

MAC0210 - Exercício Programa 1  
Nome: Eduardo Hashimoto  
n USP: 6514136

## Parte 1 - Aritmética de Ponto Flutuante

Foram considerados dois números de entrada no padrão ieee, dentro do intervalo normalizado.

Os números são representados como strings.

Tentei seguir a descrição do algoritmo apresentado no livro Numerical Computing with IEEE Floating Point Arithmetic.

Iguala os expoentes dos números, alterando a mantissa e o expoente do menor. Dentro desse passo, observa se algum bit ultrapassa a casa 25 da mantissa, o que liga o “sticky bit”.

Se o “sticky bit” for ligado, então ele é adicionado a casa 26 da mantissa.

O valor do expoente do resultado é imediato, pois é igual ao expoente de ambos os números (que foram igualados).

O valor da mantissa do resultado vem da soma das mantissas, considerando as casas de 1 a 26, caso tenham os “guard” e “sticky bits”.

Então, procede-se ao arredondamento, que é um parâmetro de entrada. Por falta de tempo foi considerado apenas o arredondamento para cima, que procura se há um bit igual 1 após a 23 casa. Se houver a própria casa 23 da mantissa passa a ser 1.

No arredondamento para baixo não é preciso fazer nada, pois ele é como um truncamento.

O mesmo vale para o arredondamento para zero, caso o número seja positivo.

Após o arredondamento é feita normalização da mantissa.

Com os valores dos sinais, expoentes e mantissa o resultado é composto.

No primeiro exemplo, nem os guard bits nem o sticky bit são usados. Assim, estes nem chegam a possuir um valor.

Por esse motivo, independentemente do tipo de arredondamento usado, o resultado é sempre o desejado.

No segundo exemplo, no entanto, um guard bit toma o valor de 1, enquanto o outro e o sticky bit tem valor 0.

Por esse motivo, caso o arredondamento seja para baixo ou em direção a zero, o resultado será diferente do esperado (será o próprio 1).

O algoritmo da subtração é inteiramente análogo ao algoritmo da soma e não foi apresentado por falta de tempo.

## Parte 2 - Método de Newton

Primeiramente foi implementado o método de Newton para encontrar a raiz de uma função  $f$ , dado sua derivada,  $df$ , ponto de partida, tolerância e número máximo de iterações.

A função e a derivada entram como expressões que são lidas pela função inline.

Em seguida calcula-se o primeiro ponto a partir do ponto fornecido e seu erro absoluto através das expressões abaixo:

$$x(1) = x(0) - f(x(0))/df(x(0)) \text{ e } ex(1) = \text{abs}(x(1)-x(0))$$

Inicia-se então um loop, que tem como condições de parada um erro menor que a tolerância fornecida ou um número de iterações maior que o número máximo de iterações fornecidas.

Assim, há duas situações de saída: ou uma solução foi encontrada (pois o erro é menor que a tolerância) ou não foi encontrada nenhuma solução (e então o número de iterações é maior que o número máximo fornecido).

Se for o segundo caso, então a função retorna o caractere 'a' (já que 'a' não vai ser raiz de uma função). Se for o primeiro, então retorna a raiz da função (arredondada o que pode dar problemas, se a função cruzar o eixo x "freneticamente", mas tive que fazer uma solução rapidamente).

A função newton\_basis então usa a função do método de newton sobre diferentes pontos de partida para observar qual raiz da função é encontrada. Cada um desses pontos de partida é um ponto de uma matriz p1 x p2, fornecidos, e tem um valor que identifica a raiz que o método encontrou.



