Relatório de Métodos Numéricos Parte 1

UFPE - Centro de Informática - Engenharia da Computação

Aluno: Flávio Lins da Mota Nascimento (flmn@cin.ufpe.br)

Professor: Ricardo Martins

Monitor Chefe: Victor Mellia

**Objetivo**

Implementar um programa que receba um arquivo de entrada, calcule os métodos especificados nele e gere um arquivo de saída com as respectivas respostas.

**Funcionalidades**

O programa deve calcular os seguintes métodos:

* Euler
* Euler Inverso
* Euler Aprimorado
* Runge-Kutta
* Adam-Bashforth
* Adam-Multon
* Fórmula Inversa

**Arquivos**

O projeto possui os seguintes arquivos:

* Programa Principal (p.py)
* Entradas Testes (entrada.txt)
* Requisitos Monitoria (requirements.txt)
* Executor dos Requisitos (RUNME)

**Linguagem de Programação**

Foi utilizada a linguagem Python, como sugerida pelo professor e pelos monitores, para botar em prática esse projeto. Python facilita a manipulação de funções com uso da biblioteca *sympy* versão 1.1.1. Com a biblioteca é possível converter uma *string* numa expressão matemática (função *parse\_expr*) e escolher valores a serem aplicados no lugar das variáveis da função (função *subs*). E a versão do interpretador Python utilizada nas compilações foi a 3.6.5, podendo também funcionar nas versões 3.5 em diante.

**Código**

Lista de funções cobradas:

* euler
* euler\_inverso
* euler\_aprimorado
* runge\_kutta
* adam\_bashforth
* adam\_bashforth\_by\_euler
* adam\_bashforth\_by\_euler\_inverso
* adam\_bashforth\_by\_euler\_aprimorado
* adam\_bashforth\_by\_runge\_kutta
* adam\_multon
* adam\_multon\_by\_euler
* adam\_multon\_by\_euler\_inverso
* adam\_multon\_by\_euler\_aprimorado
* adam\_multon\_by\_runge\_kutta
* formula\_inversa
* formula\_inversa\_by\_euler
* formula\_inversa\_by\_euler\_inverso
* formula\_inversa\_by\_euler\_aprimorado
* formula\_inversa\_by\_runge\_kutta

Lista de funções complementares:

* aplica\_fyt
* calc\_eul
* euler\_retorno\_n\_primeiros
* calc\_eul\_inv
* euler\_inverso\_retorno\_n\_primeiros
* calc\_eul\_apri
* euler\_apri\_retorno\_n\_primeiros
* calc\_runge
* runge\_retorno\_n\_primeiros
* runge\_kutta
* prep\_param\_to\_bashforth
* adam\_bashforth\_execute\_out
* aplica\_bf2\_fyt
* aplica\_bf3\_fyt
* aplica\_bf4\_fyt
* aplica\_bf5\_fyt
* aplica\_bf6\_fyt
* aplica\_bf7\_fyt
* aplica\_bf8\_fyt
* prep\_param\_to\_multon\_and\_form\_inv
* adam\_multon\_execute\_out
* aplica\_mt2\_fyt
* aplica\_mt3\_fyt
* aplica\_mt4\_fyt
* aplica\_mt5\_fyt
* aplica\_mt6\_fyt
* aplica\_mt7\_fyt
* aplica\_mt8\_fyt
* form\_inv\_execute\_out
* aplica\_fi2\_fyt
* aplica\_fi3\_fyt
* aplica\_fi4\_fyt
* aplica\_fi5\_fyt
* aplica\_fi6\_fyt
* what\_method

**Lógica**

Ao receber uma entrada qualquer, primeiro se identifica o método e se passa a linha de entrada como parâmetro. Para cada método existe uma função com o mesmo nome do método listada pela monitoria, onde os parâmetros são separados e passados para fazer cálculo dos valores de alguns “y(t)” em casos mais simples. Alguns modos precisam também de um número exato de valores “y(t)”s anteriores para iniciar outro método posteriormente, então nesse caso são calculados os “n” primeiros e em seguida a entrada é preparada em uma nova forma com os valores iniciais. Continuando os casos é necessário que sejam aplicados valores numa função específica “f(t,y)” que também pode depender da ordem solicitada. Para finalmente serem enviados os valores para as funções que executam a impressão na tela o resultado correto.

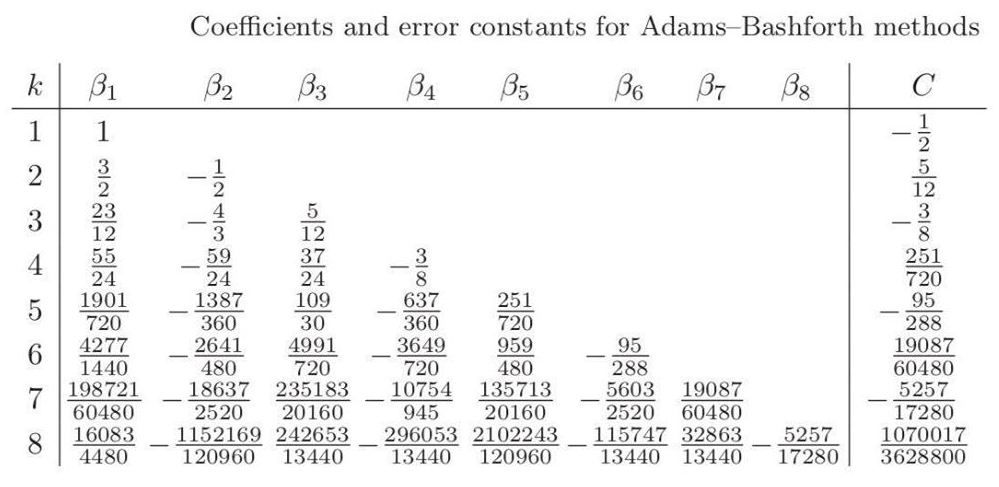
**Projeto na Íntegra**

https://github.com/Flavio13Lins/Numerical-Methods

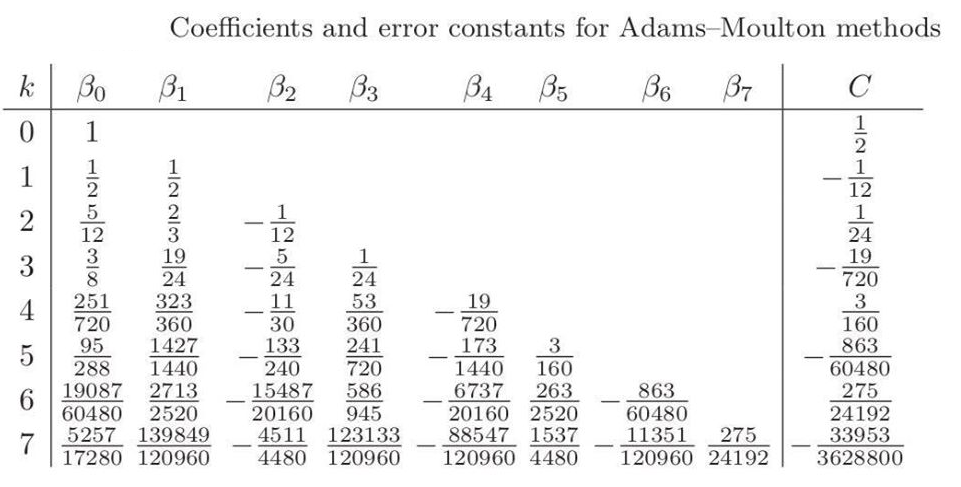
**Material de Apoio**

Para Métodos como de Adam-Bashforth, Adam-Multon e Fórmula Inversa, foram usadas tabelas de apoio para conseguir valores exatos de coeficientes em funções dependendo da ordem desejado do cálculo. Ou seja, o cálculo é pré-definido, não foi utilizado a teoria do cálculo de sistemas de equações lineares algébricas por matrizes. As tabelas utilizadas são as seguintes:

* **Adam-Bashforth**



* **Adam-Multon**



* **Fórmula Inversa:**

