**PERGUNTA FINAL:**

Nós precisamos descobrir a quantidade de carbono que vai para cada compartimento, a partir da quantidade de carbono total disponível no timestep atual (**t)**, sendo:

Carbono total disponível →. Esta variável é um input para o módulo, já que é calculada em outra parte do modelo

Sapwood → St

Heartwood → Ht

Leaves → Lt

Roots → Rt

A quantidade de carbono em um compartimento Xt é dada pela quantidade de carbono no timestep anterior Xt-1 somado ao incremento de carbono neste mesmo compartimento Xinct:

Xt = Xt-1 + Xinct

sendo o incremento de carbono a diferença entre a quantidade de carbono no timestep atual menos a quantidade de carbono no timestep anterior.

Xinct = Xt + Xt-1

A quantidade de carbono no timestep anterior (Xt-1) é conhecida, sendo calculada pelo próprio módulo no timestep anterior.

Portanto, nós precisamos descobrir qual é o incremento de carbono em cada compartimento Xinct.

A partir de derivações matemáticas de fórmulas presentes na literatura que determinam as relações alométricas entre os compartimentos (isto é, relações de tamanho, por exemplo: a quantidade de carbono nas raízes deve ser proporcional à quantidade de carbono nas folhas), determinou-se que o primeiro cálculo a ser feito será o de incremento de carbono nas folhas Linct. Para isso utilizaremos o método matemático *bisection method.* As relações alométricas acima citadas são dadas pelas variáveis **tau1, tau2, tau3 e SS,** as quais são utilizadas dentro da bisection method (as equações utilizadas para o cálculo destas variáveis estão descritas mais abaixo no documento).

os valores 0 e 10 foram definidos por nós e serão usados na formulação deste método.

Tendo o valor de Linct , calculamos o valor de incremento para raízes Rinct:

sendo que ltor é uma constante de valor conhecido.

A partir do Rinct e do Linct podemos calcular o incremento no sapwood Sinct:

Assim, sabendo todos os incrementos nós podemos calcular a quantidade de carbono atual em todos os compartimentos. Estas variáveis devem ser atualizadas para que no próximo timestep elas sejam equivalentes a Xt-1.

Na *bisection method* as variáveis **a** e **b** são inputs determinados a partir da chamada da função, no nosso caso **a=0 e b=10**. Estas variáveis serão usadas dentro de uma iteração e, assim, devem ser atualizadas dentro da própria função, já que o fortran não admite a atribuição de novos valores a uma varíavel de valor anteriormente definido. O resultado desta função será o **Linct** que ao final da iteração terá o mesmo valor de **midpoint** (variável interna da função). A bisection method utiliza outra função, a função **f** que depende das variáveis alométricas (descritas abaixo).

**Bisection method:**

se **f(a) \* f(b) > 0** então

esta função não tem resolução e retorna o valor de -2

se não **b** e **a** entrarão em um looping de iteração que só se encerrará quando a tolerância (**tol =** 0.001**)** for atingida de modo que:

enquanto **(b-a)/2** for maior que **tol:**

**midpoint = (a+b)/2**

se **f(midpoint) = 0** então

**midpoint =0** (a função se encerra já que 0 < 0.001)

porém se **f(a)\*f(midpoint)<0** então

**b=midpoint** (retorna para o início do looping)

se nenhuma das condições anteriores forem atingidas então

**a=midpoint** (retorna para o início do looping)

quando o looping for encerrado teremos o valor de **midpoint**.

Para retornarem para o início do looping **a** e **b** recebem um novo valor **(midpoint).** Isto não é aceito pelo fortran e não sabemos como resolver.

**Função f** (o x dentro dessa equação será o valor input para a função quando a mesma é chamada dentro de **bisection method**, ou seja, será **a,** ou **b** ou **midpoint):**

**f(x)=tau1\*((SS – x – x)/ltor + Ht-1****) - (((SS – x – x)/ltor)/(Lt-1****+ x) \* tau3)\*\*tau2)**

**ltor** é uma constante de valor conhecido e as outras variáveis provêm das equações alométricas:

**tau1 = ((k2\*\*(2/k3))\*4 / 3.14159 / dw**

**k2** e **k3** são constantes de valor conhecido e **dw** é uma variável input (é um dos atributos funcionais variantes e seu valor virá a partir da determinação dos PLSs)

**tau2 = (1 + 2)/k3**

**tau3 = klatosa/dw/SLA**

**klatosa** é uma constante e **SLA** é uma variável input (é um dos atributos funcionais variantes e seu valor virá a partir da determinação dos PLSs)

**SS** = **(St-1 + Ctotalt - Lt-1 )**/**ltor + Rt-1**

**ltor** é uma constante de valor conhecido

Após esses cálculos a quantidade de carbono em todos os compartimentos to timestep atual (Xt) deverá ser o input para a próxima iteração e se tornará Xt-1