


## Chapitre 2

# Algorithmes locaux pour l'électrostatique

 LA THÉORIE DE MAXWELL de l'électromagnétisme décrit le comportement des champs électrique et magnétique par des équations différentielles. Celles-ci sont purement *locales*. Cependant toutes les simulations dont nous avons parlé à présent utilisent la connaissance totale du champ électrostatique, se basant sur le fait que la propagation des interactions est instantanée. On doit en outre, pour prendre en compte des variations de permittivité, augmenter la complexité du calcul, ou faire des approximations pas toujours bien maîtrisées. Pour éviter les problèmes d'instabilités numériques, peut-on imaginer un algorithme de Monte-Carlo?

Les équations de Maxwell décrivant l'électrostatique sont

$$\mathbf{rot} \mathbf{E} = 0 \quad (\text{II-2.1a})$$

$$\mathbf{div} \mathbf{E} = \frac{\rho}{\varepsilon_0} \quad (\text{II-2.1b})$$

L'équation (II-2.1b) est appelée **loi de Gauss**. Ces équations étant locales, il est naturel de chercher à utiliser à profit ce caractère pour effectuer des calculs numériques. L'intérêt du caractère local réside dans le fait qu'il n'est nécessaire de ne connaître qu'un faible nombre de paramètres en chaque point du système. On calcule ensuite l'énergie comme une somme d'énergies locales faciles à évaluer ce qui conduit à un algorithme de complexité  $\mathcal{O}(N)$  (voir la section 1.2).

Dans ce chapitre, nous montrons comment une résolution locale numérique des équations de Coulomb doit être abordée. Nous revenons à la résolution des équations de Maxwell et montrons que la solution générale doit comprendre un terme de modes transverses. La section suivante est consacrée à l'illustration par un exemple simple des principes exposés dans la section 2.2, dont nous détaillons la construction et discutons les résultats.

## 2.1 Méthode naïve

Le chapitre précédent a présenté des méthodes de résolution consistant à calculer les forces électrostatiques grâce à l'équation de Poisson, par diverses méthodes (résolution matricielle, transformée de Fourier) et à en déduire le mouvement de chaque particule. Ces méthodes sont efficaces. Mais peuvent-elles se généraliser à un milieu où la permittivité  $\varepsilon$  n'est pas uniforme? On doit résoudre l'équation de Poisson modifiée

$$-\operatorname{div}(\varepsilon \mathbf{grad} \phi) = \rho(\mathbf{r}) \quad (\text{II-2.2})$$

à chaque étape.

Une simulation se basant sur le lagrangien écrit à partir du potentiel  $\phi$

$$\mathcal{L} = \frac{1}{2} \left[ \frac{1}{\mu_0} \left( \frac{\partial \phi}{\partial t} \right)^2 - \varepsilon(\mathbf{r})(\mathbf{grad} \phi)^2 - \rho(\mathbf{r})\phi(\mathbf{r}) \right] \quad (\text{II-2.3})$$

a été envisagée (Allen et coll., 2001). Pour effectuer le calcul numériquement, on doit connaître  $\phi$  en tout point ; il faut se placer sur un réseau, comme dans une méthode de trame. On attribue à chaque maille les valeurs de  $\phi$  et  $\varepsilon$  en ce point et utilise des opérateurs discrets pour résoudre numériquement l'équation (II-2.2). Cette méthode est lente puisqu'elle revient à inverser une matrice de grande taille. Elle suppose connue en chaque point la totalité du système, c'est-à-dire que les interactions se propagent à une vitesse infinie.

Ce schéma d'algorithme est correct au premier abord. Cependant lors de la simulation Monte-Carlo, le terme  $(\partial\phi/\partial t)^2$  tend vers 0, ce qui conduit à une valeur finale de l'énergie calculée par l'algorithme opposée à l'énergie minimale du système\*. On ne peut donc pas utiliser le lagrangien (II-2.3) pour effectuer une simulation numérique locale.

---

\*On a ainsi un système dont les particules de même signe s'attirent selon une force en  $1/r^2$ . En ne mettant que des charges de même signe, on a donc un système soumis à la gravitation, les charges jouant le rôle des masses et  $\phi$  celui du potentiel gravitationnel.

## 2.2 Introduction d'un champ transverse

Pour simuler localement un système en interaction coulombienne, l'approche la plus naturelle échoue. La solution consiste à revenir à l'équation qui contraint le système (loi de Gauss) et à exprimer l'ensemble de ses solutions.

### 2.2.a Solution générale de l'équation de Gauss

L'équation de Gauss

$$\operatorname{div}(\varepsilon \mathbf{E}) = \rho \quad (\text{II-2.1b})$$

ne possède pas une solution unique comme sous-entendu dans l'écriture de l'équation (II-2.2), mais sa solution générale s'écrit

$$\mathbf{D} = -\varepsilon(\mathbf{r}) \operatorname{grad} \phi + \operatorname{rot} \mathbf{Q} \quad (\text{II-2.4})$$

où on a posé  $\mathbf{D} = \varepsilon \mathbf{E}$  et  $\mathbf{Q}$  est un champ non déterminé mais *a priori* non nul. Dans l'espace de Fourier cette équation s'écrit

$$\tilde{\mathbf{D}}(\mathbf{k}) = -i\mathbf{k} \tilde{\varepsilon}(\mathbf{k}) * \tilde{\phi}(\mathbf{k}) + i\mathbf{k} \times \tilde{\mathbf{Q}}(\mathbf{k}) \quad (\text{II-2.5})$$

ce qui nous indique que les photons décrits par le champ  $\mathbf{Q}$  sont transverses. L'écriture de la solution (II-2.4) est une décomposition du champ  $\mathbf{D}$  en une partie longitudinale et une partie transverse dans l'espace de Fourier.

### 2.2.b Énergie électrostatique

L'énergie électrostatique du système est

$$\int \frac{\mathbf{D}^2}{2\varepsilon(\mathbf{r})} d\mathbf{r} = \frac{1}{2} \iiint \left[ \varepsilon(\mathbf{r}) (\operatorname{grad} \phi)^2 + \frac{(\operatorname{rot} \mathbf{Q})^2}{\varepsilon(\mathbf{r})} \right] d\mathbf{r} \quad (\text{II-2.6})$$

car le terme  $\int \varepsilon \operatorname{grad} \phi \cdot \operatorname{rot} \mathbf{Q}$ , en tant que produit scalaire du terme direct et du terme transverse\*, est nul. Cette écriture de l'énergie exprime d'une autre manière la nature de la décomposition du champ  $\mathbf{D}$  dans une base (équation (II-2.4)).

#### Formulation variationnelle

Si l'on exprime le problème sous forme variationnelle, l'action est une fonctionnelle de  $\mathbf{D}$  contrainte par l'équation de Gauss (II-2.1b), ce que l'on impose grâce au multiplicateur de Lagrange  $\lambda(\mathbf{r})$

$$S[\mathbf{D}] = \int \left[ \frac{\mathbf{D}^2}{2\varepsilon(\mathbf{r})} - \lambda(\mathbf{r}) (\operatorname{div} \mathbf{D} - \rho(\mathbf{r})) \right] d\mathbf{r}. \quad (\text{II-2.7})$$

---

\* Ou plus techniquement par intégration par parties.

Les états stationnaires de l'action sont déterminés par

$$\frac{\delta S}{\delta \mathbf{D}} = \frac{\mathbf{D}}{\varepsilon} + \mathbf{grad} \lambda = 0 \quad (\text{II-2.8})$$

dont on déduit que  $\mathbf{D} = -\varepsilon \mathbf{grad} \lambda$ . Ceci explique que la solution de l'équation de Gauss avec  $\mathbf{Q} = 0$  est généralement retenue, car c'est celle qui sera vérifiée lorsque le système est à l'équilibre.

### Mécanique statistique

La fonction de partition du système s'écrit d'après ce qui précède

$$\mathcal{Z} = \int [\mathcal{D}\mathbf{D}(\mathbf{r})] \exp\left(-\beta \int \frac{\mathbf{D}^2}{2\varepsilon} d\mathbf{r}\right) \times \prod_{\mathbf{r}} \delta(\text{div } \mathbf{D} - \rho(\mathbf{r})). \quad (\text{II-2.9})$$

En faisant le changement linéaire de variable fonctionnelle  $\Psi = \mathbf{D} - \varepsilon \mathbf{grad} \phi$ , elle se réécrit

$$\mathcal{Z} = \exp\left(-\beta \int \varepsilon \frac{(\mathbf{grad} \phi)^2}{2} d\mathbf{r}\right) \times \int [\mathcal{D}\Psi(\mathbf{r})] \exp\left(-\beta \int \frac{\Psi^2}{2\varepsilon(\mathbf{r})} d\mathbf{r}\right) \prod_{\mathbf{r}} \delta(\text{div } \Psi) \quad (\text{II-2.10})$$

soit encore

$$\mathcal{Z} = \underbrace{\exp\left(-\beta \int \varepsilon(\mathbf{r}) \frac{(\mathbf{grad} \phi)^2}{2} d\mathbf{r}\right)}_{\mathcal{Z}_{\text{Coulomb}}[\mathbf{r}]} \times \underbrace{\int [\mathcal{D}\mathbf{Q}(\mathbf{r})] \exp\left(-\beta \int \frac{(\mathbf{rot} \mathbf{Q})^2}{2\varepsilon(\mathbf{r})} d\mathbf{r}\right)}_{\mathcal{Z}_{\text{fluct}}[\mathbf{r}]}. \quad (\text{II-2.11})$$

Lorsque le milieu est homogène,  $\mathcal{Z}_{\text{fluct}}[\mathbf{r}]$  est une constante et peut être ignorée. Les modes transverses n'interviennent dans la statistique du système que lorsque  $\varepsilon(\mathbf{r})$  est inhomogène. C'est le terme  $\mathcal{Z}_{\text{Coulomb}}[\mathbf{r}]$  qui contient toute l'information sur les positions des charges.

### Représentation des degrés transverses

Les degrés transverses du champ  $\mathbf{D}$  sont contenus dans le terme  $\mathbf{rot} \mathbf{Q}$ . D'après l'écriture dans l'espace de Fourier de  $\tilde{\mathbf{D}}(\mathbf{k})$ , ils sont orthogonaux à  $\mathbf{k}$ . Pour les extraire de  $\mathbf{D}$ , on calcule  $\mathbf{k} \times \tilde{\mathbf{D}} = (\mathbf{k} \cdot \tilde{\mathbf{Q}})\mathbf{k} - \mathbf{k}^2 \tilde{\mathbf{Q}}$ . Or la jauge de Coulomb donne  $\text{div } \mathbf{Q} = 0$ . Les degrés de liberté de  $\mathbf{k} \times \tilde{\mathbf{D}}$ , ou encore ceux de  $\mathbf{rot} \mathbf{D}$  dans l'espace direct, sont les degrés de liberté de  $\mathbf{Q}$ . Pour les faire évoluer dans un algorithme de type Monte-Carlo il suffit de faire évoluer  $\mathbf{rot} \mathbf{D}$  : il n'y a pas besoin de simuler  $\mathbf{Q}$ .

## 2.3 Exemple de réalisation sur réseau cubique

Dans cette section, nous appliquons les résultats précédents à la réalisation d'un algorithme de simulation locale de l'électrostatique dans lequel on alterne entre mouvement de particules et du champ (au lieu du potentiel) dans un algorithme de Monte-Carlo Metropolis. Il s'agit d'un espace de permittivité  $\varepsilon(\mathbf{r})$  quelconque mais invariante dans le temps et indépendante des positions des particules. La méthode se généralise pour lever ces restrictions. Nous nous restreignons à un système élémentaire uniquement pour ne pas surcharger la description de l'algorithme.

### 2.3.a Discrétisation sur réseau cubique

On se place sur un réseau cubique de maille  $a$  et par souci de clarté nous considérerons que les particules sont astreintes à se déplacer uniquement sur les mailles. Là encore cette restriction n'est faite que dans un souci de simplicité.

#### Variable locales

L'algorithme que nous voulons réaliser doit avoir comme variables les quantités suivantes

- 1 – la position et la charge des particules ;
- 2 – la permittivité locale ;
- 3 – le champ  $\mathbf{D}$ .

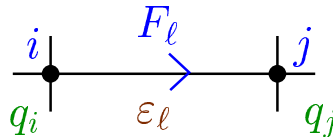


fig. 2.1: Variables locales de l'algorithme.  $\ell = ij$  et on a  $ji = -\ell$ , avec  $F_{-\ell} = -F_\ell$  et  $\varepsilon_{-\ell} = \varepsilon_\ell$ .

Plusieurs réalisations de ces requêtes sont possibles, nous choisissons pour notre exemple de présenter un algorithme où certaines variables sont associées à *un point* et les autres à *un lien* du réseau. Chaque site  $i$  du réseau possède une variable  $q_i$  de charge, et peut éventuellement posséder un indicateur d'occupation permettant de détecter la présence d'une particule. Ceci est utile lorsque l'on veut inclure des particules neutres. Chaque lien  $\ell = ij$  entre les sites  $i$  et  $j$  du réseau possède une permittivité  $\varepsilon_\ell^*$  et un *flux* du champ  $\mathbf{D}$  (voir la figure 2.1). Cette description en terme

---

\* La permittivité est donc dans ce modèle un tenseur diagonal dont les axes principaux sont les directions du réseau.

de flux se fait sans perte de généralité car seul le produit scalaire  $\mathbf{D}_\ell \cdot \boldsymbol{\ell}$  intervient. Le produit scalaire  $\mathbf{D} \cdot d\mathbf{S}$  s'écrivant pour un lien

$$F_\ell \stackrel{\text{déf.}}{=} a^2 \mathbf{D}_\ell \cdot \frac{\boldsymbol{\ell}}{a}. \quad (\text{II-2.12})$$

Notons que le flux est *orienté*, ainsi si on note  $-\ell$  le lien  $ji$  lorsque  $\ell$  représente  $ij$  on a  $F_{-\ell} = -F_\ell$ .

### Contrainte : la loi de Gauss

Nous avons basé toute notre discussion sur la loi de Gauss (II-2.1b) qui est la contrainte imposée au système. L'algorithme que nous proposons fait évoluer l'ensemble des variables du système en la respectant : les mouvements locaux du système se font de telle sorte que si la contrainte est respectée avant le mouvement elle le sera après. En partant d'une configuration correcte, la contrainte sera respectée pendant toute la simulation. Avec les variables que nous avons introduites (voir la figure 2.2), elle s'écrit

$$\forall i, \sum_{\ell=ij} F_{ij} = q_i. \quad (\text{II-2.13})$$

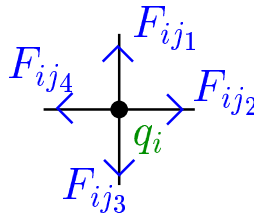


fig. 2.2: Loi de Gauss pour notre exemple d'algorithme.

### Fonction d'énergie

L'énergie du système (II-2.6) s'exprime dans notre système de variables

$$E = \sum_{\ell} \frac{F_\ell^2}{2a \varepsilon_\ell}. \quad (\text{II-2.14})$$

On la calcule au début de la simulation. Ensuite l'évolution étant locale, il suffit de calculer la variation de  $E$  induite par les modifications des flux  $F_\ell$ .



fig. 2.3: Mouvement d'une particule. La contrainte de Gauss est respectée si  $F'_\ell = F_\ell - q$ .

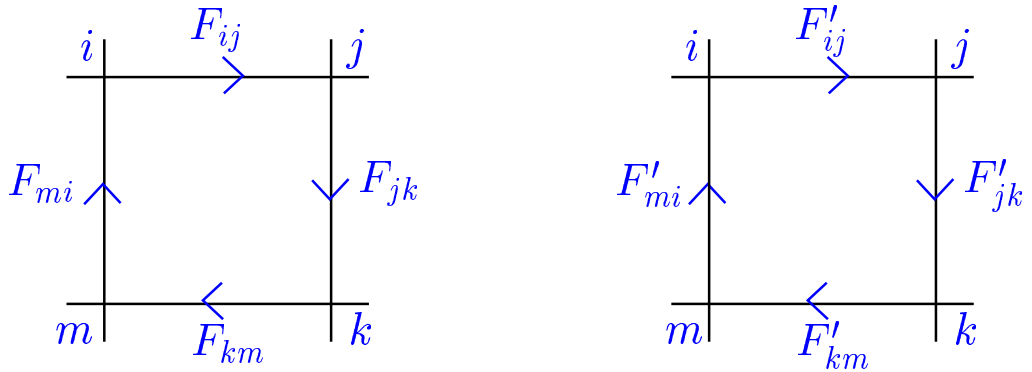


fig. 2.4: Mouvement de fluctuation du champ. Les quatre flux de la plaquette  $ijkm$  sont modifiés du même incrément aléatoire  $\delta F$  dans un sens de la marche donné. Sur ce schéma  $F'_{ij} = F_{ij} + \delta F$ ,  $F'_{jk} = F_{jk} + \delta F$ ,  $F'_{km} = F_{km} + \delta F$  et  $F'_{mi} = F_{mi} + \delta F$ .

### 2.3.b Mouvements de Monte–Carlo

#### Mouvement d'une particule

Le mouvement d'une particule se fait d'un site à un site voisin. Lors du mouvement, la contrainte de Gauss doit être conservée. Cela se traduit simplement par le fait qu'il faut changer le flux  $F_\ell$  entre le site de départ et le site d'arrivée par la valeur de la charge déplacée (figure 2.3).

$$\boxed{F'_\ell = F_\ell - q}. \quad (\text{II-2.15})$$

La variation d'énergie servant dans la simulation de Monte–Carlo Metropolis est

$$\Delta E = \frac{F'^2_\ell - F^2_\ell}{2a \varepsilon_\ell}. \quad (\text{II-2.16})$$

#### Fluctuations du champ

Les mouvements des particules sont responsables des variations du champ direct (terme  $\varepsilon \mathbf{grad} \phi$ ). Les modes transverses, on l'a vu dans la section 2.2.b, sont

représentés par  $\mathbf{rot D}$ . Pour les faire évoluer sur le réseau on doit modifier

$$\iint \mathbf{rot D} \, d\mathbf{S} = \oint \mathbf{D} \cdot d\boldsymbol{\ell}. \quad (\text{II-2.17})$$

Il faut que la circulation de  $\mathbf{D}$  le long de tous les chemins fermés fluctue. Or dans un réseau cubique, la circulation d'un champ de vecteurs le long d'un chemin formé de liens est la somme des circulations de ce champ sur les contours formés des quatre liens de plaquettes du réseau (voir la figure 2.4). Chaque circulation sur le tour d'une plaquette est un degré de liberté du système. On effectue le mouvement de Monte–Carlo—Metropolis lié à ce degré de liberté en ajoutant  $\delta\Gamma$  à la circulation.

Notons  $i, j, k$  et  $m$  les quatre sites d'une plaquette  $P$ . La circulation le long du chemin (orienté)  $ijkm$  est

$$\Gamma_{ijkm} = a^{-1}(F_{ij} + F_{jk} + F_{km} + F_{mi}). \quad (\text{II-2.18})$$

Si on modifie  $F_{ij}$  de  $\delta F$ , alors pour respecter la loi de Gauss, il faut modifier  $F_{mi}$  et  $F_{jk}$  de  $\delta F$  aussi (on s'interdit de modifier d'autre flux que ceux de  $P$ ). Ces deux modifications nous amènent aussi à modifier  $F_{km}$  de  $\delta F$ . On retrouve le fait qu'un seul degré de liberté est associé à chaque plaquette. On pose

$$\boxed{\delta F = \frac{a}{4}\delta\Gamma}; \quad (\text{II-2.19})$$

le mouvement de Monte–Carlo consiste à modifier  $F_{ij}$ ,  $F_{jk}$ ,  $F_{km}$  et  $F_{mi}$  de  $\delta F$ . La variation d'énergie du système utilisée par Monte–Carlo Metropolis est

$$\Delta E = \frac{1}{2a} \sum_{\ell \in P} \frac{F'_\ell{}^2 - F_\ell^2}{\varepsilon_\ell}. \quad (\text{II-2.20})$$

Un degré de liberté supplémentaire doit être pris en compte lorsque le système est à conditions aux limites périodiques (l'espace est alors le tore à trois dimensions), voir dans l'article II–A.

### 2.3.c Résultats

#### Vérification de la convergence

Dans un premier temps on vérifie que le système reproduit le bon ensemble statistique. On utilise pour cela une permittivité scalaire et uniforme, et on place 36 particules (18 de charges positives et 18 de charges négatives de même valeur) sur un réseau cubique  $6 \times 6 \times 6$ . On calcule le facteur de structure et on le compare au facteur de structure du même système qui évolue avec un algorithme Monte–Carlo utilisant une résolution matricielle de l'équation de Poisson. Les résultats sont proches. On en conclut que l'algorithme converge vers une statistique correcte alors que la dynamique est fictive.



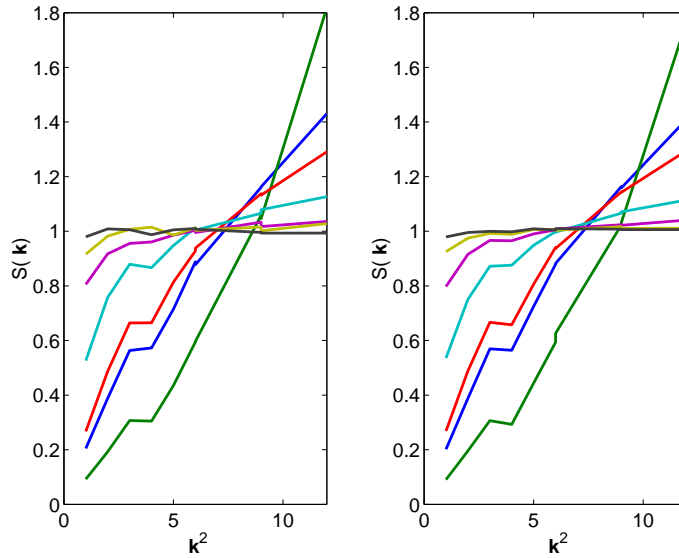


fig. 2.5: Comparaison des facteurs de structure d'un même système évoluant avec l'algorithme décrit dans la section précédente (à gauche) et avec un algorithme de résolution de l'équation de Poisson (à droite). Les courbes sont obtenues à différentes températures.

### Étude de l'équilibration du système

D'après sa construction, l'algorithme présenté est bien de complexité  $\mathcal{O}(N)$ . Il nous reste à nous assurer qu'il relaxe rapidement vers l'état stationnaire, autrement dit que la faible complexité ne se traduit pas par un préfacteur énorme. Le temps de convergence est évalué avec le temps caractéristique de décroissance des fonctions de corrélation densité–densité et charge–charge de systèmes de différentes tailles pour un mode lent, ici le mode  $(\frac{2\pi}{L}, 0, 0)$ . Les résultats sont présentés sur la figure 2.6. On constate que les fonctions de corrélation charge–charge convergent toutes à la même vitesse, conséquence de l'écrantage, alors que les fonctions de corrélation densité–densité décroissent d'autant plus lentement que le système est grand. Le temps d'équilibration de l'algorithme pour ce modèle est proportionnel à  $L^2$ , ce qui caractérise une évolution diffusive.

## 2.4 Discussion

Nous avons décrit une méthode générale de simulation des interactions coulombiennes et présenté un algorithme dans un cadre simple. L'intérêt de cette approche

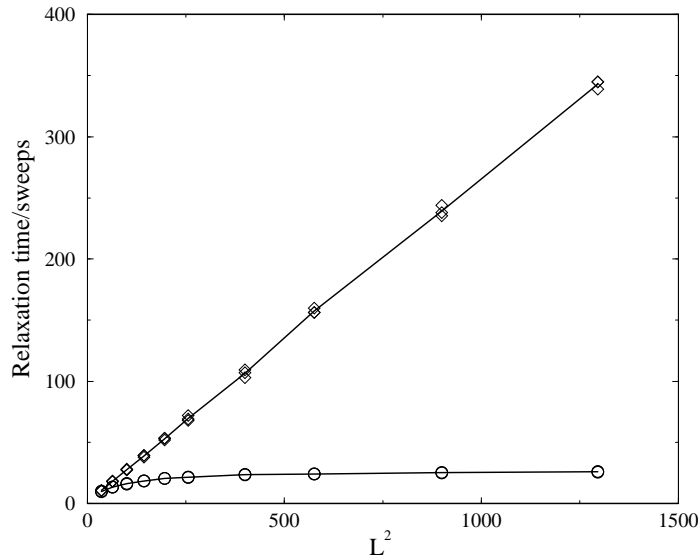


fig. 2.6: Temps d'équilibration des fonctions de corrélation densité–densité ( $\diamond$ ) et charge–charge ( $\circ$ ) en fonction de  $L^2$  du système. Le réseau est cubique de taille  $L \times L \times L$ , avec conditions aux limites périodiques et contient  $L^3/4$  particules.  $L$  varie de 6 à 36.

des simulations numériques électrostatiques est multiple.

- I** — Le système prend en compte la permittivité au niveau local sans ajouter d'ingrédient supplémentaire. La fonction de partition  $\mathcal{Z}$  contient *de facto* tous les effets polaires permanents (par exemple les effets d'image, les interactions entre dipôles permanents de type Keesom contenu dans  $\mathcal{Z}_{\text{fluct}}$ , etc.)
- II** — Comme l'algorithme est local, il est facilement transposable à des machines de calcul parallèle.
- III** — L'algorithme est rapide, la complexité de chaque étape de Monte–Carlo est d'ordre  $\mathcal{O}(N)$ , ce qui devrait permettre de simuler en des temps accessibles des systèmes de grande taille.
- IV** — La section 2.3 illustre la simplicité technique de la programmation, à mettre en regard avec les algorithmes plus complexes présentés dans le chapitre 1.

L'algorithme que nous avons décrit est très élémentaire, mais il s'adapte facilement à des systèmes où la permittivité est transportée par les particules ou autorisée à fluctuer (mouvement de Monte–Carlo d'un autre type que ceux présentés), et peut même être anisotrope (mais toujours de directions principales celles de la maille). Les interactions de volume exclu peuvent être facilement ajoutées.