

APLICAÇÃO DE PARALELISMO OPENMP EM CÁLCULO NUMÉRICO

WILLIAM INSAURRIAGA DA SILVA¹; CARLOS ALBERTO VAZ DE MORAIS
JUNIOR²

¹Univeridade Federal de Pelotas 1 – wins.trab@gmail.com 1

²Univeridade Federal de Pelotas – carlosavjr@gmail.com

1. INTRODUÇÃO

O cálculo computacional consiste de um conjunto de ferramentas ou métodos usados para se obter a solução de problemas matemáticos de forma aproximativa. Esses métodos podem ser aplicados principalmente em problemas que não apresentam uma solução exata, e que portanto precisam ser resolvidos numericamente (SANCHES I.J; FURLAN D. 2007). Podemos ver o seu uso em diversas áreas, incluindo matemática, física entre outras.

Sabemos que o desenvolvimento científico tem exigido simulações e ou cálculos numéricos com resultados cada vez mais sofisticados. A computação científica se depara com o desafio de superar as diversas barreiras que surgem em virtude da limitação física dos computadores. Portanto, a computação de alto desempenho que tem como objetivo aumentar a velocidade de processamento de simulações e ou cálculos numéricos e viabilizar a obtenção de resultados complexos e/ou com elevado grau de detalhes. Para otimização de cálculos, começa a se pensar em multiprocessamento, que é a capacidade de um sistema operacional executar simultaneamente dois ou mais processos (paralelamente). Pressupõe a existência de dois ou mais processadores. As técnicas de processamento paralelo mais difundidas atualmente são as técnicas de memória distribuída e as técnicas de memória compartilhada. Entende-se por técnicas de memória distribuída aquelas que se aplicam aos computadores que possuem arquitetura de memória distribuída, ou seja, máquinas com vários processadores que possuem seu próprio recurso de memória e são interconectados através de uma rede local (SENA M.C ; CALDAS J.A., 2008), (HERMANN M., 2002).

Portanto, o presente trabalho tem como objetivo discutir a aplicação de programação em paralelo em problemas de cálculo numérico. Em particular, o trabalho aqui proposto busca a aplicação de paralelismo OpenMP (“Open Multi-Processing”) na resolução numérica de equações auto-consistentes.

2. METODOLOGIA

Em 1977, Sherrington e Kirkpatrick (KIRKPATRICK; SHERRINGTON 1977) propuseram um dos modelos magnéticos desordenados mais conhecidos para o estudo do problema vidro de spin. Estes introduziram um modelo de Ising com interações de longo alcance, de modo que as impurezas podiam interagir com todas as outras impurezas da rede de acoplamentos magnéticos. Tal modelo é caracterizado principalmente por ser exatamente solúvel. O modelo Sherrington-Kirkpatrick é dado pelo Hamiltoniano

$$H = - \sum_{ij} J_{ij} S_i S_j - h \sum_i S_i ,$$

onde o primeiro termo introduz desordem não trivial e o segundo termo é o campo magnético externo aplicado. O termo J_{ij} é uma variável com distribuição de probabilidade gaussiana. A partir de cálculos analíticos, os parâmetros de ordem magnetização e vidro de spin são obtidos e apresentados abaixo, como segue:

$$m = \int Dz \tanh(\beta H(z))$$

e

$$q = \int Dz \tanh^2(\beta H(z)) ,$$

com $Dz = dz \exp(-z^2/2) / \sqrt{2\pi}$ e $H(z) = J\sqrt{q}z + J_0m + h$.

Em particular, m e q são equações auto-consistentes, em que $m = fm(m, q)$ e $q = fq(m, q)$. Portanto, para obtenção da solução para os parâmetros de ordem m e q , é necessário o cálculo numérico destas equações a partir do uso de rotinas de cálculo de zero de funções bem como rotinas de integração. Para resolução das integrais vistas nas equações apresentadas, métodos de integração numérica podem ser utilizados. Este métodos consistem basicamente na recolocação de uma função $f(x)$ por um polinômio que a aproxime de forma razoável no intervalo $[a, b]$. Assim o problema é solucionado por integração de polinômios, o que é mais simples de se realizar. Em particular, o presente trabalho utiliza o conceito da regra do trapézio, o qual aproxima a função $f(x)$ por um polinômio de ordem um (reta). Portanto, nessa aproximação a integral da função $f(x)$ pode ser aproximada pela área de um trapézio. Em sua forma mais simples, os resultados obtido a partir da regra do trapézio podem apresentar resultados pouco precisos. Portanto, é utilizada regra do trapézio repetida, em que dentro de um certo intervalo $[a, b]$ a regra do trapézio é aplicada repetidas vezes (PILLING S.). Para o cálculo numérico de zero de funções, o método de iteração linear é utilizado. Este método, para o caso uma variável por exemplo, consiste em buscar a raiz de uma função $f(x) = 0$. O Método da Iteração Linear (MIL) transforma o problema, procurando isolar o x da função f , de forma a se obter $x = g(x)$. A partir desse ponto, busca-se a interseção da reta x com a curva $g(x)$ (SANCHES I.J; FURLAN D., 2007). É salientado que o algoritmo para resolução numérica do problema proposto foi desenvolvido em linguagem FORTRAN. Em adição, o algoritmo foi paralelizado seguindo a sintaxe OpenMP ("Open Multi-Processing"). Em particular, a diretiva `!$OMP SECTIONS` (em conjunto com a diretiva `!$OMP PARALLEL`) foi utilizada. Esta diretiva permite atribuir a cada "thread" (conjunto de subtarefas) uma tarefa completamente diferente. A sintaxe desta diretiva é a seguinte: `!$OMP SECTIONS cláusula1, cláusula2; !$OMP SECTION ; !$OMP END SECTIONS fim_cláusula`. Cada bloco do código será executado por uma das "threads", que se inicia com a diretiva `!$OMP SECTION` e se estende até a repetição da diretiva ou até a diretiva de fechamento `!$OMP END SECTIONS`. (HERMANN S. M., 2002).

3. RESULTADOS E DISCUSSÃO

Como discutido, m e q são equações auto-consistentes, em que $m = fm(m, q)$ e $q = fq(m, q)$. Portanto, para obtenção da solução para os parâmetros de ordem m e q (ver equações acima) para um dado valor de J_0 e T , foi empregado o recurso do método numérico de iteração linear.

Basicamente, a solução do problema é obtida com os valores de m e q que satisfazem as condições $m - fm(m, q) = 0$ e $q - fq(m, q) = 0$. Em particular, as funções $fm(m, q)$ e $fq(m, q)$ dependem de uma integral, a qual é resolvida numericamente através da regra do trapézio repetida. A escolha deste método de integração se deve a maior facilidade na sua implementação, quando comparado com outros métodos. Particularmente, foi utilizada linguagem FORTRAN para confecção do algoritmo para o cálculo dos parâmetros m e q .

Em adição, a paralelização do programa é iniciada com a diretiva `!$OMP PARALLEL`. No cálculo dos parâmetros m e q , é empregada a diretiva `!$OMP SECTIONS`, a qual permite a divisão de tarefas na forma de “*threads*” (conjunto de subtarefas). Além do construtor `sections`, que indica que cada “*thread*” irá executar um bloco de instruções, é necessário incluir no código a diretiva `!$OMP SECTION`, que indica qual a instrução que cada “*thread*” irá executar. Particularmente, as sequências de instruções para o cálculo das funções $fm(m, q)$ e $fq(m, q)$ foram executadas em diferentes núcleos de um processador com dois núcleos. Portanto, cada “*thread*” executa um bloco de código diferente. No final encerra-se a paralelização utilizando o comando `!$OMP END SECTIONS` e `!$OMP PARALLEL`. Particularmente, no algoritmo proposto para o cálculo das equações auto-consistentes, não foi observado uma mudança substancial no tempo de execução do programa paralelizado quando comparado ao tempo de execução do programa em sua forma sequencial. No entanto, para problemas mais complexos, isto é, com um maior número de equações, é esperado que a paralelização do código permita um aumento significativo da velocidade de execução das instruções do algoritmo.

4. CONCLUSÕES

A partir da aplicação da sintaxe OpenMP, pode-se obter uma otimização do programa, obtendo um menor tempo de execução ou um “*speedup*” (aumento na velocidade), quando o programa paralelizado foi executado usando OpenMP em uma plataforma com mais de um processador. No algoritmo proposto para o cálculo das equações auto-consistentes, não foi observado uma mudança significativa no tempo de execução, pois o mesmo não era de grande complexidade computacional. No entanto, é esperado que em problemas com maior grau de complexidade, o aumento na velocidade de execução pode ser notável.

5. REFERÊNCIAS BIBLIOGRÁFICAS

KIRKPATRICK, S.; SHERRINGTON, D. Infinite-ranged models of spin-glasses. *Physical Review B. The American Physical Society*, v. 17 p. 4384 1977

HERNANDES M. **PARALLEL PROGRAMMING I FORTRAN 95 USING OPENMP**, School of Aeronautica Engineering, Departamento de Motopropulsión y Termofluidodinámica. Universidad Politécnica de Madrid. Disponível em: http://www.openmp.org/presentations/miguel/F95_OpenMPv1_v2.pdf

PILLING S. **Cálculos numéricos**. Universidade do Vale do Paraíba, Disponível em: http://www1.univap.br/spilling/CN/CN_Capt6.pdf

SANCHES I.J; FURLAN D. **Métodos numéricos**. Universidade do Paraná D.I: 2007. Disponível em:

<http://www.inf.ufpr.br/aurora/disciplinas/numerico/apostila.pdf>

SENA M.C ; CALDAS J.A. **TUTORIAL OpenMP C/C++** programa Campus Ambassador HPC. Disponível em:

<http://www2.ic.uff.br/~bazilio/cursos/progpar/TutorialOpenMP.pdf>