

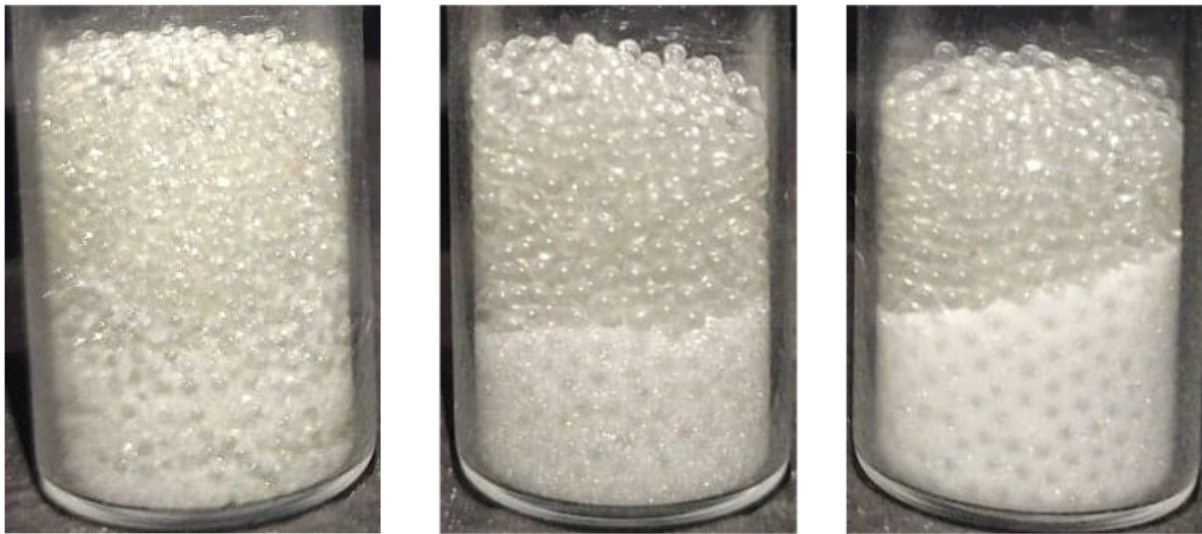
# Détection des contacts

# Détection des contacts

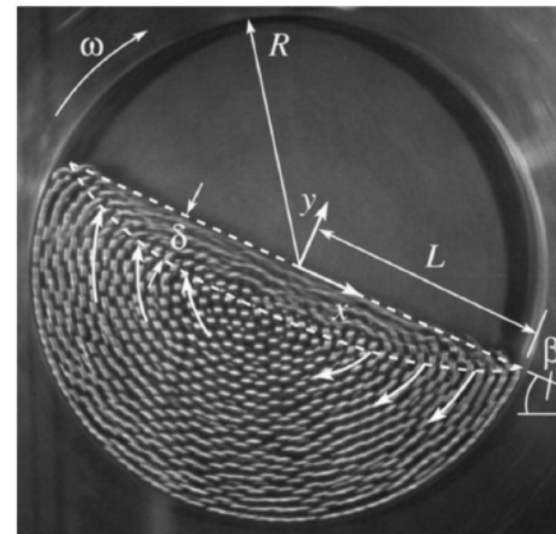
- ▶ Nous venons de voir comment modéliser les déplacements d'**une** particule.
- ▶ Généralement un système sera composé de **plusieurs** de ces particules.
- ▶ En fonction de leurs natures, ces particules vont **interagir** l'une avec les autres.
- ▶ L'interaction la plus basique en physique des matières molles est la **collision binaire**.
- ▶ Nous allons voir maintenant comment déterminer de façon efficace si deux solides sont en **contact**.

# Types des contacts

- ▶ Les systèmes que nous simulons sont composés principalement de **deux** types de solides.
- ▶ D'une part des disques (grains, colloïdes, ...) et d'autre part des courbes (parois du système).



*Pillitteri et al., Soft Matter (2020)*

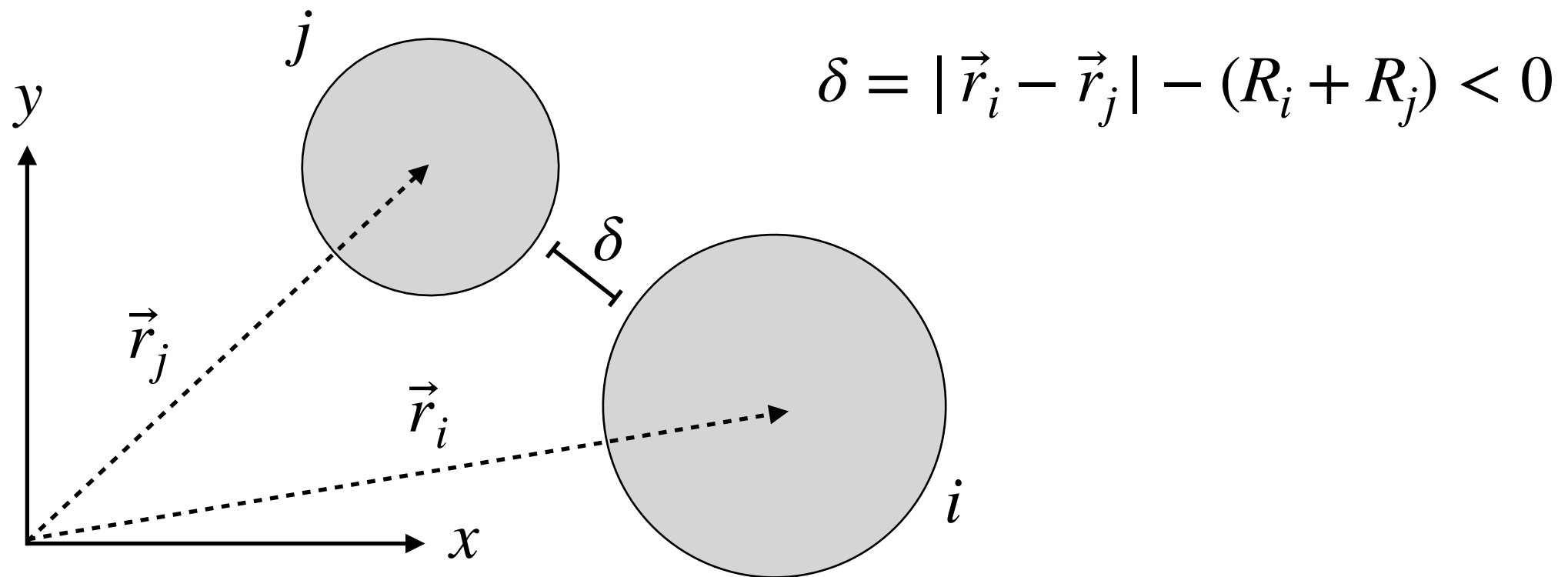


*Ottino et al., (2002)*

- ▶ Il faudra donc gérer de nombreuses **interactions** de type « disque-courbe » et de type « disque-disque ».

# Contact entre deux disques

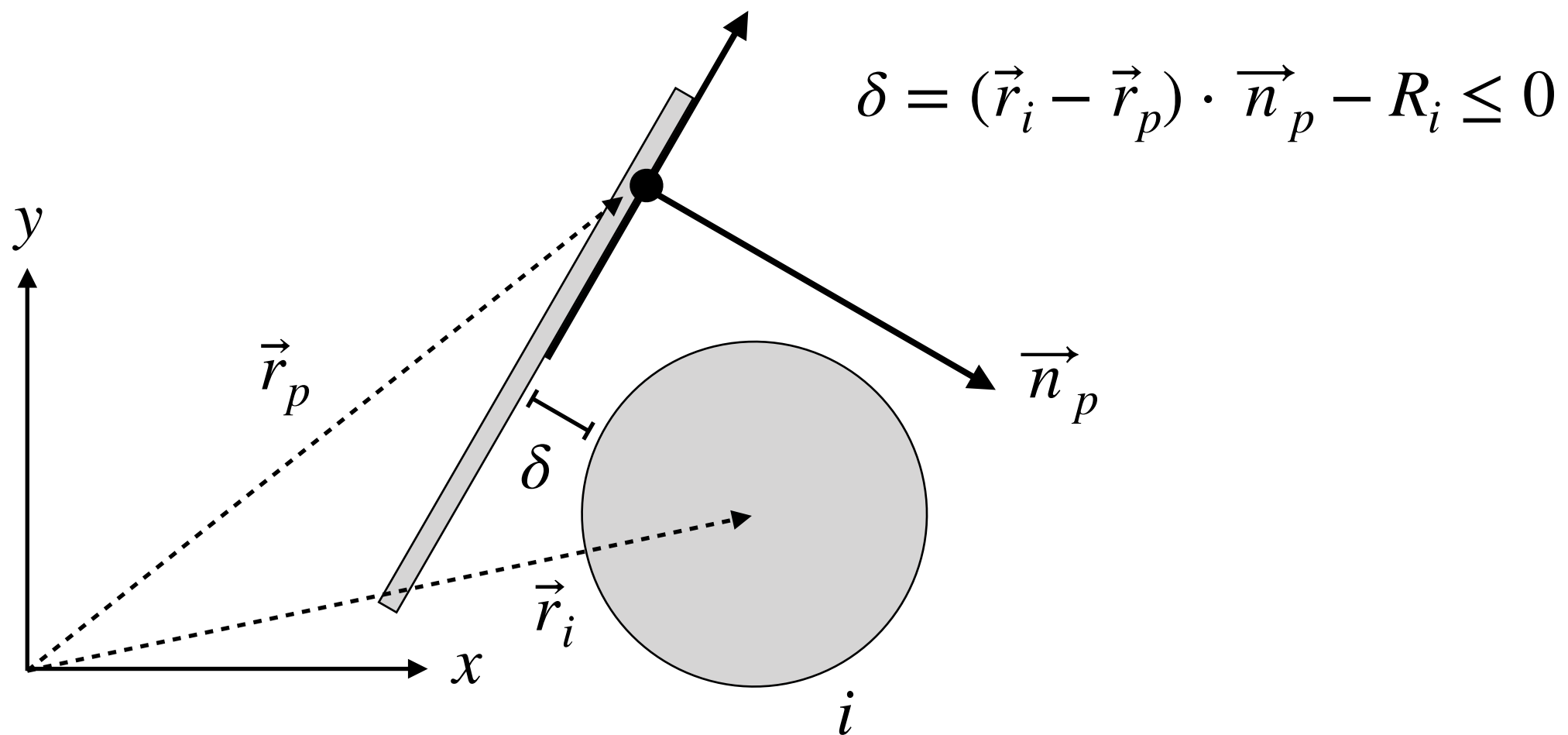
► **Deux disques** de rayons  $R_i$  et  $R_j$  placés respectivement en  $\vec{r}_i$  et  $\vec{r}_j$  sont en **contact** si,



► La grandeur  $\delta$  est la distance **surface à surface**, un élément clef qui va revenir plus tard.

# Contact entre disque une droite

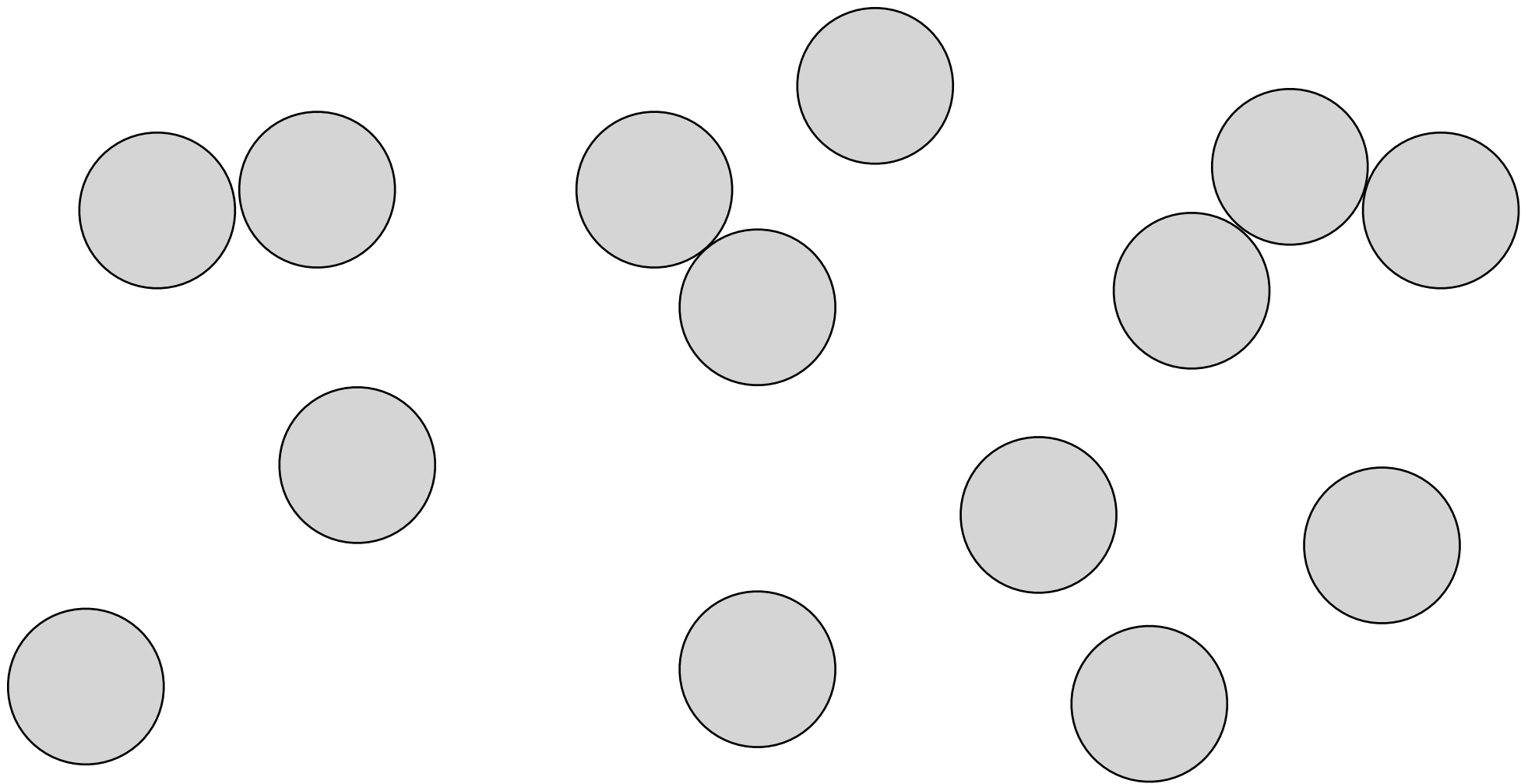
► Un **disque** de rayon  $R_i$  placé en  $\vec{r}_i$  et une **droite** placée en  $\vec{r}_p$  avec une orientation  $\vec{n}_p$  sont en **contact** si,



► **Remarque:** Par la suite, on utilisera le terme particules plutôt que disques et parois au lieu de droites.

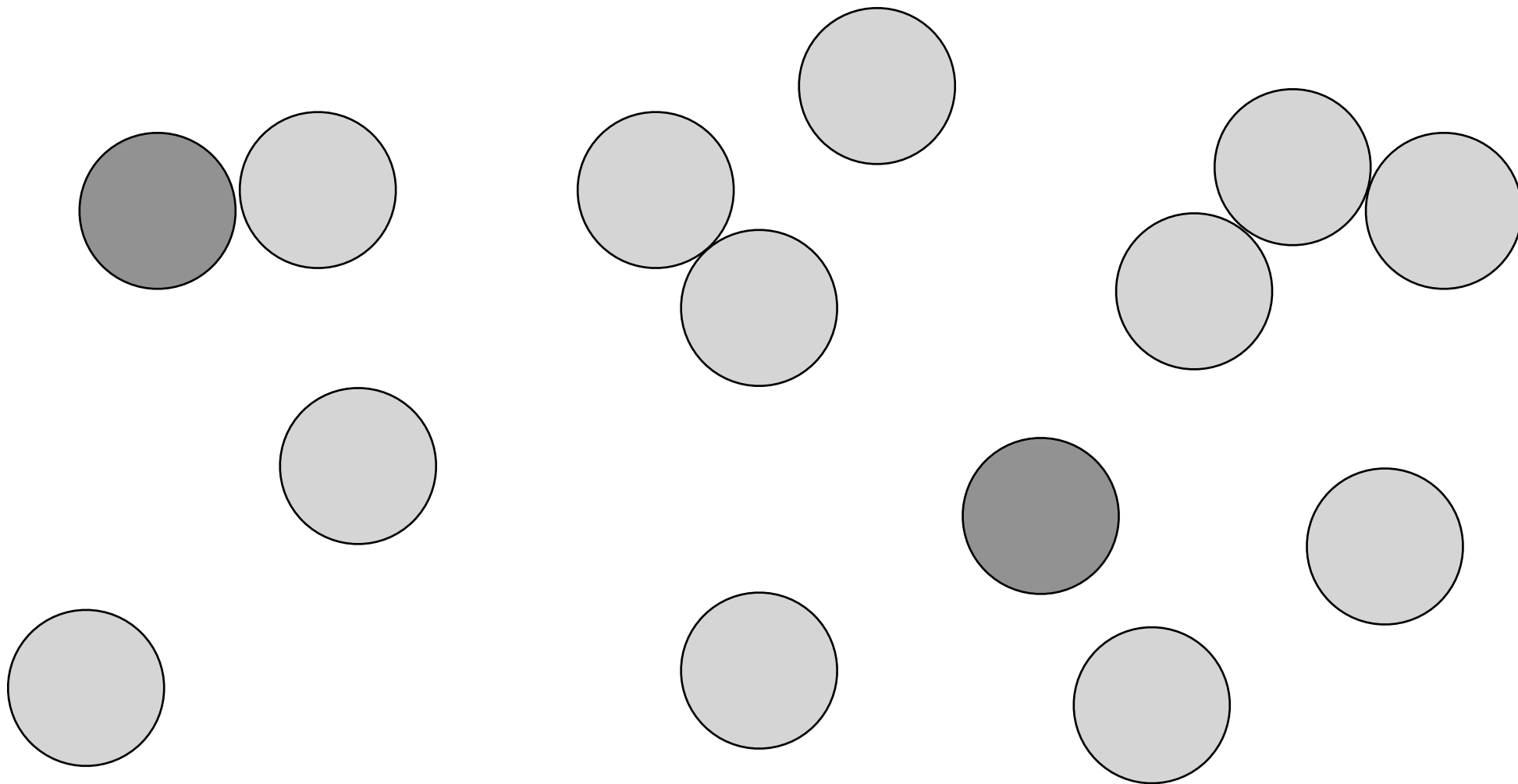
# Détection des contacts

- ▶ Dans un système composé de  $N$  particules, **combien** de tests de contact doit-on effectuer?



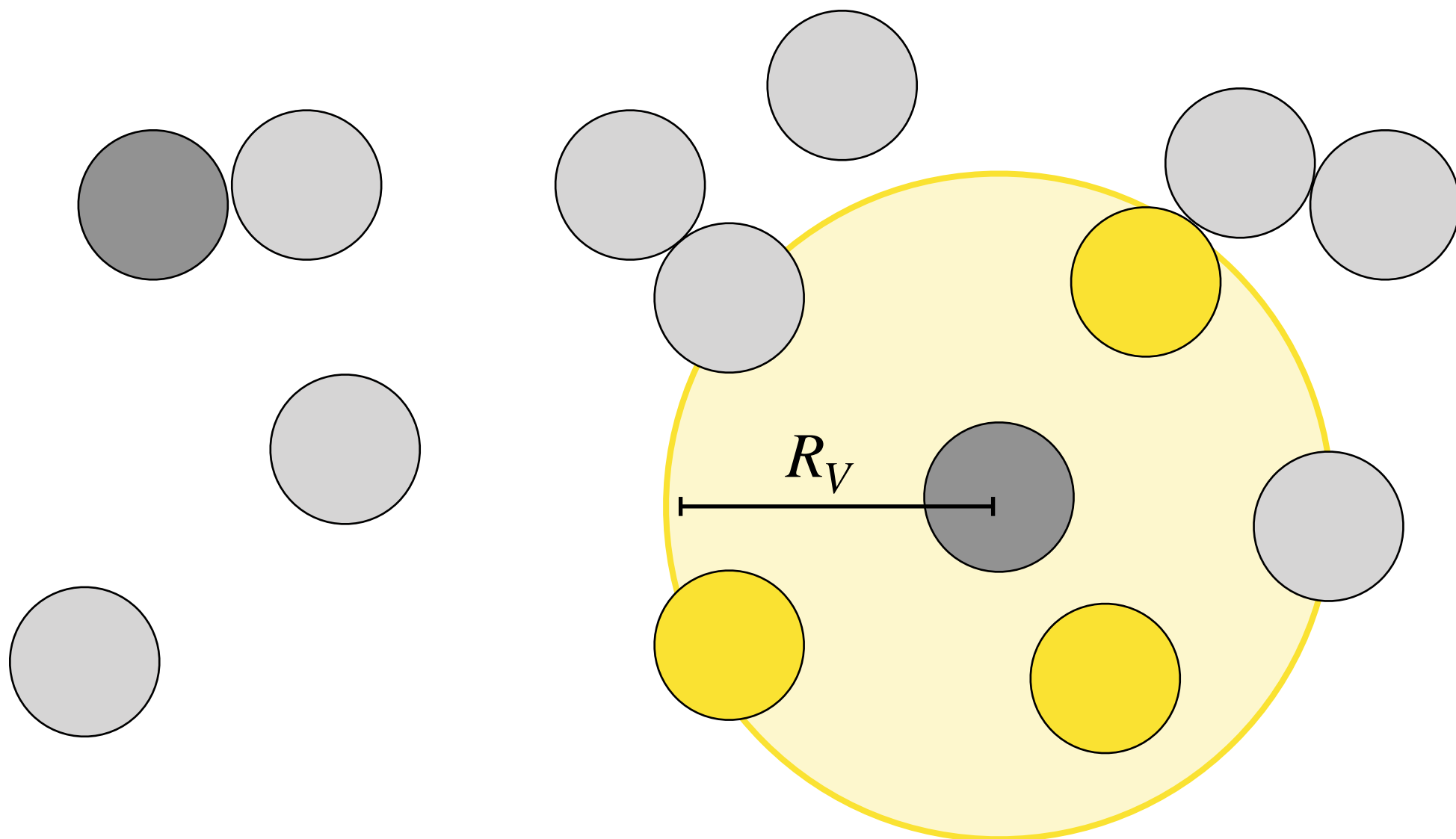
# Détection des contacts

- ▶ Est-ce que les deux particules foncées risquent d'entrer en collision **prochainement**?



# Détection des contacts

- ▶ Les prochains candidats pour une collision se situent toutes dans le **voisinage** des particules.





# Liste de Verlet

- ▶ Pour chaque particule, on peut donc construire une **liste de voisins**, appelée liste de Verlet.
- ▶ Cependant, la construction de cette liste nécessite un grand nombre de **calculs** de distance.
- ▶ Idéalement, il faudrait donc que la liste reste **valable** pendant **longtemps** (de nombreux pas de temps).
- ▶ Malheureusement, la durée de validité va dépendre de la **taille** du voisinage et de la **vitesse** des particules.

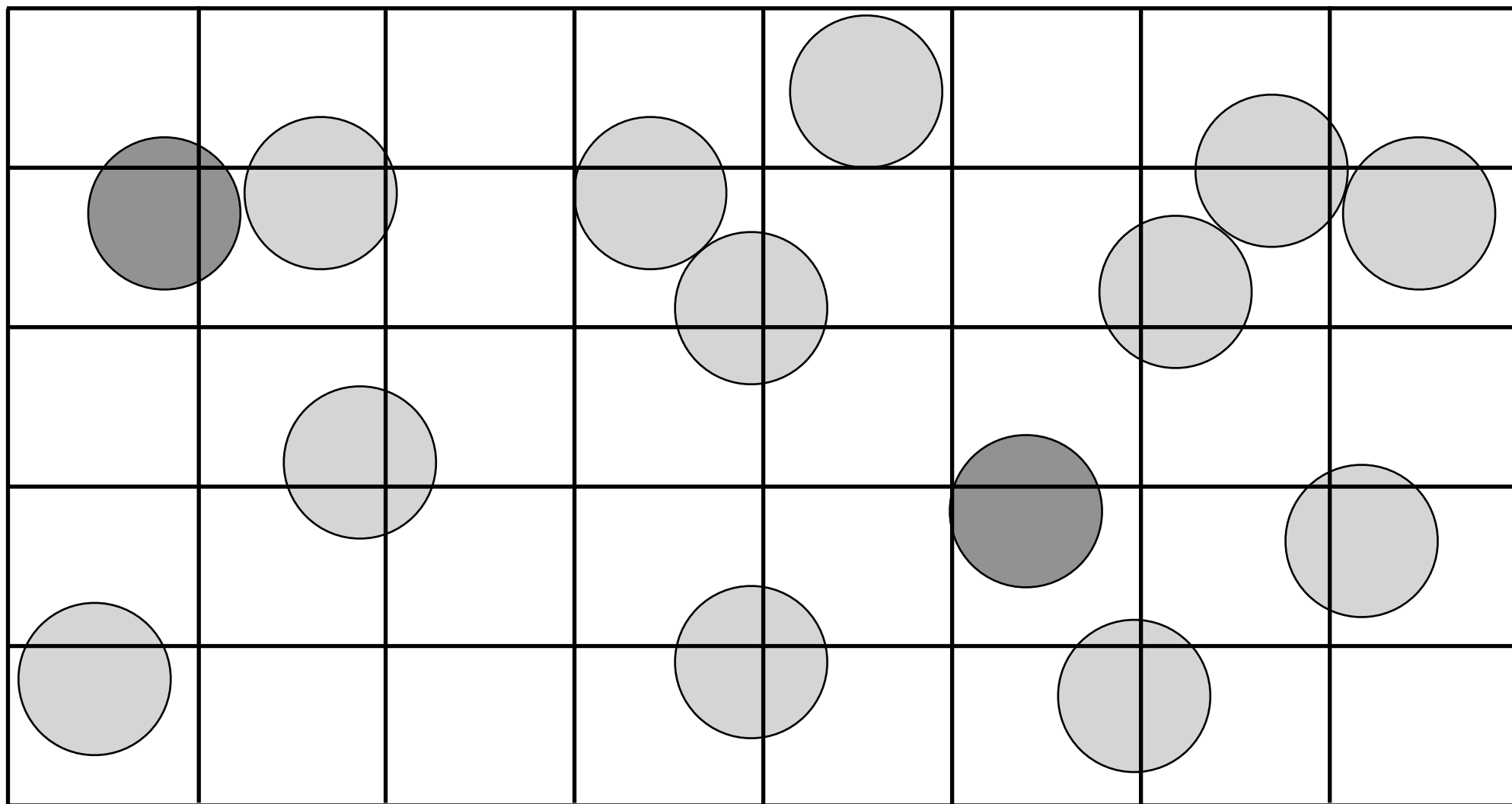
$$T_V \sim R_V / \langle v \rangle$$

# Liste de Verlet

- ▶ Si on utilise un **petit** voisinage, on a peu de tests à réaliser mais il faut souvent actualiser la liste.
- ▶ Si on utilise un **grand** voisinage, on a beaucoup de tests à réaliser mais on actualise moins souvent.
- ▶ Trouver un bon **compromis** n'est pas aisé et dépend fortement du système étudié.
- ▶ Nous allons donc essayer de trouver une **autre méthode** (encore plus) efficace.

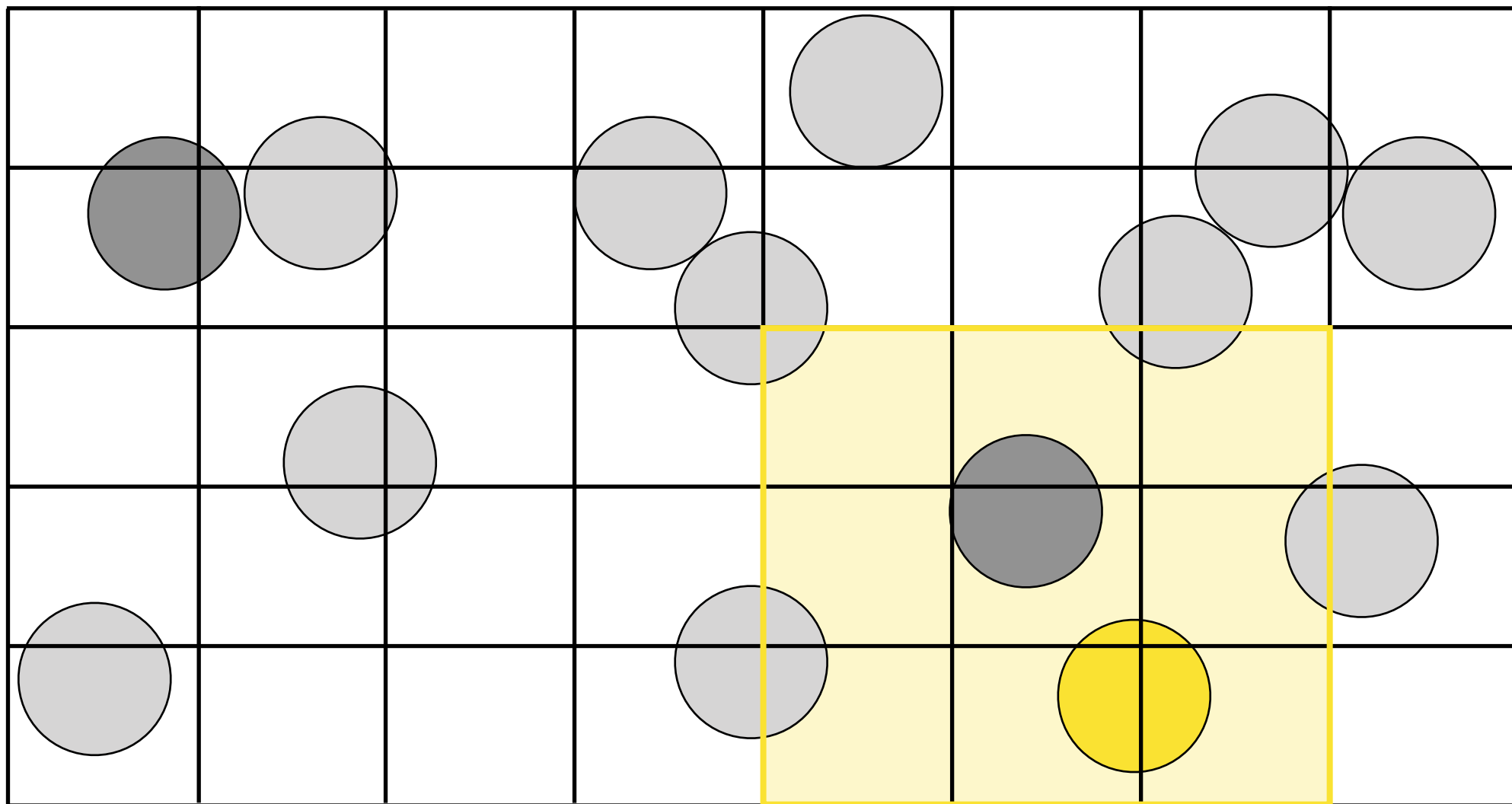
# Voisinage en cellules

- ▶ Au lieu de centrer un voisinage circulaire sur chaque particule, on place une **grille** au dessus du système.



# Voisinage en cellules

- ▶ Le **voisinage** de chaque particule est alors composé de la cellule dans laquelle il se trouve et de ses voisines.



# Voisinage en cellules

- ▶ Etablir une liste des **candidates** au contact nécessite alors trois informations:

Chaque particule connaît sa cellule

Chaque cellule connaît ses cellules voisines

Chaque cellule connaît ses particules

- ▶ Contrairement à la liste de Verlet, **classer** les particules dans leurs cellules est très rapide.
- ▶ On peut donc utiliser des cellules **petites** et classer les particules à **chaque** pas de temps.

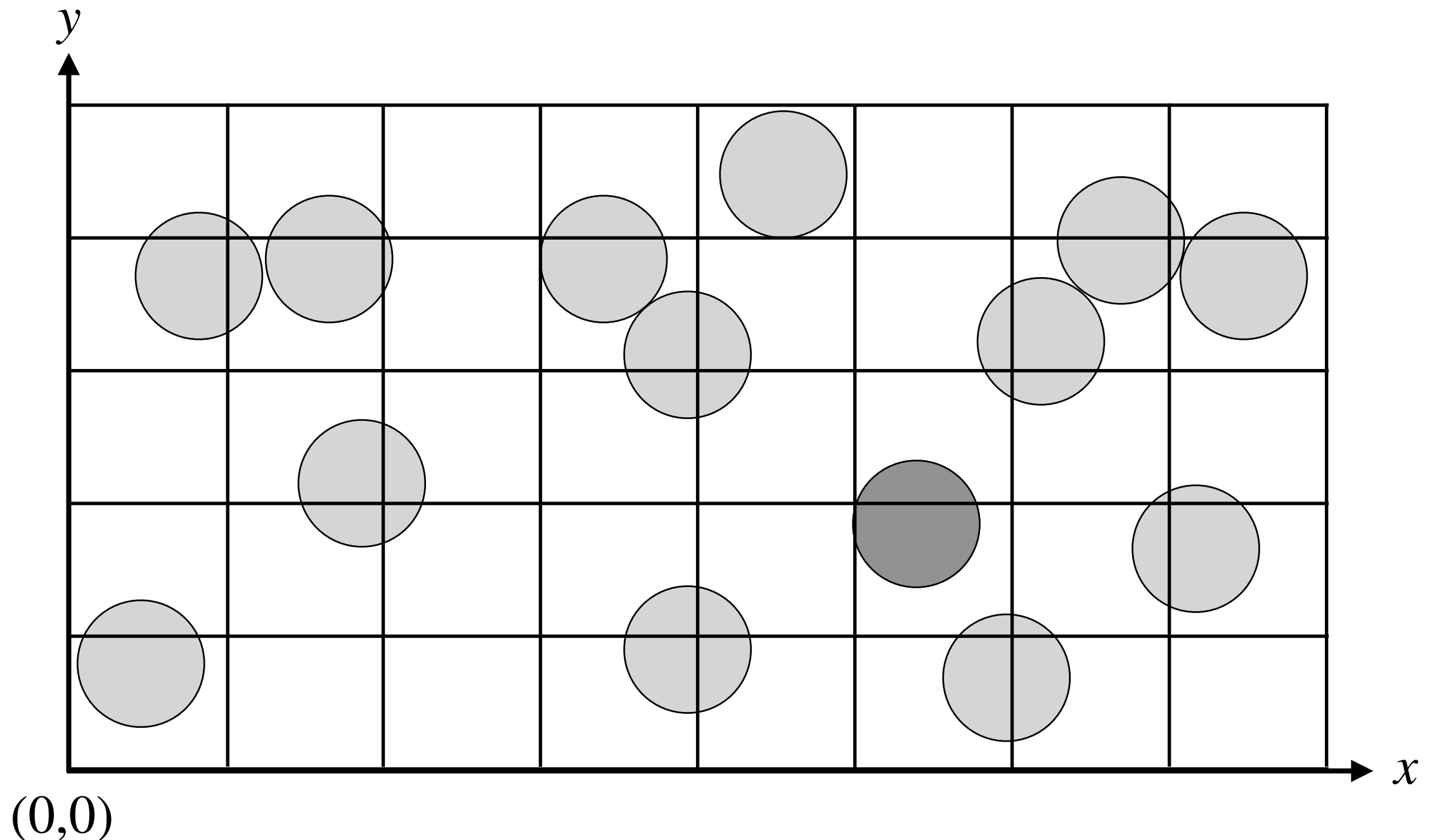
# Connaitre sa cellule

- ▶ Comme dans une matrice  $m \times n$ , chaque cellule peut être identifiée par un **couple** de nombres.

{0,2}							
{0,1}	{1,1}						
{0,0}	{1,0}	{2,0}					

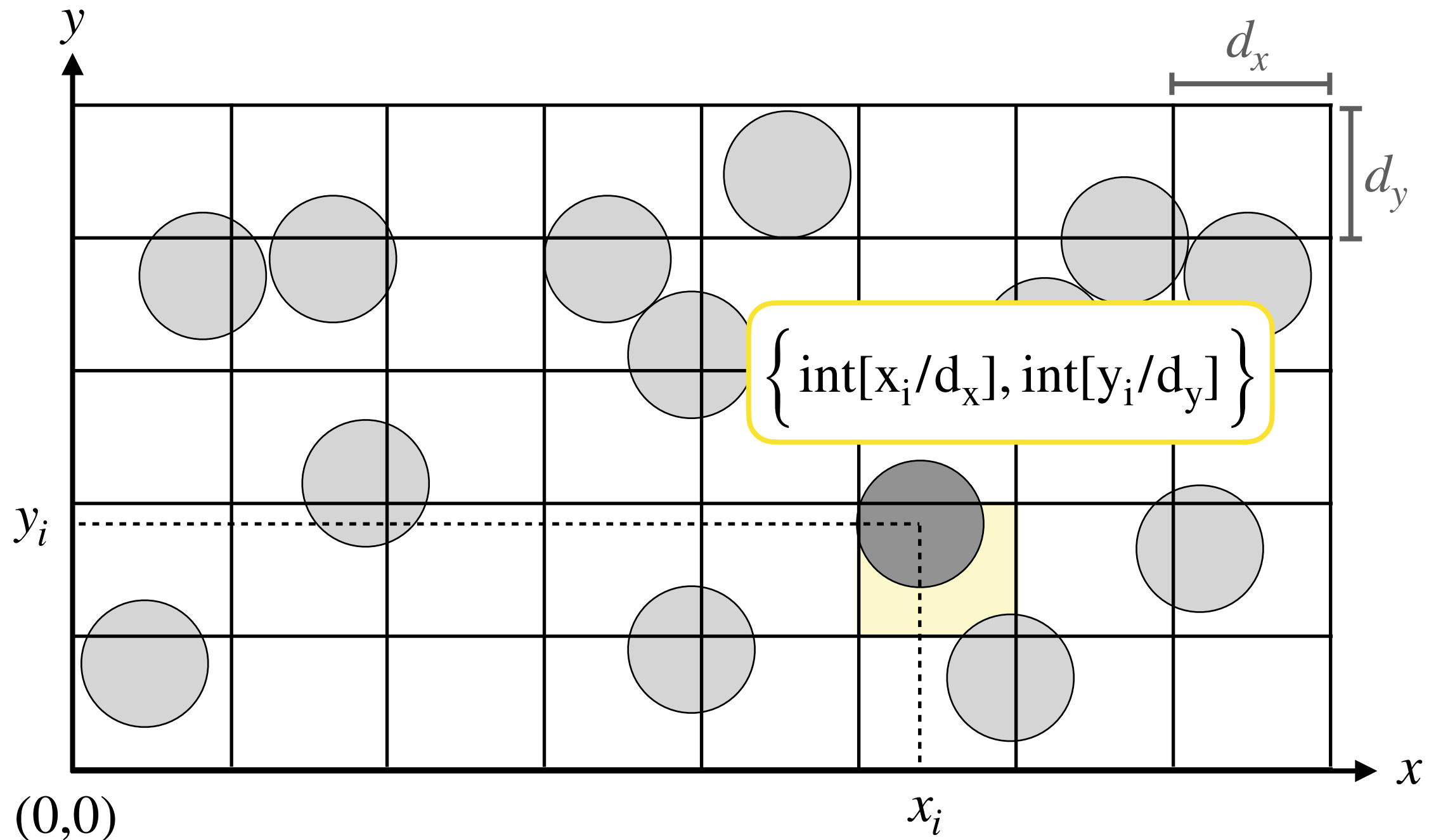
# Connaitre sa cellule

- On peut alors définir un **système d'axe** prenant son origine au coin inférieur gauche de la cellule (0,0)



# Connaitre sa cellule

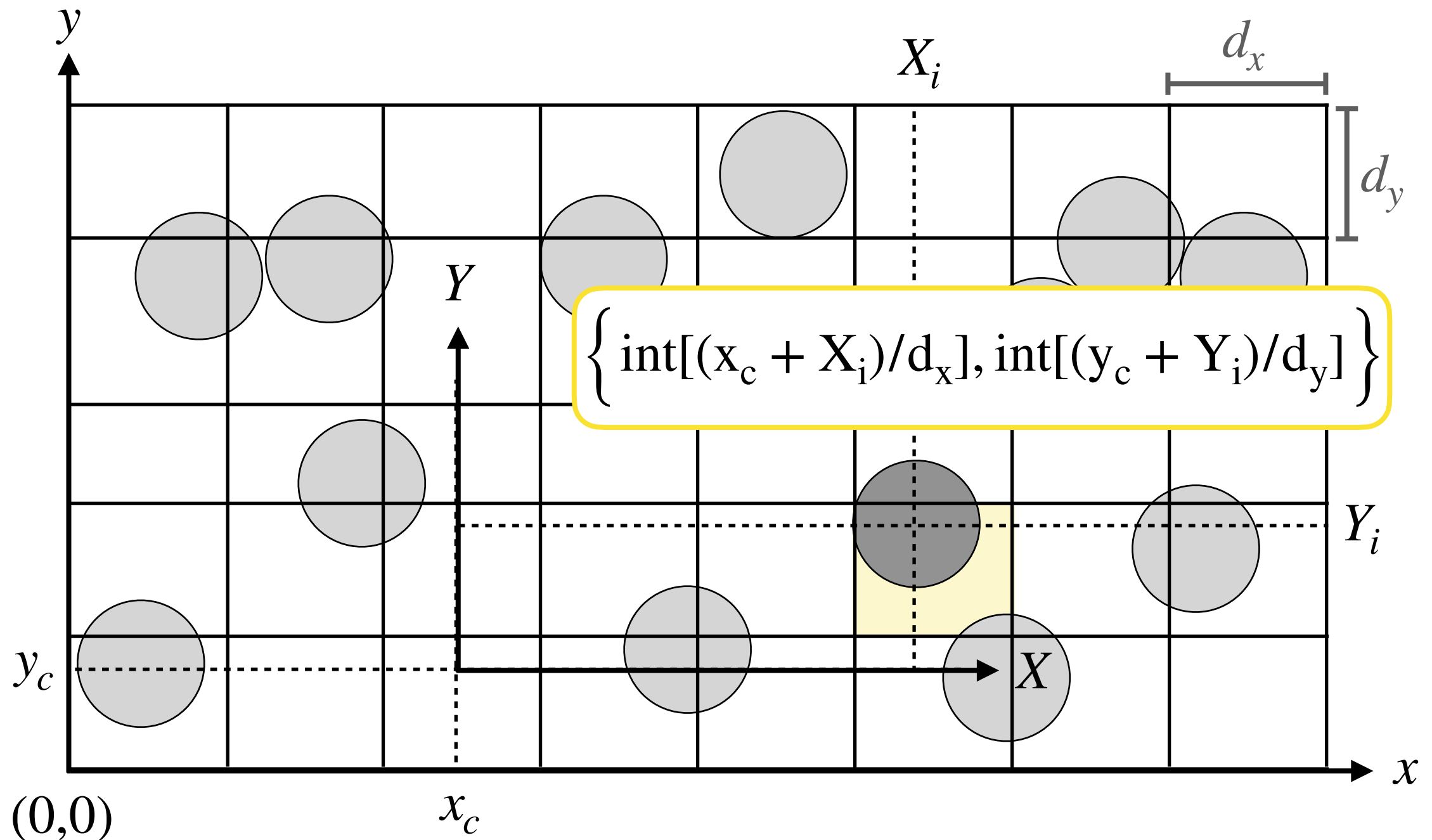
- On peut déterminer la **cellule** d'une particule à partir de sa position dans ce système d'axe.





# Connaitre sa cellule

- Si le système d'axe à son origine ailleurs, il faudra passer par un **changement** de base.



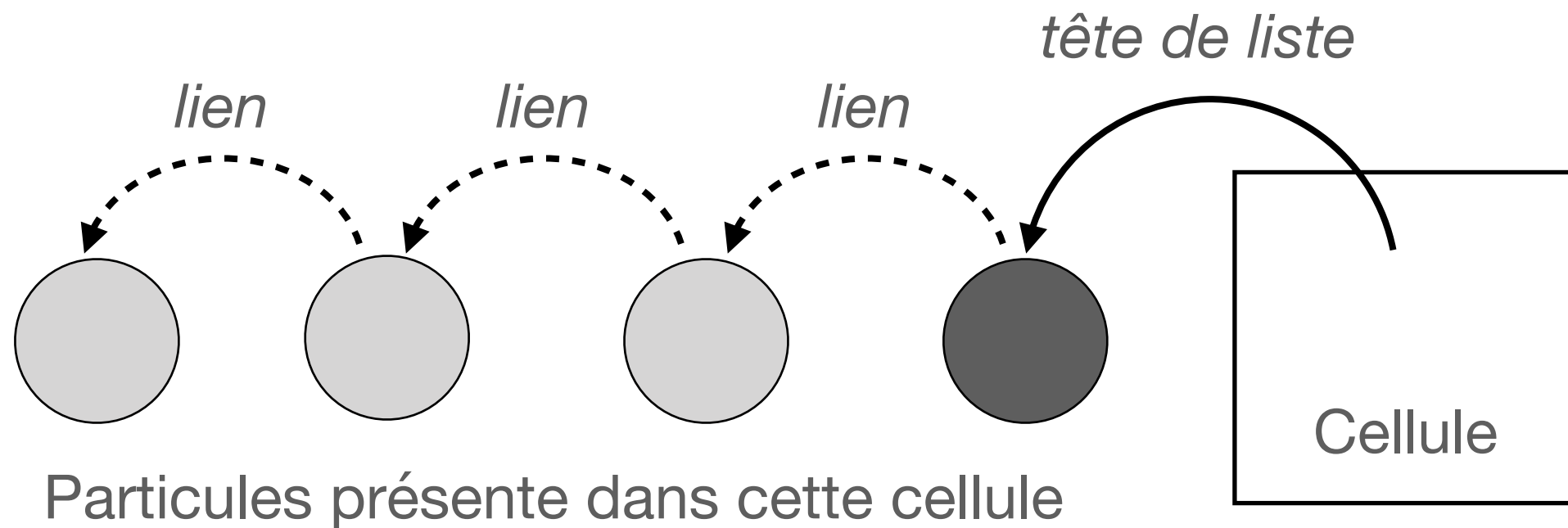
# Connaitre ses voisines

- ▶ Trouver les couples voisins est **trivial**. Attention pour les cellules le long des bords.

$\{0,2\}$				$\{i,j\}$			
$\{0,1\}$	$\{1,1\}$						
$\{0,0\}$	$\{1,0\}$	$\{2,0\}$					

# Connaitre ses particules

- ▶ Chaque cellule pourrait connaitre un **tableau** avec les indices des particules qu'elle contient.
- ▶ Pour être plus efficace, nous allons utiliser la méthode des **listes liées**.



- ▶ Chaque cellule connait une particule **TDL**. Celle-ci connait une particule qui en connait une autre...

# Connaitre ses particules

- ▶ Cette **structure** s'obtient facilement lors du classement des particules dans leurs cellules respectives.
- ▶ Dès qu'une particule est détectée dans une cellule, elle en devient la **TDL**.
- ▶ Cette nouvelle TDL **retient** qui était **TDL avant** elle (sauf si elle est la première)
- ▶ Ainsi une cellule ne doit **que retenir** qui est sa **TDL** au lieu de connaitre toutes les particules.
- ▶ Comme on applique la méthode des listes liées à des cellules, on parle de **cellules liées**.

# Cellules liées

- ▶ Dans notre algorithme, la **détection des contacts** fonctionne dès lors comme suit

On détermine la cellule contenant la particule  
On appelle la tête de liste de cette cellule  
Par récurrence, on teste les contacts dans la cellule  
On demande à la cellule qui sont ses voisines  
On répète la procédure pour chaque voisine

- ▶ Cette technique est d'autant plus efficace que le nombre de particules dans le système est **élevé**.

# Remarques

- ▶ Les cellules peuvent être indexée par un **numéro**  $k$  plutôt qu'un couple  $\{i, j\}$ :  $k = j \times m + i$ .
- ▶ La taille **optimale** des cellules correspond au diamètre  $d$  de la plus grande particule:  $d_x = d_y = d$ .
- ▶ Les **parois** du systèmes peuvent être classées dans plusieurs cellules à la fois.
- ▶ **Maintenant que les contacts sont détectés, il faut encore les gérés.**
- ▶ Nous allons voir dans la suite comment **modéliser** la force de répulsion lors d'un contact.

# Exercice

- ▶ Classez quelques particules placés aléatoirement dans un système carré divisé en neufs cellules liées.