

Manual de Implementação: Simulador de Circuitos por MNA

Pedro Augusto Pappis Bandeira

Dezembro de 2025

Resumo

Este documento apresenta um guia técnico para a implementação de um simulador de circuitos elétricos no domínio do tempo. O texto conecta a teoria da Análise Nodal Modificada (MNA) à prática de programação, utilizando uma abordagem visual para ilustrar como a topologia do circuito é traduzida, componente a componente, em um sistema algébrico linear.

Sumário

1	Fundamentação Teórica	2
2	Processo de Construção da Matriz	2
2.1	Cenário de Teste: Circuito Misto	2
3	Implementação do Algoritmo	3
4	Extensão para o Domínio do Tempo	4
4.1	Composição do Vetor de Excitação (Z)	4
4.2	Discretização do Capacitor	5
4.3	Discretização do Indutor	6
4.4	Resumo do Algoritmo Transiente	6
4.5	Validação Experimental: Circuito RLC Série	7
4.5.1	Configuração do Teste	7
4.5.2	Netlist de Entrada	7
4.5.3	Resultados e Comparação	8
4.6	Modelagem de Transformador Ideal	8
4.6.1	Equações Constitutivas	8
5	Modelagem de Chaves e Comutação	9

1 Fundamentação Teórica

Diferente da Análise Nodal clássica, restrita a equações do tipo $\mathbf{Y}\mathbf{v} = \mathbf{J}$, um simulador robusto no domínio do tempo precisa incorporar variáveis de corrente essenciais para modelar indutores, fontes de tensão e componentes complexos.

A resposta para essa necessidade é a Análise Nodal Modificada (MNA), proposta originalmente por Ho, Ruehli e Brennan [1]. Este método expande o sistema linear tradicional, criando uma estrutura matricial unificada que acomoda tanto as leis de Kirchhoff para correntes quanto as equações constitutivas dos componentes. A eficiência computacional dessa abordagem, especialmente quando combinada com técnicas de matrizes esparsas, foi fundamental para o desenvolvimento dos primeiros simuladores de grande porte, como o programa CANCER (precursor do SPICE), descrito por Nagel e Rohrer [2].

Matematicamente, o sistema é representado pela seguinte equação matricial:

$$\underbrace{\begin{bmatrix} \mathbf{G} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}}_{\text{Matriz do Sistema (A)}} \cdot \underbrace{\begin{bmatrix} \mathbf{v} \\ \mathbf{i} \end{bmatrix}}_{\text{Vetor } x} = \underbrace{\begin{bmatrix} \mathbf{I}_{\text{fontes}} \\ \mathbf{V}_{\text{fontes}} \end{bmatrix}}_{\text{Vetor } Z} \quad (1)$$

Nesta estrutura, o bloco \mathbf{G} contém as condutâncias (como na análise clássica), enquanto os blocos \mathbf{B} e \mathbf{C} gerenciam as conexões de elementos que exigem variáveis auxiliares de corrente.

2 Processo de Construção da Matriz

A grande vantagem computacional da MNA reside no método de estampagem (*stamping*). Em vez de escrever as equações de Kirchhoff manualmente para o circuito inteiro, o algoritmo processa um componente por vez.

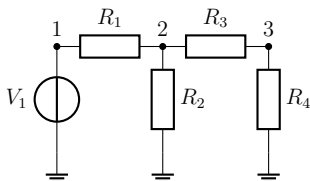
Cada elemento do circuito possui uma "assinatura" (um padrão matricial local) que é somada à matriz global. Essa acumulação progressiva garante, automaticamente, que a conservação de energia e carga seja respeitada ao final do processo.

Para cristalizar esse conceito, analisaremos a construção passo a passo da matriz para um circuito de referência.

2.1 Cenário de Teste: Circuito Misto

Considere o circuito abaixo, composto por uma fonte de tensão e quatro resistores organizados em uma topologia mista. Nosso objetivo é traduzir este diagrama esquemático para a matriz numérica \mathbf{A} .

Topologia do Circuito



Estado Inicial:

- A matriz inicia-se completamente zerada.
- A dimensão é 4×4 : 3 nós de tensão (v_1, v_2, v_3) + 1 variável de corrente para a fonte V_1 .

A seguir, visualizamos como cada componente "carimba" seus valores na matriz global. Observe que elementos conectados ao terra (GND) apenas contribuem para a diagonal do nó ativo ao qual estão ligados.

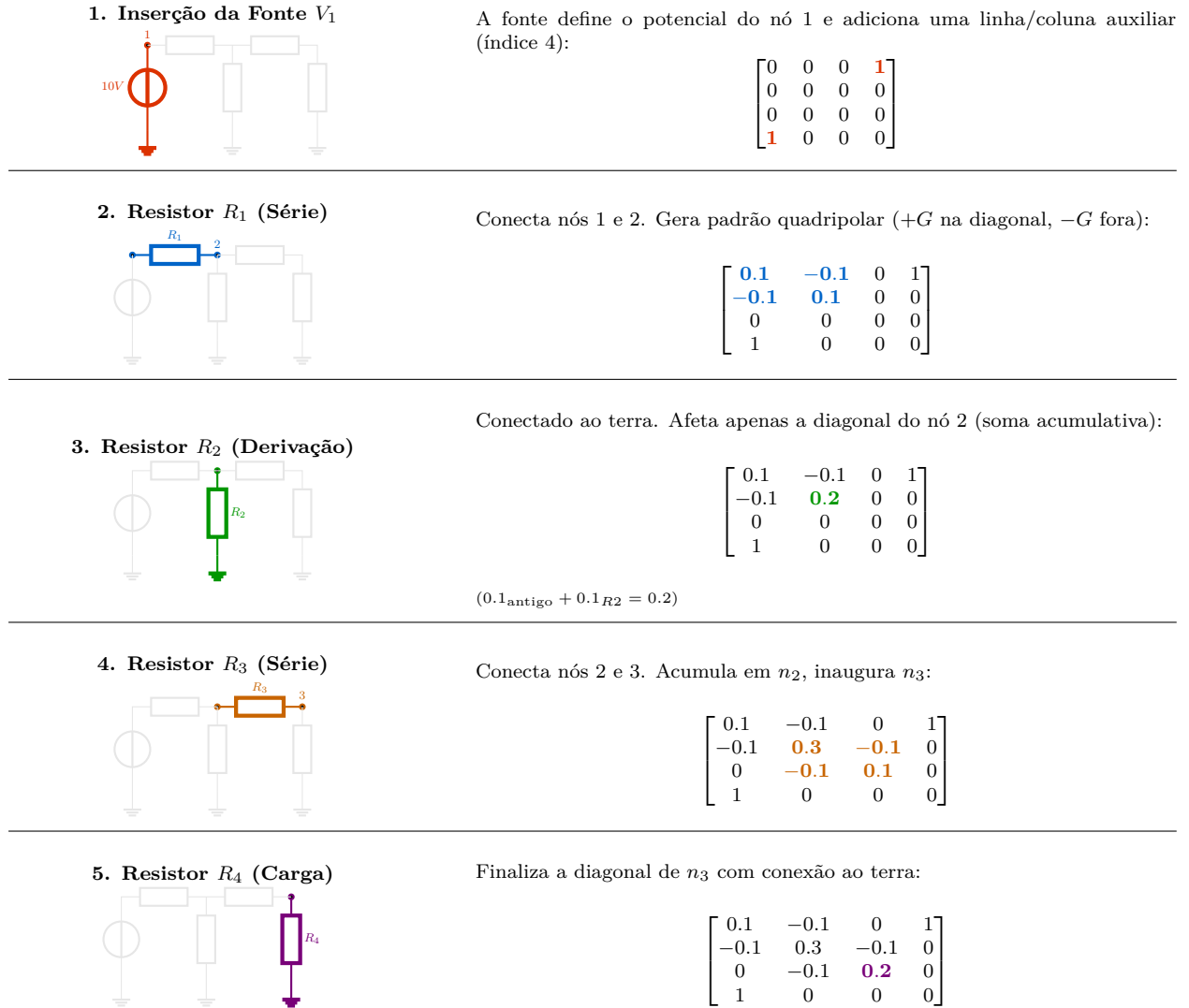


Figura 1: Rastreamento visual da construção da matriz MNA. Todos os resistores possuem condutância $G = 0.1 \text{ S}$ (10Ω).

3 Implementação do Algoritmo

Com a intuição visual estabelecida, podemos formalizar o processo em um algoritmo computacional. O código deve iterar sobre a lista de componentes (netlist) e aplicar as regras de estampagem ilustradas acima.

Três princípios fundamentais regem esta implementação:

1. **Não-Destrutividade:** Os valores são sempre somados (+), nunca sobrescritos. Isso garante a Lei das Correntes de Kirchhoff (KCL) nos nós.
2. **Mapeamento de Índices:** O "terra" (nó 0) é ignorado nas equações matriciais, pois seu potencial é conhecido (0V).
3. **Expansão Dinâmica:** A dimensão da matriz cresce conforme o número de fontes de tensão no circuito.

O pseudocódigo abaixo demonstra essa lógica estrutural:

```

1 FUNCAO MontarMatrizA(ListaComponentes, ListaNos):
2     Dimensao = NumeroDeNosAtivos + NumeroDeFontesTensao
3     A = MatrizNula(Dimensao, Dimensao)
4     k = IndiceDaPrimeiraVariavelAuxiliar
5
6     PARA CADA Componente C EM ListaComponentes FACA:
7         // Mapeia nomes de nos para indices da matriz
8         i = IndiceDoNo(C.no_positivo)
9         j = IndiceDoNo(C.no_negativo)
10
11         SE C for Resistor:
12             g = 1 / C.valor
13             // Regra: Estampagem Quadripolar (com verificacao de terra)
14             SE i != 0: A[i,i] += g
15             SE j != 0: A[j,j] += g
16             SE i != 0 E j != 0:
17                 A[i,j] -= g
18                 A[j,i] -= g
19
20         SENA O SE C for FonteDeTensao:
21             // Regra: Restricao de Tensao e Corrente Auxiliar
22             SE i != 0: A[k,i] = +1; A[i,k] = +1
23             SE j != 0: A[k,j] = -1; A[j,k] = -1
24             k = k + 1
25     FIM PARA
26     RETORNAR A
27 FIM FUNCAO

```

Listing 1: Algoritmo de Montagem da Matriz A

4 Extensão para o Domínio do Tempo

Para simular circuitos contendo capacitores e indutores, o simulador utiliza técnicas de integração numérica para discretizar as equações diferenciais no tempo. O método adotado nesta implementação é o **Euler Regressivo** (*Backward Euler*), escolhido por sua estabilidade numérica incondicional, característica fundamental para a análise de circuitos rígidos (*stiff*) típicos de eletrônica de potência [3]. Embora métodos como o Trapezoidal ofereçam maior precisão para o mesmo passo de tempo, eles podem introduzir oscilações numéricas indesejadas em circuitos com chaves, conforme discutido em análises comparativas de algoritmos de simulação [2].

Nesta abordagem, os componentes reativos são substituídos, a cada passo de tempo Δt , por circuitos equivalentes puramente algébricos, denominados **Modelos Companheiros** (*Companion Models*), permitindo sua inclusão direta na formulação de Análise Nodal Modificada (MNA).

4.1 Composição do Vetor de Excitação (Z)

Antes de detalhar os modelos individuais, é importante compreender a estrutura do vetor de excitação, denotado na implementação como **Z** (ou RHS). No domínio do tempo, este vetor deixa de conter apenas valores constantes e torna-se dinâmico.

O vetor **Z** acumula todas as funções de "forçamento" do sistema linear $\mathbf{Ax} = \mathbf{Z}$. Ele é composto pela soma de dois tipos de contribuições:

1. **Fontes Independentes:** Valores conhecidos de fontes de tensão e corrente reais conectadas ao circuito (ex: 10V, 5A ou $V(t) = \sin(\omega t)$).
2. **Fontes Históricas (I_{hist}):** Correntes virtuais oriundas dos modelos discretos de capacitores e indutores.

Matematicamente, para o solver linear, não existe distinção entre uma fonte de corrente real e a memória de um capacitor; ambas são tratadas como injeções de corrente no nó correspondente. O papel dos modelos descritos a seguir é justamente calcular o valor correto de I_{hist} com base no estado anterior (t_{n-1}) para preencher este vetor.

4.2 Discretização do Capacitor

A relação constitutiva de um capacitor linear é dada pela equação diferencial:

$$i(t) = C \frac{dv(t)}{dt} \quad (2)$$

Para discretizar esta relação, integramos ambos os lados da equação no intervalo de tempo de um passo de simulação, de t_{n-1} até t_n :

$$\int_{t_{n-1}}^{t_n} i(t) dt = C \int_{v_{n-1}}^{v_n} dv \quad (3)$$

O lado direito da equação resulta na variação exata da tensão $C(v_n - v_{n-1})$. Para o lado esquerdo, o método *Backward Euler* aproxima a integral da corrente assumindo que o valor $i(t)$ é constante e igual a $i(t_n)$ durante todo o intervalo Δt :

$$i(t_n) \cdot \Delta t \approx C (v(t_n) - v(t_{n-1})) \quad (4)$$

Isolando a corrente atual $i_n = i(t_n)$, obtemos a equação algébrica final:

$$i_n = \frac{C}{\Delta t} (v_n - v_{n-1}) \quad (5)$$

Expandindo os termos para identificar os elementos do circuito equivalente:

$$i_n = \underbrace{\left(\frac{C}{\Delta t} \right)}_{G_{eq}} v_n - \underbrace{\left(\frac{C}{\Delta t} v_{n-1} \right)}_{I_{hist}} \quad (6)$$

Esta equação representa um circuito de Norton composto por uma condutância equivalente G_{eq} em paralelo com uma fonte de corrente I_{hist} . O sinal negativo no termo histórico indica que a fonte de corrente injeta corrente no sentido contrário à queda de tensão (do nó negativo para o positivo).

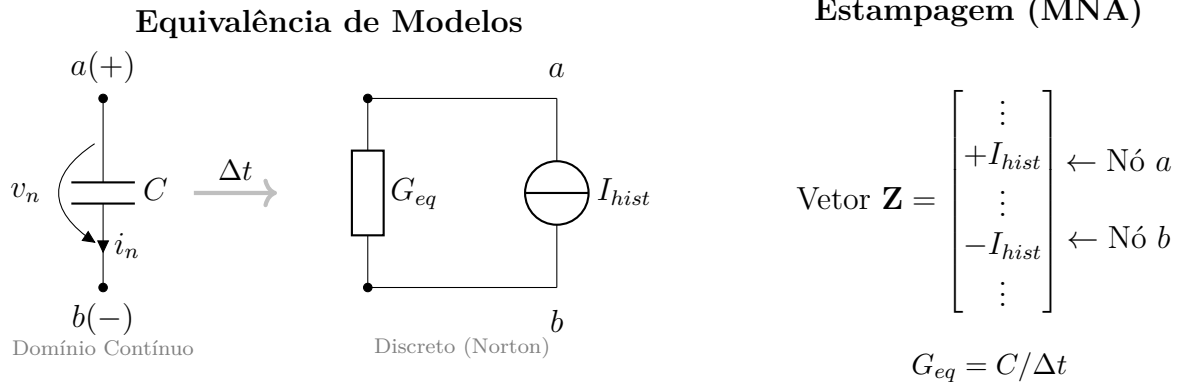


Figura 2: Modelo discreto do Capacitor. A fonte de corrente aponta para o nó positivo.

4.3 Discretização do Indutor

Partindo da lei de Faraday para o indutor:

$$v(t) = L \frac{di(t)}{dt} \quad (7)$$

Integra-se a equação no intervalo $[t_{n-1}, t_n]$:

$$\int_{t_{n-1}}^{t_n} v(t) dt = L \int_{i_{n-1}}^{i_n} di \quad (8)$$

Aplicando a aproximação *Backward Euler* para a integral da tensão (assumindo $v(t) \approx v_n$ no intervalo):

$$v_n \cdot \Delta t \approx L(i_n - i_{n-1}) \quad (9)$$

Como a Análise Nodal (MNA) resolve o sistema para tensões e correntes, isolamos a corrente atual i_n :

$$Li_n = Li_{n-1} + v_n \Delta t \implies i_n = i_{n-1} + \frac{\Delta t}{L} v_n \quad (10)$$

Reorganizando a equação para o formato de condutância ($i = Gv + I$):

$$i_n = \underbrace{\left(\frac{\Delta t}{L} \right)}_{G_{eq}} v_n + \underbrace{i_{n-1}}_{I_{hist}} \quad (11)$$

Neste caso, o termo I_{hist} é positivo, o que significa que a fonte de corrente equivalente flui no mesmo sentido da corrente convencional do ramo (do nó positivo para o negativo).

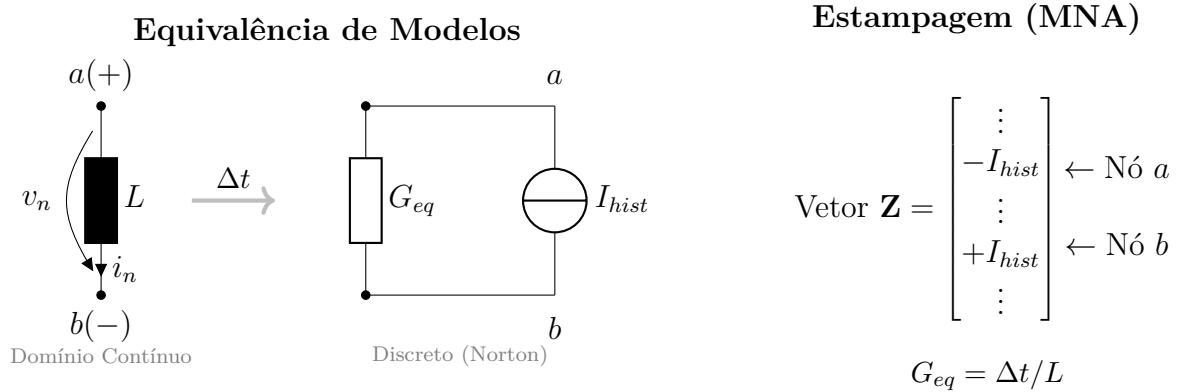


Figura 3: Modelo discreto do Indutor. A fonte de corrente auxilia o fluxo principal.

4.4 Resumo do Algoritmo Transiente

O ciclo de simulação no domínio do tempo consiste nas seguintes etapas:

1. **Inicialização:** Definição das condições iniciais ($v_C(0)$ e $i_L(0)$).

2. **Cálculo dos Modelos:** Determinação de G_{eq} e I_{hist} para cada elemento reativo com base nos valores do passo anterior.
3. **Montagem do Sistema:** Construção da matriz global \mathbf{G} e do vetor de fontes \mathbf{Z} (RHS), incluindo as contribuições históricas.
4. **Solução:** Resolução do sistema linear $\mathbf{G} \cdot \mathbf{x}_n = \mathbf{Z}_n$, onde \mathbf{x}_n contém as tensões nodais.
5. **Atualização de Estado:** Armazenamento das variáveis internas (v_C , i_L) para o próximo passo de tempo.
6. **Avanço Temporal:** Incremento do tempo $t \leftarrow t + \Delta t$ até t_{final} .

4.5 Validação Experimental: Circuito RLC Série

Para validar a integridade do motor de simulação e a precisão da discretização numérica implementada, realizou-se um estudo de caso utilizando um circuito RLC série submetido a um degrau de tensão.

Este circuito foi selecionado por possuir uma resposta transiente subamortecida com solução analítica conhecida, permitindo a quantificação direta do comportamento do método *Backward Euler*.

4.5.1 Configuração do Teste

O circuito consiste em uma fonte de tensão contínua $V_{in} = 10 \text{ V}$ conectada a um resistor $R = 1\Omega$, um indutor $L = 10\mu\text{H}$ e um capacitor $C = 10\mu\text{F}$.

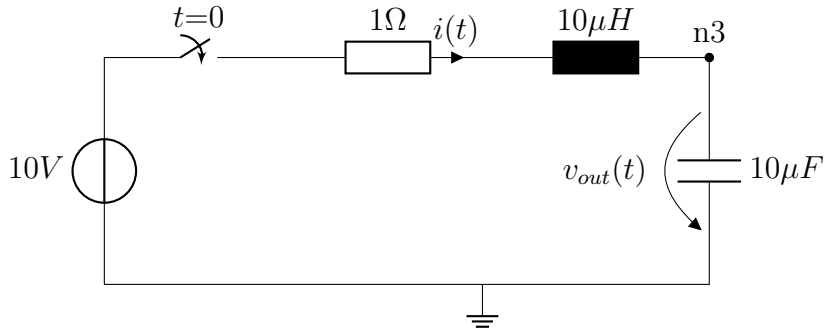


Figura 4: Topologia do circuito RLC com chaveamento em $t = 0$.

4.5.2 Netlist de Entrada

A descrição do circuito utilizando a estrutura de classes `Netlist` implementada no simulador é apresentada abaixo:

```

1 # Parametros
2 Vin = 10.0
3 R_val = 10.0
4 L_val = 1e-3
5 C_val = 1e-6
6
7 # Instanciação
8 nl = Netlist()
9 nl.add_voltage_source("Vin", Vin, ['n1', 'gnd'], type="DC")
10 nl.add_resistor("R1", R_val, ['n1', 'n2'])
11 nl.add_inductor("L1", L_val, ['n2', 'n3'])
12 nl.add_capacitor("C1", C_val, ['n3', 'gnd'])

```

Listing 2: Definição do RLC na Netlist

4.5.3 Resultados e Comparação

A simulação foi executada com um passo de tempo fixo $\Delta t = 0.5\mu s$. A Figura 5 compara os pontos discretos obtidos pelo simulador MNA desenvolvido com a curva analítica teórica (linha contínua) dada por:

$$v_c(t) = V_{in} \left[1 - e^{-\alpha t} \left(\cos(\omega_d t) + \frac{\alpha}{\omega_d} \sin(\omega_d t) \right) \right] \quad (12)$$

Onde $\alpha = R/2L = 50.000 \text{ rad/s}$ e $\omega_d = \sqrt{1/LC - \alpha^2} \approx 86.602 \text{ rad/s}$.

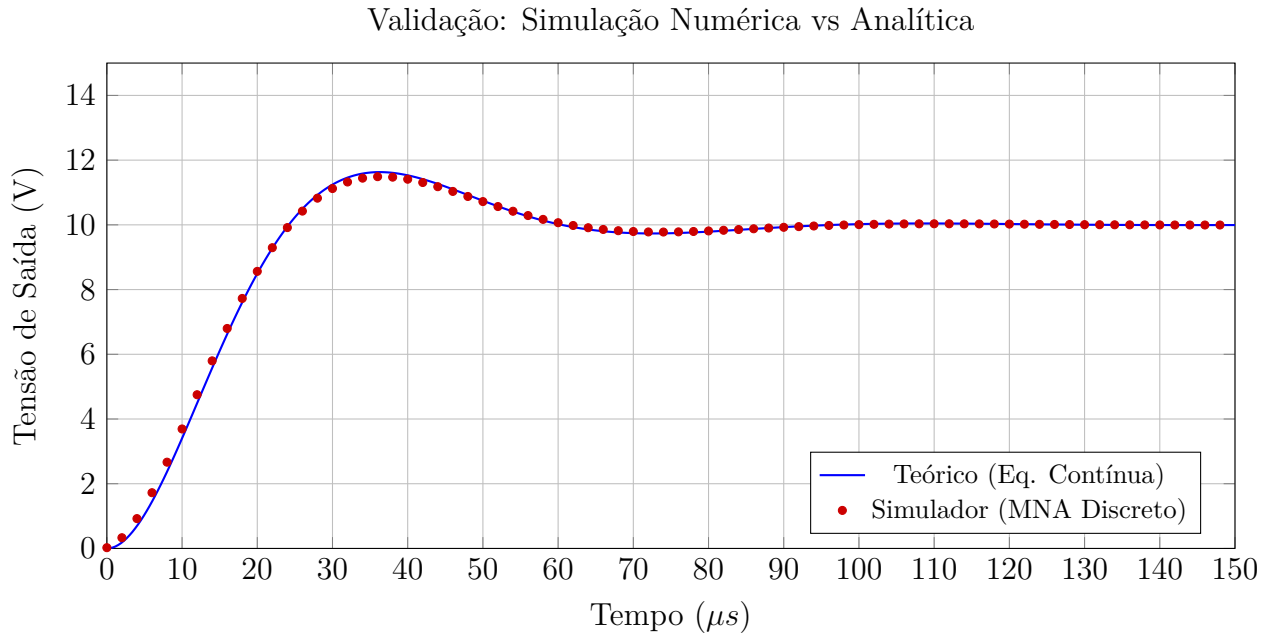


Figura 5: Comparação detalhada. A linha azul representa a solução analítica exata. Os pontos vermelhos são os valores calculados pelo simulador desenvolvido, evidenciando a precisão do método.

Observa-se uma excelente concordância entre os modelos. Pequenos desvios na amplitude são esperados devido à natureza dissipativa (estabilidade L) do método Euler Regressivo, que tende a introduzir um amortecimento numérico artificial em passos de tempo maiores.

4.6 Modelagem de Transformador Ideal

O simulador suporta transformadores ideais definidos por quatro nós (dois para o primário e dois para o secundário) e uma relação de espiras $n = N_p/N_s$.

Diferente de transformadores reais modelados por indutâncias acopladas (que poderiam ser resolvidos apenas com a matriz \mathbf{G} e a discretização de indutores), o modelo ideal impõe restrições algébricas diretas entre tensões e correntes, o que exige a expansão da matriz MNA.

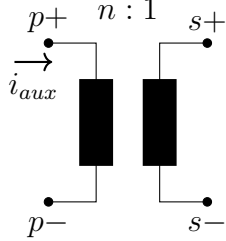
4.6.1 Equações Constitutivas

Para um transformador com relação $n : 1$, as equações fundamentais são:

$$v_p(t) = n \cdot v_s(t) \quad \text{e} \quad i_s(t) = -n \cdot i_p(t) \quad (13)$$

Para incorporar essas relações no sistema linear $\mathbf{Ax} = \mathbf{Z}$, o algoritmo introduz uma variável auxiliar i_{aux} que representa a corrente fluindo no enrolamento primário (i_p). Isso resulta na adição de uma nova linha k (equação de tensão) e uma nova coluna k (contribuição de corrente) na matriz do sistema.

Modelo Esquemático



Estampagem (MNA)

A matriz \mathbf{A} é expandida da seguinte forma:

$$\begin{bmatrix} \dots & \dots & \dots & \dots & +\mathbf{1} \\ \dots & \dots & \dots & \dots & -\mathbf{1} \\ \dots & \dots & \dots & \dots & -\mathbf{n} \\ \dots & \dots & \dots & \dots & +\mathbf{n} \\ +\mathbf{1} & -\mathbf{1} & -\mathbf{n} & +\mathbf{n} & 0 \end{bmatrix} \begin{array}{l} \leftarrow \text{Nó } p+ \\ \leftarrow \text{Nó } p- \\ \leftarrow \text{Nó } s+ \\ \leftarrow \text{Nó } s- \\ \leftarrow \text{Eq. Aux } k \end{array}$$

Interpretação da Matriz:

- **Linha k (Eq. de Tensão):** Impõe que $(v_{p+} - v_{p-}) - n(v_{s+} - v_{s-}) = 0$.
- **Coluna k (Eq. de Corrente):** Distribui a corrente i_{aux} . Note que a corrente que "sai" do secundário é multiplicada por $-n$, garantindo a conservação de potência ($v_p i_p + v_s i_s = 0$).

5 Modelagem de Chaves e Comutação

A simulação de eletrônica de potência exige o tratamento de componentes que alteram a topologia do circuito, como transistores e diodos.

Se modelássemos chaves como curtos-circuitos ($R = 0$) ou circuitos abertos ($G = 0$) ideais, a estrutura da matriz do sistema mudaria a cada comutação, exigindo complexas rotinas de reordenação. Para evitar isso e garantir uma matriz de estrutura fixa, este simulador adota o Modelo Resistivo Bimodal.

Conforme descrito por Pejović e Maksimović [4], o uso de resistências finitas para representar os estados ON e OFF (R_{on}/R_{off}) evita singularidades numéricas (como malhas de fontes de tensão ou cortes de fontes de corrente) e permite que a matriz de condutâncias \mathbf{G} mantenha dimensões constantes durante toda a simulação.

O modelo é definido por:

$$R_{sw}(S) = \begin{cases} R_{on} \approx 10^{-3}\Omega & \text{se } S = \text{ON} \\ R_{off} \approx 10^8\Omega & \text{se } S = \text{OFF} \end{cases} \quad (14)$$

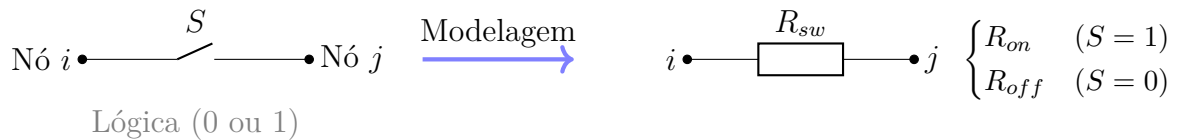


Figura 6: Linearização da chave. A substituição por condutância variável preserva a topologia da matriz MNA [4].

Matematicamente, a chave contribui para a matriz \mathbf{G} como um resistor comum, cujo valor é atualizado discretamente conforme o estado lógico S .

Referências

- [1] C.-W. Ho, A. Ruehli, and P. Brennan, “The modified nodal approach to network analysis,” *IEEE Transactions on Circuits and Systems*, vol. 22, no. 6, pp. 504–509, 1975.
- [2] L. Nagel and R. Rohrer, “Computer analysis of nonlinear circuits, excluding radiation (cancer),” *IEEE Journal of Solid-State Circuits*, vol. 6, no. 4, pp. 166–182, 1971.
- [3] J. Vlach and K. Singhal, *Computer methods for circuit analysis and design*. Van Nostrand Reinhold, 1994.
- [4] P. Pejovic and D. Maksimovic, “A new algorithm for simulation of power electronic systems using piecewise-linear device models,” *IEEE Transactions on Power Electronics*, vol. 10, no. 3, pp. 340–348, 1995.