



## **DESENVOLVIMENTO DE INTERFACE EM MATLAB PARA APRENDIZADO E COMPARAÇÃO DE MÉTODOS NUMÉRICOS**

**Giovane Malcher da Silva** – giovanemalcher.s@gmail.com

Universidade Federal do Pará

Rua Augusto Côrrea

66075-110 – Belém – Pará

**Duaymy Bruno** – duaymy@gmail.com

Universidade Federal do Pará

Rua Augusto Côrrea

66075-110 – Belém – Pará

**Wesley Barbosa Tavares** – wbartav@gmail.com

Universidade Federal do Pará

Rua Augusto Côrrea

66075-110 – Belém – Pará

**Danielle Lima Guedes** – dani.lguedes@hotmail.com

Universidade Federal do Pará

Rua Augusto Côrrea

66075-110 – Belém – Pará

**Orlando Fonseca Silva** – orfosi@ufpa.br

Universidade Federal do Pará

Rua Augusto Côrrea

66075-110 – Belém – Pará

**Resumo:** *O objetivo deste trabalho é desenvolver uma interface gráfica amigável capaz de melhorar o ensino de cálculo numérico, além de simplificar a resolução de problemas de engenharia envolvendo a aplicação de métodos numéricos. Neste sentido, fez-se o uso do software MATLAB para implementar os métodos e criar um meio interativo para a inserção dos dados referentes aos problemas, que mostre as soluções de uma forma clara, com informações que permitam não apenas melhorar a compreensão de cada método mas também avaliar seu desempenho. Assim, propõe-se uma aplicação prática do conhecimento teórico de cálculo numérico adquirido pelo estudante de engenharia. A interface criada foi submetida a testes baseados em exemplos simples e diferentes métodos foram comparados na busca de uma mesma solução.*

**Palavras-chave:** *Cálculo Numérico; interface gráfica; ferramenta didática.*



## 1. INTRODUÇÃO

A solução de problemas físicos baseia-se, primeiramente no estabelecimento do problema, posteriormente em sua modelagem e com isso, a obtenção do modelo matemático. A fase seguinte consiste na resolução do modelo obtido, ou seja, mostrar se ele tem solução ou não e se sua solução é única. Feito isso, busca-se a solução final. Para tal, usualmente são aplicados métodos analíticos. No entanto, em inúmeras situações práticas esse caminho de solução apresenta-se de difícil desenvolvimento. Uma via alternativa para esse tipo de problema é o Cálculo Numérico, o qual tem sua importância centrada no fato de que, mesmo quando a solução analítica é de difícil aquisição, as técnicas numéricas podem ser empregadas sem maiores dificuldades. No entanto, durante o emprego de tais métodos de forma manual, as frequentes mudanças de parâmetros, combinadas com um amplo número de iterações constituem uma dificuldade. A utilização de recursos computacionais permite contornar tal empecilho, além de proporcionar maior precisão devido a sua velocidade, possibilitando um número de iterações bastante elevado. Dessa forma, o presente trabalho, tem por objetivo, desenvolver uma interface gráfica amigável capaz de otimizar a utilização de métodos numéricos, além de simplificar a resolução de problemas de engenharia envolvendo suas aplicação.

## 2. ASPECTOS TEÓRICOS

A disciplina de Cálculo Numérico tem por objetivo apresentar ao aluno algoritmos e métodos pelos quais se podem resolver modelos e sistemas matemáticos que não apresentam solução precisamente definida ou por rápida análise, devido a sua complexidade. Para fins didáticos, os métodos numéricos reunidos na interface gráfica criada foram divididos em categorias, de acordo com sua aplicabilidade.

### 2.1. Métodos para Obtenção de Raízes Reais

A aquisição das raízes de certas funções não tradicionais é um trabalho árduo e, por vezes, impossível. Os métodos desta categoria são comumente utilizados para encontrar as raízes de funções complicadas. São eles os métodos da bissecção, da posição falsa, de Newton-Raphson e das secantes. Os dois primeiros são robustos e mais lentos que os outros. O método de Newton é rápido e preciso, mas utiliza a derivada da função abordada, o que, na prática, pode impor dificuldades em seu uso. O método das secantes é rápido e não se vale da derivada da função abordada.

### 2.2. Métodos para Resolução de Sistemas de Equações

As incógnitas de sistema de equações de baixa ordem são facilmente encontradas por inspeção ou métodos simples, como o da adição e da substituição. Entretanto, matrizes de ordem elevada dificultam a realização deste tipo análise, surgindo então a necessidade de métodos para resolução de sistemas de equações independentes da ordem. Neste trabalho, os métodos presentes, desta categoria, são: método de Gauss-Jacobi, Gauss-Seidel, eliminação de Gauss-Jordan, fatoração LU, regra de Cramer e Inversa da matriz.



### **2.3. Métodos para Aproximação de Funções**

Em experimentos práticos, os dados obtidos a partir de amostragens são pontuais e descontínuos. Existe, portanto, a necessidade de se descobrir uma função com a qual possamos dar a este fenômeno a continuidade desejada. Os métodos utilizados neste trabalho para esta aplicação são a interpolação polinomial, as séries de Taylor e o método dos mínimos quadrados. Outra aplicação dos métodos desta categoria é a simplificação de funções complexas

### **2.4. Métodos Para Diferenciação Numérica**

O método da diferenciação numérica obtém o valor numérico da derivada de uma função dado um ponto, através do quociente entre a diferença de dois valores da função para pontos distintos (sendo um destes, o ponto em questão), e a distância entre estes dois pontos. Embora para funções simples tenha pouca aplicação prática, para funções mais complexas pode ser de grande valor.

### **2.5. Métodos Para Integração Numérica**

Algumas funções são de difícil integração, ou seja, a expressão analítica de sua primitiva é desconhecida. Em aplicações práticas que haja a necessidade do resultado da integral definida de uma função deste tipo, os métodos de integração numérica são utilizados para encontrar uma área específica, delimitada pela função. Os métodos de integração numérica contidos neste trabalho são a Regra do Ponto Médio, a Regra do Trapézio e a Regra de Simpson.

## **3. CONSTRUÇÃO DE INTERFACES EM MATLAB**

MATLAB é uma linguagem de alto nível e um ambiente interativo para computação numérica, visualização e programação. Usando MATLAB, é possível analisar os dados, desenvolver algoritmos e criar modelos e aplicações. A linguagem, ferramentas e funções matemáticas embutidas permitem-lhe explorar múltiplas abordagens e chegar a uma solução mais rápida do que com planilhas ou linguagens de programação tradicionais, como C / C++ ou Java. O MATLAB pode ser usado para uma variedade de aplicações, incluindo processamento de sinais e comunicação, imagem e processamento de vídeo, sistemas de controle, teste e medição, finanças computacionais, e biologia computacional. Além de todas essas funções há também a *toolbox* para construção de interfaces, ferramenta esta que permite com que sejam construídas ferramentas iterativas para certas aplicações, não sendo necessário dessa forma que o usuário digite linhas de comando ou mesmo conheça a complexidade por traz do código associado à aplicação.

### **3.1 . Construção Das Interfaces de Cada Método**

A primeira etapa do projeto baseou-se na elaboração de rotinas, utilizando-se o ambiente MATLAB, referentes a 22 métodos. Estes podem ser agrupados em cinco categorias. A primeira é formada por métodos iterativos utilizados na obtenção das raízes reais, sendo eles os métodos da bissecção, da posição falsa, de Newton-Raphson e da secante. A segunda é composta dos métodos utilizados para resolução de sistemas de equações, os quais podem ser



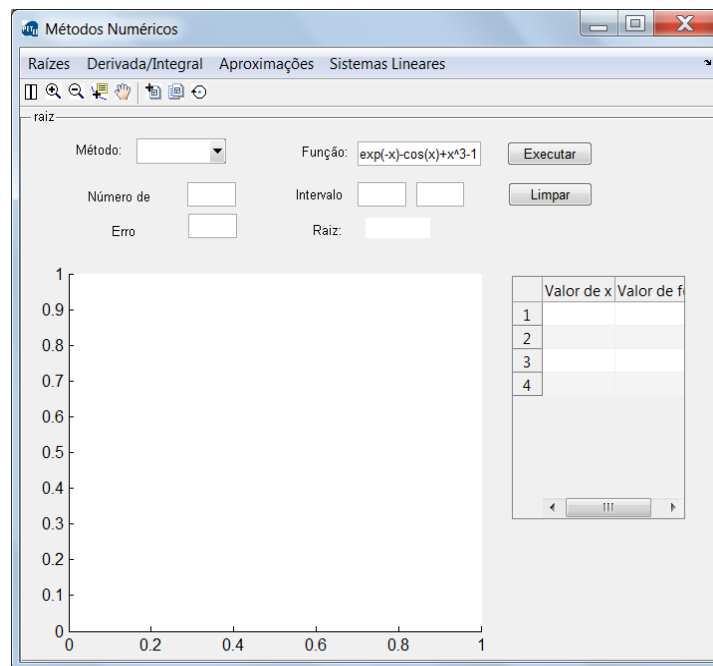
iterativos lineares (métodos de Gauss-Jacobi, Gauss-Seidel) ou diretos (eliminação de Gauss-Jordan, fatoração LU, regra de Cramer e Inversa da matriz). A terceira abrange métodos utilizados para aproximação de funções, nas quais são conhecidos apenas seus valores numéricos, por meio da interpolação polinomial, séries de Taylor e mínimos quadrados. As duas últimas categorias abrangem a diferenciação e integração numérica. A segunda etapa focou-se no desenvolvimento de interfaces gráficas de acordo com cada categoria de rotina. Para isso, utilizou-se a ferramenta *guide*, a qual inicia um editor interativo para criação e edição de interfaces, apresentando um conjunto de ferramentas que facilitam a manipulação dos objetos necessários ao seu desenvolvimento.

### 3.2 Agrupamento do Conjunto de Interfaces

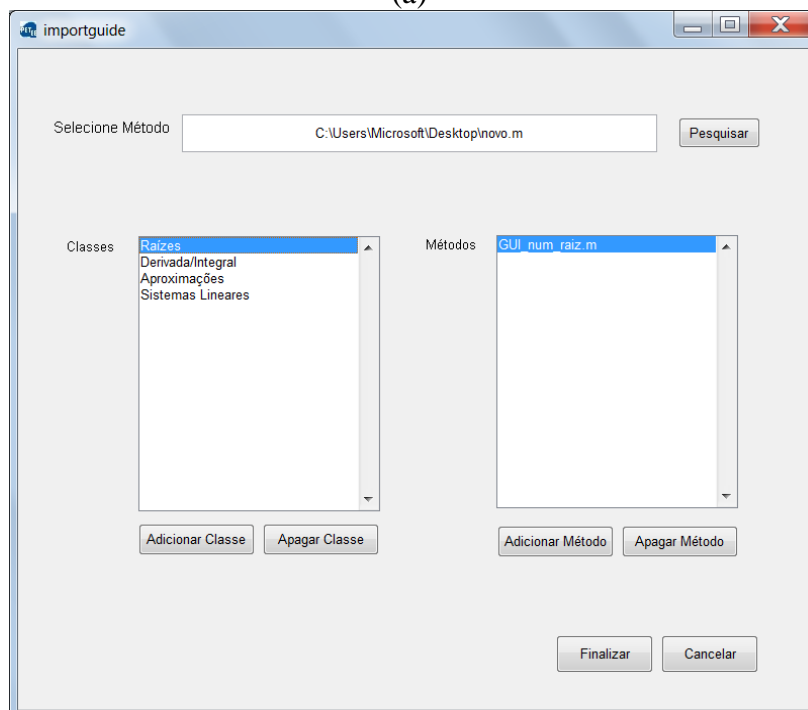
A princípio, devido ao grande número de métodos, foram desenvolvidas interfaces em janelas gráficas individuais para posteriormente realizar-se a integração de todas em uma única janela, permitindo ao usuário a utilização, paralelamente, de métodos diferentes.

A estrutura criada para a integração dos métodos consistiu na utilização de um grupo de painéis, sendo que, cada painel abriga a interface de um determinado método. No MATLAB, a função *uipanel* permite a criação de um objeto do tipo painel, que funciona como um elemento que agrupa botões, eixos e tabelas, sendo esses últimos associados ao primeiro. Desse modo, torna-se possível a habilitação de um painel e a omissão dos outros de acordo com a determinação do usuário. Essa estrutura permite a existência de várias interfaces em uma mesma janela gráfica, as quais são gerenciadas com base nas propriedades do painel. Para a exibição de um método em específico, utilizou-se a função *uimenu*, a qual cria uma barra de menus, sendo que nesta são estabelecidas hierarquias de menus e submenus, mostrados na janela gráfica principal e sua organização se deu de forma que, cada menu corresponde a uma categoria de métodos.

Concluída a fase de gerenciamento para a exibição de painéis, iniciou-se a construção de uma barra de ferramentas, a qual disponibiliza opções para análise de resultados obtidos e edição da interface principal. A princípio disponibilizou-se a possibilidade de visualização de dois painéis em simultâneo, o que provê a comparação de resultados com dois métodos diferentes. Além disso, buscou-se disponibilizar funções tais como, zoom para ampliar e diminuir a visão de eixos, cursor de dados para análise ponto a ponto de curvas e a ferramenta mão que desloca a visão de eixos bidimensionalmente. Todas as ferramentas mencionadas são para análise de resultados, no entanto, além dessas, foram desenvolvidas opções para customização da interface principal. O primeiro item consistiu na possibilidade de inserir novos métodos, recurso este que, propicia ao usuário desenvolver seus próprios métodos e com isso, incrementar a ferramenta em questão, aprimorando sua utilidade na solução de problemas de ordem prática, assim como sua aplicabilidade educacional. Para isso, foi desenvolvido um assistente, no qual se permite adicionar novas classes e métodos. Ao ser inserido um novo método, é necessário apenas fornecer o arquivo *mfile* contendo sua interface e programação. Outra opção implementada permite a exportação de métodos, ou seja, a criação de um arquivo contendo a interface de um determinado método em uma janela gráfica individual. Por fim, foi criada uma opção que restitui os métodos iniciais, excluindo todos aqueles adicionados. A interface principal, assim como o assistente para sua edição são apresentados na Figura 1.



(a)



(b)

Figura I: (a) Interface Principal e o (b) assistente para sua edição.

Com a interface principal construída, iniciou-se a inserção das classes e métodos através do assistente. A primeira classe trata de métodos iterativos para o cálculo de raízes reais, e de acordo com os testes realizados na interface gráfica, os resultados obtidos foram aproximados, concluindo assim que a programação dos métodos que foi empregada para obtenção do

resultado final foi adequada. A segunda classe agrupou os métodos para integração e diferenciação numérica. Na interface gráfica é possível fazer a diferenciação numérica por três métodos, estes são o Método da Diferença Central, Progressiva e Regressiva. No caso da integração numérica também existem três possíveis métodos de fazer a integração, que são o Método do Trapézio, o Método de Simpson e o Método de Simpson Repetido. Na terceira categoria foi criada uma classe para os métodos de aproximação e interpolação, contando com uma aba para os discretos. No caso dos métodos discretos, criou-se uma tabela para inserção das coordenadas dos pontos a serem utilizados, havendo a alternativa de se carregar um arquivo com seus valores por meio do botão *Load*. Com base nessas coordenadas, é calculada e plotada, em um eixo comum aos pontos, uma função de acordo com o algoritmo escolhido pelo usuário: pode ser uma função polinomial, exponencial ou de potência que melhor aproxima os pontos pelo método dos mínimos quadrados ou uma função polinomial que os interpola pelos métodos de Newton, de Lagrange ou da resolução do sistema linear correspondente, havendo uma tabela para mostrar informações sobre as iterações. A última categoria agrupou métodos diretos e iterativos para solução de sistemas de equações lineares. Para os métodos diretos a interface gráfica construída foi limitada para sistemas de até três equações e três incógnitas. Nesta aplicou-se um exemplo de sistemas, e as soluções obtidas foram iguais em todos os métodos dessa categoria. Nos iterativos, a interface apresentou espaço para o usuário disponibilizar não só o sistema, como também um valor inicial e o erro relativo. As Figuras 2 e 3 apresentam exemplos das interfaces construídas, sendo que primeira apresenta o conjunto de métodos para cálculo de raiz e a segunda para aproximação de funções. Pode-se perceber que sua utilização é bastante intuitiva e simples.

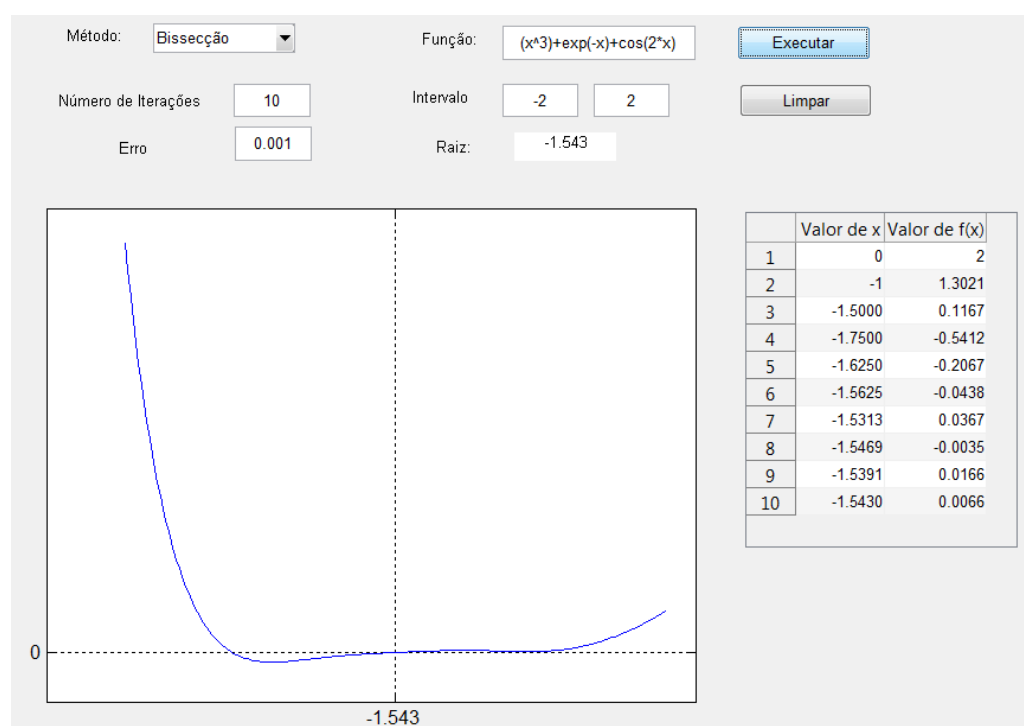


Figura 2: Cálculo de raiz através do método da Bisseção.

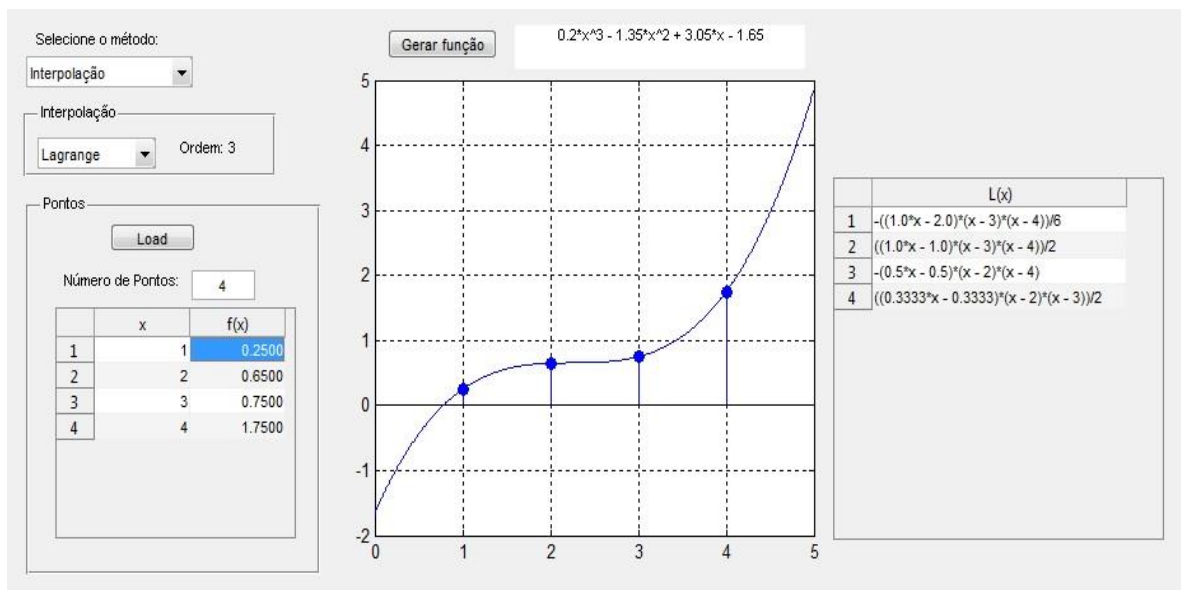


Figura 2: Cálculo de função de interpolação pelo método de Lagrange.

## 4. RESULTADOS

### 4.1. Aplicação didática e atividades propostas

A metodologia foi aplicada ao grupo de 12 alunos do 3º semestre do curso de Engenharia Elétrica da UFPA, como atividade complementar. Na aula explanatória foi destacada a importância da disciplina de Cálculo Numérico, expondo suas possíveis aplicações, e frisada a minúcia necessária durante o emprego dos métodos abrangidos pela matéria.

O roteiro do experimento complementou a metodologia aplicada, contendo atividades propostas, que foram baseadas na aula explanatória. A atividade proposta consistia em achar as raízes de certa função utilizando métodos iterativos, dado o número de iterações, chute inicial e erro desejado.

### 4.2. Análise dos questionários

A aplicação dos questionários tinha como principal objetivo avaliar a eficiência do método aplicado, e para isso verificou-se as opiniões dos alunos sobre os conhecimentos a respeito de Cálculo Numérico, assim como a compreensão dos mesmos no que condizem com as simulações realizadas nas atividades propostas da aula explicativa. A partir disso, foram coletados os dados e foram exibidos em forma de gráficos.

A primeira pergunta se referiu a opinião dos alunos quanto à capacidade do software em aprimorar seus conhecimentos de Cálculo Numérico, e todos os participantes responderam afirmativamente. A segunda pergunta buscou conhecer o impacto causado pela ferramenta no conhecimento dos alunos, e 66,67% dos graduandos responderam que o impacto foi muito significativo, ao passo que 33,33% disseram que o software ajudou a esclarecer algumas dúvidas sobre o conteúdo. A figura 3 mostra o gráfico com a porcentagem de respostas a esta pergunta.

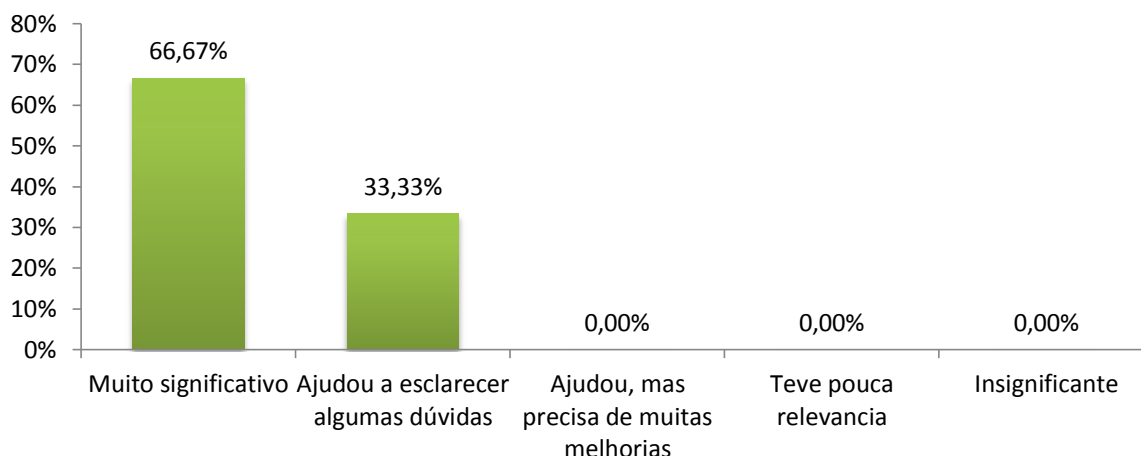


Figura 3 – Resultados da pergunta “Como você avalia o impacto desta ferramenta em seus conhecimentos?”.

A terceira pergunta tinha por finalidade saber como se deu a interação entre o aluno e a interface, quanto a sua facilidade de manuseio. Desta vez, 75% dos alunos consideraram a interface bastante intuitiva, enquanto 25% consideraram-na simples, mas se fosse lhes dada às devidas orientações. A figura 4 mostra o gráfico com a porcentagem de respostas a esta pergunta.

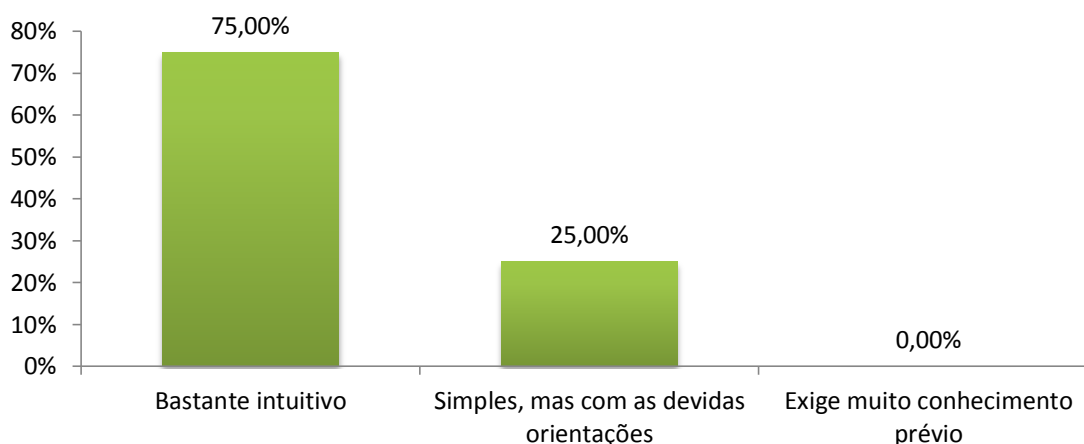


Figura 4 – Respostas a pergunta “Como você avalia o uso da interface?”.

## 5. CONSIDERAÇÕES FINAIS

De acordo com os resultados obtidos na avaliação dos alunos, pode-se constatar que o *software* desenvolvido apresenta relevância quanto ao auxílio do entendimento de tópicos teóricos de cálculo numérico ou mesmo na solução de problemas em geral. A ferramenta desenvolvida é de fácil compreensão e grande utilidade ao usuário com interesse no assunto.





A possibilidade de inserção de outros métodos na interface também amplia os horizontes de pesquisa, pois permite ao usuário utilizar novos algoritmos com as mesmas facilidades apresentadas pelo *software*.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

RUGGIERO, M. A. G., LOPES, V.L.R. Cálculo Numérico: Aspectos Teóricos e Computacionais - Pearson, 1996. 422 p.

SPERANDIO, D., MENDES, J. T., MONKEN, L. H. Cálculo Numérico: Características Matemáticas e Computacionais dos Métodos Numéricos – Pearson, 2003. 368 p.

MathWorks Products and Services. Disponível em: <<http://www.mathworks.com>>.

### INTERFACE DEVELOPMENT IN MATLAB FOR LEARNING AND COMPARISON OF NUMERICAL METHODS

**Abstract:** *The objective of this work is to develop a user-friendly graphical interface to improve the teaching of numerical calculation, in addition to simplifying the solution of engineering problems involving the application of methods numbers. So, did the use of MATLAB software to implement the methods and create an interactive tool for integrating data show that the solutions in a clear way, with information enabling not only improve understanding of each method and also to evaluate its performance. Thus, we propose a practical application of theoretical knowledge acquired by the numerical calculation of an engineering student. The interface created was subjected to tests based on simple examples and different methods were compared in the search for the same solution.*

**Key-words:** *numerical calculus; graphical interface; didactic tool.*