

## 9/4.4.3

# RSX COSH: cosinus hyperbolique

Les fonctions **COSH**, **SINH** et **TANH**, chères aux étudiants, ne font pas partie de la bibliothèque standard des CPC. Que nos amis mathématiciens se réjouissent. Nous allons voir comment les y inclure. Dans ce chapitre, nous allons étudier la fonction **COSH**. Nous étudierons **SINH** et **TANH** par la suite. La fonction **COSH** peut être simulée en utilisant une formule équivalente composée de fonctions qui font partie de la bibliothèque standard des CPC. En effet :

$$\text{COSH} = (e^x + e^{-x}) / 2$$

### COMMENT EXÉCUTER LE PROGRAMME

Comme les deux précédents, ce programme met en œuvre une RSX. Il est donc écrit en Assembleur. Si vous désirez l'utiliser sous sa forme Assembleur, entrez le listing suivant :

```

1          ORG  9000H
2
3          LOAD 9000H
4
5          ;-----
6          ; RSX COSINUS HYPERBOLIQUE
7          ; Format : !COSH,@VAR
8          ; Entree (VAR)=Angle
9          ; Sortie (VAR)=Cos hyper de (VAR)
10         ;-----
11        ;
12        ;
13        ;
14        ;-----
15        ;
16        ENTFLD: EQU 0BD61H          ;Ent->Flottant

```

```

17      COMFLO:    EQU  0BD8BH          ;Comp 2 flot
18      NEGFLO:    EQU  0BD8EH          ;Negation flot
19      ADDFLO:    EQU  0BD79H          ;Addition flot
20      DIVFLO:    EQU  0BD85H          ;Division flot
21      EXPFLO:    EQU  0BDA6H          ;Exponent flot
22      SAVHL:     DS    2              ;Sauvegarde de HL
23      Z1:        DS    5              ;Zone reel 1
24      Z2:        DS    5              ;Zone reel 2
25      Z3:        DS    5              ;Zone reel 3
26      LOGEXT:    EQU  0BCD1H          ;KL LOG EXT
27      BUF:       DS    4              ;Zone RAM pour LOG EXT
28 9015 1A90      PTRTAB:    DW  TABLE          ;Pointeur TABLE
29 9017 C32990      JP    COSH          ;Traitement du COSH
30 901A 434F53      TABLE:    DB  "COS"
31 901D C8          DB    "H"+80H
32 901E 00          DB    0              ;Fin de table
33      ;
34      ;-----
35      ; Definition de la RSX
36      ;-----
37      ;
38      DEFRSX:    EQU  $              ;Point d'entree
39 901F 011590      LD    BC,PRTAB          ;Ptr table definition
40 9022 21:190      LD    HL,BUF          ;Buffer pour LOG EXT
41 9025 CDD1BC      CALL LOGEXT          ;Definition de la RSX
42 9028 C9          RET
43      ;
44      ;-----
45      ; Traitement de COSH
46      ;-----
47      ;

```

```

48          COEH:    EQU $                ;Point d'entree
49 9029 DD6601      LD H,(IX+1)
50 902C DD6E00      LD L,(IX+0)          ;Adresse de la var.
51 902F 220090      LD (SAVHL),HL        ;Sauvegarde
52 9032 CD7690      CALL ZONE1          ;Memorisation
53 9035 210290      LD HL,Z1
54 9038 110C90      LD DE,Z3
55 903B CD8890      CALL FLODEHL        ;Sauvegarde
56 903E 210290      LD HL,Z1
57 9041 CDA6BD      CALL EXPFLO          ;E^X
58 9044 210C90      LD HL,Z3
59 9047 CD8EBD      CALL NEGFLO          ;-X
60 904A 210C90      LD HL,Z3
61 904D CDA6BD      CALL EXPFLO          ;E^-X
62 9050 210290      LD HL,Z1
63 9053 110C90      LD DE,Z3
64 9056 CD79BD      CALL ADDFLO          ;E^X+E^-X
65 9059 210200      LD HL,2
66 905C 110C90      LD DE,Z3
67 905F CD61BD      CALL ENTFLO          ;2 en flottant
68 9062 210290      LD HL,Z1
69 9065 110C90      LD DE,Z3
70 9068 CD85BD      CALL DIVFLO          ;(E^X+E^-X)/2
71          ;
72          FINPRE: EQU $
73 906B 210290      LD HL,Z1
74 906E ED5B0090     LD DE,(SAVHL)
75 9072 CD8890      CALL FLODEHL        ;Resultat
76          ;
77          ;

```

```
78          FIN:      EQU  $                ;Fin du programme
79 9075 C9          RET
80          ;
81          ;-----
82          ; Zone des sous-programmes
83          ;-----
84          ;
85          ;-----
86          ; Transfert des BC octets pointes
87          ; par HL dans le buffer Z1
88          ;-----
89          ;
90          ZONE1:    EQU  $
91 9076 110290      LD   DE,Z1
92 9079 010500      LD   BC,5
93 907C EDB0        LDIR
94 907E C9          RET
95          ;
96          ;-----
97          ; Transfert des BC octets pointes
98          ; par HL dans le buffer Z3
99          ;-----
100         ;
101         ZONE3:    EQU  $
102 907F 110C90      LD   DE,Z3
103 9082 010500      LD   BC,5
104 9085 EDB0        LDIR
105 9087 C9          RET
106         ;
```

```

107      ;-----
108      ; Transfert flottant de (HL)
109      ; dans (DE)
110      ;-----
111      ;
112      FLODEHL: EQU $
113 9088 010500      LD BC,5
114 908B EDB0      LDIR
115 908D C9      RET
116      END

```

ADDFLO	BD79 BUF	9011 COMFLO	BD8B COSH	9029
DIVFLO	BD85 DEFRSX	901F ENTFLO	BD61 EXPFLO	BDA6
FINPRE	906B FIN	9075 FLODEHL	9088 LOGEXT	BCD1
NEGFLO	BD8E PTRTAB	9015 SAVHL	9000 TABLE	901A
Z1	9002 Z2	9007 Z3	900C ZONE1	9076
ZONE3	907F			

Installez la RSX en tapant sous Basic :

```
CALL &901F
```

La fonction !COSH fait maintenant partie du Basic standard. Voici comment l'utiliser :

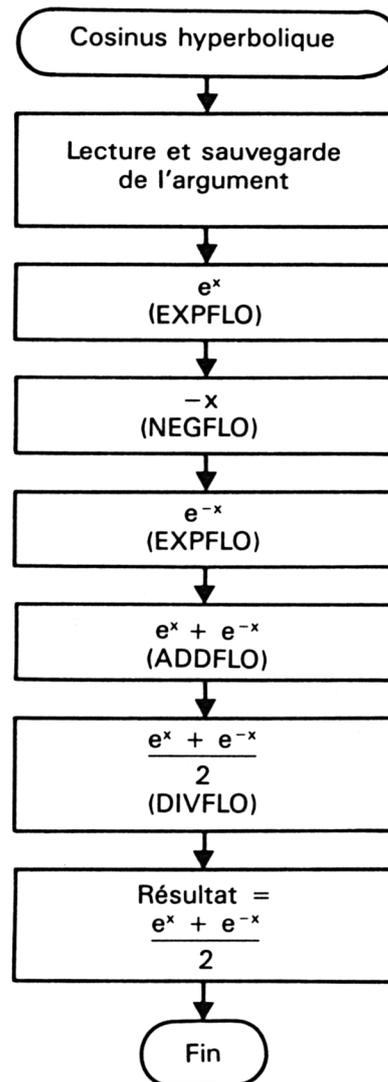
```

10 a=.5      'Argument à passer à la fonction COSH
20 !COSH,@a  'Appel de la fonction COSH
30 PRINT a   'Affichage du résultat

```

**LE PROGRAMME EN DÉTAIL**

La logique du programme obéit à l'ordinogramme suivant :



La structure du programme est identique à celle des deux précédents. Les vecteurs mathématiques utilisés sont cependant quelque peu différents :

ENTFLO:	EQU	0BD61H	;Conversion Entier → Flottant
COMFLO:	EQU	0BD8BH	;Comparaison de deux flottants
NEGFLO:	EQU	0BD8EH	;Négation d'un flottant
ADDFLO:	EQU	0BD79H	;Addition de deux flottants
DIVFLO:	EQU	0BD85H	;Division de deux flottants
EXPFLO:	EQU	0BDA6H	;Exponentiation d'un flottant

Ces adresses sont correctes pour les CPC 664. Si vous possédez un CPC 464 ou un CPC 6128, vous devez les convertir comme suit :

Point d'entrée	CPC 464	CPC 664	CPC 6128
ENTFLO	0BD40H	0BD61H	0BD64H
COMFLO	0BD6AH	0BD8BH	0BD8EH
NEGFLO	0BD6DH	0BD8EH	0BD91H
ADDFLO	0BD58H	0BD79H	0BD7CH
DIVFLO	0BD64H	0BD85H	0BD88H
EXPFLO	0BD85H	0BDA6H	0BDA9H

*Remarque :*

Pour faciliter l'utilisation de cette RSX, les chargeurs Basic sont donnés dans les trois versions (464, 664 et 6128).

La définition de la RSX doit être effectuée avant sa première utilisation. Le court programme situé à l'étiquette **DEFRSX** est chargé de cette tâche.

Ce programme est identique à celui des fonctions **ACOS** et **ASIN**. Reportez-vous à ces fonctions pour tout renseignement utile.

Lorsque l'utilisateur désire utiliser la fonction **COSH**, il place l'argument de la fonction dans une variable flottante et passe l'adresse de cette variable à la RSX **COSH**. Le résultat est retourné dans cette même variable.

Par exemple, pour connaître le cosinus hyperbolique de 0.5, tapez :

**b = 0.5: !COSH,@b:? b**

L'ordinateur affichera 1.12762597. En effet,  $\cosh(0.5) = 1.12762597$ .

Lorsque l'interpréteur rencontre l'instruction **COSH**, il recherche **COSH** dans la table RSX et exécute la routine de traitement qui lui est associée.

La première action effectuée dans cette routine consiste à déclarer que les calculs se feront en degrés.

L'adresse de la variable passée est ensuite récupérée à l'aide du registre IX et stockée dans le registre HL :

```
LD    H,(IX + 1)
LD    L,(IX + 0)    ;Adresse de la variable
```

La quantité  $e^X$  est calculée à l'aide du vecteur **EXPFLO** :

```
LD    HL,Z1
CALL  EXPFLO    ;e^X
```

La quantité  $e^{-X}$  est calculée à l'aide des vecteurs **NEGFLO** et **EXPFLO** :

```
LD    HL,Z3
CALL  NEGFLO    ; -X
LD    HL,Z3
CALL  EXPFLO    ;e^(-X)
```

Les quantités  $e^X$  et  $e^{-X}$  sont ensuite additionnées à l'aide du vecteur ADDFLO :

```
LD      HL,Z1
LD      DE,Z3
CALL    ADDFLO      ;eX + e(-X)
```

Cette valeur est enfin divisée par deux :

```
LD      HL,2
LD      DE,Z3
CALL    ENTFLO      ;2 en flottant
LD      HL,Z1
LD      DE,Z3
CALL    DIVFLO      ;(eX + e(-X))/2
```

*Remarque :*

Pour pouvoir effectuer la division, le nombre réel 2 doit préalablement être créé à l'aide du vecteur ENTFLO.

Le résultat de la fonction est stocké dans la variable passée au programme à l'aide de la routine FLODEHL qui permet de recopier le réel pointé par HL dans la zone pointée par DE :

```
LD      HL,Z1
LD      DE,(SAVHL)
CALL    FLODEHL     ;Résultat
```

Si vous préférez utiliser un chargeur Basic, voici le listing et les données de checksum correspondantes :

```
1000 '-----
1010 ' Chargeur de la RSX COSINUS HYPERBOLIQUE
1020 '-----
1030 ' VERSION CPC 464
1040 '-----
1050 '
1060 FOR I=&9000 TO &9080
1070   READ A$
1080   A=VAL("&" + A$)
1090   POKE I,A
1100 NEXT I
1110 '
1120 '-----
1130 ' Codes op de la RSX COSINUS HYPERBOLIQUE
1140 '-----
1150 '
1160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1170 DATA 0,0,0,0,0,1A,90,C3,29,90,43,4F,53,C8,0,1
1180 DATA 15,90,21,11,90,CD,D1,BC,C9,DD,66,1,DD,6E,0,22
1190 DATA 0,90,CD,76,90,21,2,90,11,C,90,CD,88,90,21,2
1200 DATA 90,CD,85,DD,21,C,90,CD,6D,BD,21,C,90,CD,85,BD
1210 DATA 21,2,90,11,C,90,CD,58,BD,21,2,0,11,C,90,CD
1220 DATA 40,BD,21,2,90,11,C,90,CD,64,BD,21,2,90,ED,5B
1230 DATA 0,90,CD,88,90,C9,11,2,90,1,5,0,ED,B0,C9,11
1240 DATA E.90.1.5.0.ED.B0.C9.1.5.0.ED.B0.C9.0.0
```

0 D7 42 D0 27 E3 4C 64 79

\* \* \*

```

1000 '-----
1010 ' Chargeur de la RSX COSINUS HYPERBOLIQUE
1020 '-----
1030 ' VERSION CPC 664
1040 '-----
1050 '
1060 FOR I=&9000 TO &9080
1070   READ A$
1080   A=VAL("&"+A$)
1090   POKE I,A
1100 NEXT I
1110 '
1120 '-----
1130 ' Codes op de la RSX COSINUS HYPERBOLIQUE
1140 '-----
1150 '
1160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1170 DATA 0,0,0,0,0,1A,90,C3,29,90,43,4F,53,C8,0,1
1180 DATA 15,90,21,11,90,CD,D1,BC,C9,DD,66,1,DD,6E,0,22
1190 DATA 0,90,CD,76,90,21,2,90,11,C,90,CD,88,90,21,2
1200 DATA 90,CD,A6,BD,21,C,90,CD,8E,BD,21,C,90,CD,A6,BD
1210 DATA 21,2,90,11,C,90,CD,79,BD,21,2,0,11,C,90,CD
1220 DATA 61,BD,21,2,90,11,C,90,CD,85,BD,21,2,90,ED,5B
1230 DATA 0,90,CD,88,90,C9,11,2,90,1,5,0,ED,B0,C9,11
1240 DATA C,90,1,5,0,ED,B0,C9,1,5,0,ED,B0,C9,0,0

```

0 D7 42 D0 BA 5 BE 64 79

```
1000 '-----
1010 ' Chargeur de la RSX COSINUS HYPERBOLIQUE
1020 '-----
1030 ' VERSION CPC 6128
1040 '-----
1050 '
1060 FOR I=&9000 TO &908D
1070   READ A$
1080   A=VAL("&"+A$)
1090   POKE I,A
1100 NEXT I
1110 '
1120 '-----
1130 ' Codes op de la RSX COSINUS HYPERBOLIQUE
1140 '-----
1150 '
1160 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1170 DATA 0,0,0,0,0,1A,90,C3,29,90,43,4F,53,C8,0,1
1180 DATA 15,90,21,11,90,CD,D1,BC,C9,DD,66,1,DD,6E,0,22
1190 DATA 0,90,CD,76,90,21,2,90,11,C,90,CD,88,90,21,2
1200 DATA 90,CD,A9,BD,21,C,90,CD,91,BD,21,C,90,CD,A9,BD
1210 DATA 21,2,90,11,C,90,CD,7C,BD,21,2,0,11,C,90,CD
1220 DATA 64,BD,21,2,90,11,C,90,CD,88,BD,21,2,90,ED,5B
1230 DATA 0,90,CD,88,90,C9,11,2,90,1,5,0,ED,B0,C9,11
1240 DATA C,90,1,5,0,ED,B0,C9,1,5,0,ED,B0,C9,0,0
```

0 D7 42 D0 93 8 94 64 79