

Um estudo para o desenvolvimento de rotinas otimizadas para funções básicas da biblioteca C-XSC

Dimas Gabriel, Carlos Amaral Hölbis

Ciência da Computação – Universidade de Passo Fundo (UPF)
CEP: 99.001-970 – Passo Fundo – RS – Brasil

{75962,holbis}@upf.br

Resumo. *Este artigo descreve um estudo em andamento sobre a otimização da biblioteca C-XSC através de rotinas otimizadas para o tratamento de funções básicas da biblioteca. Será feito uma descrição dos conceitos básicos da Computação Verificada e de Alta Exatidão. Além disso, pontos críticos para a obtenção de desempenho no C-XSC serão descritos e possíveis formas de otimização destes pontos serão abordadas.*

1. Introdução

A utilização de computadores para a resolução de problemas matemáticos fez com que novos algoritmos fossem desenvolvidos, visando, principalmente, o aumento do desempenho computacional. Entretanto, quando se trabalha com Computação Científica, especialmente com aplicações reais de grande porte utilizando números em ponto-flutuante, deve-se dar especial atenção ao controle dos erros gerados pelas computações numéricas, que muitas vezes podem apresentar resultados totalmente errados devido a inexatidão da representação numérica computacional, já que o resultado gerado é apenas uma representação do resultado exato.

A medida em que o controle dos erros é feito o desempenho na execução dos programas é prejudicado e, em algumas aplicações científicas, o tempo de execução é um fator primordial. Daí vem a importância de se ter bibliotecas científicas que tratem os erros gerados e que, ao mesmo tempo, façam os cálculos com um desempenho aceitável. Por isso, a ideia central deste trabalho é providenciar a uma biblioteca de alta exatidão (C-XSC) algumas rotinas otimizadas com o intuito de melhorar seu desempenho quando da solução de problemas que exijam um grande controle da qualidade (exatidão) do resultado que está sendo calculado.

Este trabalho originou-se da tese doutorado de Hölbis (2005) e do trabalho de Grimmer (2005) onde foi realizada a integração da biblioteca C-XSC com a biblioteca MPI em *clusters* de computadores, mas como o desempenho computacional obtido na época com os testes realizados não foi satisfatório surgiu a ideia, em conjunto com pesquisadores das Universidades de Wuppertal e Karlsruhe na Alemanha, de realizar a otimização de algumas funções da biblioteca C-XSC.

2. Computação Verificada e de Alta Exatidão

Com o objetivo de tratar os erros gerados pelas computações numéricas o paradigma da Computação Verificada foi desenvolvido, fazendo com que o próprio computador possa verificar se o cálculo realizado é ou não correto e útil para o problema que se quer solucionar. É importante ressaltar que para o desenvolvimento de programas computacionais que tenham o paradigma da Computação Verificada é obrigatório o uso de todas as suas características, ou seja, o uso da aritmética de alta exatidão, dos métodos intervalares de inclusão e da convergência garantida pelo Teorema de Ponto Fixo de Brouwer, além do uso de algoritmos apropriados.

Sobre a Computação de Alta Exatidão, na aritmética de ponto-flutuante ordinária (definida pelo padrão IEEE-754 [IEEE 1985]), a maioria dos erros ocorre em acumulações, isso é, pela execução de uma sequência de adições e subtrações. Na aritmética de ponto-fixado, contudo, a operação de acumulação é realizada sem erros. Assim, grande parte dos erros encontrados em ponto-flutuante poderiam ser evitados se a acumulação fosse realizada em ponto-fixado, em um acumulador especial [Miranker and Toupin 1986].

Com a atual tecnologia pode-se realizar facilmente essa acumulação de ponto-fixado. Necessita-se, somente, providenciar um registrador de ponto-fixado na unidade aritmética que cubra todo o domínio em ponto-flutuante. Se um registrador que possui esta característica não está disponível, então esse pode ser simulado na memória principal por software. Isso infelizmente pode resultar em perda de velocidade, que, em muitos casos, é mais importante do que o ganho em confiabilidade do resultado.

Além disso, para a garantia de um resultado correto é preciso utilizar-se de arredondamentos direcionados, isto é, arredondamento para o número de máquina anterior, superior e para o mais próximo de máquina. Atualmente existem várias bibliotecas que suportam a Computação Verificada ou algumas de suas características. Dentre estas bibliotecas pode-se citar a IntLab (toolbox intervalar para o Matlab), IntPackX (toolbox intervalar para o Maple), compilador Fortran HPC da Sun e a biblioteca matemática da Intel (MKL) que disponibiliza a aritmética intervalar em sua versão em Fortran. A escolha da biblioteca C-XSC para esta pesquisa deve-se ao fato a sua portabilidade e que esta biblioteca disponibiliza todas as características da Computação Verificada.

3. C-XSC

Abordando o conceito da Computação Verificada e da Computação de Alta Exatidão, bibliotecas de programação foram desenvolvidas para um grande número de linguagens. Uma delas é o C-XSC. Baseado e implementado para o uso com o C++, o C-XSC tem como principal característica a qualidade do resultado apresentado, que difere do resultado real em apenas um arredondamento.

É uma ferramenta para o desenvolvimento de algoritmos numéricos com a geração de resultados com alta exatidão e verificados automaticamente. Ela 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++. Ela está disponível para

ambientes computacionais que possuam um compilador C++ e obedece ao padrão ISO/IEC C++. Atualmente a versão disponível do C-XSC é a C-XSC 2.2.2 (o C-XSC é uma biblioteca livre). Maiores detalhes poderão ser obtidos em [Hammer 1985], [Hofschuster and Krämer 2001], [Klatte 1993] e [Krämer, Kulisch and Lohner 1996] ou pelo endereço <http://www.xsc.de>.

Entretanto, alguns problemas relacionados ao desempenho no uso do C-XSC estão sendo notados [Carmo 2006]. É perfeitamente aceitável que bibliotecas que implementem a Computação Verificada e de Alta Exatidão façam cálculos com um tempo computacional mais alto, já que, controles de arredondamento precisam ser realizados a todo momento e os dados em alta exatidão são representados na memória principal e não em um registrador específico. Contudo, comparações feitas entre o C-XSC e outras bibliotecas matemáticas demonstram que o desempenho da mesma está além dos padrões estabelecidos pelos próprios criadores da biblioteca. Por isso, o grande objetivo no desenvolvimento atual da biblioteca é fazer com que o seu desempenho aumente e o tempo necessário para a realização de um cálculo não seja superior à 15 vezes o tempo necessário para a realização do mesmo cálculo na maneira tradicional (sem o controle de erros).

4. Conclusões e Trabalhos Futuros

Com o objetivo de aumentar o desempenho da biblioteca, este trabalho abordará questões teóricas importantes para o desenvolvimento de novas rotinas otimizadas para o C-XSC. Um dos pontos críticos com relação ao desempenho computacional do C-XSC é a utilização do tipo de dado de alta exatidão (*dotprecision*) que utiliza cerca de 5000 bits para armazenar seus valores e o fato de, quando se trabalha com a aritmética intervalar, haver uma constante troca do arredondamento do compilador (as operações intervalares exigem arredondamentos para cima e para baixo, o que causa perda de desempenho ao programa pelo fato dele não usar apenas o arredondamento *default* que é para o número mais próximo de máquina).

O próximo passo deste trabalho será desenvolver estas (novas ou adaptadas) rotinas e inseri-las em funções básicas do C-XSC, em especial funções para a troca do tipo de arredondamento do compilador e em operações envolvendo o somatório, vetores e matrizes. Estas rotinas foram escolhidas por, no contexto da área da Computação Verificada, serem as mais utilizadas e as que demandam um maior tempo de processamento. As rotinas que serão desenvolvidas serão baseadas nas já existentes do C-XSC, aplicando conceitos de otimização utilizados em outras ferramentas computacionais sobre a implementação das mesmas e, caso possível, com a utilização de rotinas da biblioteca BLAS. Como base para este trabalho serão estudadas formas de implementação de rotinas de bibliotecas tradicionais que visam um alto desempenho computacional e de bibliotecas intervalares que, de alguma forma, foram implementadas objetivando uma melhora no seu desempenho em prol de uma exatidão considerada aceitável. Com este estudo será possível desenvolver estas novas rotinas (otimizadas) para a biblioteca C-XSC propiciando uma melhora no desempenho computacional dos programas implementados com rotinas do C-XSC, tanto em ambientes sequenciais como em ambientes paralelos.

Referências

- Carmo, Andriele B. (2006) “Paralelização de Solver Intervalar para Resolução de Sistemas Lineares Densos“, Universidade de Passo Fundo.
- Hofschuster, W., Krämer, W., Wedner, S. and Wiethoff, A. (2001) “C-XSC 2.0: A C++ Class Library for Extended Scientific Computing“, In: – WRSWT 2001/1. Annals..., Universität Wuppertal.
- Hölbig, C.A. (2005) Ambiente de Alto Desempenho com Alta Exatidão para a Resolução de Problemas. Tese (Instituto de Informática/Programa de Pós-Graduação em Computação) – Universidade Federal do Rio Grande do Sul, PortoAlegre.
- Grimmer, M. (2005) “An MPI Extension for the Use of C-XSC in Parallel Environments“, In: WRSWT 2005/3, Annals..., Universität Wuppertal.
- IEEE. (1985) IEEE 754: Standart for Binary Floating-Point Arithmetic. New York.
- Klatte, R., Kulisch, U., Wiethoff, A., Lawo, C. and Rauch, M. (1993) C-XSC – A C++ Class Library for Extended Scientific Computing. Springer-Verlag.
- Krämer, W., Kulisch, U. and Lohner, R. (1996). Numerical Toolbox for Verified Computing II: Advanced Numerical Problems. Springer-Verlag.
- Miranker, W. and Toupin, R. (1986) “Accurate Scientific Computations“, In: Symposium on Accurate Scientific Computations, Proceedings. . . Springer-Verlag, (Lecture Notes in Computer Science, 235).