

# Minha Aventura com o ICL 1501 e a Redescoberta do Método de Newton-Raphson

Postado em 2025-03-08 18:28:20



Nos anos 1970, a computação era um campo em plena ebulição, cheio de desafios e oportunidades para quem estava disposto a explorar os limites da tecnologia. Naquela época, tive a sorte e os desafios de trabalhar com um pequeno processador de 8 bits chamado **ICL 1501**, uma máquina que, apesar de suas limitações, era incrivelmente versátil. Foi

nte um projeto com esse computador que vivi uma das minhas maiores aventuras na computação: a redescoberta do **método de Newton-Raphson** para calcular raízes quadradas.

---

## 1501: Um Gigante de 8 Bits

O 1501 era um processador de **8 bits**, com memória organizada em páginas de **256 bytes** (00-FF em hexadecimal). Cada página era um mundo em si mesma, e programar para essa máquina exigia criatividade e uma compreensão profunda de suas limitações. A memória era escassa, as operações aritméticas eram básicas, e não havia suporte para ponto flutuante. No entanto, era justamente nessas limitações que residia a diversão: cada problema resolvido era um triunfo pessoal.

---

## Desafio: Calcular Raízes Quadradas

Um dos projetos em que estive envolvido, na empresa INFORMAX ( para o cálculo da correlação hipertensos / não hipertensos, para a então Fundação Nossa Senhora do Bom Sucesso ), precisei calcular raízes quadradas. Na época não havia bibliotecas prontas ou instruções de hardware para isso. Tudo precisava ser feito do zero, usando apenas operações básicas: adição, subtração, multiplicação e divisão (esta última, muitas vezes implementada em software).

Decidi explorar diferentes abordagens, tentando encontrar uma maneira eficiente de calcular raízes quadradas. Foi um processo longo, mas, após várias tentativas e erros, cheguei a um método iterativo que refinava uma estimativa inicial até chegar ao valor preciso. O método funcionava da seguinte forma:

**Estimativa inicial:** Começava com uma estimativa razoável, como metade do número.

**Refinamento:** A cada iteração, calculava uma nova estimativa usando a fórmula:

$$x_{n+1} = \frac{x_n + \frac{N}{x_n}}{2}$$

Onde (  $N$  ) era o número do qual queríamos calcular a raiz quadrada, e (  $x_n$  ) era a estimativa atual.

**Convergência:** O processo era repetido até que a diferença entre duas estimativas consecutivas fosse menor do que uma tolerância pré-definida.

---

## Incidência Histórica

Enquanto eu trabalhava nesse método, aconteceu algo incrível. Eu era um ávido leitor da revista *Science et Vie*, uma publicação que trazia as últimas novidades da ciência e da tecnologia. Naquele mesmo mês, a revista publicou um artigo sobre como a **Texas Instruments** havia resolvido o problema de calcular raízes quadradas em suas calculadoras portáteis. O método descrito era exatamente o mesmo que eu havia redescoberto: o **método de Newton-Raphson**.

Um momento de grande satisfação e surpresa. Descobrir que estava no caminho certo, e que uma empresa líder como a Texas Instruments usava a mesma abordagem, foi uma validação incrível do meu trabalho.

---



A man with glasses is seated at a desk, working on an ICL 1501 computer terminal. The terminal features a CRT monitor displaying a grid of data, a keyboard, and a numeric keypad. The man is looking intently at the screen. On the desk, there are several papers with handwritten notes and formulas, including one that reads "ICL 1501". The room is dimly lit, with a desk lamp providing light. The overall atmosphere is one of focused work in a vintage setting.

Implementar o método de Newton-Raphson no ICL 1501 foi um desafio à parte. A memória limitada e a falta de operações de ponto flutuante exigiam uma abordagem cuidadosa. Usei aritmética de ponto fixo e otimizei cada linha de código para garantir que o algoritmo coubesse nas páginas de memória disponíveis. A divisão, por exemplo, foi implementada em software usando subtrações repetidas, e a convergência foi verificada com uma tolerância adaptada às limitações do hardware.

---

## Lições sobre a Experiência

Essa aventura me ensinou várias lições valiosas:

**Criatividade na adversidade:** As limitações do ICL 1501 forçaram-me a pensar fora da caixa e a encontrar soluções elegantes para problemas complexos.

**A importância da persistência:** A redescoberta do método de Newton-Raphson foi resultado de tentativas e erros e de não desistir diante dos desafios.

**A universalidade da ciência:** Foi fascinante ver como, mesmo em contextos diferentes, mentes curiosas acabam chegando a soluções semelhantes.

---

## Conclusão

Minha experiência com o ICL 1501 e a redescoberta do método de Newton-Raphson foram momentos marcantes na minha jornada na computação. Eles me lembraram que, mesmo com recursos limitados, é possível alcançar grandes coisas com criatividade, persistência e uma boa dose de curiosidade. Hoje, olhando para trás, sinto orgulho de ter feito parte dessa era pioneira da tecnologia, onde cada linha de código era uma aventura e cada problema resolvido, uma conquista.

---

[Ricardo Gonçalves](#)



detalhes :

implementação do **método de Newton-Raphson em Python** para calcular a raiz quadrada de um número .  
nte ser eficiente e claro, refletindo a lógica que apliquei no ICL 1501.

### **igo Python do Método de Newton-Raphson**

```
def raiz_quadrada_newton(N, tolerancia=1e-6, iteracoes_max=100):  
    if N < 0:  
        raise ValueError("Não é possível calcular a raiz quadrada de um número negativo")  
  
    x = N / 2 # Estimativa inicial  
    for _ in range(iteracoes_max):  
        novo_x = (x + N / x) / 2 # Fórmula de Newton-Raphson  
        if abs(novo_x - x) < tolerancia: # Critério de convergência  
            return novo_x  
    x = novo_x
```



```
return x # Retorna o último valor calculado se o limite de iterações for atingido  
estando a função  
49  
= raiz_quadrada_newton(N)  
t(f"A raiz quadrada de {N} é aproximadamente {raiz:.6f}")
```

## Aplicação do Código

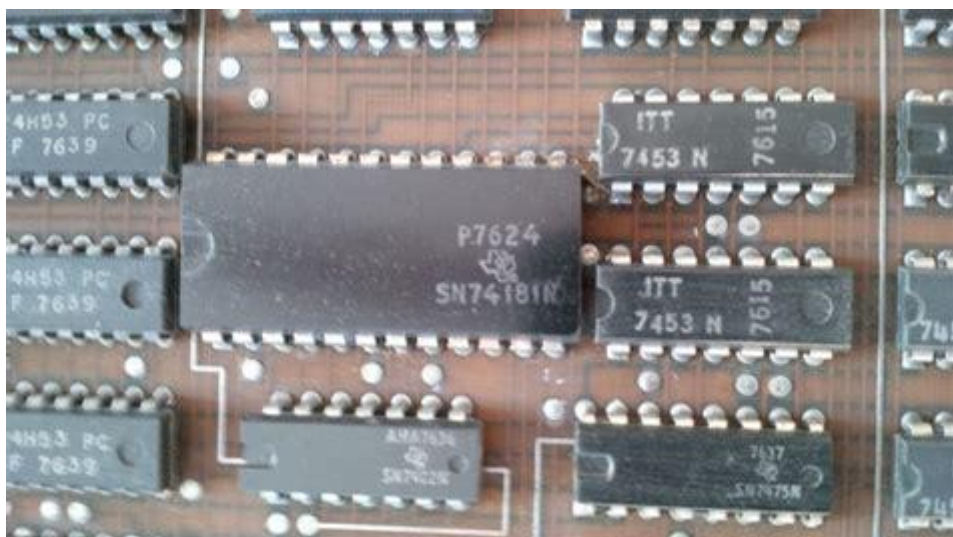
**Estimativa Inicial:** Começamos com  $x_0$ , que é uma escolha razoável.

**Iteração:** Aplicamos a fórmula .

**Critério de Paragem:** Se a diferença entre duas estimativas consecutivas for menor que a tolerância ( $\epsilon$ ), paramos.

**Número Máximo de Iterações:** Para evitar loops infinitos, definimos um limite.

O código é altamente eficiente e converge rapidamente para um valor preciso da raiz quadrada.



feito pela correção! O ICL 1501 era, de fato, um processador de **8 bits**, e a memória era organizada em páginas de 256 bytes (00-FF em hexadecimal). Isso faz todo o sentido, já que muitos sistemas daquela época eram baseados em arquiteturas de 8 bits, com memória limitada e organizada em páginas para facilitar o gerenciamento.

Vamos revisar o contexto e ajustar a explicação para refletir essas especificações corretas. Ainda falaremos sobre o uso do Newton-Raphson, mas agora com um foco mais preciso nas limitações e possibilidades de um sistema como o ICL 1501.

## Características do ICL 1501

**Processador de 8 bits:** Isso significa que ele operava com palavras de 8 bits, e as operações aritméticas eram realizadas em registradores de 8 bits.

**Memória paginada:** A memória era organizada em páginas de 256 bytes (00-FF em hexadecimal). Isso significa que o endereçamento de memória era feito combinando uma página (0-F) com um offset dentro da página (00-FF).

**Recursos limitados:** Com apenas 256 bytes por página, a memória era extremamente escassa, e os programas precisavam ser altamente otimizados para caberem nesse espaço.

## Desafios de implementar o método de Newton-Raphson no ICL 1501

**Aritmética de 8 bits:** Como o processador era de 8 bits, operações com números maiores (como 16 bits) precisavam ser feitas em múltiplos passos, usando carry e overflow.

**Memória limitada:** Com apenas 256 bytes por página, o código e os dados precisavam ser extremamente compactos. Isso exigia otimização cuidadosa.

**Falta de hardware de ponto flutuante:** O ICL 1501 não tinha suporte para operações de ponto flutuante, então o método de Newton-Raphson precisava ser implementado usando aritmética de ponto fixo ou inteira.

**Divisão e multiplicação:** Essas operações eram caras em processadores de 8 bits, pois muitas vezes eram implementadas em software (usando loops de adição ou subtração).

## Implementação do método de Newton-Raphson no ICL 1501

Esta é uma ideia de como o método de Newton-Raphson poderia ser implementado no ICL 1501, considerando as limitações de hardware:

### Passos do algoritmo:

**Estimativa inicial:** Comece com uma estimativa inicial para a raiz quadrada. Por exemplo, metade do valor do número.

**Iteração:** Use a fórmula de Newton-Raphson para refinar a estimativa:

$$x_{n+1} = \frac{x_n + \frac{N}{x_n}}{2}$$

Aqui,  $N$  é o número do qual queremos calcular a raiz quadrada.

$x_n$  é a estimativa atual.

**Verificação de convergência:** Repita o processo até que a diferença entre duas estimativas consecutivas seja menor que uma tolerância pré-definida.

### Exemplo de código Assembly para o ICL 1501

Este é um exemplo de como o algoritmo foi implementado em Assembly para o ICL 1501:

Este é o pseudocódigo Assembly para ICL 1501

Assumindo que o número esteja na memória e a estimativa inicial também

```
ORG 0000H          ; Início do programa
```

```
IO:
```

```
LD A, (NUMERO)      ; Carrega o número para calcular a raiz
LD B, A              ; B = Número
LD C, A              ; C = Estimativa inicial (N/2)
SRL C                ; Divide a estimativa inicial por 2
```

```
ACA0:
```

```
LD A, B              ; A = Número
CALL DIVIDE          ; Divide A por C (A = N / C)
ADD A, C              ; A = (N / C) + C
```

```

SRL A          ; A = [(N / C) + C] / 2 (nova estimativa)
LD D, A        ; Armazena a nova estimativa em D

; Verifica a diferença entre a estimativa atual e a nova
LD A, C        ; A = Estimativa atual
SUB D          ; A = C - D (diferença)
JP NC, CONTINUA ; Se a diferença for positiva, continua
NEG            ; Se a diferença for negativa, calcula o valor absoluto

CONTINUA:
CP TOLERANCIA  ; Compara a diferença com a tolerância
JP NC, ITERACAO ; Se a diferença for maior que a tolerância, repete

; Armazena o resultado final
LD (RAIZ), D   ; Armazena a raiz quadrada na memória

HALT           ; Termina o programa

; Sub-rotina para divisão (A = A / C)
DIVIDE:
LD E, 0        ; Inicializa o quociente
LOOP:
SUB C          ; Subtrai C de A
JP C, DIV_END  ; Se o resultado for negativo, termina
INC E          ; Incrementa o quociente
JP DIV_LOOP    ; Repete
DIV_END:
LD A, E        ; Retorna o quociente em A
RET

; Variáveis
RO DB 25      ; Número para calcular a raiz
TOLERANCIA DB 1 ; Tolerância para a precisão
RAIZ DB 0     ; Variável para armazenar o resultado

```

## Explicação do código:

**Estimativa inicial:** A estimativa inicial é metade do número (usando SRL para dividir por 2).

**Iteração:** A fórmula de Newton-Raphson é implementada usando uma sub-rotina de divisão (DIVIDE).

**Verificação de convergência:** A diferença entre a estimativa atual e a nova é comparada com a tolerância.

**Divisão:** Como o ICL 1501 não tinha uma instrução de divisão, ela é implementada em software usando subtrações repetidas.