

**MSLIB Fortran 90**

**CS SI**

Nomenclature : **M-MU-0-111-CIS**

Edition : 01      Date: 23/06/1998

Révision: 01      Date: 26/01/2000

**Volume C**

**Constantes**

<b>Rédigé par :</b> Sylvain VRESK	le : CS SI	
<b>Validé par :</b> Guylaine PRAT	le : CS SI	
<b>Pour application :</b> 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 C - Constantes**AUTEUR :** Sylvain VRESK**RESUME :**

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

**SITUATION DU DOCUMENT :** Création**VOLUME :****PAGES :** 18**PLANCHES :****FIGURES :****LANGUES :** F**CONTRAT :** Marché 870/96/Cnes/0720 BC 150 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	23/06/98	M-MU-0-111-CIS Rédacteur : V. Lépine avec la participation de G. Prat	M	Création
01	01	26/01/00	M-MU-0-111-CIS Rédacteur : S. Vresk avec la participation de G. Prat		7, 8, 9, 10, 11, 12

\* I = Inséré

S = Supprimé

M = Modifié

## Sommaire

<b>Présentation du thème C :</b>	<i>page 1</i>
<b>Notations</b>	<i>page 2</i>
<b>Index.</b>	<i>page 3</i>

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

## Liste des routines du thème C:

<b>mc_GRS1980</b> : .....	<i>page 4</i>
“Définition des constantes géodésiques du GRS 1980”.	
<b>mc_math</b> : .....	<i>page 6</i>
“Définition de constantes mathématiques”.	
<b>mc_phys</b> : .....	<i>page 8</i>
“Définition de constantes physiques”.	
<b>mc_test</b> : .....	<i>page 10</i>
“Définition de constantes de tests utilisées par les routines de la MSLIB”.	

---

## Présentation du thème C

Le thème “*Constantes*” regroupe un ensemble de routines qui définissent des constantes mathématiques, physiques, géodésiques, ou de tests.

L'utilisation des constantes, ainsi définies, doit permettre d'atteindre les objectifs suivants:

- **homogénéité** des constantes (au sein d'un même programme utilisateur, et vis à vis de celles utilisées dans la librairie MSLIB),
- utilisation de **constantes validées**, et définies avec une **précision maximale**.

## **Notations**

Sans objet.



## **Index**

Sans objet.

## Routine mc\_GRS1980

### Identification

“Définition des constantes géodésiques du **GRS 1980**”.

### Rôle

Définition des constantes relatives au système géodésique de référence (1980 = SGR 80), à savoir:

- rayon équatorial terrestre du GRS1980 ,
- inverse de l’aplatissement terrestre du GRS1980 ,
- aplatissement terrestre du GRS 1980 .

### Séquence d’appel

(voir explications dans le volume 3)

call mc\_GRS1980 ( code\_retour [ , r\_equa, inv\_apla, apla ] )

### Description des arguments

(voir explications dans le volume 3)

#### • Sorties obligatoires

tm\_code\_retour      **code\_retour**

#### • Sorties facultatives

pm_reel	[ <b>r_equa</b> ]	$a_e$ , rayon équatorial terrestre du GRS1980 (m).
pm_reel	[ <b>inv_apla</b> ]	$1/f$ , inverse de l’aplatissement terrestre du GRS1980.
pm_reel	[ <b>apla</b> ]	$f$ , aplatissement terrestre du GRS1980.

### Conditions sur les arguments

Sans objet.

### Notes d’utilisation

Sans objet.

### Références documentaires

- Algorithmes des routines du thème "Constantes" de la MSLIB; G. Prat (CS SI); référence MSLIB: M-NT-0-91-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 CONSTANTES

  use mslib
  real(pm_reel)      :: R_EQUA
  real(pm_reel)      :: INV_APLA
  real(pm_reel)      :: APLA
  type(tm_code_retour) :: CODE_RETOUR
  call mc_GRS1980 ( CODE_RETOUR,                                &
                   r_equa = R_EQUA, inv_apla = INV_APLA,        &
                   apla = APLA )

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

end program CONSTANTES
```

### Résultats attendus:

R\_EQUA = .638 10<sup>+7</sup>  
INV\_APLA = .298 10<sup>+3</sup>  
APLA = .336 10<sup>-4</sup>

CODE\_RETOUR%valeur = 0  
CODE\_RETOUR%routine = 1018

## Routine *mc\_math*

### Identification

“Définition de constantes **math**ématiques”.

### Rôle

Affectation de constantes mathématiques courantes.

### Séquence d'appel

(voir explications dans le volume 3)

call mc\_math ( code\_retour [ , pi, deux\_pi, pi\_sur2, deg\_rad, rad\_deg ] )

### Description des arguments

(voir explications dans le volume 3)

- Sorties obligatoires

tm\_code\_retour      **code\_retour**

- Sorties facultatives

pm_reel	[ <b>pi</b> ]	$\pi$
pm_reel	[ <b>deux_pi</b> ]	$2\pi$
pm_reel	[ <b>pi_sur2</b> ]	$\pi/2$
pm_reel	[ <b>deg_rad</b> ]	$\pi/180$
pm_reel	[ <b>rad_deg</b> ]	$180/\pi$

### Conditions sur les arguments

Sans objet.

### Notes d'utilisation

Sans objet.

### Références documentaires

- Algorithmes des routines du thème "Constantes" de la MSLIB; G. Prat (CS SI); référence MSLIB: M-NT-0-91-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 CONSTANTES

  use mslib

  type(tm_code_retour)  :: CODE_RETOUR
  real(pm_reel)         :: PI
  real(pm_reel)         :: DEUX_PI
  real(pm_reel)         :: PI_SUR2
  real(pm_reel)         :: DEG_RAD
  real(pm_reel)         :: RAD_DEG

  call mc_math ( CODE_RETOUR,                                &
                 pi = PI, deux_pi = DEUX_PI,                &
                 pi_sur2 = PI_SUR2, deg_rad = DEG_RAD,      &
                 rad_deg = RAD_DEG )

  ! appel a la routine utilisateur d'ecriture des resultats
  call WRITE_RESULTATS ( PI, DEUX_PI, PI_SUR2,              &
                        DEG_RAD, RAD_DEG, CODE_RETOUR )

end program CONSTANTES
```

Résultats attendus:

PI	= .314 10 <sup>+1</sup>
DEUX_PI	= .628 10 <sup>+1</sup>
PI_SUR2	= .157 10 <sup>+1</sup>
DEG_RAD	= .174 10 <sup>-1</sup>
RAD_DEG	= .573 10 <sup>+2</sup>

CODE_RETOUR%valeur	= 0
CODE_RETOUR%routine	= 1016

## Routine mc\_phys

### Identification

“Définition de constantes **physiques**”.

### Rôle

Affectation de constantes physiques courantes.

### Séquence d'appel

(voir explications dans le volume 3)

call mc\_phys ( code\_retour [ , ua , vit\_lum, i\_critique\_non\_retro, i\_critique\_retro] )

### Description des arguments

(voir explications dans le volume 3)

#### • Sorties obligatoires

tm\_code\_retour      **code\_retour**

#### • Sorties facultatives

pm_reel	[ <b>ua</b> ]	une unité astronomique <i>ua</i> (km)
pm_reel	[ <b>vit_lum</b> ]	vitesse de la lumière dans le vide <i>c</i> (m.s <sup>-1</sup> )
pm_reel	[ <b>i_critique_non_retro</b> ]	inclinaison critique non rétrograde, <i>i<sub>c_non_retro</sub></i> , solution de $1 - 5 \times (\cos i)^2 = 0$ (rad)
pm_reel	[ <b>i_critique_retro</b> ]	inclinaison critique rétrograde, <i>i<sub>c_retro</sub></i> , solu- tion de $1 - 5 \times (\cos i)^2 = 0$ (rad)

### Conditions sur les arguments

Sans objet.

### Notes d'utilisation

Sans objet.

### Références documentaires

- Algorithmes des routines du thème "Constantes" de la MSLIB; G. Prat (CS SI); référence MSLIB: M-NT-0-91-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 CONSTANTES

  use mslib

  type(tm_code_retour)      :: CODE_RETOUR
  real(pm_reel)             :: UA
  real(pm_reel)             :: VIT_LUM
  real(pm_reel)             :: I_CRITIQUE_NON_RETRO
  real(pm_reel)             :: I_CRITIQUE_RETRO

  call mc_phys ( CODE_RETOUR, ua = UA, vit_lum = VIT_LUM,      &
                 i_critique_non_retro = I_CRITIQUE_NON_RETRO, &
                 i_critique_retro = I_CRITIQUE_RETRO )

  ! appel a la routine utilisateur d'ecriture des resultats
  call WRITE_RESULTATS ( UA, VIT_LUM, I_CRITIQUE_NON_RETRO,    &
                        I_CRITIQUE_RETRO, CODE_RETOUR )

end program CONSTANTES
```

### Résultats attendus:

UA	= .1496 10 <sup>+9</sup>
VIT_LUM	= .2998 10 <sup>+9</sup>
I_CRITIQUE_NON_RETRO	= .1107 10 <sup>+1</sup>
I_CRITIQUE_RETRO	= .2034 10 <sup>+1</sup>

CODE\_RETOUR%valeur = 0  
CODE\_RETOUR%routine = 1017

## Routine mc\_test

### Identification

“Définition de constantes de **tests** utilisées par les routines de la MSLIB”.

### Rôle

Cette fonction permet d’accéder aux epsilons de test sur l’excentricité et l’inclinaison.

#### Test sur l’excentricité:

Soit  $e$  l’excentricité, à calculer éventuellement avec  $e = \sqrt{e_x^2 + e_y^2}$ .

Soit  $\epsilon_{cir}$  l’epsilon de test pour l’orbite circulaire.

- si  $e < \epsilon_{cir}$  : l’orbite est circulaire
- si  $e \geq \epsilon_{cir}$  : l’orbite est non circulaire

Soit  $\epsilon_{parab}$  l’epsilon de test pour les orbites elliptique, parabolique, ou hyperbolique.

- si  $e \leq 1 - \epsilon_{parab}$  : l’orbite est elliptique
- si  $e \geq 1 + \epsilon_{parab}$  : l’orbite est hyperbolique
- si  $1 - \epsilon_{parab} < e < 1 + \epsilon_{parab}$  : l’orbite est parabolique

#### Test sur l’inclinaison:

Soit  $i$  l’inclinaison.

Soit  $\epsilon_{equa}$  l’epsilon de test pour l’orbite équatoriale.

- si  $\sin(i) < \epsilon_{equa}$  : l’orbite est équatoriale.

Soit  $\epsilon_{critique}$  l’epsilon de test pour une inclinaison proche d’une des inclinaisons critiques ( $i_{c\_non\_retro}$  ou  $i_{c\_retro}$  : pour plus de détails se reporter à la routine mc\_phys ).

- si  $i_{c\_non\_retro} - \epsilon_{critique} < i < i_{c\_non\_retro} + \epsilon_{critique}$  : l’inclinaison est proche de l’inclinaison critique non rétrograde  $i_{c\_non\_retro}$
- si  $i_{c\_retro} - \epsilon_{critique} < i < i_{c\_retro} + \epsilon_{critique}$  : l’inclinaison est proche de l’inclinaison critique rétrograde  $i_{c\_retro}$ .

### Séquence d’appel

(voir explications dans le volume 3)

call mc\_test ( code\_retour [ , eps\_cir, eps\_parab, eps\_equa, eps\_i\_critique ] )



(voir explications dans le volume 3)

- Sorties obligatoires

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

- Sorties facultatives

pm_reel	[ <b>eps_cir</b> ]	epsilon de test pour l'orbite circulaire $\epsilon_{cir}$
pm_reel	[ <b>eps_parab</b> ]	epsilon de test pour l'orbite parabolique $\epsilon_{parab}$
pm_reel	[ <b>eps_equa</b> ]	epsilon de test pour l'orbite équatoriale $\epsilon_{equa}$
pm_reel	[ <b>eps_i_critique</b> ]	epsilon de test pour l'inclinaison critique $\epsilon_{critique}$ (rad)

### Conditions sur les arguments

Sans objet.

## Notes d'utilisation

Sans objet.

## Références documentaires

- Algorithmes des routines du thème "Constantes" de la MSLIB; G. Prat (CS SI); référence MSLIB: M-NT-0-91-CIS.

## Code retour

(voir explications dans le volume 3)

pm\_OK (0) : Retour normal.

<b>Exemple en Fortran 90 portable</b>
---------------------------------------

(voir explications dans le volume 3)

```
program CONSTANTES

  use mslib

  type(tm_code_retour)          :: CODE_RETOUR
  real(pm_reel)                 :: EPS_CIR
  real(pm_reel)                 :: EPS_PARAB
  real(pm_reel)                 :: EPS_EQUA
  real(pm_reel)                 :: EPS_I_CRITIQUE

  call mc_test ( CODE_RETOUR,                                     &
                 eps_cir = EPS_CIR, eps_parab = EPS_PARAB      &
                 eps_equa = EPS_EQUA,                           &
                 eps_i_critique = EPS_I_CRITIQUE )

  ! appel a la routine utilisateur d'ecriture des resultats
  call WRITE_RESULTATS ( EPS_CIR, EPS_PARAB, EPS_EQUA ,         &
                        EPS_I_CRITIQUE, CODE_RETOUR)

end program CONSTANTES
```

Résultats attendus:

EPS_CIR	= .100 10 <sup>-9</sup>
EPS_PARAB	= .100 10 <sup>-9</sup>
EPS_EQUA	= .100 10 <sup>-9</sup>
EPS_I_CRITIQUE	= .100 10 <sup>-1</sup>

  

CODE_RETOUR%valeur	= 0
CODE_RETOUR%routine	= 1015