

Stage découverte de l'univers Nspire

Discriminant et programmation

Activité 3

Mots-clés : discriminant, équation, programmation.

Fichier associé : Discriminant_eleve.tns

1. Objectifs

Utiliser la TI-Nspire pour le calcul du discriminant et des solutions éventuelles d'une équation du second degré. Découvrir l'environnement de calcul formel et de programmation.

2. Conduite de l'activité

1) Le « calcul manuel » du discriminant

Ouvrir une page **Calculs** pour calculer les racines de l'équation $x^2 + 4x + 3 = 0$. Pour cela, mettre en mémoire les réels 1, 4 et 3, coefficients du trinôme du second degré dans les variables **a**, **b**, **c**.

Calculer le discriminant et le mettre en mémoire dans la variable **d**. Comme **d** est ici positif, faire afficher les deux solutions de l'équation.

2) Création d'un programme

Ouvrir une nouvelle page **Calculs** pour concevoir un programme automatisant les différentes étapes de résolution d'une équation du second degré.

a) Créer un nouveau programme nommé **discr**. Ce programme a 3 variables d'entrée (**a**, **b**, **c**), 1 variable intermédiaire (**d**) et 2 variables de sortie (**x1**, **x2**) déclarées à l'aide de l'instruction **Local**.

b) L'utilisateur devra, en premier lieu, communiquer au programme les valeurs des coefficients du trinôme. Pour chacun de ces 3 coefficients, utiliser l'instruction **Request** (syntaxe : **Request** " texte ", nom_de_variable). Lors de l'exécution, le programme affichera une boîte de dialogue afin que l'utilisateur entre un nombre au clavier.

c) Calculer la valeur du discriminant et stocker le résultat dans la variable **d**. Afficher le résultat au moyen de l'instruction **Disp** (syntaxe : **Disp** " texte ", nom_de_variable).

d) Mettre en œuvre un test sur le signe de **d**. Si **d** < 0, afficher qu'il n'y a pas de solution. Si **d** > 0, afficher les valeurs des deux solutions (éventuellement confondues lorsque **d** = 0).
(Syntaxe du test est : **if** « condition » **Then** « commande à exécuter ».)

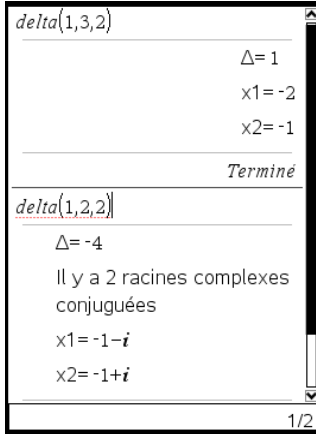
e) Tester le programme sur des cas simples.

3) Création d'un programme plus élaboré

Ouvrir une nouvelle page **Calculs** et créer un nouveau programme nommé **delta**. Passer les variables d'entrées **a**, **b**, **c** en tant que paramètres. Le programme a 1 variable intermédiaire (**d**) et 2 variables de sortie (**x1**, **x2**) déclarées à l'aide de l'instruction **Local**.

• Programme amélioré

Noter que les valeurs des coefficients a , b et c ne seront plus entrées via une boîte de dialogue mais directement lors de l'appel du programme.



Le programme propose la version terminale S avec l'emploi des nombres complexes. Il utilise aussi l'instruction Si ou si.... et sinon....

```

Define delta(a,b,c)=
Prgm
Local d,x1,x2
d:=b2-4a*c
Disp "Δ=",d
If d<0 Then
  Disp "Il y a 2 racines complexes conjuguées"
  Disp "x1=",(-b-√(-d))/(2a)
  Disp "x2=",(-b+√(-d))/(2a)
Elseif d=0 Then
  Disp "Une racine double"
  Disp "x0=",(-b)/(2a)
Else
  Disp "x1=",(-b-√(d))/(2a)
  Disp "x2=",(-b+√(d))/(2a)
EndIf
EndPrgm
  
```