

## SIMULAÇÃO PARALELA DE DINÂMICA BROWNIANA

Roberto Yuji Tanaka (Bolsista PIBIC/CNPq)  
Aluno da Universidade de Taubaté - UNITAU  
Orientador: Dr. Stephan Stephany (LAC/INPE)

O projeto objetiva acelerar o tempo de execução de um programa de simulação numérica através do emprego de técnicas de processamento de alto desempenho. Este simulador, denominado *nrgs*, foi originalmente escrito em Fortran77 pelo Dr. Enzo Granato, do Laboratório Associado de Sensores (LAS/INPE) para execução em sistemas monoprocesados. Execuções típicas deste simulador em microcomputadores ou estações de trabalho atuais levam muitas horas, algumas vezes até mais que um dia, conforme as discretizações espacial e temporal adotadas.

Os parâmetros de entrada do programa estão relacionados com o número de nós do reticulado que representa o supercondutor (discretização espacial) e o número de "time steps" da simulação (discretização temporal).

O programa *nrgs* chama uma subrotina, *calcacel3*, a qual calcula parâmetros relacionadas com a dinâmica de cada nó. Também é chamada a subrotina *noise*, a qual fornece um conjunto de números aleatórios (o número total corresponde ao cubo do número de nós) para cada passo da simulação.

Essas subrotinas constituem os trechos críticos em termos de tempo de execução, e foram, portanto, objeto principal da otimização.

Na subrotina *calcacel3* as seguintes otimizações foram possíveis:

- Eliminação de diversas variáveis que eram desnecessariamente calculadas duas vezes, a cada chamada de *calcacel3*, juntamente com 12 comandos *if* associados a estas variáveis. Estas variáveis foram substituídas por dois vetores, cujos valores são atribuídos uma única vez no início do programa. Como estas variáveis eram definidas dentro de *loops*, a eliminação das atribuições e condicionais deixou-os extremamente simplificados.
- Como estes *loops* envolviam chamadas à função intrínseca *seno*, optou-se por calcular esses senos em *loops* independentes, de forma a se poder particionar a rotina em conjuntos de comandos independentes, buscando modularidade.
- Otimização de outro *loop* da mesma rotina, com eliminação de dois outros comandos *if* e consequente simplificação do mesmo.

Na subrotina *noise*, a exemplo da rotina *calcacel3*, otimizações similares foram efetuadas.

Somente com estas modificações, para uma discretização espacial igual a 6 em cada dimensão e para 1000 *time steps*, obteve-se uma redução do tempo de execução da ordem de 6% (86,09 segundos para 81,29 segundos). Estas otimizações foram efetuadas na linguagem Fortran 77 e o programa executado numa máquina Linux multiprocessada, porém utilizando-se um único processador. O programa já foi portado para Fortran 90 (compilador PGF-90 da *Portland*), obtendo-se uma piora do tempo de execução (cêrca de 20 segundos a mais) na mesma máquina. No momento, estuda-se a causa da execução mais demorada em Fortran 90 e, paralelamente, estão sendo efetuados diversos testes utilizando-se diretivas de paralelização do HPF (High Performance Fortran) para multiprocessamento, na mesma máquina.