ÉCOLE POLYTECHNIQUE OPTION MP

CONCOURS D'ADMISSION 1997

COMPOSITION D'INFORMATIQUE

(Durée: 3 heures)

L'utilisation des calculatrices n'est pas autorisée pour cette épreuve.

***

Avertissement. On attachera une grande importance à la clarté, à la précision et à la concision de la rédaction.

Introduction. Le but de ce problème est l'étude d'une mise en œuvre efficace de la simulation du problème des N corps dans un univers plan. Le problème des N corps consiste à calculer les trajectoires de N corps ou particules qui interagissent entre eux sous l'effet de la gravité. Ce problème n'a pas de solution analytique dans le cas général, on a donc recours à une simulation numérique.

Le principe de ce type de simulation est, à un instant t donné, de calculer les forces qui s'exercent sur chaque corps, puis d'estimer les vitesses et les positions de tous les corps à l'instant $t+dt$. Pour calculer les forces, on doit, pour chacun des N corps, calculer $N-1$ interactions gravitationnelles. Le coût du calcul exact des forces est donc de l'ordre de N^2. Ce comportement quadratique limite sérieusement l'intérêt de la simulation.

Une approximation physique simple va permettre une amélioration de l'efficacité de la simulation. Étant donnés un corps c et un ensemble C de corps, l'attraction exercée sur c par les corps de C peut, pourvu que C soit relativement éloigné de c, être approximée par l'attraction exercée sur c par le centre de masse des corps de C.

Notations. Dans tous le problème, $\mathcal{U}$ est l'univers, il contient N corps. Étant donné un corps c, on note $\vec{p}_c$ le vecteur position de c. Étant donné un vecteur $\vec{u}$, on note $u.x$ et $u.y$ les composantes de $\vec{u}$ et $||\vec{u}||$ sa norme euclidienne $\sqrt{u.x^2 + u.y^2}$.

Première partie : Quadtrees

Dans la pratique, les ensembles C de corps sont des cellules carrées et ces ensembles sont organisés hiérarchiquement. L'univers est d'abord assimilé à une grosse cellule carrée de côté $D_{\mathcal{U}}$. Chaque cellule qui contient deux corps ou plus est ensuite subdivisée en quatre sous-cellules carrées de même taille, la subdivision s'arrêtant lorsqu'une cellule contient un ou zéro corps.

La figure 1 ci-contre montre un exemple de division d'un univers en cellules.

Une représentation structurée de la répartition des corps en cellules et sous-cellules est un arbre dont les nœuds internes ont quatre fils et dont les feuilles contiennent zéro ou un corps. La division en sous-cellules n'est poussée qu'autant que nécessaire, c'est-à-dire que chaque nœud qui a des fils représente une cellule qui contient deux corps ou plus. Un tel arbre est un *quadtree (adaptatif)*.

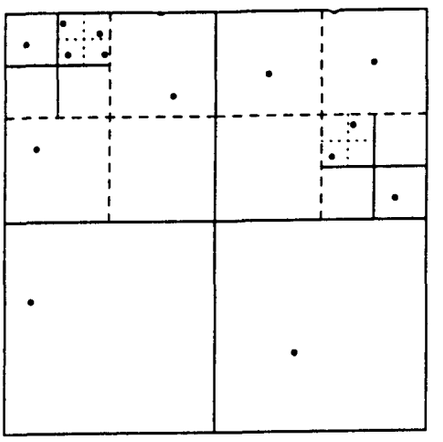

Figure 1

La figure 2 ci-dessous montre la représentation arborescente de l'univers déjà décrit sous forme de schéma sur la figure 1.

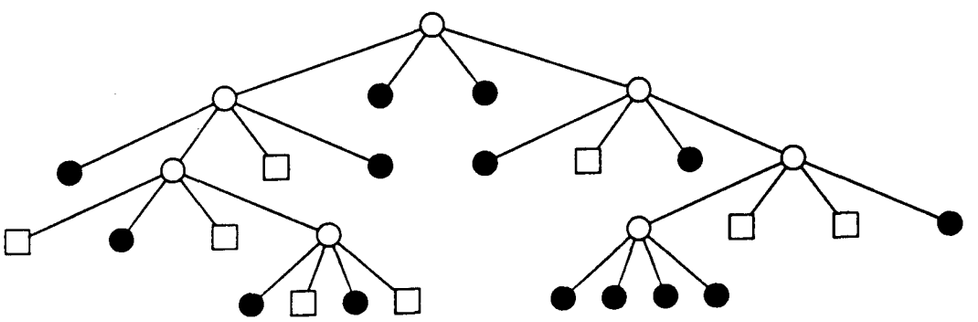

Figure 2

(Dans la figure 2 ci-dessus, les cellules vides sont des carrés et les cellules qui ne contiennent qu'un corps sont des cercles pleins.)

La profondeur d'un nœud d'un quadtree est le nombre d'arcs à parcourir pour rejoindre la racine à partir de ce nœud. La profondeur d'un quadtree est la plus grande profondeur des nœuds de ce quadtree. Par exemple, la profondeur du quadtree dessiné ci-dessus est quatre.

1. On étudie quelques propriétés des quadtrees.

a) Dessiner la division en cellules de deux univers de profondeur trois qui contiennent le plus et le moins de corps possibles.

b) Donner, en la justifiant brièvement, une minoration fonction de N, $f(N)$, de la profondeur d'un quadtree contenant N corps. Cette minoration sera atteinte, c'est-à-dire que pour tout entier p_0, il existe un quadtree contenant N_0 corps et dont la profondeur est p_0, avec $f(N_0) = p_0$.

c) Montrer qu'il n'existe pas de majoration fonction de N de la profondeur d'un quadtree.

d) Soit δ, la quantité

$$\delta = \min_{c \in \mathcal{U}, c' \in \mathcal{U}}^{c \neq c'} (Max(|p_c.x - p_{c'}.x|, |p_c.y - p_{c'}.y|))$$

Donner une borne supérieure de la profondeur d'un quadtree représentant $\mathcal{U}$ en fonction de δ et de $D_{\mathcal{U}}$. On supposera $\delta > 0$.

e) On suppose que les représentations en machine des grandeurs homogènes à une distance sont des nombres flottants du même ordre de grandeur. C'est-à-dire que leur valeur absolue est assimilable à un entier codé en base deux à l'aide de m bits multiplié par un ordre de grandeur constant 2e. En déduire un majorant absolu de la profondeur des quadtrees manipulés par notre programme.

Le candidat doit à partir de maintenant choisir un langage de programmation. S'il choisit Caml, il poursuivra le problème par la deuxième partie-Caml ci-dessous ; s'il choisit Pascal, il poursuivra le problème par la deuxième partie-Pascal.

Deuxième partie-Caml : Construction du quadtree

Un vecteur de l'espace plan est représenté par un enregistrement dont le type est le suivant :

```
type vecteur = {x:float ; y:float};;
```

À titre d'exemple, le vecteur nul est défini comme suit :

```
let vecteur_nul = {x = 0.0 ; y = 0.0};;
```

Un corps c est la donnée d'une masse m_c, d'un vecteur position $\vec{p}_c$, d'une vitesse $\vec{v}_c$ et d'une accélération $\vec{a}_c$. Le type des corps est un type enregistrement :

```
type corps = { mass:float ;
  mutable pos:vecteur ; mutable vel:vecteur ; mutable acc:vecteur};;
```

Rappel. Les champs position, vitesse et accélération sont mutables car ils changeront au cours du temps. On pourra donc, par exemple, donner une nouvelle valeur `v` à l'accélération d'un corps `c` par l'opération `c.acc <- v`.

Dans toute la suite du problème l'univers $\mathcal{U}$ est un tableau rangé dans la variable globale **univers**. La taille `N` du tableau **univers** est également une variable globale du programme. On suppose que tous les corps occupent des positions deux à deux distinctes à tout instant.

2. Écrire les fonctions qui réalisent les opérations suivantes sur les vecteurs :

a) Les fonctions **add** et **sub**, de type `vecteur -> vecteur -> vecteur`, prennent deux vecteurs $\vec{u}$ et $\vec{v}$ en arguments et renvoient respectivement la somme vectorielle $\vec{u} + \vec{v}$ et la différence $\vec{u} - \vec{v}$ comme résultats.

b) La fonction **scal**, de type `float -> vecteur -> vecteur` prend un scalaire m et un vecteur $\vec{u}$ et renvoie le produit $m \cdot \vec{u}$.

c) La fonction **carre**, de type `vecteur -> float`, renvoie le carré de la norme euclidienne d'un vecteur $\vec{u}$ passé en argument.

3. En Caml, le type `arbre` des quadtrees est le suivant :

```
type arbre = Noeud of cellule | Feuille of corps | Vide
and cellule =
  {mutable cm_mass:float;        (* masse des corps de la cellule *)
   mutable cm_pos:vecteur;       (* centre de masse des corps de la cellule *)
           filles:arbre vect     (* 4 sous-arbres *)
};;
```

On observe que les sous-cellules d'une cellule sont rangées dans un tableau et sont donc numérotées. Préciser un ordre de numérotation des sous-cellules et écrire les deux fonctions suivantes :

a) La fonction `indice_fille` est de type `vecteur->vecteur->int`. Étant données une position `p` et une cellule C, centrée en `p_c`, telles que C contient la position `p`, l'appel `indice_fille p_c p` renvoie l'indice de la sous-cellule de C qui contient la position `p`.

b) La fonction `position_fille` est de type `vecteur->float->int->vecteur`. Étant donnée une cellule C, de côté `taille` et centrée en `p_c`, l'appel `position_fille p_c taille i` renvoie la position du centre de la sous-cellule d'indice `i` de la cellule C.

4. L'arbre-univers, représentation arborescente de $\mathcal{U}$, sera construit par insertion successive de tous les corps de l'univers dans une cellule racine dont le centre est à l'origine des coordonnées et dont le côté $D_{\mathcal{U}}$ est donné par la variable globale `taille_univers`.

```
let construire_arbre () =
   let arbre_univers = ref Vide in
   for i=0 to N-1 do
     arbre_univers :=
       insere_corps univers.(i)  !arbre_univers vecteur_nul taille_univers
   done ;
   !arbre_univers
;;
```

a) Écrire la fonction `insere_corps`, de type `corps->arbre->vecteur->float->arbre`. Cette fonction prend en arguments un corps à insérer, un arbre où insérer ce corps, ainsi que la position du centre et la valeur du côté de la cellule qu'il représente. Elle renvoie un quadtree adaptatif qui contient le nouveau corps en plus de tous les autres. On ne cherchera pas à positionner correctement les champs `cm_mass` et `cm_pos` des cellules.

b) Prouver que la fonction `insere_corps` termine, à condition que le côté de la cellule racine soit suffisamment grand et que tous les corps occupent des positions distinctes.

c) Donner la complexité asymptotique en fonction de N de `construire_arbre`. On se placera dans les conditions de la question 1. e), c'est-à-dire que la profondeur des quadtrees est bornée indépendamment de N.

Troisième partie-Caml : Calcul des forces

5. Il reste à positionner correctement les champs `cm_mass` et `cm_pos` dans l'arbre-univers. Écrire la fonction `barycentres` de type `arbre->unit` qui réalise cette opération. On rappelle que la position $\vec{p}_C$ du centre de masse ou barycentre des corps de la cellule C est donnée par la formule suivante :

$$\vec{p}_C = \frac{1}{M_C} \cdot \sum_{c' \in C} m_{c'} \cdot \vec{p}_{c'}, \quad \text{avec} \quad M_C = \sum_{c' \in C} m_{c'}$$

6. Selon les lois de la mécanique, l'accélération du corps c soumis à l'attraction des autres corps de l'univers est calculée en sommant les contributions de tous les autres corps.

$$\vec{a}_c = \sum_{c' \in \mathcal{U}}^{c' \neq c} a_{c' \to c}, \quad \text{avec} \quad a_{c' \to c} = \frac{m_{c'}}{||\vec{p}_{c'} - \vec{p}_c||^3} \cdot (\vec{p}_{c'} - \vec{p}_c)$$

On considère maintenant un corps c et une cellule C de côté D, dont la masse est M_C et le centre de masse est positionné en $\vec{p}_C$. Soit $\vec{r}$ le vecteur $\vec{p}_C - \vec{p}_c$ et r sa norme. On estime que la force gravitationelle exercée par les corps de C sur c peut être assimilée à celle exercée par leur centre de masse affecté de la masse M_C quand le rapport D/r est strictement inférieur à une constante θ. On a alors :

$$\sum_{c' \in C} a_{c' \to c} \approx \frac{M_C}{r^3} \cdot \vec{r}$$

L'accélération d'un corps c sera estimée en sommant les contributions des autres corps au cours d'un parcours partiel de l'arbre univers. Si, au cours de ce parcours, une cellule C satisfait le critère d'approximation, alors on remplacera la contribution de ses corps par celle de son centre de masse. Dans le cas contraire on continuera de subdiviser la cellule C. La fonction `grav_approx` de type `vecteur -> arbre -> float -> vecteur` réalise ce parcours partiel, elle est appelée sur tous les corps de l'univers :

```
let ajuste_acc_approx arbre_univers =
  for i=0 to N-1 do
    univers.(i).acc <-
      grav_approx univers.(i).pos arbre_univers taille_univers ;
  done ;;
```

a) Écrire la fonction **grav_approx**. On supposera que la constante θ est rangée dans une variable globale `theta`.

b) Soit un corps c dont on calcule l'accélération par la méthode approximative. Montrer qu'il existe une fonction de θ seul, $K(\theta)$, qui majore le nombre de cellules d'une taille donnée qui sont subdivisées pendant le calcul de l'accélération de c.

c) Donner la complexité asymptotique en N des fonctions **grav_approx** et **ajuste_acc_approx**, en supposant que la profondeur des quadtrees est bornée (cf. la question 1. e)).

Deuxième partie-Pascal : Construction du quadtree

Un vecteur de l'espace plan est représenté par un enregistrement du type suivant :

```
vecteur = record x,y : real end ;
```

À titre d'exemple, le vecteur nul est une variable globale de type **vecteur** et initialisée comme suit :

```
vecteur_nul.x := 0.0;
vecteur_nul.y := 0.0;
```

Un corps c est la donnée d'une masse m_c, d'un vecteur position $\vec{p}_c$, d'une vitesse $\vec{v}_c$ et d'une accélération $\vec{a}_c$. Le type des corps est un type enregistrement :

```
corps = record
  mass : real ;
  pos,vel,acc : vecteur
end ;
```

Dans toute la suite du problème l'univers $\mathcal{U}$ est un tableau rangé dans la variable globale **univers**. La taille `N` du tableau **univers** est une constante du programme. On suppose que tous les corps occupent des positions deux à deux distinctes à tout instant.

2. Écrire les procédures et la fonction qui réalisent les opérations suivantes sur les vecteurs :

a) Les procédures **add** et **sub** prennent trois vecteurs $\vec{u}$, $\vec{v}$ et $\vec{r}$ en arguments et rangent respectivement la somme vectorielle $\vec{u}+\vec{v}$ et la différence $\vec{u}-\vec{v}$ dans le vecteur $\vec{r}$.

b) La procédure **scal** prend un scalaire m et deux vecteurs $\vec{u}$ et $\vec{r}$ en arguments et range le produit $m \cdot \vec{u}$ dans $\vec{r}$.

c) La fonction **carre** renvoie le carré de la norme euclidienne d'un vecteur $\vec{u}$ passé en argument.

N.B. On se conformera aux déclarations de type suivantes:

```
procedure add(u,v : vecteur ; var r : vecteur);
procedure sub(u,v : vecteur ; var r : vecteur);
procedure scal(m : real ; u : vecteur ; var r : vecteur);
function carre(u : vecteur) : real;
```

3. En Pascal, le type **arbre** des quadtrees est le suivant :

```
type nature = (Feuille, Noeud) ;
type arbre = ^cellule;
cellule = record
  case nature:nature of
    Feuille: (corps:1..N);
    Noeud:
      (cm_mass:real ;   { masse des corps de la cellule }
       cm_pos:vecteur ; { centre de masse }
       filles:array [0..3] of arbre) { quatre sous-arbres }
end ;
```

On observe que les sous-cellules d'une cellule sont rangées dans un tableau et sont donc numérotées. Préciser un ordre de numérotation des sous-cellules et écrire les deux fonctions suivantes :

a) La fonction **indice_fille** dont la déclaration est :

```
function indice_fille (p_c,p:vecteur):integer ;
```

Étant données une cellule C centrée en `p_c` et une position de corps `p` située dans C, telles que C contient la position `p`, la fonction **indice_fille** renvoie l'indice de la sous-cellule de C qui contient la position `p`.

b) La fonction **position_fille** dont la déclaration est :

```
function position_fille(p:vecteur ; taille:real ; i:integer):vecteur;
```

La fonction **position_fille** renvoie la position du centre de la sous-cellule d'indice `i` de la cellule de côté **taille** centrée en `p`.

4. Par convention une cellule vide est représentée par le pointeur **nil**. Pour ne pas parsemer ses programmes d'appels à la primitive **new** et s'assurer que l'on utilise uniquement des cellules bien formées, il est vivement recommandé d'utiliser les deux fonctions suivantes pour allouer des feuilles et des nœuds internes :

```
function cree_feuille(c:1..N):arbre ;
var
  p:arbre ; { pointeur resultat }
```

```
begin
  new(p) ;
  p^.nature := Feuille ; { d'abord affecter le champ de selection }
  p^.corps := c ;
  cree_feuille := p
end;

function cree_noeud(m:real ; pos:vecteur ; f0,f1,f2,f3:arbre):arbre;
var
  p:arbre ; { pointeur resultat }
begin
  new(p) ;
  p^.nature := Noeud ; { d'abord affecter le champ de selection }
  p^.cm_mass := m ;
  p^.cm_pos := pos ;
  p^.filles[0] := f0 ;
  p^.filles[1] := f1 ;
  p^.filles[2] := f2 ;
  p^.filles[3] := f3 ;
  cree_noeud := p
end ;
```

L'arbre-univers, représentation arborescente de $\mathcal{U}$, sera construit par insertion successive de tous les corps de l'univers dans une cellule racine dont le centre est à l'origine des coordonnées et dont le côté $D_{\mathcal{U}}$ est donné par la variable globale **taille_univers**.

```
procedure construire_arbre;
var
  i:integer;
begin
  arbre_univers := nil ; { arbre vide }
  for i := 1 to N do
    insere_corps(i,arbre_univers,vecteur_nul,taille_univers);
end ;
```

a) Écrire la procédure **insere_corps**, dont la déclaration est :

```
procedure insere_corps(c:1..N ; var a:arbre ; p:vecteur ; taille:real);
```

Cette procédure prend en arguments un corps à insérer, un arbre où insérer ce corps, ainsi que la position du centre et la valeur du côté de la cellule qu'il représente. Elle modifie son deuxième argument qui doit maintenant être un quadtree adaptatif qui contient le nouveau corps en plus de tous les autres. On ne cherchera pas à positionner correctement les champs **cm_mass** et **cm_pos** des cellules.

b) Prouver que la procédure **insere_corps** termine, à condition que le côté de la cellule racine soit suffisamment grand et que tous les corps occupent des positions distinctes.

c) Donner la complexité asymptotique en fonction de N de **construire_arbre**. On se placera dans les conditions de la question 1. e), c'est-à-dire que la profondeur des quadtrees est bornée indépendamment de N.

Troisième partie-Pascal : Calcul des forces

5. Il reste à positionner correctement les champs **cm_mass** et **cm_pos** dans l'arbre-univers. Écrire la procédure **barycentres** qui réalise cette opération. On rappelle que la position $\vec{p}_C$ du centre

de masse ou barycentre des corps de la cellule C est donnée par la formule suivante :

$$\vec{p}_C = \frac{1}{M_C} \cdot \sum_{c' \in C} m_{c'} \cdot \vec{p}_{c'}, \quad \text{avec} \quad M_C = \sum_{c' \in C} m_{c'}$$

On se conformera à la déclaration de type suivante :

```
procedure barycentres(a : arbre);
```

6. Selon les lois de la mécanique, l'accélération du corps c soumis à l'attraction des autres corps de l'univers est calculée en sommant les contributions de tous les autres corps.

$$\vec{a}_c = \sum_{c' \in \mathcal{U}}^{c' \neq c} a_{c' \to c}, \quad \text{avec} \quad a_{c' \to c} = \frac{m_{c'}}{||\vec{p}_{c'} - \vec{p}_c||^3} \cdot (\vec{p}_{c'} - \vec{p}_c)$$

On considère maintenant un corps c et une cellule C de côté D, dont la masse est M_C et le centre de masse est positionné en $\vec{p}_C$. Soit $\vec{r}$ le vecteur $\vec{p}_C - \vec{p}_c$ et r sa norme. On estime que la force gravitationelle exercée par les corps de C sur c peut être assimilée à celle exercée par leur centre de masse affecté de la masse M_C quand le rapport D/r est strictement inférieur à une constante θ. On a alors :

$$\sum_{c' \in C} a_{c' \to c} \approx \frac{M_C}{r^3} \cdot \vec{r}$$

L'accélération d'un corps c sera estimée en sommant les contributions des autres corps au cours d'un parcours partiel de l'arbre univers. Si, au cours de ce parcours, une cellule C satisfait le critère d'approximation, alors on remplacera la contribution de ses corps par celle de son centre de masse. Dans le cas contraire on continuera de subdiviser la cellule C. La fonction **grav_approx** réalise ce parcours partiel, elle est appelée sur tous les corps de l'univers :

```
function grav_approx(p:vecteur ; a:arbre ; taille:real):vecteur;
 ...

procedure ajuste_acc_approx;
var
  i:integer;
begin
  for i := 1 to N do
  begin
    univers[i].acc :=
      grav_approx(univers[i].pos,arbre_univers,taille_univers) ;
  end
end;
```

a) Écrire la fonction **grav_approx**. On supposera que la constante θ est rangée dans une variable globale **theta**.

b) Soit un corps c dont on calcule l'accélération par la méthode approximative. Montrer qu'il existe une fonction de θ seul, $K(\theta)$, qui majore le nombre de cellules d'une taille donnée qui sont subdivisées pendant le calcul de l'accélération de c.

c) Donner la complexité asymptotique en N des fonctions **grav_approx** et **ajuste_acc_approx**, en supposant que la profondeur des quadtrees est bornée (cf. la question **1.e**)).

www.tunisie-etudes.info

Ce document a été téléchargé depuis
www.tunisie-etudes.info

Des documents gratuits, devoirs, examens, cours, exercices, corrigés... Ainsi que toute une rubrique pour vous aider à trouver un emploi sans oublier les avis de concours en direct

Notre page Twitter :
http://www.twitter.com/TunisieEtudes
Notre page FaceBook :
http://www.facebook.com/TunisieEtudes



Merci d'avoir choisi www.tunisie-etudes.info
Bonne lecture et bon travail