

## AMBIENTES INTERVALARES

MARILINE LORINI<sup>1</sup>; LUCAS MENDES TORTELLI<sup>2</sup>; ERICO ALVES GREHS<sup>2</sup>;  
 MAURICIO DORNELES CALDEIRA BALBONI<sup>2</sup>; ALICE FONSECA FINGER<sup>2</sup>;  
 ALINE BRUM LORETO<sup>3</sup>

<sup>1</sup>Universidade Federal de Pelotas (UFPEL) – mlorini@inf.ufpel.edu.br

<sup>2</sup>Universidade Federal de Pelotas (UFPEL) - lmtortelli@inf.ufpel.edu.br, eagrehs@inf.ufpel.edu.br  
 baalbis@gmail.com; affinger@inf.ufpel.edu.br

<sup>3</sup>Universidade Federal de Pelotas (UFPEL) – aline.loreto@inf.ufpel.edu.br

### 1. INTRODUÇÃO

A precisão de cálculos matemáticos, em diversos ramos da ciência e tecnologia, tem sido tema de pesquisa de trabalhos científicos, buscando sempre o desenvolvimento de algoritmos aritméticos com o objetivo de alcançar melhor exatidão dentro de um processamento de dados numéricos (SANTOS, 2001).

A matemática intervalar vem para auxiliar este processo e tem como objetivo, segundo MOORE (1966), proporcionar um controle de erros numéricos. Utilizando-se intervalos para representação dos números reais, é possível controlar a propagação de erros de arredondamento ou truncamento, entre outros, em procedimentos numéricos computacionais. Portanto, resultados intervalares carregam consigo a segurança de sua qualidade.

Cálculos numéricos em computadores devem ser realizados por meio das linguagens ou bibliotecas que tenham definidos o tipo de intervalo e as operações sobre o tipo, usualmente denominadas de linguagens XSC (*eXtended for Scientific Computation*) (KLATTE, 1993).

Existem diversos ambientes de programação para Matemática Intervalar, como: C-XSC; Maple Intervalar; IntLab e IntPy. O objetivo desse trabalho é analisar os ambientes e escolher qual é o melhor (ou mais adequado) para utilizar em trabalhos futuros, considerando suas características enquanto *softwares* de boa qualidade, as quais são: manutenibilidade; operacionalidade; portatibilidade; reutilizabilidade; eficiência; rentabilidade e avaliabilidade (ROCHA, 1993). Além de, segundo GONÇALVES (2003), avaliar características técnicas como a apresentação clara de objetivos e indicação das possibilidades de uso, facilidade de instalação e desinstalação, fornecimento do manual de utilização, atualização de conteúdo via internet e destacando aqueles que se enquadram nas categorias de código aberto, gratuitos ou livres e tipos de licença.

### 2. METODOLOGIA

O controle do erro numérico é realizado através do uso de intervalos no lugar de números reais. Segundo pesquisas teóricas, KULISCH (2008) e KULISCH e MIRANKER (1981) propuseram que a implementação da aritmética intervalar seja realizada através da chamada aritmética de exatidão máxima, o que significa a busca para que resultados numéricos ou sejam um número de ponto flutuante ou estejam entre dois números de ponto flutuantes consecutivos.

Descreve-se brevemente os ambientes intervalares a serem analisados segundo critérios de qualidade. São eles:

### **C-XSC (eXtended for Scientific Computation)**

O ambiente de programação C-XSC (*C++ for eXtended Scientific Computing*) é uma ferramenta de programação de fácil uso, especialmente para aplicações científicas e de engenharia. Trata-se de uma biblioteca numérica para a Computação Científica baseada na linguagem C++. É uma ferramenta gratuita que possui a alta geração de resultados com exatidão e verificados automaticamente (KLATTE et al., 1993). Fornece um grande número de tipos de dados numéricos e operadores predefinidos. Estes tipos são implementados como classes da linguagem C++. Assim, o C-XSC permite a programação de alto nível de aplicações numéricas em C++. Disponível para muitos ambientes computacionais que possuem um compilador C-XSC. As características mais importantes do C-XSC são (HOLBIG, 2005):

- Aritmética intervalar para números reais, complexos, intervalares e intervalares complexos com propriedades definidas matematicamente;
- Tipos de dados com alta exatidão;
- Operadores aritméticos predefinidos com alta exatidão;
- Aritmética de múltipla precisão dinâmica e funções padrão;
- Controle de arredondamento para os dados de entrada e saída;
- Bibliotecas de rotinas para resolução de problemas numéricos;
- Resultados numéricos com rigor matemático.

C-XSC é particularmente adequado para o desenvolvimento de algoritmos numéricos que proporcionam resultados altamente precisos e verificados automaticamente. Todas as operações básicas (intervalo) são de exatidão máxima (KLATTE et al., 1993).

### **Maple Intervalar**

O software matemático Maple é um ambiente interativo com uma interface amigável que, para muitas finalidades, dispensa a programação. Bibliotecas Maple, uma vez carregadas, disponibilizam os comandos e operadores necessários para cálculos específicos. Possui uma linguagem de programação fundamentada no conceito de linguagem interpretada e um mecanismo de construção e distribuição de pacotes de programas e funções. É largamente utilizado em computação científica e também na implementação de protótipos de sistemas de grande porte que, após testados no Maple, são posteriormente implementados em outras linguagens que proporcionem um processamento mais rápido do que sistemas de computação algébricos. Software proprietário (custoso) (CAMPOS, 2002).

### **IntLab**

IntLab (intervalo LABoratory) é um pacote desenvolvido para a ferramenta *Matlab*<sup>®</sup> (MATLAB). O foco principal é produzir resultados confiáveis. Qualquer resultado provou ser verdadeiro sob quaisquer circunstâncias, nomeadamente em matéria de erros de arredondamento e todos os termos de erro. A filosofia da INTLAB é que tudo é escrito em código Matlab para garantir melhor portabilidade, usa extensivamente BLAS rotinas o que garante tempos de computação rápida, comparáveis à aritmética pura de ponto flutuante. Vetor intervalo e operações de matriz são muito rápidos em INTLAB, no entanto, os cálculos não lineares e *loops* podem retardar o sistema de forma significativa, devido à sobrecarga de interpretação e uso extensivo do conceito de operador. Software proprietário (custoso) (MATLAB).

## IntPy

O pacote IntPy é composto de 2 subpacotes, *support* e *irreal*, além de um módulo que encapsula as classes de exceção. O subpacote *support* agrupa toda a funcionalidade de suporte ao IntPy e é composto de três módulos: *roundingmodule.c*, *general.py*, *stdfunc.py*. O primeiro módulo é uma extensão C para Python que manipula os modos de arredondamento do processador. Este módulo é dependência para toda funcionalidade do IntPy. O segundo módulo organiza todas as peças de software pontuais. Uma dessas peças é a função *rational2fraction* (racional) responsável por converter uma representação em *string* de números racionais em outra representação mais facilmente manipulável por computador. O módulo *stdfunc.py* merece atenção especial.

O subpacote *irreal* compreende dois módulos: *irreal.py* e *irmath.py*. O primeiro implementa a classe IReal que representa o tipo Intervalo Real. Todas as operações aritméticas e de conjuntos, relações de ordem e funções auxiliares são implementadas nessa classe. O segundo módulo implementa extensões intervalares das funções padrão. Todos esses implementados na linguagem Python reconhecidos como software livre GPL.

Segundo ROCHA (1993), um software de qualidade deve atingir determinados objetivos:

- confiabilidade conceitual: o produto precisa satisfazer às necessidades e requisitos que motivaram sua construção;
- confiabilidade da representação: se refere às características de representação do produto que afetam sua compreensão e manipulação;
- utilizabilidade: determina a conveniência e a viabilidade de utilização do produto ao longo de sua vida útil. Para que um produto seja utilizável, são necessárias a confiabilidade conceitual e a confiabilidade de representação.

### 3. RESULTADOS E DISCUSSÃO

Até o presente momento, buscou-se referências na teoria, as quais se enfocam na maior precisão dos resultados nos cálculos da matemática intervalar e não na qualidade do software.

A Tabela 1 apresenta a análise das características dos softwares nos ambientes intervalares C-XSC, Maple Intervalar, IntLab e IntPy:

Tabela 1: Análise das características de qualidade dos softwares.

Características	C-XSC	Maple Intervalar	IntLab	IntPy
Manutenibilidade	Bom	Bom	Bom	Médio
Operacionalidade	Médio	Bom	Bom	Ruim
Portatilidade	Bom	Médio	Médio	Bom
Reutilizabilidade	Excelente	Bom	Bom	Excelente
Eficiência	Excelente	Ruim	Bom	Excelente
Rentabilidade	Excelente	Ruim	Bom	Excelente
Avaliabilidade	Excelente	Bom	Bom	Excelente

Sobre os softwares que são pagos, como IntLab e Maple Intervalar, pode-se dizer que são de boa atualização, fácil instalação e possuem manual de utilização. Já os que são gratuitos, como C-XSC e IntPy, também possuem manual de utilização, mas suas atualizações não são tão boas quanto dos

softwares proprietários. Saliente-se que o IntPy é de fácil instalação, porém o C-XSC não.

#### 4. CONCLUSÕES

Softwares são desenvolvidos para atender às necessidades de seus usuários e devem ter uma vida útil, produtiva e longa (ROCHA, 1993). Neste contexto o presente trabalho analisou os ambientes intervalares C-XSC, Maple Intervalar, IntLab e IntPy considerando suas características enquanto softwares de boa qualidade. Enfatiza-se que qualidade de software pode ser definida como um conjunto de propriedades a serem satisfeitas em determinado grau, de modo que o software satisfaça as necessidades de seus usuários.

Assim, pelos resultados conclui-se que os melhores ambientes nos critérios citados acima são: C-XSC e IntPy. Não deixando de citar que no IntLab existe uma grande exatidão nos resultados e que o Maple é o de mais fácil utilização.

#### 5. REFERÊNCIAS BIBLIOGRÁFICAS

CAMPOS, M. A. **Probabilidade Intervalar e Cadeias de Markov Intervalares no Maple**. 2002. Universidade Federal de Pernambuco.

GONÇALVES, I. **Análise dos diferentes tipos de Softwares usados na educação**. Online. Disponível em: [http://geocities.ws/ivanete20032002/aval-sofword\\_iva.html](http://geocities.ws/ivanete20032002/aval-sofword_iva.html). Acesso em: 10 mar. 2012.

HOLBIG, C. A. **Ambiente de Alto Desempenho com Alta Exatidão para Resolução de Problemas**. 2005. Dissertação (Doutorado em Computação) – Curso de Pós-graduação em Computação, Universidade Federal do Rio Grande do Sul.

KLATTE, R, KULISCH, U., WIETHOFF, A., LAWOW, C., RAUCH, M. **C-XSC - A C++ Class Library for Extended Scientific Computing**. Springer-Verlag, Heidelberg, 1993.

KULISCH, U. and MIRANKER, L. **Computer Arithmetic in Theory and Practice**, 1st ed., Academic Press, 1981.

KULISCH, U. W. (2008, apr) **Complete interval arithmetic and its implementation on the computer**. Online. Disponível em: <http://www.math.kit.edu/iwrmm/seite/preprints/media/preprintn%20nr>

MATLAB Intlab. Disponível em: [www.mathworks.com](http://www.mathworks.com).

ROCHA, A. R.; CAMPOS, G. H. B. **Avaliação Da Qualidade de Software Educacional**. Brasília, ano 12, n. 57, p. 32- 44. jan./mar. 1993.

SANTOS, J, M. **Em Direção a Uma Representação Para Equações Algébricas: Uma Lógica Equacional Local**. 2001. Dissertação (Mestrado) Programa de Pós-graduação em Sistemas e Computação. Universidade Federal do Rio Grande Do Norte.