

電子の相互作用のシミュレーション (修正)

01ca0125 鈴木 藍
2002年 5月 27日

目 次

概要	3
レポートの目的	3
1 レポートの意図	3
2 前回の反省	3
3 前回の修正箇所	3
3.1 前回の問題点	3
3.2 考えられる不具合	3
4 スカラー場を計算する関数について	4
5 ベクトル場を計算する関数について	5
6 その他の問題	6
7 実行結果について	6
まとめ・結論	6
感想	7
参考文献	8

概要

以前、レポートで完成出来なかった電子の相互作用のシミュレーションをするプログラムを完成させた。前回のプログラムの修正点をまとめた。

レポートの目的

完成させられなかった原因と、反省、これからの行動目標を明確にする。

1 レポートの意図

前回、電子の相互作用のシミュレーションを行うプログラムを作成したが未完成であったため、独自にプログラムを修正し 完成させた。

2 前回の反省

まず、電子が相互作用する場というモデルの把握がしっかり出来ていなかった事にある。短時間ではあったが、時間の使い方をもう少し整理するべきであると思う。また、プログラムに関しては何が出来るのか、どの計算が最適かを早い段階で理解しておく必要がある。もしわからない事が出てくればその分早く対応出来るからである。

3 前回の修正箇所

修正はほとんどせず、はじめから作りなおした。

3.1 前回の問題点

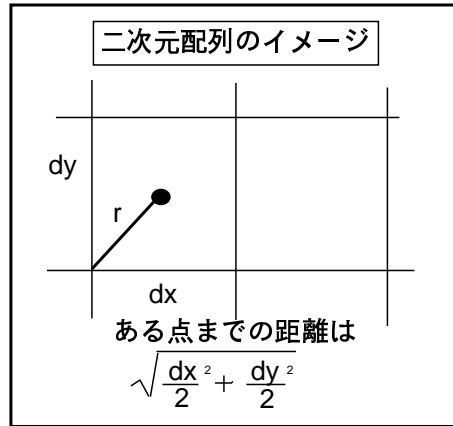
一番の問題は、プラス同士の電価を持つ電子が引合い、マイナスはプラスを追いかけて、プラスはマイナスから逃げるということである。シミュレーションのプログラムとしての機能を果たしていない。

3.2 考えられる不具合

まず計算方法として、作用される電子以外のスカラー場を計算し、そのスカラー場を元にベクトル場を作成し、その電子の移動を計算するという方法は適当であった。計算方法はあっているが、以前作成したベクトル場を計算する関数とスカラー場を計算する関数にバグがあった。今回はこのベクトル場とスカラー場を計算する関数をもう一度考え直してみた。

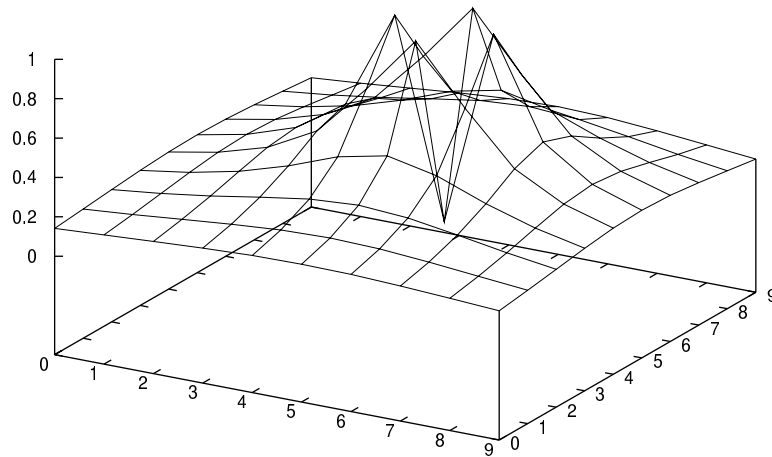
4 スカラー場を計算する関数について

前は、ある点までの半径を $\sqrt{(\frac{dx}{2})^2 + (\frac{dy}{2})^2}$ として求めていたので、求める電子のいる点のポテンシャルが無限に大きな値になるはずだが、そうはならず適当な値となり、グラフに収まっていた。今回は、その点の値を 0.0 とした。以下は 以前の計算方法のイメージである。



その点のポテンシャルを 0.0 にしたため、プロットされるグラフは以下の通りである。

"tt" ———



以下はスカラー場を求める関数のソースである。

```
ScaleField *makeScaleField(ScaleField *scalefield,
                           anElectron electrons)
{
    int i, j, k, l, inck, incl, thisX, thisY;

    thisX = (FIELD_SIZE / 2) + (int)(electrons.x);
    thisY = (FIELD_SIZE / 2) - (int)(electrons.y);

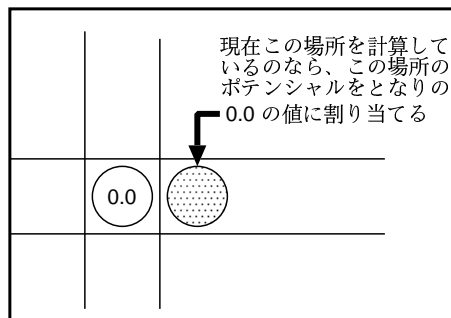
    for (i=0, k=thisY, inck=1; i<FIELD_SIZE; i++, k-=inck) {
        for (j=0, l=thisX, incl=1; j<FIELD_SIZE; j++, l-=incl) {
            if ((l==0) && (k==0))
                scalefield->potential[i][j] = 0.0;
            else
                scalefield->potential[i][j] += (electrons.q
                                                / (sqrt(SQUARE(l) + SQUARE(k))));

            if(l==0)
                incl *= -1;
        } /* for 'j' */
        if(k==0)
            inck *= -1;
    } /* for 'i' */

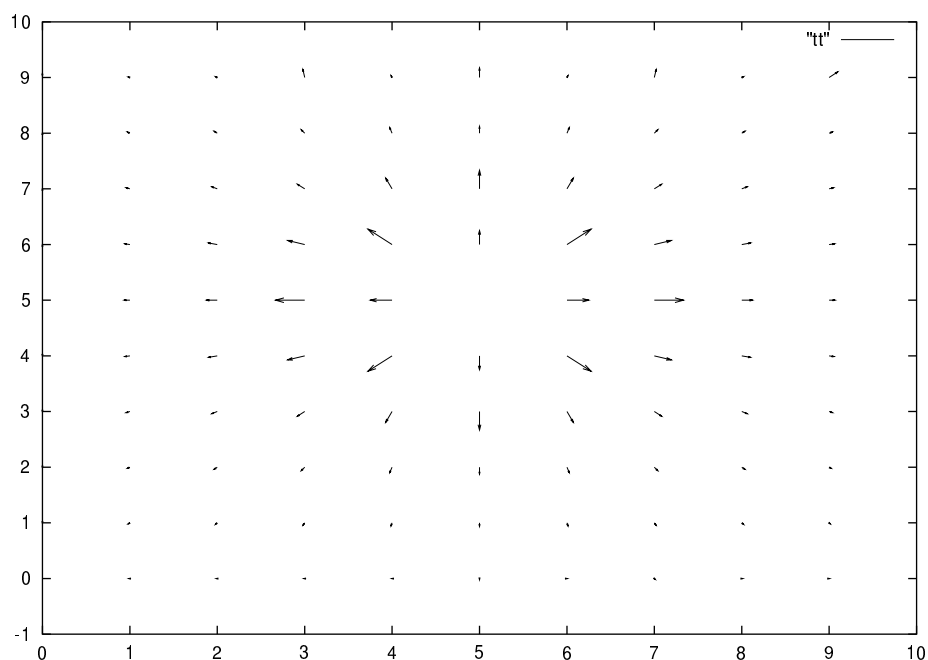
    return scalefield;
}
```

5 ベクトル場を計算する関数について

今回、一番出来の悪い関数がこれであった。境界処理が一番はじのベクトル場を計算しないという方法をとっていたが今回は隣り合うポテンシャルが無い場合、もう方のポテンシャルをない方に割り当て、計算をした。又、作用される電子のいる場所のポテンシャルは0.0になっている為、その場所には現在計算をしている場所のポテンシャルを割り当てることにした。



境界処理が長いので、ソースは割愛する。
以下はベクトル場を計算した結果である。



6 その他の問題

二次元配列を 4 分割し、正と負を表していた所で配列の添字を x, y から計算をしていたが、そこでバグを見付けた。

7 実行結果について

今回は画像を載せなかったが、以下の URL からソースごとダウンロード出来る。

<http://www.soulhack.net/TeX/electron2/>

実行方法:

- ・ 解凍して、remake へ移動
- ・ make する
- ・ `a.out {pipe} gnuplot` で実行

まとめ・結論

ようやく完成できたが、本当に時間がかかった。やはり、行動の順序や 計画を立てる事はとても重要である。

感想

正直完成できてうれしかった。不具合などは、よく見ると簡単なミスばかりだったので、設計方法の方も早めに身に着けておく必要がある。

参考文献

[1] なし