

cours 6

Fonctions inclues

1 Fonctions de conversion

`int(e)` partie entiere de e
`real(e)` convertit e en reel
`dble(e)` convertit e en double precision
`cmplx(e)` convertit e en complexe

Exemples

`int(3.14) = 3`
`cmplx(3.14) = 3.14 + 0*i`
`cmplx(3.14,2.) = 3.14 + 2.*i`

2 Fonctions incorporees arithmetiques

Le ou les arguments et le resultat sont de meme type (entiers, reels ou double precision).

- `abs(e)` valeur absolue
- arrondi
`nint(e)` entier le plus proche de e
`aint(e)` entier le plus proche de e, ecrit sous forme reelle.
- troncature `aint(e)` : partie entiere de e
- reste d'une division `mod(a,b)` avec b non nul
`mod(5,3)=2`
`mod(5.,3.)=2.`
- Extrema de valeurs scalaires
`min(e1,e2...)` = minimum des ei
`max(e1,e2...)` = maximum des ei
- transfert de signe
`sign(q,s)` avec s non nul =
 $|q|$ si $s > 0$
 $-|q|$ si $s < 0$

3 Fonctions relatives aux complexes

declaration

complex z

`z = cmplx(x,y)` (= $x+i*y$)

`real(z)` donne x
`aimag(z)` donne y
`conjg(z)` donne $x-i*y$
`abs(z)` donne $x^2 + y^2$

4 Fonctions mathematiques

Les arguments sont reels ou double precision parfois complexes mais jamais entiers.

- racine carree `sqrt(x)`
- exponentielle `exp(x)`
- `log(x)`, `log10(x)`
 (si $z = \rho e^{i\theta}$ complexe $\log(z) = \log(\rho) + i\theta'$ $0 < \theta' < \pi$)
- `cos`, `sin`, `tan` , `cosh`, `sinh`, `tanh`
- `acos(x)` , `asin(x)` pour $|x| < 1$, $0 < \text{acos} < \pi$, $|\text{asin}| < \pi/2$
- `atan(x)`
 $|\text{atan}| < \pi/2$
`atan2(y,x)` = angle sur le demi-cercle trigo $z = x + i * y$ positif
 ou -angle sur le demi-cercle trigo negatif

5 Fonctions systeme

generique

```

program tst

call system('pwd')

c date and time
call system('date')
call system('time')

c pwd
call system('pwd')

c pid
call system('ps -j|grep tst2')

c name
call system('id')

c env variable

```

```
cc      call system('setenv')
        call system('echo $SHELL')
```

```
        x = 1.
        do 10 i =1,10000000
            x = x + 1./x
10      continue
```

```
c time
        call system('date')
        call system('time')
```

```
        stop
        end
```

Une sortie de tst:

```
/usersgm/prof/caputo/gm3
Mon Oct  8 16:25:27 MEST 2001
0.0u 0.0s 0:00 0% 0+0k 0+0io 0pf+0w
/usersgm/prof/caputo/gm3
 24482 24482 24251 pts/20   0:00 tst2
uid=110(caputo) gid=100
/bin/csh
Mon Oct  8 16:25:29 MEST 2001
0.0u 0.0s 0:00 0% 0+0k 0+0io 0pf+0w
```