O uso do LATEX para realizar cálculos e simulações de circuitos elétricos

The use of LATEX for electrical circuits calculation and simulation

DOI: 10.46814/lajdv4n3-049

Recebimento dos originais: 31/03/2022
Aceitação para publicação: 18/04/2022

Delmar Broglio Carvalho
Doutor em Engenharia Elétrica pela Universidade Federal de Santa Catarina (UFSC)
Instituição: Universidade Estadual de Feira de Santana (UEFS)
Endereço: Av. Transnordestina, s/n, Campus Universitário, Bahia, CEP: 44036-900
E-mail: carvalho.db@uefs.br

João Vítor de Azambuja Carvalho
Mestrando em Engenharia Civil pela Universidade Federal do Rio Grande do Sul (UFRGS)
Instituição: Universidade Federal do Rio Grande do Sul (UFRGS)
Endereço: Av. Osvaldo Aranha, 99, Porto Alegre - RS, CEP: 90035-190
E-mail: jvazambuja.carvalho@gmail.com

RESUMO
Neste artigo estão apresentados a fundamentação teórica e os resultados obtidos com a utilização do pacote gráfico PGF/TikZ, na execução de cálculos matemáticos embutidos no documento que está em elaboração utilizando apenas o Sistema de Preparação de Documentos - LATEX. Foram implementadas novas interfaces de comandos, utilizando a estrutura do LATEX, as quais facilitam a inserção das operações matemáticas bem como a visualização dos valores numéricos. Os resultados obtidos das operações matemáticas são confiáveis, o que valida a utilização dos recursos propostos, permitindo a otimização do processo de elaboração do documento e ao mesmo tempo facilita a inserção dos valores calculados em qualquer parte do documento, bem como viabiliza a reutilização de tais valores em formulações matemáticas diretas ou através de funções.

Palavras-chave: LaTeX, PGFTikZ, circuitos elétricos.

ABSTRACT
This article presents the theoretical background and the results achieved by using the PGF/TikZ graphic package, in the execution of mathematical calculations embedded in the document that is being prepared using only the Document Preparation System - LATEX. New command interfaces were implemented, using the structure of LATEX, which subserve the insertion of mathematical operations as well as the visualization of numerical values. The results obtained from mathematical operations are reliable, which validates the use of the proposed resources, allowing the optimization of the document preparation process and at the same time facilitating the insertion of the calculated values in any part of the document, as well as enabling the reuse of such values in direct mathematical formulations or through functions.

Keywords: LaTeX, PGFTikZ, electrical circuits.

.

1 INTRODUÇÃO

O Sistema de Preparação de Documentos TEX/LATEX tem se consolidado como uma das principais ferramentas de edição de documentos científicos, sendo utilizado por vários organismos e entidades nacionais e internacionais, com destaque para editoras e revistas científicas, que disponibilizam modelos pré-determinados para que os autores submetam suas produções para análise e futura publicação [1], criando assim um padrão de formatação final dos documentos. Diferente das aplicações clássicas para a edição de documentos, o LATEX - Lamport TEX, é estruturado sobre o Processador de Texto TEX [2, 3] sendo que o mesmo pode ser considerado como uma linguagem de marcação orientada para sequências de comandos, sequências de controle, ambientes, macros, registradores, entre outros recursos, que irão produzir um documento bem elaborado e com características tipográficas de alta qualidade [4, 5]. Os modelos disponibilizados para a preparação dos documentos, na maioria das vezes, buscam oferecer ao produtor do conteúdo uma forma rápida de customizar o formato final e assim concentrar os esforços no conteúdo em elaboração, como por exemplo, a disponibilização de modelos para preparação de teses e dissertações [6] ou a customização de referências bibliográficas no formato da Associação Brasileira de Normas Técnicas (ABNT) [7]. Além dos modelos disponibilizados para a customização do documento final, existem inúmeros recursos complementares disponibilizados na forma de arquivos auxiliares, com as sequências de macros (comandos) organizados estruturalmente, denominados de pacotes, os quais acrescentam novas funcionalidades para o processo de criação de elementos textuais, gráficos, formas, figuras, interfaces de programação, entre outros. Uma das finalidades mais consolidadas no Sistema de Preparação de Documentos TEX/LATEX é a sua orientação para a tipografia de equações matemáticas, o que produz um resultado final de excelente qualidade [4].

Além da qualidade tipográfica das equações matemáticas nas Ciências Exatas e nas Engenharias, uma das necessidades mais evidenciadas, na elaboração de documentos científicos, é a utilização dos resultados de operações matemáticas, que, normalmente, são realizadas em outros ambientes destinados aos cálculos e simulações e cujos valores necessitam inserção através da cópia ou da digitação, o que provoca, muitas vezes, o retrabalho ou a produção de erros. Dentro do campo de conhecimento da Engenharia Elétrica, particularmente dentro da área de Circuitos Elétricos e Eletrônicos, são verificadas necessidades recorrentes do uso de resultados numéricos das formulações matemáticas nas produções de relatórios, ensaios, tutoriais, materiais didáticos, artigos, entre outros. Como é uma área dependente da representação gráfica para os circuitos elétricos e eletrônicos, os quais são objetos de estudo e análise, o uso do LATEX trouxe avanços na edição de documentos com o uso de pacotes para o desenho destes circuitos, como o pacote CircuiTi*k*Z [8], baseado nos comandos gráficos Ti*k*Z e a biblioteca ***{circuits}*** do pacote gráfico PGF/Ti*k*Z [9]. Normalmente, os cálculos

realizados na análise de circuitos elétricos e eletrônicos consistem da aplicação direta das formulações oriundas da aplicação da Lei de Ohm e das análises pelas Leis de Kirchhoff, onde as variáveis de tensão, corrente e potência são objeto de determinação [10], cujos valores necessitam de uniformidade na visualização no documento final, seja em *notação científica* ou em *notação de engenharia*, levando muitas vezes a repetitivos processos de correção. Com o uso do Sistema de Preparação de Documentos TEX/LATEX a representação gráfica de circuitos elétricos e eletrônicos e a tipografia de equações matemáticas são facilmente inseridos em um documento, enquanto que a realização de cálculos diretamente no documento em elaboração ainda é uma lacuna a ser superada, pois, assim, são dispensáveis outros ambientes ou aplicativos usados para esta finalidade.

Este trabalho visa apresentar a viabilidade da utilização do pacote gráfico PGF/Ti*k*Z na execução de cálculos matemáticos voltados para a análise e simulação de circuitos elétricos e eletrônicos, bem como apresentar um conjunto de macros voltadas para a formatação numérica, implementações de funções e simulações e, assim, contribuir efetivamente para ampliar o uso do LATEX, permitindo o uso integrado dos recursos gráficos e de cálculo no processo de criação de documentos científicos na área de Circuitos Elétricos e Eletrônicos.

2 MATERIAL E MÉTODOS

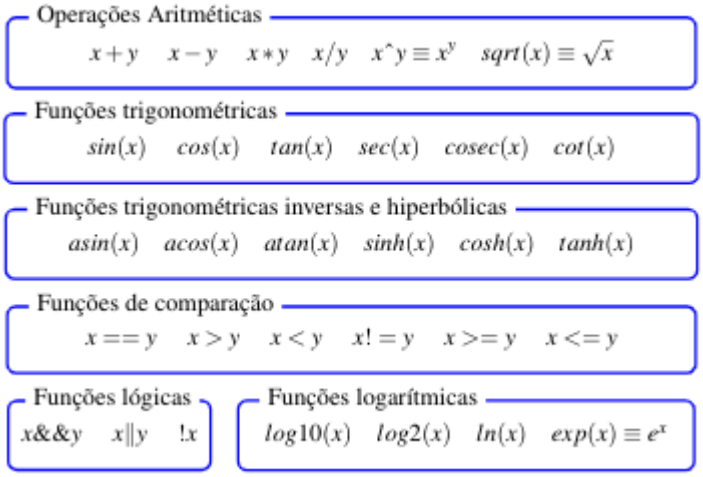
2.1 O PACOTE GRÁFICO PGF/TI*K*Z

O pacote gráfico PGF/Ti*k*Z desenvolvido por Till Tantau [11, 12] é um poderoso recurso de macros para a criação de inúmeros tipos de gráficos, facilitando a forma de inserção destes elementos no texto em elaboração. A estrutura do pacote possui três camadas (*System layer*, *Basic layer e Frontend layer*) com funcionalidades bem distintas, cujo acesso aos recursos é realizado na camada *Frontend layer* através dos comandos Ti*k*Z. O pacote gráfico PGF/Ti*k*Z possui uma estrutura voltada para a criação de diferentes tipos de gráficos, sendo que possui blocos de comandos específicos voltados para cálculos matemáticos envolvidos nas diferentes operações gráficas. Os blocos de comandos, organizados na forma de bibliotecas, são divididos em dois pacotes: Aritmética de ponto fixo e Aritmética de ponto flutuante, os quais podem ser acessados independente do uso para a geração de gráficos. O acesso aos recursos é obtido com a inclusão do comando para uso de pacotes, inserido no preâmbulo do documento em elaboração, através do comando **\usepackage{pgfmath}** ou **\usepackage{tikz}.** Os princípios e noções das nomenclaturas utilizadas na matemática são fundamentais para o uso e entendimento dos resultados obtidos através dos cálculos efetuados utilizando os recursos do pacote. Os conceitos aplicados à avaliação das expressões matemáticas são importantes, uma vez que essas, inseridas no documento base, são processadas por um analisador ou *"parser"* que acessa os recursos matemáticos do pacote PGF/Ti*k*Z. Os conceitos, regras e métodos

necessários na elaboração de documentos com o LATEX, em sua forma geral, não serão abordados neste artigo, podendo ser consultadas as referências [1, 13] para aprofundar estes conhecimentos.

As principais funções matemáticas disponíveis no pacote PGF/Ti*k*Z, são mostradas na Figura 1, sendo possível customizá-las ou incluir novas funções para atender determinados cálculos, como no caso das operações com números complexos, utilizadas nas análises senoidal e de frequência complexa aplicadas aos circuitos elétricos. Além das funções apresentadas, existem inúmeras outras disponibilizadas no pacote que podem ser consultadas no documento de referência [9].

Figura 1: Principais funções matemáticas disponíveis no pacote gráfico PGF/TikZ.



2.2 IMPLEMENTAÇÃO DA INTERFACE DE CÁLCULOS COM O PACOTE GRÁFICO PDF/TIKZ

O pacote PGF/TikZ utiliza a macro **\pgfmathparse{}**, para realizar as operações matemáticas, a qual é a interface disponibilizada para os usuários, para dar acesso à biblioteca matemática. Os resultados das operações são armazenados automaticamente na macro denominada **\pgfmathresult.** Para preservar o conteúdo dos resultados das operações realizadas e utilizá-los ao longo da edição do documento faz-se necessário atribuir o conteúdo da macro **\pgfmathresult** para outras macros, as quais podem ser consideradas como "variáveis" que armazenam os resultados das respectivas operações. Esses resultados podem ser armazenados nas dimensões disponíveis no TEX: registradores, contadores ou macros. Para o armazenamento de valores não inteiros é recomendado que sejam utilizadas as macros. Com a finalidade de diferenciar as macros internas do pacote PGF/Ti*k*Z das macros criadas ao longo do documento, que irão armazenar resultados, adotou-se a denominação "comando" para macros internas do pacote PGF/Ti*k*Z e a denominação "variáveis" para as macros de armazenamento. As configurações necessárias para usar a biblioteca de cálculo do pacote gráfico PGF/Ti*k*Z são realizadas através do pacote de chave-valor denominado **\pgfkeys{}**, com a seguinte sintaxe:

\pgfkeys{/pgf/fpu=true,/pgf/fpu/output format=sci}. O primeiro parâmetro aciona a biblioteca de ponto flutuante, podendo assumir os valores true ou false, e o segundo parâmetro define o formato de saída que será armazenado em **\pgfmathresult**. Os formatos de saída podem ser: *float*, *fixed* ou *sci*. Após a realização de um determinado cálculo é necessário desativar a biblioteca de ponto flutuante, uma vez que o pacote PGF/Ti*k*Z utiliza como padrão a biblioteca de ponto fixo para outras ações, principalmente nos ambientes gráficos [9]. Assim, para evitar que o documento base contenha inúmeros blocos de invocação do comando **\pgfmathparse{<expressão>}** e os ajustes necessários através do pacote chave-valor **\pgfkeys{}**, principalmente na ativação da biblioteca de ponto flutuante, será implementada uma nova macro para realizar essas ações, simplificando o processo de cálculo. Para as macros criadas, que remetem a um conjunto de comandos, será adotada a palavra "função". E, portanto, essa nova função que contemple uma sequência de comandos, terá a forma apresentada na Figura 2a, sendo utilizada a macro LATEX **\newcommand{}[]{}** para agregar os comandos necessários em um único bloco. A nova função, designada de **\calcula{}{}**, implementa uma nova interface de cálculo, que será utilizada para invocar os recursos matemáticos disponíveis no pacote PGF/TikZ. Na criação de novos comandos podem ser informados determinados parâmetros de entrada e conforme a sintaxe estabelecida para a macro **\newcommand{}[]{}**, estes parâmetros são referenciados sequencialmente pelo símbolo (#). A nova interface proposta possui dois parâmetros de entrada, o primeiro parâmetro (#1) é a variável que irá armazenar o resultado da operação e o segundo parâmetro (#2) é a expressão que será avaliada por **\pgfmathparse{}**. O resultado da avaliação da expressão será automaticamente armazenado em **\pgfmathresult**, cujo valor será transferido para a variável de retorno através da invocação do comando **\pgfmathsetmacro{}{}**. A sintaxe para invocar a nova função **\calcula{}{}** é apresentada na Figura 2b.

Figura 2: Código LATEX para (a) implementar e (b) invocar a função **\calcula{}{}**.

```
\newcommand{\calcula}[2]{
    \pgfkeys{/pgf/fpu=true,/pgf/fpu/output format=sci}
    \pgfmathparse{#2}
    \pgfmathsetmacro{#1}{\pgfmathresult}
    \pgfkeys{/pgf/fpu=false}
}
```

(a) Código LaTeX para implementar a interface de cálculos.

```
\calcula{<variaveldearmazenamento>}{<expressaoaseravaliada
>}
```

(b) Sintaxe para invocar a nova função.

2.3 IMPLEMENTAÇÃO DA INTERFACE PARA A VISUALIZAÇÃO DE DADOS NUMÉRICOS

Após os cálculos serem realizados, ao longo da elaboração do documento, os resultados podem ser visualizados em diferentes formatos, sendo necessária a formatação desses. O pacote PGF/Ti*k*Z disponibiliza recursos para realizar essa ação através do uso do comando **\pgfmathprintnumber{<x>}**. As diferentes possibilidades de formatos para a visualização e o arredondamento dos números são configuradas no pacote de chave-valor **\pgfkeys{}**, conforme sintaxe disponível no documento de referência [9]. Uma possível configuração para visualização dos valores numéricos tem os seguintes parâmetros: **\pgfkeys{/pgf/number format/.cd,fixed, precision=2, set decimal separator={,}}.**

Na configuração apresentada, foi estabelecido que o formato será em ponto fixo (**fixed**), com dois dígitos decimais (**precision=2**), bem como foi realizado o ajuste para a vírgula ser o símbolo para o ponto decimal conforme o padrão brasileiro. O símbolo padrão, utilizado no pacote gráfico PGF/Ti*k*Z para o separador decimal é o ponto. Os formatos para a visualização podem ser: **fixed**, **sci**, **frac** ou **std**. Sendo esta última uma alternativa mais flexível para a formatação uma vez que instrui o comando **\pgfmathprintnumber{}** para utilizar um algoritmo interno de seleção entre os formatos **fixed** ou **sci** dependendo da ordem de magnitude do número que será mostrado. O formato **sci**, formato científico, por padrão apresenta um número na base 10, podendo ser alterado para expressar o número com representação **10*E*±1** com a seguinte sintaxe: **\pgfkeys{/pgf/number format/.cd,sci,sci E,precision=2, set decimal separator={,}}**.

Figura 3: Código LATEX para (a) implementar e (b) invocar a função **\mostranumero{}{}**.

```
\newcommand{\mostranumero}[2]{
\pgfkeys{/pgf/number format/.cd,std,precision=#2,
set decimal separator={,}}
\pgfmathprintnumber{#1} \ignorespaces
}
```

(a) Código LATEX para implementar a interface de apresentação de um número.

```
\mostranumero{<variavel de armazenamento>}...
{<numero de digitos decimais>}
```

(b) Sintaxe para invocar a função de apresentação de um número.

Para evitar inúmeros blocos com a invocação das configurações e da apresentação de um determinado número, foi criada uma nova interface para agrupar os comandos em uma única chamada, conforme sintaxe mostrada na Figura 3a. A nova função implementada possui a designação **\mostranumero{}{}** e possui dois parâmetros de entrada, o primeiro parâmetro (#1) é a variável que contém o resultado de uma determinada operação e o segundo parâmetro (#2) é a precisão, em dígitos

decimais, com os quais serão mostrados os valores no documento. A nova função **\mostranumero{}{}** possui a sintaxe mostrada na Figura 3b. A função de apresentação de um número deve ser usada na posição do texto na qual deve ser inserido o valor numérico, como também pode ser usada dentro dos ambientes matemáticos.

2.4 IMPLEMENTAÇÃO DA INTERFACE DE CÁLCULOS UTILIZANDO O AMBIENTE *\TIKZMATH{}*

Outra forma de utilizar o pacote PGF/Ti*k*Z para realizar cálculos é através dos recursos do ambiente **\tikzmath{}**. Com este recurso é possível ampliar o conjunto de funções necessárias ao desenvolvimento do trabalho e customizar determinados blocos de cálculos que normalmente são recorrentes na elaboração de um documento. As principais funções matemáticas apresentadas na Figura 1, bem como os recursos disponíveis no pacote PGF/Ti*k*Z podem ser invocados dentro deste ambiente. A criação de funções dentro do ambiente **\tikzmath{}**, com finalidades para realizar determinados conjuntos de cálculos, podem ser invocadas em qualquer etapa do documento desde que dentro dos ambientes **\tikzmath{}**. A sintaxe para a criação de funções, neste ambiente, possui a forma apresentada na Figura 4.

Figura 4: Sintaxe para a criação de funções no ambiente **\tikzmath{}.**

```
\tikzmath{
function <nome da funcao A>(<parametros de entrada>){
    ...
    ...
    return <parametro de retorno funcao A>;
    };
function <nome da funcao B>(<parametros de entrada>){
    ...
    ...
    return <parametro de retorno funcao B>;
    };
}
```

Os parâmetros de entrada, bem como as “variáveis” criadas internamente nas funções são tratadas como macros. As funções são criadas utilizando a palavra reservada “*function*” e devem retornar os resultados através da palavra reservada “*return*” seguida da macro que contém o valor a ser retornado. Todas as sequências de cálculos realizadas dentro do ambiente **\tikzmath{}** devem ser seguidas da marcação de finalização designada por ponto e vírgula (;). As chamadas para as novas funções seguem o mesmo padrão utilizado para invocar as funções do pacote PGF/Ti*k*Z, como mostrado na Figura 5.

Figura 5: Invocação de funções implementadas no ambiente **\tikzmath.**

```
\tikzmath{
    \novamacrox=novafuncaoA(<parametros de entrada>);
    \novamacroy=novafuncaoB(<parametros de entrada>);
}
```

Na sintaxe apresentada, as "variáveis" denominadas de **\novamacrox** e **\novamacroy** irão receber os valores de retorno das funções denominadas **novafuncaoA{}** e **novafuncaoB{}** respectivamente, conforme os parâmetros informados.

2.5 PRECISÃO DOS RESULTADOS DAS OPERAÇÕES USANDO O PACOTE PGF/TIKZ

A biblioteca de ponto flutuante contida no pacote PGF/Ti*k*Z possui uma precisão razoável, que atende a maioria das operações de cálculos, provendo compatibilidade com o padrão IEEE de precisão dupla [9]. As operações realizadas pela biblioteca apresentam precisão relativa de 1.10−4, embora algumas operações, como a soma, possuam precisão relativa de 1.10−6. Para demonstrar os aspectos relativos à precisão, serão usadas as seguintes operações envolvendo a constante π, definida na biblioteca, e usando do conceito de *erro relativo*:

$$E_{rel} = \left| \frac{(\pi * 1,0) - \pi}{\pi} \right| = 9,5494 \cdot 10^{-8} \qquad (1)$$

A operação de multiplicação na Equação (1) introduziu um erro de representação para o número (1, 0) e, consequentemente, produziu uma mudança no valor da constante π na operação de multiplicação levando a um erro diferente de *zero*. Outro erro que pode ocorrer é o *erro de aproximação*, neste caso, será utilizada a função exponencial para demonstração:

$$E_{rel} = \left| \frac{e^{1,0} - e}{e} \right| = 0,0000117354 \qquad (2)$$

Neste exemplo, o valor de e, definido como uma constante na biblioteca, tem seu valor igual a 2, 718281828 e para a expressão ($e^y \equiv exp(x)$) é utilizada a aproximação pela Série de Maclaurin, conforme citado no documento de referência [9]. Notadamente, o truncamento da Série de Maclaurin introduziu um erro de aproximação. A função *exp(x)*, contida na biblioteca, também é usada na função de potência (x^y), a qual é aproximada por ($e^{y \cdot ln(x)}$), quando o expoente *y* não é inteiro. Portanto, os resultados das operações de potenciação com expoentes não inteiros terão a precisão limitada pelo

truncamento utilizado na função *exp(x)*. Como o pacote PGF/TikZ oferece a interface para a realização dos cálculos e implementações de novos recursos, é possível desenvolver algoritmos numéricos para sobrepor algumas limitações impostas pelas funções nativas.

3 RESULTADOS E DISCUSSÃO

3.1 IMPLEMENTAÇÃO DE UMA NOVA FUNÇÃO EXPONENCIAL

No caso da função exponencial (*exp(x)*), foi apresentada a situação do erro relativo à aproximação, que no caso, foi causado pelo truncamento da Série de Maclaurin [14]. Como mencionado, é possível implementar novas funções e assim obter um melhor desempenho para determinadas funções que apresentam certas limitações. Neste sentido, foi implementada uma nova função para efeitos de demonstração dessa capacidade. Essa nova implementação foi realizada considerando o truncamento da Série de Maclaurin com 12 termos, e foi obtido o seguinte resultado para o *erro relativo*:

$$E_{rel} = \left| \frac{e_{ni}^{1,0} - e}{e} \right| = 0,00000286946 \qquad (3)$$

Verifica-se, neste caso, uma redução de 75, 55% no valor do *erro relativo*, com a nova implementação para a função exponencial (e_{ni}^{x}). A nova implementação foi realizada com o código mostrado na Figura 6.

A nova função exponencial possui dois parâmetros de entrada: a variável que irá armazenar o resultado (#1) e o argumento (*x*) (#2). Internamente foi declarada uma variável **\soma** que acumula o resultado e uma variável \X que recebe o argumento da função exponencial, sendo também utilizada no processo do somatório, conforme a expressão geral dada por:

$$e_{ni}^{x} = 1 + x + \sum_{i=2}^{n} \frac{x^i}{i!} \qquad (4)$$

Para manter a compatibilidade com o pacote PGF/Ti*k*Z e demonstrar os recursos disponíveis optou-se por utilizar o comando **\pgfmathsetmacro{}{}** para realizar as operações de transferência de dados para as "variáveis". O somatório previsto na Equação 4 foi implementado através de um laço **\foreach \counter in {<range>}{<code>}**, conforme sintaxe prevista em [9]. A invocação do comando

\xdef garante que a variável **\soma** seja atualizada e disponível externamente ao laço de repetição **\foreach**.

Figura 6: Código LATEX para implementar a nova função exponencial.

```
\newcommand{\novaEXP}[2]{
\pgfkeys{/pgf/fpu=true,/pgf/fpu/output format=fixed}
\pgfmathsetmacro{\soma}{0}
\pgfmathsetmacro{\X}{#2}
\pgfmathsetmacro{\soma}{1+\X}
\foreach \i in {2,...,11}{
    \pgfmathparse{(\X^\i)/\i!}
    \pgfmathparse{\soma+\pgfmathresult}
    \pgfmathsetmacro{\soma}{\pgfmathresult}
    \xdef\soma{\soma}
}
\pgfmathsetmacro{#1}{\soma}
\pgfkeys{/pgf/fpu=false}
}
```

3.2 IMPLEMENTAÇÃO DA INTERFACE PARA REALIZAR OPERAÇÕES MATEMÁTICAS COM NÚMEROS COMPLEXOS

O conjunto dos números complexos (C) possui importante aplicação na análise de circuitos, principalmente quando considerados os conceitos de função-excitação complexa, fasores e frequência complexa [10] e, portanto, requerem que as operações aritméticas possam ser efetuadas sobre este conjunto. O pacote gráfico PGF/Ti*k*Z não fornece suporte para as operações básicas (soma, subtração, multiplicação e divisão) com números complexos, sendo necessário criar as funções e assim poder operar com este conjunto. Na abordagem que será utilizada foi considerado que um determinado número complexo (**z**) pode ser representado na forma cartesiana ou retangular ($\mathbf{z} = a + j \cdot b$) como também na forma polar ($\mathbf{z} = C\angle\theta$), onde $C = \sqrt{a^2 + b^2}$ e $\theta = tan^{-1}(b/a)$. Assim, as quatro variáveis (a,b,C,θ) serão armazenadas em um vetor, o qual representará um determinado número complexo, permitindo assim a utilização das representações cartesiana ou polar, para apresentar os resultados das operações realizadas com os números complexos. Para dar suporte a esta representação foi desenvolvida uma nova função, utilizando o ambiente ***\tikzmath()***, denominada de ***\complexo{}{}{}***, cujo código é apresentado na Figura 7.

A função ***\complexo{}{}{}*** possui três parâmetros de entrada, o primeiro parâmetro (#1) é a variável, do tipo vetor, onde serão armazenados os valores relativos à representação do número complexo (a,b,C,θ), o segundo parâmetro (#2) é a parte real e o terceiro parâmetro (#3) é a parte imaginária. Neste caso, os dados de entrada devem ser inseridos na forma retangular.

Figura 7: Código LATEX para implementar a função **\complexo{}{}{}**.

```
\newcommand{\complexo}[3]{
\pgfkeys{/pgf/fpu=true,/pgf/fpu/output format=sci}
\tikzmath{
    \M=modulo(#2,#3);
    \A=fase(#2,#3);
    \flag=(#2/abs(#2));
    #1{1}=#2;
    #1{2}=#3;
    #1{3}=\M;
    #1{4}=\A+((\flag-1)*(-90));
}
\pgfkeys{/pgf/fpu=false}
}
```

A função ***\complexo{}{}{}*** realiza internamente os cálculos necessários para a transformar para a forma polar através da chamada às funções módulo e fase, bem como da correção do ângulo de fase quando o vetor resultante se encontra no segundo ou quarto quadrantes do plano cartesiano. O código desenvolvido para as funções módulo e fase é apresentado na Figura 8.

Figura 8: Código LATEX para implementar as funções módulo e fase no ambiente **\tikzmath{}.**

```
\tikzmath{
function modulo(\a,\b){
   \r=sqrt(abs(\a)^2+abs(\b)^2);
   return \r;
};
function fase(\a,\b){
   \r=atan(\b/\a);
   return \r;
};
}
```

As funções **modulo** e **fase** possuem como parâmetros de entrada as partes real e imaginária de um número complexo e retornam respectivamente o módulo e o ângulo de fase do respectivo número. Para demonstrar a aplicação das funções desenvolvidas, serão criados dois números complexos **\x** e **\y**, conforme o código mostrado na Figura 9.

Figura 9: Código LATEX para invocar a função **\complexo{}{}{}.**

```
\complexo{\x}{1}{5}
\complexo{\y}{-2}{7}
```

Após a invocação da função um número complexo **\x** foi criado, sendo atribuído o valor "1" como parte real e o valor "5" como parte imaginária. O resultado da operação após a invocação da função ***\complexo{}{}*** irá produzir um vetor que terá quatro parâmetros: parte real, parte imaginária, módulo e fase: **\x** ={ 1; 5; 5,1; 78,69}. Da mesma forma foi criado um número complexo **\y** sendo

atribuído o valor " 2" como parte real e o valor "7" como parte imaginária. O resultado da operação após a invocação da função ***\complexo{}{}*** irá produzir um vetor que terá quatro parâmetros: parte real, parte imaginária, módulo e fase: **\y** ={ 2; 7; 7, 28; 105, 95}.

Para a visualização do vetor, que contém os quatro valores, foi implementada uma função denominada ***\mostracomplexo{}***, conforme o código apresentado na Figura 10a.

A nova função de visualização utiliza a função ***\mostranumero{}*** (Figura 3) e agrupa itens de formatação para visualização de um vetor, no caso com quatro elementos. A função possui um parâmetro de entrada (#1), sendo esse a variável, do tipo vetor, que contém os quatro parâmetros. Os valores armazenados em uma determinada variável do tipo complexo, podem ser inseridos ao longo do texto conforme apresentado na Figura 10b .

Os recursos desenvolvidos para armazenar as duas formas de representação de um determinado número complexo, buscam facilitar a inserção dos valores em qualquer etapa da elaboração do texto, principalmente dentro dos ambientes matemáticos como mostrado nas Equações (5) e (6) para as variáveis complexas **\x** e **\y.**

$$\mathbf{x} = \begin{cases} 1 + j \cdot 5 & \text{forma retangular} \\ 5,1\angle 78,69^{\circ} & \text{forma polar} \end{cases} \tag{5}$$

$$\mathbf{y} = \begin{cases} -2 + j \cdot 7 & \text{forma retangular} \\ 7,28\angle 105,95^{\circ} & \text{forma polar} \end{cases} \tag{6}$$

Figura 10: Código LATEX para implementar e invocar a função \mostracomplexo{}.

```
\newcommand{\mostracomplexo}[1]{
$\{\mostranumero{#1{1}}{2};\mostranumero{#1{2}}{2};...
\mostranumero{#1{3}}{2};\mostranumero{#1{4}}{2}\}$
}
```

(a) Implementação da função \mostracomplexo{}.

```
... a vari{\'{a}}vel $\mathbf{\backslash x}=\mostracomplexo
{\x}$ possui quatro valores...
```

(b) Exemplo de inserção no texto dos valores armazenados em uma variável do tipo complexo.

As Equações (5) e (6) foram inseridas no documento conforme o código apresentado na Figura 11.

Figura 11: Código LATEX implementado para apresentar um número complexo na forma retangular e polar.

```
\begin{equation}
\mathbf{\backslash x}= \left\{
\begin{array}{ll}
    \mostracomplexor{\x} &  \mbox{forma retangular}\\
    \mostracomplexop{\x} &  \mbox{forma polar}
\end{array}
\right.
\end{equation}
```

As funções ***\mostracomplexor{}*** e ***\mostracomplexop{}*** coletam as informações armazenadas na variável complexa (parâmetro de entrada) e acrescentam elementos de formatação (retangular ou polar) associados à chamada da função ***\mostranumero{}***, conforme o código apresentado na Figura 12.

Figura 12: Códigos LATEX implementados para visualização de números complexos.

```
\newcommand{\mostracomplexor}[1]{
\mostranumero{#1{1}}{2} +j\cdot \mostranumero{#1{2}}{2}
}
```

(a) Função para apresentar um número complexo na forma retangular.

```
\newcommand{\mostracomplexop}[1]{
\mostranumero{#1{3}}{2} \angle \mostranumero...
{#1{4}}{2}^{\circ}
}
```

(b) Função para apresentar um número complexo na forma polar.

As operações matemáticas (+,-,*,/), que foram implementadas para números complexos, serão demonstradas utilizando os valores armazenados nas variáveis x e y (Figura 9). O resultado da soma entre os números complexos (**\x + \y**) foi armazenado em uma nova variável **\Somaxy**, que após a operação assumiu os seguintes valores:

$$\backslash\mathbf{Somaxy} = \begin{cases} -1 + j \cdot 12 & \text{forma retangular} \\ 12{,}04 \angle 94{,}76^{\circ} & \text{forma polar} \end{cases} \quad (7)$$

O resultado da subtração entre os números complexos (**\x - \y**) foi armazenado em uma nova variável **\Subxy**, que após a operação assumiu os seguintes valores:

$$\backslash\mathbf{Subxy} = \begin{cases} 3 + j \cdot -2 & \text{forma retangular} \\ 3{,}61 \angle -33{,}69^{\circ} & \text{forma polar} \end{cases} \quad (8)$$

O resultado da multiplicação entre os números complexos (**\x * \y**) foi armazenado em uma nova variável **\Mulxy**, que após a operação assumiu os seguintes valores:

$$\textbf{\Mulxy} = \begin{cases} -37 + j \cdot -3 & \text{forma retangular} \\ 37,12 \angle 184,64^{\circ} & \text{forma polar} \end{cases} \quad (9)$$

O resultado da divisão entre os números complexos (**\x / \y**) foi armazenado em uma nova variável **\Divxy**, que após a operação assumiu os seguintes valores:

$$\textbf{\Divxy} = \begin{cases} 0,62 + j \cdot -0,32 & \text{forma retangular} \\ 0,7 \angle -27,25^{\circ} & \text{forma polar} \end{cases} \quad (10)$$

Os resultados das operações matemáticas entre os dois números complexos **\x** e **\y** foram obtidos através da invocação das funções criadas para realizar as operações, conforme mostrado na Figura 13. Verifica-se que os resultados apresentados nas Equações (7), (8), (9) e (10) são condizentes com valores obtidos por outras aplicações que realizam operações matemáticas com números complexos. A Figura 14 apresenta o código LATEX que foi implementado para a função ***\complexoDiv{}{}{}***.

Figura 13: Código LATEX para realizar as operações matemáticas com números complexos.

```
\complexoSoma{\Somaxy}{\x}{\y}
\complexoSub{\Subxy }{\x}{\y}
\complexoMul{\Mulxy}{\x}{\y}
\complexoDiv{\Divxy}{\x}{\y}
```

Figura 14: Código LATEX que implementa a função **\complexoDiv{}{}{}**.

```
\newcommand{\complexoDiv}[3]{
\pgfkeys{/pgf/fpu=true,/pgf/fpu/output format=sci}
\tikzmath{
    \den= abs(#3{1})^2+abs(#3{2})^2;
    \re=((#2{1}*#3{1})+(#2{2}*#3{2}))/\den;
    \imag=((#2{2}*#3{1})-(#2{1}*#3{2}))/\den;
    \M=modulo(\re,\imag);
    \A=angle(\re,\imag);
    \flag=(\re/abs(\re));
    #1{1}=\re;
    #1{2}=\imag;
    #1{3}=\A;
    #1{4}=\A+((\flag-1)*(-90));
}
\pgfkeys{/pgf/fpu=false}
}
```

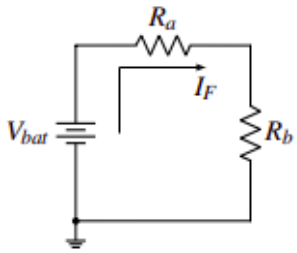
As funções implementadas para realizar as operações matemáticas entre números complexos possuem a mesma sintaxe, ou seja, possuem três parâmetros: o primeiro parâmetro (#1) é a variável

que irá receber o resultado da operação; o segundo (#2) e terceiro (#3) parâmetros são os números complexos criados pela invocação da função ***\complexo{}***, conforme mostrado na Figura 9. As operações matemáticas foram implementadas conforme as regras algébricas estabelecidas para o conjunto dos números complexos (C) [10].

3.3 APLICAÇÃO DA INTERFACE DE CÁLCULOS PARA ANÁLISES DE CIRCUITOS ELÉTRICOS

Nas seções anteriores foram estabelecidas as bases conceituais e, implementados os recursos que possibilitam a inclusão de cálculos matemáticos, diretamente no documento em elaboração, utilizando o pacote gráfico PGF/Ti*k*Z. Nesta seção serão apresentados exemplos de aplicação do uso desses recursos na resolução de cálculos utilizados nos equacionamentos de circuitos elétricos. A primeira aplicação na qual será demonstrada a viabilidade do uso desses recursos é no equacionamento de um circuito *CC* em série, com dois resistores de valores nominais fixos, conforme mostrado na Figura 15.

Figura 15: Circuito CC em série.



O circuito apresentado na Figura 15 é um exemplo clássico na introdução da análise de circuitos, com aplicação direta da Lei de Ohm, Lei de Kirchhoff para tensões (LKT) ou da Lei de Kirchhoff para correntes (LKC) [10]. Como indicado no diagrama do circuito, a variável principal para investigação é a corrente total fornecida pela fonte (*Vbat*), denominada de (I_F). Assim, utilizando a LKT, a equação da malha principal é dada por:

$$V_{bat} - R_a \cdot I_F - R_b \cdot I_F = 0 \qquad (11)$$

e, portanto, a equação para determinar a corrente *IF* é dada por:

$$I_F = \frac{V_{bat}}{R_a + R_b} \tag{12}$$

Considerando que os valores nominais atribuídos aos componentes são: *Vbat* = 12*V* , *Ra* = 1 *k*Ω e *Rb* = 500 Ω, o resultado obtido para a corrente *IF* será:

$$I_F = \frac{12}{(1 \cdot 10^3 + 5 \cdot 10^2)} = 8 \cdot 10^{-3}[\mathrm{A}] \tag{13}$$

Para o resultado mostrado na Equação (13) foi inserido, no documento, o código LATEX conforme apresentado na Figura 16.

Figura 16: Código LATEX para calcular e apresentar o valor da corrente IF.

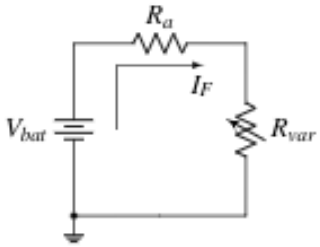
```
\calcula{\Vbat}{12}      % atribui valor fonte
\calcula{\Ra}{1e3}       % atribui valor Ra
\calcula{\Rb}{0.5e3}     % atribui valor Rb
\calcula{\If}{\Vbat/(\Ra+\Rb)}  % calcula valor corrente
\begin{equation}
    I_F=\frac{\mostranumero{\Vbat}{2}}
    {(\mostranumero{\Ra}{2}+\mostranumero{\Rb}{2})}=...
    \mostranumero{\If}{2} [A]
\end{equation}
```

No código LATEX apresentado na Figura 16 foi utilizada inicialmente a função ***\calcula{}{}*** para atribuir valores às variáveis, conforme formulado na Equação (11) e, posteriormente, a mesma função foi utilizada para realizar o cálculo da corrente total (*IF*). Embora no LATEX seja possível utilizar outros comandos para atribuição de valores para as macros, optou-se por esta abordagem visando uniformizar o processo de cálculo e garantir que todas as variáveis sejam tratadas pelo comando ***\pgfmathparse{}***. Na sequência, foi criado o ambiente **{equation}** que proporciona a visualização da Equação (13) com os dados armazenados nas variáveis e os resultados das operações matemáticas realizadas. Foi utilizada a função ***\mostranumero{}{}***, para a visualização dos resultados, configurada para notação científica automática, conforme a grandeza dos valores a serem mostrados. Os valores visualizados na Equação (13) foram obtidos das "variáveis" criadas no documento, e, os resultados apresentados, são oriundos das operações matemáticas sobre estas variáveis, ficando evidenciada a exatidão destes resultados, os quais podem ser auditados por aplicações computacionais ou calculadoras eletrônicas.

3.4 APLICAÇÃO DA INTERFACE DE CÁLCULO PARA SIMULAR A VARIAÇÃO DA RESISTÊNCIA

Com o objetivo de avaliar a influência do resistor *Rb* sobre a corrente total (I_F) fornecida pela fonte (V_{bat}), podem ser usados os recursos gráficos do pacote gráfico PGF/Ti*k*Z e as interfaces de cálculos implementadas, simultaneamente para obter um resultado visual que traduza esse comportamento, como mostrado na Figura 18a. Neste caso, o resistor fixo *Rb*, no circuito na Figura 15, será alterado para um resistor variável (*Rvar*), conforme mostrado na Figura 17.

Figura 17: Circuito CC com um resistor variável.

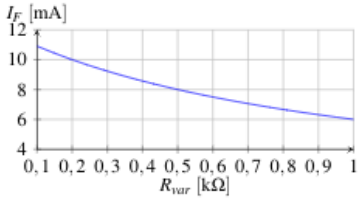


Para efeitos da análise que será realizada, este resistor terá resistência mínima de 100Ω e resistência máxima de 1*k*Ω. A Equação (12) pode ser reescrita conforme a Equação (14).

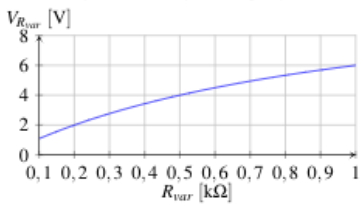
$$I_F(R_{var}) = \frac{V_{bat}}{R_a + R_{var}} \qquad (14)$$

Como pode ser visualizado na Figura 18a, a corrente fornecida pela fonte (I_F) tem seu valor alterado pela variação do valor da resistência *Rvar*, como previsto pela Equação (14).

Figura 18: Influência da variação de Rvar sobre a corrente IF e VRvar .



(a) Gráfico da corrente fornecida pela fonte I_F



(b) Gráfico da tensão sobre R_{var}

O resultado gráfico do comportamento da corrente I_F, foi obtido inserindo no documento em elaboração o código mostrado na Figura 19a. Foi utilizado o ambiente **{tikzpicture}**, que permite a execução de comandos gráficos, previstos no pacote gráfico PGF/Ti*k*Z [9], como também a inclusão de novas funções utilizando a palavra reservada ***declare function***. Para o exemplo que está sendo demonstrado foi criada uma nova função denominada: ***correntetotal()***, que possui três parâmetros de entrada (**\vf**,**\ra**,**\rv**), com nomenclatura diferente das variáveis criadas anteriormente através da função ***\calcula{}{}***. A expressão matemática da nova função implementa computacionalmente a Equação (14), utilizando os parâmetros de entrada e seguindo o padrão da sintaxe estabelecida no pacote gráfico PGF/Ti*k*Z. Para obter uma formatação gráfica mais adequada à área de Circuitos Elétricos foi utilizado o pacote gráfico PGFPlots [15], pois facilita a parametrização e a inserção das unidades utilizadas para corrente, tensão e resistência, além de permitir uma interface mais intuitiva para a criação dos eixos coordenados. A visualização do comportamento da função implementada (***correntetotal()***) é realizada dentro do ambiente **{axis}**, cujo principal parâmetro de configuração é denominado *domain*, sobre o qual é determinado os limites "mínimo" e "máximo" envolvidos na análise da função implementada. Como estamos simulando um comportamento unidimensional, por padrão, é utilizada a variável "*x*" como variável independente que será utilizada no cômputo dos valores a serem obtidos pela função ***correntetotal()***. Os valores que a variável "*x*" irá assumir são obtidos do parâmetro *domain*, que foi configurado para os valores mínimo e máximo da resistência atribuídos a *Rvar*. O comando **\plot** irá realizar internamente a avaliação da função ***correntetotal()***, utilizando os valores atribuídos para os parâmetros de entrada da função. Neste caso, foram atribuídos como parâmetros de entrada as "variáveis": *Vbat* e *Ra*, cujos valores foram previamente armazenados (Figura 16) e, a "variável" *x*, que irá cumprir o papel de *Rvar* no circuito.

Figura 19: Código LATEX para implementar simulações.

```
\begin{tikzpicture}[
    declare function={
        correntetotal(\vf,\ra,\rv)= \vf/(\ra+\rv);
    }
    ]
    \begin{axis}[axistocurrent,domain=100:1000,...
                 xlabel=$R_{var}$,ylabel=$I_F$]
        \plot [blue]{correntetotal(\Vbat,\Ra,x};
    \end{axis}
\end{tikzpicture}
```

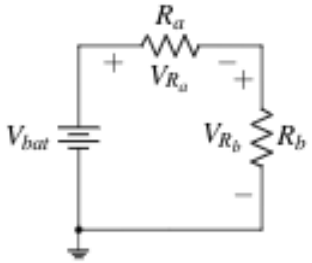
(a) Função para mostrar o comportamento da corrente I_F.

```
\begin{tikzpicture}[
    declare function={
        tensaorx(\vf,\ra,\rv)= (\vf*\rv)/(\ra+\rv));
    }
    ]
    \begin{axis}[axistovoltage,domain=100:1000,...
                 xlabel=$R_{var}$,ylabel=$V_{R_{var}}$]
        \plot [blue]{tensaorx(\Vbat,\Ra,x};
    \end{axis}
\end{tikzpicture}
```

(b) Função para mostrar o comportamento da tensão sobre R_{var}.

Uma outra abordagem que pode ser dada para a análise do circuito mostrado na Figura 15 é o conceito de divisor de tensão, onde a tensão fornecida pela fonte (*Vbat*) é distribuída proporcionalmente aos resistores associados em série. Neste caso, as variáveis envolvidas são as tensões sobre os resistores: *VRa* e *VRb*, como mostrado na Figura 20.

Figura 20: Circuito resistivo com foco na queda de tensão nos elementos.



Considerando que a corrente total fornecida pela fonte (*IF*) possui expressão dada pela Equação (12) e utilizando a Lei de Ohm, a queda de tensão sobre os resistores é dada por:

$$V_{R_a} = R_a \cdot \frac{V_{bat}}{R_a + R_b} = V_{bat} \cdot \left(\frac{R_a}{R_a + R_b} \right) \quad (15)$$

$$V_{R_b} = R_b \cdot \frac{V_{bat}}{R_a + R_b} = V_{bat} \cdot \left(\frac{R_b}{R_a + R_b} \right) \quad (16)$$

Utilizando os valores nominais atribuídos aos componentes: *V*bat = 12V, *Ra* = 1 kΩ e *Rb* = 500 Ω, os resultados obtidos para as tensões *V*Ra e *V*Rb são os seguintes:

$$V_{R_a} = 12std \cdot \left(\frac{1 \cdot 10^3}{1 \cdot 10^3 + 5 \cdot 10^2} \right) = 8\ [\mathrm{V}] \quad (17)$$

$$V_{R_b} = 12std \cdot \left(\frac{5 \cdot 10^2}{1 \cdot 10^3 + 5 \cdot 10^2} \right) = 4\ [\mathrm{V}] \quad (18)$$

A visualização das Equações (17) e (18) segue a mesma sintaxe utilizada para visualização da Equação (13), utilizando os valores calculados previamente para as tensões *V*Ra e *V*Rb. Se a resistência *Rb* for substituída por uma resistência variável *Rvar*, conforme mostrado no circuito da Figura 17, podemos avaliar graficamente o comportamento da variação da tensão sobre este resistor variável, conforme mostrado na Figura 18b e utilizando como referência a expressão geral da queda de tensão sobre o resistor:

$$V(R_{var}) = V_{bat} \cdot \left(\frac{R_{var}}{R_a + R_{var}} \right) \qquad (19)$$

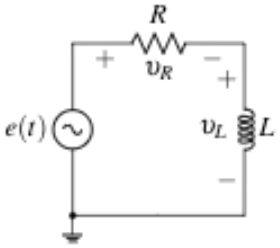
O código inserido no documento para realizar esta simulação é apresentado na Figura 19b, sendo implementada uma função dentro do ambiente **{tikzpicture}** denominada: ***tensaorx()***, que possui três parâmetros de entrada (**\vf**,**\ra**,**\rx**), seguindo a mesma sintaxe utilizada para a função ***correntetotal()***. A expressão matemática da nova função implementa computacionalmente a Equação (19). Uma outra alteração que foi introduzida no código apresentado na Figura 19b é uma customização do ambiente **{axis}** através do parâmetro *axistovoltag*e, que configura o eixo *y* para a unidade de tensão.

Os gráficos apresentados na Figura 18 para os comportamentos da corrente fornecida pela fonte (*IF*) e da tensão sobre o resistor variável *Rvar* são numericamente iguais aos valores que seriam obtidos aplicando os valores para a resistência variável nas Equações (14) e (19), demonstrando que os recursos de cálculos e simulações implementados são viáveis e produzem os resultados numéricos com a exatidão prevista, considerando aqueles que seriam obtidos por outros aplicativos computacionais ou calculadoras eletrônicas.

3.5 APLICAÇÃO DA INTERFACE DE CÁLCULO PARA ANÁLISE DE CIRCUITOS CA

No exemplo apresentado nesta seção serão utilizados os recursos da nova interface de cálculo para números complexos, bem como os conceitos de tensão e corrente senoidais, tensão e corrente de pico (*Vm, Im*), tensão e corrente *rms* (*Vrms*, *Irms*), fasores, frequência angular (ω), impedância (**Z**) e reatância (**X**), sendo que tais conceitos são abordados em profundidade na bibliografia correspondente [10]. Na representação fasorial adotada, o módulo do fasor será o valor *rms* (*root mean square*) do sinal senoidal. O diagrama apresentado na Figura 21 representa um circuito série de corrente alternada, denominado circuito $R-L$.

Figura 21: Circuito série R−L de corrente alternada.



Para o desenvolvimento do equacionamento, serão usados os conceitos de fasores e da notação fasorial e atribuídos aos elementos, do circuito da Figura 21, os valores abaixo:

$$f = 60\,[\text{Hz}],\ \omega = 2 \cdot \pi \cdot f \Rightarrow\ \omega = 376{,}99\,[\text{rad/s}] \quad (20)$$
$$e(t) = 179{,}6\,\text{sen}(376{,}99 \cdot t) \Rightarrow \mathbf{E} = 127\angle 0^\circ\,[\text{V}] \quad (21)$$
$$R = 1\,\Omega \Rightarrow \mathbf{Z_R} = 1\angle 0^\circ\,[\Omega] \quad (22)$$
$$L = 1 \cdot 10^{-2}\,[\text{H}] \Rightarrow \mathbf{Z_L} = 3{,}77\angle 90^\circ\,[\Omega] \quad (23)$$

Os fasores são utilizados para representar as tensões e correntes senoidais, bem como as impedâncias e reatâncias. A resistência não é considerada um fasor pois não há variação do seu valor com o tempo [10], mas será usada aqui para efeitos de cálculos. A utilização da representação fasorial permite que sejam aplicadas, aos circuitos *CA*, as Leis de Kirchhoff para realizar as análises de tensões e correntes resultantes da mesma forma que na análise *CC*. No caso do circuito mostrado na Figura 21 será demonstrada a aplicação da interface de cálculo para determinar a corrente total fornecida pela fonte, as tensões nos elementos e a impedância total vista pela fonte. Inicialmente será determinada a impedância total (**ZT**) associada ao circuito como mostrado na Equação (24).

$$\mathbf{Z_T} = \mathbf{Z_R} + \mathbf{Z_L} \quad (24)$$
$$\mathbf{Z_T} = 1\angle 0^\circ + 3{,}77\angle 90^\circ = 3{,}9\angle 75{,}14^\circ\,[\Omega]$$

Após a determinação do valor da impedância total (**ZT**), a corrente total fornecida pela fonte (**IF**) é calculada conforme a Equação (25).

$$\mathbf{I_F} = \frac{\mathbf{E}}{\mathbf{Z_T}} \quad (25)$$
$$\mathbf{I_F} = \frac{127\angle 0^\circ}{3{,}9\angle 75{,}14^\circ} = 32{,}56\angle -75{,}14^\circ\,[\text{A}]$$

Uma vez obtido o valor para a corrente, as tensões nos elementos do circuito da Figura 21 são calculadas conforme as Equações (26) e (27).

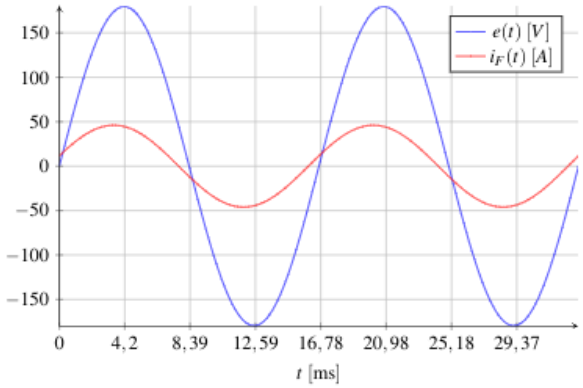
$$\mathbf{V_R} = \mathbf{I_F} \cdot \mathbf{R} \quad (26)$$
$$\mathbf{V_R} = 32{,}56\angle -75{,}14^\circ \cdot 1\angle 0^\circ = 32{,}56\angle -75{,}14^\circ\,[\text{V}]$$
$$\mathbf{V_L} = \mathbf{I_F} \cdot \mathbf{L} \quad (27)$$
$$\mathbf{V_L} = 32{,}56\angle -75{,}14^\circ \cdot 3{,}77\angle 90^\circ = 122{,}75\angle 14{,}86^\circ\,[\text{V}]$$

No domínio do tempo, a tensão da fonte (*e*(*t*)) e corrente total (*iF* (*t*)) são expressas pelas Equações (28) e (29) respectivamente.

$$e(t) = 179,6 \cdot \text{sen}(376,99 \cdot t)\ [\text{V}] \quad (28)$$
$$i_F(t) = 46,05 \cdot \text{sen}(376,99 \cdot t - 75,14)\ [\text{A}] \quad (29)$$

A representação gráfica do comportamento no domínio do tempo, das tensões e correntes, é um recurso de análise que auxilia na interpretação e na validação dos resultados. A Figura 22 apresenta o comportamento, no domínio do tempo, da tensão fornecida pela fonte ($e(t)$) e da corrente total fornecida (iF (t)), comprovando o efeito provocado pelo indutor no circuito, o qual produz um atraso de fase da corrente em relação à tensão [10].

Figura 22: Formas de onda para o circuito série R−L da Figura 21.



Os resultados numéricos apresentados nas Equações (20) a (27), foram obtidos pela inserção, no documento, do código LATEX conforme mostrado na Figura 23. Foram utilizadas as funções da interface de cálculo desenvolvidas para operações com números reais (Figura 2) e para operações com números complexos (Figuras 7 e 13). Para a visualização dos valores numéricos, associados às equações, foram utilizadas as funções desenvolvidas para esta finalidade conforme apresentado na Figura 12.

Figura 23: Código LATEX para efetuar as operações matemáticas relativas à análise do circuito série R−L.

```
\calcula{\erms}{127} % atribui valor fonte
\calcula{\R}{1}      % atribui valor R
\calcula{\L}{10e-3}  % atribui valor L
\calcula{\ePico}{\erms*sqrt(2)} % calcula valor pico
\calcula{\fang}{2*pi*60}        % calcula freq. angular
\complexo{\efasor}{\erms}{0}    % cria fasor da tensao
\complexo{\Rfasor}{\R}{0}       % cria fasor de R
\complexo{\Lfasor}{0}{\fang*\L} % cria fasor de L
\complexoSoma{\Ztotal}{\Rfasor}{\Lfasor} % calcula Z total
\complexoDiv{\ifonte}{\efasor}{\Ztotal}  % calcula I
\complexoMul{\vrfasor}{\ifonte}{\Rfasor} % calcula V de R
\complexoMul{\vlfasor}{\ifonte}{\Lfasor} % calcula V de L
```

A representação gráfica do comportamento no domínio do tempo (Figura 22) foi obtida com a inserção do código LATEX apresentado na Figura 24. Foi utilizado o ambiente **{tikzpicture}**, conforme apresentado na Seção 3.4, sendo criada uma função denominada ***funcaosenoidal()***, que possui três parâmetros de entrada: valor de pico, frequência angular, defasagem e tempo. O pacote PGF/Ti*k*Z, por padrão, assume que os argumentos para as funções trigonométricas são informados em graus, é necessário incluir nos argumentos de ângulo, caso necessário, que os mesmos estão sendo informados em radianos, incluindo o sufixo "r" após o argumento. A faixa de tempo para análise foi ajustada no parâmetro *domain*=0:0.035, visando abranger dois ciclos da função senoidal, considerando a frequência adotada de 60 *Hz*. As formas de onda ($e(t)$) e ($iF\ (t)$) obtidas no gráfico através do comando **\plot** utilizaram a mesma função de cálculo (***funcaosenoidal()***), sendo modificados os parâmetros de entrada conforme as variáveis associadas a cada grandeza em análise. Os argumentos utilizados são as variáveis resultantes dos cálculos realizados com a nova interface de cálculo proposta, conforme o código mostrado na Figura 23.

Figura 24: Código LATEX para realizar a visualização do comportamento no domínio do tempo.

```
\begin{tikzpicture}[
    declare function={
    funcaosenoidal(\vm,\freq,\defasagem,\t)=\vm*sin...
    ((\freq*\t)r+\defasagem r);}
    ]
    \begin{axis}[axistotime,domain=0:0.034,xlabel=$t$,
    legend pos=north east]
        \plot [blue]{funcaosenoidal(\efasor{3}*sqrt(2),...
        \fang,\efasor{4},x};
        \plot [red]{funcaosenoidal(\ifonte{3}*sqrt(2),...
        \fang,\ifonte{4},x};
        \legend{$e(t) [V]$,$i_F(t) [A]$}
    \end{axis}
\end{tikzpicture}
```

Os resultados obtidos e apresentados são condizentes com as condições estabelecidas, principalmente em relação à precisão utilizada, e apresentam valores numéricos compatíveis com aqueles que seriam obtidos utilizando outras aplicações computacionais ou calculadoras eletrônicas.

4 CONCLUSÃO

Este trabalho apresentou uma nova interface de cálculo para o Sistema de Preparação de Documentos TEX/LATEX utilizando o pacote gráfico PGF/Ti*k*Z, cujas funcionalidades foram exploradas com aplicações na Análise de Circuitos Elétricos. Foram apresentados resultados numéricos, inseridos no documento em elaboração, utilizando a nova interface de cálculo e cujos valores apresentam exatidão quando comparados àqueles que seriam obtidos por aplicações

computacionais voltadas à execução de cálculos ou através do uso de calculadoras eletrônicas. O uso e as funcionalidades da interface implementada foram comprovados através de exemplos didáticos, demonstrando recursos que facilitam a inserção de cálculos matemáticos diretamente no documento em elaboração, tornando-se uma alternativa viável e apropriada para otimizar o processo de edição com a utilização dos resultados de cálculos e operações matemáticas em qualquer etapa do documento, bem como a reutilização de tais valores diretamente ou através de funções.

REFERÊNCIAS

1. Peruzzo J. Uso Do Latex Na Elaboração De Trabalhos Acadêmicos. 1ª ed. Concórdia: Edição do Autor; 2020. 420p.

2. Knuth DE, Bibby DR. The TeXbook. 20th ed. New York: Addison-Wesley Publishing Copany; 1991. 483p.

3. Flom PL. LATEX for academics and researchers who (think they) don't need it. The PracTEX Journal. 2005;28(4):1–9.

4. Cherem YA. A utilização do sistema de preparação de textos LaTeX na produção de textos acadêmicos no Brasil: uma investigação preliminar e perspectivas. Informação & Informação. 2015 jun;20(1):228, doi: 10.5433/1981-8920.2015v20n1p228

5. Lobão Neto RA, Neto MCM. IFBATEX - Ambiente WEB para edição de documentos LATEX. Revista de Sistemas e Computação - RSC. 2014:38–52.

6. Abraham ER, Teixeira Machado S, Mendes dos Reis JG, Franco Gonçalves R, Terra da Silva M. Modelo LaTeX para teses e dissertações em Programa de Pós-Graduação: construção e avaliação de artefato. AtoZ: novas práticas em informação e conhecimento. 2015;4(2):84, doi: 10.5380/atoz.v4i2.43554

7. Araujo LC. O pacote abntex2cite. CTAN; 2018[acesso em 12 ago 2020]. Disponível em: https://www.ctan.org/pkg/abntex2.

8. Redaelli MA, Lindner S, Erhardt S, Giannetti R. CircuiTikZ. CTAN; 2020[acesso em 20 jul 2020]. Disponível em: http://mirrors.ibiblio.org/CTAN/graphics/pgf/contrib/circuitikz/doc/circuitikzmanual.pdf.

9. Tantau T. The TikZ and PGF Packages. CTAN; 2020[acesso em 20 mai 202]. Disponível em: http://mirrors.ibiblio.org/CTAN/graphics/pgf/base/doc/pgfmanual.pdf.

10. Boylestad RL. Introdução à análise de circuitos. 12th ed. São Paulo: Pearson Prentice Hall; 2012. 980p.

11. Mertz A, Slough W. A TikZ tutorial: Generating graphics in the spirit of TEX. TUGboat. 2009;30(2):214–226.

12. Surhone LM, Tennoe MT, Henssonow SF. Pgf/ Tikz. Betascript Publishing; 2010. 76p.

13. Kottwitz S. LaTeX Beginner's Guide Create high-quality and professional-looking texts, articles, and books for business and science using LaTeX. 1ª ed. Birminghan: Packt Publishing ltd; 2011. 336p.

14. Gilat A, Subramaniam V. Métodos Numéricos para Engenheiros e Cientistas: Uma Introdução com Aplicações Usando o MATLAB. Porto Alegre: Bookman Companhia Ed; 2008. 480p.

15. Feuersänger C. Manual for Package PGFPLOTS. CTAN; 2020[acesso em 20 jul 2020]. Disponível em: http://mirrors.ibiblio.org/CTAN/graphics/pgf/contrib/pgfplots/doc/pgfplots.pdf.