

Física computacional

adaptado de Steve Kooning

Matheus Roos

3 de setembro de 2023

1 Apêndice A (Adaptado ao Fortran 90)

1.1 Arquivos

Existem cinco categorias de arquivos incluídos neste repositório de "Física Computacional": programa principal (program main), módulo (procedimento de módulos), arquivos de dados, scripts Gnuplot (em construção) e *script* Makefile. Eles foram escritos de acordo com os padrões FORTRAN-90 (com atributos do padrão 2003 e 2008) e, portanto, devem ser executados em qualquer máquina com um compilador FORTRAN-90 (gFortran e iFort, por exemplo).

1. Os programas principais consistem nos códigos dos exemplos (chamados 'CHAPnx.f90, onde 'n' indica o número do capítulo e 'x' é uma letra para diferenciar códigos dentro do mesmo capítulo) e exercícios propostos ao final dos capítulos, bem como o projeto final de cada capítulo. Estes programas fazem referência aos módulos e aos *scripts* de plot do software gráfico Gnuplot (que é o padrão na atualidade em contraste com o que o original do livro). Os programas principais também podem gerar um arquivo de dados com extensão '.dat'¹
2. Os módulos trazem constantes físicas e matemáticas bem como um conjunto de sub-rotinas e funções que são utilitários. Estes procedimentos são gerais para todos os códigos, as sub-rotinas inerentes a apenas um capítulo encontram-se dentro do arquivo Fortran que contém o programa principal, estando ali como procedimentos externos.
Como temos que compilar ao menos dois módulos, mais o programa principal, fica mais cômodo agrupar todos estes comandos em um *script* Makefile.
3. Os primeiros cinco códigos utilitários FORTRAN comuns são chamados e, portanto, devem ser desvinculados de cada um dos programas de física. UTIL.f90 contém as rotinas para menus, Entrada/Saída, abertura/fechamento de arquivos, etc. SETUP.f90 contém todas as variáveis e rotinas que dependem do hardware e do compilador, por exemplo, comprimento da tela e número da unidade terminal. Pode ser necessário editar este arquivo para obter a saída mais eficiente e atraente. (Veja a seção A.3.) O código utilitário restante são os arquivos gráficos: 'templates#.gnu', que fornecem as instruções de plotagem no Gnuplot.

¹Poderia ser também '.txt', preferi adotar o usual em Fortran.

Os dezesseis códigos físicos para os exemplos e projetos são denominados 'EXMPLn.f90' e 'PROJn.f90', onde 'n' indica o capítulo. Com duas exceções, estes não requerem edição para serem executados (ver A.3).

Todos os três arquivos de dados possuem a extensão .DAT e contêm dados para serem lidos em EXMPL5.F0R em tempo de execução. Eles não requerem edição.

Todos os três arquivos de dados possuem a extensão .DAT e contêm dados para serem lidos em EXMPL5.F0R em tempo de execução. Eles não requerem edição.

1.2 Compilação

Para compilar os diferentes tipos de arquivos, vou utilizar um *script* Makefile. Então para compilar um programa Fortran do exemplo 1 (Example1.f90), por exemplo, devemos escrever no terminal:

```
make SOURCE=Example1
```

1.3 Execução

Execução: Aqui discutimos as convenções que são seguidas, mais ou menos estreitamente, por todos os programas; as exceções são indicadas nos Apêndices B e C. Todos os programas são projetados para serem executados de forma interativa. Cada um começa com uma página de título descrevendo o problema físico a ser investigado e o resultado que será produzido. A seguir, o menu principal é exibido, oferecendo a opção de inserir valores de parâmetros, examinar valores de parâmetros, executar o programa ou encerrar o programa. Se você optar por executar o programa, a execução continuará e a saída será enviada aos dispositivos solicitados. Para muitos dos programas, o cálculo tem um fim definido.

Input: Há dois momentos em que o programa espera pela entrada. Primeiro, o programa faz uma pausa para você examinar a saída da tela (por exemplo, a página de título, uma página de resultados, um gráfico); a execução é retomada quando você pressiona a tecla Return. Segundo, o programa fará uma pausa para que você insira um valor de parâmetro. Todos os valores de parâmetros têm valores padrão exibidos entre colchetes []; para aceitar o valor padrão, pressione "return" no prompt. Se você optar por inserir um novo valor, mas usar o formato incorreto, o programa solicitará novamente a entrada. Se você inserir um valor fora da faixa definida no menu, o programa solicitará a entrada, informando a faixa de valores permitida.

Número real: Os números reais devem ser inseridos no formato E9.2. Isto é um comprometer para permitir valores padrão. As funções de entrada do usuário (GETFLT e GETINT) aceitam valores padrão lendo a entrada em uma variável de string e verificando o comprimento zero. Se a string não estiver vazia, o valor do parâmetro deverá ser lido a partir da string que atua como uma unidade interna. Portanto, a leitura deve ser formatada, pois as leituras direcionadas à lista de arquivos internos não são padrão. Observe que alguns compiladores não insistem no exponencial (ou seja, 1,23 e 1,23E+0 são ambos compreendidos) ou permitem leituras direcionadas a listas para unidades internas, de modo que as restrições de formatação podem ser relaxadas. Felizmente, os números inteiros podem ser inseridos de maneira "natural", como numerais justificados à esquerda, sem decimal. Isso ocorre porque o programa coloca um ponto decimal após o último caractere não vazio na sequência de entrada e, em seguida, lê o número da sequência usando o formato F7.0.

Em muitos programas, você será solicitado a inserir informações fora do menu principal. Isso acontece por vários motivos. Primeiro, se os parâmetros de entrada não passarem na verificação, o programa solicitará novamente a entrada. Em segundo lugar, pode ser necessária orientação na escolha dos valores dos parâmetros (por exemplo, a faixa de ondas parciais em PROJ4 depende da energia de entrada), e o programa solicita esses valores após os valores padrão razoáveis terem sido

calculados. Terceiro, os cálculos muitas vezes podem permitir a alteração de parâmetros durante a execução (por exemplo, o intervalo de tempo pode ser alterado durante a evolução do tempo), e o programa solicita esses valores durante uma execução.

Por favor, note que a nossa “programação defensiva” não tem sido hermética. É possível travar o programa ou obter resultados sem sentido inserindo parâmetros inadequados. Os parâmetros padrão para cada código estão listados no Apêndice B ou C entre colchetes []. Consulte A.8 para obter detalhes sobre como alterar os valores padrão.

Output: Para acomodar uma variedade de situações, existem várias opções para saída em tempo de execução. A saída de texto pode ser exibida na tela e/ou enviada para um arquivo. A saída gráfica pode ser exibida na tela. Os dados utilizados para gerar gráficos também podem ser enviados para um arquivo. Você pode escolher o tipo de saída alterando os valores padrão em SETUP.f90 ou alterando os parâmetros de E/S no menu em tempo de execução.

1.4 Gráficos

Uma desvantagem da programação em FORTRAN é que não existem gráficos FORTRAN padrão. Isso coloca você em uma de três situações:

1. você não tem nenhum pacote gráfico ou
2. seu pacote gráfico é executado independentemente do programa que cria os dados (ou seja, ele lê os dados de um arquivo) ou
3. seu pacote gráfico pode ser chamado a partir de um FORTRAN. programa.

Se você não tiver nenhum pacote gráfico, deverá editar SETUP.f90 e definir GRFTRM=O, GRFHRD=0 e GRFFIL=0 para que nenhum gráfico seja produzido. Como a saída gráfica melhora muito o uso desses programas, você pode querer considerar adquirir tal pacote (vou utilizar o Gnuplot).

Para aqueles usuários cujo pacote gráfico deve ler os dados de um arquivo, os programas (mediante solicitação) enviarão os dados gráficos para um arquivo. Se você editar SETUP.f90 para que os dados gráficos GRFFIL=1, GRFHRD=0 e GRFTRM=O sejam sempre enviados para um arquivo, mas nunca para a tela.

Finalmente, para aqueles que possuem um pacote compatível com FORTRAN, você terá que modificar nossos códigos para funcionar com seu pacote. Para facilitar esta tarefa, fornecemos um conjunto de rotinas (contidas em GRAPHIT.HI e GRAPHIT.LO) para atuar como interface entre os programas de física e um pacote gráfico geral. Os programas de física chamam as rotinas nos arquivos GRAPHIT, que por sua vez chamam as rotinas do pacote gráfico. Dessa forma, você só precisa modificar um código GRAPHIT em vez de todos os códigos físicos. O código está documentado para facilitar a modificação. Caso contrário, use GRAPHIT.HI ou GRAPHIT.LO, o que for mais apropriado, como modelo para criar rotinas para chamar seu pacote gráfico.

1.5 Estrutura do programa

Referências

- [1] G. Herzberg. *Spectra of Diatomic Molecules*. 1950.
- [2] A. Messiah. *Quantum Mechanics*. 1968.