

PÓS GRADUAÇÃO – ENGENHARIA ELÉTRICA  
MESTRADO – IAAA  
INTELIGÊNCIA ARTIFICIAL APLICADA  
À AUTOMAÇÃO E ROBÓTICA

Programação Científica– PEL 216  
Prof. Dr. Reinaldo Augusto da Costa Bianchi

DESCIDA DE GRADIENTE  
“MÉTODOS NUMÉRICO”

FLÁVIO INFANTI  
Matrícula: 118.310-2

AULA 04 – 09/08/2019

## **1      OBJETO**

Dando continuidade ao aprendizado da disciplina de Programação Científica, com o objetivo de aprofundar-se nos estudos referentes à Programação Orientada a Objeto C++, iremos desenvolver o Algoritmo de Descida do Gradiente conforme apresentado durante aula realizada em 09/08/2019.

## **2      INTRODUÇÃO**

A maioria das tarefas em Machine Learning é na verdade solucionar problemas relacionados aos Métodos Numéricos, os quais contamos com duas técnicas de otimização que são: o Método de Newton-Raphson para o cálculo das raízes de uma função e o Algoritmo de Descida do Gradiente para encontrar o valor mínimo de uma função, seguindo a direção do gradiente.

### **2.1    Algoritmo de Descida de Gradiente**

A Descida do Gradiente é uma ferramenta padrão para otimizar funções complexas iterativamente dentro de um programa de computador. Para uma determinada função arbitrária, seu objetivo é encontrar seu mínimo que às vezes pode ser considerado um mínimo local ou global.

A derivada é um conceito de Cálculo e refere-se à inclinação da função em um determinado ponto. Precisamos conhecer a inclinação para que possamos conhecer a direção (sinal) para mover os valores dos coeficientes gerando desta forma a próxima iteração.

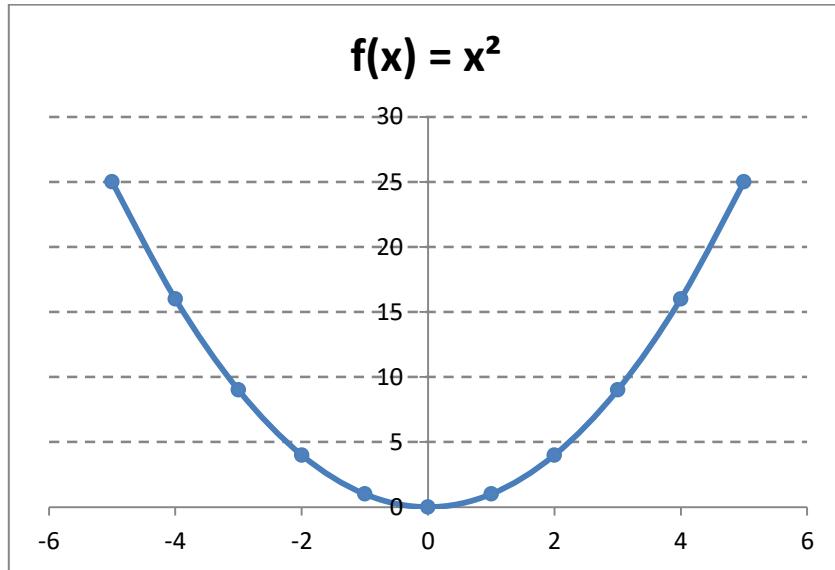
Verificando que a direção está em declive, podemos atualizar os valores dos coeficientes. Um parâmetro de taxa de aprendizagem ( $\beta$ ) deve ser especificado e controla o quanto os coeficientes podem mudar em cada atualização.

Este processo se repetirá até que o erro seja zero ou próximo o suficiente de zero, indicando que chegamos no mínimo da função.

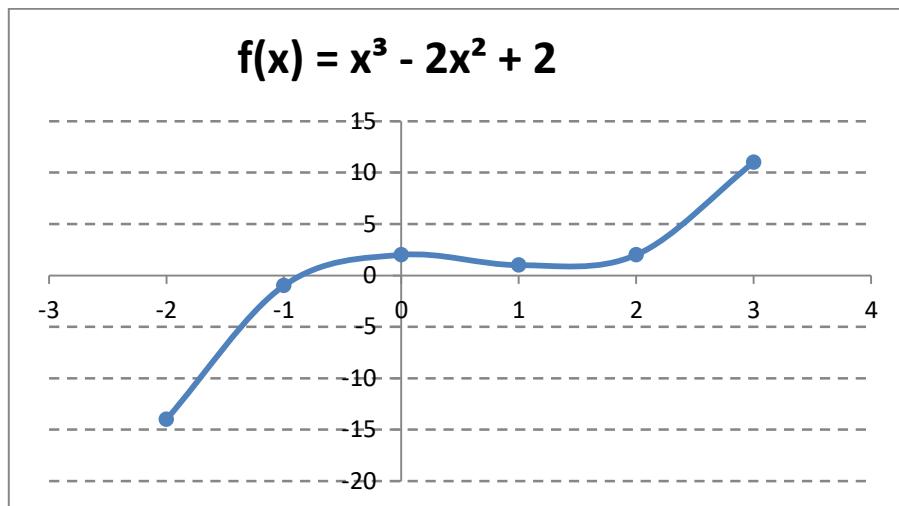
### 3 PROPOSTA DE IMPLEMENTAÇÃO.

Para implementação do algoritmo, partiu-se da lista de exercício fornecida em sala de aula com duas funções proposta para serem resolvidas e a ideia aplicada no desenvolvimento do programa foi gerar um “menu” onde pode ser escolhido a solução desejada, segue abaixo a descrição das duas funções:

- Opção 1 →  $f(x) = x^2$



- Opção 2 →  $f(x) = x^3 - 2x^2 + 2$



## 4 RESULTADOS

Segue abaixo, telas obtidas durante simulação realizada pelo programa desenvolvido:

<b>Entrada de Dados: <math>x_0 = 2</math>; Erro = 1; <math>\beta = 1</math></b>	
$f(x) = x^2$	$f(x) = x^3 - 2x^2 + 2$
<pre>==&gt; Numero de iteracao = 1 x0 .....: = 2.000000 x0 (novo) .: = 1.000000 Funcao ....: f(x) = 2.000000 ^ 2 = 4.000000 Derivada ..: f'(x) = 2 * 2.000000 = 4.000000  ==&gt; Numero de iteracao = 2 x0 .....: = 1.000000 x0 (novo) .: = 0.500000 Funcao ....: f(x) = 1.000000 ^ 2 = 1.000000 Derivada ..: f'(x) = 2 * 1.000000 = 2.000000  A solucao final eh: 0.500000  Tempo de execucao : 13.612 s</pre>	<pre>==&gt; Numero de iteracao = 1 x0 .....: = 2.000000 x0 (novo) .: = 1.500000 Funcao ....: f(x) = (2.000000 ^3) - (2 * 2.000000^2) + 2 = 2.000000 Derivada ..: f'(x) = (3 * 2.000000 ^2) - (4 * 2.000000) = 4.000000  A solucao final eh: 1.500000  Tempo de execucao : 14.195 s</pre>

<b>Entrada de Dados: <math>x_0 = 2</math>; Erro = 1; <math>\beta = 0.5</math></b>	
$f(x) = x^2$	$f(x) = x^3 - 2x^2 + 2$
<pre>==&gt; Numero de iteracao = 1 x0 .....: = 2.000000 x0 (novo) .: = 1.500000 Funcao ....: f(x) = 2.000000 ^ 2 = 4.000000 Derivada ..: f'(x) = 2 * 2.000000 = 4.000000  A solucao final eh: 1.500000  Tempo de execucao : 15.981 s</pre>	<pre>==&gt; Numero de iteracao = 1 x0 .....: = 2.000000 x0 (novo) .: = 1.750000 Funcao ....: f(x) = (2.000000 ^3) - (2 * 2.000000^2) + 2 = 2.000000 Derivada ..: f'(x) = (3 * 2.000000 ^2) - (4 * 2.000000) = 4.000000  A solucao final eh: 1.750000  Tempo de execucao : 12.084 s</pre>

<b>Entrada de Dados: <math>x_0 = 2</math>; Erro = 1; <math>\beta = 0.1</math></b>	
$f(x) = x^2$	$f(x) = x^3 - 2x^2 + 2$
<pre>==&gt; Numero de iteracao = 1 x0 .....: = 2.000000 x0 (novo) .: = 1.900000 Funcao ....: f(x) = 2.000000 ^ 2 = 4.000000 Derivada ..: f'(x) = 2 * 2.000000 = 4.000000  A solucao final eh: 1.900000  Tempo de execucao : 11.510 s</pre>	<pre>==&gt; Numero de iteracao = 1 x0 .....: = 2.000000 x0 (novo) .: = 1.950000 Funcao ....: f(x) = (2.000000 ^3) - (2 * 2.000000^2) + 2 = 2.000000 Derivada ..: f'(x) = (3 * 2.000000 ^2) - (4 * 2.000000) = 4.000000  A solucao final eh: 1.950000  Tempo de execucao : 9.902 s</pre>

**Entrada de Dados:  $x_0 = 2$ ; Erro = 0.001;  $\beta = 1$**

$f(x) = x^2$	$f(x) = x^3 - 2x^2 + 2$
<pre> ==&gt; Numero de iteracao = 1 x0 .....: = 2.000000 x0 (novo) ..: = 1.000000 Funcao ....: f(x) = 2.000000 ^ 2 = 4.000000 Derivada ...: f'(x) = 2 * 2.000000 = 4.000000  ==&gt; Numero de iteracao = 2 x0 .....: = 1.000000 x0 (novo) ..: = 0.500000 Funcao ....: f(x) = 1.000000 ^ 2 = 1.000000 Derivada ...: f'(x) = 2 * 1.000000 = 2.000000  ==&gt; Numero de iteracao = 3 x0 .....: = 0.500000 x0 (novo) ..: = 0.250000 Funcao ....: f(x) = 0.500000 ^ 2 = 0.250000 Derivada ...: f'(x) = 2 * 0.500000 = 1.000000  ==&gt; Numero de iteracao = 4 x0 .....: = 0.250000 x0 (novo) ..: = 0.125000 Funcao ....: f(x) = 0.250000 ^ 2 = 0.062500 Derivada ...: f'(x) = 2 * 0.250000 = 0.500000  . . .  ==&gt; Numero de iteracao = 7 x0 .....: = 0.031250 x0 (novo) ..: = 0.015625 Funcao ....: f(x) = 0.031250 ^ 2 = 0.000977 Derivada ...: f'(x) = 2 * 0.031250 = 0.062500  ==&gt; Numero de iteracao = 8 x0 .....: = 0.015625 x0 (novo) ..: = 0.007813 Funcao ....: f(x) = 0.015625 ^ 2 = 0.000244 Derivada ...: f'(x) = 2 * 0.015625 = 0.031250  ==&gt; Numero de iteracao = 9 x0 .....: = 0.007813 x0 (novo) ..: = 0.003906 Funcao ....: f(x) = 0.007813 ^ 2 = 0.000061 Derivada ...: f'(x) = 2 * 0.007813 = 0.015625  ==&gt; Numero de iteracao = 10 x0 .....: = 0.003906 x0 (novo) ..: = 0.001953 Funcao ....: f(x) = 0.003906 ^ 2 = 0.000015 Derivada ...: f'(x) = 2 * 0.003906 = 0.007813  ==&gt; Numero de iteracao = 11 x0 .....: = 0.001953 x0 (novo) ..: = 0.000977 Funcao ....: f(x) = 0.001953 ^ 2 = 0.000004 Derivada ...: f'(x) = 2 * 0.001953 = 0.003906 </pre> <p>A solucao final eh: 0.000977</p> <p>Tempo de execucao : 15.328 s</p>	<pre> ==&gt; Numero de iteracao = 1 x0 .....: = 2.000000 x0 (novo) ..: = 1.500000 Funcao ....: f(x) = (2.000000 ^ 3) - (2 * 2.000000^2) + 2 = 2.000000 Derivada ...: f'(x) = (3 * 2.000000 ^ 2) - (4 * 2.000000) = 4.000000  ==&gt; Numero de iteracao = 2 x0 .....: = 1.500000 x0 (novo) ..: = 0.333333 Funcao ....: f(x) = (1.500000 ^ 3) - (2 * 1.500000^2) + 2 = 0.875000 Derivada ...: f'(x) = (3 * 1.500000 ^ 2) - (4 * 1.500000) = 0.750000  ==&gt; Numero de iteracao = 3 x0 .....: = 0.333333 x0 (novo) ..: = 2.148148 Funcao ....: f(x) = (0.333333 ^ 3) - (2 * 0.333333^2) + 2 = 1.814815 Derivada ...: f'(x) = (3 * 0.333333 ^ 2) - (4 * 0.333333) = -1.000000  A solucao final eh: 2.148148  Tempo de execucao : 9.429 s </pre>

<b>Entrada de Dados:</b> $x_0 = 2$ ; Erro = 0.001; $\beta = 0.5$	
$f(x) = x^2$	$f(x) = x^3 - 2x^2 + 2$
<pre> ==&gt; Numero de iteracao = 1 x0 .....: = 2.000000 x0 (novo) .: = 1.500000 Funcao ....: f(x) = 2.000000 ^ 2 = 4.000000 Derivada ..: f'(x) = 2 * 2.000000 = 4.000000  ==&gt; Numero de iteracao = 2 x0 .....: = 1.500000 x0 (novo) .: = 1.125000 Funcao ....: f(x) = 1.500000 ^ 2 = 2.250000 Derivada ..: f'(x) = 2 * 1.500000 = 3.000000  ==&gt; Numero de iteracao = 3 x0 .....: = 1.125000 x0 (novo) .: = 0.843750 Funcao ....: f(x) = 1.125000 ^ 2 = 1.265625 Derivada ..: f'(x) = 2 * 1.125000 = 2.250000  ==&gt; Numero de iteracao = 4 x0 .....: = 0.843750 x0 (novo) .: = 0.632813 Funcao ....: f(x) = 0.843750 ^ 2 = 0.711914 Derivada ..: f'(x) = 2 * 0.843750 = 1.687500  . . .  ==&gt; Numero de iteracao = 20 x0 .....: = 0.008457 x0 (novo) .: = 0.006342 Funcao ....: f(x) = 0.008457 ^ 2 = 0.000072 Derivada ..: f'(x) = 2 * 0.008457 = 0.016913  ==&gt; Numero de iteracao = 21 x0 .....: = 0.006342 x0 (novo) .: = 0.004757 Funcao ....: f(x) = 0.006342 ^ 2 = 0.000040 Derivada ..: f'(x) = 2 * 0.006342 = 0.012685  ==&gt; Numero de iteracao = 22 x0 .....: = 0.004757 x0 (novo) .: = 0.003568 Funcao ....: f(x) = 0.004757 ^ 2 = 0.000023 Derivada ..: f'(x) = 2 * 0.004757 = 0.009514  ==&gt; Numero de iteracao = 23 x0 .....: = 0.003568 x0 (novo) .: = 0.002676 Funcao ....: f(x) = 0.003568 ^ 2 = 0.000013 Derivada ..: f'(x) = 2 * 0.003568 = 0.007135 </pre> <p>A solucao final eh: 0.002676</p> <p>Tempo de execucao : 10.979 s</p>	<pre> ==&gt; Numero de iteracao = 1 x0 .....: = 2.000000 x0 (novo) .: = 1.750000 Funcao ....: f(x) = (2.000000 ^ 3) - (2 * 2.000000^2) + 2 = 2.000000 Derivada ..: f'(x) = (3 * 2.000000 ^ 2) - (4 * 2.000000) = 4.000000  ==&gt; Numero de iteracao = 2 x0 .....: = 1.750000 x0 (novo) .: = 1.467857 Funcao ....: f(x) = (1.750000 ^ 3) - (2 * 1.750000^2) + 2 = 1.234375 Derivada ..: f'(x) = (3 * 1.750000 ^ 2) - (4 * 1.750000) = 2.187500  ==&gt; Numero de iteracao = 3 x0 .....: = 1.467857 x0 (novo) .: = 0.747513 Funcao ....: f(x) = (1.467857 ^ 3) - (2 * 1.467857^2) + 2 = 0.853443 Derivada ..: f'(x) = (3 * 1.467857 ^ 2) - (4 * 1.467857) = 0.592385  ==&gt; Numero de iteracao = 4 x0 .....: = 0.747513 x0 (novo) .: = 1.242343 Funcao ....: f(x) = (0.747513 ^ 3) - (2 * 0.747513^2) + 2 = 1.300141 Derivada ..: f'(x) = (3 * 0.747513 ^ 2) - (4 * 0.747513) = -1.313725  A solucao final eh: 1.242343  Tempo de execucao : 8.894 s </pre>

**Entrada de Dados:**  $x_0 = 2$ ; Erro = 0.001;  $\beta = 0.1$

$f(x) = x^2$	$f(x) = x^3 - 2x^2 + 2$
<pre> ==&gt; Numero de iteracao = 1 x0 .....: = 2.000000 x0 (novo) ..: = 1.900000 Funcao ....: f(x) = 2.000000 ^ 2 = 4.000000 Derivada ...: f'(x) = 2 * 2.000000 = 4.000000  ==&gt; Numero de iteracao = 2 x0 .....: = 1.900000 x0 (novo) ..: = 1.805000 Funcao ....: f(x) = 1.900000 ^ 2 = 3.610000 Derivada ...: f'(x) = 2 * 1.900000 = 3.800000  ==&gt; Numero de iteracao = 3 x0 .....: = 1.805000 x0 (novo) ..: = 1.714750 Funcao ....: f(x) = 1.805000 ^ 2 = 3.258025 Derivada ...: f'(x) = 2 * 1.805000 = 3.610000  ==&gt; Numero de iteracao = 4 x0 .....: = 1.714750 x0 (novo) ..: = 1.629012 Funcao ....: f(x) = 1.714750 ^ 2 = 2.940367 Derivada ...: f'(x) = 2 * 1.714750 = 3.429500  . . . .  ==&gt; Numero de iteracao = 87 x0 .....: = 0.024281 x0 (novo) ..: = 0.023067 Funcao ....: f(x) = 0.024281 ^ 2 = 0.000590 Derivada ...: f'(x) = 2 * 0.024281 = 0.048561 </pre>	<pre> ==&gt; Numero de iteracao = 1 x0 .....: = 2.000000 x0 (novo) ..: = 1.950000 Funcao ....: f(x) = (2.000000 ^ 3) - (2 * 2.000000^2) + 2 = 2.000000 Derivada ...: f'(x) = (3 * 2.000000 ^ 2) - (4 * 2.000000) = 4.000000  ==&gt; Numero de iteracao = 2 x0 .....: = 1.950000 x0 (novo) ..: = 1.899830 Funcao ....: f(x) = (1.950000 ^ 3) - (2 * 1.950000^2) + 2 = 1.809875 Derivada ...: f'(x) = (3 * 1.950000 ^ 2) - (4 * 1.950000) = 3.607500  ==&gt; Numero de iteracao = 3 x0 .....: = 1.899830 x0 (novo) ..: = 1.849084 Funcao ....: f(x) = (1.899830 ^ 3) - (2 * 1.899830^2) + 2 = 1.638452 Derivada ...: f'(x) = (3 * 1.899830 ^ 2) - (4 * 1.899830) = 3.228744  ==&gt; Numero de iteracao = 4 x0 .....: = 1.849084 x0 (novo) ..: = 1.797214 Funcao ....: f(x) = (1.849084 ^ 3) - (2 * 1.849084^2) + 2 = 1.484002 Derivada ...: f'(x) = (3 * 1.849084 ^ 2) - (4 * 1.849084) = 2.861002  . . . .  ==&gt; Numero de iteracao = 7 x0 .....: = 1.686553 x0 (novo) ..: = 1.624532 Funcao ....: f(x) = (1.686553 ^ 3) - (2 * 1.686553^2) + 2 = 1.108412 Derivada ...: f'(x) = (3 * 1.686553 ^ 2) - (4 * 1.686553) = 1.787170  ==&gt; Numero de iteracao = 8 x0 .....: = 1.624532 x0 (novo) ..: = 1.553428 Funcao ....: f(x) = (1.624532 ^ 3) - (2 * 1.624532^2) + 2 = 1.009101 Derivada ...: f'(x) = (3 * 1.624532 ^ 2) - (4 * 1.624532) = 1.419186  ==&gt; Numero de iteracao = 9 x0 .....: = 1.553428 x0 (novo) ..: = 1.463503 Funcao ....: f(x) = (1.553428 ^ 3) - (2 * 1.553428^2) + 2 = 0.922360 Derivada ...: f'(x) = (3 * 1.553428 ^ 2) - (4 * 1.553428) = 1.025704  ==&gt; Numero de iteracao = 10 x0 .....: = 1.463503 x0 (novo) ..: = 1.314617 Funcao ....: f(x) = (1.463503 ^ 3) - (2 * 1.463503^2) + 2 = 0.850909 Derivada ...: f'(x) = (3 * 1.463503 ^ 2) - (4 * 1.463503) = 0.571513  ==&gt; Numero de iteracao = 11 x0 .....: = 1.314617 x0 (novo) ..: = 2.419401 Funcao ....: f(x) = (1.314617 ^ 3) - (2 * 1.314617^2) + 2 = 0.815509 Derivada ...: f'(x) = (3 * 1.314617 ^ 2) - (4 * 1.314617) = -0.073816 </pre>
A solucao final eh: 0.018788	A solucao final eh: 2.419401
Tempo de execucao : 11.476 s	Tempo de execucao : 11.647 s

## 5 CONCLUSÕES

Como principal conclusão podemos afirmar que:

- Otimização por Descida de Gradiente é um método iterativo;
- Não possui garantia de encontrar parâmetros ótimos;
- Esta técnica utiliza somente derivadas de 1<sup>a</sup> ordem.
- Alterando  $\beta$  pode-se evitar mínimo local.

## 6 REFERÊNCIAS BIBLIOGRAFIA

- DEITEL, H.M. ; DEITEL, P.J. - C++ How to program: Third edition, © 2001.
- ZIVIANI, N. Projeto de Algoritmos com Implementações em Java e C++. Thomson, 2007.
- CHAPRA, Steven C. Applied Numerical Methods,
- PRESS, W.H.; TEUKOLSKY, S.A. Numerical Recipes in C++. Second edition, 2002.

## ANEXO – LISTAGEM DO PROGRAMA

```
//== PEL216 - PROGRAMACAO CIENTIFICA
//== Prof. Dr. Reinaldo Augusto da Costa Bianchi
//== Aluno: Flavio Infanti nº 118.310-2
//== AULA 04 - 09/08/2019
//==
//== DESCIDA DE GRADIENTE

// Bibliotecas
#include <stdio.h>
#include <stdlib.h>
#include <locale.h>

// Executa o Módulo de |X|
float Abs( float x )
{
    x = ( x * x ) / -(x);
    return x;
}
```

```

int main()
{
    int opcao;
    printf("=====\\n");
    printf("=                               =\\n");
    printf("=             DESCIDA DE GRADIENTE      =\\n");
    printf("=                               =\\n");
    printf("=     ESCOLHA A OPCAO DESEJADA        =\\n");
    printf("=                               =\\n");
    printf("=     1- Funcao f(x) = x^2            =\\n");
    printf("=                               =\\n");
    printf("=     2- Funcao f(x) = x^3 - 2x^2 + 2 =\\n");
    printf("=                               =\\n");
    printf("=                               =\\n");
    printf("=     0- Sair do Programa           =\\n");
    printf("=                               =\\n");
    printf("=====\\n\\n");

    scanf("%i",&opcao);

//ZERA VARIAVEIS
float x0 = 0., xnovo = 0., Fxnovo = 0., beta=0,
Fdxnovo = 0., E = 0.;// Fdx0 = derivada de Fx0
int k = 0;

switch(opcao)
{
    // ESCOLHE O EXERCICIO A) F(X)= X^2, SENDO xO = 2
    case 1:
        printf("\nDigite o valor para x0: ");
        scanf( "%f", &xnovo );
        printf( "Digite o Erro Esperado: " );
        scanf( "%f", &E );
        printf( "Digite a taxa de aprendizado (0<= beta <= 1): " );
        scanf( "%f", &beta );
        do{

            x0 = xnovo;
            Fxnovo = ( x0 * x0); //inserir sua função principal aqui
            Fdxnovo = 2. * ( x0 ); //inserir a derivada da função principal aqui

            xnovo = x0 - (( Fxnovo / Fdxnovo )*beta);
            k += 1;
}
}

```

```

        printf( "\n==> Numero de iteracao = %d", k );
        printf( "\n  x0 .....:= %f", x0);
        printf( "\n  x0 (novo) .:= %f", xnovo );
        printf( "\n  Funcao ....: f(x) = %f ^ 2 = %f", x0, x0 * x0);
        printf( "\n  Derivada ...: f'(x) = 2 * %f = %f\n", x0, 2 * x0);

    } while( Abs( xnovo - x0 ) >= E || Abs( Fxnovo ) >= E );
    printf("\n\nA solucao final eh: %f\n", xnovo );
    break;

// ESCOLHE O EXERCICIO B) F(X)=X^3 -2* X^2 + 2, SENDO x0 = 2
case 2:
    printf( "Digite o valor para x0: " );
    scanf( "%f", &xnovo );
    printf( "Digite o Erro Esperado: " );
    scanf( "%f", &E );
    printf( "Digite a taxa de aprendizado (0<= beta <= 1): " );
    scanf( "%f", &beta );
    do{
        x0 = xnovo;
        Fxnovo = ( x0 * x0 * x0 ) - 2. * x0 * x0 + 2.; //inserir sua função principal aqui
        Fdxnovo = 3. * ( x0 * x0 ) - 4. * x0; //inserir a derivada da função principal aqui

        xnovo = x0 - (( Fxnovo / Fdxnovo )*beta);
        k += 1;

        printf( "\n==> Numero de iteracao = %d", k );
        printf( "\n  x0 .....:= %f", x0);
        printf( "\n  x0 (novo) .:= %f", xnovo );
        printf( "\n  Funcao ....: f(x) = (%f ^3) - (2 * %f^2) + 2 = %f", x0,x0,(( x0 * x0 * x0 ) - 2. * x0 * x0 + 2));
        printf( "\n  Derivada ...: f'(x) = (3 * %f ^2 - (4 * %f) = %f\n", x0,x0,((3*x0*x0)-(4*x0)));

    } while( Abs( xnovo - x0 ) >= E || Abs( Fxnovo ) >= E );
    printf("\n\nA solucao final eh: %f\n", xnovo );
    break;
}

system("pause");
return 0;
}

```