

MSLIB Fortran 90

CS SI

Nomenclature : **M-MU-0-121-CIS**

Edition : 01 Date: 01/02/2000

Révision: 00 Date: 01/02/2000

Volume M

Manœuvres et transferts orbitaux

Rédigé par : Sylvain VRESK	le : CS SI	
Validé par : Guylaine PRAT	le : CS SI	
Pour application : Eric LE DÉ	le : Cnes (DTS/MPI/MS/MN)	

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 M - Manœuvres et transferts orbitaux**AUTEUR :** Sylvain VRESK**RESUME :**

Ce document rassemble les notices d'utilisation des routines du thème "Manœuvres et transferts orbitaux".

SITUATION DU DOCUMENT : Création**VOLUME :****PAGES :** 16**PLANCHES :****FIGURES :****LANGUES :** F**CONTRAT :** Marché 870/96/Cnes/0720 BC150 L23**SYSTEME HOTE :** Frame5/MSLIB

MODIFICATION

ETAT DOCUMENT				PAGES REVISEES	
ED.	REV.	DATE	REFERENCE ORIGINE (pour chaque édition)	ETAT PAGE *	NUMERO DES PAGES
01	00	01/02/00	M-MU-0-121-CIS Rédacteur : S. Vresk avec la participation de G. Prat		Création

* I = Inséré

S = Supprimé

M = Modifié

Sommaire

Présentation du thème M :..... *page 1*

Notations *page 2*

Index. *page 3*

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

Liste des routines du thème M:

mm_impul_car : *page 4*

“Pour un satellite, calcul du bulletin d’orbite en coordonnées cartésiennes,
suite à la réalisation d’une manœuvre orbitale modélisée par une mono-
impulsion quelconque”.

mm_impul_kep : *page 7*

“Pour un satellite, calcul du bulletin d’orbite en paramètres képlériens suite
à la réalisation d’une manœuvre orbitale modélisée par une mono-
impulsion quelconque”.

Présentation du thème M

Le thème “*Manœuvres et transferts orbitaux*” regroupe une série de routines de calculs de manœuvres et de transferts orbitaux.

Notations

Sans objet.

Index

Sans objet.

Routine mm_impul_car

Identification

“Pour un satellite, calcul du bulletin d’orbite en coordonnées **cartésiennes**, suite à la réalisation d’une manœuvre orbitale modélisée par une mono-**impulsion** quelconque”.

Rôle

Calcule le bulletin d’orbite d’un satellite après la réalisation d’une manœuvre orbitale modélisée par une mono-impulsion quelconque. L’utilisateur choisit le repère dans lequel il veut donner les composantes de l’impulsion : repère géocentrique inertiel, repère (t , n , w) ou repère (q , s , w).
Nota : par définition, la position d’un satellite est constante avant et après impulsion.

Séquence d’appel

(voir explications dans le volume 3)

call mm_impul_car (pos_car, vit_car_avant, rep, impul, vit_car_apres, code_retour)

Description des arguments

(voir explications dans le volume 3)

• Entrées obligatoires

pm_reel(3)	pos_car	position \vec{P} , du satellite, au moment de l’impulsion (coordonnées cartésiennes dans le repère géocentrique inertiel)(m)
pm_reel(3)	vit_car_avant	vitesse \vec{V}_{avant} , du satellite, avant l’impulsion (coordonnées cartésiennes dans le repère géocentrique inertiel) (m.s ⁻¹)
integer	rep	indicateur du repère dans lequel est défini le vecteur impulsion.
pm_reel(3)	impul	vecteur impulsion $\Delta \vec{V}$ (m.s ⁻¹)

• Sorties obligatoires

pm_reel(3)	vit_car_apres	vitesse \vec{V}_{apres} , du satellite, après l’impulsion (coordonnées cartésiennes dans le repère géocentrique inertiel)(m.s ⁻¹)
tm_code_retour	code_retour	

Conditions sur les arguments

Sans objet.

Notes d'utilisation

Selon le calcul souhaité, l'argument **rep** sera initialisé par l'appelant à la valeur de l'un des paramètres suivants de la MSLIB : **pm_rep_geo** si le vecteur impulsion est donné en repère géocentrique inertiel, **pm_rep_qsw** si le vecteur impulsion est donné en repère (q , s , w) ou **pm_rep_tnw** si le vecteur impulsion est donné en repère (t , n , w).

Références documentaires

- Algorithmes des routines du thème "Manœuvres et transferts orbitaux" de la MSLIB; S. Vresk (CS SI); référence MSLIB: M-NT-0-343-CIS.

Code retour (voir explications dans le volume 3)

pm_OK	(0) : Retour normal.
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_ind_rep	(-1808) : La valeur donnée pour l'indicateur du repère est incorrecte.

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

```
program MANOEUVRES
  use mslib

  real(pm_reel), dimension(3)      :: POS_CAR
  real(pm_reel), dimension(3)      :: VIT_CAR_AVANT
  integer                          :: REP
  real(pm_reel), dimension(3)      :: IMPUL
  real(pm_reel), dimension(3)      :: VIT_CAR_APRES
```

```
type(tm_code_retour)          :: CODE_RETOUR

POS_CAR(1)      = 22930599.568_pm_reel
POS_CAR(2)      = -28611399.924_pm_reel
POS_CAR(3)      = -1574752.970_pm_reel
VIT_CAR_AVANT(1) = 2321.066_pm_reel
VIT_CAR_AVANT(2) = 21.253_pm_reel
VIT_CAR_AVANT(3) = 146.840_pm_reel
REP             = pm_rep_tnw
IMPUL(1)        = -4._pm_reel
IMPUL(2)        = 1.2_pm_reel
IMPUL(3)        = 0._pm_reel

call mm_impul_car ( POS_CAR, VIT_CAR_AVANT, REP, IMPUL,      &
                   VIT_CAR_APRES, CODE_RETOUR )

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

end program MANOEUVRES
```

Résultats attendus:

```
VIT_CAR_APRES(1)  = .232 104
VIT_CAR_APRES(2)  = .224 102
VIT_CAR_APRES(3)  = .147 103
```

```
CODE_RETOUR%valeur = 0
CODE_RETOUR%routine = 1095
```

Routine mm_impul_kep

Identification

“Pour un satellite, calcul du bulletin d’orbite en paramètres **képlériens** suite à la réalisation d’une manœuvre orbitale modélisée par une mono-**impulsion** quelconque”.

Rôle

Calcule le bulletin képlérien d’un satellite après la réalisation d’une manœuvre orbitale modélisée par une mono-impulsion quelconque. L’utilisateur choisit le repère dans lequel il veut donner les composantes de l’impulsion : repère géocentrique inertiel, repère (t, n, w) ou repère (q, s, w).

Nota : par définition, la position d’un satellite est constante avant et après impulsion.

Séquence d’appel

(voir explications dans le volume 3)

call mm_impul_kep (mu, kep_avant, rep, impul, kep_apres, code_retour)

Description des arguments

(voir explications dans le volume 3)

• Entrées obligatoires

pm_reel	mu	constante gravitationnelle μ ($\text{m}^3.\text{s}^{-2}$)
tm_orb_kep	kep_avant	bulletin képlérien ($a, e, i, \omega, \Omega, M$) _{avant} du satellite avant impulsion (m, rad)
integer	rep	indicateur du repère dans lequel est défini le vecteur impulsion
pm_reel(3)	impul	vecteur impulsion $\Delta \vec{V}$ ($\text{m}.\text{s}^{-1}$)

• Sorties obligatoires

tm_orb_kep	kep_apres	bulletin képlérien ($a, e, i, \omega, \Omega, M$) _{apres} du satellite après impulsion (m, rad)
tm_code_retour	code_retour	

Conditions sur les arguments

Sans objet.

Notes d'utilisation

Selon le calcul souhaité, l'argument **rep** sera initialisé par l'appelant à la valeur de l'un des paramètres suivants de la MSLIB : **pm_rep_geo** si le vecteur impulsion est donné en repère géocentrique inertiel, **pm_rep_qsw** si le vecteur impulsion est donné en repère (q , s , w) ou **pm_rep_tnw** si le vecteur impulsion est donné en repère (t , n , w).

Références documentaires

- Algorithmes des routines du thème "Manœuvres et transferts orbitaux" de la MSLIB; S. Vresk (CS SI); référence MSLIB: M-NT-0-343-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_negatif	(-1201) : L'excentricité (e) est négative.
pm_err_e_circul	(-1202) : L'excentricité (e) est proche de 0 ; l'orbite est circulaire.
pm_err_i_negatif	(-1301) : L'inclinaison (i) est négative.
pm_err_i_equa	(-1302) : $\sin(i)$ est proche de 0 ; l'orbite est équatoriale ($i=0$ ou $i=\pi$).
pm_err_i_sup_pi	(-1305) : L'inclinaison (i) est supérieure à π .
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_ind_rep	(-1808) : La valeur donnée pour l'indicateur du repère est incorrecte.
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_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 MANOEUVRES
  use mslib

  real(pm_reel)                :: MU
  type(tm_orb_kep)             :: KEP_AVANT
  integer                      :: REP
  real(pm_reel), dimension(3)  :: IMPUL
  type(tm_orb_kep)             :: KEP_APRES
  type(tm_code_retour)         :: CODE_RETOUTOUR

  MU = 398600.64e+09_pm_reel
  KEP_AVANT%a = .24435100e+8_pm_reel
  KEP_AVANT%e = .73054_pm_reel
  KEP_AVANT%i = .121468_pm_reel
  KEP_AVANT%pom = .3116285E+1_pm_reel
  KEP_AVANT%gom = .5747544e+1_pm_reel
  KEP_AVANT%M = .179748e+1_pm_reel
  REP = pm_rep_tnw
  IMPUL(1) = -4._pm_reel
  IMPUL(2) = 1.2_pm_reel
  IMPUL(3) = 0._pm_reel

  call mm_impul_kep ( MU, KEP_AVANT, REP, IMPUL, KEP_APRES,      &
                    CODE_RETOUTOUR )

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

```

end program MANOEUVRES

Résultats attendus:

KEP_APRES%a	= .244 10 ⁸
KEP_APRES%e	= .731
KEP_APRES%i	= .121
KEP_APRES%pom	= .311 10 ¹
KEP_APRES%gom	= .575 10 ¹
KEP_APRES%M	= .180 10 ¹

CODE_RETOUR% valeur	= 0
CODE_RETOUR%routine	= 1096