

Criando conteúdos educacionais digitais interativos em matemática e estatística com o uso integrado de tecnologias: GeoGebra, JavaView, HTML, CSS, MathML e JavaScript

Creating interactive digital educational contents in mathematics and statistics with the integrated use of technologies: GeoGebra, JavaView, HTML, CSS, MathML and JavaScript

HUMBERTO JOSÉ BORTOLOSSI¹

Resumo

Existem várias maneiras de se construir conteúdos digitais educacionais interativos. Ao se fazer uma escolha por uma determinada tecnologia de desenvolvimento, questões como o tempo, a facilidade de atualização, a portabilidade, a curva de aprendizagem e o poder de expressão das ferramentas de programação devem ser consideradas. O objetivo desse artigo é mostrar, através de vários exemplos, como o uso integrado do GeoGebra com outras tecnologias gratuitas (JavaView, HTML, CSS, MathML e JavaScript) se coloca como uma excelente opção nesse cenário. Esperamos que o nosso relato de experiência seja útil para outras equipes preocupadas com custos, tempo e qualidade didática no desenvolvimento de seus softwares e sistemas online de ensino.

Palavras-chave: GeoGebra; objetos de aprendizagem; tecnologias de desenvolvimento de software.

Abstract

There are several ways to build interactive digital educational contents. When making a choice for a particular technology of development, issues such as time, ease of change, portability, learning curve and power of expression of the programming tools should be taken into account. The aim of this paper is to show, through several examples, how the integrated use of GeoGebra with other free technologies (JavaView, HTML, CSS, MathML and JavaScript) stands up as an excellent option in this scenario. We hope that our experience report will be useful for other teams concerned about cost, time and didactic quality in the development of their applications and online teaching systems.

Keywords: GeoGebra; learning objects; software development technologies.

Introdução

No desenvolvimento de um *software* educacional, diversos aspectos devem ser considerados. Entre eles: as especificidades do conteúdo a ser explorado, o *design* pedagógico, o *design* gráfico, a interface com o usuário, os custos de produção e o tempo de desenvolvimento.

Enquanto que o conteúdo e a didática certamente são quesitos muito importantes, aspectos técnicos também devem merecer igual atenção. Afinal, a escolha das

¹ Universidade Federal Fluminense – <http://www.professores.uff.br/hjbortol/>

tecnologias empregadas no desenvolvimento tem influência direta nos recursos disponíveis para a construção das componentes instrucionais que irão compor o *software*. De fato, dependendo das tecnologias escolhidas, certos instrumentos didáticos podem ser mais difíceis de se implementar (geometria espacial e a própria notação matemática bidimensional são exemplos clássicos dessa situação). Mais ainda: a escolha das tecnologias tem um impacto direto nos custos e no tempo de desenvolvimento do projeto (aqui vale lembrar que o próprio *site* oficial do GeoGebra mantém um fórum dedicado exclusivamente aos aspectos tecnológicos do programa²).

O objetivo desse artigo é mostrar, através de vários exemplos, como o uso integrado do GeoGebra com outras tecnologias gratuitas (JavaView, HTML, CSS, MathML e JavaScript) se coloca como uma excelente opção nesse cenário. A escolha desse conjunto de tecnologias e os exemplos apresentados são resultados da experiência do autor com o desenvolvimento de *softwares* educacionais em matemática e estatística pelo projeto de pesquisa “Conteúdos Digitais para o Ensino e Aprendizagem da Matemática do Ensino Médio” (<http://www.uff.br/cdme/>), com financiamento do MEC/MCT/FNDE pelo edital de apoio à produção de conteúdos educacionais digitais multimídia para o ensino médio de 2007 (Projeto CONDIGITAL MEC-MCT).

1. Breve descrição das tecnologias envolvidas

Nesta seção faremos uma breve descrição das tecnologias que serão usadas nos exemplos seguintes, com indicações de *download* e referências onde o leitor poderá encontrar descrições mais aprofundadas.

GeoGebra

Criado pelo austríaco Markus Hohenwater, o GeoGebra (<http://www.geogebra.org>) é um *software* de matemática dinâmica gratuito multiplataforma que integra recursos gráficos, numéricos, simbólicos e estatísticos. Sua interface gráfica permite que objetos sejam construídos e manipulados visualmente ou através de comandos de programação.

JavaView

Criado pelos alemães Konrad Polthier, Klaus Hildebrandt, Felix Kälberer, Matthias Nieser e Ulrich Reitebuch, o JavaView (<http://www.javaview.de>) é um *software* de

² <http://www.geogebra.org/forum/viewforum.php?f=20>

matemática gratuito multiplataforma que permite criar, modificar e visualizar objetos tridimensionais. Sua biblioteca numérica oferece vários recursos sofisticados em geometria espacial, geometria computacional e geometria diferencial (cálculo de volumes, seções planas, planificações, cálculo de curvaturas, geodésicas, etc.).

HTML

HTML (acrônimo para *HyperText Markup Language*) é uma linguagem de marcação usada para especificar a estrutura de um documento. Um navegador de internet (*web browser*) nada mais é do que um *software* que interpreta estas marcações de estrutura e, então, constrói uma página *web* com recursos de hipermídia com os quais o usuário pode interagir. Para mais informações, recomendamos o livro (BROOKS, 2007).

CSS

CSS (acrônimo para *Cascading Style Sheets*) é uma linguagem de estilo usada para especificar a aparência (*layout*, cor e fonte) dos vários elementos de um documento que foi definido por uma linguagem de marcação (como a linguagem HTML). Ela foi criada com o objetivo de separar a estrutura do documento de sua aparência. Para mais informações, recomendamos o livro (GRANNEL, 2007).

MathML

MathML (acrônimo para *Mathematical Markup Language*) é uma linguagem de marcação usada para especificar a estrutura de expressões matemáticas em um documento. Seu objetivo principal é fazer para a matemática o que a linguagem HTML faz para textos e imagens. Para mais informações, recomendamos o livro (SANDHU, 2003).

JavaScript

JavaScript é uma linguagem de programação interpretada disponível nos navegadores de internet. Sua sintaxe é parecida com a da linguagem C. A linguagem JavaScript disponibiliza uma série de recursos de interface gráfica (tais como botões, campos de entrada e seletores), viabilizando assim a construção de páginas *web* mais interativas. Mais ainda, a linguagem JavaScript permite modificar e integrar, de forma dinâmica, o conteúdo e a aparência dos vários elementos que compõem o documento.

Resumindo: na construção dos *softwares* que apresentaremos a seguir, os recursos matemáticos foram implementados com o auxílio do GeoGebra e do JavaView, a

estrutura foi elaborada usando-se HTML e MathML, a aparência com a linguagem CSS, a interatividade e dinamismo com a linguagem JavaScript.

2. Portabilidade

As tecnologias descritas na seção anterior são todas portáteis, isto é, as páginas *web* interativas e dinâmicas criadas com estas tecnologias podem ser executadas nos principais navegadores (Firefox, Google Chrome, Internet Explorer, Opera) em diferentes sistemas operacionais (Linux, Windows e Mac OS). As atividades podem ser armazenadas em um servidor para uso online pela internet ou, alternativamente, elas podem ser baixadas ou disponibilizadas em CDs e *pen drives* para uso off-line (isto é, em um navegador, mas sem a necessidade de uma conexão com a internet).

3. Exemplo: superfícies e sólidos de revolução

A Parte 5 do *software* “Superfícies e Sólidos de Revolução” (disponível no endereço <http://www.uff.br/cdme/ssr/>) oferece um ambiente onde o usuário pode modelar interativamente um objeto 3D que é uma superfície de revolução a partir de uma foto de seu perfil (Figura 1).

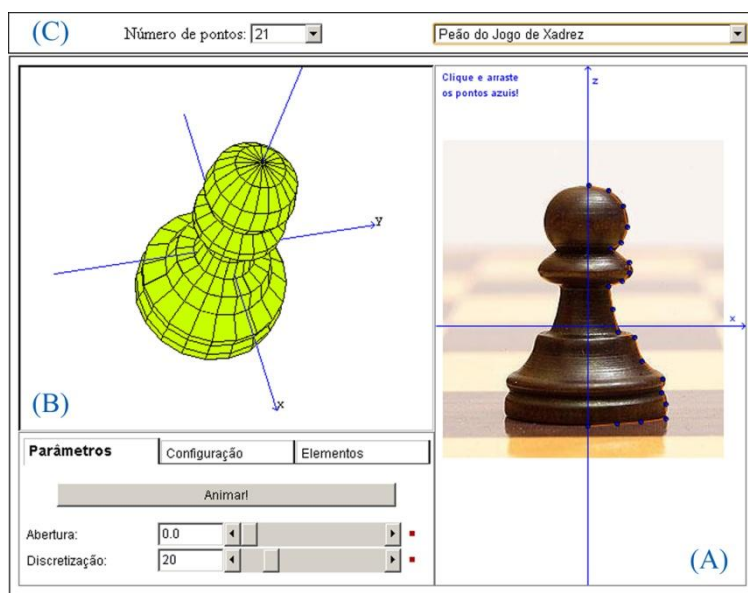


FIGURA 1: Modelando um objeto 3D com uma superfície de revolução

FONTE: <http://www.uff.br/cdme/ssr/ssr-html/ssr-g-br.html>

O objetivo da atividade é promover o entendimento dos elementos geométricos principais que constituem uma superfície de revolução (sua curva geratriz e seu eixo de revolução) e, também, promover a percepção da presença e do uso de superfícies de

revolução no dia a dia.

Aqui temos um excelente exemplo de como a combinação das tecnologias descritas na Seção 1 podem gerar um *software* interativo com alto poder de expressão. O *software* é composto de três áreas principais (em um documento HTML) que se comunicam entre si usando a linguagem JavaScript. A Parte (A) é feita no GeoGebra, usando o recurso que permite inserir uma figura de fundo. Nesta área também está disponível um caminho poligonal. O usuário pode clicar e arrastar os vértices desse caminho de forma a ajustá-los ao perfil da figura. Ao fazê-lo, as coordenadas dos vértices são transmitidas para a Parte (B) feita no JavaView. Na Parte (B), o usuário pode interagir com o modelo 3D: girá-lo, ampliá-lo, reduzi-lo, transladá-lo e abri-lo. O usuário também pode configurar cores, efeitos de iluminação, transparência e discretização. Na Parte (C), escrita em JavaScript, temos dois elementos de interface gráfica, mais precisamente, dois menus suspensos onde o usuário pode escolher o número de vértices do caminho poligonal e a foto que será inserida no GeoGebra.

4. Exemplo: rodas da fortuna

O *software* “Rodas da Fortuna” (disponível no endereço <http://www.uff.br/cdme/rdf/>) oferece seis aplicativos orientados para a simulação de experimentos aleatórios no computador (geração de números pseudoaleatórios), com ênfase em probabilidade geométrica.

O aplicativo da Parte 1 simula uma roda da fortuna onde o espaço amostral e as probabilidades podem ser modificadas, permitindo a simulação de experimentos clássicos como o lançamento de uma moeda e o lançamento de um dado (Figura 2). Estão disponíveis três geradores de números pseudoaleatórios: o gerador congruente linear (GCL), o *Mersenne Twister* e o *Mersenne Twister 53*.

O aplicativo da Parte 4 é um experimento em probabilidade geométrica (Figura 3). Usando geradores pseudoaleatórios, o aplicativo sorteia pares ordenados (x, y) no quadrado unitário $Q = [0, 1] \times [0, 1]$, onde as variáveis x e y estão uniformemente distribuídas no intervalo $[0, 1]$. O aplicativo conta então o número de pares ordenados sorteados em Q e o número de pares ordenados sorteados que estão no círculo de centro $P = (1/2, 1/2)$ e raio $1/2$. A partir destas contagens, o programa calcula a razão r do número de pontos no círculo pelo número total de pontos sorteados e, também, o número $r \times 4$. Como a probabilidade de um ponto sorteado Q estar no círculo de centro

P e raio 1/2 é igual a $p = (\text{área do círculo})/(\text{área do quadrado}) = [\pi \times (1/2)^2]/1 = \pi/4$ segue-se que (teoricamente) o número $r \times 4$ calculado pelo programa deve convergir para o número π .

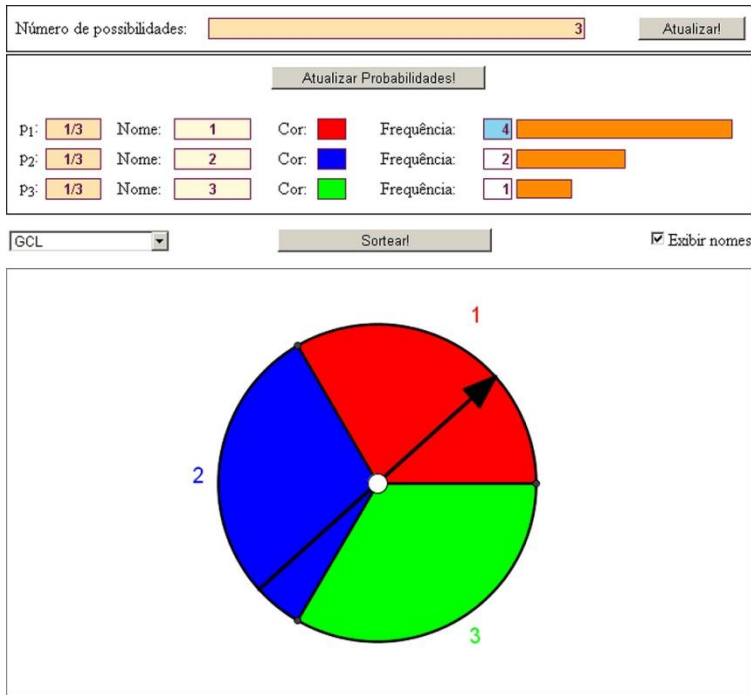


FIGURA 2: Rodas da fortuna e simulação
FONTE: <http://www.uff.br/cdme/rdf/rdf-html/rdf-c-br.html>

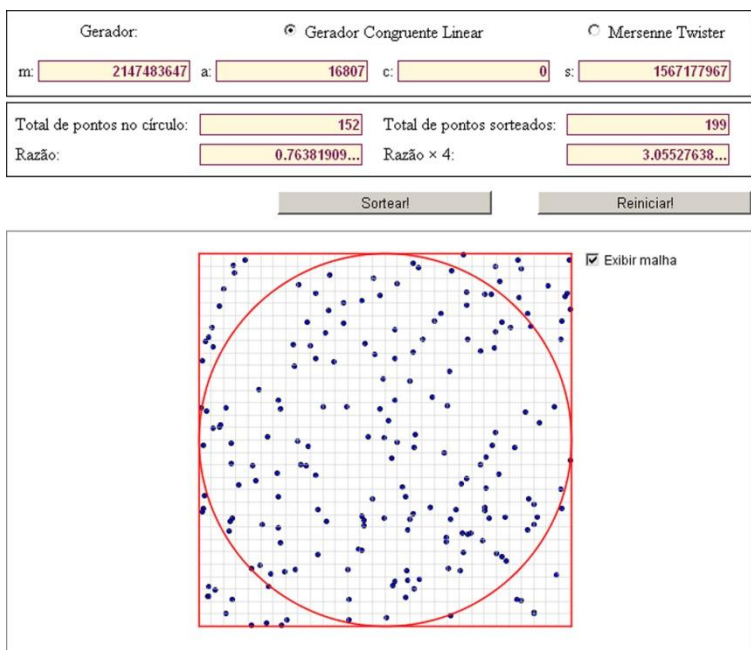


FIGURA 3: Probabilidade geométrica e a área de um círculo
FONTE: <http://www.uff.br/cdme/rdf/rdf-html/rdf-s-br.html>

O aplicativo da Parte 6 simula um caso particular do problema da agulha de Buffon

(Figura 4), proposto pelo naturalista francês George Louis Leclerc (1707-1788). Mais precisamente, usando geradores de números pseudoaleatórios, o aplicativo lança aleatoriamente uma série de agulhas sobre as ripas retangulares, calculando o número de agulhas que ficaram sobre uma reta divisória. O programa calcula a frequência relativa r (razão) do número de agulhas sobre as ripas, que deve convergir para $2/\pi$ ou, equivalentemente, $2/r$ deve convergir para π .

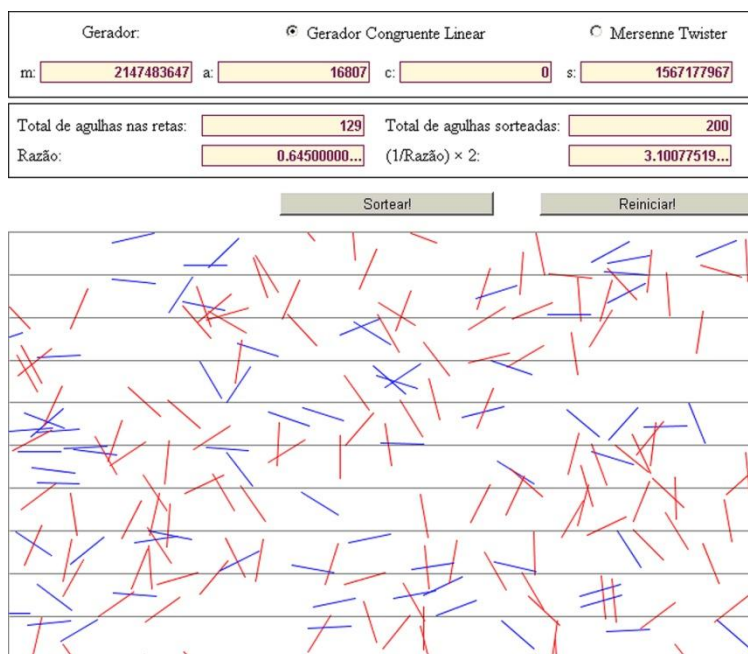


FIGURA 4: Probabilidade geométrica e o problema da agulha de Buffon
FONTE: <http://www.uff.br/cdme/rdf/rdf-html/rdf-b-br.html>

Nos três casos, os desenhos (da roda da fortuna, do círculo inscrito em um quadrado e dos pontos e das agulhas sendo lançadas) são feitos com o auxílio do GeoGebra. A interface gráfica (botões, seletores, campos de entrada, menus suspensos) é implementada em JavaScript. A aparência (incluindo a mudança de cores e o histograma do aplicativo da Parte 1) é controlada via CSS. Uma novidade aqui é o uso de sons: usando-se JavaScript (e um pequeno programa escrito em Java), foi possível incorporar no aplicativo da Parte 1 o som de uma roleta girando.

5. Exemplo: anatomia de uma função quadrática

A Parte 1 do *software* “Anatomia de Uma Função Quadrática” (disponível no endereço <http://www.uff.br/cdme/fqa/>) traz um conjunto de recursos que permitem visualizar e calcular (numericamente e simbolicamente) os elementos principais associados a uma função quadrática (Figura 5).

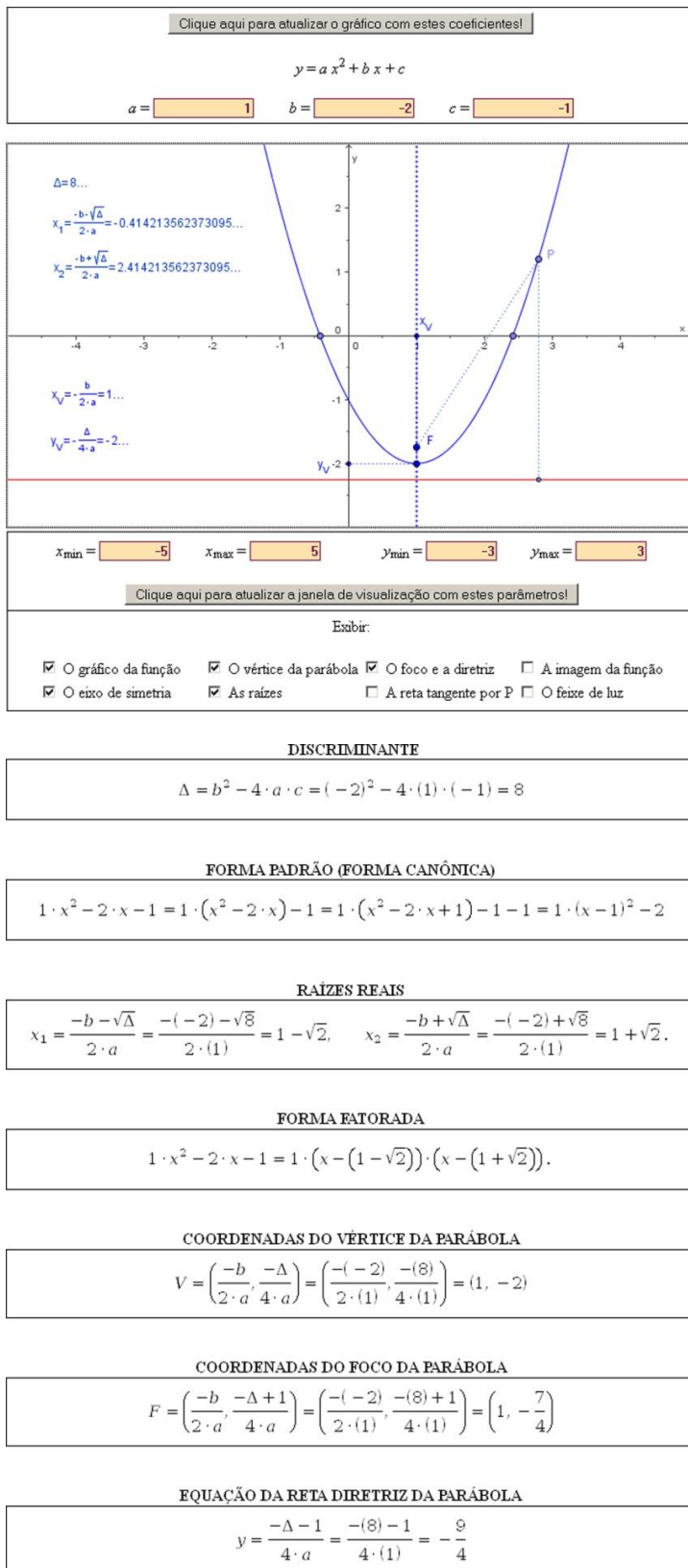


FIGURA 5: Elementos geométricos e algébricos associados a uma função quadrática

FONTE: <http://www.uff.br/cdme/rdf/rdf-html/rdf-b-br.html>

Os elementos geométricos exibidos pelo aplicativo são os seguintes: o gráfico da função quadrática, o vértice da parábola (que é gráfico da função quadrática), o foco e a diretriz

dessa parábola, a imagem da função, o eixo de simetria, as raízes reais (quando existirem), a reta tangente por um ponto do gráfico da função e o feixe de luz (um conjunto de retas paralelas ao eixo y , que incidem no gráfico da função e convergem no foco da parábola). Na parte algébrica, temos os cálculos (simbólicos e sem aproximações) do discriminante, das raízes reais (quando existirem), da forma padrão, da forma fatorada, das coordenadas dos vértices da parábola, das coordenadas dos focos da parábola e da equação da reta diretriz da parábola. Ao modificar os coeficientes da função quadrática (que nessa interface devem ser números inteiros) nos campos “ $a =$ ”, “ $b =$ ” e “ $c =$ ”, todos os elementos gráficos, simbólicos e numéricos são atualizados automaticamente.

Como nos exemplos anteriores, o *software* “Anatomia de Uma Função Quadrática” foi implementado usando-se o GeoGebra para a parte gráfica, o JavaScript para a interface gráfica e a linguagem CSS para controle da aparência. As fórmulas e expressões matemáticas são formatadas com a linguagem MathML.

Considerações finais

Como ilustram os exemplos apresentados nesse artigo, a combinação e uso dos melhores recursos do GeoGebra, JavaView, HTML, CSS, MathML e JavaScript permitem, em um ambiente de desenvolvimento rápido, criar, a baixo custo, *softwares* matemáticos expressivos, portáteis e extensíveis. Com as novas versões dessas tecnologias (GeoGebra versão 5, JavaView versão 4, HTML5 e CSS3), mais e mais recursos ficam disponíveis, possibilitando assim a construção de atividades mais dinâmicas, mais interativas e mais didáticas.

Por fim, esperamos que o relato de experiência feito aqui seja útil para outras equipes preocupadas com custos, tempo e qualidade didática no desenvolvimento de seus *softwares* e sistemas online de ensino.

Referências

- BROOKS, D. R. (2007). *An Introduction to HTML and JavaScript for Scientists and Engineers*. London: Springer-Verlag.
- GRANNELL, C. (2007). *The Essential Guide to CSS and HTML Web Design*. New York: APress.
- SANDHU, P. (2003). *The MathML Handbook*. Hingham: Charles River Media.