

tsolve

Solveur d'équations trigonométriques

Calculatrices : 89 92 92+ V200

Niveaux : 1ere 1ere S Tale Tale S Sup

Descriptif : Fonction - Permet de résoudre proprement les équations trigonométriques en prenant bien en compte les conditions supplémentaires

Mots-clefs : équations, trigonométrie

Auteur : Hubert Bayet

Date de dernière révision : Septembre 2002

Présentation :

Il s'agit d'un ensemble dont le programme principal est *ts* ; il permet de résoudre des équations trigonométriques. Bien que votre calculatrice le fasse par l'utilisation de la fonction *solve*, les résultats qu'elle retourne dépendent souvent d'un paramètre interne @n (désignant un entier relatif quelconque). De plus, elle ne prend pas toujours en compte les conditions supplémentaires qu'on lui indique, en particulier les conditions portant sur les intervalles de solutions. Grâce à *tsolve*, ceci est amélioré.

Mode d'emploi :

Il vous suffit de taper : *ts(equation,variable,condition)*.

Par exemple : *ts(sin(x)=1/2, x, x>0 and x<π/2)* affichera $\pi/6$

Note : il peut toutefois y avoir quelques dysfonctionnements, auquel cas il est conseillé d'indiquer les conditions sous la forme $x>...$ and $x<...$ (ou $x>=...$ and $x<=...$) lorsque l'intervalle est éloigné de l'origine.

Sources :

```
NI ist (cond,nva)
Func
Local nn1,nnx,nns,nnp,nnq,con
cond|nva=nnx→con
nSolve(sin(nnx*π)=0,nnx)|con→nns
If getType(nns)="NUM" Then
exact(int(nns))→nns
{}→nn1
For nnq,-1,1,2
nns+(nnq+1)/2→nnp
Loop
If (con|nnx=nnp)=false: Exit
```

```

augment(nn1,{nnp})→nn1
nnp+nnq→nnp
EndLoop
EndFor
Else
{}→nn1
EndIf
Return nn1

EndFunc

Remplan (tts,ttchv)
Func
Local ttm,ttsc,ttsc1,ttsc2,ttq,ttr,ttw,ttsn
string(tts)→ttsc
Loop
dim(ttsc)→ttm
inString(ttsc,"@")→ttr
If ttr=∅:Exit
ttr+1→ttq
Loop
ttq+1→ttq
If ttq=ttm: Exit
ord(mid(ttsc,ttq,1))→ttw
If ttw>57 or ttw<48: Exit
EndLoop
left(ttsc,ttr-1)→ttsc1
right(ttsc,ttm-ttq+1)→ttsc2
ttsc1&ttchv&ttsc2→ttsc
EndLoop
expr(ttsc)→ttsn
Return ttsn
EndFunc

Ts (tsopb,tsov,tsot)
Prgm
Local
tsoc,tsod,tsoz,tsol,tsoon,tsos,tsn,tsocc,tsodd,tsos2,tsoqq,tss0,tsopp,t
sp,tsq,tsdeb,tsorm,tsond,tsoll,tsww,tsow,tsor,tsoe,tseq2,tsol2

tsot|tsov=tsoz→tsoon

left(tsopb|tsov=tsoz)-right(tsopb|tsov=tsoz)→tsoc
when(tsoon,tsoc,1)→tsod
solve(tsod=∅,tsoz)→tsos
If inString(string(tsos),"@">>∅ Then
Lbl bbon
remplan(tsos,"tsn")→tsos
string(tsos)→tss0: {"Exact"}→tseq
Loop
inString(tss0,"or")→tsor
dim(tss0)→tsow
If tsor=∅ Then: tsso→tsdeb: tsow→tsp
Else: tsor→tsp: left(tss0,tsp-1)→tsdeb
EndIf
inString(tsdeb,"and")→tsq

```

```

expr(left(tsdeb,tsq-1))→tsorm
expr(right(tsdeb,tsp-tsq-3))→tsond
nlist(tsnd,tsn)→tsoll
dim(tsoll)→tsoqq
If tsoqq≠∅ Then
For tsoe,1,tsoqq
tsorm|tsn=tsoll[tsoe]→tsopp
solve(tsopp,tsoz)→tsopp
exp>List(tsopp,tsoz)→tsww
augment(tseq,tsww)→tseq
EndFor
EndIf
If tsor=∅: Exit
right(tss0,tsow-tsp-1)→tss0
EndLoop
Else
tsos→tsos2: tCollect(tsoc)→tsocc
when(tsoon,tsocc,1)→tsodd
solve(tsodd=∅,tsoz)→tsos
If inString(string(tsos), "e")>∅
Goto bbon
tsos or tsos2→tsos
tsod*tsodd→tsod
exp>List(tsos,tsoz)→tsol
tsol/(π)→tsol2
exact(tsol2,1E-∅∅7)→tsol2
π*tsol2→tsol2
dim(tsol)→tsp
{}→tseq
If tsp>∅ Then
For tsq,1,tsp
tsol[tsq]→tsoqq
If abs(tsod|tsoz=tsoqq)<1E-∅13
augment(tseq,{tsoqq})→tseq
EndFor
tseq/(π)→tseq2
exact(tseq2,1E-∅∅7)→tseq2
tseq2*π→tseq2
augment({"soit environ"},tseq2)→tseq2
augment({"approx"},tseq)→tseq
augment(tseq,tseq2)→tseq
EndIf
EndIf
Disp tseq

EndPrgm

```