Projet 3, Informatique Scientifique, 2004

Simulation 2D d'un condensat de Bose-Einstein

1 Introduction

Le refroidissement d'un gaz à des températures proches de 0 Kelvins (zéro absolu) induit un état particulier de la matière, caractérisé par l'existence d'un même état quantique pour tous les atomes. Cet état de la matière a été prédit en 1925 par Einstein et réalisé expérimentalement par une équipe américaine en 1995, récompensée d'ailleurs par le prix Nobel de physique. Dans un condensat de Bose-Einstein, le comportement identique des atomes est similaire à celui des photons dans un rayon laser, ce qui laisse présager dans l'avenir la possibilité de réaliser des lasers à atomes, avec de nombreuses applications pratiques imaginées.

Nous allons essayer de simuler numériquement l'apparition de tourbillons quantiques dans un condensat de Bose-Einstein en rotation. Cette expérience à été réalisé au Laboratoire Kastler-Brossel de l'ENS Paris.

Un état stable du condensat est obtenu en intégrant en temps l'équation instationnaire suivante :

$$\begin{cases} \dfrac{\partial u}{\partial t} - \Delta u - i(2\Omega)(\mathbf{A}.\nabla u) - G(x,y,u,\mu_\varepsilon(u))u = 0, \quad \text{sur} \quad \mathcal{D} \\ u = 0, \quad \text{sur} \quad \Gamma = \partial\mathcal{D}. \end{cases} \tag{1}$$

avec $G(x,y,u,\mu) = \frac{1}{\varepsilon^2}\left(\rho_{TF}(x,y) - |u|^2\right) - \mu$ et avec avec la contrainte $||u|| = \displaystyle\int_{\mathcal{D}} |u|^2 = 1$. Dans l'équation (1)

– la variable $u \in \mathbb{C}$ est complexe et on note $i = \sqrt{-1}$ l'unité imaginaire et

$$u = Re(u) + iIm(u), \quad \bar{u} = Re(u) - iIm(u), \quad |u|^2 = u\bar{u}.$$

– les opérateurs intervenant sont définis comme :

$$\mathbf{A} = \begin{pmatrix} y \\ -x \end{pmatrix}, \quad \nabla = \begin{pmatrix} \dfrac{\partial}{\partial x} \\ \dfrac{\partial}{\partial y} \end{pmatrix}, \quad \Delta = \frac{\partial}{\partial x^2} + \frac{\partial}{\partial y^2} \tag{2}$$

– le multiplicateur de Lagrange a l'expression :

$$\mu_\varepsilon(u) = \int_{\mathcal{D}} \left[|\nabla u|^2 - \frac{1}{\varepsilon^2}\rho_{TF}(x,y)|u|^2 \right] - 2\Omega L_z, \tag{3}$$

où

$$L_z = -\int_{\mathcal{D}} Im\left[(\mathbf{A}.\nabla u)\bar{u}\right], \tag{4}$$

– le domaine $\mathcal{D}$ est une ellipse définie comme :

$$\mathcal{D} = \{\rho_{TF}(x,y) \geq 0\}, \quad \rho_{TF}(x,y) = \rho_0 - (x^2 + \beta y^2), \quad \rho_0 = \sqrt{\frac{2\beta}{\pi}}. \tag{5}$$

– les données du problème sont $\varepsilon, \beta, \Omega$.
L'équation (1) sera avancée en temps jusqu'à l'obtention d'un état stationnaire. La méthode de résolution suivante sera utilisée :

• la condition initiale ($t = 0$) sera calculée comme :

$$u^0 = \sqrt{\rho_{TF}(x,y)} \in \mathbb{R} \tag{6}$$

• pour l'avancement en temps (de $t_n = n\Delta t$ à $t_{n+1} = (n+1)\Delta t$) nous utiliserons un schéma d'Euler implicite ($u^\star = u^*, u^\bullet = u^{n+1}$) ou semi-implicite non symétrique ($u^\star = u^n, u^\bullet = u^{n+1}$) ou semi-implicite symétrique ($u^\star = u^n, u^\bullet = u^n$)

$$\frac{u^* - u^n}{\Delta t} - \Delta u^* - i(2\Omega)(\mathbf{A}.\nabla u^\bullet) - G(x,y,u^\star(x,y),\mu_\varepsilon(u^n))u^* = 0 \tag{7}$$

avec $G(x,y,u,\mu) = \frac{1}{\varepsilon^2}\left(\rho_{TF}(x,y) - |u^\star|^2\right) - \mu)$.

remarque dans le cas semi-implicite à chaque itération en temps, il suffit de résoudre un problème linéaire.

suivi d'une projection pour assurer la contrainte de la norme unité :

$$u^{n+1} = \frac{u^*}{||u^*||}, \quad ||u^*|| = \int_{\mathcal{D}} |u^*|^2. \tag{8}$$

• le calcul s'arrête quand $|E(u^{n+1}) - E(u^n)| < 10^{-5}$, avec l'énergie $E(u) \in \mathbb{R}$ calculée suivant :

$$E(u) = \int_{\mathcal{D}} e(x,y,|\nabla u|^2,|u|^2) \tag{9}$$

avec

$$e(x,y,\nu,\mu) = \frac{1}{2}\nu - \frac{1}{2\varepsilon^2}\rho_{TF}(x,y)\mu + \frac{\mu^2}{4\varepsilon^2} \tag{10}$$

où $\nu = |\nabla u|^2$, $\mu = |u|^2$.

Pour résoudre l'équation (7) non linéaire dans le cas implicite nous utiliserons la méthode de Newton avec :

$$F(u^*) = \frac{u^* - u^n}{\Delta t} - \Delta u^* - i(2\Omega)(\mathbf{A}.\nabla u^*) - \frac{1}{\varepsilon^2}\left(\rho_{TF}(x,y) - |u^*|^2\right)u^* - \mu_\varepsilon u^* \tag{11}$$

$$\left[\frac{\partial F}{\partial u^*}\right].q = \frac{q}{\Delta t} - \Delta q - i(2\Omega)(\mathbf{A}.\nabla q) - \frac{1}{\varepsilon^2}\left(\rho_{TF}(x,y) - 2|u^*|^2\right) q + \frac{1}{\varepsilon^2}\left(u^* \cdot u^*\right) \bar{q} - \mu_\varepsilon q \tag{12}$$

Par conséquent, la méthode de Newton s'écrit :

$$\begin{cases} \theta^{(0)} = u^n \\ \theta^{(k+1)} = \theta^{(k)} - q, \end{cases} \tag{13}$$

avec q solution de

$$\left[\frac{\partial F}{\partial u^*}\right]_{u^*=\theta^{(k)}}.q = F(u^*)|_{u^*=\theta^{(k)}}. \tag{14}$$

Questions

Les questions 1,2,3 sont a faire sur le problème instationnaire avec le schéma semi-implicite. Pour que l'algorithme converge vers la solution, il faut que le pas de temps soit suffisamment petit.

Question 1

Ecrire un interpreteur de formule arithmétique pour que l'utilisateur donne la formule de G et la formule de e et on utilise interpreteur génere les valeurs aux sommets de la triangulation. Par exemple si l'utilisateur pose $G(x,y,u,v) = 1.2 - (x*x+3*y*y) - u*u - v*u$ l'interpreteur devra répondre $-9.05 - 6i$ si $x = 1, y = 2, u = 0.5 + 2i, v = 2$. L'interpreteur devra se présenter comme une fonction C++ :

```
  typedef complex<double> Complex;                    //    cf. #include <complex>
  Complex interpret(const R2 p, Complex u,Complex v, const char* rho);
//    exemple d'appel
  Complex rhoTF = interpret( R2(1.,2.), Complex(0.5,2), Complex(2) ,"1.2-(x*x+3*y*y) + v*u");
  cout <<rhoTF<<endl;
```

Question 2

Ecrire la formulation variationnelle de chaque itération en temps ; construire les matrices virtuelles correspondantes. en utilisant la fonction *interpret* appliquée à G et à ses dérivées partielles.

Valider vos matrices dans un programme qui résout une EDP dont on connaît la solution analytiquement.

Tester la convergence en temps, vous utiliserez l'interpreteur pour définir la fonction e introduite en (10) afin d'évaluer le test d'arret pour le jeux de paramètres suivant

$$\varepsilon = 0.02, \quad \beta = 1., \quad \Omega = 20$$

construire un maillage du domaine $\mathcal{D}$ et résoudre le problème (1).
Visualiser les iso-lignes de $|u|$.

Remarque : afin d'optimiser votre programme, nous vos conseillons des stocker les valeurs de $G(x, y, u^\star(x, y), \mu)$ dans un tableau dimensioner au nombre de points d'intégration total, c'est à dire $3 \times nt$ pour la formule des milieux des arêtes.

Question 3

Ecrire une interface java pour votre programme qui automatise au maximum l'entrée des données. Il faudra rentrer quelques paramètres dans des champs éditables et aussi pouvoir changer les fonctions ρ et e

Les questions suivantes peuvent être faites dans un ordre quelconque.

Question 4

Paramètrer les frontières et le nombres de points de discrétisation dans des champs éditable java, générer un fichier .edp et appeler `FreeFem++` par un script pour générer le maillage et ensuite appeler votre module de résolution de l'EDP. Automatiser aussi la sortie graphique.

Question 5

Programmer le schéma implicite en temps

Ecrire la formulation variationnelle de chaque itération de la méthode de Newton (14) ; construire les matrices virtuelles correspondantes. en utilisant la fonction *interpret* appliquée à F et à ses dérivées partielles.

Valider vos matrices dans un programme qui résout une EDP dont on connaît la solution.

Construire l'algorithme de Newton et tester la convergence pour le jeux de paramètres suivant

$$\varepsilon = 0.02, \quad \beta = 1., \quad \Omega = 20$$

construire un maillage du domaine $\mathcal{D}$ et résoudre le problème (1).
Visualiser les iso-lignes de $|u|$ a convergence en temps.

Question 6

Algèbre de fonction et/ou différentiation automatique
Recrire la fonction `interpret` afin qu'elle calcule automatiquement les dérivées partielles à partir de la seule connaissance de la formule définissant la fonction.