

# 2020 年度計算物理学 II 第 3 回レポート課題

## 数値計算プログラミングの基礎

2020/8/3 更新

### ■注意事項

- manaba で一つの pdf ファイルとして提出してください。
- (1) と (2)(と (3)) で作成した Fortran や C のプログラムと問の答え・グラフをまとめてください。
- レポートは TeX で作成するのが簡単かと思います。また、プログラムを TeX に貼り付ける時は verbatim 環境を使います。

```
¥begin{verbatim}
```

プログラムをコピー&ペースト

一行が長いプログラムは右側で切れてしまうので適宜改行するようにしてください。

```
¥end{verbatim}
```

- TeX を使いたくない人は Word などで作っても構いませんが一つの pdf ファイルにしてください。
- (3) は発展問題です。(1) と (2) が終わってまだ時間のある人は解いてください。ただし、(3) に取り掛かる前に (1) と (2) の部分のレポートの作成を先に行ってください。

(1) 統計力学で出てくる Stirling の公式は大きな整数  $n$  に対して

$$\log n! = n \log n - n + \frac{1}{2} \log(2\pi n) \quad (1)$$

で与えられます (右辺第 3 項は省略されることもあります)。以下の (a)-(c) を計算するプログラムを一つにまとめて作成してください。(gnuplot の作成に使ったコマンドファイルは提出不要 (8/3 更新))

- (a) 左辺と右辺の値を計算するプログラムを作成し、 $n = 10, 10^2, 10^3, 10^4, 10^5$  の場合の左辺と右辺の値を出力してください。
- (b)  $n = 1 \sim 100$  程度の範囲で左辺と右辺それぞれをグラフにプロットしてください。違いは見えますか。右辺第 3 項を抜くとどうですか。
- (c)  $n = 1 \sim 100$  程度の範囲で Stirling 公式での近似の誤差を対数目盛でプロットしてください。誤差は

$$\frac{|\log n! - n \log n + n - \frac{1}{2} \log(2\pi n)|}{\log n!} \quad (2)$$

で与えられます。

### ■ヒント

- 階乗は単精度の整数で表現するのは難しいので倍精度実数で表現しますが、倍精度実数にも表現できる最大値があります。この問題で必要な量は  $n!$  ではなく  $\log n!$  ですので、 $n!$  を計算してから対数を取るのではなく、対数を取ってから階乗を直接計算するようにします。 $(\log n! = \sum_{k=1}^n \log k)$
- Fortran では組み込み関数の  $\log(x)$  の引数  $x$  は実数です。整数から倍精度実数への型変換を `dbble()` を用いて行います。

- C 言語で倍精度実数の絶対値を取る関数は fabs です。
- gnuplot で y 軸を対数目盛にするには set logscale y

(2) データファイル mat.dat に  $10 \times 10$  次元の実行列が書かれています。これをプログラムで読み込んで、転置行列との積をとることで、この行列が直交行列であることを示してください。データファイル mat.dat は

```
$ wget https://wwwnucl.ph.tsukuba.ac.jp/~hinohara/compphys2-20/report/mat.dat
```

でダウンロードしてください。

### ■ヒント

- ファイルからの配列の読み込み・行列積の計算は演習問題 15 を参考にしてください。

### (3) (発展問題)

量子力学の角運動量の合成で現れる Clebsch-Gordan 係数は以下の公式で与えられます。(通常  $\langle j_1 m_1 j_2 m_2 | j_1 j_2; JM \rangle$  という記法が使われますがここでは CG と書きます)

$$\begin{aligned} \text{CG}(j_1, m_1, j_2, m_2, J, M) &= \delta_{M, m_1 + m_2} \sqrt{\frac{(2J+1)(j_1+j_2-J)!(j_1-j_2+J)!(-j_1+j_2+J)!}{(j_1+j_2+J+1)!}} \\ &\quad \times \sqrt{\frac{(j_1+m_1)!(j_1-m_1)!(j_2+m_2)!(j_2-m_2)!(J+M)!(J-M)!}{(-1)^z}} \\ &\quad \times \sum_z \frac{z!(j_1+j_2-J-z)!(j_1-m_1-z)!(j_2+m_2-z)!(J-j_2+m_1+z)!(J-j_1-m_2+z)!}{1} \end{aligned}$$

ただし整数  $z$  に関する和については  $z$  を含む階乗の項の全てが負にならない範囲で和をとることとします。また 0 の階乗は 1 とします ( $0! = 1$ )。  $j_1, j_2, J$  は角運動量、  $m_1, m_2, M$  はそれぞれの角運動量の第三成分で、

- $j_1, j_2, J$  はゼロまたは正の整数あるいは半整数 ( $0, 1/2, 1, 3/2, 2, \dots$ ) の値を取ります
- $m_1, m_2, M$  はそれぞれ  $|m_1| \leq j_1, |m_2| \leq j_2, |M| \leq J$  を満たす整数あるいは半整数の値をとります。ただし  $j$  が整数なら  $m$  も整数、  $j$  が半整数なら  $m$  も半整数となります。例えば  $j_1 = 3/2$  の時は  $m_1 = 3/2, 1/2, -1/2, -3/2$  の値が可能、  $j_2 = 2$  のときは  $m_2 = 2, 1, 0, -1, -2$  の値が可能です。与えられた  $j$  に対して  $m$  の値がこの範囲外であれば CG 係数の値はゼロとします。
- $j_1, j_2, J$  の値については一行目の分子にある階乗の項の値が負になるものは許されません (三角不等式)。階乗の項のどれかが負になる場合は CG 係数の値はゼロとなります。

まずは手計算で  $j_1 = j_2 = 1/2$  の場合のゼロではない CG 係数の値をすべて求めてください (ゼロではない CG 係数を与える  $m_1, m_2, J, M$  の組は全部で 6 つあります。レポートにはこの手計算の結果は不要です)。その後、CG 係数を出力するプログラムを作成し、 ( $j_1 = 1/2, j_2 = 1/2$  の場合に手計算と数値計算の値が一致したのを確認した後)、  $j_1 = 9/2, j_2 = 7/2, J = 1$  の場合のゼロではない CG 係数を与える  $m_1, m_2, M$  の組と CG 係数の値を全て求めてください。

### ■ヒント

- $j$  や  $m$  は整数か半整数で扱いにくいいため、プログラムの中では  $j_1, m_1, j_2, m_2, J, M$  はすべて値を 2 倍したのを使います。プログラム中での  $j1=1$  は  $j_1 = 1/2$  に対応し、  $j2=4$  は  $j_2 = 2$  に対応する、というようにすると半整数をプログラム中で整数として扱えます。

- ただしもとの2倍の値を使っていることを忘れないでください。例えば  $(j_1 - m_1)!$  を計算するときにプログラム中では、階乗を計算する関数が `factorial(n)` であれば、`factorial((j1-m1)/2)` とする必要があります。
- Fortran では平方根を計算する組み込み関数 `sqrt(x)` の引数は実数です。階乗は倍精度実数で計算するとよいでしょう。
- $z$  の取りうる最小値と最大値を決定します。 $z$  の最大値と最小値は `min` 関数と `max` 関数を用いて決定できます。これが決まれば `do` 文あるいは `for` 文で和の計算ができます。例えば自然数  $a, b$  を使って  $\sum_z (-z + a)!(z + b)!$  という項があった場合、 $z$  の取りうる範囲は  $-b \leq z \leq a$  です。
- Clebsch-Gordan 係数の値は教科書やインターネットで値が調べられます。 $j = 5/2$  までの値は

[https://en.wikipedia.org/wiki/Table\\_of\\_Clebsch-Gordan\\_coefficients](https://en.wikipedia.org/wiki/Table_of_Clebsch-Gordan_coefficients)

に掲載されています。

- ゼロでない値のみを出力するには、多重 `do` 文 (`for` 文) で取りうる全ての値を計算し、`if` 文で CG 係数の値がゼロでない場合のみ (例えば絶対値が十分に小さい正の数よりは大きい) 出力するようにします。
- Fortran の関数副プログラムの内部では `print` 文や `write` 文が使えないのでデバッグがやりにくいです。途中経過でチェックしたい値は関数の返り値に一時的に設定して直後に `return` 文を書くことで主プログラム側からチェックをしたり、あるいは主プログラムに関数の内容を直接書いてデバッグをし、動作確認が終わってから関数副プログラムに移すなどの方法が考えられます。