



VISUALIZAÇÃO DE FUNÇÕES MATEMÁTICAS EM TEMPO REAL: UMA ABORDAGEM DE APOIO AO ENSINO DE MATEMÁTICA AOS ALUNOS INGRESSANTES EM ENGENHARIA

Gabriel Inácio de Mello – gabrielinaciodemello@gmail.com
Faculdade Municipal "Professor Franco Montoro", FMPFM
Centro Regional Universitário de Espírito Santo do Pinhal, UNIPINHAL
Rua dos Estudantes s/n, Cachoeira de Cima
CEP 13840-000 – Mogi Guaçu - SP

José Tarcísio Franco de Camargo – jtfc@bol.com.br
Faculdade Municipal "Professor Franco Montoro", FMPFM
Centro Regional Universitário de Espírito Santo do Pinhal, UNIPINHAL
Rua dos Estudantes s/n, Cachoeira de Cima
CEP 13840-000 – Mogi Guaçu - SP

Estéfano Vizconde Veraszto – estefanovv@cca.ufscar.br
Universidade Federal de São Carlos
Depto. de Ciências da Natureza, Matemática e Educação, UFSCar, CCA
Rodovia Anhanguera, Km 174
CEP 13604-900 – Araras - SP

Gilmar Barreto – gbarreto@dsif.fee.unicamp.br
Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação, UNICAMP, FEEC
Av. Albert Einstein, 400, Cidade Universitária Zeferino Vaz, Barão Geraldo
CEP 13083-852 – Campinas - SP

Resumo: *Um problema recorrente no ensino de engenharia encontra-se na dificuldade apresentada pelos alunos ingressantes em manipular funções matemáticas. Esta dificuldade, muitas vezes, acaba por limitar ou comprometer o aprendizado de fundamentos matemáticos essenciais à formação do engenheiro em seus anos iniciais de estudo. Neste sentido, este artigo apresenta uma estratégia computacional para o apoio à visualização de funções matemáticas, com o intuito de facilitar o ensino desta disciplina. O software desenvolvido para esta abordagem possui código aberto, de fácil manutenção e atualização, podendo ser adaptado por seus próprios usuários de acordo com suas necessidades específicas. O mesmo é bastante intuitivo, sendo executado em navegadores web, podendo ser acessado pela internet em diversas plataformas. O aplicativo foi criado com as linguagens de programação HTML e Javascript em conjunto com a API WEBGL. O software desenvolvido no projeto é capaz de receber funções e dados do usuário, realizar os cálculos necessários e mostrar os resultados em três dimensões através de uma interface de visualização, auxiliando assim os usuários a entenderem melhor o comportamento das funções matemáticas.*

Palavras-chave: *Software educativo, Visualização gráfica, ensino de engenharia*

1. INTRODUÇÃO

O ensino de engenharia nos anos iniciais de um curso de graduação é frequentemente prejudicado pela deficiência em matemática apresentada pelos alunos oriundos do ensino médio. O ingressante em um curso de engenharia, muitas vezes chega ao ensino superior sem ter desenvolvido o grau de abstração necessário ao aprendizado de conceitos matemáticos fundamentais à sua formação profissional. Esta situação é notável, principalmente, no ensino de funções matemáticas e suas respectivas representações gráficas (RODRIGUES et al, 2012)

Neste contexto, o uso da computação gráfica e de softwares específicos vem se tornando cada vez mais frequente em diversas áreas, como ensino de ciências, ensino de engenharia ou ainda, em novas propostas metodológicas. Por outro lado, problemas relacionados ao ensino de Matemática tem incentivado cada vez mais trabalhos apoiados na utilização de tecnologias capazes de trazer inovação na área. Nesse sentido, algo que tem sido cada vez mais recorrente é o uso de softwares no ensino desta disciplina (CAMARGO et al, 2011a, 2011b; MOTA et al, 2013; SANTOS et al, 2013; VERASZTO et al 2012, 2013).

Segundo Valente (2013), para introduzir o computador no meio educacional são necessários quatro fatores essenciais: o computador, o software, o professor capacitado e o aluno. Neste quadro, o software é um fator tão importante quanto os outros, pois sem o software o computador jamais poderá ser utilizado na educação.

É dentro desse contexto que esse artigo propõe a criação de um software para auxiliar o ensino de funções matemáticas, cujo potencial é destinado ao ensino de diferentes disciplinas de matemática em cursos como os de engenharia ou informática, que necessitem da manipulação de funções. Com esse aplicativo, pretende-se contribuir para o ensino de disciplinas de matemática em cursos de engenharia e informática. Essa justificativa apoia-se no fato de que muitos estudantes, de maneira geral, chegam aos bancos universitários com grandes deficiências em matemática, que vão desde o domínio de operações básicas com números naturais até a manipulação de funções reais passando pela teoria polinomial e também pela geometria plana e trigonometria (ROJAS, RITTO & BARBOSA, 2008).

2. FUNDAMENTAÇÃO TEÓRICA

2.1. O ensino de matemática

Conforme apontado, muitos alunos ingressam no ensino superior com grandes deficiências em matemática. Este problema muitas vezes têm origem em muitos dos conteúdos que seriam a base dos estudos do ensino superior e que nunca foram vistos pelos alunos ao longo de sua trajetória escolar (ROJAS, RITTO & BARBOSA, 2008).

Esse fato ainda é agravado pela quantidade de matéria curricular exigida nos cursos de graduação, o que implica em pouco tempo para que o aluno ultrapasse a defasagem conceitual e se encarregue de estudar novos conteúdos sem ter os fundamentos necessários.

2.2. Utilização de tecnologias no ensino de matemática

Atualmente temos a tecnologia espalhada em todos os setores sociais. Processos administrativos são controlados por computadores, da mesma forma que transações bancárias não mais se efetuam sem a interligação de sistemas inteligentes. Indústrias são automatizadas e todo o sistema de comunicação global é gerenciado por softwares e hardwares de elevado



poder computacional. Até a maneira com que adquirimos informações vem sendo modificada nos últimos anos. Hoje em dia temos acesso à internet e a programação televisiva das mais variadas natureza que nos permitem ter contato, sem ao menos sair de casa, com assuntos antes inacessíveis (CAMARGO et al, 2011a). Por outro lado, essa dinâmica do processamento e compartilhamento da informação parece que ainda não chegou aos bancos escolares. O ensino ainda é praticado de forma tradicional, centrado em professores que geralmente utilizam o discurso, o giz e a lousa na tentativa de ensinar (VERASZTO et al, 2012).

Pensando no cenário rapidamente descrito, é possível concluir que a facilidade de utilização de sistemas computacionais é um fator que pode contribuir também para que processos de ensino-aprendizagem sejam feitos de forma consciente. Se a mídia consegue utilizar recursos para entreter, profissionais da área de educação também precisam aprender a utilizar esses recursos para ensinar. Conforme afirmam Calil, Veiga e Carvalho, "*a informática abre possibilidades de mudanças dentro do conhecimento e a relação desse com o sujeito que aprende superando os problemas da prática do ensino tradicional*" (p.19).

A tentativa de deixar as aulas mais lúdicas, de tornar visível aquilo que é difícil de abstrair, é algo que os educadores devem tomar como desafio primordial da sua tarefa profissional.

Nesse sentido, no próximo tópico serão apresentadas algumas considerações acerca do potencial de algumas ferramentas tecnológicas para o cumprimento dos objetivos deste trabalho.

2.3. Recursos tecnológicos de apoio à educação

Os recursos tecnológicos utilizados para o desenvolvimento deste trabalho são fundamentados em dois "pacotes": WEBGL e THREE.JS. Estas são ferramentas (*softwares*) de grande potencial para o desenvolvimento de aplicações na área da educação, especialmente dentro do contexto tratado neste artigo.

WEBGL: Acesso ao hardware de vídeo através de paginas web

O WEBGL é uma especificação de API desenvolvida especialmente para navegadores *web*. O objetivo dessa especificação é estender a capacidade do HTML/Javascript de forma que seja possível acessar o hardware de vídeo e gerar gráficos 3D interativos, sintetizando imagens através do componente HTML 5 denominado "canvas".

Devido ao fato de ser uma especificação, cada navegador *web* deve implementar a sua versão da API, de maneira que ela possa ser utilizada sem a utilização de *plug-ins*. O WEBGL foi construído com base no OPENGL ES 2.0, o que permite que ele seja implantado também em navegadores *web* de dispositivos embarcados, tais como *smartphones* ou *tablets*, por exemplo, contribuindo para a aplicação das novas tecnologias em educação através do uso de dispositivos portáteis.

O uso do WEBGL vem se tornando cada vez mais frequente por desenvolvedores de jogos e aplicações interativas, pois permite a criação de aplicações multiplataforma de maneira simples e rápida, devido ao fato de que a abstração de hardware e sistema operacional fica por conta dos navegadores. Por ser uma API baseada em OPENGL ES, a migração de aplicações embarcadas para aplicações *web* se torna simples, e hoje é possível ver vários exemplos de aplicações e jogos para dispositivos móveis sendo portadas para WEBGL.



THREE.JS: Uma biblioteca para facilitar o desenvolvimento WEBGL

Apesar de o WEBGL ser uma API de alto nível, sua estrutura e sintaxe são consideradas complexas e o desenvolvimento de aplicações tridimensionais interativas com esta API pode se tornar muito trabalhoso e demorado. Assim, para tornar o desenvolvimento mais simples e intuitivo, foi criada a "THREE.JS", uma biblioteca que abstrai as chamadas ao hardware de vídeo em um nível muito maior que o da API WEBGL.

A THREE.JS fornece diversos recursos, desde os mais simples até os mais complexos, o que a torna uma ferramenta versátil para que os desenvolvedores criem aplicações de alto nível de complexidade com uma codificação relativamente simples.

A biblioteca surgiu em abril de 2010, quando foi publicada como um projeto de código aberto no Git Hub pelo desenvolvedor Ricardo Cabello, que planejava criar o seu próprio "motor 3D". Logo após a sua publicação, vários desenvolvedores começaram a utilizar esta biblioteca e a publicar diversas demonstrações de aplicações interativas tridimensionais utilizando-a, o que a tornou uma das mais populares entre as disponíveis para esse tipo de desenvolvimento *web*.

Os recursos presentes na THREE.JS são divididos em várias categorias e subcategorias, que juntos formam uma plataforma ideal para o desenvolvimento de aplicações tridimensionais interativas. Os recursos inclusos são: *Renderers, Scenes, Cameras, Animation, Lights, Materials, Shaders, Objects, Geometry, Loaders, Utilities e Export/Import* (Vierros, 2013).

A THREE.JS foi criada com o propósito de facilitar ao máximo o desenvolvimento das aplicações, portanto, com poucas linhas de código é possível criar uma aplicação funcional. O processo para criação da aplicação consiste em alguns passos que definem a composição geral de uma aplicação tridimensional. São eles: definir a tela de renderização, definir uma câmera para a cena, criar e inserir objetos na cena, inserir luzes para tornar a cena visível, e iniciar o loop de síntese das imagens.

A THREE.JS oferece diversas vantagens para o desenvolvimento de aplicações tridimensionais interativas, tais como uma quantidade relativamente grande de recursos pré-programados, estrutura simples e organizada, uma grande quantidade de desenvolvedores ativos no projeto, constantes atualizações, documentação ampla e detalhada e uma extensa biblioteca de exemplos. A mesma conta também com uma crescente comunidade de usuários, que organizam fóruns e eventos para trocar informações sobre a biblioteca.

A maior desvantagem de se utilizar a THREE.JS é o fato de esta ser uma biblioteca de alto nível de abstração, a qual, em projetos muito específicos, pode não conter as funcionalidades necessárias para o desenvolvimento do projeto, sendo então necessário que tais funcionalidades sejam programadas no código fonte da biblioteca.

3. METODOLOGIA

Conforme previamente descrito, este trabalho apresenta uma proposta que tem por objetivo facilitar o aprendizado da matemática pelo aluno ingressante em um curso de engenharia ou ciências exatas, através do uso de um *software* interativo aplicado à visualização de funções matemáticas. Através deste instrumento, espera-se que o aluno venha a desenvolver e consolidar a abstração necessária ao aprendizado de diversos fundamentos matemáticos.



O trabalho pretende alcançar este objetivo permitindo que o aluno interaja, em tempo real, com diversas funções matemáticas, através de gráficos e de suas respectivas formulações algébricas (equações). Assim, ao poder visualizar e manipular um gráfico em tempo real a partir de sua expressão algébrica, o estudante pode vir a compreender melhor esta ciência (a matemática), a qual é fundamental para o desenvolvimento de boa parte das habilidades profissionais que devem ser desenvolvidas pelo engenheiro.

O desenvolvimento deste objeto de aprendizagem foi fundamentado no uso de software livre, tendo como meta não apenas a construção do produto em si, mas também o uso e a divulgação de ferramentas públicas, compatíveis com as mais diversas plataformas computacionais, acessíveis a todos e de fácil utilização, manutenção e atualização. As "ferramentas" WebGL e Three.js se encaixam perfeitamente dentro desta filosofia.

Através do aplicativo o usuário pode inserir uma função matemática e os dados que deseja visualizar e, em tempo real, os resultados são mostrados em um espaço cartesiano tridimensional. O software é executado através de um navegador *web*, o que permite que ele seja facilmente distribuído pela internet e funcione em múltiplas plataformas.

Para criar o espaço cartesiano tridimensional é utilizada a API WEBGL em conjunto com a biblioteca THREE.JS, que permitem acessar o hardware de vídeo e criar gráficos tridimensionais através de um navegador *web*. Para realizar os cálculos matemáticos é utilizada a linguagem de programação Javascript. Para criar a interface com o usuário foi utilizada a linguagem HTML.

Todas as ferramentas utilizadas no desenvolvimento do software são gratuitas, limitando o custo da aplicação a recursos computacionais e desenvolvedores capacitados. Este artigo não pretende apresentar conceitos gerais sobre computação gráfica. Diversas informações a esse respeito podem ser encontradas em Barzel (1992), Foley e outros (1994), Hetem Junior (2006), Rogers (1985), Rogers e Adams (1990) e Schineider e outros (2003).

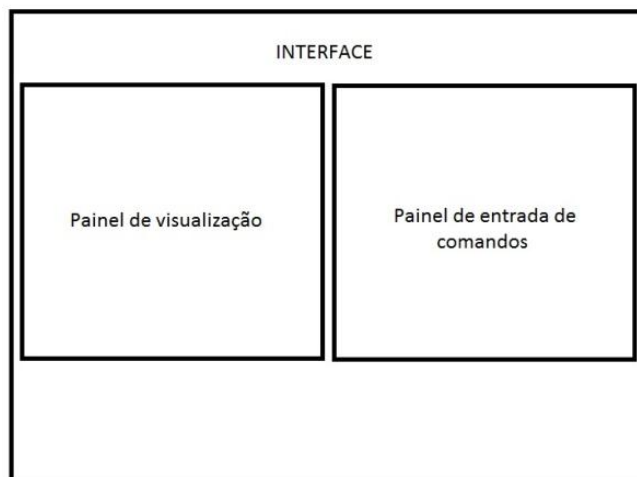
4. RESULTADOS DO TRABALHO

A aplicação desenvolvida nesse projeto partiu da premissa de ser, a princípio, uma ferramenta que auxiliasse o meio educacional, sendo simples de ser utilizada, fácil de ser distribuída e, também, multiplataforma. Partindo dessas premissas foi criado um visualizador de funções matemáticas que é executado em um navegador *web*, o qual pode ser acessado por qualquer dispositivo que possua acesso à internet além de um navegador compatível.

4.1. A interface

Para facilitar o uso da aplicação, a interface de usuário foi projetada de maneira simples, sendo dividida em apenas dois painéis: o painel de visualização e o painel de entrada de comandos. Como pode ser visto na Figura 1, o painel de visualização fica à esquerda da tela, enquanto o painel de entrada de comandos fica à direita.

Figura 1 - Projeto de interface.



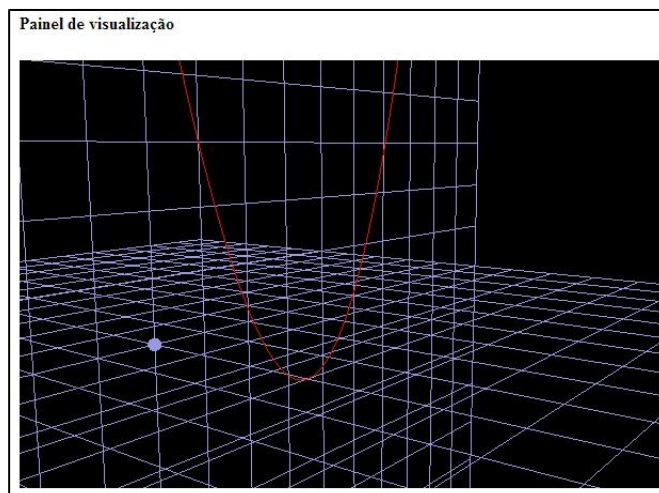
Painel de visualização

O painel de visualização é o local onde o usuário pode visualizar o resultado dos comandos que são inseridos no painel de entrada de comandos. Os resultados são exibidos em um espaço cartesiano de três dimensões, onde as linhas do espaço são mostradas na cor azul e a função resultante dos comandos inseridos é mostrada na cor vermelha.

Os resultados são exibidos em tempo real, e sempre que o usuário realizar alguma alteração nos comandos inseridos, o painel de visualização é atualizado. O usuário pode interagir com o painel de visualização. Utilizando o mouse ele pode ajustar a visualização da maneira que desejar, movendo a câmera nas direções vertical e horizontal, aproximando ou afastando a câmera do centro do espaço cartesiano ou, também, girando a câmera em torno do centro do espaço.

Na Figura 2 é mostrado o painel de visualização exibindo o gráfico de uma função matemática no espaço cartesiano.

Figura 2 - Painel de visualização.



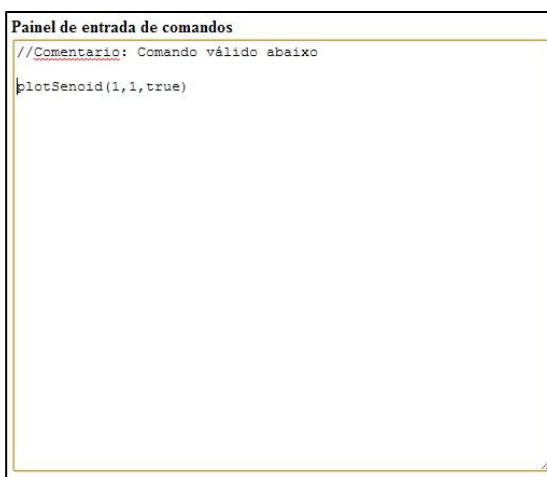
Painel de entrada de comandos

O painel de entrada de comandos é uma caixa de texto onde o usuário insere os comandos e os dados necessários para a aplicação calcular e exibir os resultados obtidos no painel de visualização. Para inserir os comandos, o usuário deve clicar com o botão esquerdo do mouse sobre o painel de entrada de comandos, assim o uso do teclado será habilitado.

Quando o usuário termina de escrever um comando válido, o resultado é automaticamente exibido no painel de visualização. Caso o usuário escreva um comando inválido, a aplicação irá ignorar automaticamente o comando e não exibirá resultado algum no painel de visualização.

Assim, como é visto em linguagens de programação, o usuário pode inserir linhas de comentários no painel de entrada de comandos, bastando que o mesmo escreva duas barras seguidas “//” antes de iniciar a linha de comentário, dessa forma a aplicação irá ignorar a linha de comentário e não executará os comandos presentes nela. Na Figura 3 é mostrado o painel de entrada de comandos com uma linha de comentário escrita, e em seguida um comando válido escrito.

Figura 3 - Painel de entrada de comandos.



```
Painel de entrada de comandos
//Comentario: Comando válido abaixo
plotSenoid(1,1,true)
```

Comandos válidos

A aplicação foi projetada para receber comandos e dados do usuário, para então realizar o processamento e exibir os resultados obtidos. Atualmente, a aplicação suporta a entrada de quatro comandos, mas foi projetada de forma a prever em trabalhos futuros a possibilidade de suporte a mais comandos, sem que seja totalmente reescrita.

Os quatro comandos suportados atualmente são: *plotTime*, *plot3d*, *plot2d* e *plotSenoid*. Para que os comandos sejam válidos, o usuário também deve fornecer os parâmetros necessários para cada comando realizar os seus respectivos cálculos. Os parâmetros são escritos entre parênteses, logo após o nome do comando, por exemplo:

Comando(parametro1,parametro2).

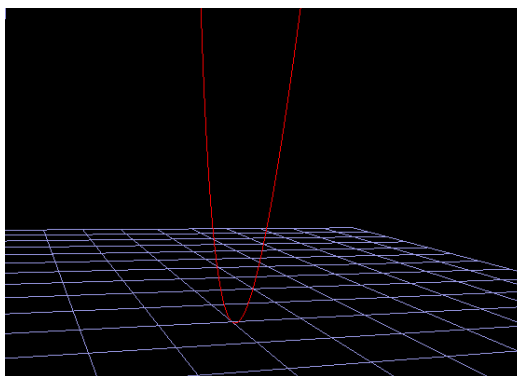
A sintaxe utilizada para escrever os comandos é a mesma da linguagem Javascript, suportando todas as funções matemáticas previstas para a linguagem.

Comando *plotTime*

O comando *plotTime* foi criado para que o usuário possa inserir equações especiais em função de uma variável de tempo. Para este comando ser válido, o usuário deve inserir cinco parâmetros. O primeiro parâmetro é o valor inicial da variável tempo, que será incrementada de maneira automática pela aplicação. O segundo parâmetro é o valor final do tempo, que irá determinar até qual valor a variável tempo será incrementada. O terceiro parâmetro é uma função que representará o valor de X no espaço cartesiano. O quarto parâmetro é uma função que representará o valor de Y no espaço cartesiano. O quinto parâmetro é uma função que representará o valor de Z no espaço cartesiano. Para esse comando também existe uma palavra reservada, *'time'*, que retorna o valor da variável tempo sempre que ela é incrementada.

A Figura 4 mostra o resultado do comando *plotTime* no painel de visualização.

Figura 4 - `plotTime(-10,10,'time',-Math.pow(time,2) , '3')`.

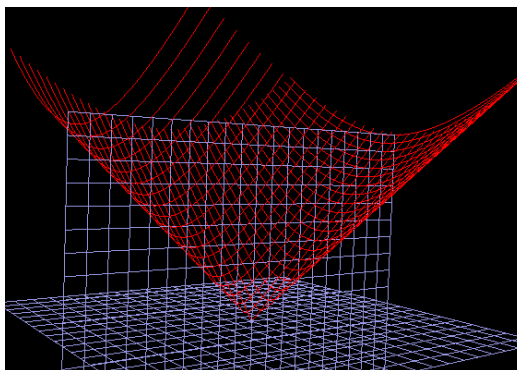


Comando *plot3d*

O comando *plot3d* foi criado para que o usuário possa inserir funções de três dimensões na aplicação. Para este comando ser executado corretamente o usuário deve inserir um parâmetro junto ao comando. O parâmetro necessário é uma função matemática de três dimensões, que será calculada e mostrada no painel de visualização.

Na Figura 5 pode ser visto o resultado do comando *plot3d*.

Figura 5 - `plot3d('Math.sqrt(Math.pow(x,2)+ Math.pow(y,2))')`.

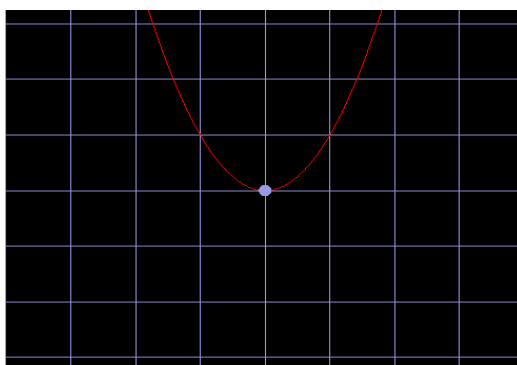


Comando *plot2d*

O comando *plot2d* foi criado para que o usuário possa visualizar o gráfico de uma função de duas dimensões. Para o comando ser válido, o usuário deve inserir três parâmetros. O primeiro parâmetro é o valor inicial da variável X, que será incrementada automaticamente pela aplicação. O segundo parâmetro é o valor final da variável X, que irá determinar até qual valor a variável X será incrementada. O terceiro parâmetro é uma função em duas dimensões descrita em termos de X.

A Figura 6 mostra o resultado do comando *plot2d*.

Figura 6 - `plot2d(-2,2,'Math.pow(x,2)')`.

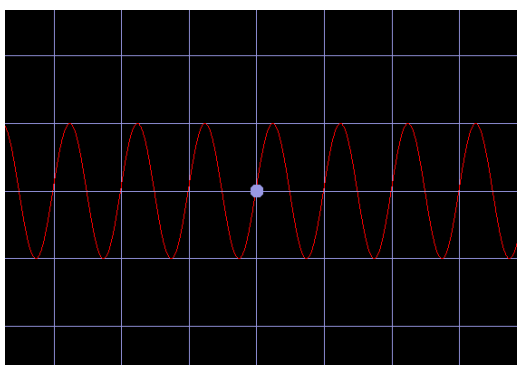


Comando *plotSenoid*

O comando *plotSenoid* foi criado para que o usuário possa visualizar o gráfico de uma onda senoidal estática ou em movimento. Para o comando ser válido, o usuário deve inserir três parâmetros. O primeiro parâmetro é o valor que representa a amplitude da onda. O segundo parâmetro é o valor que representa o comprimento da onda. O terceiro parâmetro é uma variável booleana que deve receber o valor “*true*” ou “*false*”. O valor *true* fará com que a onda seja mostrada em movimento no painel de visualização. O valor *false* fará com que a onda seja mostrada de forma estática no painel de visualização.

A Figura 7 mostra o resultado do comando *plotSenoid*.

Figura 7 - `plotSenoid(1,1,true)`.



4.2. Análise dos resultados

A intenção de utilização do software vislumbra um conjunto de ações possíveis que foram verificadas em testes realizados em disciplinas dos Cursos de Ciência da Computação, Engenharia Química e Engenharia Ambiental da Faculdade Municipal "Prof. Franco Motoro". A saber:

1. inovar no ensino de matemática, trazendo subsídios para que aluno consiga visualizar situações muitas vezes difícil de abstrair. Trata-se de um avanço em relação ao ensino tradicional, que muitas vezes isola o aluno, não permitindo o aprendizado daquele que possui maiores dificuldades lógicas e matemáticas;
2. promover um distanciamento entre a tendência tradicional de controlar o progresso do aprendizado em virtude da promoção da autonomia pessoal;
3. auxiliar o aluno desenvolver atitudes, habilidades e comportamentos destinados à auto regulação do processo de aprendizagem;
4. permitir com que a instrução seja relevante para o estudante, criando um contexto educacional capaz de estimular a autonomia do aprendiz;
5. promover atividades que levem o estudante a assumir crescente responsabilidade no desenvolvimento e reestruturação do seu próprio conhecimento.

Como projetos futuros a serem desenvolvidos, podem ser apontadas as seguintes melhorias:

1. Implementar novas rotinas para a visualização do comando *plot3d*, para que o gráfico do comando possa ser exibido na forma “*wireframe*” (“gráfico de arame”) ou “*flat*” (malhas de polígonos).
2. Programar chamadas CSS3D para aumentar a compatibilidade da aplicação em navegadores que não suportem chamadas WEBGL.
3. Melhorar a interface do usuário, de forma a torná-la mais intuitiva.
4. Implementar novos comandos para processar novos tipos de funções matemáticas.
5. Revisar as rotinas de síntese de imagens, para otimizar o desempenho.

5. CONSIDERAÇÕES FINAIS

Este artigo propôs o desenvolvimento de um aplicativo para auxiliar o ensino de funções matemáticas. O software proposto pode ser executado em navegadores *web* além de ser acessado pela internet em diversas plataformas.

Para alcançar este objetivo, foram utilizadas as linguagens HTML, para criar a interface com o usuário, e Javascript, para realizar os cálculos matemáticos necessários. Para criar o espaço de visualização tridimensional foram utilizadas a API WEBGL em conjunto com a biblioteca THREE.JS.

A linguagem HTML serviu perfeitamente para o propósito de se criar a interface com o usuário. Contando com uma sintaxe simples e diversos recursos de interação pré-programados, foi possível realizar a codificação da interface de forma simples e rápida.

O uso da API WEBGL, em conjunto com a biblioteca THREEJS para criar o espaço de visualização tridimensional se mostrou eficiente, tanto em desempenho quanto em facilidade de codificação. O fato da biblioteca THREE.JS possuir uma grande quantidade de



exemplos e uma documentação bem estruturada, fez com que o processo de desenvolvimento se tornasse ágil e eficiente.

Durante o desenvolvimento desse projeto não foram encontradas dificuldades que tornassem o mesmo inviável. Dessa forma, ao final do projeto é possível constatar que o desenvolvimento de aplicações interativas tridimensionais que auxiliem o ensino é plausível, possibilitando a sua expansão para além do ambiente acadêmico, atingindo também as empresas desenvolvedoras de software e as instituições de ensino em geral.

6. REFERÊNCIAS

- BARZEL, R. Physically-Based Modeling for Computer Graphics. London: Academic Press Inc., 1992.
- CALIL, Alessandro Marques; VEIGA, Janaína & CARVALHO, Carlos Vitor de Alencar. Aplicação do Software GRAPHMATICA no Ensino de Funções Polinomiais de 1º grau no 9º ano do Ensino Fundamental. REVISTA PRÁXIS. ano II, nº 4 - agosto 2010, p.17-27.
- CAMARGO, J. T. F.; BORTOLOTTI, J. A.; BARROS FILHO, J.; VERASZTO, E. V. Ferramentas gráficas para o ensino de técnicas de negociação de ações. Anais do XXXIX COBENGE: Congresso Brasileiro de Educação em Engenharia, Blumenau, SC, 2011a.
- CAMARGO, J. T. F.; BORTOLOTTI, J. A.; BARROS FILHO, J.; VERASZTO, E. V. Visualização de tendências de mercado: desenvolvimento de uma ferramenta gráfica a partir de software livre. Anais do XXXIX COBENGE: Congresso Brasileiro de Educação em Engenharia, Blumenau, SC, 2011b.
- FOLEY, J. D. e outros. Introduction to Computer Graphics. New York: Addison-Wesley Publishing Company Inc., 1994.
- HETEM JUNIOR, A. Computação Gráfica. Rio de Janeiro: LTC, 2006 (Coleção Fundamentos da Informática).
- PARENT, R. Computer Animation – Algorithms and techniques; Morgan Kaufmann Publishers; 2002.
- MOTA, G. A.; LUQUE, M.; VERASZTO, E. V.; CAMARGO, J. T. F. Desenvolvimento de uma animação gráfica para auxiliar o ensino das Leis de Newton. Anais do XX SNEF: Simpósio Nacional do Ensino de Física, São Paulo, SP, 2013.
- RODRIGUES, N. R. N. M. Alunos Ingressantes nas Engenharias e a aprendizagem Básica em Matemática. Anais do XL COBENGE 2012: O engenheiro professor e o desafio de educar. Belém-PA. 2012.
- ROGERS, D. F. Procedural Elements for Computer Graphics. New York: McGraw-Hill, 1985.
- ROGERS, D. F. & ADAMS, J. A. Mathematical Elements for Computer Graphics. 2nd Ed. New York: McGraw-Hill, 1990.
- ROJAS, A.; RITTO, A.C.A & BARBOSA, A.C.C. O software livre para o ensino da matemática em instituições de ensino superior – uma tecnologia social. Cadernos do IME: Série Informática: Vol.25: Julho de 2008, p.15-26.
- SANTOS, J. S.; CAMARGO, J. T. F.; VERASZTO, E. V. Desenvolvimento de simulações educativas aplicadas a dispositivos portáteis: um exemplo voltado ao ensino de física. Anais do XLI COBENGE: Congresso Brasileiro de Educação em Engenharia, Gramado-RS, 2013.



SCHNEIDER, P. J. et al. Geometric Tools for Computer Graphics. Morgan Kaufmann Publishers; 2003.

VALENTE, J. A. Diferentes usos do Computador na Educação. Disponível em: <http://ffalm.br/gied/site/artigos/diferentesusoscomputador.pdf>. Acessado em: 5 nov. 2013.

VERASZTO, E. V.; CAMARGO, J. T. F.; BARROS FILHO, J.; GARCIA, F. G.; AMARAL, S. F.; BORTOLOTTI, J. A. Estrategias para el desarrollo de contenido educativo 3D. La Revista Icono 14: Revista de Comunicación y Nuevas Tecnologías, v. 1, p. 198-212, Madrid, España, 2012.

VERASZTO, E. V.; CAMARGO, J. T. F.; LOPES, J. C.; SANTOS, J. S. Ensino da física e recursos tecnológicos: uma proposta do uso da computação gráfica. Anais do XX SNEF: Simpósio Nacional do Ensino de Física, São Paulo, SP, 2013.

VISUALIZATION OF MATHEMATICAL FUNCTIONS IN REAL TIME: AN APPROACH TO SUPPORT THE TEACHING OF MATHEMATICS TO BEGINNER STUDENTS IN ENGINEERING

Abstract: *A recurrent problem in engineering education lies in the difficulty of the new students in manipulating mathematical functions. This difficulty often turns out to limit or compromise the learning of mathematical foundations essential to the formation of the engineer in his early years of study. Thus, this paper presents a computational strategy to support the visualization of mathematical functions, in order to facilitate the teaching of this discipline. The software developed for this approach has open source, easy to maintain and upgrade and can be adapted by their own users according to their specific needs. It is quite intuitive, running in web browsers, and can be accessed via internet on various platforms. The applicative was created with the programming languages HTML and Javascript, together with the WebGL API. The software developed in the project is able to receive user data and functions, perform the required calculations and show the results in three dimensions through a visualization interface, thus helping users to better understand the behavior of mathematical functions.*

Keywords: *Educational software, Graphical visualization, Engineering education*