative.

pm_err_mu_nul	(-1002) : La constante gravitationnelle est proche de 0.
pm_err_a_negatif	(-1101) : Le demi-grand axe (a) ou le paramètre (p) de la parabole est négatif.
pm_err_a_nul	(-1102) : Le demi-grand axe (a) ou le paramètre (p) de la parabole est proche de 0 .
pm_err_e_negatif	(-1201) : L'excentricité (e) est négative.
pm_err_i_negatif	(-1301) : L'inclinaison (i) est négative.
pm_err_i_sup_pi	(-1305) : L'inclinaison (i) est supérieure à pi.
pm_err_conv_kepler_ellip	(-1902) : L'algorithme itératif utilisé pour la résolution de l'équation de Kepler (orbite elliptique) n'a pas réussi à converger vers la bonne solution. Contacter l'assistance utilisateur MSLIB.
pm_err_conv_kepler_hyperb	(-1903) : L'algorithme itératif utilisé pour la résolution de l'équation de Kepler (orbite hyperbolique) n'a pas réussi à converger vers la bonne solution. Contacter l'assistance utilisateur MSLIB.
pm_warn_e_parab	(+1206) : L'excentricité (e) est proche de 1 : les calculs ont été faits en considérant que l'orbite était parabolique.

Exemple en Fortran 90 portable

(voir explications dans le volume 3)

```

program CHANGE_VAR
  use mslib
  real(pm_reel)                :: MU
  type(tm_orb_kep)             :: KEP
  real(pm_reel), dimension(3)  :: POS_CAR
  real(pm_reel), dimension(3)  :: VIT_CAR
  type(tm_code_retour)         :: CODE_RETOUR
  real(pm_reel), dimension(6,6) :: JACOB

  MU      = .39860047e15_pm_reel
  KEP%a   = 24464560._pm_reel
  KEP%e   = .7311_pm_reel
  KEP%i   = .122138_pm_reel
  KEP%pom = 3.10686_pm_reel
  KEP%gom = 1.00681_pm_reel
  KEP%M   = .048363_pm_reel

  call mv_kep_car ( MU, KEP, POS_CAR, VIT_CAR, CODE_RETOUR,      &
                  jacob= JACOB)

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

```

end program CHANGE_VAR

Résultats attendus:

POS_CAR(1) = -.108 10⁺⁷
POS_CAR(2) = -.677 10⁺⁷
POS_CAR(3) = -.332 10⁺⁶
VIT_CAR(1) = .936 10⁺⁴
VIT_CAR(2) = -.331 10⁺⁴
VIT_CAR(3) = -.119 10⁺⁴

JACOB =

$$\begin{bmatrix} -0.440 \cdot 10^{-1} & -0.277 & -0.136 \cdot 10^{-1} & -0.191 \cdot 10^{-3} & 0.677 \cdot 10^{-4} & 0.243 \cdot 10^{-4} \\ 0.200 \cdot 10^8 & 0.193 \cdot 10^8 & -0.811 \cdot 10^6 & 0.218 \cdot 10^5 & 0.580 \cdot 10^4 & -0.188 \cdot 10^4 \\ -0.281 \cdot 10^6 & 0.178 \cdot 10^6 & -0.271 \cdot 10^7 & -0.100 \cdot 10^4 & 0.635 \cdot 10^3 & -0.968 \cdot 10^4 \\ 0.674 \cdot 10^7 & -0.103 \cdot 10^7 & -0.767 \cdot 10^6 & 0.337 \cdot 10^4 & 0.941 \cdot 10^4 & 0.268 \cdot 10^3 \\ 0.677 \cdot 10^7 & -0.108 \cdot 10^7 & 0. & 0.331 \cdot 10^4 & 0.936 \cdot 10^4 & 0. \\ 0.567 \cdot 10^8 & -0.201 \cdot 10^8 & -0.720 \cdot 10^7 & 0.806 \cdot 10^4 & 0.507 \cdot 10^5 & 0.249 \cdot 10^4 \end{bmatrix}$$

CODE_RETOUR%valeur = 0
CODE_RETOUR%routine = 1031

Routine mv_kep_cir

Identification

“Passage des paramètres **képlériens** aux paramètres orbitaux dits adaptés aux orbites **circulaires** non équatoriales.”

Rôle

Passage des paramètres képlériens $(a, e, i, \omega, \Omega, M)$ aux paramètres orbitaux dits adaptés aux orbites circulaires non équatoriales $(a, e_x, e_y, i, \Omega, \omega+M)$.

Le calcul du jacobien de la transformation $\frac{\partial(a, e_x, e_y, i, \Omega, \omega + M)}{\partial(a, e, i, \omega, \Omega, M)}$ est optionnel.

Séquence d'appel

call mv_kep_cir (kep, cir, code_retour [, jacob])

Description des arguments

• Entrées obligatoires

tm_orb_kep **kep** paramètres képlériens $(a, e, i, \omega, \Omega, M)$ (m, rad.)

• Sorties obligatoires

tm_orb_cir **cir** paramètres orbitaux $(a, e_x, e_y, i, \Omega, \omega+M)$ (m, rad.)

tm_code_retour **code_retour**

• Sorties facultatives

pm_reel(6,6) [**jacob**] jacobienne de la transformation $\frac{\partial(a, e_x, e_y, i, \Omega, \omega + M)}{\partial(a, e, i, \omega, \Omega, M)}$

Conditions sur les arguments

Sans objet

Notes d'utilisation

- Applicable aux orbites circulaires ou elliptiques.
- Applicable aux orbites équatoriales ou non.
- La transformation inverse peut être effectuée par **mv_cir_kep**.

Références documentaires

- Algorithmes des routines du thème "changement de Variables pour les paramètres orbitaux" de la MSLIB; B. Revelin (CS SI); référence MSLIB: M-NT-0-97-CIS.

Code retour

pm_OK (0) : Retour normal.

Exemple de résultats

Pour la description des types dérivés, se reporter à 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.

Entrées:

kep%a	= 24464560._pm_reel
kep%e	= 0.7311_pm_reel
kep%i	= 0.122138_pm_reel
kep%pom	= 3.10686_pm_reel
kep%gom	= 1.00681_pm_reel
kep%M	= 0.048363_pm_reel

Résultats attendus:

cir%a	= 24464560.
cir%ex	= -0.731
cir%ey	= 0.254 10 ⁻¹
cir%i	= 0.122
cir%gom	= 1.009
cir%pso_M	= 3.155

code_retour%valeur = 0

Routine mv_kep_cir_equa

Identification

“Passage des paramètres **képlériens** aux paramètres orbitaux dits adaptés aux orbites **circulaires équatoriales**.”

Rôle

Passage des paramètres képlériens $(a, e, i, \omega, \Omega, M)$ aux paramètres orbitaux dits adaptés aux orbites circulaires équatoriales $(a, \tilde{e}_x, \tilde{e}_y, i_x, i_y, \omega + \Omega + M)$

Le calcul du jacobien de la transformation $\frac{\partial(a, \tilde{e}_x, \tilde{e}_y, i_x, i_y, \omega + \Omega + M)}{\partial(a, e, i, \omega, \Omega, M)}$ est optionnel.

Séquence d'appel

call mv_kep_cir_equa (kep, cir_equa, code_retour [, jacob])

Description des arguments

• Entrées obligatoires

tm_orb_kep **kep** paramètres képlériens $(a, e, i, \omega, \Omega, M)$ (m, rad.)

• Sorties obligatoires

tm_orb_cir_equa **cir_equa** paramètres orbitaux $(a, \tilde{e}_x, \tilde{e}_y, i_x, i_y, \omega + \Omega + M)$ (m, rad.)

tm_code_retour **code_retour**

• Sorties facultatives

pm_reel(6,6) [**jacob**] jacobienne de la transformation $\frac{\partial(a, \tilde{e}_x, \tilde{e}_y, i_x, i_y, \omega + \Omega + M)}{\partial(a, e, i, \omega, \Omega, M)}$

Conditions sur les arguments

Sans objet

Notes d'utilisation

- Applicable aux orbites circulaires ou elliptiques.
- Applicable aux orbites équatoriales ou non.
- La transformation inverse peut être effectuée par **mv_cir_equa_kep**.

Références documentaires

- Algorithmes des routines du thème "changement de Variables pour les paramètres orbitaux" de la MSLIB; B. Revelin (CS SI); référence MSLIB: M-NT-0-97-CIS.

Code retour

pm_OK (0) : Retour normal.

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.

Entrées:

kep%a	= 24464560._pm_reel
kep%e	= 0.7311_pm_reel
kep%i	= 0.122138_pm_reel
kep%pom	= 3.10686_pm_reel
kep%gom	= 1.00681_pm_reel
kep%M	= 0.048363_pm_reel

Résultats attendus:

cir_equa%a	= 24464560.
cir_equa%ex	= -0.412
cir_equa%ey	= -0.604
cir_equa%ix	= 6.525 10 ⁻²
cir_equa%iy	= 0.103
cir_equa%pso_M	= 4.162

code_retour%valeur = 0

Routine mv_kep_equa

Identification

“Passage des paramètres **képlériens** aux paramètres orbitaux dits adaptés aux orbites **équatoriales** non circulaires.”

Rôle

Passage des paramètres képlériens (a ou p , e , i , ω , Ω , M) aux paramètres orbitaux dits adaptés aux orbites équatoriales non circulaires (a ou p , e , $\omega + \Omega$, i_x , i_y , M).

Le calcul du jacobien de la transformation $\frac{\partial(a \text{ ou } p, e, \omega + \Omega, i_x, i_y, M)}{\partial(a \text{ ou } p, e, i, \omega, \Omega, M)}$ est optionnel.

Séquence d'appel

call mv_kep_equa (kep, equa, code_retour [, jacob])

Description des arguments

• Entrées obligatoires

tm_orb_kep **kep** paramètres képlériens (a ou p , e , i , ω , Ω , M) (m, rad.)

• Sorties obligatoires

tm_orb_equa **equa** paramètres orbitaux (a ou p , e , $\omega + \Omega$, i_x , i_y , M) (m, rad.)

tm_code_retour **code_retour**

• Sorties facultatives

pm_reel(6,6) [**jacob**] jacobienne de la transformation $\frac{\partial(a \text{ ou } p, e, \omega + \Omega, i_x, i_y, M)}{\partial(a \text{ ou } p, e, i, \omega, \Omega, M)}$

Conditions sur les arguments

Sans objet

Notes d'utilisation

- Applicable aux orbites circulaires, elliptiques, paraboliques ou hyperboliques.
- Applicable aux orbites équatoriales ou non.
- La transformation inverse peut être effectuée par **mv_equa_kep**.

Références documentaires

- Algorithmes des routines du thème "changement de Variables pour les paramètres orbitaux" de la MSLIB; B. Revelin (CS SI); référence MSLIB: M-NT-0-97-CIS.

Code retour

pm_OK (0) : Retour normal.

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.

Entrées:

kep%a	= 24464560._pm_reel
kep%e	= 0.7311_pm_reel
kep%i	= 0.122138_pm_reel
kep%pom	= 3.10686_pm_reel
kep%gom	= 1.00681_pm_reel
kep%M	= 0.048363_pm_reel

Résultats attendus:

equa%a	= 24464560.
equa%e	= 0.731
equa%pgom	= 4.114
equa%ix	= 6.525 10 ⁻²
equa%iy	= 0.103
equa%M	= 4.836 10 ⁻²

code_retour%valeur	= 0
--------------------	-----

Routine mv_kepler_bar

Identification

“Résolution des équations de **Kepler** et de **Barker**”.

Rôle

- Résolution itérative des équations de Kepler :

$$M = E - e \sin E \text{ (orbite elliptique)}$$

$$M = e \sinh E - E \text{ (orbite hyperbolique)}$$

par la méthode de Newton .

- Résolution de l'équation de Barker:

$$6M = 3D + D^3 \text{ (orbite parabolique)}$$

avec:

M : anomalie moyenne

e : excentricité

E : anomalie excentrique

D : tangente (anomalie vraie/2)

Séquence d'appel

(voir explications dans le volume 3)

call mv_kepler_bar (anom_M, e, anom_E_D, code_retour)

Description des arguments

(voir explications dans le volume 3)

- Entrées obligatoires

pm_reel	anom_M	anomalie moyenne M (rad.)
---------	---------------	-----------------------------

pm_reel	e	excentricité e
---------	----------	------------------

- Sorties obligatoires

pm_reel	anom_E_D	anomalie excentrique E (cas elliptique et hyperbolique) ou D (cas parabolique) (rad.).
---------	-----------------	--

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

Conditions sur les arguments

Sans objet.

Notes d'utilisation

La sortie **anom_E_D** vaut **D** lorsque la valeur du code retour vaut **pm_warn_e_parab** (voir rubrique "Code retour").

Références documentaires

- Algorithmes des routines du thème "changement de Variables pour les paramètres orbitaux" de la MSLIB; B. Revelin (CS SI); référence MSLIB: M-NT-0-97-CIS.
- Résolution numérique de l'équation de Kepler; R. Epenoy (Cnes); référence MSLIB: M-ST-0-90-CN.

Code retour (voir explications dans le volume 3)

pm_OK	(0) : Retour normal.
pm_err_e_negatif	(-1201) : L'excentricité (e) est négative.
pm_err_conv_kepler_ellip	(-1902) : L'algorithme itératif utilisé pour la résolution de l'équation de Kepler (orbite elliptique) n'a pas réussi à converger vers la bonne solution. Contacter l'assistance utilisateur MSLIB.
pm_err_conv_kepler_hyperb	(-1903) : L'algorithme itératif utilisé pour la résolution de l'équation de Kepler (orbite hyperbolique) n'a pas réussi à converger vers la bonne solution. Contacter l'assistance utilisateur MSLIB.
pm_warn_e_parab	(+1206) : L'excentricité (e) est proche de 1 : les calculs ont été faits en considérant que l'orbite était parabolique.

Exemple en Fortran 90 portable

(voir explications dans le volume 3)

```
program CHANGE_VAR

  use mslib

  real(pm_reel)          :: ANOM_M
  real(pm_reel)          :: E
  real(pm_reel)          :: ANOM_E_D
  type(tm_code_retour)    :: CODE_RETOUR

  ANOM_M = .57327_pm_reel
  E      = 4._pm_reel

  call mv_kepler_bar ( ANOM_M, E, ANOM_E_D, CODE_RETOUR)

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

end program CHANGE_VAR
```

Résultats attendus:

ANOM_E_D = .190

CODE_RETOUR%valeur = 0

CODE_RETOUR%routine = 1024

Routine mv_kepler_gene

Identification

“Résolution de l’équation de **Kepler** généralisée”.

Rôle

Résolution de l’équation de Kepler généralisée:

$$PSO_M = PSO_E - E_x \sin (PSO_E) + E_y \cos (PSO_E)$$

avec :

- Orbite circulaire ou elliptique et NON équatoriale:

$$\begin{aligned} PSO_M &= \omega + M \\ (E_x, E_y) &= (e_x, e_y) = (e \cos \omega, e \sin \omega) \\ PSO_E &= \omega + E \end{aligned}$$

- Orbite circulaire ou elliptique et équatoriale ou non:

$$\begin{aligned} PSO_M &= \omega + \Omega + M \\ (E_x, E_y) &= (\tilde{e}_x, \tilde{e}_y) = (e \cos(\omega + \Omega), e \sin(\omega + \Omega)) \\ PSO_E &= \omega + \Omega + E \end{aligned}$$

L’algorithme résout itérativement l’équation par la méthode de Newton .

Séquence d’appel

(voir explications dans le volume 3)

call mv_kepler_gene (pso_M, ex, ey, pso_E, code_retour)

Description des arguments

(voir explications dans le volume 3)

- Entrées obligatoires

pm_reel	pso_M	position sur orbite (PSO_M) , $\omega + M$ ou $\omega + \Omega + M$ (rad.)
pm_reel	ex	composante sur x du vecteur excentricité (E_x) , e_x ou \tilde{e}_x
pm_reel	ey	composante sur y du vecteur excentricité (E_y) , e_y ou \tilde{e}_y

- Sorties obligatoires

pm_reel	pso_E	position sur orbite (PSO_E) , $\omega + E$ ou $\omega + \Omega + E$ (rad.)
tm_code_retour	code_retour	

Conditions sur les arguments

Sans objet.

Notes d'utilisation

Ne s'applique pas aux orbites paraboliques ou hyperboliques.

Références documentaires

- Algorithmes des routines du thème "changement de Variables pour les paramètres orbitaux" de la MSLIB; B. Revelin (CS SI); référence MSLIB: M-NT-0-97-CIS.
- Résolution numérique de l'équation de Kepler; R. Epenoy (Cnes); référence MSLIB: M-ST-0-90-CN.

Code retour

(voir explications dans le volume 3)

pm_OK	(0) : Retour normal.
pm_err_e_non_ellip	(-1208) : L'excentricité (e) n'appartient pas à l'intervalle [0, 1 [; l'orbite n'est pas elliptique.
pm_err_conv_kepler_gene	(-1904) : L'algorithme itératif utilisé pour la résolution de l'équation de Kepler généralisée (équation avec le vecteur excentricité) n'a pas réussi à converger vers la bonne solution. Contacter l'assistance utilisateur MSLIB.

Exemple en Fortran 90 portable

(voir explications dans le volume 3)

```
program CHANGE_VAR

  use mslib

  real(pm_reel)                :: PSO_M
  real(pm_reel)                :: EX
  real(pm_reel)                :: EY
  real(pm_reel)                :: PSO_E
  type(tm_code_retour)         :: CODE_RETOUT
```

```
PSO_M = 1.2566371_pm_reel  
EX     = 0.005_pm_reel  
EY     = 0.005_pm_reel
```

```
call mv_kepler_gene ( PSO_M, EX, EY, PSO_E, CODE_RETOUR)
```

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

```
end program CHANGE_VAR
```

Résultats attendus:

PSO_E = .126 10⁺¹

CODE_RETOUR%valeur = 0

CODE_RETOUR%routine = 1023

Routine mv_kepler_std

Identification

“Résolution de l’équation de **Kepler standard** (orbite elliptique)”.

Rôle

Résolution de l’équation de Kepler :

$$M = E - e \sin E$$

avec:

M : anomalie moyenne

E : anomalie excentrique

e : excentricité

L’algorithme résout itérativement l’équation par la méthode de Newton.

Séquence d’appel

(voir explications dans le volume 3)

call mv_kepler_std (anom_M, e, anom_E, code_retour)

Description des arguments

(voir explications dans le volume 3)

• Entrées obligatoires

pm_reel	anom_M	anomalie moyenne M (rad.)
pm_reel	e	excentricité e

• Sorties obligatoires

pm_reel	anom_E	anomalie excentrique E (rad.)
tm_code_retour	code_retour	

Conditions sur les arguments

Sans objet.

Notes d’utilisation

Sans objet.

Références documentaires

- Algorithmes des routines du thème "changement de Variables pour les paramètres orbitaux" de la MSLIB; B. Revelin (CS SI); référence MSLIB: M-NT-0-97-CIS.
- Résolution numérique de l'équation de Kepler; R. Epenoy (Cnes); référence MSLIB: M-ST-0-90-CN.

Code retour (voir explications dans le volume 3)

pm_OK (0) : Retour normal.

pm_err_e_negatif (-1201) : L'excentricité (e) est négative.

pm_err_e_non_ellip (-1208) : L'excentricité (e) n'appartient pas à l'intervalle [0, 1 [; l'orbite n'est pas elliptique.

pm_err_conv_kepler_ellip (-1902) : L'algorithme itératif utilisé pour la résolution de l'équation de Kepler (orbite elliptique) n'a pas réussi à converger vers la bonne solution. Contacter l'assistance utilisateur MSLIB.

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

```
program CHANGE_VAR
  use mslib
  real(pm_reel) :: ANOM_M
  real(pm_reel) :: E
  real(pm_reel) :: ANOM_E
  type(tm_code_retour) :: CODE_RETOUR

  ANOM_M = 2.618_pm_reel
  E = 0.9_pm_reel

  call mv_kepler_std ( ANOM_M, E, ANOM_E, CODE_RETOUR)

  ! appel a la routine utilisateur d'ecriture des resultats
  call WRITE_RESULTATS (ANOM_E, CODE_RETOUR)
end program CHANGE_VAR
```

Résultats attendus:

ANOM_E = .286 10⁺¹

CODE_RETOUR%valeur = 0

CODE_RETOUR%routine = 1014