2 - Determinação da Raiz de $x^3 + 4x^2 - 10 = 0$

Método Iterativo Simples

O método iterativo simples tem como objetivo encontrar uma raiz de uma função f(x) através da transformação:

$x = g(x)$

A partir de uma aproximação inicial x_0, define-se a sequência:

$x_{n+1} = g(x_n)$

A sequência converge para a raiz se existir um valor x^* tal que $x^* = g(x^*)$ e se a condição de convergência for satisfeita:

$|g'(x^*)| < 1$

Caso contrário, o processo diverge.

Implementação

O programa em linguagem **C** aplica o método iterativo simples para cada uma das funções $g_i(x)$ e mostra o comportamento da sequência de aproximações.

Parâmetros utilizados

- Valor inicial: $x_0 = 1.5$
- Precisão: $\varepsilon = 10^{-12}$
- Máximo de iterações: 1000

Código-Fonte

```
#include <stdio.h>
#include <math.h>
```

```c
#define EPSILON 1e-12
#define MAX 1000

double g1(double x) {
    return (x - pow(x,3) - 4*pow(x,2) + 10);
}

double g2(double x) {
    return sqrt((10/x) - 4*x);
}

double g3(double x) {
    return sqrt(10 - pow(x,3))*0.5;
}

double g4(double x) {
    return sqrt(10 / (4 + x));
}

double g5(double x) {
    return (2*pow(x,3) + 4*pow(x,2) + 10) / (3*pow(x,2) + 8*x);
}

void iterativo(double (*funcao)(double)) {
    double x0 = 1.5;
    double x1;
    int iter = 0;

    do {
        x1 = funcao(x0);
        iter++;
        printf("x%d = %.10f\n", iter, x1);

        if (iter >= MAX) {
            printf("Não convergiu em %d iterações\n", MAX);
            break;
        }

        if (isnan(x1) || isinf(x1)) {
            printf("Divergiu\n");
            break;
        }

        if (fabs(x1 - x0) < EPSILON) {
```

```
                printf("Convergiu em %d iterações\n", iter);
                break;
            }

            x0 = x1;

        } while (1);

        printf("\n");
    }

    int main() {
        iterativo(g1);
        iterativo(g2);
        iterativo(g3);
        iterativo(g4);
        iterativo(g5);
        return 0;
    }
```

Resultados Obtidos

$g_1(x)$

```
x1 = -0.8750000000
x2 = 6.7324218750
x3 = -469.7200120017
x4 = 102754555.1873851120
x5 = -1084933870531746352594944.0000000000
x6 =
1277055591444378074254579861314550183250535909418315265493330570988486656.
0000000000
x7 =
-20827129085810249974571791836278090335490162217383667597099544813057580176
72956495671066147677854017685570863387435107174994275938956662374341691407
01482595022418148424793832168821821935870461554196696031984495886336 0.000
0000000
x8 = -nan
Divergiu
```

Conclusão: Divergiu rapidamente. Os valores explodem numericamente até se tornarem `NaN` .

Motivo: |g'(x)| > 1 próximo da raiz, tornando o método instável.

$g_2(x)$

```
x1 = 0.8164965809
x2 = 2.9969088058
x3 = -nan
Divergiu
```

Conclusão: Divergiu.

Motivo: o argumento de sqrt(10/x - 4x) torna-se negativo, saindo do domínio da função real.

$g_3(x)$

```
x1 = 1.2869537676
x2 = 1.4025408035
x3 = 1.3454583740
...
x40 = 1.3652300134
x41 = 1.3652300134
Convergiu em 41 iterações
```

Conclusão: Convergiu lentamente, mas de forma estável.

Motivo: |g'(x)| está próximo de 1, reduzindo a velocidade de convergência.

$g_4(x)$

```
x1 = 1.3483997249
x2 = 1.3673763720
x3 = 1.3649570154
...
x12 = 1.3652300134
x13 = 1.3652300134
x14 = 1.3652300134
Convergiu em 14 iterações
```

Conclusão: Convergência rápida e estável.

Motivo: |g'(x)| < 1 no intervalo de interesse, método eficiente.

$g_5(x)$

```
x1 = 1.3733333333
x2 = 1.3652620149
x3 = 1.3652300139
x4 = 1.3652300134
x5 = 1.3652300134
Convergiu em 5 iterações
```

Conclusão: Convergência extremamente rápida e precisa.
Em apenas 5 iterações, a aproximação atingiu o erro solicitado (10^{-12}).

Resultados e Análise

1. **$g_1(x)$** diverge porque a derivada $g_1'(x)$ é grande em módulo próximo da raiz, violando $|g'(x)| < 1$.
2. **$g_2(x)$** diverge devido a problema de domínio: a raiz quadrada recebe valor negativo.
3. **$g_3(x)$** converge lentamente, pois o fator 1/2 reduz oscilações, mas $|g'(x)|$ ainda está próximo de 1.
4. **$g_4(x)$** apresenta convergência rápida e estável.
5. **$g_5(x)$** converge quase imediatamente, demonstrando que a forma iterativa é ideal (semelhante a Newton-Raphson).

Conclusão

- As funções **$g_1(x)$** e **$g_2(x)$ divergem**, não sendo adequadas para o método iterativo simples.
- As funções **$g_3(x)$**, **$g_4(x)$** e **$g_5(x)$ convergem**, com diferentes velocidades.
- A **melhor forma iterativa** é **$g_5(x)$**, apresentando **convergência extremamente rápida** (5 iterações) e alta precisão.

A raiz real aproximada é:

$x^* \approx 1.3652300134$

Portanto, a forma iterativa mais eficiente para resolver a equaçãoé:

$g5(x) = (2x^3 + 4x^2 + 10) / (3x^2 + 8x)$