

Protótipo de um ambiente integrado de código aberto voltado para computação numérica

Daniel Leal Souza^{1,2}, Tiago Carvalho Martins², Victor Dmitriev²

¹Área de Ciências Exatas e Tecnologia (ACET), Centro Universitário do Pará (CESUPA)
Av. Gov. José Malcher, 1963 – 66060-230 – Belém, PA, Brasil

²Laboratório de Nanofotônica e Nanoeletrônica, Universidade Federal do Pará (UFPA)
E-mail: daniel.leal.souza@gmail.com, tiagocm@ufpa.br, victor@ufpa.br

APRESENTAÇÃO

Diversos campos da ciência, em especial, as áreas das ciências exatas, naturais e as engenharias buscam por soluções na computação científica, mais especificamente na análise numérica [1]. Em geral, as aplicações voltadas para a análise numérica possuem natureza de código fechado, o que inibe outros programadores de customizar a aplicação em si, além de possuir licenças de preços muito elevados.

A utilização de software livre na computação científica tem se intensificado nos últimos anos. Diversas bibliotecas para análise numérica como a GSL (GNU Scientific Library) e LAPACK, as quais podem ser utilizadas para resolver um grande número de problemas de várias áreas da matemática, física e da engenharia [2], mostram a viabilidade da utilização de software livre em ambientes voltados para a computação numérica.

Com base nas ferramentas de código aberto utilizadas no meio acadêmico-científico, foi construído um ambiente de desenvolvimento integrado (IDE) sob a licença *GNU Lesser General Public License (LGPL)*, escrito na linguagem Java. O software encontra-se em fase de protótipo, mas já integra em um único ambiente, diversas ferramentas *open source*. Utiliza-se os compiladores da série GNU GCC for C, G++ e GFortran; o interpretador da linguagem Python; as bibliotecas LAPACK, SciPy, NumPy e GSL; o depurador GNU Debugger; o sistemas de plotagem GnuPlot e o programa de *ray tracing* POV-Ray. Para a comunicação da IDE com as aplicações, é utilizada a classe *java.lang.ProcessBuilder*. A interface gráfica, bem como o assistente de plotagem podem ser vistos na Fig. 1.

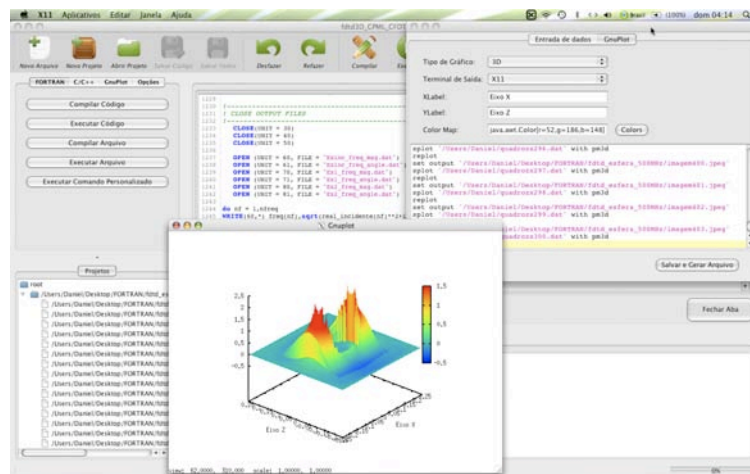


Fig. 1. IDE executando um script GnuPlot gerado por código Fortran

SIMULAÇÃO

A IDE foi utilizada na construção da aplicação escrita em Fortran, utilizada para simular a incidência de um pulso gaussiano com frequência central de 500 MHz sobre uma esfera dielétrica com perdas, de raio 10 cm, permissividade elétrica relativa igual a 30 e de condutividade elétrica igual a 0,3 S/m. O ambiente computacional consiste de um domínio de

análise truncado por CPML (*Convolutional Perfectly Matched Layers*), com incremento espacial de 0,5 cm e incremento temporal de 0,962 ns. As Fig. 2 e 3 foram geradas a partir dos arquivos de saída *.dat* gerados a partir código Fortran construído na IDE.

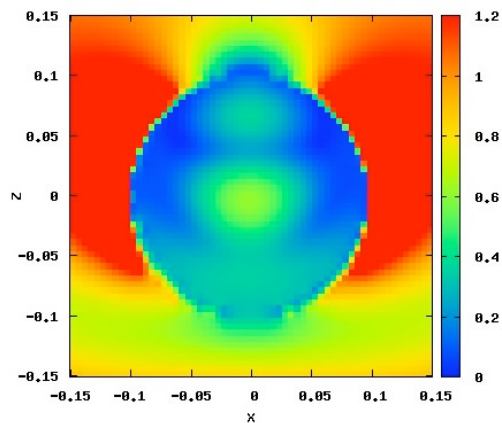


Fig. 2. Distribuição de campo elétrico total no domínio da frequência (obtida por transformada de Fourier dos campos no domínio do tempo), na frequência de 500 MHz

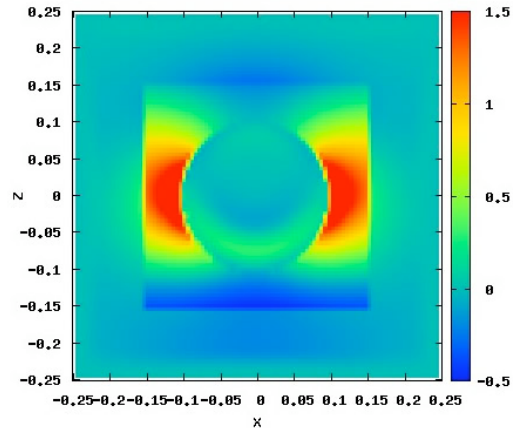


Fig. 3. Distribuição de campo elétrico total no domínio do tempo, na iteração 295 da simulação

Na Fig. 3 é mostrada a distribuição de campo elétrico total no domínio do tempo, localizada no plano $y = 0$, em que a onda propaga-se no sentido positivo do eixo z . Na Fig. 2, é mostrada a distribuição de campo elétrico total no domínio da frequência, em 500 MHz.

A experiência realizada na IDE, mostrou que as etapas de desenvolvimento, desde a edição de código, compilação, depuração, execução e plotagem de gráficos, além de animações ocorreram sem problemas e de forma estável. O protótipo estará disponível em breve para download, incluindo o código fonte permitindo que outros colaboradores possam aprimorar as funcionalidades já existentes ou inserir novas, dando assim, continuidade ao projeto. Este projeto visa construir uma IDE customizável, intuitiva, flexível e compatível com as principais linguagens voltadas para o ambiente científico e ferramentas que auxiliem no seu desenvolvimento. Embora as ferramentas apresentadas neste trabalho já tenham atingido os objetivos esperados, alguns testes ainda precisam ser feitos, além de outras implementações, para que possamos aprimorar a aplicação e torna-lá mais robusta.

Além do aprimoramento dos recursos já disponíveis, e de eventuais correções de *bugs*, iniciativas futuras envolvem a integração com a API Java Plugin Framework, o que tornará possível a inserção de novas funcionalidades através de *plugins*, a inserção das linguagens de alto nível GNU Octave e R, a adição de novas funcionalidades para trabalhos envolvendo análise numérica em Python e uma biblioteca de rotinas e funções, cujo objetivo será a criação de uma interface editável que acessa um banco de funções e rotinas pré-moldadas, além de permitir a entrada de novas através do usuário.

Palavras-chave: *Ambiente de Desenvolvimento Integrado, Computação Numérica, Software Livre, FDTD.*

Referências

- [1] A. S. Charao, C. Schepke, *Comparação entre Java e C++ na Computação Numérica*, em Quinto Workshop em Sistemas Computacionais de Alto Desempenho, 2004.
- [2] R. L. Burden, J. D. Faires, “Análise Numérica”, Cengage Learning Editores, 2003.
- [3] A. Taflove, S. C. Hagness, *Computational Electrodynamics the Finite-Difference Time-Domain Method*, Artech House, Boston, 2005.