MAGNIER
Louis

NF04 : Projet

Sujet : Volume d'un Prisme droit

Problème :

La fonction de cet algorithme est de calculer le volume d'un prisme droit pour lequel l'utilisateur choisirait la forme de la base (via la Superformule de Gielis) ainsi que la hauteur du prisme.

Spécifications :

- Données :

L'utilisateur pourra choisir chaque paramètre de la Superformule (a; b; m; n_1; n_2; n_3) ainsi que la hauteur du prisme. Toutes ces valeurs seront saisies au clavier dans des boîtes de dialogue et sont des réels positifs (différent de 0 pour a; b et n_1)

- Résultats et Affichages :

Le résultat attendu est le volume du prisme droit. Celui-ci est un réel positif et sera affiché dans une boîte d'affichage.

Pour ce qui est des autres affichages : On affichera dans chaque boîte de dialogue un message pour guider l'utilisateur.

Analyse :

Après avoir lu le sujet, il est possible de dégager 3 grandes étapes : Tout d'abbord déterminer la forme de la base puis, calculer l'aire de cette dernière et enfin calculer le Volume et l'afficher.
Dans le but de structurer cette analyse, nous la diviserons en 3 parties (correspondant à nos 3 grandes étapes).

I/ Déterminer la forme de la base :

Ici, l'utilisateur sera libre du choix de la base. Il devra donc saisir chacun des paramètres de la Superformule de Gielis ainsi que la hauteur dans de boîtes de dialogues adaptés. Toutes les valeurs saisies seront stockées dans des variables portant les noms des paramètres de la formule (a ; b ; m ; n1 ; n2 ; n3 et hauteur)

II / Calculer l'aire de la base :

Pour cette partie, il est suggéré, dans le sujet, d'utiliser une méthode d'intégration numérique. Il existe plusieurs types d'intégration numérique : la méthode des rectangles, la méthode des trapèzes et la méthode de Simpson. Nous utiliserons cette dernière car le fait d'approcher notre fonction à l'aide de polynôme de degrés 2 rend l'approximation bien plus précise. Dans le cas général, la méthode de Simpson indique :

$$\int_a^b f(x)\,dx = \frac{h}{3}\left[(f(a)+f(b)) + 4\sum_{i=1}^{n-1} f(a+ih) + 2\sum_{j=2}^{n-2} f(a+jh)\right]$$

avec $h = \frac{b-a}{n}$ $i = 2k+1$ (nombre impair) et $j = 2k$ (nombre pair)

Dans notre cas, la surface est donnée par $\frac{1}{2}\int_0^{2\pi} r^2(t)\,dt$. Soit :

$$\frac{1}{2}\int_0^{2\pi} r^2(t)\,dt = \frac{1}{2} \times \frac{\frac{2\pi-0}{n}}{3}\left[(r^2(0)+r^2(2\pi)) + 4\sum_{i=1}^{n-1} r^2\left(i \times \frac{2\pi-0}{n}\right) + 2\sum_{i=2}^{n-2} r^2\left(j \times \frac{2\pi-0}{n}\right)\right]$$

Étant donné que l'on désire une précision la plus grande possible, on fixera la variable n (qui correspond au nombre de division de l'ensemble) à 50. On stockera différentes données : a et b (respectivement inté 1 et inté 2) et h (largeur). Le calcul sera divisé en quatres parties :

- calcul de $r^2(0)+r^2(2\pi)$ stocké dans la variable <u>somme 1</u>.
- calcul de $4\sum_{i=1}^{n-1} r^2\left(i\times\frac{2\pi-0}{n}\right)$ stocké dans la variable <u>somme 2</u>. Pour effectuer ce calcul, on fera la somme de chaque terme pour i allant de 1 à n-1 de 2 en 2. Puis on multipliera cette somme par 4.
- calcul de $2\sum_{j=2}^{n-2} r^2\left(j\times\frac{2\pi-0}{n}\right)$ stocké dans la variable <u>somme 3</u>. Pour j allant de 2 à n-2 de 2 en 2. Puis on multipliera cette somme par 2.
- Addition de somme 1, somme 2 et somme 3 dans la variable <u>Aire</u>.

On multipliera successivement Aire par largeur/3 et 1/2.

Seulement, à de multiples reprises nous devrons calculer $r^2(t)$. Pour cela, nous ferons appel à un sous-algorithme (fonction) dont les paramètres d'entrés seront <u>coeff</u> (cette variable représente t de $r^2(t)$) et les paramètres de la Superformule. Ce sous-algorithme aura pour paramètre de sortie <u>passage</u>. On aura également plusieurs étapes dans ce sous-algorithme:

- Calcul de $\left|\frac{\cos\left(\frac{mt}{4}\right)}{a}\right|^{n_2}$ On calculera d'abbord cos à l'aide d'un sous-algorithme (Calcul Cosinus) à qui on transmettra m et coeff et qui nous renverra <u>cosinus</u>. Après cela, on divisera cosinus par a puis on determinera, à l'aide d'une condition, sa valeur absolue. Pour finir, on élevera cosinus à la puissance n2 à l'aide d'une boucle "Pour"
- Calcul de $\left|\frac{\sin\left(\frac{mt}{4}\right)}{b}\right|^{n_3}$. Sur le même principe, on fera appel à un sous algorithme pour calculer sin (calcul sinus). Ce sous-algorithme nous renverra la variable <u>sinus</u>. On divisera ensuite sinus par b, puis on determinera sa valeur absolue et enfin on l'élevera à la puissance n3 à l'aide d'une boucle "pour"

- Finallement : calcul de $\frac{1}{\sqrt[n_1]{\left|\frac{\cos(\frac{mt}{4})}{a}\right|^{n_2}+\left|\frac{\sin(\frac{mt}{4})}{b}\right|^{n_3}}}$. Pour cela, on commencera par additionner cosinus et sinus dans la variable passage. Après cela, on fera appel à un dernier Sous-algorithme (calcul racine nième) qui calculera la racine n-ième. Ses paramètres d'entrées sont passage et n1 et le paramètre de sortie est passage. Après cela, il nous restera à multiplier passage par elle même et à inverser cette même variable.

III / Calculer le volume et l'afficher :

Pour calculer le volume, on multipliera simplement Aire par hauteur. Finallement, on affichera le volume dans une boîte de dialogue.

<u>Analyse</u> : Sous algorithme Calcul cosinus.

Ce sous-algorithme permet de calculer cosinus $(\frac{mt}{4})$. Dans ce sous-algorithme, on utilisera la formule du devellopement limité de cosinus que l'on calculera au rang $n=20$ (pour obtenir une bonne precision). On initialisera chaque variable pour correspondre au rang $n=1$ et on fera des sommes répétées dans une boucle "Pour". Quelques variables supplémentaires seront nécessaire : <u>n</u>, <u>signe</u>, <u>x</u>, <u>puisx</u> et <u>factorielle</u>.

<u>Analyse</u> Sous algorithme Calcul sinus.

Ce sous algorithme permet de calculer sinus $(\frac{mt}{4})$. Dans ce sous algorithme, on utilisera la formule du devellopement limité de sinus que l'on calculera au rang $n=20$. On initialisera chaque variable pour correspondre au rang $n=1$ et on effectuera

MAGNIER
Louis

des sommes répétées dans une boucle "Pour". Quelques variables supplémentaires seront nécessaire : n ; signe ; x ; puisx et factorielle

Analyse : Sous-algorithme Calcul racine n-ième

Ce Sous-algorithme permet de calculer une racine n-ième. Il repose sur un principe assez simple : On sait que $\sqrt[n]{A} = x \iff x^n = A$. Si on considère cela avec nos paramètres, on a : $x^{n1} =$ passage. On va donc chercher x grâce à plusieurs boucles "Tant que" qui représenteront chacune un niveau de precision. Dans chacune de ces boucle, on calculera x^{n1} dans la variable puisx à l'aide de la variable x (qui sera incrementer à chaque tour de boucle.) Le calcul de x^{n1} se fera dans une boucle "Pour" imbrique dans la boucle "Tant que" susdite. On devra également détailler quelques cas particuliers notamment quand n1 ou passage valent 1 et quand passage vaut 0.

Algorithme Principal

Variables
a; b; m; n1; n2; n3; Aire; largeur; inté 1; inté 2; passage; somme 1;
somme2; somme 3; coeff; volume; hauteur: réel
n; i; j: entier.

Instructions:
Ecrire (Fenêtre, "Le programme a pour but de calculer le volume d'un prisme droit. Veuillez suivre les informations"!)
Ecrire (Fenêtre, "entrez a: réel positif différent de 0"!)
Lire (Fenêtre! a)
Ecrire (Fenêtre, "entrez b: réel positif différent de 0"!)
Lire (Fenêtre! b)
Ecrire (Fenêtre, "entrez m: réel positif"!)
Lire (Fenêtre! m)
Ecrire (Fenêtre, "entrez n1: réel positif différent de 0"!)
Lire (Fenêtre! n1)
Ecrire (Fenêtre, "entrez n2: réel positif"!)
Lire (Fenêtre! n2)
Ecrire (Fenêtre, "entrez n3: réel positif"!)
Lire (Fenêtre! n3)
Ecrire (Fenêtre, "entrez la valeur de la hauteur"!)
Lire (Fenêtre! hauteur)
inté 1 ← 0
inté 2 ← 2π
n ← 50
largeur ← (inté 2 - inté 1) / n
somme 2 ← 0
somme 3 ← 0
fonction (inté 1; a; b; m; n1; n2; n3! passage)
somme 1 ← passage
fonction (inté 2; a; b; m; n1; n2; n3! passage)
somme 1 ← somme 1 + passage
Pour i allant de 1 à n-1 par pas de 2 faire:
coeff ← inté 1 + (i * largeur)
fonction (coeff; a; b; m; n1; n2; n3! passage)
somme2 ← somme2 + passage
Fin pour
Pour j allant de 2 à n-2 par pas de 2 faire
coeff ← inté 1 + (j * largeur)
fonction (coeff; a; b; m; n1; n2; n3! passage)
somme 3 ← somme 3 + passage.
Fin pour
somme 2 ← somme 2 * 4
somme 3 ← somme 3 * 2
Aire ← (somme 1 + somme 2 + somme 3) * (largeur / 3) * (1/2)
Volume ← Aire × hauteur.
Ecrire (Fenêtre, "Le volume du prisme est de", Volume)
Fin principal.

Sous - Algorithme fonction

PF : coeff ; a ; b ; n ; n1 ; n2 ; n3 : réel
PS : passage : réel

Variables :
cosinus, sinus : réel
i : entier

Instructions :

```
    calcul cosinus (n ; coeff ! cosinus)
    cosinus ← cosinus / a
    Si (cosinus < 0) alors :
        cosinus ← - cosinus
    Finsi
    Si (n2 = 0) alors :
        cosinus ← 1
    Sinon si (n2 = 1) alors :
        cosinus ← cosinus
    Sinon
        Pour i allant de 2 à n2 par pas de 1 faire :
            cosinus ← cosinus * cosinus
        Fin pour
    Finsi
    calcul sinus (n ; coeff ! sinus)
    sinus ← sinus / b
    Si (sinus < 0) alors :
        sinus ← - sinus
    Finsi
    Si (n3 = 0) alors :
        sinus ← 1
    Sinon si (n3 = 1) alors :
        sinus ← sinus
    Sinon
        Pour i allant de 2 à n3 par pas de 1 faire :
            sinus ← sinus * sinus
        Fin pour
    Finsi
    passage ← cosinus + sinus
    calcul racine n-ième (passage ; n1 ! passage)
    passage ← passage * passage
    passage ← 1 / passage
Fin fonction.
```

Sous algorithme : calcul cosinus:

PE: n, coeff : réel
PS: cosinus : réel

Variables:
x, puisx : réel
n, signe, factorielle : entier

Instructions:

```
cosinus ← 1
signe ← -1
factorielle ← 2
x ← (n * coeff)/4
puisx ← x * x
Pour n allant de 1 à 20 par pas de 1 faire:
    cosinus ← cosinus + ((signe * puisx)/factorielle)
    signe ← -signe
    factorielle ← factorielle * (2* n+1) * (2* n+2)
    puisx ← puisx * x * x.
Fin pour
```

Fin calcul cosinus.

Sous Algorithme : calcul sinus:

PE: n, coeff : réel
PS: sinus : réel

Variables:
x, puisx : réel
n, signe, factorielle : entier

Instructions:

```
x ← (n * coeff)/4
sinus ← x
signe ← -1
factorielle ← 6
puisx ← x * x * x
Pour n allant de 1 à 20 par pas de 1 faire:
    sinus ← sinus + ((signe * puisx)/factorielle)
    signe ← -signe
    factorielle ← factorielle * (2* n+2) * (2* n+3)
    puisx ← puisx * x * x
Fin pour
```

Fin calcul sinus.

MAGNIER
Louis

Sous - Algorithme : Calcul racine n-ième

PE : passage ; n1 : réel
PS : passage : réel

Variables :
x ; puisx : réel
i : entier

Instructions :

```
puisx ← 1
x ← 0
Si (passage = 0) alors
    passage ← 0
Sinon si (n1 = 1) alors :
    passage ← passage
Sinon si (passage = 1) alors
    passage ← passage
Sinon :
    Tant que (puisx < passage) faire :
        puisx ← 1
        x ← x+1
        Pour i allant de 1 à n1 par pas de 1 faire :
            puisx ← puisx * x
        Fin pour
    Fin tant que.
    Tant que (puisx > passage) faire :
        puisx ← 1
        x ← x - 0,1
        Pour i allant de 1 à n1 par pas de 1 faire :
            puisx ← puisx * x
        Fin pour
    Fin tant que
    Tant que (puisx < passage) faire :
        puisx ← 1
        x ← x + 0,01
        Pour i allant de 1 à n1 par pas de 1 faire :
            puisx ← puisx * x
        Fin pour
    Fin Tant que
    Tant que (puisx > passage) faire :
        puisx ← 1
        x ← x - 0,001
        Pour i allant de 1 à n1 par pas de 1 faire :
            puisx ← puisx * x
        Fin pour
    Fin tant que
    passage ← x
Fin si
Fin Calcul racine n-ième.
```