

**Ano Letivo:** 2022/2023

**Atividade 05-** **Métodos Numéricos para Derivação e Integração**

**Relatório**

Análise Matemática II

**Docente:** Arménio Correia

**Trabalho realizado por:**

Martim Alexandre Vieira Antunes

**nº:** 2022141890 **Curso:** LEI

Pedro Lino Neves Faneca

**nº:** 2022134142 **Curso:** LEI

**Índice**

1.Introdução-------------------------------------------------------------------------------------------------2

2.Métodos Numéricos para Derivação---------------------------------------------------------------3

2.1-Diferenças Finitas em 2 pontos----------------------------------------------------------4

2.1.1-Progressivas----------------------------------------------------------------------4

2.1.2-Regressivas-----------------------------------------------------------------------5

2.2-Diferenças Finitas em 3 pontos----------------------------------------------------------6

2.2.1-Progressivas----------------------------------------------------------------------6

2.2.2-Regressivas-----------------------------------------------------------------------7

2.2.3-Centradas-------------------------------------------------------------------------8

2.2.4-2ªDerivada------------------------------------------------------------------------9

3.Métodos Numéricos para Integração-------------------------------------------------------------10

3.1-Regra dos Trapézios----------------------------------------------------------------------11

3.2-Regra de Simpson-------------------------------------------------------------------------12

4.Conclusão------------------------------------------------------------------------------------------------13

5.Bibliografia----------------------------------------------------------------------------------------------14

6.Autoavaliação e heteroavaliação------------------------------------------------------------------15

**1.Introdução**

Este trabalho surge do âmbito da unidade curricular de Análise Matemática 2, do curso de Engenharia Informática do Instituto Superior de Engenharia de Coimbra.

O principal objetivo deste trabalho é a implementação de funções, através do desenvolvimento de uma app em Matlab, para algumas fórmulas de Derivação e Integração Numérica, nomeadamente: Diferenças finitas em 2 pontos (Progressivas e Regressivas) e 3 pontos (Progressivas, Regressivas e Centradas), 2ª derivada e também regra dos Trapézios e regra de Simpson.

**2.Métodos Numéricos para Derivação**

Existem vários métodos numéricos para a derivação, que são utilizados para calcular aproximações das derivadas de funções. Esses métodos são úteis quando não é possível calcular a derivada analiticamente ou quando se deseja obter uma estimativa numérica precisa da derivada.

O método mais simples para aproximar a derivada é usar o método de diferenças finitas. O método das diferenças finitas é um método de resolução de equações diferenciais que se baseia na aproximação de derivadas por diferenças finitas.

**2.1- Diferenças Finitas em 2 pontos**

As diferenças finitas em 2 pontos são um conjunto de métodos para estimar derivadas de uma função utilizando os valores da função em dois pontos consecutivos.

**2.1.1- Progressivas**

Este método utiliza a diferença entre os valores da função em dois pontos consecutivos para estimar a derivada.

**Fórmula:**

![Uma imagem com relógio

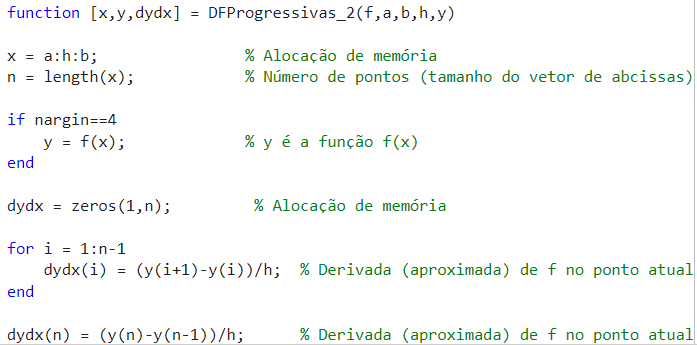
Descrição gerada automaticamente](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4RDSRXhpZgAATU0AKgAAAAgABAE7AAIAAAAEQklOAIdpAAQAAAABAAAISpydAAEAAAAIAAAQwuocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFkAMAAgAAABQAABCYkAQAAgAAABQAABCskpEAAgAAAAM3NQAAkpIAAgAAAAM3NQAA6hwABwAACAwAAAiMAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAyMDowNToyOCAyMDo1NDoxMQAyMDIwOjA1OjI4IDIwOjU0OjExAAAAQgBJAE4AAAD/4QsWaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLwA8P3hwYWNrZXQgYmVnaW49J++7vycgaWQ9J1c1TTBNcENlaGlIenJlU3pOVGN6a2M5ZCc/Pg0KPHg6eG1wbWV0YSB4bWxuczp4PSJhZG9iZTpuczptZXRhLyI+PHJkZjpSREYgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyIvPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6eG1wPSJodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvIj48eG1wOkNyZWF0ZURhdGU+MjAyMC0wNS0yOFQyMDo1NDoxMS43NDk8L3htcDpDcmVhdGVEYXRlPjwvcmRmOkRlc2NyaXB0aW9uPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIj48ZGM6Y3JlYXRvcj48cmRmOlNlcSB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6bGk+QklOPC9yZGY6bGk+PC9yZGY6U2VxPg0KCQkJPC9kYzpjcmVhdG9yPjwvcmRmOkRlc2NyaXB0aW9uPjwvcmRmOlJERj48L3g6eG1wbWV0YT4NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgPD94cGFja2V0IGVuZD0ndyc/Pv/bAEMABwUFBgUEBwYFBggHBwgKEQsKCQkKFQ8QDBEYFRoZGBUYFxseJyEbHSUdFxgiLiIlKCkrLCsaIC8zLyoyJyorKv/bAEMBBwgICgkKFAsLFCocGBwqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKv/AABEIADcAqwMBIgACEQEDEQH/xAAfAAABBQEBAQEBAQAAAAAAAAAAAQIDBAUGBwgJCgv/xAC1EAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADAQEBAQEBAQEBAAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQIDEQQFITEGEkFRB2FxEyIygQgUQpGhscEJIzNS8BVictEKFiQ04SXxFxgZGiYnKCkqNTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/APpGiiigAooooAKKKKACiiigAorG8W6zJ4f8J6jqcEDzPbwO6hCvynacMdxHAOOmT7Gua0kala2+gaLeX2qNdanI93eNfTq0qJGilkRkJ2qXKcZzgt60LVg9DvqK53WfGdpot0sEmn6jdM7vGhtoVbe6JvZVBYE4XvjHbOeK2IdStJp4rcTol1LCJ1tnYLKE45KdcAkA+9G4bFqiiormBbq1lgZ5EEilC0TlGXI6hhyD7igCWisTwlqc+qeH0e9cPdW80tpO4GN7xOULY7Z25/GtugAoorzyLW7258Wa/rN4dVtdN8PxFFtlmiEDssO9xIoYlm+YEY4GBk54pXsOzex6HRXO2Goy6R4Ws7rUhcX15cx/aJliwfmK72xuIVUUcDJHQDqaLHxvpl3oVzrE0F9ZWNuqO0tzat86OoYMoXcWGCOnTvVNWbXYlO6TXU6Kiuf8UXsulLp2rQyuscd3FBcRbjtkilYIcr0yrMrA9Rgjua6Cl0GYXiTxHPos1haadpkmqX987iK2SZYvlQZdizccZH51DZeJZfsWu6pqaRxabYXLxWxjU75FjAVyecEmTcoxjpWbPpVhq3xgE01jbTf2dpYdneFWPmySfISSOqrEcemau+P49ng92jULFFd200wUcbFuEZz+WSaFsvP8Nbfp+Idbdv8AK/6nRWkk01nFJdQiCZ0DPEH3bCe2cDOKmrg/iNG+oRf2dBpGo3csscey7igMkVupmXeRjkSBQTkDOMY61V1u6g8NaXoUGkW15YQS6pJcND57xnyl3MS/PyRsxTIbhQ3IzxRe+v8AW4bf15HayasIPEMOl3EWwXULSW0wbIkZPvoRjggEEdcjPTFaNcfqlxcahrHguR7f7NdvcyTywLKsnlKLdw43DggFlGR6itvVY7J9X0hrvTri6nSdzbzxRlltm2HLOQflBGRznk0AatFYT6ZemRiLbIJOP+J3cj9AvFamnwyQWgSZNjZJx9pef/x5wDQBzfjn7VqsVt4atNOvJV1GWNprxIwYIYlkVnDtnglQcDHOateIbdrXX9E1sLmCzeW3uSB/q45VA3/QMqZ9ASegNWp7XxO1xIbbV9JjhLkxpJpUrsq54BYXABOO+Bn0FR/ZPFn/AEG9F/8ABPL/APJNC0ApXXhPVLrxEmqt4hyITMLWJ7NT9nWVQpCkMMkbeCQepyDmsrV49dHxEW4sUuobRIYbJbpLTzjkkyvweNjAKhfPDY9DXRfZPFn/AEG9F/8ABPL/APJNH2TxZ/0G9F/8E8v/AMk0LS3kD1uFtNfHxpfxui+ULC2IXzmKgmSfJA29eBn6Cs3wy0Wlah4qmltZrO0N958Ze1eJXQQRglcqAfmDdOprS+yeLP8AoN6L/wCCeX/5Jo+yeLP+g3ov/gnl/wDkmj/K35f5Der+78rB4O02fT/Do+2xmO5u55ruWJusZlkL7T7gED8Kvf2DpH/QKsf/AAGT/CseW416DUIbCfxR4djvJwTFbvpkgkkA6lVN1k9D09KtfZPFn/Qb0X/wTy//ACTQI0L+5TRtGluLexnuEto8ra2UQZ2H91F4/Kuc0HQJ7zwPq0GpRPa3GvSXU8kcow8QlyEDDsQm3I7Vp/ZPFn/Qb0X/AME8v/yTR9k8Wf8AQb0X/wAE8v8A8k0mrp366DTta3Qy/wCzNS8T+D7OxTVJNIljga01GBYFcs3l7GXnBGPvKR1BB5FJrnhrV5/B8ejWV2k0kk8CNIsQSOCGPaeELHOdgBGTkt6dNX7J4s/6Dei/+CeX/wCSaPsniz/oN6L/AOCeX/5JpvVt9xLRJdjCm06/vPDul6Jerd/bLu9jurpbm5E7QRxyCR/nHG3coRR15Hvjuqwvsniz/oN6L/4J5f8A5JragEq28YuXSSYIBI8aFFZsckKSSBntk49TT/r+vuAz9F8OaT4eS4XRrJLUXUpmmKksXc9ySSfw6Cr9xbxXVtJb3MaywyoUkjcZDKRggj0qSil0sHW5DZ2wsrOK2SSSRYlCK0rbmwOmT3479fWm3On2d6c3lpBcHY0f72IN8rY3LyOhwMjvgVYoo3DYpw6VawagLyNMSLCIIlwAsMY52oB0BIGfXA9BVyiigAooooAKKKKACiiigAooooA898WX9ynjo3lrLbxxeHNHmvJGuI2kUPLkKMBl5xGe/f3rttJu5b/R7S6uI1inlhVpY1bIjfHzL+ByPwqC68OaTerqIurQSf2mqLd5dv3gUYUdflx7Y9etXbOzgsLRLa0TZEmcDJJ5OSSTySSSSTySaForf11/z/AHq7/10/r5k1FFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB/9k=)

**Legenda:**

* f’(xk) = Aproximação do valor da derivada no ponto de abcissa xk;
* f(xk+1) = Valor da função na próxima abcissa;
* f(xk) = Valor da função no ponto de abcissa atual;
* h = Valor de cada subintervalo (passo).

**Algoritmo:**

* Alocar memória para **x**;
* Definir o número de pontos **(n)**;
* Se forem recebidos 4 elementos, y recebe o valor de **f(x)**;
* Alocar memória para a derivada;
* Para **i** de **1** a **n-1**, calcular a derivada (aproximada) de f no ponto atual, para a iésima iteração;
* Calcular a derivada (aproximada) de f no ponto atual, em n.

**Função (MATLAB):**

**2.1.2- Regressivas**

Este método também utiliza a diferença entre os valores da função em dois pontos consecutivos, mas agora os pontos são deslocados para trás em relação ao ponto de interesse.

**Fórmula:**

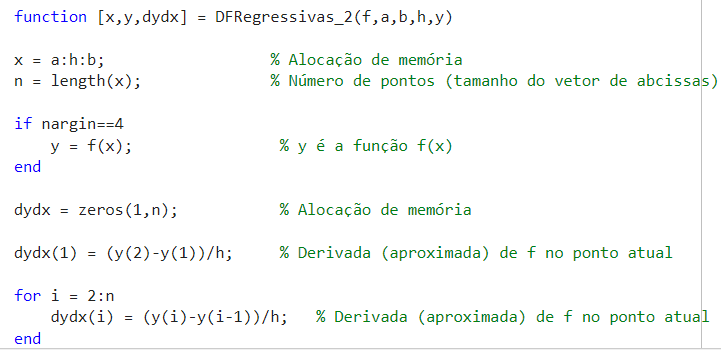


**Legenda:**

* *f’(xk)*= Aproximação do valor da derivada no ponto de abcissa ***xk***;
* *f(xk)*= Valor da função no ponto de abcissa atual;
* *f(xk-1)*= Valor da função na abcissa anterior;
* *h* = Valor de cada sub-intervalo (passo).

**Algoritmo:**

* Alocar memória para ***x***;
* Definir o número de pontos (***n***);
* Se forem inseridos 4 elementos, ***y*** recebe o valor de ***f(x)***;
* Alocar memória para a derivada;
* Calcular a derivada (aproximada) de f no ponto atual, em ***1***;
* Para ***i*** de ***2*** a ***n***, calcular a derivada (aproximada) de f no ponto atual, para a iésima iteração.

**Função (MATLAB):**

**2.2-Diferenças Finitas em 3 pontos**

As diferenças finitas em 3 pontos são um conjunto de métodos para estimar derivadas de uma função utilizando os valores da função em três pontos consecutivos.

**2.2.1-Progressivas**

**Fórmula:**

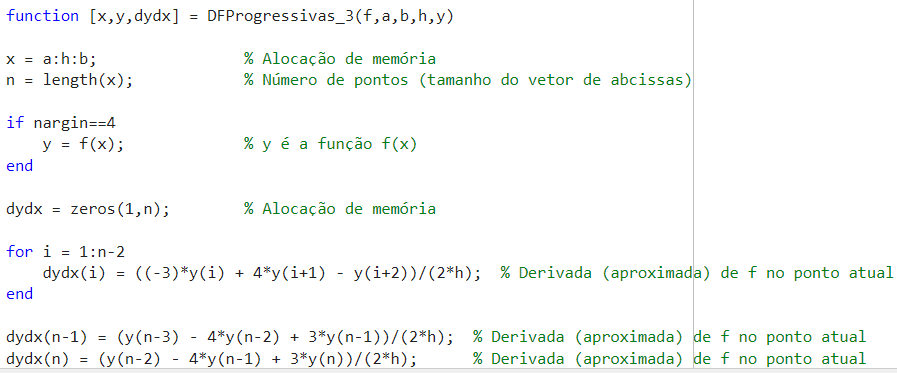


**Legenda:**

* *f’(xk)*= Aproximação do valor da derivada no ponto de abcissa ***xk***;
* *f(xk)*= Valor da função no ponto de abcissa atual;
* *f(xk+1)*= Valor da função na próxima abcissa;
* *f(xk+2)*=Valor da função 2 abcissas à frente;
* *h* = Valor de cada subintervalo (passo).

**Algoritmo:**

* Alocar memória para ***x***;
* Definir o número de pontos (***n***);
* Se forem inseridos 4 elementos, ***y*** recebe o valor de ***f(x)***;
* Alocar memória para a derivada;
* Para ***i*** de ***1*** a ***n-2***, calcular a derivada (aproximada) de f no ponto atual, para a iésima iteração;
* Calcular a derivada (aproximada) de f no ponto atual, em ***n-1***;
* Calcular a derivada (aproximada) de f no ponto atual, em ***n***.

**Função (MATLAB):**

**2.2.2-Regressivas**

**Fórmula:**

![Uma imagem com faca

Descrição gerada automaticamente](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAYABgAAD/4RDSRXhpZgAATU0AKgAAAAgABAE7AAIAAAAEQklOAIdpAAQAAAABAAAISpydAAEAAAAIAAAQwuocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAFkAMAAgAAABQAABCYkAQAAgAAABQAABCskpEAAgAAAAM3MgAAkpIAAgAAAAM3MgAA6hwABwAACAwAAAiMAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAyMDowNToyOCAyMzowODoyNQAyMDIwOjA1OjI4IDIzOjA4OjI1AAAAQgBJAE4AAAD/4QsWaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLwA8P3hwYWNrZXQgYmVnaW49J++7vycgaWQ9J1c1TTBNcENlaGlIenJlU3pOVGN6a2M5ZCc/Pg0KPHg6eG1wbWV0YSB4bWxuczp4PSJhZG9iZTpuczptZXRhLyI+PHJkZjpSREYgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyIvPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6eG1wPSJodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvIj48eG1wOkNyZWF0ZURhdGU+MjAyMC0wNS0yOFQyMzowODoyNS43MTc8L3htcDpDcmVhdGVEYXRlPjwvcmRmOkRlc2NyaXB0aW9uPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIj48ZGM6Y3JlYXRvcj48cmRmOlNlcSB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6bGk+QklOPC9yZGY6bGk+PC9yZGY6U2VxPg0KCQkJPC9kYzpjcmVhdG9yPjwvcmRmOkRlc2NyaXB0aW9uPjwvcmRmOlJERj48L3g6eG1wbWV0YT4NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgPD94cGFja2V0IGVuZD0ndyc/Pv/bAEMABwUFBgUEBwYFBggHBwgKEQsKCQkKFQ8QDBEYFRoZGBUYFxseJyEbHSUdFxgiLiIlKCkrLCsaIC8zLyoyJyorKv/bAEMBBwgICgkKFAsLFCocGBwqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKv/AABEIADgBHwMBIgACEQEDEQH/xAAfAAABBQEBAQEBAQAAAAAAAAAAAQIDBAUGBwgJCgv/xAC1EAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADAQEBAQEBAQEBAAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQIDEQQFITEGEkFRB2FxEyIygQgUQpGhscEJIzNS8BVictEKFiQ04SXxFxgZGiYnKCkqNTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/APpGiiigAooooAKKKKACiiigAoooJwMmgAormtO8eaTqWn6jfQxXkdrYAEzSwFVnBztMR/izjgdeR610FtJLLaxyXEPkSsoLRbt2w+me9AEtFFFABRRVC51VbPWLWyuYyiXgKwT7uGkAJMZHY7QSPXDdMDIBfooooAKKKy9A8RWHiW0uLrSnaSCC5e2MhAw7JjJXnleeDQBqUVhw+LLC48WS+H4IbuS4hU+ZOsOYUYKrFC/Y4ZT0xzjOau6Tqi6vBJc28ZFr5jJDKT/rgpwWA7LkHB7jn0oAv0UUUAFFFUdY1MaPp5vpojJbRMDcMp5ij7vjuB1I9MnnGCAXqKRWDKGUggjII700zRiZYTIolZSyoWG4gYBIHoMj8xQA+iorq6hsrOa6upBHBBG0kjnoqgZJ/IViaV4y07VtTt7GOKeCW7tftlqZdhW4izjcpRm6ccHB56UdbB0udBRWHr/iyw8O3dna3UN3cT3jfJHaQ+YyruVS5H90F1Hc89Kuxaotxrk+n28e8WsatcS7sCNm5VAO5xyemAV9aAL9FFIxYIdgBbHAJwCfrQAtFUdI1WPV7IzIjRSRyNDPC5+aKRThlP8AQ9wQe9XqACmSzxQbPPlSPzHCJvYDcx6AZ6n2rD8b68fDng/UL+ISfaFgYW5SFpAJCMLnAIAzjk4FYuj+F7q6OhT3up32o2tncS3rnUlZZRNsCIgV1DbVy7ZPfGODQtWD0R3NFc1pt8l5rWuateybbXS5TaQ5ztjCIGlfHqS2Pog96a3juyTVhp0um6klz5kMbr5Kt5Yl+4zbWOAcN7jacgcZFrbz/XYO509FUrfVLK+1C906Fy9xZhBcRtEwC7wSvJGDkA9Cay/DM7wahrGiSOzpp06G3LEkiGRAyrk9dp3KPYCgDoaKoyaqkWsGxktbpUW2Nw140eLdQDjaXzw3fHpzSf29o/8A0FbH/wACU/xoAv1heNrx9P8ABGrXkVzJbPBau6yRBd2QOB8wI5OB0zzxWxb3VvdxebaTxzx5xvicMM/UVjeIvDk/iGa2il1SSDTFYNdWKwqwudrqy5c8qMrggdQTSkrq3ccXZ3Odm0iXQ/BPhfS57mWdF1GxSUShBsAIOwbVHG9R1yfepviDqwtrW4gsdYkh1XbbpaWsMjIUklm2CRsH5x/snj5Twc11Ou6SutaPLZmTyZMrJDKBnypEYMjY74YA470NpFlqEcc2s6XYzXbRBJS0Syj3UMwyVyT1/Kqbu3fvf8F/kSlZL0t+JymsXc3hnwlfGx8Q3U9zLqcdvFcXGJPs7O6KUG8HcFG49+cjrXX6TqQ1Wx+0C0u7TDsnl3kPlv8AKcZx6Hsaq6n4Y03VLaztp4vLtbORnjt4QqRklGTBGOgDkjGOal0zQ7XTNMgsi0l4IV2LLdkO+OwzgDAHGABSXn/X9fqN9LFLxVq2qaRZSXOnQ24ghtpZ5bi4G5QygbI9oZWy2SMjOMdOaoeKrqa48LaLfPbva3bajYSCBzlomaVAyn6BmBrV1jw2uqm28nUbzTFt3MgWyWEK7Y4Zg8bAkdvTr1xRc6JJdXekxzTGSy00ibMjbpJ5VUqhY4xgZLZ7tjpjkW6v3X4O4Pb5M1Lhp1izaRxyyZ+7LIUGPqFP8qqibV882Nlj/r9f/wCNVfooAxfE326WxitLKxuLqK4k2XRtpI1dIsZIG91+993g5AJPXFc58L5Gl0rX4UtZrJBrF1sfMZC5b7oAZuV+mPTNd3JGk0TRyoro4KsrDIYHqCKwbzwpDFp0kHhRrXw5NM6tNPZ2MR8xRnKkYHXPXqKWqv5/8D/Ie9vL/gnN+DobtPBGu65HqFzcPqEl5cRLIsYDYJVH+VAckIvfHsK1La607T/Anh2K61M2Nu1pEQqOUecLDuKhlIK9NxI9Md66PStKttI0S10u1U/ZraFYVDckgDHPue9UNJ8Ow2ukRaXqNvbXttYyYsjKgcrGPuZBHDKDtyOoAPfAbW6Xl+F/8xLo35/ic14LluIdBh1vV9fvpZZdJN1JZySF0ijLFkky2cMEAX3wxOTk1seA/EFzreg2o1BLuW6+yxzyXkluEhmL5OEYYB29Dgds85zWpd+G9MuNDvdKgto7G2vkKTfY41jJBGCeBjOOM4pul+GrHSZbswGWSK6lMv2eVg0cTHrsXHGTyc5p31fb/h/8/wAAe39eRo3cssNnLLAiSSIpYK7FQfxAP8qydN1D/hIPAUGoXsaR/b9PEssa/dXemSOe3NacthC9vLFAPsxkXaZIUUMB+II/MVkHw3JaeDE8OabeSGHyRa/aLgr5kcONpxtUAtt4HTk5PTFRJXi13/4P/AKi7NMXwbNcy/D3RJWAkuG06E4kbaGOwdTg4/KoJ5dV/wCEzsSbKz3/AGC4wPtj4xvh7+V9K6K3gjtbaK3gQJFEgRFHRVAwBTWtIXvo7xkzPHG0Stk8KxUkY6dVH5VpJ3nzev6mcVaNitf3sVjoN1ea5HHHbwwu9wqZmXYAc8bQW47Yri7WS1tPihojpCLiPUdKdNPWGAwLYQrhuYzz8wIGTjBGMV6GQGUhhkHgg96zodCsLC3uRotna6ZPOhXz7a2RSDjgkAYOCc4NTs7/ANbP/P8Aq5W6t/XQ5izs59Q+L+p3I1K68vTLKGHYFixmVjI0f3M7cKh67uetTaHdLa6L4pvbu8SxYapdGS6cBhCFwqsR3woXitnw3oD6Fb3LXd82o395L511dvGIzIwUKPlHAAVRx9aWDQxBq2pMUhm07VAJJ4JBnEoUIeCMFWUDI7Fe+7gtbRdvzaYXvq+6/KxyOkb9T8UX8sHiLUINOstTgt7WMXDy/aXEQaVDuzkHIz2AU4xkmtXRfFM1z4v1exf7Ze263/2SAw26tHa7IgzGRwBgFiwGc9B689JbaNpdlcCez020t5guwSRQKrBfTIGccdKy9O8GWGlaomoWdxeCdfN8wGUBZy8jSfPgc4Zzj9c4oWjXb/hv8vxB7ef/AA5F4eynjTxXGn+q8+3kx6OYFDfoq101ZehaVJpsNzNdsj3t9O1xcsn3QxAAUZ6hVVVB74zxmtSjogOK8a3sGsahYeErYvJdz3lvPdR+U21bZWMjMWxtwfL29epFdrRRQtFYHuclpcN9az+KtM094oNQe5a8s3mUlCJUG1iO4Dq4P0pNC0HxBYazf3t/PYl9RnilnmhZmYJGgURBSgGCRndn+I4HeuqaCFrhJ2iQzIpRZCo3KpxkA+hwPyFSULS3pb7gev8AXc860W11U+OLi7voNWt4L6/lniNupSMrH+5VZwR90qgdeR14BzXQ+HUNz4l8Raqo/cTTx2sTdnEKYYj/AIGzL/wE10h560yCCK2gSG3jSKKNQqIihVUegA6ULRJf1/VgeruVJF1R9YKo9qmmG2I3KG+0LNngjOV249RnPtSf2ddf9Bm+/wC+IP8A43V+igCK3heGLZJcSXDZzvlCg/T5QB+lZ174V8PalePd6joOmXdzJjfNPZxu7YGBliMngAfhWtRQBhf8IN4T/wChX0X/AMF8X/xNH/CDeE/+hX0X/wAF8X/xNbtFAGF/wg3hP/oV9F/8F8X/AMTR/wAIN4T/AOhX0X/wXxf/ABNbtFAGF/wg3hP/AKFfRf8AwXxf/E0f8IN4T/6FfRf/AAXxf/E1u0UAYX/CDeE/+hX0X/wXxf8AxNH/AAg3hP8A6FfRf/BfF/8AE1u0UAYX/CDeE/8AoV9F/wDBfF/8TR/wg3hP/oV9F/8ABfF/8TW7RQBgt4J8IopZ/DGiqqjJJ0+LAH/fNYFnD4Du9XtLB/CFhatqEbS2E0+mQCO7UDJ2EZIOOcMFOD0rupoo54JIplDRyKVdT3BGCK87t4NDvfEXh+HS7q3tdI0KV47KJJjJJdTFSMDksI1Ab5j94+2CRb6/1/Wlge1zT0zR/Buqa3qmlx+DNPhm0xkWV5dOt9j7xldpXPbnkAjIrX/4Qbwn/wBCvov/AIL4v/ia8+tLl4tLi16O/uba78Qa+06MJ2VEtY2JYlBww8mI9Qeor1yN1kjV0ztYAjIxwfahaq/9bJg9JW/rexif8IN4T/6FfRf/AAXxf/E0f8IN4T/6FfRf/BfF/wDE1u0UAYX/AAg3hP8A6FfRf/BfF/8AE0f8IN4T/wChX0X/AMF8X/xNbtFAGF/wg3hP/oV9F/8ABfF/8TR/wg3hP/oV9F/8F8X/AMTW7RQBhf8ACDeE/wDoV9F/8F8X/wATR/wg3hP/AKFfRf8AwXxf/E1u0UAYX/CDeE/+hX0X/wAF8X/xNH/CDeE/+hX0X/wXxf8AxNbtFAGbp3hzQ9HuGuNJ0bT7GZk2NJbWqRsVyDjKgHGQOPYVpUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFADXRJY2jlVXRgVZWGQQeoIrKtPCfhywulubHQNLtp0ztlhso0ZcjBwQMjgkUUUAW30fTJLW3tX060a3tSDBEYFKREDA2jGFwPSrlFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAH//Z)

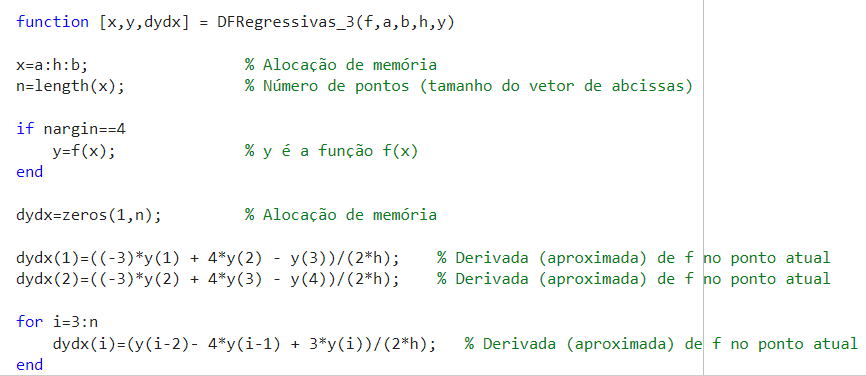
**Legenda:**

* *f’(xk)*= Aproximação do valor da derivada no ponto de abcissa ***xk***;
* *f(xk-2)*= Valor da função 2 abcissas atrás;
* *f(xk-1)*= Valor da função na abcissa anterior;
* *f(xk)*= Valor da função no ponto de abcissa atual;
* *h* = Valor de cada subintervalo (passo).

**Algoritmo:**

* Alocar memória para ***x***;
* Definir o número de pontos (***n***);
* Se forem inseridos 4 elementos, ***y*** recebe o valor de ***f(x)***;
* Alocar memória para a derivada;
* Calcular a derivada (aproximada) de f no ponto atual, em ***1***;
* Calcular a derivada (aproximada) de f no ponto atual, em ***2***;
* Para ***i*** de ***3*** a ***n***, calcular a derivada (aproximada) de f no ponto atual, para a iésima iteração.

**Função (MATLAB):**

****

**2.2.3-Centradas**

**2.2.3-Centradas**

Este método utiliza a diferença simétrica entre os valores da função em pontos adjacentes para obter uma aproximação mais precisa da derivada.

**Fórmula:**

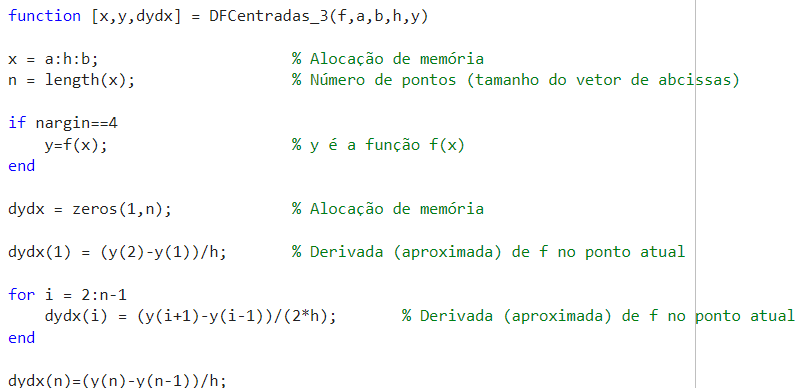


**Legenda:**

* *f’(xk)*= Aproximação do valor da derivada no ponto de abcissa ***xk***;
* *f(xk+1)*= Valor da função na próxima abcissa;
* *f(xk-1)*= Valor da função na abcissa anterior;
* *h* = Valor de cada subintervalo (passo).

**Algoritmo:**

* Alocar memória para ***x***;
* Definir o número de pontos (***n***);
* Se forem inseridos 4 elementos, ***y*** recebe o valor de ***f(x)***;
* Alocar memória para a derivada;
* Calcular a derivada (aproximada) de f no ponto atual, em ***1***.
* Para ***i*** de ***2*** a ***n-1***, calcular a derivada (aproximada) de f no ponto atual, para a iésima iteração;
* Calcular a derivada (aproximada) de f no ponto atual, em ***n***.

**Função (MATLAB):**

**2.3-2ªDerivada**

**Fórmula:**



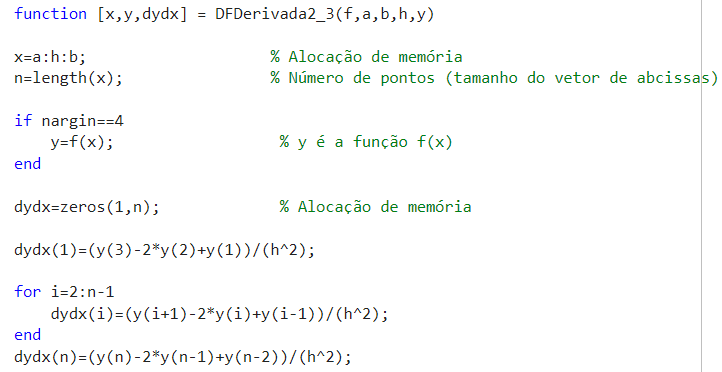
**Legenda:**

* *f’’(xk)*= Aproximação do valor da 2ª derivada no ponto de abcissa ***xk***;
* *f(xk+1)*= Valor da função na próxima abcissa;
* *f(xk)*= Valor da função no ponto de abcissa atual;
* *f(xk-1)*= Valor da função na abcissa anterior;
* *h* = Valor de cada subintervalo (passo).

**Algoritmo:**

* Alocar memória para x;
* Definir o número de pontos (n);
* Se forem inseridos 4 elementos, y recebe o valor de f(x);
* Alocar memória para a derivada;
* Calcular a derivada (aproximada) de f no ponto atual, em 1.
* Para i de 2 a n-1, calcular a derivada (aproximada) de f no ponto atual, para a iésima iteração;
* Calcular a derivada (aproximada) de f no ponto atual, em n.

**Função (MATLAB):**

****

**3.Métodos Numéricos para Integração**

Integração numérica é uma técnica utilizada para calcular uma aproximação numérica de uma integral definida de uma função. O integral definido é uma operação matemática que envolve o cálculo da área sob a curva de uma função em um intervalo específico.

A necessidade de técnicas de integração numérica surge quando não é possível encontrar uma solução analítica para a integral, ou quando a função é muito complexa para ser integrada de forma exata.

Existem vários métodos de integração numérica, cada um com suas próprias características e aplicabilidades. Alguns dos métodos mais comuns incluem:

* Regra do Trapézio: Este método divide o intervalo de integração em vários segmentos de igual comprimento e aproxima a área sob a curva em cada segmento por um trapézio. A soma das áreas dos trapézios fornece uma estimativa da integral.
* Regra de Simpson: Este método também divide o intervalo de integração em segmentos, mas em vez de utilizar trapézios, utiliza polinómios de grau 2 para aproximar a curva em cada segmento. A soma das áreas dos polinómios de grau 2 fornece uma estimativa mais precisa do integral.
* Quadratura de Gauss: Este método utiliza uma abordagem baseada em pontos de integração específicos e pesos correspondentes. Esses pontos e pesos são escolhidos de forma a obter uma estimativa precisa do integral para uma determinada classe de funções.
* Regra de Monte Carlo: Este método utiliza amostragem aleatória para estimar o integral. São gerados pontos aleatórios dentro do intervalo de integração, e a média dos valores da função nesses pontos é utilizada para estimar o integral.

Esses são apenas alguns exemplos de métodos de integração numérica. A escolha do método mais adequado depende da função a ser integrada, das características do problema e do nível de precisão desejado. A integração numérica desempenha um papel importante em várias áreas, como física, engenharia, economia, entre outras, onde a integração analítica pode ser difícil ou impossível de ser realizada.



**3.1-Regra dos Trapézios**

A regra dos trapézios é um método numérico para calcular uma aproximação da integral definida de uma função. Ela divide o intervalo de integração em vários trapézios e estima a área sob a curva da função utilizando a soma das áreas desses trapézios. Quanto mais subintervalos existirem mais precisa se torna a aproximação.

**Fórmula:**



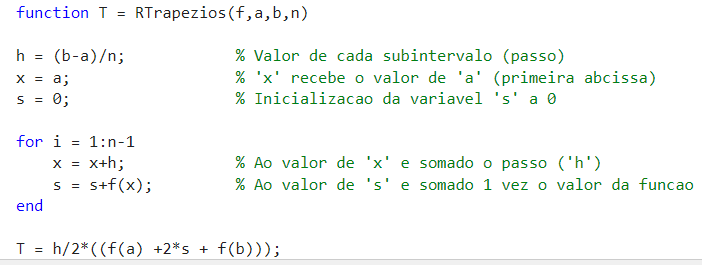
**Legenda:**

* *IT (f)* = Cálculo da Regra dos Trapézios;
* *f(x0)*= Valor da função na abcissa ***x0***;
* *f(x1)*= Valor da função na abcissa ***x1***;
* *f(xn-1)*= Valor da função na abcisa ***xn-1***;
* *f(xn)*= Valor da função na abcissa ***xn***;
* *h* = Valor de cada subintervalo (passo).

**Algoritmo:**

* Calcular o passo (**h**);
* Atribuir o valor de **a** a **x** ;
* Inicializar **s** com o valor **0**;
* Para **i** de **1** a **n-1**:
  + Somar o passo (**h**) a **x**;
  + Somar o valor da função em x (**f(x)**) a **s**.
* Cálculo da Regra dos Trapézios.

**Função (MATLAB):**

****

**3.2-Regra de Simpson**

A regra de Simpson é um método numérico para calcular uma aproximação do integral definido de uma função. Esta regra utiliza uma fórmula de interpolação polinomial para aproximar a curva da função por meio de segmentos de parábolas. O integral sob cada parábola é calculado e somado para obter a aproximação do integral total.

**Fórmula:**

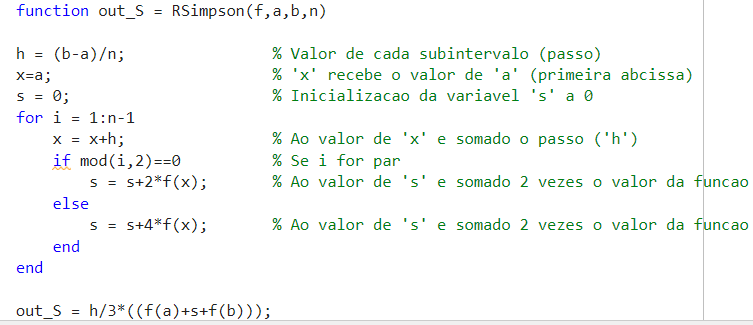


**Legenda:**

* + - *IS (f)* 🡪 Cálculo da Regra de Simpson;
    - *h* 🡪 Valor de cada subintervalo (passo);
    - *f(x0)*🡪 Valor da função na abcissa ***x0***;
    - *f(x1)*🡪 Valor da função na abcissa ***x1***;
    - *f(xn-1)*🡪 Valor da função na abcissa ***xn-1***;
    - *f(xn)*🡪 Valor da função na abcissa ***xn***;
    - *n* 🡪 Número de subintervalos.

**Algoritmo:**

* Cálculo do passo (**h**);
* Atribuir o valor de **a** a **x**;
* Inicializar **s** com o valor **0**;
* Para **i** de **1** a **n-1**:
  + Somar o passo (**h**) a **x**;
  + Se **i** for par, soma-se **2** vezes o valor de **f(x)** a **s**;
  + Se **i** for ímpar, soma-se 4 vezes o valor de **f(x)** a **s**;
* Calcular a Regra de Simpson.

**Função (MATLAB)**

**4.Conclusão**

Com este trabalho podemos concluir que os métodos numéricos têm diversas aplicações, quer para derivadas, quer para integrais, o que facilita a resolução de problemas mais complicados das diferentes áreas da matemática e ciências.

Como já visto anteriormente, verificamos que quanto maior for o número de subintervalos ***n***, menor é o erro dos métodos. Já com o tamanho de cada subintervalo, ***h*** , o efeito é o contrário: quanto menor o tamanho introduzido, menor o erro dos métodos.

Relativamente à comparação entre os vários métodos da integração numérica, verificamos que o que apresenta menor erro é, e consequentemente, melhor aproximação ao valor exato, é a **Regra de Simpson** comparando com a Regra dos Trapézios.

Em relação às fórmulas da derivação numérica, a comparação encontrada foi que a melhor aproximação ao valor real se origina a partir das fórmulas que utilizam **3 pontos**, comparativamente às que apenas utilizam 2 pontos.

**5.Bibliografia**

* Ficheiros de suporte disponibilizados pelo professor;
* Formulário da cadeira;

**6.Autoavalição e heteroavaliação**

Chegando ao fim do trabalho, estamos contentes pelo resultado final.

Pelo esforço e trabalho aplicados a esta atividade, achamos que merecemos numa escala de 0 a 5 valores, um 4.25.

A nível de grupo, não houve quaisquer problemas e ambos trabalhámos bem. O aluno Martim Antunes foi quem distribui as tarefas, explicou como secalhar ficava melhor e quem preocupava-se de sempre colocar um “dedinho” dele no final, mesmo não lhe tivesse sido atribuído aquela tarefa, aperfeiçoar ainda mais.

Por isso achamos que o aluno Martim Antunes mereça um 4.25 e o aluno Pedro Faneca um 3.75.