INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO

RELATÓRIO FINALDEINICIAÇÃO CIENTÍFICA

CAMPUS VOTUPORANGA

SITE PARA AUXÍLIO NO ESTUDO DO CÁLCULO NÚMERICO NA ENGENHARIA

ALUNO: ISABELA CASSIA DOMINICAL PARRA

ORIENTADOR: GUSTAVO CABRELLI NIRSCHL

MODALIDADE DE IC: PIBIFSP

VIGENCIA: 01/03/2016 A 30/11/2016

NOVEMBRO DE 2016.

RESUMO

Atualmente, a maioria dos softwares apresenta somente a resposta final, escondendo toda a rotina de cálculo. Objetivando ajudar alunos de Engenharia e auxiliar professores em aula, estão sendo desenvolvidos alguns programas disponibilizados on-line que auxiliam no estudo de Cálculo Numérico aplicado à Engenharia Civil. Diferentemente da maioria dos softwares, estes têm como foco principal contribuir com os estudos, sendo que é possível gerar um relatório PDF explicativo contendo a resolução interna realizada pelo programa, uma breve explicação da teoria envolvida e o algoritmo principal do método exposto. Foram desenvolvidos programas para a resolução de sistemas lineares por Fatoração LU com pivoteamento parcial, por Gauss-Jacobi e por Gauss-Seidel, para a interpolação e aproximação de dados, e também para integração e derivação numérica. Os programas estão disponibilizados no próprio site da instituição (Instituto Federal de São Paulo - Campus Votuporanga), sendo parte do grupo de pesquisa cadastrado no CNPq: NEV (Núcleo de Engenharia Virtual).

Palavras-chave:*Cálculo Numérico. Fatoração LU. Gauss- Seidel. Gauss- Jacobi. Interpolação.Aproximação.IntegraçãoNumérica. DerivaçãoNumérica.*

ABSTRACT

Currently, most softwares shows only the final solution, hiding all routine calculation. Aiming to help engineering students and assist teachers in the classroom, they are being developed some online programs that assist in Numerical Calculus study applied to Civil Engineering. Unlike most, these softwaresare mainly focused on contributing to the studies, so it is possible to generate an explanatory PDF report containing the internal resolution performed by the program, a brief explanation of the theory involved, and the main algorithm of the performed method. Programs have been developed for solving linear systems LU factorization with partial pivoting for Gauss-Jacobi and Gauss-Seidel, for interpolation and approximation of data, and also for integration and numerical differentiation. Programs are available on the Web site of the institution (Instituto Federal de São Paulo - Campus Votuporanga), being part of the research group registered in the CNPq: NEV (Virtual Engineering Center).

Keywords: *Numerical calculation. LU factorization.Gauss-Seidel.Gauss-Jacobi.Interpolation.Approach. Numericalintegration. NumericalDerivation***.**

SUMÁRIO

LISTA DE FIGURAS

LISTA DE ABREVIATURAS

NEV: Núcleo de Engenharia Virtual

LU: LowerUpper

PVI: Problema de Valor Inicial

EDO: Equação diferencial ordinária

PVC: Problema de Valor de Contorno

LISTA DE SÍMBOLOS

a_{ij}Termo da matriz dos coeficientes

b_iTermo da matriz independente

*m*Fator da linha

*P*Polinômio interpolador

$a_n$coeficiente do polinômio

L Polinômio de Lagrange

n número de pontos menos 1

d diferenças dividas

h valor do intervalo

*ai*coeficientes do polinômioaproximador

CAPÍTULO 1

INTRODUÇÃO

Atualmente, com o avanço das tecnologias, é possível utilizar programas de computador comerciais ou criar softwares de acordo com a necessidade. A maioria dos softwares apresenta somente os resultados finais. Pode-se citar como exemplo o FTOOL (2012), programa de análise estrutural e o TQS (2015), sobre dimensionamento de concreto armado.

O grupo de pesquisadores que formam o NEV – Núcleo de Engenharia Virtual (http://vtp.ifsp.edu.br/nev) elabora, entre outros, páginas da internet com programas que, além de mostrar os resultados finais, apresentam todo o desenvolvimento dos cálculos, a teoria básica para entendimento do método e também o algoritmo principal utilizado, auxiliando alunos, profissionais recém-formados e professores que quiserem utilizar os programas no ensino-aprendizado.

No caso deste relatório, serão apresentadas páginas para a resolução de sistemas lineares por Fatoração LU com pivoteamento parcial, por Gauss-Jacobi, e por Gauss-Seidel, para a interpolação polinomial, interpolação por Lagrange e usando a Fórmula de Newton, aproximação de dados de função polinomial, exponencial e potencial, derivação numérica e integração numérica pela Regra dos Trapézios e pela Formula de Simpson, conteúdo que podem ser encontrados em Quadros e Bortoli (2009).

A evolução tecnológica motivou a utilização da programação direta em páginas de internet, por meio da linguagem HTML/Javascript, que pode ser estudada em W3... (2015).

CAPÍTULO 2

APRESENTAÇÃO TEÓRICA

Na engenharia, sabe-se que os métodos analíticos de resolução dos problemas nem sempre são eficazes. Nestes casos,utilizam-se métodos numéricos, que nem sempre são exatos, mas, se usados corretamente, têm suficiente precisão. Além disso, os métodos numéricossão usados na criação dos programas computacionais.

2.1. Sistemas Lineares por Fatoração LU

O Método da Fatoração LU é um método derivado do Método da Eliminação de Gauss que apresenta a vantagem de que o escalonamento é feito independentemente do vetor dos termos independentes. Um exemplo prático desta vantagem pode ser encontrado na Engenharia Civil. Para calcular os deslocamentos "x" dos nós de uma estrutura, um dos procedimentos é via Método dos Elementos Finitos, resolvendo um sistema linear da forma Ax=b, em que "A" é a matriz de rigidez, fixa para a estrutura, e o vetor "b" contém as ações externas, naturalmente variáveis. Sendo assim, pelo Método da Fatoração LU, é possível escalonar uma só vez a matriz "A" e utilizá-la para quaisquer vetores "b".

De acordo com Ruggiero (1996), no Método da Fatoração LU, para resolver um sistema linear Ax=b, realiza-se a fatoração A=LU, sendo que o sistema fica (LU)x=b. Se y=Ux, resolver o sistema é equivalente a resolver Ly=b e, depois, Ux=y.

LU é um termo que vem do inglês "lower" e "upper", já que o método se baseia na decomposição da matriz A na forma A = LU, onde "L" é uma matriz triangular inferior (todos os elementos acima da diagonal são nulos) com elementos da diagonal principal iguais a 1 e "U" é uma matriz superior (todos os elementos abaixo da diagonal são nulos).

Para encontrar “L” e “U”, faz-se o escalonamento da matriz “A”, levando em consideração o pivoteamento parcial, ou seja, fazendo a troca de linhas de forma que o pivô seja o maior valor em módulo da coluna. Antes de resolver o sistema Ly=b, as mesmas trocas de linhas realizadas na matriz “A” têm que ser feitas no vetor “b”. A matriz “A” escalonada é a própria matriz “U” e a matriz “L” contém os multiplicadores da matriz “A” escalonada abaixo da diagonal, também com as devidas trocas de linhas realizadas.

Abaixo segue o esquema da resolução de uma matriz 3x3 genérica:

$$\begin{bmatrix} a11 & a12 & a13 \\ a21 & a22 & a23 \\ a31 & a32 & a33 \end{bmatrix}$$ →matriz “A” referente ao sistema Ax=b

Escalonando a matriz “A”, têm-se:

- Primeira coluna

Coeficientes: $m_{21} = \frac{a_{21}}{-} \; e \; m_{31} = \frac{a_{31}}{-}$

Linha 2: $a'_{22} = a_{22} - m_{21} * a_{12}$

$a'_{23} = a_{23} - m_{21} * a_{13}$

Linha 3: $a'_{32} = a_{32} - m_{31} * a_{12} a'_{33} = a_{33} - m_{31} * a_{13}$

- Segunda coluna

Coeficientes: $m_{32} = \frac{a_{32}}{-}$

Linha 3: $a'_{33} = a_{33} - m_{32} * a_{23}$

$b'_{3} = b_{3} - m_{32} * b_{2}$

$$\begin{bmatrix} 1 & 0 & 0 & b1 \\ m21 & 1 & 0 & b2 \\ m31 & m32 & 1 & b3 \end{bmatrix}$$ →matriz aumentada referente ao sistema Ly=b

$$\begin{bmatrix} a11 & a12 & a13 & y1 \\ 0 & a'22 & a'23 & y2 \\ 0 & 0 & a'33 & y3 \end{bmatrix}$$ →matriz aumentada referente ao sistema Ux=y b

Para achar os valores de x, é necessário que se resolva o sistema Ly=b, encontrando assim os valores de y e, depois, por retro substituição, que se resolva o sistema Ux=y.

Vale destacar que, no exemplo genérico anterior, não houve troca de linhas (pivoteamento parcial). Se houver troca de linhas no escalonamento de “A”, a mesma troca deve ser feita no vetor “b”.

2.2. Sistemas Lineares Via Gauss-Jacobi

Para resolver um sistema linear pelo método de Gauss-Jacobi, é necessário isolar o “x” de todas as equações, de acordo com os elementos da diagonal principal, como mostrado a seguir:

$$\{ \qquad a_{22} \quad a_{22} \qquad .$$

Para sistemas que apresentem zeros nas diagonais principais, pode-se permutar linhas até que todos os elementos da diagonal sejam não-nulos, pois, com elementos nulos nas diagonais, as divisões teriam o zero como denominador.

Para prosseguir a resolução, é necessário chutar um valor inicial para todos os “x”. Feito isso, basta ir substituindo os valores nas equações e ir encontrando novos valores, que serão utilizados na próxima iteração.A resolução chega ao fim quando as soluções convergirem segundo um erro menor que o estabelecido.Para os casos de soluções não convergirem,torna-se necessárioum número máximo de iterações.

2.3. Sistemas Lineares Via Gauss-Seidel

O Método de Gauss-Seidel tem quase o mesmo esquema de solução do Método de Gauss-Jacobi, sendo que os “x” anteriores dentro da iteração já resolvidos devem ser utilizados na iteração atual, e não os da iteração anterior.

2.4. Interpolação Polinomial

Segundo Quadros e Bortoli (2009), a interpolação de (n+1) de pontos do tipo (x, f(x)) leva a um polinômio p(x) de grau menor ou igual a n, na forma da Eq. 1 a seguir.

$$p(x) = a_0 + a_1 * x + a_2 * x^2 + \ldots + a_n * x^n$$

Desta maneira, conforme os autores citados, a obtenção do polinômio parte de um sistema de (n+1) equações da forma mostrada a seguir:

Pode-se usar qualquer método numérico para resolver o sistema. No programa referente a este relatório, o sistema é resolvido por Eliminação de Gauss.

2.5. Interpolação por Lagrange

Por este método, o polinômio interpolador é suposto de uma forma que depende de outros polinômios, assim como mostrado a seguir:

$$p_n(x) = y_0 L_0(x) + y_1 L_1(x) + \cdots + y_n L_n(x)$$

No caso, n é o número de pontos a interpolar menos 1. Os polinômios“ L”, conhecidos como polinômios de Lagrange, são supostos da seguinte forma:

$$L_k(x) = \frac{(x - x_0)(x - x_1) \ldots (x - x_{k-1})(x - x_{k+1}) \ldots (x - x_n)}{(x_k - x_0)(x_k - x_1) \ldots (x_k - x_{k-1})(x_k - x_{k+1}) \ldots (x_k - x_n)}$$

O “xi” e o “yi” são justamente os pontos conhecidos, a partir dos quais se quer achar um polinômio interpolador.

2.6. Interpolação pela Fórmula de Newton

Por este método, o polinômio interpolador é suposto da seguinte forma, chamado de fórmula de Newton:

$$p_n(x) = d_0 + d_1(x - x_0) + d_2(x - x_0)(x - x_1) + \cdots + d_n(x - x_0)(x - x_1) \ldots (x - x_{n-1})$$

No caso, n é o número de pontos a interpolar menos 1. Os coeficientes “ dk“ são chamados de diferenças divididas, da forma detalhada abaixo:

$$k = 0 \rightarrow d_0 = f[x_0]$$

$k = 1 \rightarrow d_1 = f[x_1, x_0]$
$k = 2 \rightarrow d_2 = f[x_2, x_1, x_0]$
(...)
$k = n \rightarrow d_n = f[x_n, x_{n-1}, ..., x_0]$

Essas diferenças divididas, da forma “ $f[x_n, x_{n-1}, ..., x_0]$ ” são calculadas da seguinte maneira:

$f[x_0] = f(x_0)$
(...)
$f[x_n] = f(x_n)$

$$f[x_1, x_0] = \frac{f[x_1] - f[x_0]}{x_1 - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$
$$f[x_2, x_1] = \frac{f[x_2] - f[x_1]}{x_2 - x_1} = \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$
(...)
$$f[x_n, x_{n-1}] = \frac{f[x_n] - f[x_{n-1}]}{x_n - x_{n-1}} = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

$$f[x_2, x_1, x_0] = \frac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0} = \frac{calculados\_anteriormente}{x_2 - x_0}$$
$$f[x_3, x_2, x_1] = \frac{f[x_3, x_2] - f[x_2, x_1]}{x_3 - x_1} = \frac{calculados\_anteriormente}{x_3 - x_1}$$
(...)
$$f[x_n, x_{n-1}, x_{n-2}] = \frac{f[x_n, x_{n-1}] - f[x_{n-1}, x_{n-2}]}{x_n - x_{n-2}} = \frac{calculados\_anteriormente}{x_n - x_{n-2}}$$

E assim por diante.

2.7. Integração pela Regra dos Trapézios

A integral de uma função “ f ” no intervalo [a,b] pode ser aproximada pela área de um trapézio, conforme a figura a seguir:

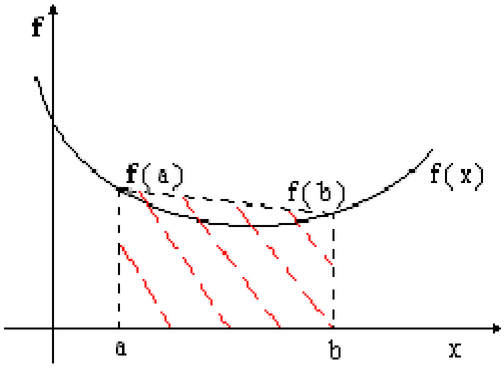


Figura: Área sob f(x) aproximada por um trapézio. Fonte: Quadros e Bortoli (2009).

Considerando o cálculo da área do trapézio mostrado na figura anterior, pode-se calcular a integral aproximada como:

$$\int_a^b f(x)\,dx \approx [f(a)+f(b)]\,\frac{b-a}{2}$$

A função f(x) e suas derivadas à primeira e à segunda, devem ser contínuas no intervalo [a,b] para que este processo seja válido.

Quanto mais se dividir a área em “n” trapézios de alturas iguais, mais preciso será o valor da integral. Ao fim deve se somar todas as áreas, o que pode ser simplificado pela equação a seguir:

$$T(f,h) = \frac{h}{2}\left[f(x_0) + 2\,\left(f(x_1) + \cdots + f(x_{n-1})\right) + f(x_n)\right]$$

Em que:
n é o número de intervalos que se quer usar
h = (b-a)/n
$f(x_0) = f(a)$
$f(x_n) = f(b)$
$x_i = x_{i-1} + h$

Se afunção estiver na forma de uma tabela de valores x,f(x), podem-se usar exatamente esses intervalos para aplicar a regra dos trapézios, não sendo necessário encontrar uma função aproximada. É preciso calcular a integral para cada intervalo, caso esses seja diferentes, e somar as integrais no final.

2.8. Integração pela Fórmula de Simpson

A integração pela Fórmula de Simpson utiliza-se de um polinômio de grau 2 que aproxima a função original f(x), segundo Ruggiero (2006), por interpolação polinomial de grau 2 com funções de Lagrange. Depois de simplificações a integral é dada pela fórmula abaixo:

$$\int_{x_0}^{x_2} f(x)\,dx \simeq \frac{h}{3}\left[f(x_0) + 4f(x_1) + f(x_2)\right]$$

Quanto maior o número de intervalos maior será a precisão da integral, Quadros e Bortoli (2009) apresentam o cálculo aproximado da integral de f(x), com “2n” intervalos constantes de amplitude “h” como:

$$S(f,h) = \frac{h}{3}\left\{\begin{array}{l} f(x_0) + 4\left[f(x_1) + f(x_3) + f(x_5) + \ldots + f(x_{2n-1})\right] \\ + 2\left[f(x_2) + f(x_4) + f(x_6) + \ldots + f(x_{2n-2})\right] + f(x_{2n}) \end{array}\right\}$$

Portanto, se quisermos integrar uma função f(x) de a até b, usando intervalos constantes de amplitude “h”:

2n é o número de intervalos que se quer usar (quanto mais, melhor)
h = (b-a)/2n
$f(x_0) = f(a)$
$f(x_{2n}) = f(b)$
$x_i = x_{i-1} + h$ (i=1 a 2n)

Obviamente e segundo Ruggiero (1996), “n” tem que ser par, ou seja, o número de intervalos tem que ser par ou o número de pontos “np” tem que se ímpar, uma vez que np=2n+1.

É possível o cálculo a partir de uma tabela de valores com intervalos diferentes, mas, a cada 3 pontos, os intervalos têm que obrigatoriamente ser iguais. Ruggiero (1996) mostra a expressão genérica para o cálculo de cada intervalo com 3 pontos:

$$\int_{x_{2k-2}}^{x_{2k}} f(x)\,dx = \frac{h}{3}\left[f(x_{2k-2}) + 4f(x_{2k-1}) + f(x_{2k})\right]$$

2.9. Derivação Numérica

A derivada em um ponto x_0 de uma função y=f(x) representa a taxa de variação instantânea de y em relação a x neste ponto x_0 e pode ser denotada como $f'(x_0)$. Graficamente, a derivada é interpretada como o coeficiente angular da reta tangente ao gráfico de f(x) em um ponto x_0.

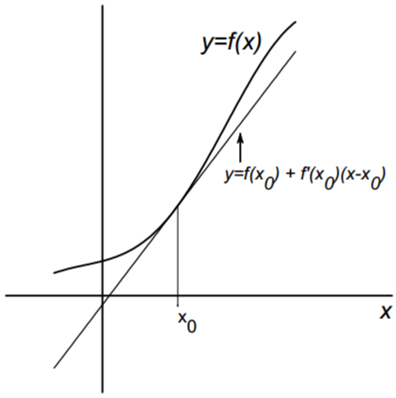


Figura: Função f(x) e reta tangente a ela. Fonte: http://www.if.ufrj.br/~tkodama/Fisica_I/expansaotaylor.pdf.

Chamando $(x-x_0)$ de "h", ou seja, uma variação em torno de x, são obtidas 3 possibilidades para o cálculo numérico aproximado do valor da derivada da função num ponto x_i (chamado Método das Diferenças Finitas):

$f_i' = \dfrac{f_{i+1} - f_i}{h}$ → DIFERENÇA ASCENDENTE ou avançada (1ª ORDEM)

$f_i' = \dfrac{f_i - f_{i-1}}{h}$ → DIFERENÇA DESCENDENTE ou atrasada (1ª ORDEM)

$f_i' = \dfrac{f_{i+1} - f_{i-1}}{2h}$ → DIFERENÇA CENTRAL ou centrada (2ª ORDEM - MAIS EXATA)

Segundo Quadros e Bortoli (2009), para as diferenças ascendente e descendente, o erro é da ordem de h. Para a diferença central, o erro é da ordem de h^2.

Sendo assim, o que se faz é, dada uma função f(x) e a necessidade de encontrar f'(x) num ponto x, criar uma tabela com 3 pontos, sendo o ponto central igual (x,f(x)), o ponto à esquerda igual a (x-h,f(x-h)) e o ponto à esquerda igual a (x+h,f(x+h)), escolhendo um h pequeno, e aplicar as funções anteriores.

A partir das diferenças centrais, Quadros e Bortoli (2009) apresentam a expressão numérica para o cálculo da derivada a segunda:

$$f_i^{(2)} = \frac{f_{i+1} - 2\,f_i + f_{i-1}}{h^2}$$

2.10. Aproximação Polinomial

Ao contrário da interpolação, na qual todos os pontos têm que estar sobre a função interpoladora, há casos em que se precisa aproximar os dados por meio de uma função mais simples, mesmo que os dados não façam parte da função agora aproximadora.

As funções aproximadoras permitem extrapolar dados, ou seja, encontrar dados que aconteceriam caso a função seguisse uma tendência. A extrapolação tenta estimar os valores fora do domínio de pontos conhecidos, a partir da função aproximadora.

Pode-se usar uma função de ajuste do tipo polinomial:

$$g(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

Na função g(x), as funções $\phi_i(x)$ devem ser preestabelecidas, de acordo com a aproximação que se quer chegar.

Chamando M a função dos erros mínimos quadrados, podemos escrever:

$$M(a_0, a_1, \ldots, a_m) = \sum_{i=0}^{n} r_i^2(x) = \sum_{i=0}^{n} [f(x_i) - g(x_i)]^2$$

O Método dos Mínimos Quadrados deverá determinar os valores dos coeficientes a_i. Neste caso, (n+1) é o número de pontos conhecidos, a partir dos quais se encontrará a função de ajuste:

$$M = (f(x_0) - g(x_0))^2 + (f(x_1) - g(x_1))^2 + \ldots + (f(x_n) - g(x_n))^2$$
$$M = (f(x_0) - (a_0 + a_1 x_0))^2 + (f(x_1) - (a_0 + a_1 x_1))^2 + \ldots + (f(x_n) - (a_0 + a_1 x_n))^2$$

Aplicando a derivada parcial da função M em relação de cada coeficiente “ ai ” e fazendo ajustes, obtém-se um sistema de equações lineares:

$$a_0(n+1) + a_1 \sum_0^n x_i = \sum_0^n f(x_i)$$

$$a_0 \sum_0^n x_i + a_1 \sum_0^n x_i^2 = \sum_0^n f(x_i) x_i$$

Podendo ser escrito na forma:

$$\begin{bmatrix} (n+1) & \sum_0^n x_i \\ \sum_0^n x_i & \sum_0^n x_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_0^n f(x_i) \\ \sum_0^n f(x_i) x_i \end{bmatrix}$$

Resolvendo o sistema e achando todos valores de “ ai ”, é só substituir e encontrar o polinômio.

2.11. Aproximação Exponencial

A aproximação de dados por funções especiais como as exponenciais é feita a partir de uma linearização dos dados e depois, a partir dessa reta, aplicam-se processos de ajuste linear para encontrar a função linear e sua consequente função especial.

Utilizando um conjunto de dados (x,y), sendo os valores de y maiores ou diferentes de zero, de uma função exponencial:

$y = a*e^{bx}$

Para linearizar este tipo de função, o que se faz é aplicar o logaritmo em ambos os lados:

$\ln(y) = \ln(a*e^{bx}) = \ln(a)+\ln(e^{bx}) = \ln(a)+bx$

Se fizermos uma mudança de variáveis como abaixo, chegamos uma função z que é linear:

$z = \ln(y)$
$a_0 = \ln(a)$
$a_1 = b$

$z = a_0 + a_1x$

Assim, calculam-se os pontos (z,x) e se faz um ajuste linear para encontrar a_0 e a_1. Lembrando que, no caso, o MMQ linear é aplicado da seguinte maneira:

$$\begin{bmatrix} (n+1) & \sum_0^n x_i \\ \sum_0^n x_i & \sum_0^n x_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_0^n z_i \\ \sum_0^n z_i * x_i \end{bmatrix}$$

Achando os valores de " a0 " e " a1 ", basta fazer " e^{a0} " e então substituir na equação inicial.

2.12. Aproximação Potencial

A aproximação de dados por funções especiais como as potenciaisé feita a partir de uma linearização dos dados e depois, a partir dessa reta, aplicam-se processos de ajuste linear para encontrar a função linear e sua consequente função especial.

Utilizando um conjunto de dados (x,y), sendo os valores de "x" e"y" maiores ou diferentes de zero, de uma função potencial:

$y = a*x^b$

Para linearizar este tipo de função, o que se faz é aplicar o logaritmo em ambos os lados:

$\ln(y) = \ln(a*x^b) = \ln(a)+\ln(x^b) = \ln(a)+b\ln(x)$

Se fizermos uma mudança de variáveis como abaixo, chegamos uma função z que é linear:

$z = \ln(y)$
$k = \ln(x)$
$a0 = \ln(a)$
$a1 = b$

z = a0 + a1k

Assim, calculam-se os pontos (z,k) e se faz um ajuste linear para encontrar a_0 e a_1. Lembrando que, no caso, o MMQ linear é aplicado da seguinte maneira:

$$\begin{bmatrix} (n+1) & \sum_0^n k_i \\ \sum_0^n k_i & \sum_0^n k^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_0^n z_i \\ \sum_0^n z_i * k_i \end{bmatrix}$$

Achando os valores de “ a0 ” e “ a1 ”, basta fazer " e^{a0} " e " e^{a1} " e então substituir na equação inicial.

2.13. Método de Euler para Solução de PVI's De Ordem 1

Considere o seguinte esquema para a resolução de Problema de valor inicial (PVI) de ordem 1:

$$\text{(PVI)} \begin{cases} y' = f(x,y) \\ y(x_0) = y_0 \end{cases}$$

Conforme Ruggiero (1996), o Método de Euler baseia-se em expansões em séries de Taylor de 1ª ordem (menos precisas). O que se faz é calcular o valor da função em um intervalo de pontos, que satisfazem à Equação diferencial ordinária(EDO) e ao PVI.

Se considerarmos um intervalo pequeno xi+1-xi e lembrando que f'(x)=f(x,y):
y(xi+1) = y(xi)+h.f(xi,y(xi)) (Taylor de 1ª ordem: desconsiderados f'' pra frente)

Em outra notação:

$$y_{i+1} = y_i + hf(x_i, y_i), \quad i = 0,1,\ldots,n-1$$

Assim, o método de Euler é uma sucessão de “n” intervalos de x e, quanto menor o intervalo h, melhor é a aproximação. Assim, note que, quanto menor o intervalo, mais pontos de f(x) você terá e esses pontos terão melhor aproximação em relação ao valor analítico. O número de intervalos será dado por:
n = (xn – x0) / h

2.14. Método de Runge-Kutta de 2ª ordem para o cálculo de PVI'S de ordem 1'

Podem ser deduzidas fórmulas de Euler usando as expansões de Taylor de 2ª ordem, 3ª e assim por diante. Porém, tais fórmulas necessitam do cálculo das derivadas sucessivas de $f(x_i,y_i)$. Buscando evitar isso, surgem as fórmulas de Runge-Kutta. No caso de Runge-Kutta de 2ª ordem, depois dos tratamentos matemáticos, o PVI geral fica:

$$y_{i+1} = y_i + \frac{h}{2}(k_1 + k_2), \quad k_1 = f(x_i, y_i) \text{ e } k_2 = f(x_i + h, y_i + hk_1), \quad i = 0,1,\ldots,n-1$$

2.15. Método das diferenças finitas para a resolução de PVC'S de Ordem 2

Um Problema de valor de contorno (PVC) de ordem 2 generalizado pelo seguinte:

$$\begin{cases} y''+P(x)y'+Q(x)y=f(x) \\ y(x_o)=y_o, \quad y(x_n)=y_n. \end{cases}$$

Conforme Ruggiero (1996), a idéia básica do Método das Diferenças Finitas é transformar o problema de resolver uma equação diferencial num problema de resolver um sistema de equações algébricas, usando para isto aproximações das derivadas que aparecem na equação, por diferenças finitas.

Então, o problema passa a ser a resolução do seguinte sistema linear (de 1 a n-1 porque y_0 e y_n são conhecidos):

$$y''_1+P(x_1)*y'_1+Q(x_1)*y_1=f(x_1)$$
$$y''_2+P(x_2)*y'_2+Q(x_2)*y_2=f(x_2)$$
(...)
$$y''_{n-1}+P(x_{n-1})*y'_{n-1}+Q(x_{n-1})*y_{n-1}=f(x_{n-1})$$

O sistema é possível de ser resolvido porque podemos aproximar as derivadas de y conhecendo-se y em alguns pontos. Relembrando, as aproximações mais usadas para a primeira e segunda derivadas em x_i são obtidas a partir da diferença central, relembrando o item 10.1 e na notação de Ruggiero (1996):

$$y'(x_i) \simeq \frac{y_{i+1}-y_{i-1}}{2h}$$
$$y''(x_i) \simeq \frac{y_{i+1}-2y_i+y_{i-1}}{h^2}$$

Para cada linha:

$$y''_1=(y_2-2y_1+y_0)/h^2=y_0/h^2-2y_1/h^2+y_2/h^2$$
$$y''_2=(y_3-2y_2+y_1)/h^2=y_1/h^2-2y_2/h^2+y_3/h^2$$
(...)
$$y''_{n-1}=(y_n-2y_{n-1}+y_{n-2})/h^2=y_{n-2}/h^2-2y_{n-1}/h^2+y_n/h^2$$

$$y'_1=(y_2-y_0)/2h=-y_0/2h+y_2/2h$$
$$y'_2=(y_3-y_1)/2h=-y_1/2h+y_3/2h$$
(...)
$$y'_{n-1}=(y_n-y_{n-2})/2h=-y_{n-2}/2h+y_n/2h$$

Para a primeira linha:

$$y_0/h^2 - 2y_1/h^2 + y_2/h^2 + P(x_1)*(-y_0/2h + y_2/2h) + Q(x_1)*y_1 = f(x_1)$$

Ou:

$$y_0/h^2 - P(x_1)*(y_0/2h) - 2y_1/h^2 + Q(x_1)*y_1 + y_2/h^2 + P(x_1)*(y_2/2h) = f(x_1)$$

Ou:

$$y_0[1/h^2 - P(x_1)/2h] + y_1[-2/h^2 + Q(x_1)] + y_2[1/h^2 + P(x_1)/2h] = f(x_1)$$

Como y_0 é conhecido:

$$y_1[-2/h^2 + Q(x_1)] + y_2[1/h^2 + P(x_1)/2h] = f(x_1) - y_0[1/h^2 - P(x_1)/2h]$$

Para a linha 2:

$$y_1/h^2 - 2y_2/h^2 + y_3/h^2 + P(x_2)*(-y_1/2h + y_3/2h) + Q(x_2)*y_2 = f(x_2)$$

Ou:

$$y_1/h^2 - P(x_2)*(y_1/2h) - 2y_2/h^2 + Q(x_2)*y_2 + y_3/h^2 + P(x_2)*(y_3/2h) = f(x_2)$$

Ou:

$$y_1[1/h^2 - P(x_2)/2h] + y_2[-2/h^2 + Q(x_2)] + y_3[1/h^2 + P(x_2)/2h] = f(x_2)$$

Semelhantemente, para a linha n-2:

$$y_{n-3}[1/h^2 - P(x_{n-2})/2h] + y_{n-2}[-2/h^2 + Q(x_{n-2})] + y_{n-1}[1/h^2 + P(x_{n-2})/2h] = f(x_{n-2})$$

Semelhantemente, para a linha n-1:

$$y_{n-2}[1/h^2 - P(x_{n-1})/2h] + y_{n-1}[-2/h^2 + Q(x_{n-1})] + y_n[1/h^2 + P(x_{n-1})/2h] = f(x_{n-1})$$

Como y_n é conhecido:

$$y_{n-2}[1/h^2 - P(x_{n-1})/2h] + y_{n-1}[-2/h^2 + Q(x_{n-1})] = f(x_{n-1}) - y_n[1/h^2 + P(x_{n-1})/2h]$$

Sendo assim, o problema se resume a resolver o seguinte sistema linear:

$$\begin{bmatrix} d_1 & c_1 & 0 & 0 & 0 & 0 \\ a_2 & d_2 & c_2 & 0 & 0 & 0 \\ 0 & a_3 & d_3 & c_3 & 0 & 0 \\ 0 & 0 & \dots & \dots & \dots & 0 \\ 0 & 0 & 0 & a_{n-2} & d_{n-2} & c_{n-2} \\ 0 & 0 & 0 & 0 & a_{n-1} & d_{n-1} \end{bmatrix} \text{x} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \dots \\ y_{n-2} \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} f(x_1) - y_0[1/h^2 - P(x_1)/2h] \\ f(x_2) \\ f(x_3) \\ \dots \\ f(x_{n-2}) \\ f(x_{n-1}) - y_n[1/h^2 + P(x_{n-1})/2h] \end{bmatrix}$$

Em que:

$d_i = -2/h^2 + Q(x_i)$ $\qquad 1 \le i \le (n-1)$

$c_i = 1/h^2 + P(x_{i-1})/2h$ $\qquad 1 \le i \le (n-2)$

$a_i = 1/h^2 - P(x_{i+1})/2h$ $\qquad 2 \le i \le (n-1)$

CAPÍTULO 3

RESULTADOS

Na página do NEV na internet já se encontram os programas publicados. O acesso é em http://vtp.ifsp.edu.br/nev/, como apresentado na Fig. 1 a seguir. Escolhido qual programa acessar, é preciso clicar no link para que abra a interface do programa:

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus Votuporanga

NEV - NÚCLEO DE ENGENHARIA VIRTUAL - DESDE 17/06/2015

GRUPO CAPES: NEVE - NÚCLEO DE ENGENHARIA VIRTUAL E EXPERIMENTAL (IFSP CAMPUS VOTUPORANGA)
Esta página tem o objetivo de publicar os resultados de pesquisas da área de Edificações do IFSP campus Votuporanga, com foco em trabalhos de cunho virtual, como softwares, vídeos, maquetes virtuais, etc.
Contato - prof. Gustavo Cabrelli Nirschl: nirschl@gmail.com

Área: Cálculo Numérico

Título e link	Data	Autor	Orientador	Tipo	Curso
DERIVAÇÃO NUMÉRICA DE FUNÇÃO - TAYLOR	21/10/2016	Isabela Cassia Dominical Parra	Prof. Gustavo Cabrelli Nirschl	Iniciação Científica com Bolsa Institucional	Engenharia Civil
INTEGRAÇÃO DE FUNÇÃO - FÓRMULA DE SIMPSON	21/10/2016	Isabela Cassia Dominical Parra	Prof. Gustavo Cabrelli Nirschl	Iniciação Científica com Bolsa Institucional	Engenharia Civil
INTEGRAÇÃO POR MEIO DA TABELA DE PONTOS - FÓRMULA DE SIMPSON	08/09/2016	Isabela Cassia Dominical Parra	Prof. Gustavo Cabrelli Nirschl	Iniciação Científica com Bolsa Institucional	Engenharia Civil
INTEGRAÇÃO POR MEIO DA TABELA DE PONTOS- REGRA DOS TRAPÉZIOS	08/09/2016	Isabela Cassia Dominical Parra	Prof. Gustavo Cabrelli Nirschl	Iniciação Científica com Bolsa Institucional	Engenharia Civil
INTEGRAÇÃO DE FUNÇÃO - REGRA DOS TRAPÉZIOS	08/09/2016	Isabela Cassia Dominical Parra	Prof. Gustavo Cabrelli Nirschl	Iniciação Científica com Bolsa Institucional	Engenharia Civil
INTERPOLAÇÃO POLINOMIAL POR FORMA DE NEWTON	08/09/2016	Isabela Cassia Dominical Parra	Prof. Gustavo Cabrelli Nirschl	Iniciação Científica com Bolsa Institucional	Engenharia Civil
INTERPOLAÇÃO POLINOMIAL POR LAGRANGE	04/08/2016	Isabela Cassia Dominical Parra	Prof. Gustavo Cabrelli Nirschl	Iniciação Científica com Bolsa Institucional	Engenharia Civil
INTERPOLAÇÃO POLINOMIAL	04/08/2016	Isabela Cassia Dominical Parra	Prof. Gustavo Cabrelli Nirschl	Iniciação Científica com Bolsa Institucional	Engenharia Civil
SISTEMAS LINEARES PELO MÉTODO DE GAUSS-SEIDEL	28/04/2016	Isabela Cassia Dominical Parra	Prof. Gustavo Cabrelli Nirschl	Iniciação Científica com Bolsa Institucional	Engenharia Civil
SISTEMAS LINEARES PELO MÉTODO DE GAUSS-JACOBI	28/04/2016	Isabela Cassia Dominical Parra	Prof. Gustavo Cabrelli Nirschl	Iniciação Científica com Bolsa Institucional	Engenharia Civil
SISTEMAS LINEARES POR FATORAÇÃO LU	17/04/2016	Isabela Cassia Dominical Parra	Prof. Gustavo Cabrelli Nirschl	Iniciação Científica com Bolsa Institucional	Engenharia Civil

Figura 1: Links de acesso para os programas, na página do NEV.

3.1 Sistemas Lineares por Fatoração LU

A página para Sistemas Lineares por Fatoração LU apresenta-se como a Fig. 2 a seguir.

Figura 2: Página para Sistemas Lineares por Fatoração LU.

Para iniciar os cálculos, tem-se duas opções: ou gerar o sistema na própria página, a partir da ordem digitada (botão “GERAR”), e digitar os valores (Fig. 3 a seguir) de cada elemento ou criar um arquivo TXT, seguindo o padrão apresentado ao se clicar no botão “EXEMPLO de TXT” (Fig. 4 a seguir).

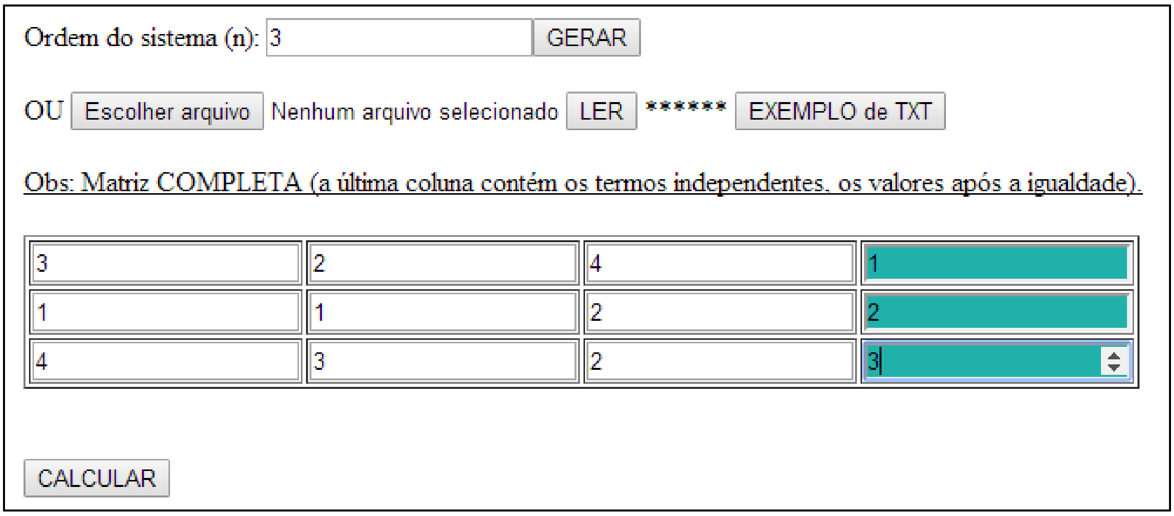


Figura 3: Exemplo de sistema gerado com ordem 3, cujos valores têm que ser digitados

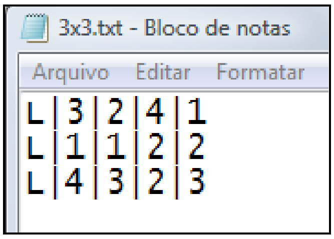


Figura 4:Exemplo de arquivo TXT que pode ser lido pela página.

Depois de ler ou digitar o sistema, clica-se no botão "CALCULAR" (veja Fig. 3 anterior), aparecendo os sistemas triangulares equivalentes e a solução final. A Fig. 5 a seguir apresenta um exemplo para um sistema de ordem 3.

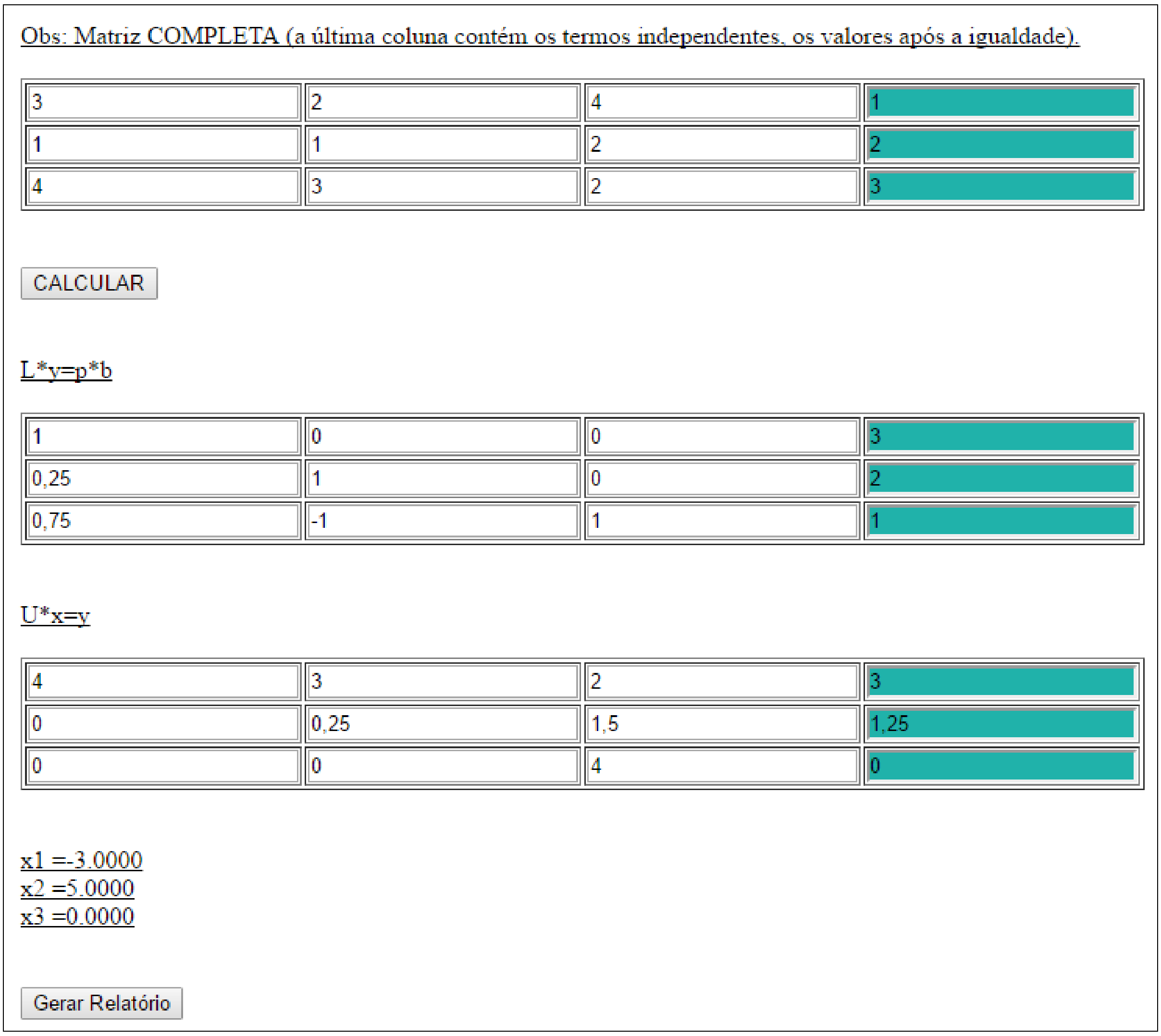


Figura 5: Exemplo da resolução de um sistema de ordem 3.

Por fim, como principal característica da página criada, pode-se gerar o relatório, clicando no botão “Gerar Relatório” ao fim da página, conforme a Fig. 5 anterior. O relatório é em PDF e contém a resolução detalhada. A figura a seguir (Fig. 6) mostra o relatório para a matriz exemplificada na Fig. 3.

A matriz segue o modelo abaixo

$$\begin{bmatrix} a11 & a12 & \ldots & a1n & b1 \\ a21 & a22 & .. & a2n & b2 \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ an1 & an2 & \ldots & ann & bn \end{bmatrix}$$

Ordem da Matriz =3

Matriz Completa:

3.000	2.000	4.000	*1.000*
1.000	1.000	2.000	*2.000*
4.000	3.000	2.000	*3.000*

A ordem inicial das linhas (vetor p) é:

1
2
3

A matriz L inicial é a matriz identidade:

1.000	0.000	0.000
0.000	1.000	0.000
0.000	0.000	1.000

Para resolução do sistema linear pelo método da fatoração LU o escalonamento é feito independentemente do vetor dos termos independentes. Este método fatora o sistema linear Ax=B em Ly=B e em Ux=y.A matriz A (matriz dos coeficientes) irá se decompor em A=L*U. Onde:
- L é uma matriz triangular inferior em que todos os elementos acima da diagonal são nulos, os elementos da diagonal principal são iguais a 1 e os elementos abaixo da diagonal principal são os valores dos multiplicadores (m), resultante do escalonamento da matriz A.
- U é a matriz triangular superior que resultou do escalonamento com pivoteamento parcial da matriz A.

Abaixo segue a resolução:

O Pivô é:
4.000

A nova ordem das linhas será (vetor p atualizado):

3
2

1

A matriz "A" com a troca de linhas fica:

4.000	3.000	2.000
1.000	1.000	2.000
3.000	2.000	4.000

A matriz L com a troca de linhas abaixo da diagonal fica:

1.000	0.000	0.000
0.000	1.000	0.000
0.000	0.000	1.000

m[2,1]=a'[2,1]/a'[1,1]=1.0000/4.0000=0.2500
a'[2,1]=a[2,1]-m[2,1]*a'[1,1]=1.0000-(0.2500)*(4.0000)=0.0000
a'[2,2]=a[2,2]-m[2,1]*a[1,2]=1.0000-(0.2500)*(3.0000)=0.2500
a'[2,3]=a[2,3]-m[2,1]*a[1,3]=2.0000-(0.2500)*(2.0000)=1.5000

m[3,1]=a'[3,1]/a'[1,1]=3.0000/4.0000=0.7500
a'[3,1]=a[3,1]-m[3,1]*a'[1,1]=3.0000-(0.7500)*(4.0000)=0.0000
a'[3,2]=a[3,2]-m[3,1]*a[1,2]=2.0000-(0.7500)*(3.0000)=-0.2500
a'[3,3]=a[3,3]-m[3,1]*a[1,3]=4.0000-(0.7500)*(2.0000)=2.5000

A matriz "A" escalonada até aqui fica:

4.000	3.000	2.000
0.000	0.2500	1.500
0.000	-0.2500	2.500

A matriz L atualizada fica:

1.000	0.000	0.000
0.2500	1.000	0.000
0.7500	0.000	1.000

O Pivô é:
0.2500

A nova ordem das linhas será (vetor p atualizado):

3
2
1

A matriz "A" com a troca de linhas fica:

4.000	3.000	2.000
0.000	0.2500	1.500
0.000	-0.2500	2.500

A matriz L com a troca de linhas abaixo da diagonal fica:

1.000	0.000	0.000
0.2500	1.000	0.000
0.7500	0.000	1.000

m[3,2]=a'[3,2]/a'[2,2]=-0.2500/0.2500=-1.0000
a'[3,1]=a[3,1]-m[3,2]*a'[2,1]=0.000-(-1.0000)*(0.000)=0.000
a'[3,2]=a[3,2]-m[3,2]*a'[2,2]=0.000-(-1.0000)*(0.000)=0.000
a'[3,3]=a[3,3]-m[3,2]*a[2,3]=2.5000-(-1.0000)*(1.5000)=4.0000

A matriz "A" escalonada até aqui fica:

4.000	3.000	2.000
0.000	0.2500	1.500
0.000	0.000	4.000

A matriz L atualizada fica:

1.000	0.000	0.000
0.2500	1.000	0.000
0.7500	-1.000	1.000

O vetor b com a troca de linhas determinada pelo vetor p fica:

3.000
2.000
1.000

Então, o sistema triangular inferior a resolver (L*y=b com as linhas trocadas) fica:

1.000	0.000	0.000	*3.000*
0.2500	1.000	0.000	*2.000*
0.7500	-1.000	1.000	*1.000*

Com o triângulo inferior formado pode se encontrar os valores de y:
soma=0
y1=(b[1]-(soma))/L[11]=(3.0000-(0.0000))/1.0000=3.0000
soma=0
soma=soma+L[2,1]*y[1]=0.0000+(0.2500*3.0000)=0.7500
y2=(b[2]-(soma))/L[22]=(2.0000-(0.7500))/1.0000=1.2500
soma=0
soma=soma+L[3,1]*y[1]=0.0000+(0.7500*3.0000)=2.2500
soma=soma+L[3,2]*y[2]=2.2500+(-1.0000*1.2500)=1.0000
y3=(b[3]-(soma))/L[33]=(1.0000-(1.0000))/1.0000=0.0000

Que resulta em y = :

3.000
1.250
0.000

E o sistema triangular superior a resolver (U*x=y com as linhas trocadas) fica:

4.000	3.000	2.000	*3.000*
0.000	0.2500	1.500	*1.250*
0.000	0.000	4.000	*0.000*

Com o triângulo superior formado pode se encontrar os valores de x:

x3=(y[3]/a'[33])=0.0000/4.0000=0.0000
soma=0
soma=soma+a[2,3]*x[3]=0.0000+(1.5000*0.0000)=0.0000
x2=(y[2]-(soma))/a'[22]=(1.2500-(0.0000))/0.2500=5.0000
soma=0
soma=soma+a[1,2]*x[2]=0.0000+(3.0000*5.0000)=15.0000
soma=soma+a[1,3]*x[3]=15.0000+(2.0000*0.0000)=15.0000
x1=(y[1]-(soma))/a'[11]=(3.0000-(15.0000))/4.0000=-3.0000

Resultando na solução final do sistema, x = :

-3.000
5.000
0.000

A seguir apresenta o algoritmo principal para a resolução de sistema linear pelo método de faltoração LU com Pivoteamento Parcial:

```
Dado n, A[n,n] e b[n,1]
Para i=1 até i=n faça
p[i]=i
Fim do laço
Para k=1 até k=n-1 faça //para cada coluna
   pv=|a[k,k]| //pv é o pivô
   r=k //r é o n° da linha que está o pivô
   Para i=k+1 até i=n faça //para cada linha
      Se |a[i,k]|>pv então //encontra o maior pivô em módulo
      pv=|a[i,k]|
        r=i
        Fim do condicional
    Fim do laço
   Se pv=0 então
   "Todos os pivôs são nulos." PARE.
   Se r for diferente de k, faça
   Troque a linha k com a linha r
   atemp=p(k)
   p(k)=p(r)
   p(r)=aux
    Para c=1 até c=n faça //coluna 1 a n+1
      atemp=a[k,c]; //variável temporária
      a[k,c]=a[r,c];
      a[r,c]=atemp;
    Fim do Laço
   Fim do condicional
```

```
  Para i=k+1 até i=n faça //para cada linha, menos a anterior (que tem o pivô)
  m[i,k]=a[[i,k]/a[k,k]
  a[i,k]=m[i,k]
    Para j=k+1 até j=n faça //para cada coluna, a partir da k
    a[i,j]=a[i,j]-m[i,k]*a[k,j]
    Fim do laço
  Fim do laço
Fim do laço
c=Pb
Para i=1 até i=n faça
 r=p(i)
 c[i]=b[r]
Fim do laço
Ly=c
Para i=1 até i=n faça
soma=0
  Para j=1 até j=i-1 faça
  soma=soma+a[i,j]*x[j]
  Fim do laço
y[i]=c[i]-soma
Ux=y
Para i=n até i=1 faça
soma=0
  Para j=i+1 até j=n faça
  soma=soma+a[i,j]*x[j]
  Fim do laço
x[i]=(y[i]-soma)/a[i,i]
Fim do laço
```

REFERÊNCIAS BIBLIOGRÁFICAS

RUGGIERO, M.A.G; LOPES, V.L.R. Cálculo numérico: aspectos teóricos e computacionais. 2.ed. São Paulo: Makron Books, 1996.

SWOKOWSKI. Cálculo com Geometria Analítica. São Paulo. Makron Books, 1995.

QUADROS, Régis S. de; BORTOLI, Álvaro L. de. Fundamentos de Cálculo Numérico para Engenheiros. Porto Alegre, 2009. Disponível em < http://www.inf.ufrgs.br/~rlflupchinski/files/20112/NUMERICO/numerico-bortoli.pdf >. Acesso em outubro de 2014.

Figura 6: Relatório em PDF gerado com o exemplo dos dados da Fig. 5.

3.2 Sistemas Lineares Via Gauss-Jacobi

A página para Sistemas Lineares por Gauss-Jacobi apresenta-se como a Fig. 7 a seguir.

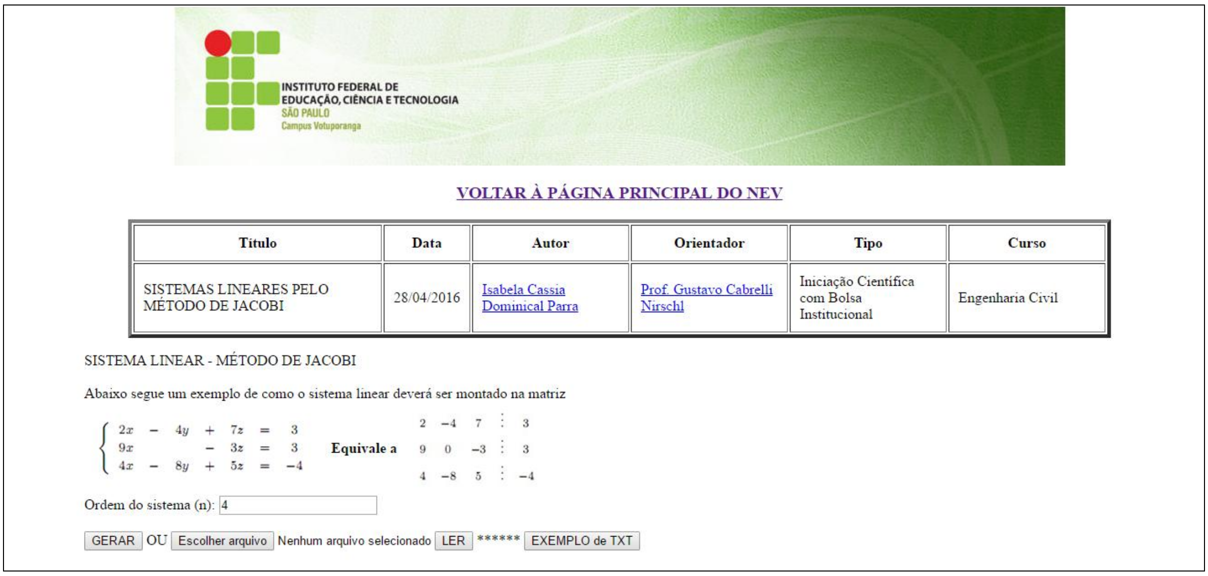


Figura 7: Página para Sistemas Lineares por Gauss- Jacobi.

Para iniciar os cálculos, tem-se duas opções: ou gerar o sistema na própria página, a partir da ordem digitada (botão "GERAR"), e digitar os valores (Fig. 8 a seguir) de cada elemento ou criar um arquivo TXT, seguindo o padrão apresentado ao se clicar no botão "EXEMPLO de TXT" (Fig. 9 a seguir).

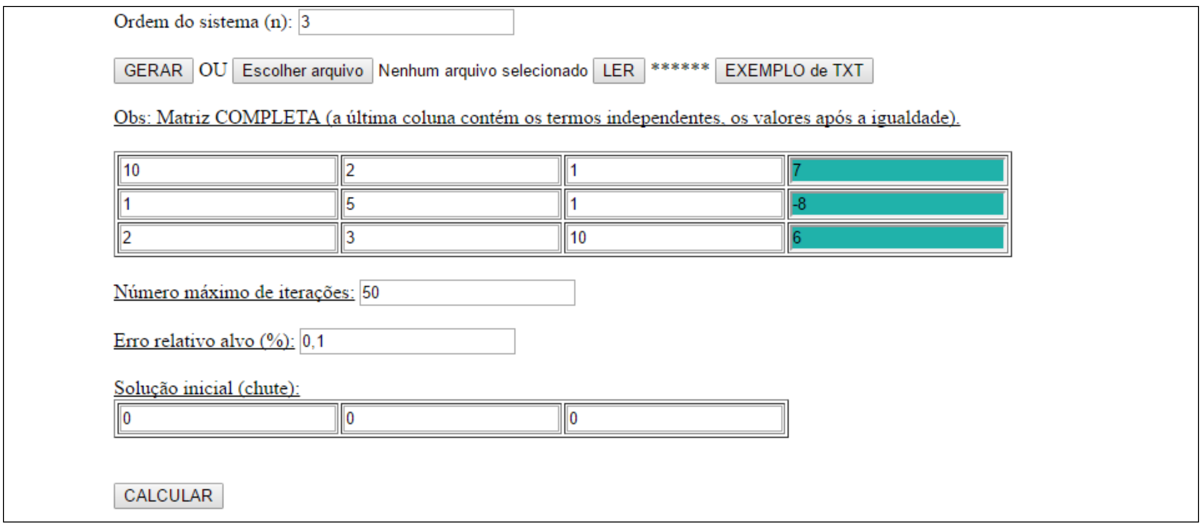


Figura 8: Exemplo de sistema gerado com ordem 3, cujos valores têm que ser digitados.

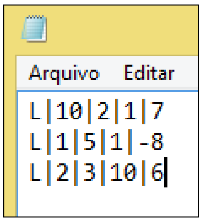
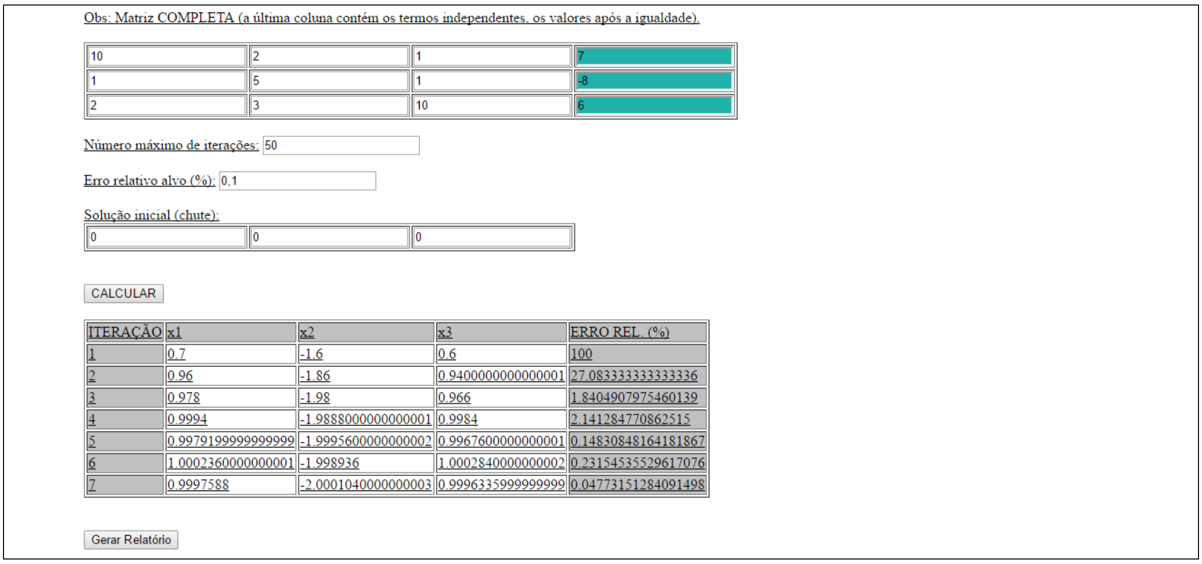


Figura 9: Exemplo de arquivo TXT que pode ser lido pela página.

Depois de ler ou digitar o sistema, clica-se no botão “CALCULAR” (veja Fig. 8anterior), aparecendo uma tabela com os resultados de cada iteração. A Fig. 10 a seguir apresenta um exemplo para um sistema de ordem 3.



ITERAÇÃO	x1	x2	x3	ERRO REL. (%)
1	0.7	-1.6	0.6	100
2	0.96	-1.86	0.9400000000000001	27.083333333333336
3	0.978	-1.98	0.966	1.8404907975460139
4	0.9994	-1.9888000000000001	0.9984	2.141284770862515
5	0.9979199999999999	-1.9995600000000002	0.9967600000000001	0.14830848164181867
6	1.0002360000000001	-1.998936	1.0002840000000002	0.23154535529617076
7	0.9997588	-2.0001040000000003	0.9996335999999999	0.04773151284091498

Figura 10: Exemplo da resolução por Gauss-Jacobi de um sistema de ordem 3.

Por fim, como principal característica da página criada, pode-se gerar o relatório, clicando no botão “Gerar Relatório” ao fim da página, conforme a Fig. 10 anterior. O relatório é em PDF e contém a resolução detalhada. A figura a seguir (Fig. 11) mostra o relatório para a matriz exemplificada na Fig. 8.

A matriz segue o modelo abaixo

$$\begin{bmatrix} a11 & a12 & \ldots & a1n & b1 \\ a21 & a22 & .. & a2n & b2 \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ an1 & an2 & \ldots & ann & bn \end{bmatrix}$$

Ordem da Matriz =3

Matriz Completa:

10.00	2.000	1.000	*7.000*
1.000	5.000	1.000	*-8.000*
2.000	3.000	10.00	*6.000*

O número máximo de iterações é: 50

O erro relativo alvo é: 0.1

A solução inicial, que é um valor estimado, é:

0.000	0.000	0.000

Para resolução do sistema linear pelo método de Gauss-Jacobi, é necessário isolar o vetor x, de acordo com o exemplo abaixo:

$$\begin{cases} a_{11} * x_1 + a_{12} * x_2 + \cdots a_{1n} * x_n = b_1 \\ a_{21} * x_1 + a_{22} * x_2 + \cdots a_{2n} * x_n = b_2 \\ \ldots \\ a_{n1} * x_1 + a_{n2} * x_2 + \cdots a_{nn} * x_n = b_n \end{cases} \longrightarrow \begin{cases} x_1 = \frac{b_1}{a_{11}} - \frac{1}{a_{11}}(a_{12} * x_2 + \cdots a_{1n} * x_n) \\ x_2 = \frac{b_2}{a_{22}} - \frac{1}{a_{22}}(a_{21} * x_1 + \cdots a_{2n} * x_n) \\ \ldots \\ x_n = \frac{b_n}{a_{nn}} - \frac{1}{a_{nn}}(a_{n1} * x_1 + \cdots + a_{nn-1} * x_{n-1}) \end{cases}$$

Para prosseguir a resolução, é necessário escolher um valor inicial para todos os "x".
Basta substituir os valores iniciais de "x" nas equações e encontrar as soluções das iterações.
Deve-se continuar subtituindo os valores encontrados até que haja uma convergência, ou seja, até que se encontre um erro menor do que o Erro Relativo alvo.
Para o caso de a solução demorar para convergir ou não convergir, determina-se um número máximo de iterações.
Obs.: Este método não permite matriz que tenha zeros na diagonal principal.

Abaixo segue a resolução:

soma=0
soma=soma+a[1,2]*x2[0]=0.0000+(2.0000*0.0000)=0.0000
soma=soma+a[1,3]*x3[0]=0.0000+(1.0000*0.0000)=0.0000
x1[1]=(b[1]/a[1,1])-soma=(7.0000/10.0000)-(0.0000)=0.7000
soma=0
soma=soma+a[2,1]*x1[0]=0.0000+(1.0000*0.0000)=0.0000
soma=soma+a[2,3]*x3[0]=0.0000+(1.0000*0.0000)=0.0000
x2[1]=(b[2]/a[2,2])-soma=(-8.0000/5.0000)-(0.0000)=-1.6000
soma=0
soma=soma+a[3,1]*x1[0]=0.0000+(2.0000*0.0000)=0.0000
soma=soma+a[3,2]*x2[0]=0.0000+(3.0000*0.0000)=0.0000
x3[1]=(b[3]/a[3,3])-soma=(6.0000/10.0000)-(0.0000)=0.6000
Erro relativo máximo:Er[1]=|x1[1]-x1[0]|/|x1[1]*100=(|0.7000-(0.0000)|/|0.7000|)*100=100.0000

soma=0
soma=soma+a[1,2]*x2[1]=0.0000+(2.0000*-1.6000)=-0.3200
soma=soma+a[1,3]*x3[1]=-0.3200+(1.0000*0.6000)=-0.2600
x1[2]=(b[1]/a[1,1])-soma=(7.0000/10.0000)-(-0.2600)=0.9600
soma=0
soma=soma+a[2,1]*x1[1]=0.0000+(1.0000*0.7000)=0.1400
soma=soma+a[2,3]*x3[1]=0.1400+(1.0000*0.6000)=0.2600
x2[2]=(b[2]/a[2,2])-soma=(-8.0000/5.0000)-(0.2600)=-1.8600
soma=0
soma=soma+a[3,1]*x1[1]=0.0000+(2.0000*0.7000)=0.1400
soma=soma+a[3,2]*x2[1]=0.1400+(3.0000*-1.6000)=-0.3400
x3[2]=(b[3]/a[3,3])-soma=(6.0000/10.0000)-(-0.3400)=0.9400
Erro relativo máximo:Er[2]=|x3[2]-x3[1]|/|x3[2]*100=(|0.9400-(0.6000)|/|0.9400|)*100=36.1702

soma=0
soma=soma+a[1,2]*x2[2]=0.0000+(2.0000*-1.8600)=-0.3720
soma=soma+a[1,3]*x3[2]=-0.3720+(1.0000*0.9400)=-0.2780
x1[3]=(b[1]/a[1,1])-soma=(7.0000/10.0000)-(-0.2780)=0.9780
soma=0
soma=soma+a[2,1]*x1[2]=0.0000+(1.0000*0.9600)=0.1920
soma=soma+a[2,3]*x3[2]=0.1920+(1.0000*0.9400)=0.3800
x2[3]=(b[2]/a[2,2])-soma=(-8.0000/5.0000)-(0.3800)=-1.9800
soma=0
soma=soma+a[3,1]*x1[2]=0.0000+(2.0000*0.9600)=0.1920
soma=soma+a[3,2]*x2[2]=0.1920+(3.0000*-1.8600)=-0.3660
x3[3]=(b[3]/a[3,3])-soma=(6.0000/10.0000)-(-0.3660)=0.9660
Erro relativo máximo:Er[3]=|x2[3]-x2[2]|/|x2[3]*100=(|-1.9800-(-1.8600)|/|-1.9800|)*100=6.0606

soma=0
soma=soma+a[1,2]*x2[3]=0.0000+(2.0000*-1.9800)=-0.3960
soma=soma+a[1,3]*x3[3]=-0.3960+(1.0000*0.9660)=-0.2994
x1[4]=(b[1]/a[1,1])-soma=(7.0000/10.0000)-(-0.2994)=0.9994
soma=0
soma=soma+a[2,1]*x1[3]=0.0000+(1.0000*0.9780)=0.1956
soma=soma+a[2,3]*x3[3]=0.1956+(1.0000*0.9660)=0.3888
x2[4]=(b[2]/a[2,2])-soma=(-8.0000/5.0000)-(0.3888)=-1.9888
soma=0

soma=soma+a[3,1]*x1[3]=0.0000+(2.0000*0.9780)=0.1956
soma=soma+a[3,2]*x2[3]=0.1956+(3.0000*-1.9800)=-0.3984
x3[4]=(b[3]/a[3,3])-soma=(6.0000/10.0000)-(-0.3984)=0.9984
Erro relativo máximo:Er[4]=|x3[4]-x3[3]|/|x3[4]*100=(|0.9984-(0.9660)|/|0.9984|)*100=3.2452

soma=0
soma=soma+a[1,2]*x2[4]=0.0000+(2.0000*-1.9888)=-0.3978
soma=soma+a[1,3]*x3[4]=-0.3978+(1.0000*0.9984)=-0.2979
x1[5]=(b[1]/a[1,1])-soma=(7.0000/10.0000)-(-0.2979)=0.9979
soma=0
soma=soma+a[2,1]*x1[4]=0.0000+(1.0000*0.9994)=0.1999
soma=soma+a[2,3]*x3[4]=0.1999+(1.0000*0.9984)=0.3996
x2[5]=(b[2]/a[2,2])-soma=(-8.0000/5.0000)-(0.3996)=-1.9996
soma=0
soma=soma+a[3,1]*x1[4]=0.0000+(2.0000*0.9994)=0.1999
soma=soma+a[3,2]*x2[4]=0.1999+(3.0000*-1.9888)=-0.3968
x3[5]=(b[3]/a[3,3])-soma=(6.0000/10.0000)-(-0.3968)=0.9968
Erro relativo máximo:Er[5]=|x2[5]-x2[4]|/|x2[5]*100=(|-1.9996-(-1.9888)|/|-1.9996|)*100=0.5381

soma=0
soma=soma+a[1,2]*x2[5]=0.0000+(2.0000*-1.9996)=-0.3999
soma=soma+a[1,3]*x3[5]=-0.3999+(1.0000*0.9968)=-0.3002
x1[6]=(b[1]/a[1,1])-soma=(7.0000/10.0000)-(-0.3002)=1.0002
soma=0
soma=soma+a[2,1]*x1[5]=0.0000+(1.0000*0.9979)=0.1996
soma=soma+a[2,3]*x3[5]=0.1996+(1.0000*0.9968)=0.3989
x2[6]=(b[2]/a[2,2])-soma=(-8.0000/5.0000)-(0.3989)=-1.9989
soma=0
soma=soma+a[3,1]*x1[5]=0.0000+(2.0000*0.9979)=0.1996
soma=soma+a[3,2]*x2[5]=0.1996+(3.0000*-1.9996)=-0.4003
x3[6]=(b[3]/a[3,3])-soma=(6.0000/10.0000)-(-0.4003)=1.0003
Erro relativo máximo:Er[6]=|x3[6]-x3[5]|/|x3[6]*100=(|1.0003-(0.9968)|/|1.0003|)*100=0.3523

soma=0
soma=soma+a[1,2]*x2[6]=0.0000+(2.0000*-1.9989)=-0.3998
soma=soma+a[1,3]*x3[6]=-0.3998+(1.0000*1.0003)=-0.2998
x1[7]=(b[1]/a[1,1])-soma=(7.0000/10.0000)-(-0.2998)=0.9998
soma=0
soma=soma+a[2,1]*x1[6]=0.0000+(1.0000*1.0002)=0.2000
soma=soma+a[2,3]*x3[6]=0.2000+(1.0000*1.0003)=0.4001
x2[7]=(b[2]/a[2,2])-soma=(-8.0000/5.0000)-(0.4001)=-2.0001
soma=0
soma=soma+a[3,1]*x1[6]=0.0000+(2.0000*1.0002)=0.2000
soma=soma+a[3,2]*x2[6]=0.2000+(3.0000*-1.9989)=-0.3996
x3[7]=(b[3]/a[3,3])-soma=(6.0000/10.0000)-(-0.3996)=0.9996
Erro relativo máximo:Er[7]=|x3[7]-x3[6]|/|x3[7]*100=(|0.9996-(1.0003)|/|0.9996|)*100=0.0651

ITERAÇÃO	x1	x2	x3	ERRO RELATIVO(%)

1	0.7000	-1.6000	0.6000	100.0000
2	0.9600	-1.8600	0.9400	36.1702
3	0.9780	-1.9800	0.9660	6.0606
4	0.9994	-1.9888	0.9984	3.2452
5	0.9979	-1.9996	0.9968	0.5381
6	1.0002	-1.9989	1.0003	0.3523
7	0.9998	-2.0001	0.9996	0.0651

A seguir apresenta o algoritmo principal para a resolução de sistema linear pelo método de Gauss-Jacobi:

```
Dado n, A[n,n],b[n,1],xi[n,k], nit e Era
//ler a solução inicial como x(„0,1 ), x(„0,2 ) ... x(„0,n ) – iteração 0
Er(0)=100 //erro relativo inicial, para entrar no "loop enquanto"
k=1 //número da iteração
Enquanto Er(k-1)>Era faça //para erro relativo maior que o alvo
Para i=1 até n faça //para cada linha
soma=0
 Para j=1 até n faça //para cada coluna
  Se j==i então //não na diagonal
   Se aii=0 então // na diagonal
   "Erro: Algum elemento da diagonal é nulo."
    PARE
    Fim do condicional
   soma=soma+a(i,j)*x(k-1,j)/a(i,i)
   Fim do condicional
 x(k,i)=(b(i)/a(i,i))-soma
 Fim do laço
Fim do laço
Er = (|x(k,1)-x(k-1,1)|/|x(k,1)|)*100
Para c=2 até n faça //para cada linha
 Se ((|x(k,c)-x(k-1,c)|/|x(k,c)|)*100 > Er(k)) então //encontra o maior erro relativo entre os x
 Er(k) = (|x(k,c)-x(k-1,c)|/|x(k,c)|)*100
 Fim do condicional
Fim do laço
k=k+1
 Se (k-1)=nit então //número máximo de iterações atingido
 SAIR DO ENQUANTO
 Fim do condicional
Fim do loop Enquanto
```

REFERÊNCIAS BIBLIOGRÁFICAS

RUGGIERO, M.A.G; LOPES, V.L.R. Cálculo numérico: aspectos teóricos e computacionais. 2.ed. São Paulo: Makron Books, 1996.

SWOKOWSKI. Cálculo com Geometria Analítica. São Paulo. Makron Books, 1995.

QUADROS, Régis S. de; BORTOLI, Álvaro L. de. Fundamentos de Cálculo Numérico para Engenheiros.

Porto Alegre, 2009. Disponível em < http:// .inf.ufrgs.br/ rlflupchinski/files/20112/NUMERICO/ numerico-bortoli.pdf >. Acesso em outubro de 2014.

Figura 11: Relatório em PDF gerado com o exemplo dos dados da Fig. 10.

3.3 Sistemas Lineares Via Gauss-Seidel

A página para Sistemas Lineares por Gauss-Seidel apresenta-se como a Fig. 12 a seguir.

Figura 12: Página para Sistemas Lineares por Gauss- Jacobi.

Para iniciar os cálculos, tem-se duas opções: ou gerar o sistema na própria página, a partir da ordem digitada (botão “GERAR”), e digitar os valores (Fig. 13 a seguir) de cada elemento ou criar um arquivo TXT, seguindo o padrão apresentado ao se clicar no botão “EXEMPLO de TXT” (Fig. 14 a seguir).

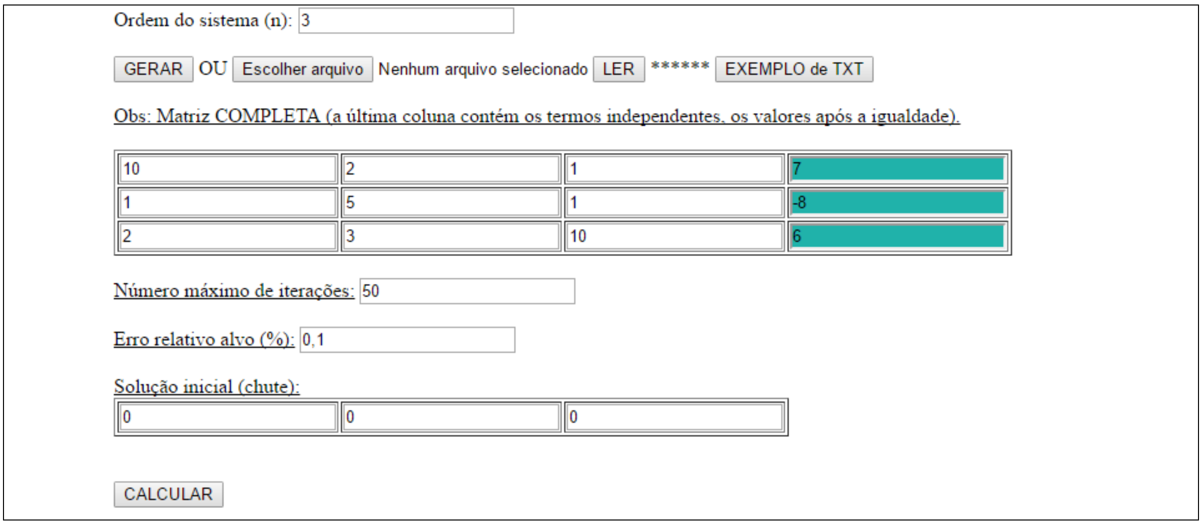


Figura 13: Exemplo de sistema gerado com ordem 3, cujos valores têm que ser digitados.

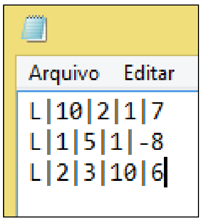


Figura 14: Exemplo de arquivo TXT que pode ser lido pela página.

Depois de ler ou digitar o sistema, clica-se no botão “CALCULAR” (veja Fig. 13anterior), aparecendo uma tabela com os resultados de cada iteração. A Fig. 15 a seguir apresenta o mesmo exemplo apresentado pelo método de Gauss-Jacobi, um sistema de ordem 3.

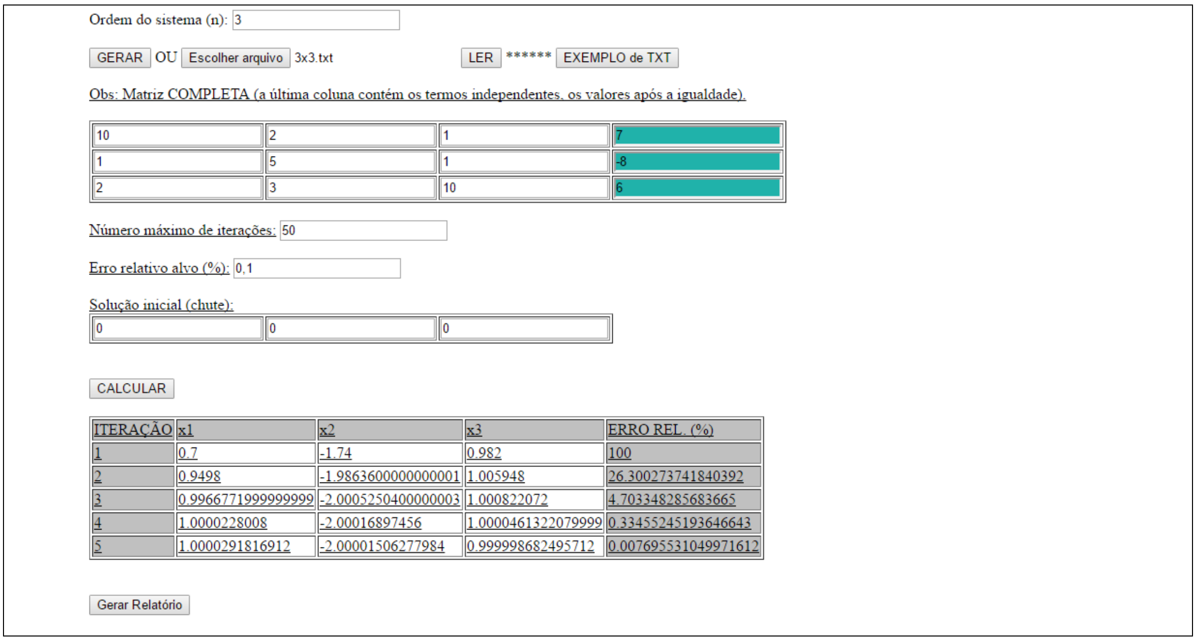


Figura 15: Exemplo da resolução por Gauss-Jacobi de um sistema de ordem 3.

Por fim, como principal característica da página criada, pode-se gerar o relatório, clicando no botão “Gerar Relatório” ao fim da página, conforme a Fig. 15 anterior. O relatório é em PDF e contém a resolução detalhada. A figura a seguir (Fig. 16) mostra o relatório para a matriz exemplificada na Fig. 13.

A matriz segue o modelo abaixo

$$\begin{bmatrix} a11 & a12 & \ldots & a1n & b1 \\ a21 & a22 & .. & a2n & b2 \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ an1 & an2 & \ldots & ann & bn \end{bmatrix}$$

Ordem da Matriz =3

Matriz Completa:

10.00	2.000	1.000	*7.000*
1.000	5.000	1.000	*-8.000*
2.000	3.000	10.00	*6.000*

O número máximo de iterações é: 50

O erro relativo alvo é: 0.1

A solução inicial, que é um valor estimado, é:

0.000	0.000	0.000

Para resolução do sistema linear pelo método de Gauss-Seidel, é necessário isolar o vetor x, de acordo com o exemplo abaixo:

$$\begin{cases} a_{11} * x_1 + a_{12} * x_2 + \cdots a_{1n} * x_n = b_1 \\ a_{21} * x_1 + a_{22} * x_2 + \cdots a_{2n} * x_n = b_2 \\ \ldots \\ a_{n1} * x_1 + a_{n2} * x_2 + \cdots a_{nn} * x_n = b_n \end{cases} \longrightarrow \begin{cases} x_1 = \frac{b_1}{a_{11}} - \frac{1}{a_{11}}(a_{12} * x_2 + \cdots a_{1n} * x_n) \\ x_2 = \frac{b_2}{a_{22}} - \frac{1}{a_{22}}(a_{21} * x_1 + \cdots a_{2n} * x_n) \\ \ldots \\ x_n = \frac{b_n}{a_{nn}} - \frac{1}{a_{nn}}(a_{n1} * x_1 + \cdots + a_{nn-1} * x_{n-1}) \end{cases}$$

Para prosseguir a resolução, é necessário escolher um valor inicial para todos os "x".
Na primeira iteração, basta substituir o valores iniciais de x na primeira equação e encontrar a solução de "x1".
Para encontrar os outros "x" da iteração 1, devem-se utilizar os "x" encontrados anteriormente na mesma iteração.
Na segunda iteração, devem-se utilizar os valores da iteração anterior para a primeira equação e, a partir da segunda equação, devem-se substituir os valores encontrados anteriormente.
E assim por diante,continuar subtituindo os valores encontrados até que haja uma convergência, ou seja, até que se encontre um erro menor do que o Erro Relativo alvo.
Para o caso de a solução demorar para convergir ou não convergir, determina-se um número máximo de iterações.

Obs.: Este método não permite matriz que tenha zeros na diogonal principal.

Abaixo segue a resolução:

soma=0
soma=soma+a[1,2]*x2[0]=0.0000+(2.0000*0.0000)=0.0000
soma=soma+a[1,3]*x3[0]=0.0000+(1.0000*0.0000)=0.0000
x1[1]=(b[1]/a[1,1])-soma=(7.0000/10.0000)-(0.0000)=0.7000
soma=0
soma=soma+a[2,1]*x1[0]=0.0000+(1.0000*0.7000)=0.1400
soma=soma+a[2,3]*x3[0]=0.1400+(1.0000*0.0000)=0.1400
x2[1]=(b[2]/a[2,2])-soma=(-8.0000/5.0000)-(0.1400)=-1.7400
soma=0
soma=soma+a[3,1]*x1[0]=0.0000+(2.0000*0.7000)=0.1400
soma=soma+a[3,2]*x2[0]=0.1400+(3.0000*-1.7400)=-0.3820
x3[1]=(b[3]/a[3,3])-soma=(6.0000/10.0000)-(-0.3820)=0.9820
Erro relativo máximo:Er[1]=|x1[1]-x1[0]|/|x1[1]*100=(|0.7000-(0.0000)|/|0.7000|)*100=100.0000

soma=0
soma=soma+a[1,2]*x2[1]=0.0000+(2.0000*-1.7400)=-0.3480
soma=soma+a[1,3]*x3[1]=-0.3480+(1.0000*0.9820)=-0.2498
x1[2]=(b[1]/a[1,1])-soma=(7.0000/10.0000)-(-0.2498)=0.9498
soma=0
soma=soma+a[2,1]*x1[1]=0.0000+(1.0000*0.9498)=0.1900
soma=soma+a[2,3]*x3[1]=0.1900+(1.0000*0.9820)=0.3864
x2[2]=(b[2]/a[2,2])-soma=(-8.0000/5.0000)-(0.3864)=-1.9864
soma=0
soma=soma+a[3,1]*x1[1]=0.0000+(2.0000*0.9498)=0.1900
soma=soma+a[3,2]*x2[1]=0.1900+(3.0000*-1.9864)=-0.4059
x3[2]=(b[3]/a[3,3])-soma=(6.0000/10.0000)-(-0.4059)=1.0059
Erro relativo máximo:Er[2]=|x1[2]-x1[1]|/|x1[2]*100=(|0.9498-(0.7000)|/|0.9498|)*100=26.3003

soma=0
soma=soma+a[1,2]*x2[2]=0.0000+(2.0000*-1.9864)=-0.3973
soma=soma+a[1,3]*x3[2]=-0.3973+(1.0000*1.0059)=-0.2967
x1[3]=(b[1]/a[1,1])-soma=(7.0000/10.0000)-(-0.2967)=0.9967
soma=0
soma=soma+a[2,1]*x1[2]=0.0000+(1.0000*0.9967)=0.1993
soma=soma+a[2,3]*x3[2]=0.1993+(1.0000*1.0059)=0.4005
x2[3]=(b[2]/a[2,2])-soma=(-8.0000/5.0000)-(0.4005)=-2.0005
soma=0
soma=soma+a[3,1]*x1[2]=0.0000+(2.0000*0.9967)=0.1993
soma=soma+a[3,2]*x2[2]=0.1993+(3.0000*-2.0005)=-0.4008
x3[3]=(b[3]/a[3,3])-soma=(6.0000/10.0000)-(-0.4008)=1.0008
Erro relativo máximo:Er[3]=|x1[3]-x1[2]|/|x1[3]*100=(|0.9967-(0.9498)|/|0.9967|)*100=4.7033

soma=0
soma=soma+a[1,2]*x2[3]=0.0000+(2.0000*-2.0005)=-0.4001
soma=soma+a[1,3]*x3[3]=-0.4001+(1.0000*1.0008)=-0.3000
x1[4]=(b[1]/a[1,1])-soma=(7.0000/10.0000)-(-0.3000)=1.0000
soma=0
soma=soma+a[2,1]*x1[3]=0.0000+(1.0000*1.0000)=0.2000
soma=soma+a[2,3]*x3[3]=0.2000+(1.0000*1.0008)=0.4002

x2[4]=(b[2]/a[2,2])-soma=(-8.0000/5.0000)-(0.4002)=-2.0002
soma=0
soma=soma+a[3,1]*x1[3]=0.0000+(2.0000*1.0000)=0.2000
soma=soma+a[3,2]*x2[3]=0.2000+(3.0000*-2.0002)=-0.4000
x3[4]=(b[3]/a[3,3])-soma=(6.0000/10.0000)-(-0.4000)=1.0000
Erro relativo máximo:Er[4]=|x1[4]-x1[3]|/|x1[4]*100=(|1.0000-(0.9967)|/|1.0000|)*100=0.3346

soma=0
soma=soma+a[1,2]*x2[4]=0.0000+(2.0000*-2.0002)=-0.4000
soma=soma+a[1,3]*x3[4]=-0.4000+(1.0000*1.0000)=-0.3000
x1[5]=(b[1]/a[1,1])-soma=(7.0000/10.0000)-(-0.3000)=1.0000
soma=0
soma=soma+a[2,1]*x1[4]=0.0000+(1.0000*1.0000)=0.2000
soma=soma+a[2,3]*x3[4]=0.2000+(1.0000*1.0000)=0.4000
x2[5]=(b[2]/a[2,2])-soma=(-8.0000/5.0000)-(0.4000)=-2.0000
soma=0
soma=soma+a[3,1]*x1[4]=0.0000+(2.0000*1.0000)=0.2000
soma=soma+a[3,2]*x2[4]=0.2000+(3.0000*-2.0000)=-0.4000
x3[5]=(b[3]/a[3,3])-soma=(6.0000/10.0000)-(-0.4000)=1.0000
Erro relativo máximo:Er[5]=|x2[5]-x2[4]|/|x2[5]*100=(|-2.0000-(-2.0002)|/|-2.0000|)*100=0.0077

ITERAÇÃO	x1	x2	x3	ERRO RELATIVO(%)
1	0.7000	-1.7400	0.9820	100.0000
2	0.9498	-1.9864	1.0059	26.3003
3	0.9967	-2.0005	1.0008	4.7033
4	1.0000	-2.0002	1.0000	0.3346
5	1.0000	-2.0000	1.0000	0.0077

A seguir apresenta o algoritmo principal para a resolução de sistema linear pelo método de Gauss-Seidel:

```
Dado n, A[n,n],b[n,1],xi[n,k], nit e Era
//ler a solução inicial como x(„0,1 ), x(„0,2 ) ... x(„0,n ) – iteração 0
1: Er(0)=100 //erro relativo inicial, para entrar no "loop enquanto"
2: k=1 //número da iteração
3: Enquanto Er(k-1)>Era faça //para erro relativo maior que o alvo
4: Para i=1 até n faça //para cada linha
5: soma=0
6:   Para j=1 até n faça //para cada coluna
7:     Se j==i então //não na diagonal
8:       Se aii=0 então // na diagonal
9:       "Erro: Algum elemento da diagonal é nulo."
10:        PARE
11:        Fim do condicional
12:    Se i<j então
13:    soma=soma+a(i,j)*x(k-1,j)/a(i,i)
14:    Fim do condicional
15:    Senão
16:    soma=soma+a(i,j)*x(k,j)/a(i,i)
17:    Fim do condicional
```

```
 x(k,i)=(b(i)/a(i,i))-soma
 Fim do laço
Fim do laço
Er = (|x(k,1)-x(k-1,1)|/|x(k,1)|)*100
Para c=2 até n faça //para cada linha
 Se (|x(k,c)-x(k-1,c)|/|x(k,c)|)*100 > Er(k) então //encontra o maior erro relativo entre os x
 Er(k) = (|x(k,c)-x(k-1,c)|/|x(k,c)|)*100
 Fim do condicional
Fim do laço
k=k+1
 Se (k-1)=nit então //número máximo de iterações atingido
 SAIR DO ENQUANTO
 Fim do condicional
Fim do loop Enquanto
```

REFERÊNCIAS BIBLIOGRÁFICAS

RUGGIERO, M.A.G; LOPES, V.L.R. Cálculo numérico: aspectos teóricos e computacionais. 2.ed. São Paulo: Makron Books, 1996.

SWOKOWSKI. Cálculo com Geometria Analítica. São Paulo. Makron Books, 1995.

QUADROS, Régis S. de; BORTOLI, Álvaro L. de. Fundamentos de Cálculo Numérico para Engenheiros. Porto Alegre, 2009. Disponível em < http://www.inf.ufrgs.br/ rlflupchinski/files/20112/NUMERICO/numerico-bortoli.pdf >. Acesso em outubro de 2014.

Figura 16: Relatório em PDF gerado com o exemplo dos dados da Fig. 15.

3.4 Interpolação Polinomial

A página para fazer interpolação polinomial de dados apresenta-se como a Fig.17 a seguir.

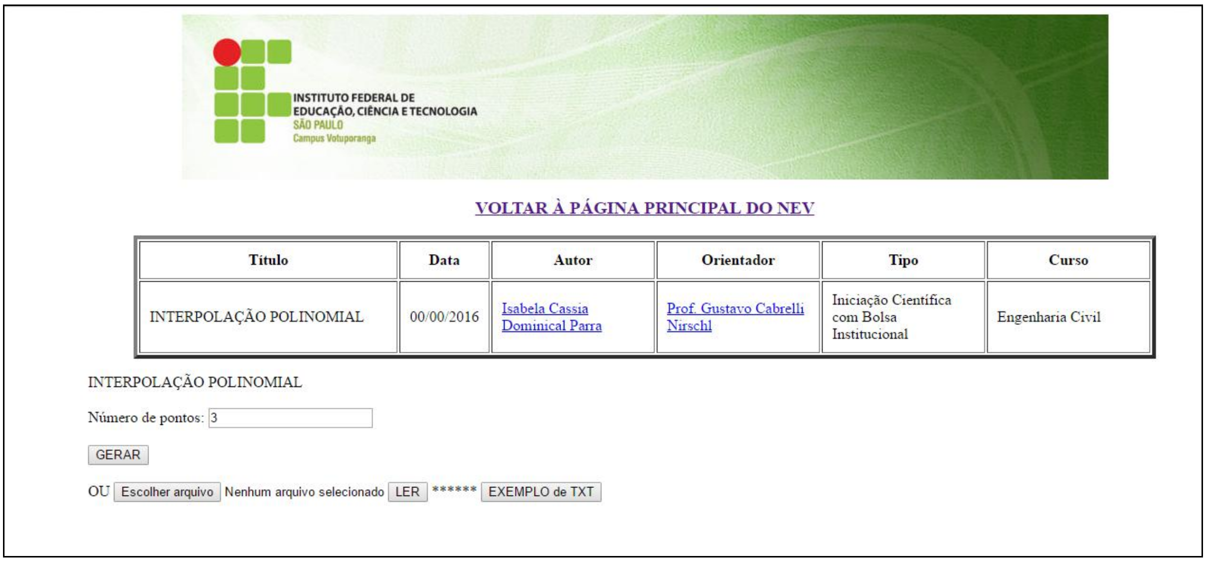


Figura 17: Página para Interpolação Polinomial.

Para iniciar os cálculos, tem-se duas opções: ou gerar a tabela de dados na própria página, a partir do número de pontos escolhido, clicando no botão “GERAR”, e digitar os valores (Fig. 18 a seguir) de cada coordenada dos pontos, ou criar um arquivo TXT, seguindo o padrão apresentado ao se clicar no botão “EXEMPLO de TXT” (Fig. 19 a seguir).

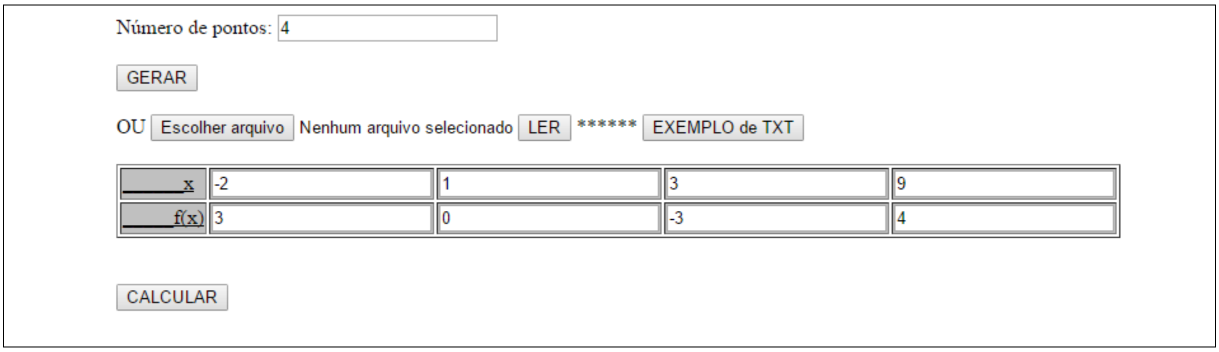


Figura 18: Tabela de pontos digitada na página.

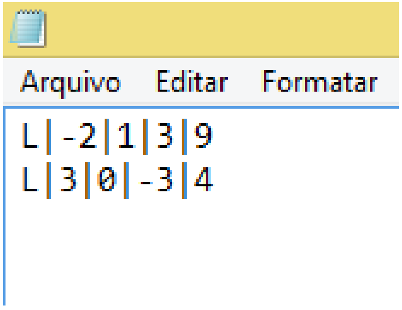


Figura 19:Exemplo de TXT

Depois de ler ou digitar a tabela de pontos, clica-se no botão “CALCULAR”, mostrado na Fig. 18 anterior, e irá aparecer o polinômio formado e o gráfico gerado a partir dos pontos apresentados. A Fig. 20 a seguir apresenta um exemplo da tabela com 4 pontos da Fig. 18.

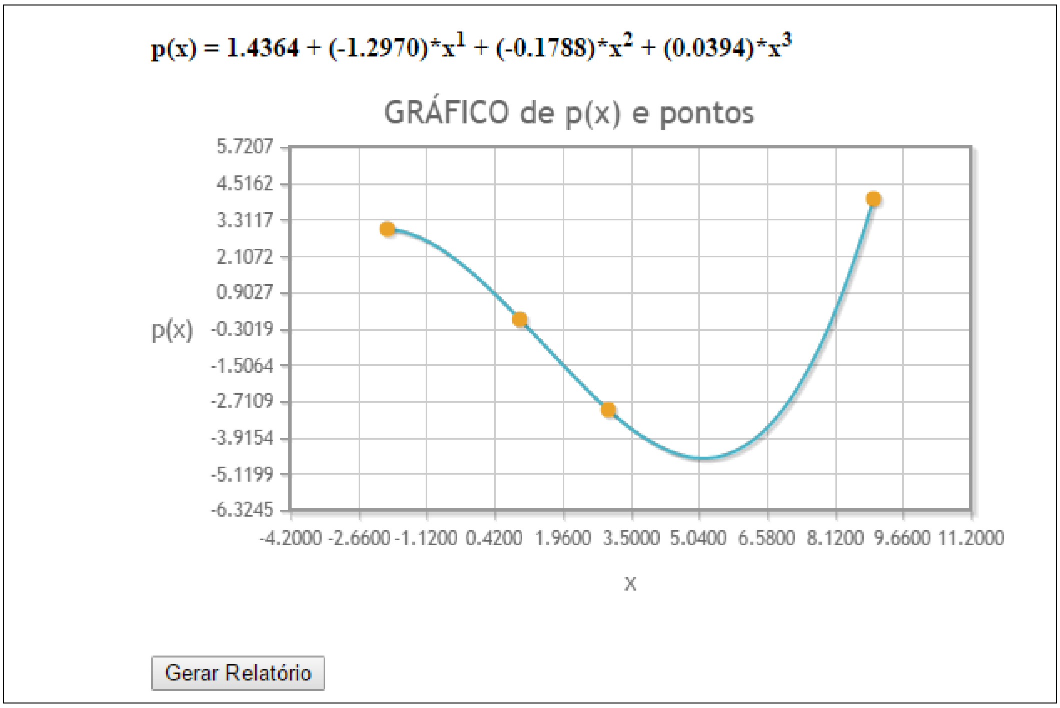


Figura 20:Polinômio formado e a Interpretação gráfica – Resultado Final.

Por fim, como principal objetivo para a criação da página, é possível gerar o relatório, clicando no botão “Gerar Relatório”, depois do resultado final, conforme a Fig. 20 anterior. O relatório é em PDF e contém a resolução detalhada, além de uma breve explicação teórica e o algoritmo principal para o estudo de cálculo numérico. A figura a seguir, Fig. 21, mostra o relatório para os pontos exemplificados na Fig. 18.

Número de pontos=4

Tabela de Pontos:

x	f(x)
-2.0000	3.0000
1.0000	0.0000
3.0000	-3.0000
9.0000	4.0000

Para se obter um polinômio de grau n, por meio da interpolação polinomial, é necessário n+1 pontos do tipo (x,f(x)).
Com os pontos devem-se formar um sistema igual o da figura abaixo:

$$\begin{cases} a_0 + a_1x_0 + a_2x_0^2 + \cdots + a_nx_0^n &= f(x_0) \\ a_0 + a_1x_1 + a_2x_1^2 + \cdots + a_nx_1^n &= f(x_1) \\ & \vdots \\ a_0 + a_1x_n + a_2x_n^2 + \cdots + a_nx_n^n &= f(x_n) \end{cases}$$

Com o sistema formado, é preciso montar a matriz, igual a que segue abaixo, resolvê-la e encontrar o polinômio.

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^n \\ 1 & x_1 & x_1^2 & \ldots & x_1^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^n \end{bmatrix}$$

O polinômio formado será igual a Pn(x)=a0+a1*x+...+an*(x^n).
Aqui o sistema será resolvido por Eliminação de Gauss.

Abaixo segue o sistema formado:

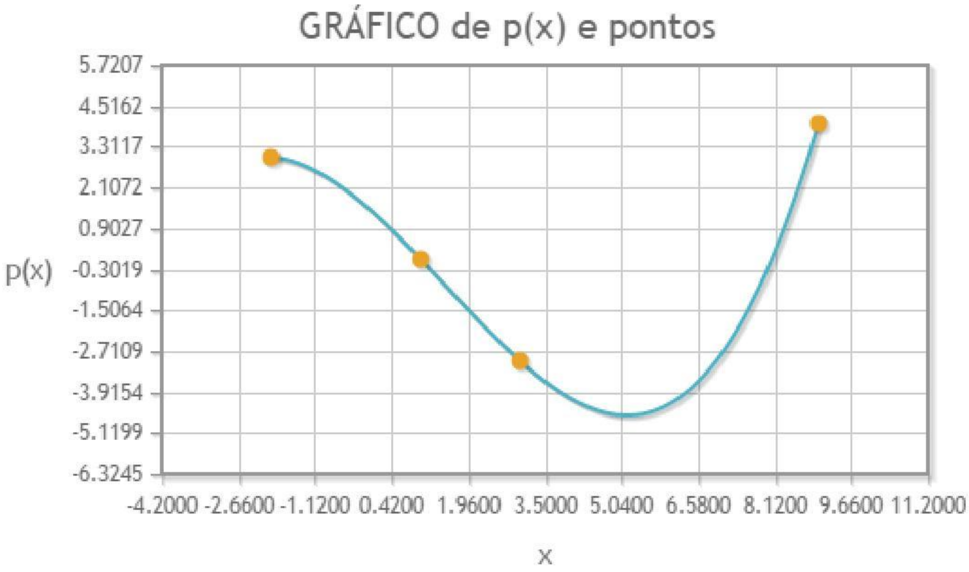
1.0000	-2.0000	4.0000	-8.0000	*3.0000*
1.0000	1.0000	1.0000	1.0000	*0.0000*
1.0000	3.0000	9.0000	27.0000	*-3.0000*
1.0000	9.0000	81.0000	729.0000	*4.0000*

Este sistema pode ser resolvido, por exemplo, por Gauss, que pode ser estudado em http://vtp.ifsp.edu.br/nev/Sistema-gauss/sistemagauss.html. Para tanto, utilize o arquivo txt gerado juntamente com este relatório, que pode ser lido pelo referido programa.

Resolvendo o sistema acima encontra se os seguintes valores de a:
a3=0.0394
a2=-0.1788
a1=-1.2970
a0=1.4364

Com isso o polinômio formado é:
p(x) = 1.4364 + (-1.2970)*x^1 + (-0.1788)*x^2 + (0.0394)*x^3

Segue abaixo gráfico da função:



A seguir apresenta o algoritmo principal para montar a matriz dos coeficientes para encontrar o polinômio interpolador:

```
Para i=1 até n+1 faça //linhas
  Para j=1 até n+1 faça //colunas
  a[i,j]=X[i-j]^(j-1)
  fim do para j
Fim do para i
Execute o algoritmo de Eliminação de Gauss. //Acesse http://vtp.ifsp.edu.br/nev/Sistema-gauss/sistemagauss.html
```

REFERÊNCIAS BIBLIOGRÁFICAS

RUGGIERO, M.A.G; LOPES, V.L.R. Cálculo numérico: aspectos teóricos e computacionais. 2.ed. São Paulo: Makron Books, 1996.

SWOKOWSKI. Cálculo com Geometria Analítica. São Paulo. Makron Books, 1995.

QUADROS, Régis S. de; BORTOLI, Álvaro L. de. Fundamentos de Cálculo Numérico para Engenheiros. Porto Alegre, 2009. Disponível em < http://www.inf.ufrgs.br/~rlflupchinski/files/20112/NUMERICO/numerico-bortoli.pdf >. Acesso em outubro de 2014.

Figura 21: Relatório gerado em PDF para o exemplo apresentado na Fig. 18.

Nota-se, no relatório da Fig. 21 anterior, que é necessário resolver um sistema linear para encontrar o polinômio interpolador. Como o objetivo deste programa é mostrar o passo-a-passo da obtenção do polinômio a partir de uma tabela de dados e não necessariamente como se resolve um sistema linear, estes cálculos são aqui ocultos. Porém, caso o usuário tenha interesse, é gerado o arquivo txt com os dados do sistema linear a ser resolvido, sendo que outra página de internet já publicada criada por pesquisadores deste grupo (http://vtp.ifsp.edu.br/nev/) é capaz de ler este arquivo e resolver o sistema via Método de Gauss, mostrando todos os cálculos.

3.5. Interpolação por Lagrange

A página para fazer Interpolação Polinomial por Lagrangeapresenta-se como a Fig.22 a seguir.

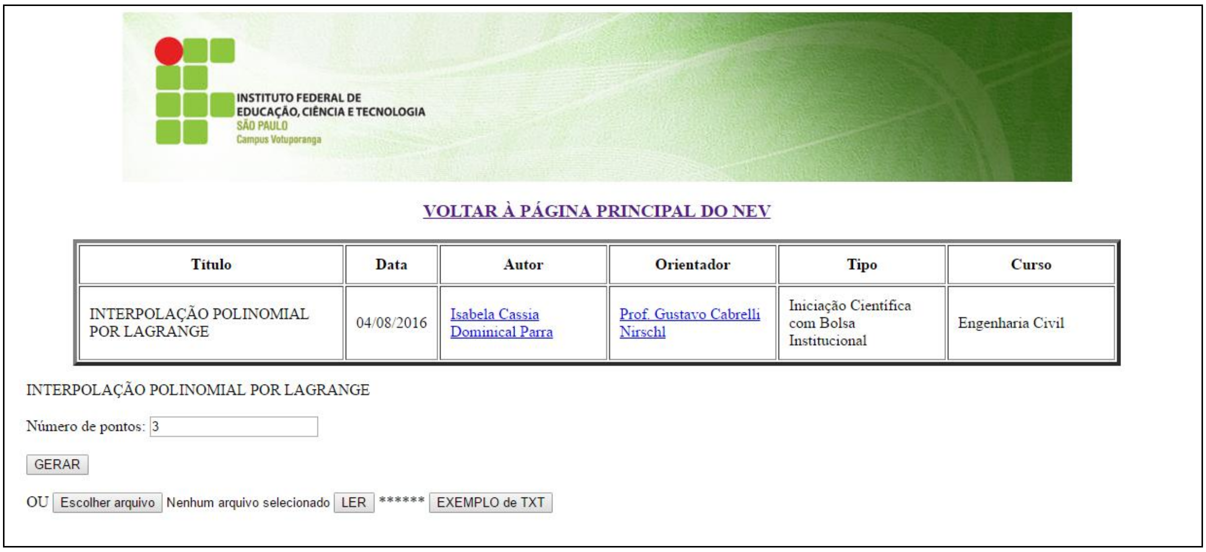


Figura 22: Página para Interpolação Polinomial por Lagrange.

Para iniciar os cálculos, tem-se duas opções: ou gerar a tabela de dados na própria página, a partir do número de pontos escolhido, clicando no botão “GERAR”, e digitar os valores (Fig. 23 a seguir) de cada coordenada dos pontos, ou criar um arquivo TXT, seguindo o padrão apresentado ao se clicar no botão “EXEMPLO de TXT” (Fig. 24 a seguir).

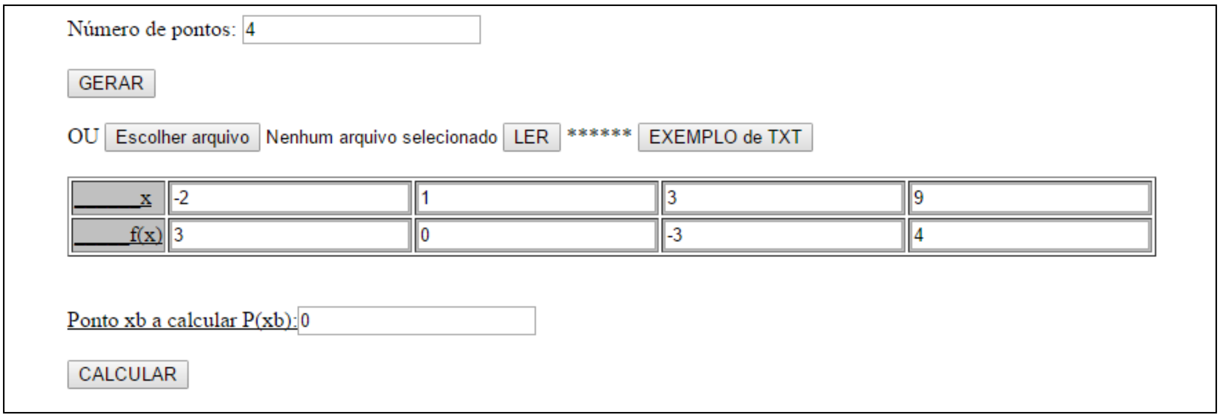


Figura 23: Tabela de pontos digitada na página.

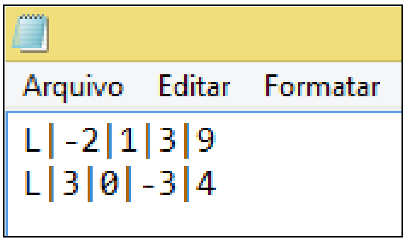


Figura 24:Exemplo de TXT

Depois de ler ou digitar a tabela de pontos, clica-se no botão “CALCULAR”, mostrado na Fig. 23 anterior, e irá aparecer o resultado do P(xb), uma vez que com esse método computacionalmente é difícil encontrar o polinômio e o gráfico gerado a partir dos pontos apresentados. A Fig. 25 a seguir apresenta um exemplo da tabela com 4 pontos da Fig. 23.

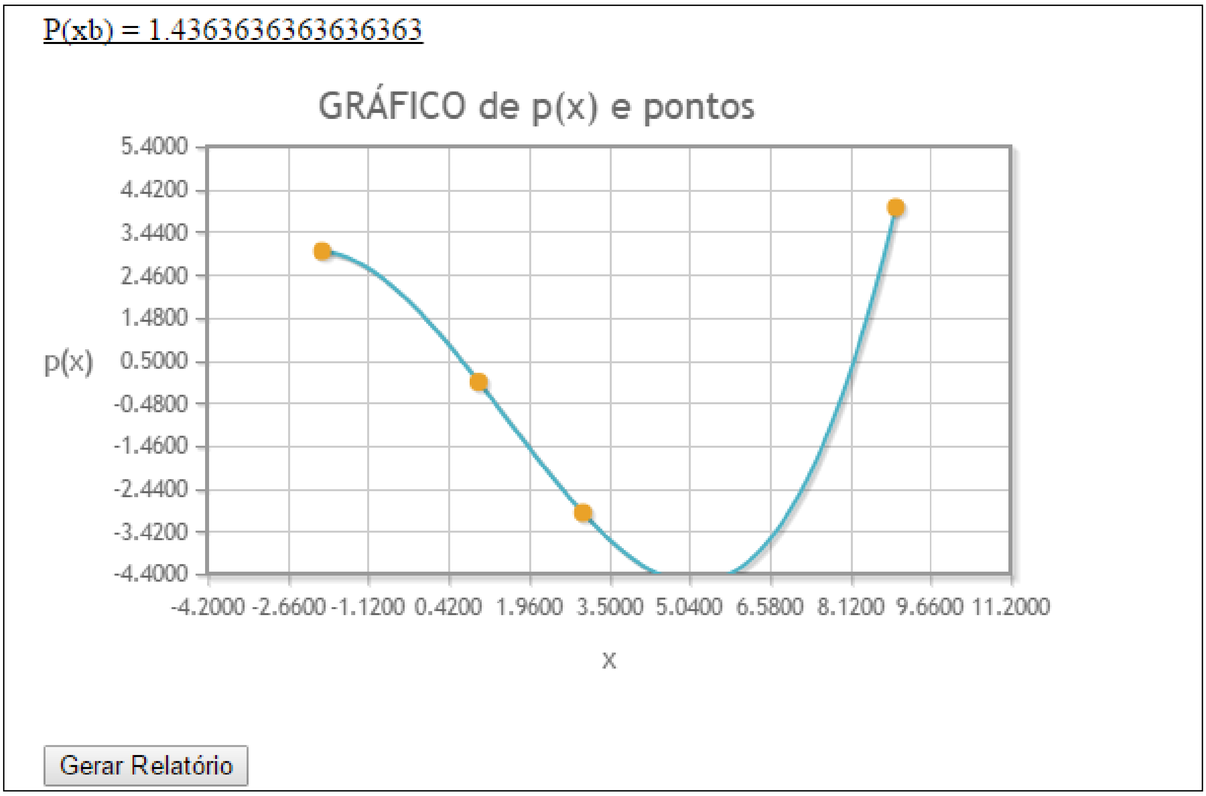


Figura 25: Valor do polinômio no ponto xb e a Interpretação gráfica – Resultado Final.

Por fim, como principal objetivo para a criação da página, é possível gerar o relatório, clicando no botão “Gerar Relatório”, depois do resultado final, conforme a Fig. 25 anterior. O relatório é em PDF e contém a resolução detalhada, além de uma breve explicação teórica e o algoritmo principal para o estudo de cálculo numérico. A figura a seguir, Fig. 26, mostra o relatório para os pontos exemplificados na Fig. 23.

Número de pontos=4

Tabela de Pontos:

x	f(x)
-2.0000	3.0000
1.000	0.000
3.0000	-3.0000
9.0000	4.0000

Por este método, o polinômio interpolador depende de outros polinômios “especiais”.
O polinômio formado é do tipo: "P(x)=y[0]*L[0](x)+y[1]*L[1](x)+...+y[n]*L[n](x)".
No caso, n é o número de pontos a interpolar menos 1. Os polinômios de Lagrange (L[k](x)) são da forma apresentada a seguir:

$$L_k(x) = \frac{(x-x_0)(x-x_1)\dots(x-x_{k-1})(x-x_{k+1})\dots(x-x_n)}{(x_k-x_0)(x_k-x_1)\dots(x_k-x_{k-1})(x_k-x_{k+1})\dots(x_k-x_n)}$$

Abaixo segue a resolução para encontrar o polinômio interpolador:

L[0](x)=(x-(x[1]))(x-(x[2]))(x-(x[3]))/(x[0]-(x[1]))(x[0]-(x[2]))(x[0]-(x[3]))=
= (x-(1.000))(x-(3.0000))(x-(9.0000))/-165.0000
L[1](x)=(x-(x[0]))(x-(x[2]))(x-(x[3]))/(x[1]-(x[0]))(x[1]-(x[2]))(x[1]-(x[3]))=
= (x-(-2.0000))(x-(3.0000))(x-(9.0000))/48.0000
L[2](x)=(x-(x[0]))(x-(x[1]))(x-(x[3]))/(x[2]-(x[0]))(x[2]-(x[1]))(x[2]-(x[3]))=
= (x-(-2.0000))(x-(1.000))(x-(9.0000))/-60.0000
L[3](x)=(x-(x[0]))(x-(x[1]))(x-(x[2]))/(x[3]-(x[0]))(x[3]-(x[1]))(x[3]-(x[2]))=
= (x-(-2.0000))(x-(1.000))(x-(3.0000))/528.0000

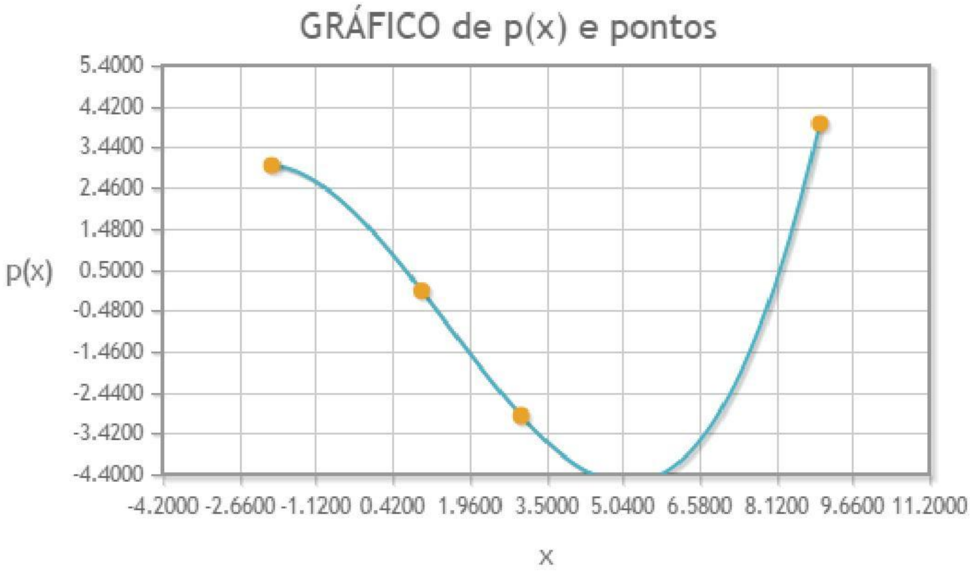
**P[4](x)=f[0](x)*L[0](x)+f[1](x)*L[1](x)+f[2](x)*L[2](x)+f[3](x)*L[3](x)=
= (3*(x-(1.000))(x-(3.0000))(x-(9.0000))/-165.0000)+(0*(x-(-2.0000))(x-(3.0000))(x-(9.0000))/48.0000)+(-3*(x-(-2.0000))(x-(1.000))(x-(9.0000))/-60.0000)+4*(x-(-2.0000))(x-(1.000))(x-(3.0000))/528.0000**

Uma vez que computacionalmente é difícil encontrar o polinômio, a seguir segue a resolução realizada para resolver P(x) num ponto específico, no caso, xb=0.000, baseada no algoritmo apresentado ao fim do PDF:

Pn=0
sup=1
inf=1
sup=sup*(xb-xp[1])=1*(0.000-(1.000))=-1.000
inf=inf*(xp[0]-xp[1])=1*(-2.0000-(1.000))=-3.0000
sup=sup*(xb-xp[2])=-1.000*(0.000-(3.0000))=3.0000

```
inf=inf*(xp[0]-xp[2])=-3.0000*(-2.0000-(3.0000))=15.0000
sup=sup*(xb-xp[3])=3.0000*(0.000-(9.0000))=-27.0000
inf=inf*(xp[0]-xp[3])=15.0000*(-2.0000-(9.0000))=-165.0000
Pn=((sup/inf)*fxp[0])+Pn=((-27.0000/-165.0000)*3)+0.000=0.4909
sup=1
inf=1
sup=sup*(xb-xp[0])=1*(0.000-(-2.0000))=2.0000
inf=inf*(xp[1]-xp[0])=1*(1.000-(-2.0000))=3.0000
sup=sup*(xb-xp[2])=2.0000*(0.000-(3.0000))=-6.0000
inf=inf*(xp[1]-xp[2])=3.0000*(1.000-(3.0000))=-6.0000
sup=sup*(xb-xp[3])=-6.0000*(0.000-(9.0000))=54.0000
inf=inf*(xp[1]-xp[3])=-6.0000*(1.000-(9.0000))=48.0000
Pn=((sup/inf)*fxp[1])+Pn=((54.0000/48.0000)*0)+0.4909=0.4909
sup=1
inf=1
sup=sup*(xb-xp[0])=1*(0.000-(-2.0000))=2.0000
inf=inf*(xp[2]-xp[0])=1*(3.0000-(-2.0000))=5.0000
sup=sup*(xb-xp[1])=2.0000*(0.000-(1.000))=-2.0000
inf=inf*(xp[2]-xp[1])=5.0000*(3.0000-(1.000))=10.0000
sup=sup*(xb-xp[3])=-2.0000*(0.000-(9.0000))=18.0000
inf=inf*(xp[2]-xp[3])=10.0000*(3.0000-(9.0000))=-60.0000
Pn=((sup/inf)*fxp[2])+Pn=((18.0000/-60.0000)*-3)+0.4909=1.3909
sup=1
inf=1
sup=sup*(xb-xp[0])=1*(0.000-(-2.0000))=2.0000
inf=inf*(xp[3]-xp[0])=1*(9.0000-(-2.0000))=11.0000
sup=sup*(xb-xp[1])=2.0000*(0.000-(1.000))=-2.0000
inf=inf*(xp[3]-xp[1])=11.0000*(9.0000-(1.000))=88.0000
sup=sup*(xb-xp[2])=-2.0000*(0.000-(3.0000))=6.0000
inf=inf*(xp[3]-xp[2])=88.0000*(9.0000-(3.0000))=528.0000
Pn=((sup/inf)*fxp[3])+Pn=((6.0000/528.0000)*4)+1.3909=1.4364
```

Segue abaixo gráfico da função dada a partir dos pontos:



A seguir apresenta-se o algoritmo principal para encontrar o valor de P(xb):

```
Pn=0;
para j=0 até n {
sup=1;
inf=1;
para i=0 até n {
se i ≠ j então
sup=sup*(xb-xp[i]);
inf=inf*(xp[j]-xp[i]);
} fim do condicional
} fim do para i
Pn=((sup/inf)*fxp[j])+Pn;
} fim do para j
```

REFERÊNCIAS BIBLIOGRÁFICAS

RUGGIERO, M.A.G; LOPES, V.L.R. Cálculo numérico: aspectos teóricos e computacionais. 2.ed. São Paulo: Makron Books, 1996.

SWOKOWSKI. Cálculo com Geometria Analítica. São Paulo. Makron Books, 1995.

QUADROS, Régis S. de; BORTOLI, Álvaro L. de. Fundamentos de Cálculo Numérico para Engenheiros. Porto Alegre, 2009. Disponível em < http://www.inf.ufrgs.br/~rlflupchinski/files/20112/NUMERICO/numerico-bortoli.pdf >. Acesso em outubro de 2014.

Figura 26: Relatório gerado em PDF para o exemplo apresentado na Fig. 23.

3.6. Interpolação pela Forma de Newton

A página para fazer Interpolação Polinomial pela Forma de Newton apresenta-se como a Fig.27 a seguir.

Figura 27: Página para Interpolação Polinomialpela Forma de Newton.

Para iniciar os cálculos, tem-se duas opções: ou gerar a tabela de dados na própria página, a partir do número de pontos escolhido, clicando no botão “GERAR”, e digitar os valores (Fig. 28 a seguir) de cada coordenada dos pontos, ou criar um arquivo TXT, seguindo o padrão apresentado ao se clicar no botão “EXEMPLO de TXT” (Fig. 29 a seguir).

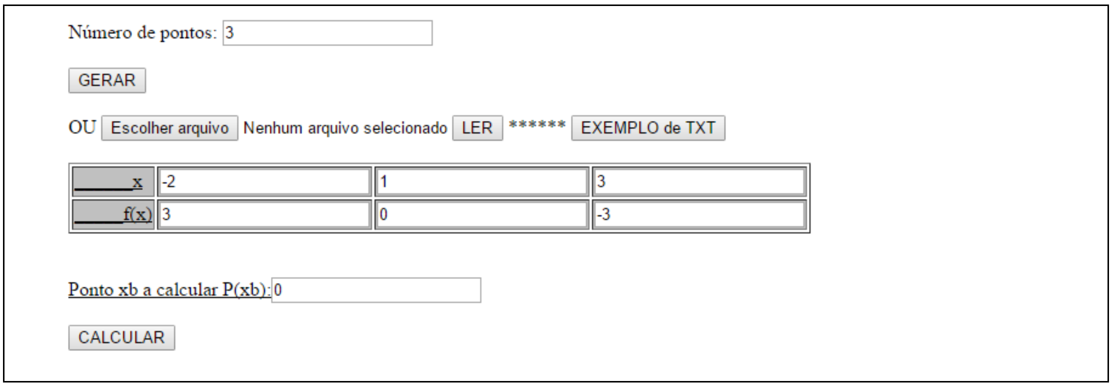


Figura 28: Tabela de pontos digitada na página.

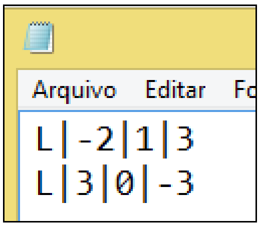


Figura 29: Exemplo de TXT

Depois de ler ou digitar a tabela de pontos, clica-se no botão “CALCULAR”, mostrado na Fig. 28 anterior, e irá aparecer o resultado do P(xb), uma vez que com esse método computacionalmente é difícil encontrar o polinômio e o gráfico gerado a partir dos pontos apresentados. A Fig. 25 a seguir apresenta um exemplo da tabela com 3 pontos da Fig. 28.

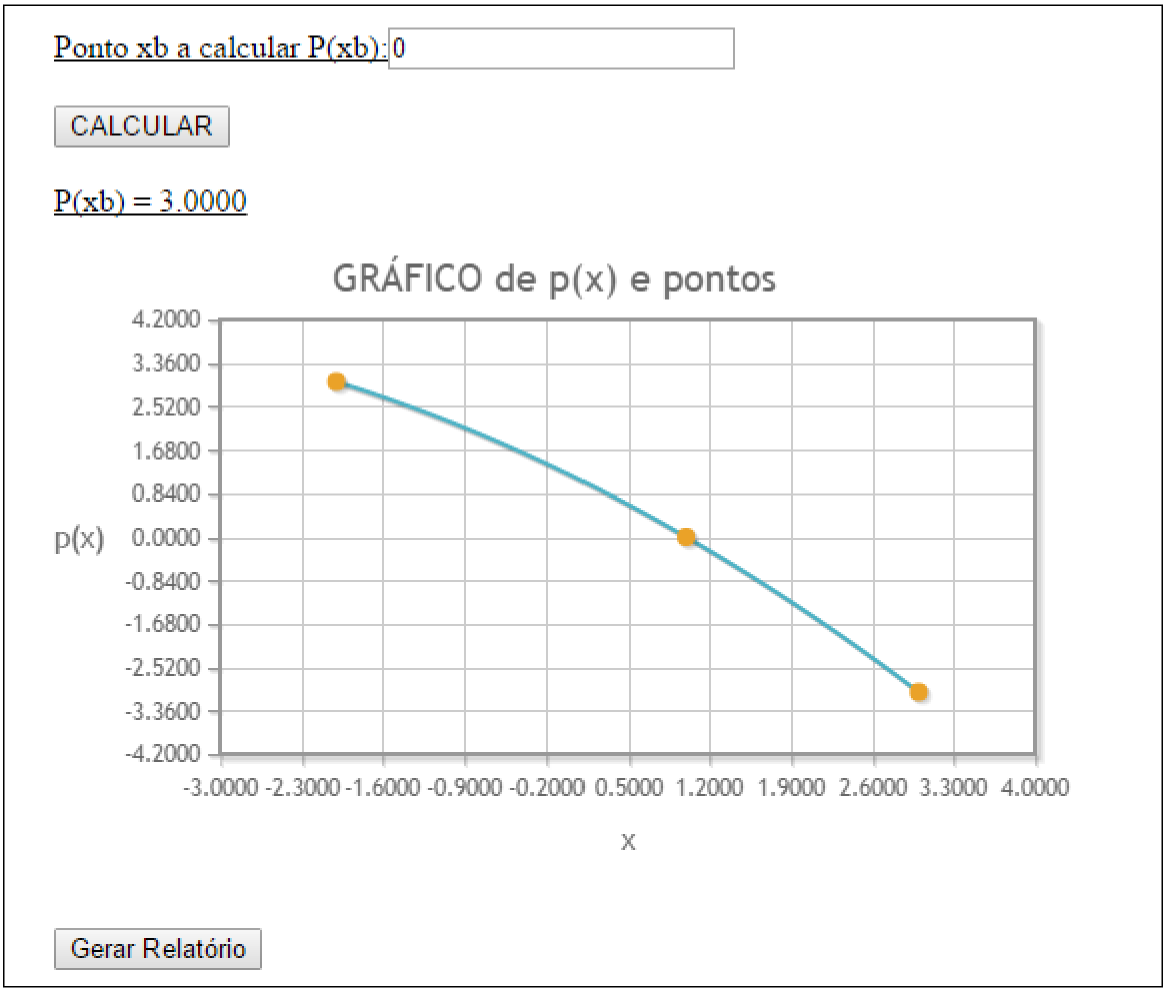


Figura 30: Valor do polinômio no ponto xb e a Interpretação gráfica – Resultado Final.

Por fim, como principal objetivo para a criação da página, é possível gerar o relatório, clicando no botão “Gerar Relatório”, depois do resultado final, conforme a Fig. 30 anterior. O relatório é em PDF e contém a resolução detalhada, além de uma breve explicação teórica e o algoritmo principal para o estudo de cálculo numérico. A figura a seguir, Fig. 31, mostra o relatório para os pontos exemplificados na Fig. 28.

NEV: Núcleo de Engenharia Virtual
Título: Interpolação Polinomial por Forma de Newton
Data: 08/09/2016
Autor: Isabela Cassia Dominical Parra
Orientador: Prof. Gustavo Cabrelli Nirschl
Tipo: Iniciação Científica
Curso: Engenharia Civil

Número de pontos=3

Tabela de Pontos:

x	f(x)
-2.0000	3.0000
1.000	0.000
3.0000	-3.0000

Por este método, o polinômio interpolador é da seguinte forma:
"P(x)=d[0]+ d[1]*(x-x[0])+ d[2]*(x-x[0])*(x-x[1])+...+d[n]*(x-x[0])*(x-x[1])*...*(x-x[n-1])"
Em que, n é o número de pontos a interpolar menos 1. Os coeficientes d[k] são chamados de diferenças divididas apresentadas a seguir:

$$k=0 \rightarrow d_0 = f[x_0]$$
$$k=1 \rightarrow d_1 = f[x_1, x_0]$$
$$k=2 \rightarrow d_2 = f[x_2, x_1, x_0]$$
$$(...)$$
$$k=n \rightarrow d_n = f[x_n, x_{n-1}, ..., x_0]$$

As diferenças divididas são resolvidas como mostra abaixo:

$$f[x_0] = f(x_0)$$
$$(...)$$
$$f[x_n] = f(x_n)$$
****
$$f[x_1, x_0] = \frac{f[x_1] - f[x_0]}{x_1 - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$
$$f[x_2, x_1] = \frac{f[x_2] - f[x_1]}{x_2 - x_1} = \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$
$$(...)$$
$$f[x_n, x_{n-1}] = \frac{f[x_n] - f[x_{n-1}]}{x_n - x_{n-1}} = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$
****
$$f[x_2, x_1, x_0] = \frac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0} = \frac{calculados\_anteriormente}{x_2 - x_0}$$
$$f[x_3, x_2, x_1] = \frac{f[x_3, x_2] - f[x_2, x_1]}{x_3 - x_1} = \frac{calculados\_anteriormente}{x_3 - x_1}$$
$$(...)$$
$$f[x_n, x_{n-1}, x_{n-2}] = \frac{f[x_n, x_{n-1}] - f[x_{n-1}, x_{n-2}]}{x_n - x_{n-2}} = \frac{calculados\_anteriormente}{x_n - x_{n-2}}$$

Abaixo segue a resolução das diferenças divididas:

f[x0]=f(x0)= 3.0000= d[0]
f[x1]=f(x1)=0.000

f[x2]=f(x2)=-3.0000
f[x1;x0]=(f[x1]-f[x0])/(x1-x0)=(0.000-(3.0000))/3.0000=-1.000= d[1]
f[x2;x1]=(f[x2]-f[x1])/(x2-x1)=(-3.0000-(0.000))/2.0000=-1.5000
f[x2;x1;x0]=(f[x2;x1]-f[x1;x0])/(x2-x0)=(-1.5000-(-1.000))/5.0000=-0.1000= d[2]

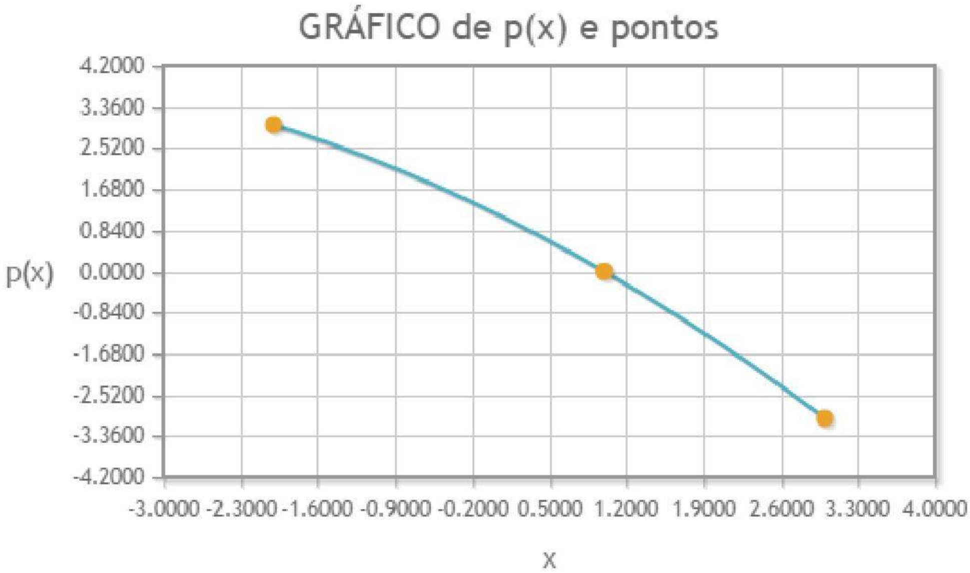
P(x) = d[0]+d[1]*(x-x[0])+d[2]*(x-x[0])(x-x[1]) =
= 3.0000+(-1.000*(x-(-2)))+(-0.1000*(x-(-2))(x-(1)))

Uma vez que, computacionalmente, é dificil simplificar o polinômio, a seguir segue a resolução realizada para resolver P(x) num ponto específico, no caso, xb=0.000, baseada no algoritmo apresentado ao fim do PDF:

lmax=2
fn[0,0]=f(x0)= 3.0000
fn[1,0]=f(x1)= 0.000
fn[2,0]=f(x2)= -3.0000
lmax=1
fn[0,1]=(fn[1,0]-fn[0,0])/(xp[1]-xp[0])=(0.000-(3.0000))/(1.000-(-2.0000))=-1.000
fn[1,1]=(fn[2,0]-fn[1,0])/(xp[2]-xp[1])=(-3.0000-(0.000))/(3.0000-(1.000))=-1.5000
fn[0,2]=(fn[1,1]-fn[0,1])/(xp[2]-xp[0])=(-1.5000-(-1.000))/(3.0000-(-2.0000))=-0.1000

Pn = d[0]+d[1]*(xb-x[0])+d[2]*(xb-x[0])(xb-x[1]) =
= 3.0000+(-1.000*(0.000-(-2)))+(-0.1000*(0.000-(-2))(0.000-(1.000)))=3.0000

Segue abaixo gráfico da função dada a partir dos pontos:



A seguir apresenta o algoritmo principal para encontrar o valor de P(xb):

1: //coluna c=0 da tabela de diferenças divididas
2: lmax=n-1; //linha máxima

```
3: fn=[]; //criar o vetor das diferenças divididas
4: para l=0 até lmax {
5: fn[l,0]=fxp[l];
6: } fim do para l
7: lmax=lmax-1;
8: //colunas 1 a np-1 da tabela de diferenças divididas
9: para c=1 até n {
10: para l=0 até lmax {
11: fn[l,c]=(fn[(l+1),(c-1)]-fn[l,(c-1)])/(xp[l+c]-xp[l]);
12: } fim do para l
13: lmax=lmax-1;
14: } fim do para c
15: //cálculo do polinômio no ponto xb
16: Pn=fn[0,0];
17: Pn=Number(Pn); //para garantir que seja número (necessário no Javascript)
18: m=1;
19: para c=1 até n {
20: m=m*(xb-xp[c-1]);
21: Pn=Pn+(m*fn[0,c]);
22: } fim do para c
```

REFERÊNCIAS BIBLIOGRÁFICAS

RUGGIERO, M.A.G; LOPES, V.L.R. Cálculo numérico: aspectos teóricos e computacionais. 2.ed. São Paulo: Makron Books, 1996.

SWOKOWSKI. Cálculo com Geometria Analítica. São Paulo. Makron Books, 1995.

QUADROS, Régis S. de; BORTOLI, Álvaro L. de. Fundamentos de Cálculo Numérico para Engenheiros. Porto Alegre, 2009. Disponível em < http://www.inf.ufrgs.br/~rlflupchinski/files/20112/NUMERICO/numerico-bortoli.pdf >. Acesso em outubro de 2014.

Figura 31: Relatório gerado em PDF parao exemplo apresentado na Fig. 28.

3.7. Integração pela Regra dos Trapézios

3.7.1. Pela Função

A página para fazer Integração pela Regra do Trapézio a partir de uma função apresenta-se como a Fig.32 a seguir.

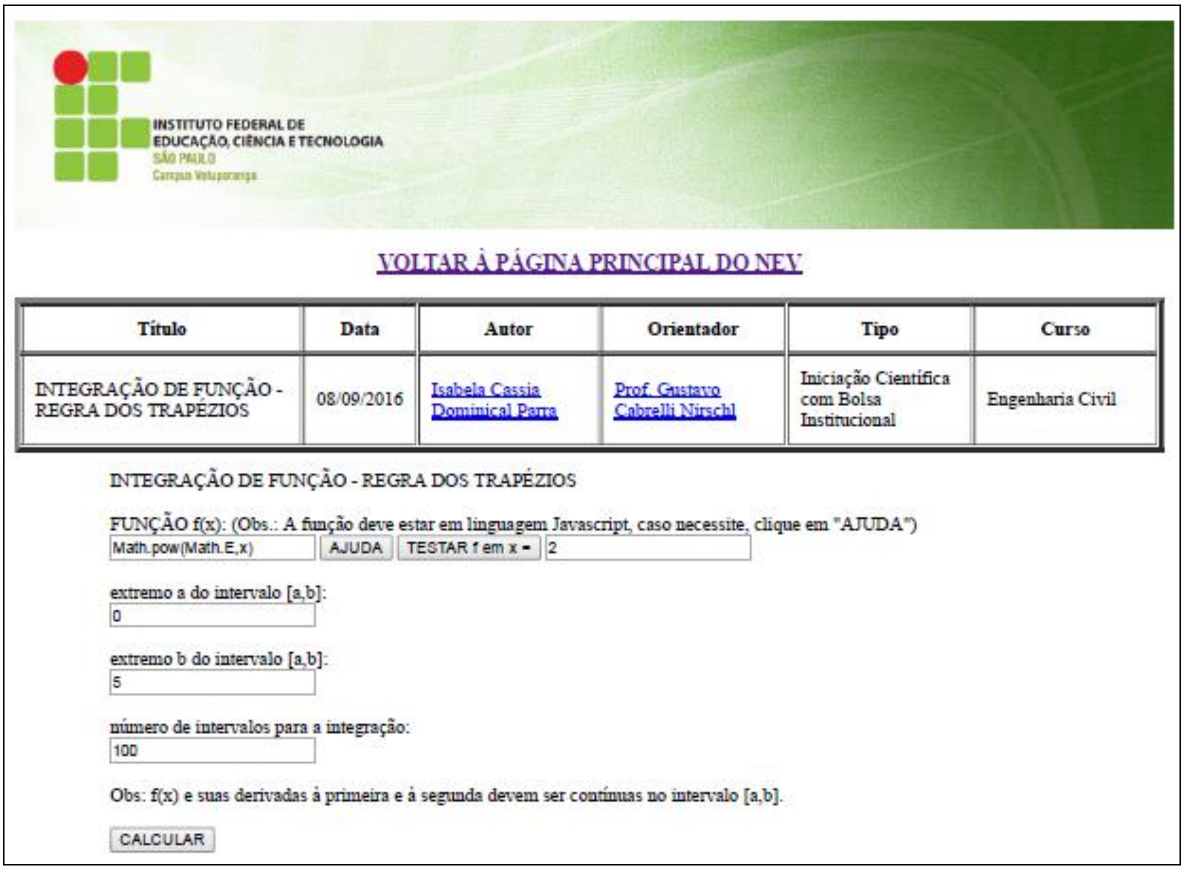


Figura 32: Página para Integração pela Regra do Trapézio a partir de uma função.

Para iniciar os cálculos é preciso escrever a função, por meio do botão "AJUDA", é possível visualizar as principais funções matemáticas na linguagem Javascript, como esta deverá ser escrita (Fig. 33 abaixo). O botão "TESTAR f em x=" é para testar se a função foi digitada corretamente na linguagem Javascript. Depois de digitado a função, escolhido o intervalo de integração e o número de intervalos, clica-se no botão "CALCULAR", mostrado na Fig. 32 anterior, e irá aparecer o resultado da integral e o gráfico gerado a partir da função. A Fig. 34 a seguir apresenta um exemplo da Fig. 32.

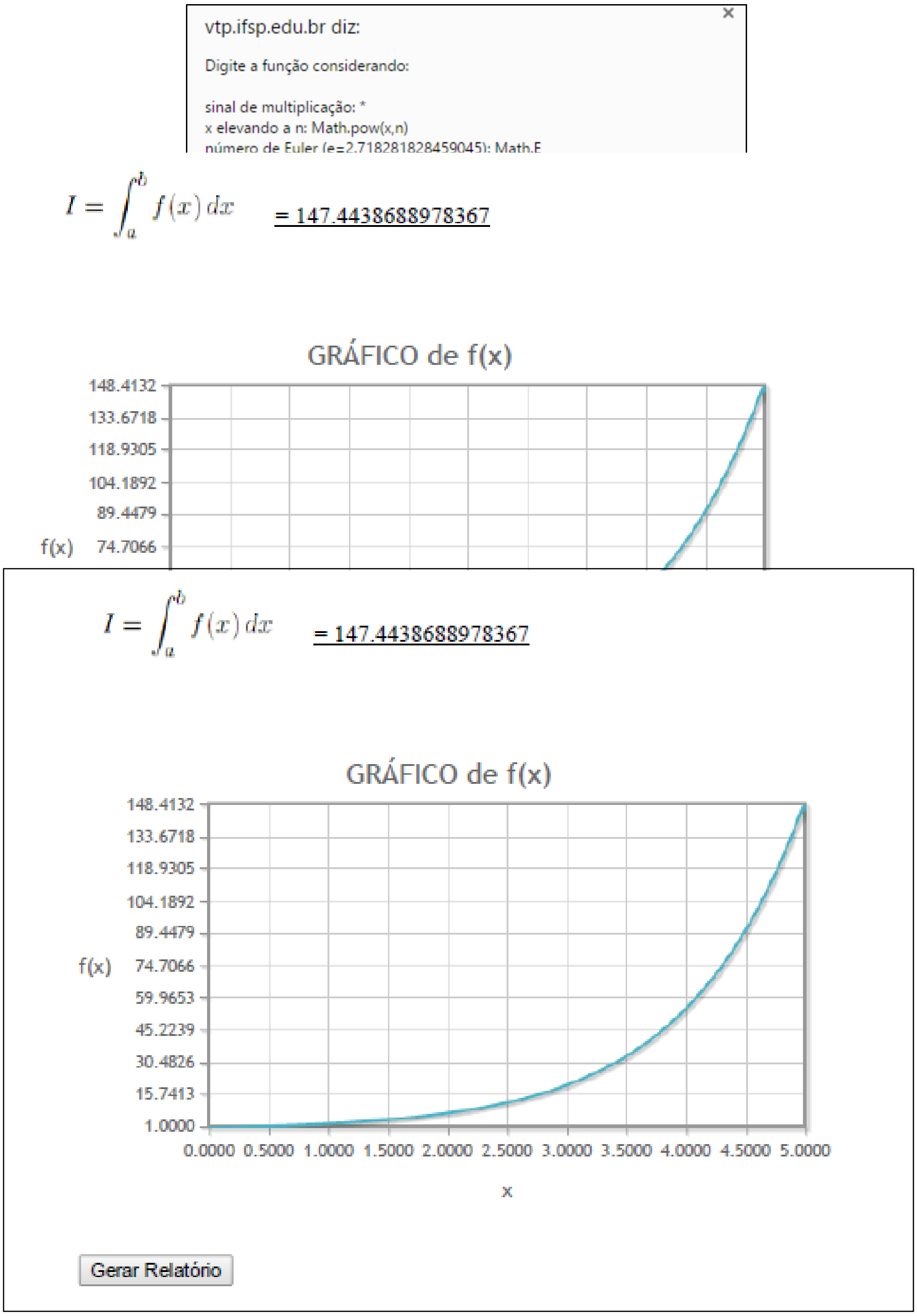


Figura 34: Valor da Integral e a Interpretação gráfica – Resultado Final.

Por fim, como principal objetivo para a criação da página, é possível gerar o relatório, clicando no botão “Gerar Relatório”, depois do resultado final, conforme a Fig. 34 anterior. O relatório é em PDF e contém a resolução detalhada, além de uma breve explicação teórica e o algoritmo principal para o estudo de cálculo numérico. A figura a seguir, Fig. 35, mostra o relatório para o exemplo da Fig. 32.

NEV: Núcleo de Engenharia Virtual
Título: Integração de função - Regra dos Trapézios
Data: 08/09/2016
Autor: Isabela Cassia Dominical Parra
Orientador: Prof. Gustavo Cabrelli Nirschl
Tipo: Iniciação Científica
Curso: Engenharia Civil

$$I = \int_a^b f(x)\, dx$$

Função f(x): Math.pow(Math.E,x)

Extremo "a" do intervalo [a,b]: 0
Extremo "b" do intervalo [a,b]: 5
Número de intervalos para a integração: 100

Para resolver a integral pela Regra dos Trapézios a partir de uma Função em um intervalo [a,b], deve-se determinar o números de intervalos da integração (quanto maior o número, mais preciso é o valor da integral).
Com o número de intervalos, é preciso encontrar todos os "x", e posteriormente encontrar todos "f(x)".
Com todos os pontos (x,f(x)) determinados basta resolver a fórmula a seguir.
Obs:A resolução da integral por meio do algoritmo é realizada de dois em dois pontos, sempre somando com o valor anterior.

$$T(f,h) = \frac{h}{2}\left[f(x_0) + 2\ \left(f(x_1) + \cdots + f(x_{n-1})\right) + f(x_n)\right]$$

Segue a resolução pela Regra dos Trapézios para encontrar alguns pontos, e em seguida a tabela com todos os pontos formados:

h=(b-a)/n=(5.0000-(0.000))/100=0.05000
x[0]=a=0.000
x[1]=x[0]+h=0.0000+0.0500=0.0500
x[2]=x[1]+h=0.0500+0.0500=0.1000
x[3]=x[2]+h=0.1000+0.0500=0.1500
x[4]=x[3]+h=0.1500+0.0500=0.2000
x[5]=x[4]+h=0.2000+0.0500=0.2500
E assim por diante para encontrar os outros 95 pontos...

Para encontrar os f(x) basta substituir os x encontrados na função:
f(x[0])=1.0000
f(x[1])=1.0513
f(x[2])=1.1052
f(x[3])=1.1618
f(x[4])=1.2214
f(x[5])=1.2840
E assim por diante para encontrar os outros 95 f(x)...

	x	f(x)
0	0.000	1.000

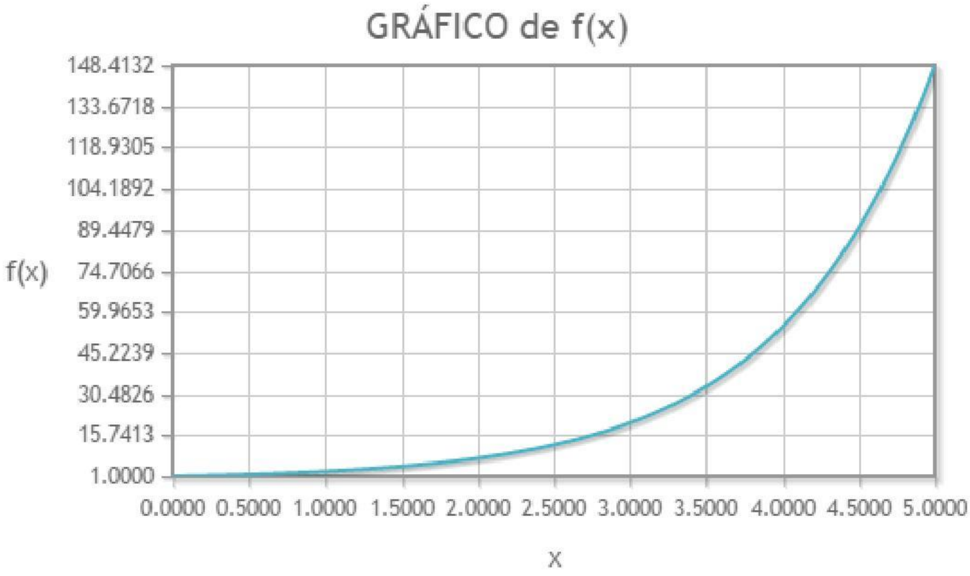
1	0.05000	1.0513
2	0.1000	1.1052
3	0.1500	1.1618
4	0.2000	1.2214
5	0.2500	1.2840
6	0.3000	1.3499
7	0.3500	1.4191
8	0.4000	1.4918
9	0.4500	1.5683
10	0.5000	1.6487
11	0.5500	1.7333
12	0.6000	1.8221
13	0.6500	1.9155
14	0.7000	2.0138
15	0.7500	2.1170
16	0.8000	2.2255
17	0.8500	2.3396
18	0.9000	2.4596
19	0.9500	2.5857
20	1.0000	2.7183
21	1.0500	2.8577
22	1.1000	3.0042
23	1.1500	3.1582
24	1.2000	3.3201
25	1.2500	3.4903
26	1.3000	3.6693
27	1.3500	3.8574
28	1.4000	4.0552
29	1.4500	4.2631
30	1.5000	4.4817
31	1.5500	4.7115
32	1.6000	4.9530
33	1.6500	5.2070
34	1.7000	5.4739
35	1.7500	5.7546
36	1.8000	6.0496
37	1.8500	6.3598

38	1.9000	6.6859
39	1.9500	7.0287
40	2.0000	7.3891
41	2.0500	7.7679
42	2.1000	8.1662
43	2.1500	8.5849
44	2.2000	9.0250
45	2.2500	9.4877
46	2.3000	9.9742
47	2.3500	10.4856
48	2.4000	11.0232
49	2.4500	11.5883
50	2.5000	12.1825
51	2.5500	12.8071
52	2.6000	13.4637
53	2.6500	14.1540
54	2.7000	14.8797
55	2.7500	15.6426
56	2.8000	16.4446
57	2.8500	17.2878
58	2.9000	18.1741
59	2.9500	19.1060
60	3.0000	20.0855
61	3.0500	21.1153
62	3.1000	22.1980
63	3.1500	23.3361
64	3.2000	24.5325
65	3.2500	25.7903
66	3.3000	27.1126
67	3.3500	28.5027
68	3.4000	29.9641
69	3.4500	31.5004
70	3.5000	33.1155
71	3.5500	34.8133
72	3.6000	36.5982
73	3.6500	38.4747
74	3.7000	40.4473

75	3.7500	42.5211
76	3.8000	44.7012
77	3.8500	46.9931
78	3.9000	49.4024
79	3.9500	51.9354
80	4.0000	54.5982
81	4.0500	57.3975
82	4.1000	60.3403
83	4.1500	63.4340
84	4.2000	66.6863
85	4.2500	70.1054
86	4.3000	73.6998
87	4.3500	77.4785
88	4.4000	81.4509
89	4.4500	85.6269
90	4.5000	90.0171
91	4.5500	94.6324
92	4.6000	99.4843
93	4.6500	104.5850
94	4.7000	109.9472
95	4.7500	115.5843
96	4.8000	121.5104
97	4.8500	127.7404
98	4.9000	134.2898
99	4.9500	141.1750
100	5.0000	148.4132

T(f,h)=0.05000/2*[1.0000+2*(1.0513+...+141.1750)+148.4132] = 147.4

Segue abaixo gráfico da função:



A seguir apresenta o algoritmo principal para encontrar o valor da Integral:

```
Dado f(x), a, b e n:
//cálculo da integral
x=[]; //criar o vetor
inte=0;
h=(b-a)/n;
x[0]=a;
Para i de 1 até n
x[i]=x[i-1]+h;
Fim do para
Para i de 1 até n
x[i]=x[i-1]+h;
vf(f,x[i-1]);
fxim=y;
vf(f,x[i]);
fxi=y;
inte=inte+((h/2)*(fxim+fxi));
Fim do para
```

REFERÊNCIAS BIBLIOGRÁFICAS

RUGGIERO, M.A.G; LOPES, V.L.R. Cálculo numérico: aspectos teóricos e computacionais. 2.ed. São Paulo: Makron Books, 1996.

SWOKOWSKI. Cálculo com Geometria Analítica. São Paulo. Makron Books, 1995.

QUADROS, Régis S. de; BORTOLI, Álvaro L. de. Fundamentos de Cálculo Numérico para Engenheiros. Porto Alegre, 2009. Disponível em < http://www.inf.ufrgs.br/~rlflupchinski/files/20112/NUMERICO/numerico-bortoli.pdf >. Acesso em outubro de 2014.

Figura 35: Relatório gerado em PDF para o exemplo apresentado na Fig. 32.

3.7.2. Pela Tabela de Pontos

A página para fazer Integração pela Regra do Trapézio a partir de uma tabela de pontos apresenta-se como a Fig.36 a seguir.

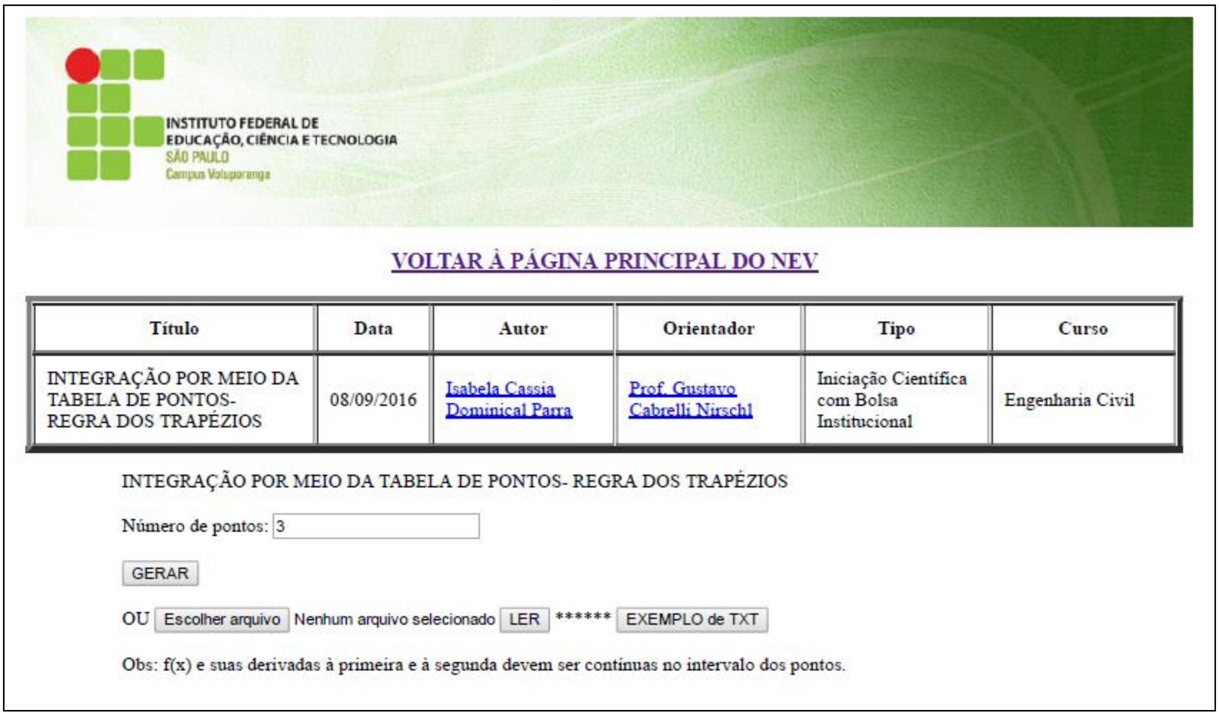


Figura 36: Página para Integração pela Regra do Trapézioa partir de uma tabela de pontos.

Para iniciar os cálculos, tem-se duas opções: ou gerar a tabela de dados na própria página, a partir do número de pontos escolhido, clicando no botão “GERAR”, e digitar os valores (Fig. 37 a seguir) de cada coordenada dos pontos, ou criar um arquivo TXT, seguindo o padrão apresentado ao se clicar no botão “EXEMPLO de TXT” (Fig. 38 a seguir).

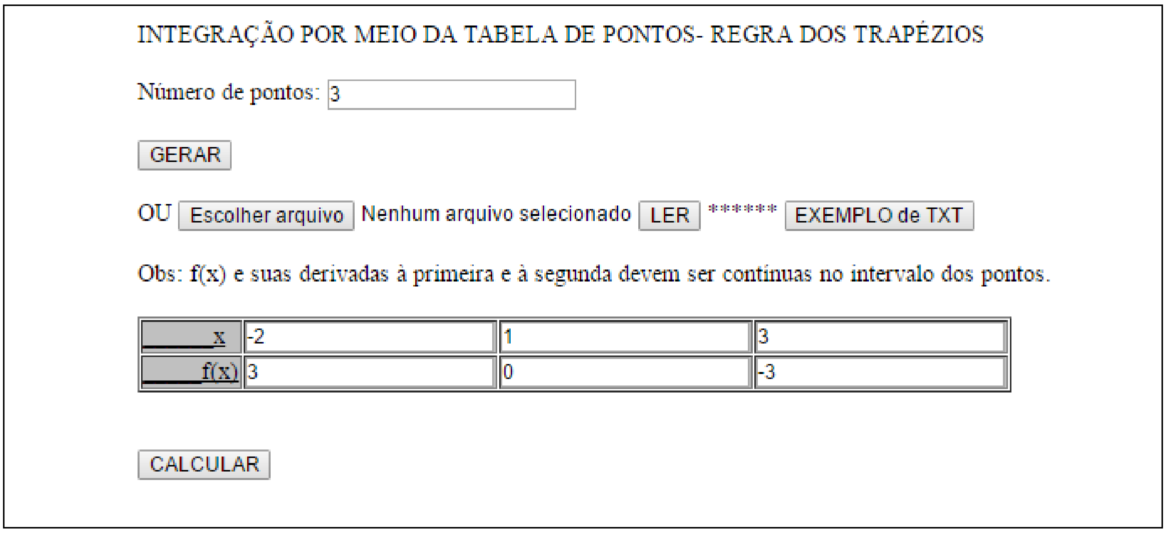


Figura 37: Tabela de pontos digitada na página.

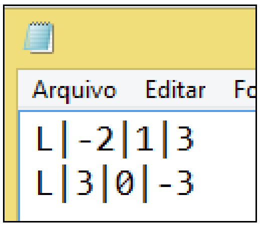


Figura 38: Exemplo de TXT

Depois de ler ou digitar a tabela de pontos, clica-se no botão "CALCULAR", mostrado na Fig. 37 anterior, e irá aparecer o resultado da integral e o gráfico dos pontos gerado. A Fig. 39 a seguir apresenta um exemplo da tabela com 3 pontos da Fig. 37.

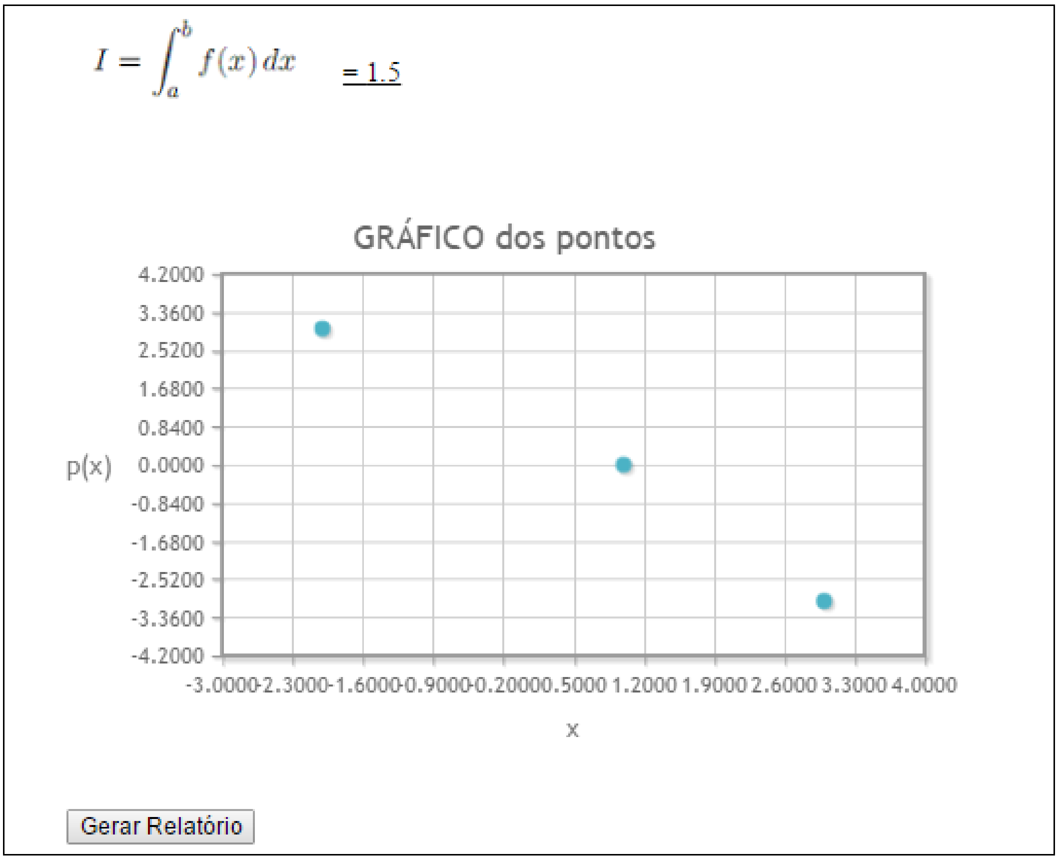


Figura 39: Valor da Integral da função e a Interpretação gráfica – Resultado Final.

Por fim, como principal objetivo para a criação da página, é possível gerar o relatório, clicando no botão “Gerar Relatório”, depois do resultado final, conforme a Fig. 39 anterior. O relatório é em PDF e contém a resolução detalhada, além de uma breve explicação teórica e o algoritmo principal para o estudo de cálculo numérico. A figura a seguir, Fig. 40, mostra o relatório para os pontos exemplificados na Fig. 37.

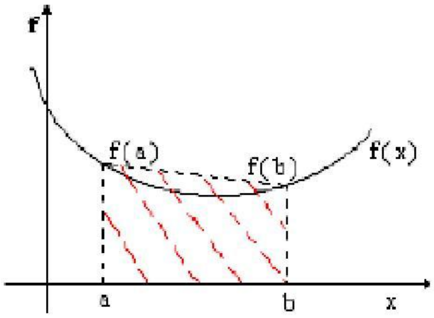
$$I = \int_a^b f(x)\,dx$$

Número de pontos=3

Tabela de Pontos:

x	f(x)
-2.0000	3.0000
1.000	0.000
3.0000	-3.0000

Para resolver a integral pela Regra dos Trapézios utiliza-se a área do trápezio abaixo da curva, delimitadas pelo intervalo [a,b], como mostrado a seguir:



Quanto mais dividir a área em trapézios menores mais aproximada é a integral de f(x).
O cálculo aproximado da integral de f(x), com "n" intervalos, de amplitude "h" diferente, é calculado de dois em dois pontos, somando sempre o valor anteiror.
T(f,h)= Tanterior + (h/2*(f(x[i-1])+f(x[i])))
Obs.: Nos casos, em que os "h" forem iguais, pode-se utilizar a fórmula a seguir:

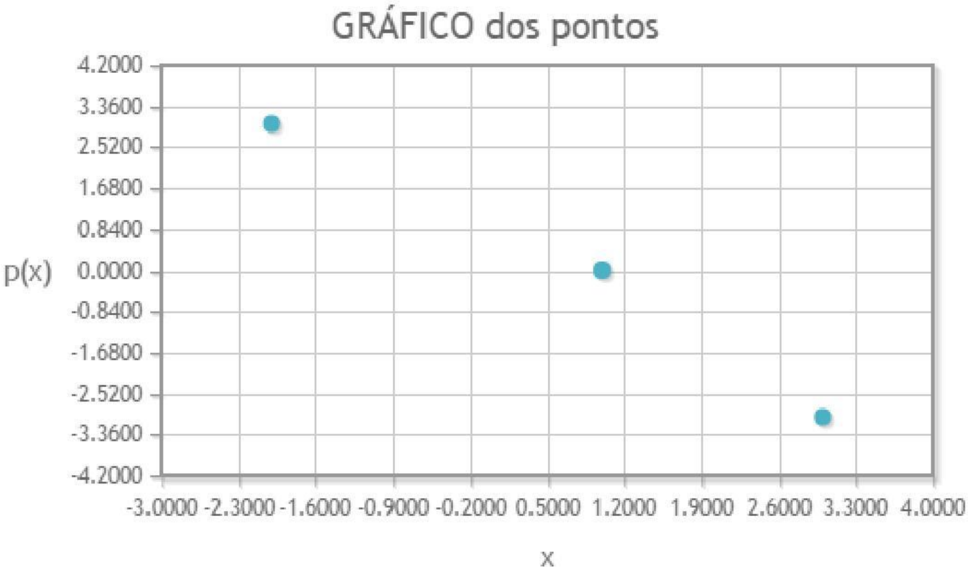
$$T(f,h) = \frac{h}{2}\left[f(x_0) + 2\ (f(x_1) + \cdots + f(x_{n-1})) + f(x_n)\right]$$

Abaixo segue a resolução da Regra dos Trapézios:

inte=0
h[1]=(x[1]-x[0])=(1.000-(-2.0000))=3.0000
inte=inte+((h[1]/2)*(f(x[0])+f(x[1])))=0.000+(3.0000/2)*(3.0000+(0.000)))=4.5000
h[2]=(x[2]-x[1])=(3.0000-(1.000))=2.0000

inte=inte+((h[2]/2)*(f(x[1])+f(x[2])))=4.5000+(2.0000/2)*(0.000+(-3.0000)))=1.5000

Segue abaixo gráfico da função dada a partir dos pontos:



A seguir apresenta o algoritmo principal para o cálculo de integral de uma função, pela Regra dos Trapézios, a partir de uma tabela de pontos:

```
//Tendo n (número de intervalos, constantes) e os extremos a e b da integral:
inte=0;
h=(b-a)/n;
x(0)=a;
para i=1 até n {
x(i)=x(i-1)+h
inte=inte+((h/2)*(f(x(i-1))+f(x(i)))
} fim do para i
//No caso de intervalos diferentes (numa tabela de n+1 pontos):
inte=0;
para i=1 até n {
h(i)=x(i)-x(i-1);
inte=inte+((h(i)/2)*(f(x(i-1))+f(x(i)))
} fim do para i
```

REFERÊNCIAS BIBLIOGRÁFICAS

RUGGIERO, M.A.G; LOPES, V.L.R. Cálculo numérico: aspectos teóricos e computacionais. 2.ed. São Paulo: Makron Books, 1996.

SWOKOWSKI. Cálculo com Geometria Analítica. São Paulo. Makron Books, 1995.

QUADROS, Régis S. de; BORTOLI, Álvaro L. de. Fundamentos de Cálculo Numérico para Engenheiros. Porto Alegre, 2009. Disponível em < http://www.inf.ufrgs.br/~rlflupchinski/files/20112/NUMERICO/numerico-bortoli.pdf >. Acesso em outubro de 2014.

Figura 40: Relatório gerado em PDF para o exemplo apresentado na Fig. 37.

3.8. Integração pela Fórmula de Simpson

3.8.1. Pela Função

A página para fazer Integração pela Fórmula de Simpson a partir de uma função apresenta-se como a Fig.41 a seguir.

Figura 41: Página para Integração pela Fórmula de Simpson a partir de uma função.

Para iniciar os cálculos é preciso escrever a função, por meio do botão "AJUDA", é possível visualizar as principais funções matemáticas na linguagem Javascript, como esta deverá ser escrita (Fig. 42 abaixo). O botão "TESTAR f em x=" é para testar se a função foi digitada corretamente na linguagem Javascript. Depois de digitado a função, escolhido o intervalo de integração e o número de intervalos, clica-se no botão "CALCULAR", mostrado na Fig. 41 anterior, e irá aparecer o resultado da integral e o gráfico gerado a partir da função. A Fig. 43 a seguir apresenta um exemplo da Fig. 41.

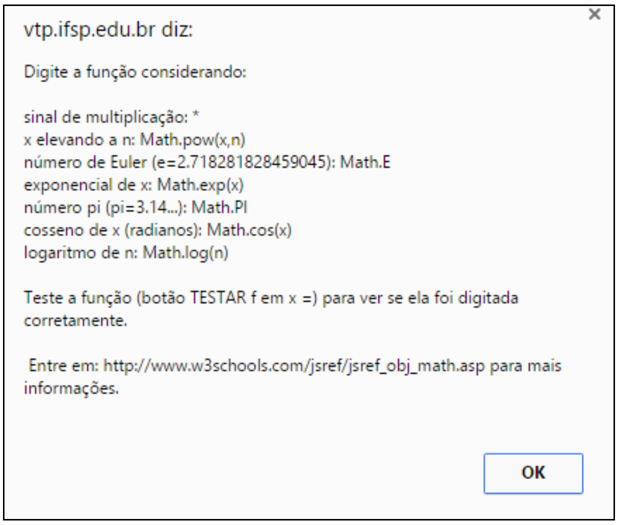


Figura 42: Funções matemáticas na linguagem Javascript.

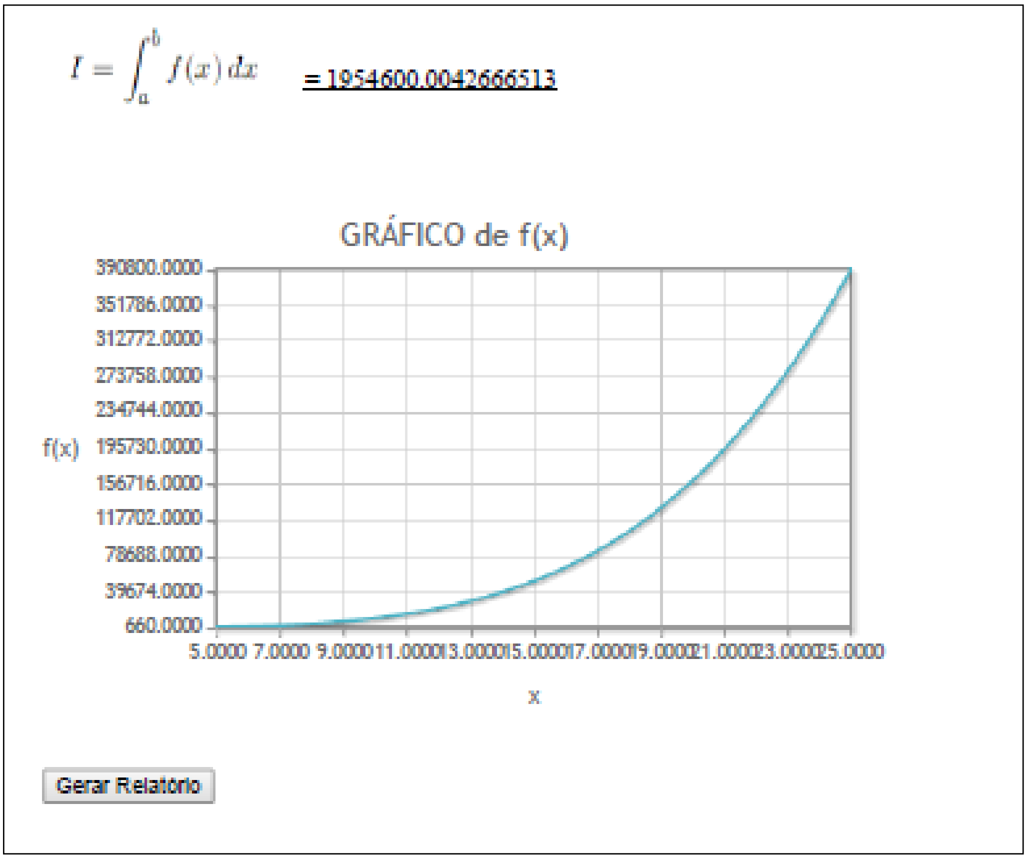


Figura 43: Valor da Integral e a Interpretação gráfica – Resultado Final.

Por fim, como principal objetivo para a criação da página, é possível gerar o relatório, clicando no botão “Gerar Relatório”, depois do resultado final, conforme a Fig. 43 anterior. O relatório é em PDF e contém a resolução detalhada, além de uma breve explicação teórica e o algoritmo principal para o estudo de cálculo numérico. A figura a seguir, Fig. 44, mostra o relatório para o exemplo da Fig. 41.

NEV: Núcleo de Engenharia Virtual
Título: Integração de função - Fórmula de Simpson
Data: 21/10/2016
Autor: Isabela Cassia Dominical Parra
Orientador: Prof. Gustavo Cabrelli Nirschl
Tipo: Iniciação Científica
Curso: Engenharia Civil

$$I = \int_a^b f(x)\,dx$$

Função f(x): Math.pow(x,4)+7*x

Extremo "a" do intervalo [a,b]: 5
Extremo "b" do intervalo [a,b]: 25
Número de intervalos para a integração: 100

Para resolver a integral pela Fórmula de Simpson a partir de uma Função em um intervalo [a,b], deve-se determinar o números de intervalos da integração (quanto maior o número, mais preciso é o valor da integral), este deverá ser par.
Com o número de intervalos, é preciso encontrar todos os "x", e posteriormente encontrar todos "f(x)".
Com todos os pontos (x,f(x)) determinados basta resolver a fórmula a seguir.

$$S(f,h) = \frac{h}{3}\left\{\begin{matrix} f(x_0) + 4\left[f(x_1) + f(x_3) + f(x_5) + \ldots + f(x_{2n-1})\right] \\ + 2\left[f(x_2) + f(x_4) + f(x_6) + \ldots + f(x_{2n-2})\right] + f(x_{2n}) \end{matrix}\right\}$$

Segue a resolução pela Fórmula de Simpson para encontrar alguns pontos, e em seguida a tabela com todos os pontos formados:

h=(b-a)/2*n=(25.0000-(5.0000))/100=0.2000
x[0]=a=5.0000
x[1]=x[0]+h=5.0000+0.2000=5.2000
x[2]=x[1]+h=5.2000+0.2000=5.4000
x[3]=x[2]+h=5.4000+0.2000=5.6000
x[4]=x[3]+h=5.6000+0.2000=5.8000
x[5]=x[4]+h=5.8000+0.2000=6.0000
E assim por diante para encontrar os outros 95 pontos...

Para encontrar os f(x) basta substituir os x encontrados na função:
f(x[0])=660.0000
f(x[1])=767.5616
f(x[2])=888.1056
f(x[3])=1022.6496
f(x[4])=1172.2496
f(x[5])=1338.0000
E assim por diante para encontrar os outros 95 f(x)...

	x	f(x)
0	5.0000	660.0000

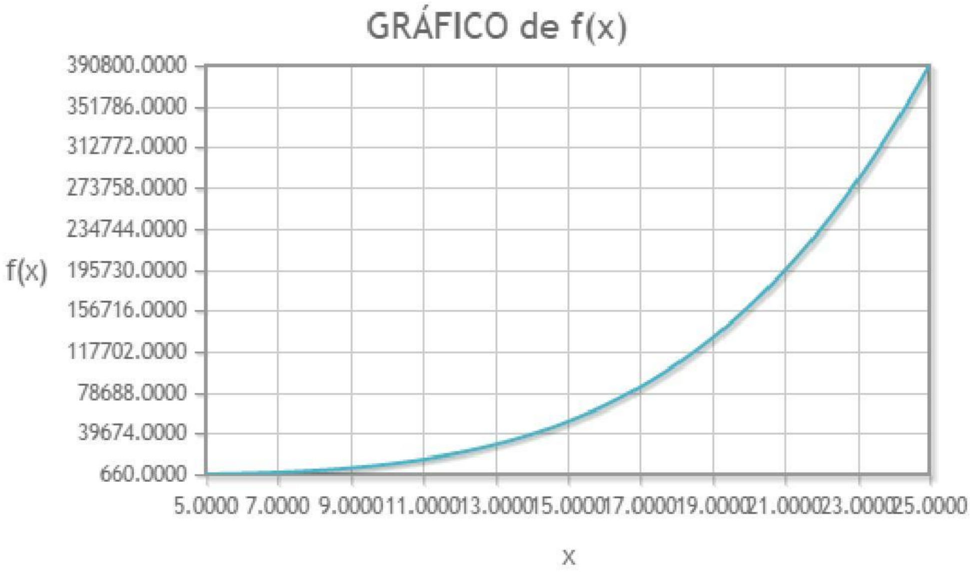
1	5.2000	767.5616
2	5.4000	888.1056
3	5.6000	1022.6496
4	5.8000	1172.2496
5	6.0000	1338.0000
6	6.2000	1521.0336
7	6.4000	1722.5216
8	6.6000	1943.6736
9	6.8000	2185.7376
10	7.0000	2450.0000
11	7.2000	2737.7856
12	7.4000	3050.4576
13	7.6000	3389.4176
14	7.8000	3756.1056
15	8.0000	4152.0000
16	8.2000	4578.6176
17	8.4000	5037.5136
18	8.6000	5530.2816
19	8.8000	6058.5536
20	9.0000	6624.0000
21	9.2000	7228.3296
22	9.4000	7873.2896
23	9.6000	8560.6656
24	9.8000	9292.2816
25	10.0000	10070.0000
26	10.2000	10895.7216
27	10.4000	11771.3856
28	10.6000	12698.9696
29	10.8000	13680.4896
30	11.0000	14718.0000
31	11.2000	15813.5936
32	11.4000	16969.4016
33	11.6000	18187.5936
34	11.8000	19470.3776
35	12.0000	20820.0000
36	12.2000	22238.7456
37	12.4000	23728.9376

38	12.6000	25292.9376
39	12.8000	26933.1456
40	13.0000	28652.0000
41	13.2000	30451.9776
42	13.4000	32335.5936
43	13.6000	34305.4016
44	13.8000	36363.9936
45	14.0000	38514.0000
46	14.2000	40758.0896
47	14.4000	43098.9696
48	14.6000	45539.3856
49	14.8000	48082.1216
50	15.0000	50730.0000
51	15.2000	53485.8816
52	15.4000	56352.6656
53	15.6000	59333.2896
54	15.8000	62430.7296
55	16.0000	65648.0000
56	16.2000	68988.1536
57	16.4000	72454.2816
58	16.6000	76049.5136
59	16.8000	79777.0176
60	17.0000	83640.0000
61	17.2000	87641.7056
62	17.4000	91785.4176
63	17.6000	96074.4576
64	17.8000	100512.1856
65	18.0000	105102.0000
66	18.2000	109847.3376
67	18.4000	114751.6736
68	18.6000	119818.5216
69	18.8000	125051.4336
70	19.0000	130454.0000
71	19.2000	136029.8496
72	19.4000	141782.6496
73	19.6000	147716.1056
74	19.8000	153833.9616

75	20.0000	160140.0000
76	20.2000	166638.0416
77	20.4000	173331.9456
78	20.6000	180225.6096
79	20.8000	187322.9696
80	21.0000	194628.0000
81	21.2000	202144.7136
82	21.4000	209877.1616
83	21.6000	217829.4336
84	21.8000	226005.6576
85	22.0000	234410.0000
86	22.2000	243046.6656
87	22.4000	251919.8976
88	22.6000	261033.9776
89	22.8000	270393.2256
90	23.0000	280002.0000
91	23.2000	289864.6976
92	23.4000	299985.7536
93	23.6000	310369.6416
94	23.8000	321020.8736
95	24.0000	331944.0000
96	24.2000	343143.6096
97	24.4000	354624.3296
98	24.6000	366390.8256
99	24.8000	378447.8016
100	25.0000	390800.0000

T(f,h)=0.2000/3*[660.0000+4*(767.5616+...+378447.8016)+2*(888.1056+...+366390.8256)+390800.0000] = 1954600.0043

Segue abaixo gráfico da função:



A seguir apresenta o algoritmo principal para encontrar o valor da Integral:

```
Dado f(x), a, b e m:
n=m/2
//cálculo da integral
x=[]; //criar o vetor
h=(b-a)/(2*n);
x[0]=a;
Para i de 1 até (2*n)
x[i]=x[i-1]+h;
Fim do para
Para i de 1 até n
x[i]=x[i-1]+h;
Fim do para
inte=0;
Para k de 1 até n
vf(f,x[(2*k)-2]);
f1=y;
vf(f,x[(2*k)-1]);
f2=y;
vf(f,x[(2*k)]);
f3=y;
inte=inte+((h/3)*(f1+(4*f2)+f3));
Fim do para
```

REFERÊNCIAS BIBLIOGRÁFICAS

RUGGIERO, M.A.G; LOPES, V.L.R. Cálculo numérico: aspectos teóricos e computacionais. 2.ed. São Paulo: Makron Books, 1996.

SWOKOWSKI. Cálculo com Geometria Analítica. São Paulo. Makron Books, 1995.

QUADROS, Régis S. de; BORTOLI, Álvaro L. de. Fundamentos de Cálculo Numérico para Engenheiros. Porto Alegre, 2009. Disponível em < http://www.inf.ufrgs.br/~rlflupchinski/files/20112/NUMERICO/numerico-bortoli.pdf >. Acesso em outubro de 2014.

Figura 44: Relatório gerado em PDF para o exemplo apresentado na Fig. 41.

3.8.2. Pela Tabela de Pontos

A página para fazer Integração pela Formula de Simpson a partir de uma tabela de pontos apresenta-se como a Fig.45 a seguir.

Figura 45: Página para Integração pela Formula de Simpson a partir de uma tabela de pontos.

Para iniciar os cálculos, tem-se duas opções: ou gerar a tabela de dados na própria página, a partir do número de pontos escolhido, clicando no botão “GERAR”, e digitar os valores (Fig. 46 a seguir) de cada coordenada dos pontos, ou criar um arquivo TXT, seguindo o padrão apresentado ao se clicar no botão “EXEMPLO de TXT” (Fig. 47 a seguir).

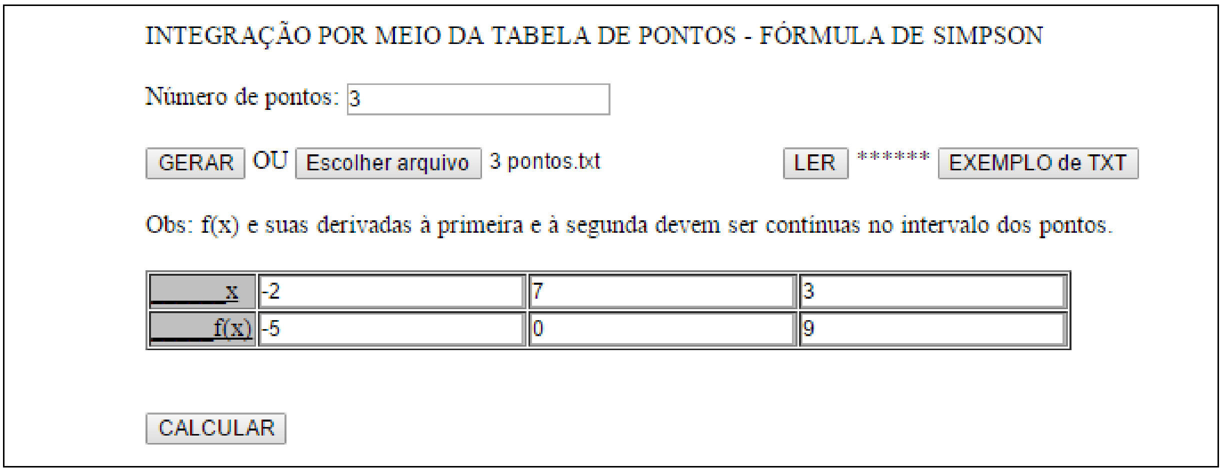


Figura 46: Tabela de pontos digitada na página.

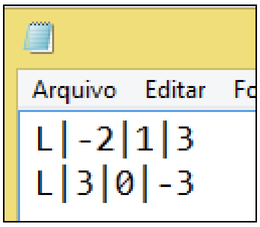


Figura 47: Exemplo de TXT

Depois de ler ou digitar a tabela de pontos, clica-se no botão “CALCULAR”, mostrado na Fig. 46 anterior, e irá aparecer o resultado da integral e o gráfico dos pontos gerado. A Fig. 48 a seguir apresenta um exemplo da tabela com 3 pontos da Fig. 46.

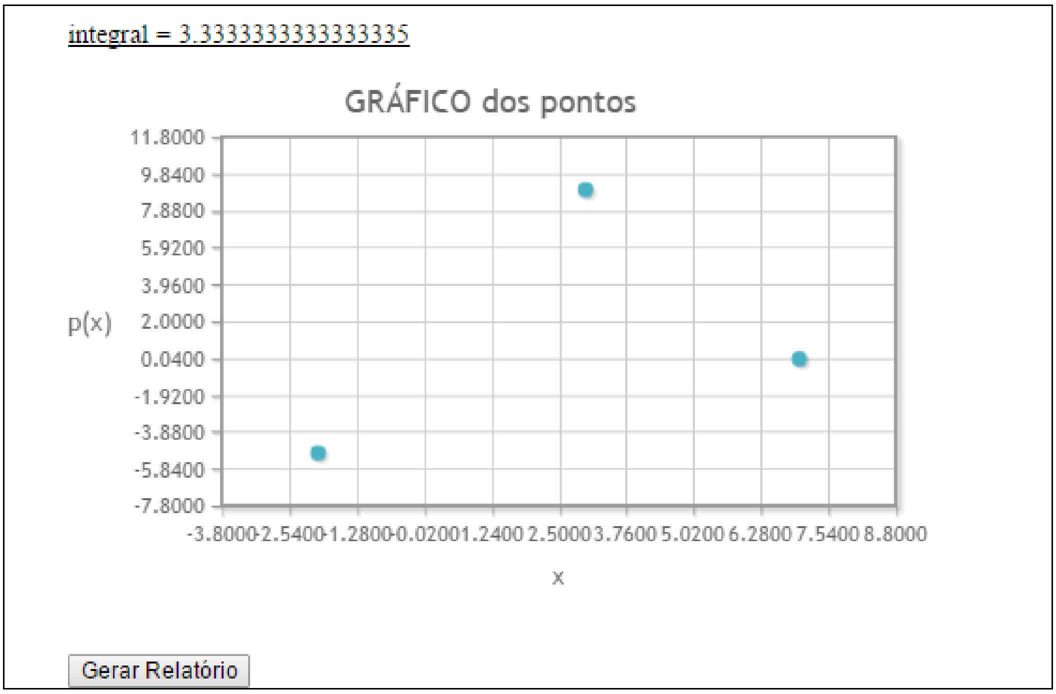


Figura 48: Valor da Integral da função e a Interpretação gráfica – Resultado Final.

Por fim, como principal objetivo para a criação da página, é possível gerar o relatório, clicando no botão “Gerar Relatório”, depois do resultado final, conforme a Fig. 48 anterior. O relatório é em PDF e contém a resolução detalhada, além de uma breve explicação teórica e o algoritmo principal para o estudo de cálculo numérico. A figura a seguir, Fig. 49, mostra o relatório para os pontos exemplificados na Fig. 46.

$$I = \int_a^b f(x)\, dx$$

Número de pontos=3

Tabela de Pontos:

x	f(x)
-2.0000	-5.0000
7.0000	0.000
3.0000	9.0000

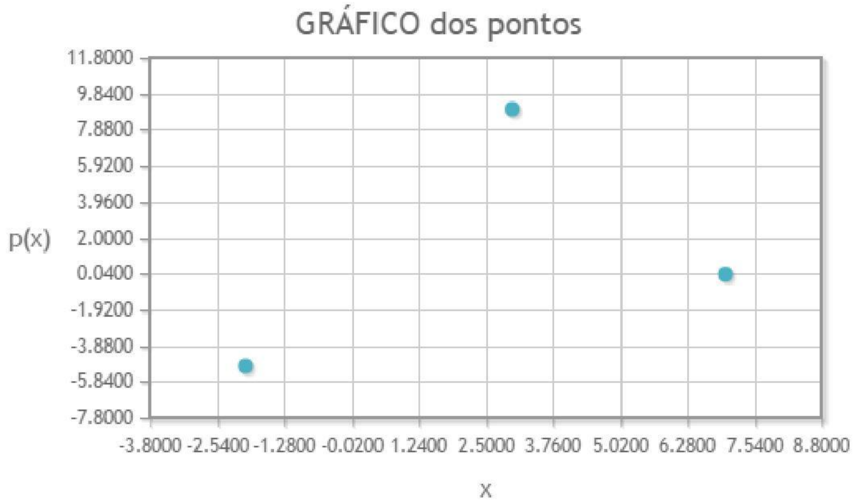
Para resolver a integral pela fórmula de Simpson utiliza-se um polinômio de 2° grau. O polinômio de grau 2 que aproxima a função original f(x) pode ser obtido, por exemplo, por Lagrange.
O cálculo aproximado da integral de f(x), com "2n" intervalos de amplitude "h" diferentes, é calculado de três em três pontos, somando sempre o valor anteiror.
T(f,h)= Tanterior + (h/3*(f(x[2*i-2])+(4*f(x[2*i-1])+f(x[2*i])))
Obs.: Nos casos, em que os "h" forem iguais, pode-se utilizar a fórmula a seguir:

$$S(f,h) = \frac{h}{3}\left\{ \begin{array}{l} f(x_0) + 4\left[f(x_1) + f(x_3) + f(x_5) + \ldots + f(x_{2n-1})\right] \\ + 2\left[f(x_2) + f(x_4) + f(x_6) + \ldots + f(x_{2n-2})\right] + f(x_{2n}) \end{array} \right\}$$

Abaixo segue a resolução da Fórmula de Simpson:

n=(np-1)/2=(3-1)/2=1
inte=0
h[1]=(x[2]-x[0])/2=(3.0000-(-2.0000))/2=2.5000
inte=inte+((h[1]/3)*(f(x[0])+(4*f(x[1]))+f(x[2])))=0.000+((2.5000/3)*(-5.0000+4*(0.000)+(0.0000)))=3.3333

Segue abaixo gráfico da função dada a partir dos pontos:



A seguir apresenta o algoritmo principal para o cálculo de integral de uma função, pela Fórmula de Simpson, a partir de uma tabela de pontos:

```
//No caso de intervalos diferentes (numa tabela de (2n+1) pontos (0 a 2n)):
inte=0;
para k=1 até n
h(k)=(x(2k)-x(2k-2))/2;
inte=inte+((h(i)/3)*(f(x(2k-2))+4f(x(2k-1)+f(x(2k)))
 fim do para k
```

REFERÊNCIAS BIBLIOGRÁFICAS

RUGGIERO, M.A.G; LOPES, V.L.R. Cálculo numérico: aspectos teóricos e computacionais. 2.ed. São Paulo: Makron Books, 1996.

SWOKOWSKI. Cálculo com Geometria Analítica. São Paulo. Makron Books, 1995.

QUADROS, Régis S. de; BORTOLI, Álvaro L. de. Fundamentos de Cálculo Numérico para Engenheiros. Porto Alegre, 2009. Disponível em < http://www.inf.ufrgs.br/~rlflupchinski/files/20112/NUMERICO/numerico-bortoli.pdf >. Acesso em outubro de 2014.

Figura 49: Relatório gerado em PDF para o exemplo apresentado na Fig. 46.

3.9. Derivação Numérica

3.9.1. Pela Função

A página paraDerivação Numérica a partir de uma função apresenta-se como a Fig.50 a seguir.

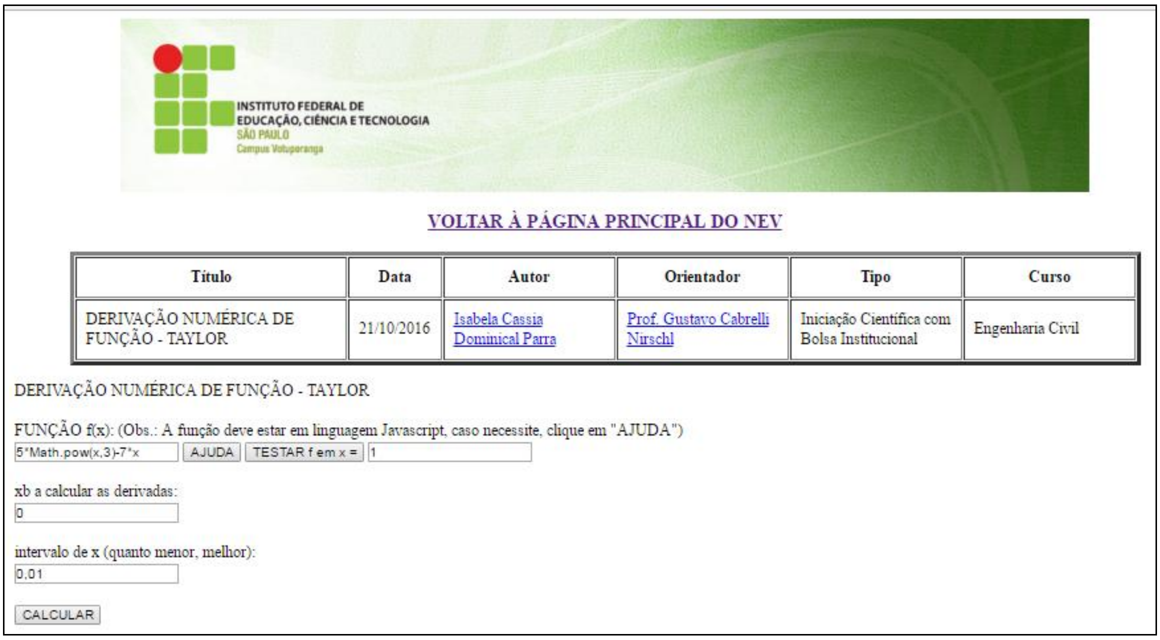


Figura 50: Página para Derivação Numérica a partir de uma função.

Para iniciar os cálculos é preciso escrever a função, por meio do botão “AJUDA”, é possível visualizar as principais funções matemáticas na linguagem Javascript, como esta deverá ser escrita (Fig. 51 abaixo). O botão “TESTAR f em x=” é para testar se a função foi digitada corretamente na linguagem Javascript. Depois de digitado a função, o “xb” e o intervalo entre os x, clica-se no botão “CALCULAR”, mostrado na Fig. 50 anterior, e irá aparecer os resultados das derivadas. A Fig. 52 a seguir apresenta um exemplo da Fig. 50.

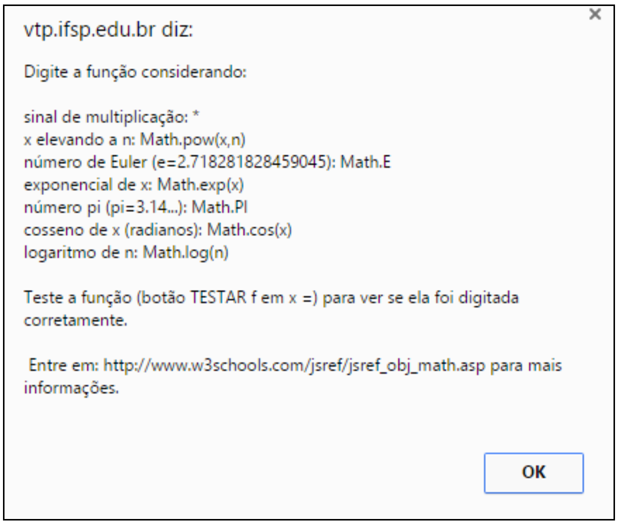


Figura 51: Funções matemáticas na linguagem Javascript.

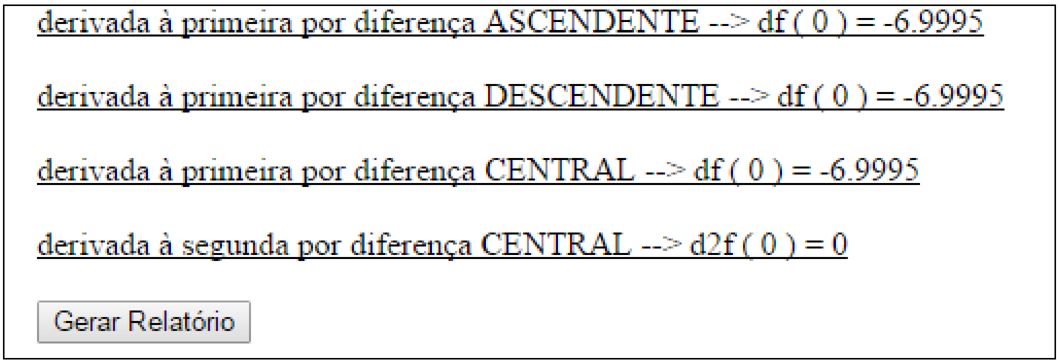


Figura 52: Valores das Derivadas – Resultado Final.

Por fim, como principal objetivo para a criação da página, é possível gerar o relatório, clicando no botão "Gerar Relatório", depois do resultado final, conforme a Fig. 52 anterior. O relatório é em PDF e contém a resolução detalhada, além de uma breve explicação teórica e o algoritmo principal para o estudo de cálculo numérico. A figura a seguir, Fig. 53, mostra o relatório para o exemplo da Fig. 50.

NEV: Núcleo de Engenharia Virtual
Título: Derivação Numérica de Função - Taylor
Data: 21/10/2016
Autor: Isabela Cassia Dominical Parra
Orientador: Prof. Gustavo Cabrelli Nirschl
Tipo: Iniciação Científica
Curso: Engenharia Civil

Função f(x): 5*Math.pow(x,3)-7*x

xb a calcular as derivadas: 0
Intervalo de x: 0.01

A derivada em um ponto x0 de uma função y=f(x) representa a taxa de variação instantânea de y em relação a x neste ponto x0 e pode ser denotada como f'(x0)
Graficamente, a derivada é interpretada como o coeficiente angular da reta tangente ao gráfico de f(x) em um ponto x0.
Com uma variação em torno de x, são obtidas 3 possibilidades para o cálculo numérico aproximado do valor da derivada à primeira da função num ponto x0 (chamado Método das Diferenças Finitas).
Abaixo estão as 3 possibilidades (Ascendente, Descendente, Central):

*****Diferença Ascendente ou Avançada (1ª Ordem)**

f '[x(i)]=(f[x(i+1)]-f[x(i)])/h

Abaixo segue a resolução para o exemplo:

f '[0]=(f[0+0.01]-f[0])/h=-0.07000-(0.000)/0.01=-6.9995

*****Diferença Descendentes ou Atrasada (1ª Ordem)**

f '[x(i)]=(f[x(i)]-f[x(i-1)])/h

Abaixo segue a resolução:

f '[0]=(f[0]-f[0-0.01])/h=0.000-(0.07000)/0.01=-7.000

*****Diferença Central ou Centrada (2ª Ordem)**

f '[x(i)]=(f[x(i+1)]-f[x(i-1)])/2*h

Abaixo segue a resolução:

f '[0]=(f[0+0.01]-f[0-0.01])/2*h=-0.07000-(0.07000)/0.02=-6.9995

Além das derivadas à primeira, segue uma possibilidade de cálculo de derivada à segunda, por DIFERENÇA CENTRAL:

*****Diferença Central**

f ''[x(i)]=(f[x(i+1)]-2*f[x(i)]+f[x(i-1)])/h^2

Abaixo segue a resolução:

f ''[0]=(f[0+0.01]-2*f[0]+f[0-0.01])/h^2=-0.07000-(2*0.000)+(0.07000)/0.0001=-0.07000

A seguir apresenta-se o algoritmo principal para encontrar o valor das derivadas:

1: Dado f(x), xb, e h:
2: //cálculo da derivada 1ª em xb, ascendente
3: f 'asc=(f(xb+h)-f(xb))/h
4: //cálculo da derivada 1ª em xb, descendente
5: f 'des=(f(xb)-f(xb-h))/h
6: //cálculo da derivada 1ª em xb, central
7: f 'cen1=(f(xb+h)-f(xb-h))/2*h
8: //cálculo da derivada 2ª em xb, central
9: f ''cen2=(f(xb+h)-(2*f(xb))+f(xb-h))/Math.pow(h,2)

REFERÊNCIAS BIBLIOGRÁFICAS

RUGGIERO, M.A.G; LOPES, V.L.R. Cálculo numérico: aspectos teóricos e computacionais. 2.ed. São Paulo: Makron Books, 1996.

SWOKOWSKI. Cálculo com Geometria Analítica. São Paulo. Makron Books, 1995.

QUADROS, Régis S. de; BORTOLI, Álvaro L. de. Fundamentos de Cálculo Numérico para Engenheiros. Porto Alegre, 2009. Disponível em < http://www.inf.ufrgs.br/~rlflupchinski/files/20112/NUMERICO/numerico-bortoli.pdf >. Acesso em outubro de 2014.

Figura 53: Relatório gerado em PDF para o exemplo apresentado na Fig. 50.

3.9.2. Pela Tabela de Pontos

A página para Derivação Numérica a partir de uma tabela de pontos apresenta-se como a Fig.54 a seguir.

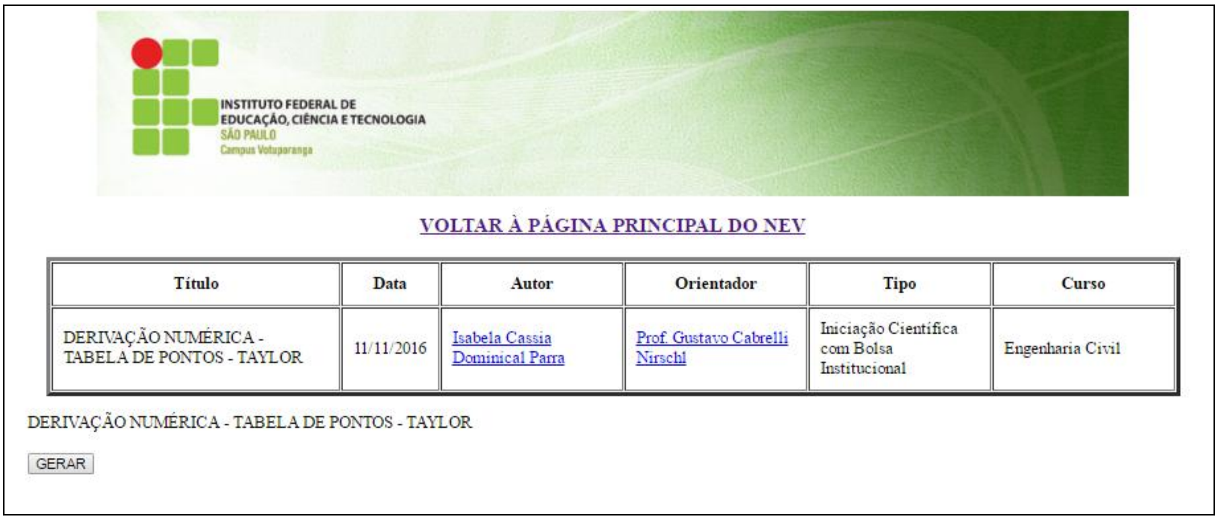


Figura 54: Página para Derivação Numérica a partir de uma tabela de pontos.

Para iniciar os cálculos, clica-se no botão “GERAR”, e digite os valores (Fig. 55 a seguir) de cada coordenada dos pontos.

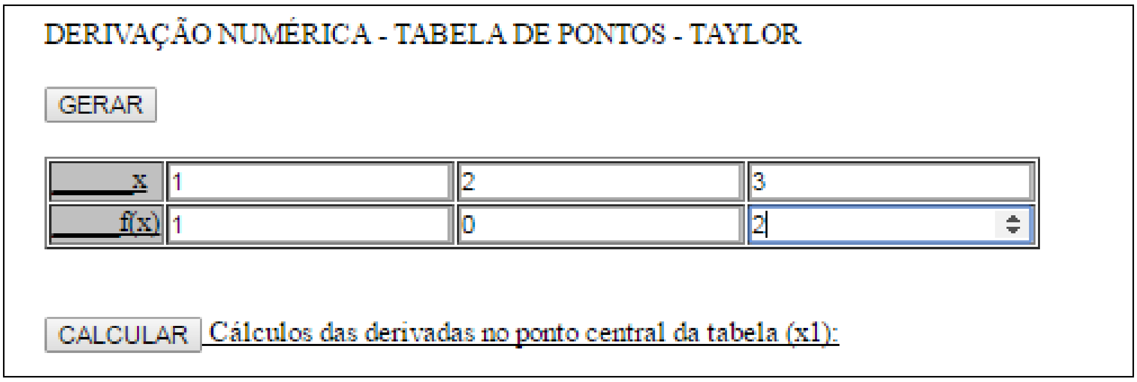


Figura 55: Tabela de pontos digitada na página.

Depois de digitar a tabela de pontos, clica-se no botão “CALCULAR”, mostrado na Fig. 55 anterior, e irá aparecer os resultados das derivadas. A Fig. 56 a seguir apresenta um exemplo da tabela da Fig. 55.

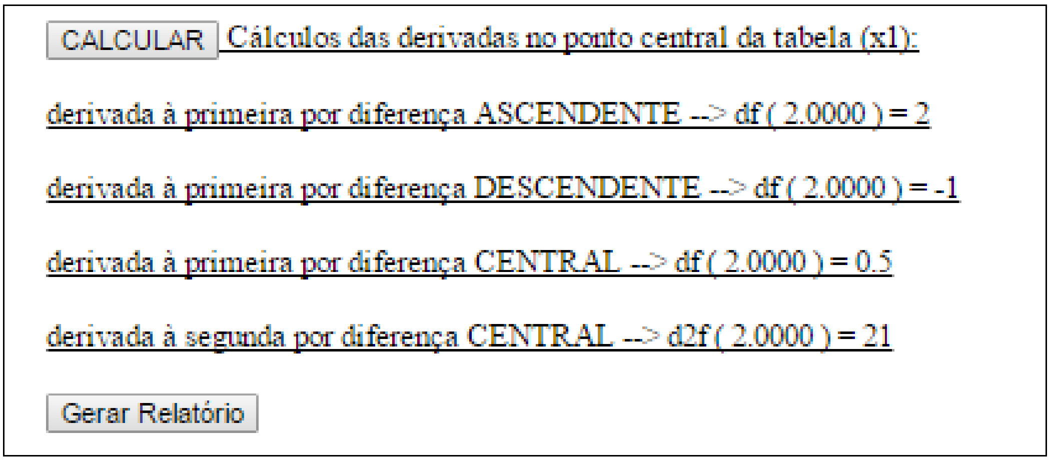


Figura 56: Valores das Derivadas – Resultado Final.

Por fim, como principal objetivo para a criação da página, é possível gerar o relatório, clicando no botão “Gerar Relatório”, depois do resultado final, conforme a Fig. 56 anterior. O relatório é em PDF e contém a resolução detalhada, além de uma breve explicação teórica e o algoritmo principal para o estudo de cálculo numérico. A figura a seguir, Fig. 57, mostra o relatório para os pontos exemplificados na Fig. 55.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Votuporanga

NEV: Núcleo de Engenharia Virtual
Título: Derivação numérica Tabela de pontos - Taylor
Data: 11/11/2016
Autor: Isabela Cassia Dominical Parra
Orientador: Prof. Gustavo Cabrelli Nirschl
Tipo: Iniciação Científica
Curso: Engenharia Civil

Tabela de Pontos:

x	f(x)
1.000	1.000
2.0000	0.000
3.0000	2.0000

A tabela deve ter três pontos, sendo o intervalo de x constante (h = x2-x1 = x1-x0). As derivadas sempre serão calculadas no ponto central da tabela (x1).
Segundo o chamado Método das Diferenças Finitas, existem 3 possibilidades para o cálculo numérico aproximado do valor da derivada no ponto x1:
Ascendente, Descendente e Central.

*****Derivada à primeira por Diferença Ascendente ou Avançada (1ª Ordem)**

f '[x1]=(f[x2]-f[x1])/h

f '[x1]=(f[x2]-f[x1])/h=(2.0000-(0.000))/1.000=2.0000

*****Derivada à primeira por Diferença Descendente ou Atrasada (1ª Ordem)**

f '[x1]=(f[x1]-f[x0])/h

f '[x1]=(f[x1]-f[x0])/h=(0.000-(1.000))/1.000=-1.000

*****Derivada à primeira por Diferença Central ou Centrada (2ª Ordem)**

f '[x1]=(f[x2]-f[x0])/2*h

f '[x1]=(f[x2]-f[x0])/2*h=(2.0000-(1.000))/1.000=0.5000

Além das derivadas à primeira, segue uma possibilidade de cálculo de derivada à segunda, por DIFERENÇA CENTRAL:

*****Derivada à segunda por Diferença Central:**

f '[x1]=(f[x2]-2*f[x1]+f[x0])/h^2

f '[x1]=(f[x2]-2*f[x1]+f[x0])/h^2=(2.0000-2*(0.000)+(1.000))/1=21.0000

A seguir, apresenta-se o algoritmo principal para encontrar os valores das derivadas:

```
Dado x0, x1 e x2 e f(x0), f(x1) e f(x2):
//cálculo da derivada à 1ª em x1, ascendente
f'asc=(f(x2)-f(x1))/h
//cálculo da derivada à 1ª em x1, descendente
f'des=(f(x1)-f(x0))/h
//cálculo da derivada à 1ª em x1, central
f'cen1=(f(x2)-f(x0))/2*h
//cálculo da derivada à 2ª em x1, central
f'cen2=(f(x2)-(2*f(x1))+f(x0))/Math.pow(h,2)
```

REFERÊNCIAS BIBLIOGRÁFICAS

RUGGIERO, M.A.G; LOPES, V.L.R. Cálculo numérico: aspectos teóricos e computacionais. 2.ed. São Paulo: Makron Books, 1996.

SWOKOWSKI. Cálculo com Geometria Analítica. São Paulo. Makron Books, 1995.

QUADROS, Régis S. de; BORTOLI, Álvaro L. de. Fundamentos de Cálculo Numérico para Engenheiros. Porto Alegre, 2009. Disponível em < http://www.inf.ufrgs.br/~rlflupchinski/files/20112/NUMERICO/numerico-bortoli.pdf >. Acesso em outubro de 2014.

Figura 57: Relatório gerado em PDF para o exemplo apresentado na Fig. 55.

3.10. Aproximação Polinomial

A página para Aproximação Polinomialapresenta-se como a Fig.58 a seguir.

Figura 58: Página para Aproximação Polinomial.

Para iniciar os cálculos, tem-se duas opções: ou gerar a tabela de dados na própria página, a partir do número de pontos escolhido, clicando no botão “GERAR”, e digitar os valores (Fig. 59 a seguir) de cada coordenada dos pontos, ou criar um arquivo TXT, seguindo o padrão apresentado ao se clicar no botão “EXEMPLO de TXT” (Fig. 60 a seguir).

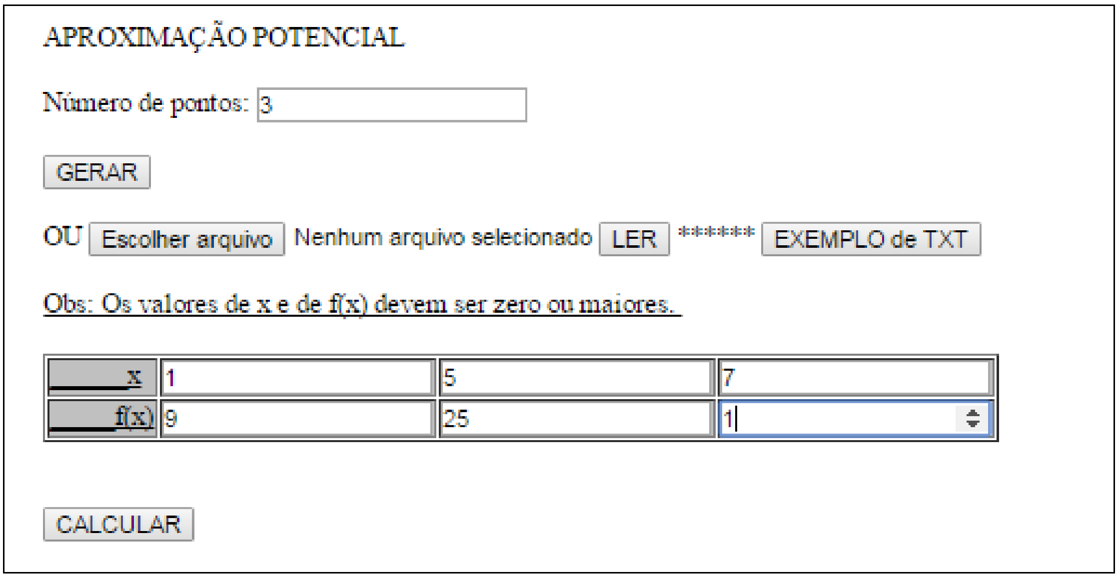


Figura 59: Tabela de pontos digitada na página.

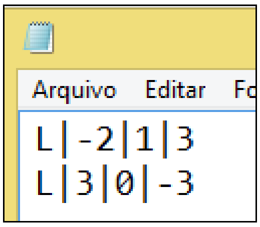


Figura 60: Exemplo de TXT

Depois de ler ou digitar a tabela de pontos, clica-se no botão “CALCULAR”, mostrado na Fig. 59 anterior, e irá aparecer o polinômio, o gráfico com os pontos gerado e a função aproximadora. A Fig. 61 a seguir apresenta um exemplo da tabela com 3 pontos da Fig. 59.

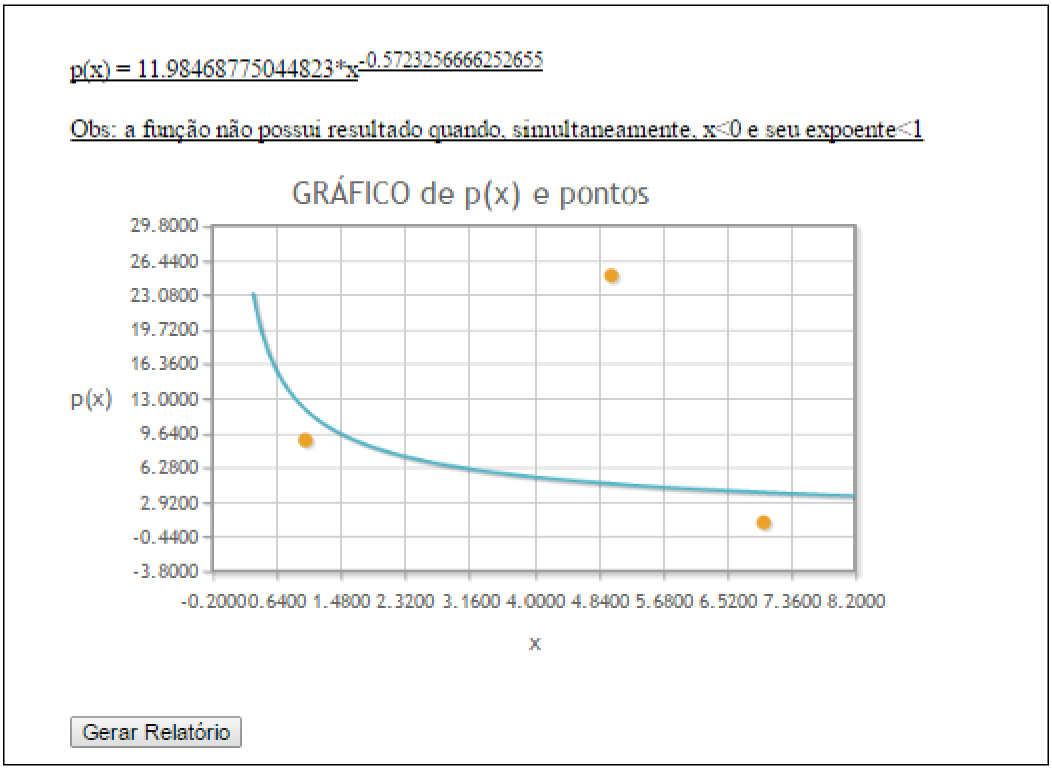


Figura 61: Polinômio, o gráfico com os pontos e a função aproximadora– Resultado Final.

Por fim, como principal objetivo para a criação da página, é possível gerar o relatório, clicando no botão “Gerar Relatório”, depois do resultado final, conforme a Fig. 61 anterior. O relatório é em PDF e contém a resolução detalhada, além de uma breve explicação teórica e o algoritmo principal para o estudo de cálculo numérico. A figura a seguir, Fig. 62, mostra o relatório para os pontos exemplificados na Fig. 60.

Número de pontos=3

Tabela de Pontos:

x	f(x)
1.000	9.0000
5.0000	25.0000
7.0000	1.000

Uma função aproximadora é uma função que segue a tendência dos dados, mas não necessariamente passa por todos os pontos.
Além do uso para extrapolação do domínio, a funçao aproximadora também é indicada quando o número de pontos é muito grande.
A aproximação por uma função potencial, do tipo y=a*(x^b), é calculada pelo Método dos Mínimos Quadrados com alguns ajustes.
Primeiramente deve se aplicar o logaritmo neperiano (Ln) dos dois lados da função, tornando-a do tipo ln(y)=ln(a)+b*ln(x).
E assim se faz uma analogia entre a função exponencial e uma polinomial de grau 1 (linear) do tipo z=a0+a1*k:
z=ln(y)=ln(f(x))
a0=ln(a)
a1=b
k=ln(x)
A partir desta função "linearizada", aplica-se o Método dos Mínimos Quadrados.

Abaixo segue a tabela de pontos depois de aplicados os logaritmos:

k	z
0.000	2.1972
1.6094	3.2189
1.9459	0.000

Aplicando o método dos mínimos quadrados linear, encontra-se um sistema de equações lineares, que pode ser escrito da seguinte forma:

$$\begin{bmatrix} (n+1) & \sum_{0}^{n} k_i \\ \sum_{0}^{n} k_i & \sum_{0}^{n} k_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_{0}^{n} z_i \\ \sum_{0}^{n} z_i * k_i \end{bmatrix}$$

Abaixo segue o sistema formado na forma matricial:

3.0000	3.5553	*5.4161*
3.5553	6.3767	*5.1805*

Este sistema pode ser resolvido, por exemplo, por Gauss, que pode ser estudado em http://vtp.ifsp.edu.br/nev/Sistema-gauss/sistemagauss.php?. Para tanto, utilize o arquivo txt gerado juntamente com este relatório, que pode ser lido pelo referido programa.

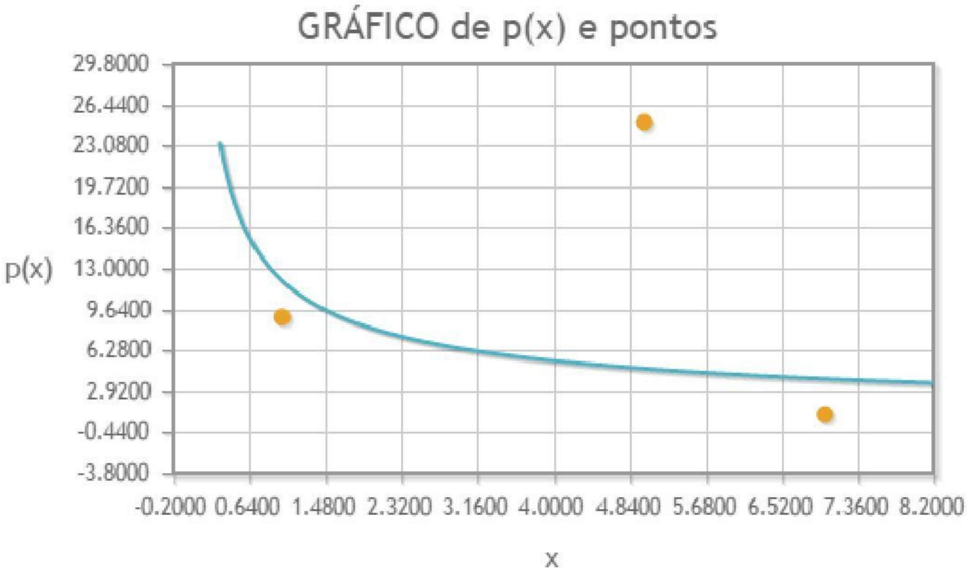
Resolvendo o sistema acima encontram-se os seguintes valores de a:
a1=-0.5723
a0=2.4836

Finalmente, para voltar à função potencial original, y=a*(x^b), deve-se fazer:
a = e^a0
b = a1.

Com isso a função potencial formada é:

$p(x) = 11.9847*x^{-0.5723}$

Segue abaixo gráfico da função dada a partir dos pontos:



A seguir, apresenta-se o algoritmo principal para montar a matriz para encontrar a função aproximadora:

```
f=-1;
para c=1  até p+1 { //colunas – p é o grau do polinômio
   para l=1 até p+1 { //linhas
   e=l+f;
```

```
soma=0;
 soma1=0
   para i=0 até n { //n+1 pontos
   soma=soma+Math.pow(xp[i],e);
      soma1=soma1+(fxp[i]*Math.pow(xp[i],(l-1)));;
 } fim do para i
  a(l,c)=soma
   b(l)=soma1
} fim do para l
f=f+1;
} fim do para c
a(1,1)=n+1;
Execute o algoritmo de Eliminação de Gauss. //http://vtp.ifsp.edu.br/nev/Sistema-gauss/sistemagauss.php?
```

REFERÊNCIAS BIBLIOGRÁFICAS

RUGGIERO, M.A.G; LOPES, V.L.R. Cálculo numérico: aspectos teóricos e computacionais. 2.ed. São Paulo: Makron Books, 1996.

SWOKOWSKI. Cálculo com Geometria Analítica. São Paulo. Makron Books, 1995.

QUADROS, Régis S. de; BORTOLI, Álvaro L. de. Fundamentos de Cálculo Numérico para Engenheiros. Porto Alegre, 2009. Disponível em < http://www.inf.ufrgs.br/~rlflupchinski/files/20112/NUMERICO/numerico-bortoli.pdf . Acesso em outubro de 2014.

Figura 62: Relatório gerado em PDF para o exemplo apresentado na Fig. 60.

3.11. Aproximação Exponencial

A página para Aproximação Exponencialapresenta-se como a Fig.63a seguir.

Figura 63: Página para AproximaçãoExponencial.

Para iniciar os cálculos, tem-se duas opções: ou gerar a tabela de dados na própria página, a partir do número de pontos escolhido, clicando no botão “GERAR”, e digitar os valores (Fig. 64 a seguir) de cada coordenada dos pontos, ou criar um arquivo TXT, seguindo o padrão apresentado ao se clicar no botão “EXEMPLO de TXT” (Fig. 65 a seguir).

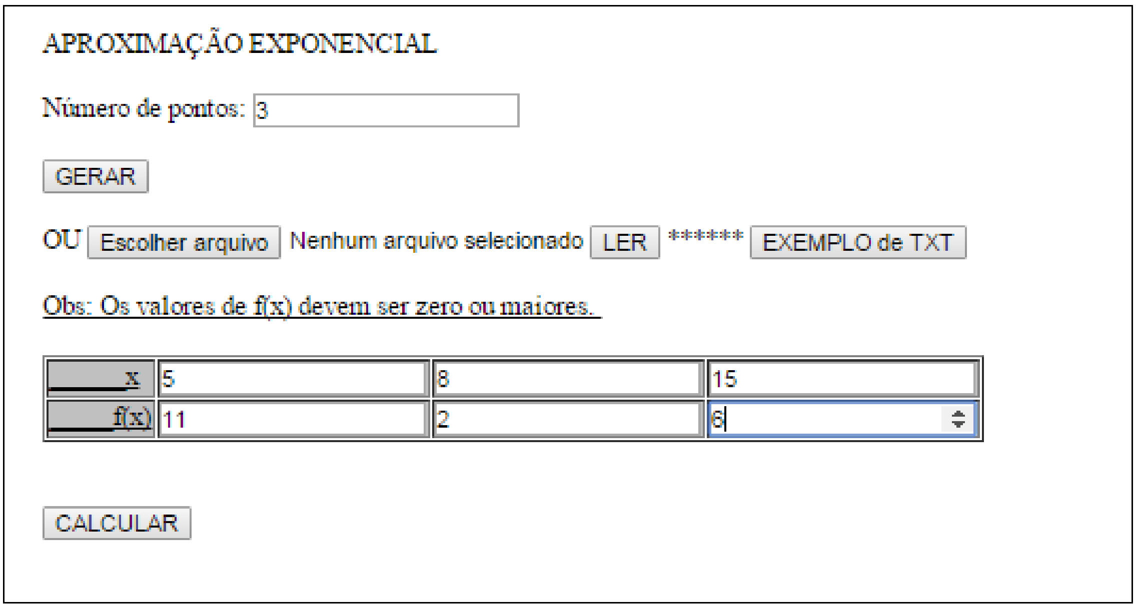


Figura 64: Tabela de pontos digitada na página.

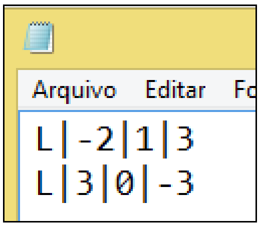


Figura 65: Exemplo de TXT

Depois de ler ou digitar a tabela de pontos, clica-se no botão “CALCULAR”, mostrado na Fig. 54 anterior, e irá aparecer o polinômio, o gráfico com os pontos gerado e a função aproximadora. A Fig. 66 a seguir apresenta um exemplo da tabela com 3 pontos da Fig. 64.

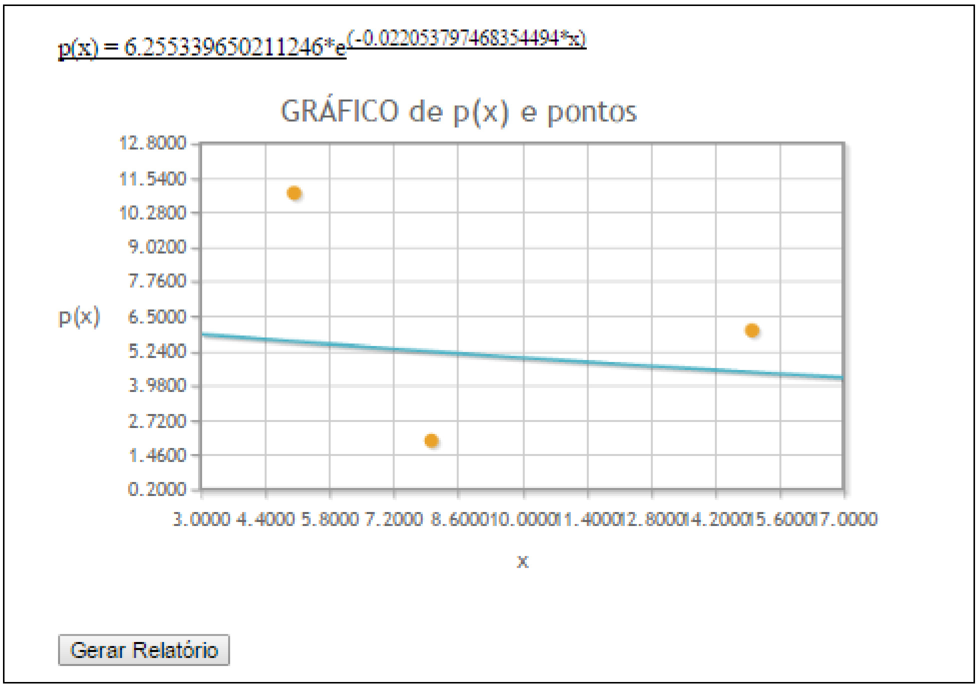


Figura 66: Polinômio, o gráfico com os pontos e a função aproximadora – Resultado Final.

Por fim, como principal objetivo para a criação da página, é possível gerar o relatório, clicando no botão “Gerar Relatório”, depois do resultado final, conforme a Fig. 66 anterior. O relatório é em PDF e contém a resolução detalhada, além de uma breve explicação teórica e o algoritmo principal para o estudo de cálculo numérico. A figura a seguir, Fig. 67, mostra o relatório para os pontos exemplificados na Fig. 64.

Número de pontos=3

Tabela de Pontos:

x	f(x)
5.0000	11.0000
8.0000	2.0000
15.0000	6.0000

Uma função aproximadora é uma função que segue a tendência dos dados, mas não necessariamente passa por todos os pontos.
Além do uso para extrapolação do domínio, a funçao aproximadora também é indicada quando o número de pontos é muito grande.
A aproximação por uma função exponencial, do tipo y=a*(e^(b*x)), é calculada pelo Método dos Mínimos Quadrados com alguns ajustes.
Primeiramente deve se aplicar o logaritmo neperiano (Ln) dos dois lados da função, tornando-a do tipo ln(y)=ln(a)+b*x.
E assim se faz uma analogia entre a função exponencial e uma polinomial de grau 1 (linear) do tipo z=a0+a1*x:
z=ln(y)=ln(f(x))
a0=ln(a)
a1=b
A partir desta função "linearizada", aplica-se o Método dos Mínimos Quadrados.

Abaixo segue a tabela de pontos depois de aplicado o logaritmo:

x	z
5.0000	2.3979
8.0000	0.6931
15.0000	1.7918

Aplicando o método dos mínimos quadrados linear, encontra-se um sistema de equações lineares, que pode ser escrito da seguinte forma:

$$\begin{bmatrix} (n+1) & \sum_0^n x_i \\ \sum_0^n x_i & \sum_0^n x_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_0^n z_i \\ \sum_0^n z_i * x_i \end{bmatrix}$$

Abaixo segue o sistema formado na forma matricial:

3.0000	28.0000	*4.8828*

28.0000	314.0000	*44.4113*

Este sistema pode ser resolvido, por exemplo, por Gauss, que pode ser estudado em http://vtp.ifsp.edu.br/nev/Sistema-gauss/sistemagauss.php?. Para tanto, utilize o arquivo txt gerado juntamente com este relatório, que pode ser lido pelo referido programa.

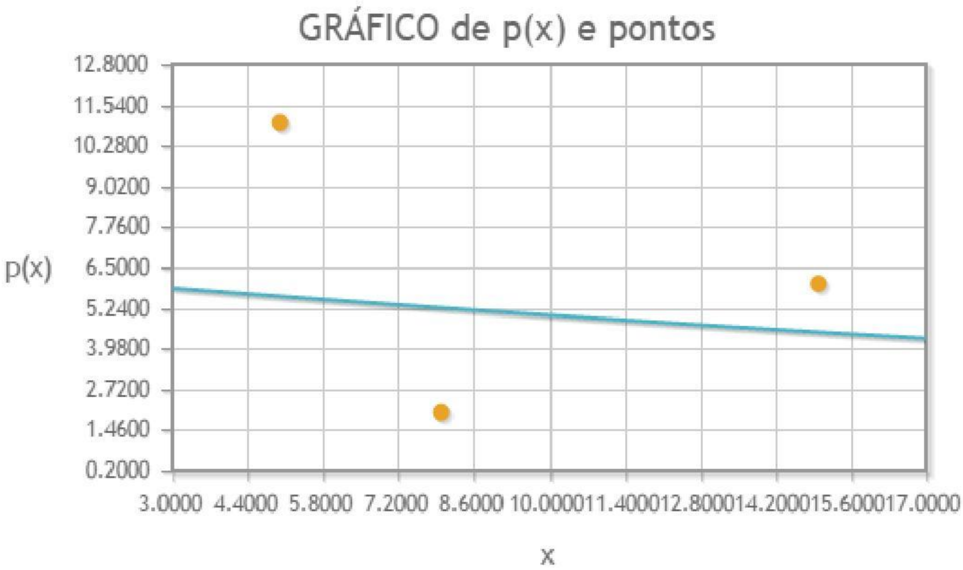
Resolvendo o sistema acima, encontram-se os seguintes valores de a:
a1=-0.02205
a0=1.8334

Finalmente, para voltar à função exponencial original, y=a*(e^(b*x)), deve-se fazer:
a = e^a0
b = a1.

Com isso a função exponencial formada é:

$p(x) = 6.2553*e^{-0.02205x}$

Segue abaixo gráfico da função dada a partir dos pontos:



A seguir, apresenta-se o algoritmo principal para montar a matriz dos coeficientes para encontrar a função aproximadora:

```
f=-1;
para c=1  até p+1 { //colunas – p é o grau do polinômio
  para l=1 até p+1 { //linhas
  e=l+f;
  soma=0;
   soma1=0
     para i=0 até n { //n+1 pontos
```

```
      soma=soma+Math.pow(xp[i],e);
         soma1=soma1+(fxp[i]*Math.pow(xp[i],(l-1)));;
 } fim do para i
   a(l,c)=soma
    b(l)=soma1
} fim do para l
f=f+1;
} fim do para c
a(1,1)=n+1;
Execute o algoritmo de Eliminação de Gauss. //http://vtp.ifsp.edu.br/nev/Sistema-gauss/
sistemagauss.php?
```

REFERÊNCIAS BIBLIOGRÁFICAS

RUGGIERO, M.A.G; LOPES, V.L.R. Cálculo numérico: aspectos teóricos e computacionais. 2.ed. São Paulo: Makron Books, 1996.

SWOKOWSKI. Cálculo com Geometria Analítica. São Paulo. Makron Books, 1995.

QUADROS, Régis S. de; BORTOLI, Álvaro L. de. Fundamentos de Cálculo Numérico para Engenheiros. Porto Alegre, 2009. Disponível em < http://www.inf.ufrgs.br/~rlflupchinski/files/20112/NUMERICO/numerico-bortoli.pdf . Acesso em outubro de 2014.

Figura 67: Relatório gerado em PDF para o exemplo apresentado na Fig. 64.

3.12. Aproximação Potencial

A página para Aproximação Potencialapresenta-se como a Fig.68a seguir.

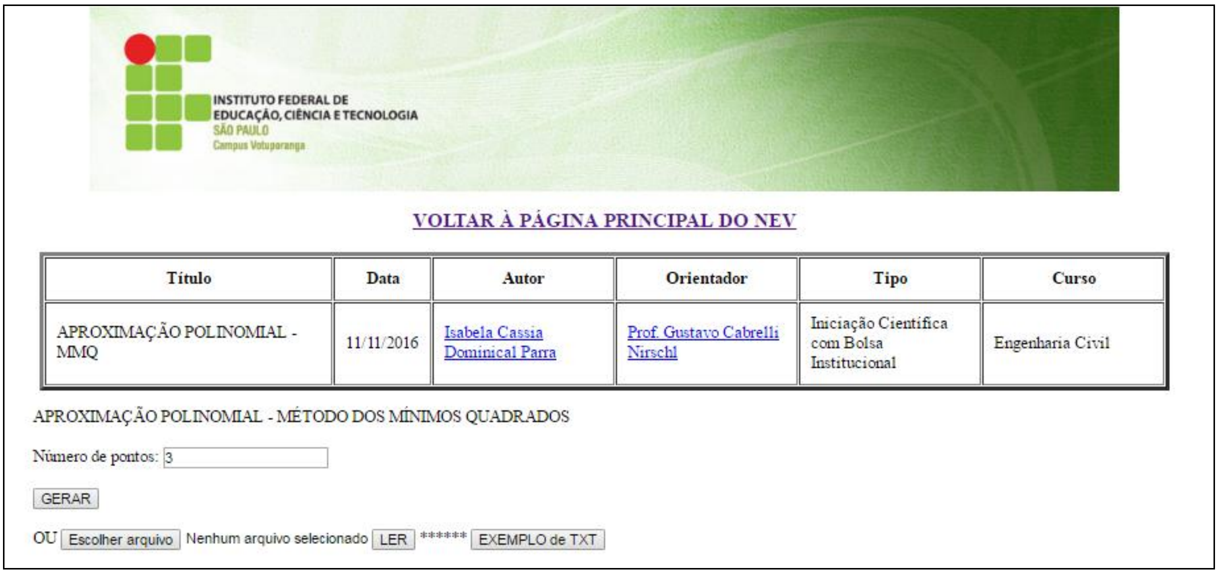


Figura 68: Página para Aproximação Potencial.

Para iniciar os cálculos, tem-se duas opções: ou gerar a tabela de dados na própria página, a partir do número de pontos escolhido, clicando no botão “GERAR”, e digitar os valores (Fig. 69 a seguir) de cada coordenada dos pontos, ou criar um arquivo TXT, seguindo o padrão apresentado ao se clicar no botão “EXEMPLO de TXT” (Fig. 70 a seguir).

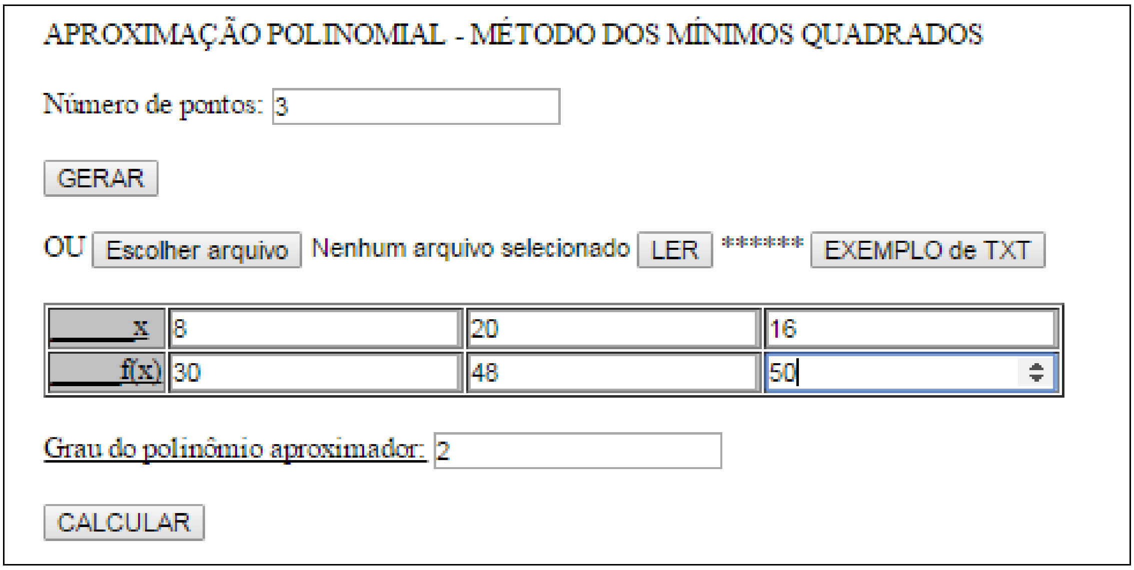


Figura 69: Tabela de pontos digitada na página.

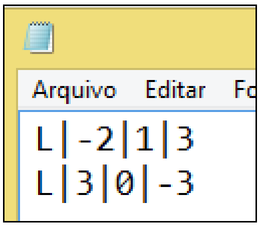


Figura 70: Exemplo de TXT

Depois de ler ou digitar a tabela de pontos deve se escolher o grau do polinômio, e depoisclica-se no botão “CALCULAR”, mostrado na Fig. 69 anterior, e irá aparecer o polinômio, o gráfico com os pontos gerado e a função aproximadora. A Fig. 71 a seguir apresenta um exemplo da tabela com 3 pontos da Fig. 69.

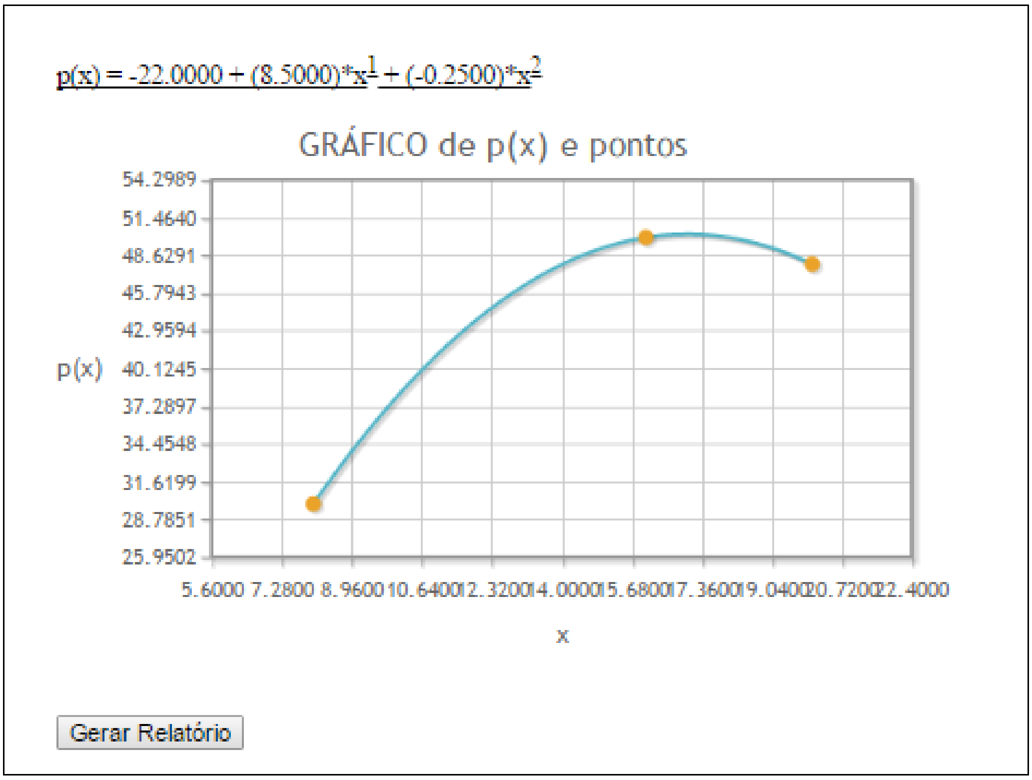


Figura 71: Polinômio, o gráfico com os pontos e a função aproximadora – Resultado Final.

Por fim, como principal objetivo para a criação da página, é possível gerar o relatório, clicando no botão “Gerar Relatório”, depois do resultado final, conforme a Fig. 71 anterior. O relatório é em PDF e contém a resolução detalhada, além de uma breve explicação teórica e o algoritmo principal para o estudo de cálculo numérico. A figura a seguir, Fig. 72, mostra o relatório para os pontos exemplificados na Fig. 70.

Número de pontos=3

Tabela de Pontos:

x	f(x)
8.0000	30.0000
20.0000	48.0000
16.0000	50.0000

Grau do Polinômio = 2

Uma função aproximadora é uma função que segue a tendência dos dados, mas não necessariamente passa por todos os pontos.
Além do uso para extrapolação do domínio, a funçao aproximadora também é indicada quando o número de pontos é muito grande, o que gera uma função interpoladora polinomial de alto grau.
Neste programa, a aproximação por polinômios é feita por meio do Método dos Mínimos Quadrados.
Seja uma função polinomial aproximadora de grau "p", do tipo g(x)= a0+ a1*(x^1)+a2*(x^2)+...+ ap*(x^p), e a função analítica f(x) que passa por todos os pontos.
Obviamente, em cada ponto, ocorre um erro dado por: r(x)= f(x)-g(x).
A função M, dos erros mínimos quadrados, é dada por:

$$M(a_0, a_1, \ldots, a_m) = \sum_{i=0}^{n} r_i^2(x) = \sum_{i=0}^{n} [f(x_i) - g(x_i)]^2$$

A fim de minimizar os erros (ou a função M), aplica-se a derivada parcial de M em relação a cada variável "ai", encontrando-se um sistema de equações lineares que pode ser escrito da forma a seguir. Sua resolução permite encontrar as variáveis "ai", que formam a função aproximadora.

$$\begin{bmatrix} (n+1) & \sum_0^n x_i & \sum_0^n x_i^2 & \cdots & \sum_0^n x_i^p \\ \sum_0^n x_i & \sum_0^n x_i^2 & \sum_0^n x_i^3 & \cdots & \sum_0^n x_i^{p+1} \\ \sum_0^n x_i^2 & \sum_0^n x_i^3 & \sum_0^n x_i^4 & \cdots & \sum_0^n x_i^{p+2} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \sum_0^n x_i^p & \sum_0^n x_i^{p+1} & \sum_0^n x_i^{p+2} & \cdots & \sum_0^n x_i^{2p} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \cdots \\ a_p \end{bmatrix} = \begin{bmatrix} \sum_0^n f(x_i) \\ \sum_0^n f(x_i)x_i \\ \sum_0^n f(x_i)x_i^2 \\ \cdots \\ \sum_0^n f(x_i)x_i^p \end{bmatrix}$$

Para os dados de entrada que temos:

3.0000	44.0000	720.0000	*128.0000*

44.0000	720.0000	12608.0000	*2000.0000*
720.0000	12608.0000	229632.0000	*33920.0000*

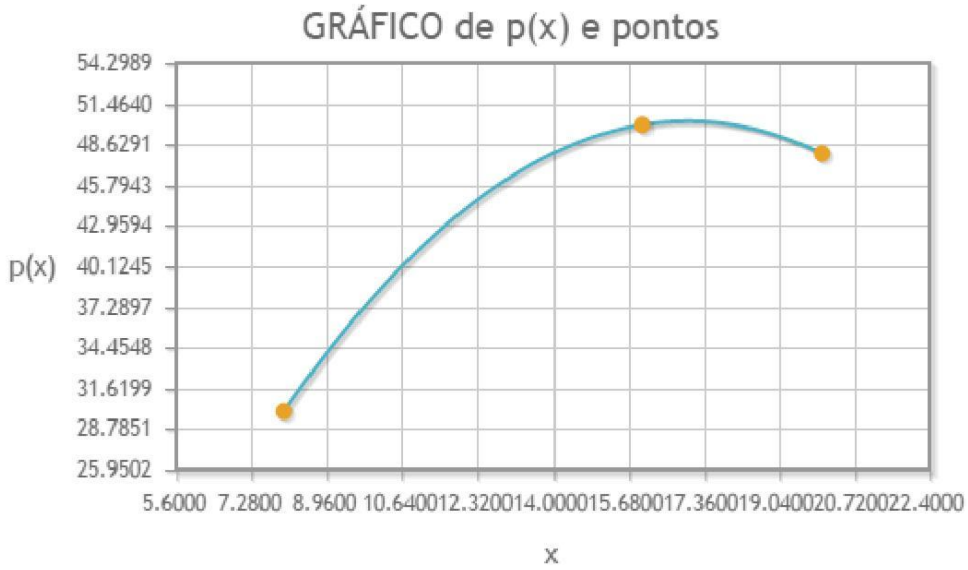
Este sistema pode ser resolvido, por exemplo, por Gauss, que pode ser estudado em http://vtp.ifsp.edu.br/nev/Sistema-gauss/sistemagauss.php?. Para tanto, utilize o arquivo txt gerado juntamente com este relatório, que pode ser lido pelo referido programa.

Resolvendo o sistema acima, encontram-se os seguintes valores de "ai":
a2=-0.2500
a1=8.5000
a0=-22.0000

Com isso a função aproximadora polinomial encontrada é:

$p(x) = -22.0000+(8.5000)*x^{1}+(-0.2500)*x^{2}$

Segue abaixo o gráfico da função aproximadora e os pontos (dados de entrada):



A seguir, apresenta-se o algoritmo principal para montar o sistema matricial que gera o polinômio aproximador:

```
f=-1;
para c=1  até p+1 { //colunas – p é o grau do polinômio
  para l=1 até p+1 { //linhas
  e=l+f;
  soma=0;
   soma1=0
     para i=0 até n { //n+1 pontos
     soma=soma+x(i)^e;
```

```
        soma1=soma1+(f(xi)*(x(i)^(l-1)));
 } fim do para i
   a(l,c)=soma
    b(l)=soma1
} fim do para l
 f=f+1;
 } fim do para c
a(1,1)=n+1;
Execute o algoritmo de Eliminação de Gauss. //http://vtp.ifsp.edu.br/nev/Sistema-gauss/
sistemagauss.php?
```

REFERÊNCIAS BIBLIOGRÁFICAS

RUGGIERO, M.A.G; LOPES, V.L.R. Cálculo numérico: aspectos teóricos e computacionais. 2.ed. São Paulo: Makron Books, 1996.

SWOKOWSKI. Cálculo com Geometria Analítica. São Paulo. Makron Books, 1995.

QUADROS, Régis S. de; BORTOLI, Álvaro L. de. Fundamentos de Cálculo Numérico para Engenheiros. Porto Alegre, 2009. Disponível em < http://www.inf.ufrgs.br/~rlflupchinski/files/20112/NUMERICO/numerico-bortoli.pdf >. Acesso em outubro de 2014.

Figura 72: Relatório gerado em PDF para o exemplo apresentado na Fig. 70.

Os programas para o Método de Euler para Solução de PVI's de Ordem 1, para o Método de Runge-Kutta para o cálculo de PVI'S de ordem 1 e para Método das diferenças finitas para a resolução de PVC'S de segunda ordemestão também postados na página no NEV.

CAPÍTULO 4

CONCLUSÕES

O relatório PDF detalhando os passos da resolução contribui significativamente para o estudo de cada método. O meio on-line foi escolhido para propiciar praticidade e ausência de custos aos estudantes e profissionais. Cabe ressaltar que este projeto faz parte de um grupo de pesquisa do CNPq e possui outras páginas com programas na área de Cálculo Numérico e Engenharia Civil (acesse o NEV: http://vtp.ifsp.edu.br/nev/).

REFERÊNCIAS BILBIOGRÁFICAS

FTOOL.Ftool para Windows, versão 3.00. PUC-Rio, 2012. Disponível em <http://www.tecgraf.puc-rio.br/ftool/>. Acesso em 29 de junho de 2015.

QUADROS, Régis S. de; BORTOLI, Álvaro L. de. **Fundamentos de Cálculo Numérico para Engenheiros**. Porto Alegre, 2009. 265p.

RUGGIERO, M.A.G; LOPES, V.L.R. **Cálculo numérico: aspectos teóricos e computacionais.**2.ed. São Paulo: Makron Books, 1996.

SWOKOWSKI. Cálculo com Geometria Analítica. São Paulo. Makron Books, 1995.

TQS. TQS Online, versão 18. [S.l.]: TQS Informática, 2015. Disponível em <http://www.tqs.com.br/>. Acesso em: 10 de junho de 2015.

W3SCHOOLS.W3schools Online. [S.l]. Disponível em <http://www.w3schools.com>. Acesso em 05 de maio de 2015.