



Laboratório Nacional de Computação Científica

GA - 018 - Métodos Numéricos

Exercício de Implementação

2

Aluno: Vinícius Prata Klôh

02 de agosto de 2018

1 Modelo Estudado e Resolução Numérica

O modelo estudado é dado pela Equação 1

$$\frac{dN}{dt} = rN * (1 - \frac{N}{K}) * (\frac{N}{A} - 1) \quad (1)$$

onde os termos dN , dt (também definido por h), r , N , K e A representam, respectivamente, a variação do tamanho da população estudada, o tamanho dos passos de tempo, a taxa intrínseca de crescimento, o tamanho da população, a capacidade de suporte e a intensidade do efeito Allee ¹.

1.1 Problema

Verificar o comportamento do crescimento populacional utilizando o método Euler Implícito para resolver numericamente o modelo da Equação 1.

Os valores adotados para os parâmetros do modelo são apresentado na Seção 2 pela Tabela 2.1. O domínio computacional foi definido por t_0 e t_{max} , onde $t_0 = 0$ e $t_{max} = 600$.

1.2 Resolução Numérica

O método de Euler é um procedimento numérico para resolver problema de EDO (Equações Diferenciais Ordinárias), com valor inicial dado, e é baseado na expansão em série de Taylor de primeira ordem.

A Resolução Numérica foi realizada seguindo os passos:

Aproximando $N(t)$ por um polinômio linear de Taylor (Equação 2):

$$N(t) = N(t_i) + \frac{dN}{dt} \Big|_{t_i} (t - t_i) + \frac{1}{2} \frac{d^2N}{dt^2} (\xi(t))(t - t_i)^2 \quad (2)$$

Aproximando $\frac{dN}{dt} \Big|_{t_i}$ (Equação 3):

$$\frac{dN}{dt} \Big|_{t_i} = \frac{N(t) - (N(t_i))}{(t - t_i)} - \frac{1}{2} \frac{d^2N}{dt^2} (\xi(t))(t - t_i) \quad (3)$$

Definindo a partição do domínio computacional por h e uniforme por simplicidade com $N(t_i)$, em que $t_i = ih, i = 1$.

Substituindo no modelo da Equação 1, temos a Equação 4:

$$\frac{N_{i+1} - N_i}{h} = \left(\frac{r(N_{i+1})^2}{A} - rN_{i+1} \right) + \left(\frac{r(N_{i+1})^3}{kA} - \frac{r(N_{i+1})^2}{k} \right) \quad (4)$$

Para realizar a linearização dos termos quadráticos e cúbico na Equação 4, foi adotado um novo índice sobrecrito k , que definem as etapas do processo iterativo para as aproximações destes termos.

¹O efeito Allee influencia na capacidade reprodutiva de uma população de acordo com a sua densidade.

Linearizando os termos $(N_{i+1})^2$ e $(N_{i+1})^3$ através do método de Picard, obtemos $(N_{i+1})^2 \approx (N_{i+1}^k)(N_{i+1}^{k+1})$ e $(N_{i+1})^3 \approx (N_{i+1}^k)^2(N_{i+1}^{k+1})$.

Os termos não lineares foram linearizados também utilizando o método de Newton, com expansão em série de Taylor. Assim, obtemos $(N_{i+1})^2 \approx 2(N_{i+1}^k)(N_{i+1}^{k+1}) - (N_{i+1}^k)^2$ e $(N_{i+1})^3 \approx 3(N_{i+1}^k)^2(N_{i+1}^{k+1}) - 2(N_{i+1}^k)^3$.

Dessa forma, dado $N_i = N_{i+1}^k$, é calculado N_{i+1}^{k+1} , conforme a Equação 5 (usando *Picard*) ou a Equação 6 (usando *Newton*), para cada passo de tempo do processo evolutivo.

$$N_{i+1}^{k+1} = \frac{Ni}{1 - hrN_{i+1}^k(\frac{1}{A} + \frac{1}{k}) + hr + \frac{hr}{kA}(N_{i+1}^k)^2} \quad (5)$$

$$N_{i+1}^{k+1} = \frac{Ni - hr(N_{i+1}^k)^2(\frac{1}{A} + \frac{1}{k}) + \frac{2hr}{kA}k(N_{i+1}^k)^3}{1 - hr[1 - 2(N_{i+1}^k)(\frac{1}{A} + \frac{1}{k}) - 3\frac{(N_{i+1}^k)^2}{kA}]} \quad (6)$$

Obervando o termo remanescente da Equação 3 percebemos que o erro para o processo evolutivo é de $o(h)$. Pelo termo remanescente $\frac{1}{2} \frac{d^2(N_{i+1}^k)}{dt^2}(\xi(t))((N_{i+1}^{k+1}) - (N_{i+1}^k))^2$ da aproximação do processo iterativo, utilizando a Equação 6, percebemos que o erro é de $o(h)^2$. A taxa de convergência do processo iterativo foi calculada pela Equação 7.

$$p = \log \frac{|(N_{i+1}^{k+1}) - (N_{i+1}^k)|}{|(N_{i+1}^k)(N_{i+1}^{k-1})|} \quad (7)$$

2 Experimentos e Resultados

2.1 Metodologia

Os experimentos numéricos apresentados neste relatório foram realizados utilizando a Equação 5 para determinar a evolução da população ao longo do tempo. Lembrando que para cada passo de tempo foi realizado o processo iterativo para aproximação dos termos não lineares. O domínio computacional considerado para o estudo foi de 0 até 600 dias ($t_0 = 0$ e $t_{max} = 600$). Os demais parâmetros são apresentados na Tabela 2.1.

Os valores adotados para dt foram escolhidos de forma que o domínio computacional fosse dividido em partições de valores inteiros, conforme $\frac{t_{max}}{2}, \frac{t_{max}}{4} \dots$

| | Caso 1 ($A < N < K$) | Caso 2 ($K < N$) | Caso 3 ($N < A$) |
|--|------------------------|--------------------|--------------------|
| K | 5000 | 1000 | 5000 |
| N | 300 | 5000 | 1000 |
| A | 200 | 200 | 2000 |
| r | 0,01/dia | | |
| Tolerância ($N_{i+1}^{k+1} - N_{i+1}^k$) | 0,001 | | |
| Max iterações | 10 | | |

Tabela 1: Casos estudados e os parâmetros adotados.

2.2 Resultados obtidos

A Figura 1 apresenta os resultados obtidos nos experimentos utilizando as diferentes configurações de parâmetros apresentados na Tabela 2.1. Para ambos os casos adotados estudados neste trabalho, o comportamento da evolução populacional seguiu as considerações sobre o modelo. Para o Caso 1 podemos perceber que, visto que a população é maior que seu limite crítico definido pelo efeito Alee, e menor do que a capacidade de suporte, há um crescimento da mesma até que K é atingido. Para o Caso 2, como a população inicial N_0 é maior do que a capacidade de suporte, mesmo com $A < N$, ela decresce até atingir capacidade de suporte. Para o último caso, como a população é menor do que seu limite crítico ($N < A$), ocorre a mortalidade por conta do efeito Alee.

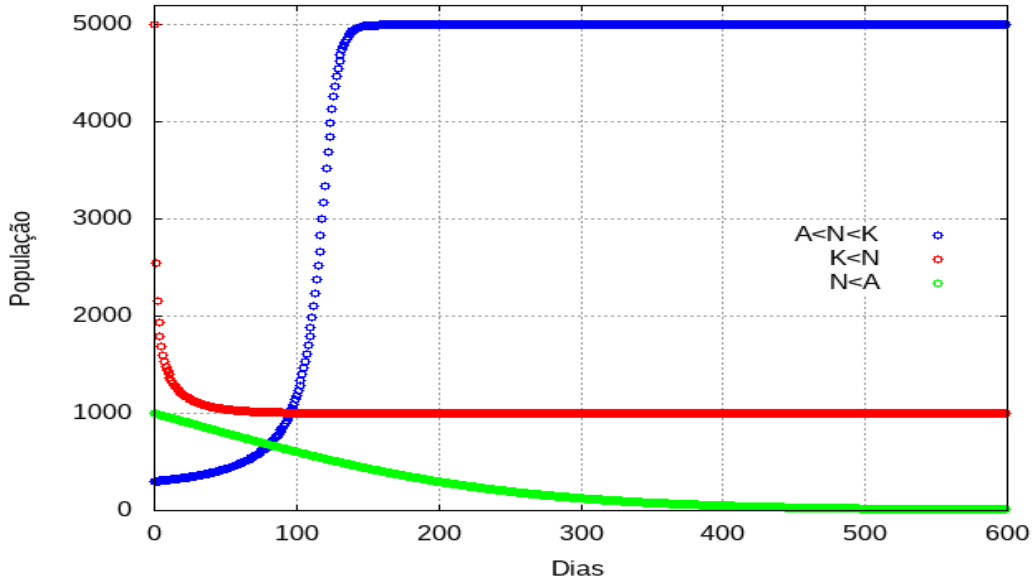


Figura 1: Evolução da população ao longo do tempo.

A taxa de convergência da Equação 7, pode ser calculada para diferentes passos de tempo. Foi escolhido o passo de tempo 70, por arbitrariedade e por permitir o cálculo utilizando 4 aproximações consecutivas obtidas nesta etapa do processo iterativo. O valor obtido foi $p \approx 2,1$, o que caracteriza convergência quadrática do método.

Por limitação do número de páginas e por similaridade entre os resultados utilizando as Equações 5 e 6, neste relatório estão contidos apenas os resultados utilizando o método de *Picard*. Os arquivos inclusos na realização do trabalho como um todo encontram-se disponíveis em https://github.com/ViniciusPrataKloh/Metodos_numericos_modelo_2.

3 Conclusão

Neste trabalho foi estudado o modelo de crescimento logístico com influência do efeito Alee. Para resolução numérica do modelo foram necessárias construções de aproximações para os termos não lineares, resultantes da utilização do método de Euler Implícito. O processo iterativo, utilizado em cada passo de tempo do processo evolutivo apresentou convergência consideravelmente rápida, visto que o número maior de iterações foi 4.

Tendo em vista as considerações sobre o modelo, podemos concluir também que os experimentos numéricos foram compatíveis com a lógica do crescimento envolvida no modelo.

Apêndice

Neste apêndice são apresentados os algoritmos construídos para execução computacional dos experimentos. A Figura 2 contém o algoritmo para o processo evolutivo e a Figura 3, para o processo iterativo.

```
int time_loop(double* n_i, const double r, const double K, const double A, const int tmax,
              const int dt, const double tol, const int iter_max){

    double kn_i, kn_ii;
    int count;
    double aux = (r * dt);

    for(count = 0; count < tmax/dt; count++){
        kn_i = n_i[count];
        printf("\nTempo= %d\n", count);

        /*
         * call iteration_loop function
         */
        if(!iteration_loop(n_i[count], &kn_i, aux, K, A, dt, tol, iter_max))
            printf("\n WARNING: The method did not converge with %d iterations. \n", iter_max);

        /*
         * Updates the approximation of the time step with the value obtained in the iterative process
         */
        n_i[count+1] = kn_i;
    }

    pFile = fopen("output_1.dat", "w");
    if(!pFile){
        printf("\n FILE ERROR: Could not open the file. \n");
        return 0;
    }
    printf(pFile, "N \n");

    for(count = 0; count < tmax/dt; count++){
        fprintf(pFile, "%d \t %.12lf\n", (count * dt), n_i[count]);
    }
    fclose(pFile);
    return 1;
}
```

Figura 2: algoritmo para o processo evolutivo do crescimento populacional.

```
int iteration_loop(double n_i, double* kn_i, const double aux, const double K, const double A,
                  const int dt, const double tol, const int iter_max){

    double kn_ii;
    double b = *kn_i; // Auxiliar
    int count = 0;

    do{
        /*
         * Equation for the iterative process
         */
        kn_ii = n_i / ( 1 - (aux*b)*(1/A + 1/K) + aux + (aux/(K*A))*(b*b) );
        if(abs((kn_ii - b) < tol)){
            *kn_i = kn_ii;
            return 1;
        }
        *kn_i = kn_ii;
        b = *kn_i;
        count++;
    } while(count <= 20);

    return 0;
}
```

Figura 3: Algoritmo para o processo iterativo.