



Simulador de Transporte de Partículas Neutras Otimizado Via OpenMP

Matheus Ribeiro Alves, Odair Pinheiro da Silva

A modelagem física do fenômeno de transporte de neutras é constituída da migração destas partículas no interior de um meio material com a probabilidade de interação com os núcleos dos átomos deste meio. Feita esta modelagem física, desenvolve-se uma modelagem matemática que possibilita a simulação da distribuição destas partículas em um determinado domínio. Para fazermos esta modelagem utilizamos a equação linearizada de Boltzmann. Nosso objetivo foi desenvolver um simulador computacional que possibilitasse a realização de experimentos simulados, modo eficiente e com baixo custo. Para problemas monoenergéticos e estacionários de transporte de partículas neutras em meios não multiplicativos na formulação de ordenadas discretas. Para tanto, uma vez representado o problema físico pelas equações linearizadas de Boltzmann, utilizamos o método numérico de malha fina Diamond Difference (DD) e o método de malha grossa Matriz Resposta (MR), da classe dos métodos espectralnodais, para construir algoritmos e desenvolver códigos para gerarem resultados numéricos para os problemas propostos. Esses códigos foram implementados em linguagem C, com estratégias de paralelização utilizando a API OpenMP para paralelizar loops e blocos de códigos. Ademais, construímos uma interface visual, seguindo os padrões de uma API-REST. Neste padrão, o Front-End (Desenvolvido em JavaScript: com ReactJS) permanece isolado do Back-End (Desenvolvido em JavaScript: com NodeJS), permitindo a execução, em um servidor isolado, de todo o código computacional desenvolvido em C. Desta forma, o usuário que possui acesso somente ao servidor da interface consegue enviar os dados, em formato JSON, e capturar os resultados calculados no servidor do Back-End. Assim, esse procedimento permite ao usuário da interface de simulação a utilização desta tecnologia em qualquer dispositivo com acesso ao Browser, tendo o código principal rodando em um computador remoto com os requisitos adequados para garantir a eficiência da simulação. Ademais, as estratégias de paralelização apresentaram, nos problemas testados, um ganho significativo de tempo de execução, que representou uma eficiência variando entre 40% e 60%. Finalizamos este projeto com resultados bastante satisfatórios quanto a eficiência do código computacional desenvolvido para o simulador. Também, quanto a implementação da interface de gerenciamento do simulador que permite ao usuário realizar simulações usando equipamentos com baixo requisito de hardware, acessando máquinas mais potentes de forma remota simplesmente usando um Browser.