Les sources des exercices de Travaux Pratiques sont disponibles sur le site de l'UE en
`http://remi.morin.myspace.luminy.univmed.fr/I2/PP`

Récupérez l'archive compressée `TPA.tgz` disponible sur le site de l'UE et décompressez-la avec
`tar -xvzf TPA.tgz`
Vous obtiendrez les codes utilisés en cours, en travaux dirigés et dans cette planche de travaux pratiques.

Exercice A.1 Partage statique et dynamique d'une tâche L'ensemble de Mandelbrot est une fractale définie comme l'ensemble des points c du plan complexe pour lesquels la suite définie par : $z_{n+1} = z_n^2 + c$ avec $z_0 = 0$ ne tend pas, en module, vers l'infini. Si nous reformulons cela sans utiliser les nombres complexes, en remplaçant z_n par le couple de réels (x_n, y_n) et c par le couple (a, b), alors nous obtenons :

$$x_{n+1} = x_n^2 - y_n^2 + a \text{ et } y_{n+1} = 2.x_n.y_n + b \text{ avec } x_0 = y_0 = 0$$

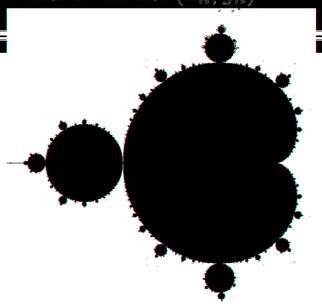
la condition devenant que ni x_n, ni y_n ne tendent vers l'infini (en valeur absolue).

Dans la pratique, nous allons calculer les premiers termes de cette suite pour un point (a, b) fixé. Si la valeur de $x^2 + y^2$ dépasse 4, nous considérerons (de manière arbitraire) que la suite diverge, et donc

erge pas, et donc que le point (a, b) est dans
qui détermine si oui ou non le point (a, b) est
premiers éléments de la suite (x_n, y_n).

des termes calculés, nous considérerons que la suite ne di
l'ensemble. Ceci nous conduit à utiliser la méthode suivante
dans l'ensemble de Mandelbrot lorsque l'on examine les pr

```
static boolean mandelbrot(double a, double b, int
  double x = 0;
  double y = 0;
  for (int t = 0; t < max; t++) {
    if (x*x + y*y > 4.0) return false;
    double nx = x*x - y*y + a;
    double ny = 2*x*y + b;
    x = nx;
    y = ny;
  }
  return true;
}
```



rchive `TPA.tgz` est donné sur la figure 4. Il
gion carrée comprise entre $(-1.5, -1)$ en bas à
ligne (et par colonne) est fixé à `taille=500`
e programme indique également le temps de
e en supprimant l'instruction `image.show()`.

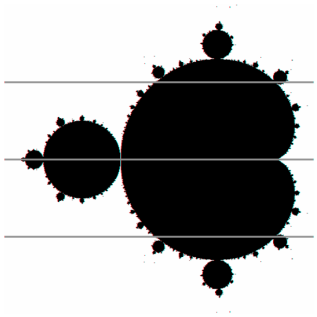
Le programme `Mandelbrot.java` disponible dans l'a
calcule et affiche l'ensemble de Mandelbrot situé dans la rég
gauche et $(0.5, 1)$ en haut à droite. Le nombre de pixels par
alors que le nombre maximal d'itérations vaut `max=200`. I
calcul de l'image. Vous pouvez annuler l'affichage de l'imag

ur que le
trentaine

ads, cha-

à la ver-
le chacun

uls aux 4
les. Quel
pport à la



Question 1. Modifiez les valeurs de `taille` ou de `max` po
calcul de l'image sur votre machine dure une
de secondes.

Question 2. Partagez ensuite le calcul de l'image sur 4 thre
cun calculant un quart de l'image.

Question 3. Quel est le gain en temps de calcul par rapport
sion séquentielle ? Quel est le temps de calcul d
des quatre threads ?

Question 4. Attribuez à présent à la volée les lignes de calc
threads au fur et à mesure qu'ils sont disponib
est le nouveau gain, en temps de calcul, par ra
version séquentielle ?

te-Carlo Nous poursuivons l'exercice I.2 de la
_PiSurQuatre.java donné dans l'archive
e utilisant 10 threads pour accélérer ce calcul.

Exercice A.2 Evaluation de $\pi/4$ par la méthode de Mont
page 1. À partir du code séquentiel du programme `Calcul`
et indiqué sur la figure 1, coder et tester une version parallèl