Projeto I: estudo de aritmética de máquina

Danrley Tavares, Filipe Bessa, Ielane Nogueira , Ruben Esteche, Tertius Ferraz
Alunos da disciplina Cálculo Numérico, turma T8.

Resumo: *Este trabalho constitui um estudo de aritmética de máquina. Para tal, foi feita uma análise de operações matemáticas básicas, como adição e subtração, utilizando técnicas de aritmética de ponto flutuante e de arredondamento em uma calculadora programada em linguagem C.*
Palavras-chave: aritmética de máquina; arredondamento; erros; algoritmo.

1. Introdução

Devido a limitações físicas, não é possível representar e realizar operações com números reais em máquinas da mesma maneira que na matemática real, pois elas armazenam um número finito de algarismos. De modo a minimizar os erros causados por essas limitações, foram propostas representações para os números reais (ditas número de máquinas), sendo a de ponto flutuante a mais utilizada atualmente.

Este estudo visa demonstrar operações aritméticas básicas (como adição e subtração) utilizando técnicas de aritmética de ponto flutuante e de arredondamento em uma calculadora programada em linguagem C, além de avaliar os erros gerados pelo emprego tais técnicas em máquinas.

2. Desenvolvimento

A. Número de ponto flutuante normalizado

Representar números reais em máquinas demanda espaço na memória e realizar operações com eles custa processamento. Por isso, foram sugeridas ao longo dos anos formas de representação distintas, sendo a mais utilizada atualmente a de ponto flutuante. Ela consiste em representar os números de maneira única, na forma de notação científica. A expressão geral para o número de ponto flutuante é apresentada a seguir:

$$x = m \times b^e$$

Onde, $m = \pm d_0,d_1d_2...d_{t-1}$, ($d_0 \neq 0$, $t \in$ N) é chamado de mantissa normalizada; $b \geq 2$, é a base; e ($e \in$ Z, $e_1 \leq e \leq e_2$) é o expoente; e t é o número de algarismos significativos da máquina. Esses parâmetros são chamados de especificação de máquina (b, t, e_1, e_2) e determinam o número de algarismos que um computador pode armazenar, além de quais são o seu maior e menor número representado. Os números mínimo e máximo podem ser encontrados através das seguintes expressões, respectivamente:

$$x_{\min} = \pm 1{,}000\ldots0 \times b^{e_1} \text{ (}e_1\text{ é o menor expoente)}$$
$$x_{\text{máx}} = \pm 9{,}999\ldots9 \times b^{e_2} \text{ (}e_2\text{ é o maior expoente)}$$

O número 4028599, por exemplo, seria representado como 4,028599 x 10^6 em uma máquina com especificações (10, 7, -8, 8).

Quando um número é maior que o $x_{máx}$ ou menor que o x_{min}, seja ele positivo ou negativo, ele não é representado pela máquina e é determinado como pertencente à região de *overflow* e *underflow*, respectivamente. O número 0 (zero) é representado como 0,000...0 x b^0.

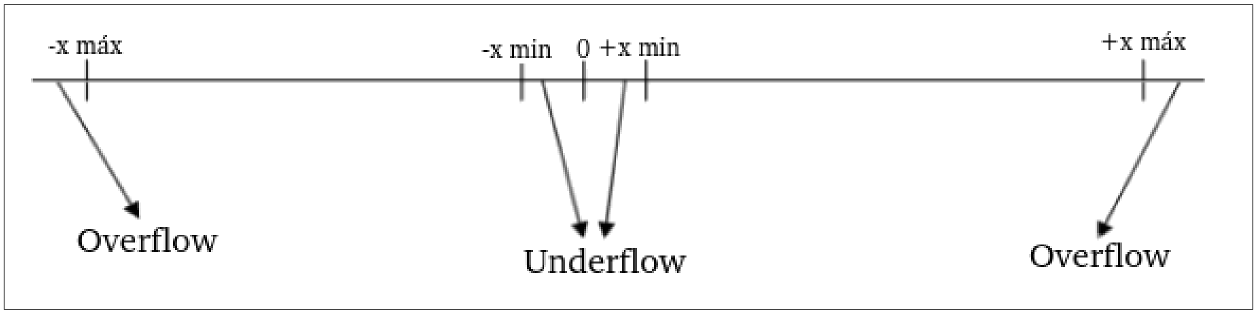

Figura 1: Representação das regiões de *overflow* e *underflow*

B. Arredondamento

Quando se faz operações em uma máquina, o resultado é frequentemente um número aproximado, pois as máquinas possuem capacidade finita para armazenar informações e somente é capaz de representar números limitados. Seja uma máquina F(b,t,e_1,e_2) em um sistema de ponto flutuante, temos que uma função O: R→ F com $x \rightarrow Ox = x'$, é chamado de arredondamento se:

i) $x = x'$, se $x \in$ F;

ii) $x \notin$ F, então entre x e x' não existem elementos de F.

A partir dessa definição tem-se que o resultado de alguma operação realizada numa máquina pode ser um número da máquina ou pode estar entre dois números consecutivos da máquina.

No caso do resultado ser um número da máquina, não há problema, pois ele será representado por si próprio, mas caso o resultado da operação esteja entre dois números consecutivos. Considere-se: u (o último significando) e u_1 (o algarismo após u). Se u_1 for maior que 5 (cinco), u sobe uma unidade; se u_1 for menor que 5, u permanece inalterado; se u_1 for igual a 5 e u for par, u sobe uma unidade, caso contrário, permanece inalterado.

Por exemplo, utilizando-se uma máquina com t = 5 para arredondar os seguintes valores de x, sendo x_1 e x_2 seus números de máquina consecutivos:

X	**x'**	**x_1**	**x_2**
5,3461x10^0	5,3461x10^0	5,3460x10^0	5,3462x10^0
5,34313x10^0	5,3431x10^0	5,3431x10^0	5,3432x10^0
5,34318x10^0	5,3432x10^0	5,3431x10^0	5,3432x10^0
5,34315x10^0	5,3431x10^0	5,3431x10^0	5,3432x10^0

Tabela 1: Exemplos de arredondamentos

C. Erro relativo

Quando são efetuadas operações em uma máquina, como a de arredondamento de números reais para que eles se adequem a um número finito de algarismos significativos após operações aritméticas, por exemplo, é gerado, automaticamente um erro. A representação desse fenômeno é de extrema importância pois ele indica o quanto as técnicas de arredondamento influenciaram na acurácia do resultado das operações aritméticas. Neste estudo, será adotado o conceito de erro relativo, pois ele expressa a perda com relação ao número sem arredondamento (o número real).

O erro relativo (δx) é denotado como o quociente entre o erro absoluto (Δx, o módulo da diferença entre o valor exato x e o valor arredondado x') e o valor exato.

$$\delta x = \frac{\Delta x}{x} = \frac{|x - x'|}{x}$$

Sejam x, x_1 e x_2 números de máquina e $x = x_1 \pm x_2$, o erro relativo de x é definido por:

$$\delta x = \frac{(\Delta x1 \pm \Delta x2)}{|x1' \pm x2'|} = \frac{(|x1 - x1'| \pm |x2 - x2'|)}{|x1' \pm x2'|}$$

C. Operações de adição e subtração

Como uma máquina trabalha com uma quantidade finita de números, não consegue utilizar a matemática real para realizar as operações. Assim, a máquina utiliza algumas operações para trabalhar, como a adição e subtração.

Dada uma máquina F(b, t, e_1, e_2). Sejam dois números de máquina $x_1 = m_1 \times b^{c1}$, $x_2 = m_2 \times b^{c2}$, e $x = x_1 + x_2$. Se $c_1 \neq c_2$, os expoentes devem ser igualados. Esse procedimento é realizado renormalizando o número de máquina com menor expoente, só então, é possível efetuar a soma de x_1 e x_2. Após esse processo, o resultado da adição é arredondado, se necessário, encontrando-se $x = m \times b^c$.

A subtração equivale à adição, sendo: $x = x_1 + (-x_2)$ ou $x = (-x_1) + x_2$.

E. Calculadora

Foi elaborado um programa em linguagem C (uma calculadora que trabalha com base 10), normalizar e, se necessário, renormalizar números, realizar operações aritméticas de soma e subtração, arredondar e imprimir as parcelas e resultado de acordo com especificações de máquina pré-determinadas.

Assim que é iniciado, o programa executa a função *main* e exibe um menu perguntando se o usuário pretende digitar o nome de um arquivo de entrada ou fechar o programa. Essa função foi implementada com laços do tipo *while* e funções *scanf*. Para realizar os comandos do menu foram utilizadas as funções *if* e *scanf* (para identificar a escolha do usuário), *fopen* (para abrir o arquivo “entrada.txt”) e *fscanf* (para identificar as especificações de máquina, o número de operações a serem realizadas, as parcelas das operações e o sinal característico da operação desejada). Logo após, com o auxílio de funções *for*, os números da mantissa normalizada (que estavam armazenados em uma variável do tipo *int*) são copiados para um vetor tipo *int*, de modo a permitir as operações algarismo-algarismo, posteriormente.

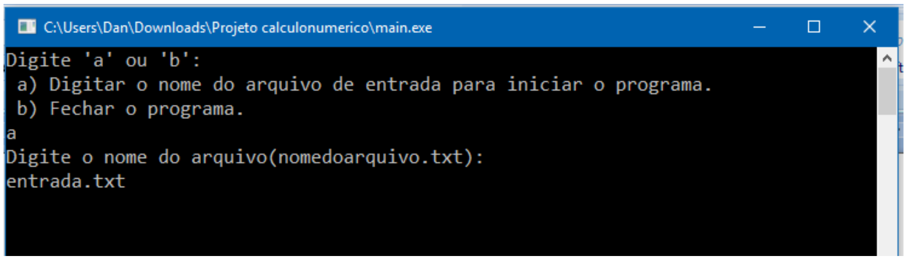

Figura 2: Menu inicial da calculadora

Então, o programa chama uma função *imprimirNum("")*, que arredonda a primeira e segunda parcela de uma operação (para se adequar às especificações de máquina) através de uma série de relações condicionais *if* e laços *for*, para serem impressos. Posteriormente, é verificado se as parcelas são números de máquina (com uso de comparações do expoente da parcela e o $e_{máx}/e_{min}$); caso contrário, a operação não é realizada e é indicado a qual região o número pertence (se de *underflow* ou *overflow*).

Caso ambas as parcelas sejam número de máquina, são comparados seus expoentes. Se um deles for menor, usam-se funções *if* e *for* para renormalizar o número com menor expoente, de modo que a operação aritmética possa ser realizada com o menor erro possível dentro das especificações da máquina. Feita a renormalização, o programa chama a função *soma("")*.

A função *soma("")* verifica se – e qual – das parcelas normalizadas é maior (através de funções *if-else*), além de comparar os sinais de ambas as parcelas e do sinal da operação desejada, modificando os sinais dos valores de cada algarismo do vetor, quando necessário, com o uso de funções condicionais *if*. Em seguida, é realizada a soma algarismo-algarismo, a partir da última posição dos vetores contendo as parcelas da operação, cujo resultado é armazenado em outro vetor tipo *int* de mesmo tamanho. Com uma série de *if* e *for* são feitos ajustes nos valores armazenados no vetor *resultado[]*, levando em conta os sinais dos algarismos e a operação realizada. Após a operação, é realizada a renormalização, e é verificado se o resultado é número de máquina (através de funções *if*). Em caso afirmativo, o programa imprime o resultado como *underflow* ou *overflow*. É chamada novamente a função *imprimirNum("")*, que arredonda o resultado, caso necessário, e o imprime na tela.

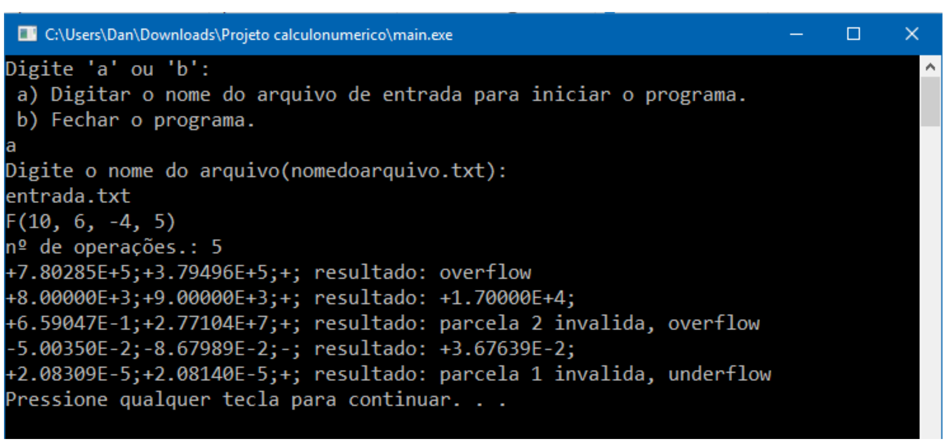

Figura 3- Calculadora realiza várias operações

As operações acima são realizadas *n* vezes (sendo *n* o número desejado de operações informado em "*entrada.txt*"), o arquivo é fechado (com a função *fclose*) e o programa é encerrado.

F. Exemplos

Sejam as variáveis x_1 e x_2 parcelas das operações; x a soma direta das parcelas; x_1', x_2' e x' os números de máquina (arredondados) e δx_1, δx_2 e δx os erros relativos associados a esses números, respectivamente. Todos os valores dos números de máquina presentes nos exemplos a seguir foram obtidos com uso da calculadora elaborada para o projeto.

Exemplo 1: Considere a máquina F (10, 4, -6, 6), $x_1 = 0.509381 \times 10^2$ e $x_2 = 7.583524 \times 10^4$, e seja efetuada uma adição. Então:
$x_1' = 5.094 \times 10^1 \rightarrow \delta x_1 = 7{,}48 \times 10^{-4}$
$x_2' = 7.583 \times 10^4 \rightarrow \delta x_2 = 6{,}91 \times 10^{-5}$
$x = 7{,}58861781 \times 10^4 \rightarrow x' = 7.589 \times 10^4 \rightarrow \delta x = 7{,}90 \times 10^{-5}$

Exemplo 2: Considere a máquina F (10, 5, -2, 4), $x_1 = 2.786453 \times 10^2$ e $x_2 = 9.574629 \times 10^4$, e seja efetuada uma adição. Então:
$x_1' = 2.7865 \times 10^4 \rightarrow \delta x_1 = 1{,}68 \times 10^{-5}$
$x_2' = 9.5746 \times 10^4 \rightarrow \delta x_2 = 6{,}91 \times 10^{-5}$
$x = 1{,}2361082 \times 10^5 \rightarrow x'$ = overflow $\rightarrow \delta x$ = ---

Exemplo 3: Considere a máquina F (10, 3, -4, 5), $x_1 = 2.786453 \times 10^2$ e $x_2 = 9.574629 \times 10^4$, e seja efetuada uma adição. Então:
$x_1' = 7.80 \times 10^3 \rightarrow \delta x_1 = 3{,}65 \times 10^{-4}$
$x_2' = 0.79 \times 10^{-2} \rightarrow \delta x_2 = 4{,}96 \times 10^{-5}$
$x = 7{,}8023839 \times 10^3 \rightarrow x' = 7.80 \times 10^3 \rightarrow \delta x = 6{,}27 \times 10^{-3}$

Através dos resultados obtidos é possível identificar que ao realizar o arredondamento das parcelas é gerado um erro relativo e que esses e erros são propagados nas operações de soma e/ou subtração, interferindo na precisão do resultado final.

3. Limitações

Durante a elaboração do projeto, descobriu-se que a etapa mais complexa seria implementar uma função para operações algarismo-algarismo, uma vez que seria necessário transformar os dados de uma variável tipo *int* em dados de um vetor tipo *int*. Além disso, foi preciso cuidar para que ao efetuar as somas dos algarismos, nenhum deles se perdesse e que fossem obedecidas as regras aritméticas (o que exigiu uma série de laços e funções condicionais).

4. Conclusões

Tendo em vista os dados apresentados neste relatório, é possível concluir que, geralmente, não há como evitar a geração de erros ao se efetuar operações em máquinas, devido às suas limitações físicas e erros associados às próprias operações. Entretanto, tais eles podem ser minimizados com o aumento de números significativos representados; cálculos mais precisos exigem máquinas mais potentes no que se refere ao armazenamento de dados e processamento.

Referências

[l] AMORIM, Guilherme. Aula 3 – Arredondamento e Operações. Disponível em: <http://www.cin.ufpe.br/ if215/slides/2014-1/>.

[2] DOS SANTOS, José Dias; DA SILVA, Zanoni Carvalho. Métodos Numéricos. 3ª ed. Pernambuco: Editora Universitária UFPE, 2010.

[3] PILING, Sergio. I – Representação dos números, aritmética de ponto flutuante e erros em máquinas digitais. Disponível em: <http://www1.univap.br/spilling/CN/CN_Capt1.pdf>.