



COBENGE 2005

XXXIII - Congresso Brasileiro de Ensino de Engenharia

"Promovendo e valorizando a engenharia em um cenário de constantes mudanças"

12 a 15 de setembro - Campina Grande Pb

Promoção/Organização: ABENGE/UFCG-UFPE

POSSIBILIDADES PARA A INTEGRAÇÃO ENTRE MATEMÁTICA E PROGRAMAÇÃO

Tânia Martins Preto - tania@inf.ufpr.br

Departamento de Informática – Setor de Ciências Exatas

Universidade Federal do Paraná - UFPR

Centro Politécnico - Caixa Postal 19011 - Jardim das Américas

81531-980, Curitiba, Paraná.

André Luiz Pires Guedes - andré@inf.ufpr.br

Sergio Scheer - scheer@ufpr.br

Curso de Pós-Graduação em Métodos Numéricos em Engenharia

Centro de Estudos de Engenharia Civil -CESEC

Universidade Federal do Paraná –UFPR

Centro Politécnico - Caixa Postal 19011 - Jardim das Américas

81531-990, Curitiba, Paraná.

Viviana Cocco Mariani - viviana.mariani@pucpr.br

Programa de Pós-Graduação em Engenharia Mecânica – PPGEM

Pontifícia Universidade Católica do Paraná – PUCPR

Rua Imaculada Conceição, 1155- Prado Velho

81611-970, Curitiba, Paraná.

Resumo: *Este artigo apresenta uma análise de dois tipos de abordagens de linguagens de programação usadas a fim de auxiliar e melhorar o ensino das disciplinas de matemática nos Cursos de Engenharia e Computação. Na primeira abordagem, a linguagem é voltada para aplicações matemáticas e não necessita do uso de técnicas de programação avançadas, possibilitando que o aluno se concentre na tarefa de elaboração da solução. Na segunda abordagem, utiliza-se uma linguagem voltada para propósitos gerais, exigindo maiores conhecimentos de programação, além de algoritmos e estruturas de dados, favorecendo assim a execução de trabalhos interdisciplinares envolvendo principalmente matemática e programação. Este artigo analisa os dois tipos de abordagens e suas contribuições na aprendizagem.*

Palavras-chaves: Programação, Matemática, Ensino, Interdisciplinaridade

1. INTRODUÇÃO

As disciplinas que enfocam conteúdos de Matemática possuem uma grande carga horária nos Cursos de Engenharia e Computação. No ensino dessas disciplinas observa-se a ocorrência de dificuldades por parte dos alunos quanto ao entendimento de alguns conteúdos, falta de percepção da sua importância na formação e o não entendimento da relação entre a Matemática e as outras disciplinas. Algumas características desejáveis dos egressos desses cursos [CR1999], [CNE/CES, 2001] são as seguintes: Capacidade para aplicar seus conhecimentos (matemáticos, científicos, tecnológicos e instrumentais) na busca de soluções em diferentes áreas; identificar, formular e resolver problemas de engenharia; modelagem e especificação de soluções computacionais para diversos tipos de problemas; desenvolver e/ou utilizar novas ferramentas e técnicas e também atuar em equipes multidisciplinares.

Este trabalho apresenta algumas abordagens para auxiliar e melhorar o ensino das disciplinas de Matemática nos Cursos de Engenharia e Computação, face à grande quantidade de ferramentas computacionais que existem, diferindo entre si na forma de utilização, objetivos, pré-requisitos para sua utilização e possibilidade de integração com outros assuntos e disciplinas.

Ambas abordagens fazem uso de um ambiente de programação através de uma linguagem de programação onde o aluno vai elaborar soluções para problemas matemáticos propostos.

A primeira abordagem trata do uso de linguagens destinadas a aplicações de Matemática, dentre outras, apresentando diversas funcionalidades para tal, exigindo que o aluno tenha conhecimentos prévios de suas funcionalidades e também de conhecimentos básicos de programação. Estas linguagens geralmente são simples, mas envolvem algum tipo de treinamento para a sua utilização. A segunda abordagem baseia-se no uso de linguagens de programação de propósitos gerais que não apresentam funcionalidades específicas, no entanto, exigem do aluno conhecimentos mais profundos sobre conceitos de programação, algoritmos e estruturas de dados, propiciando a realização de atividades interdisciplinares e a prática de conceitos presentes nas disciplinas de programação, algoritmos e estruturas de dados, além de se preocupar com a solução do problema matemático.

Vale ressaltar que os dois tipos de abordagem permitem que problemas de diferentes níveis de complexidade sejam resolvidos.

Para analisar essas abordagens será apresentada uma comparação entre a programação com recursos do Matlab e a programação com recursos em linguagem C. Existem outras abordagens que não serão analisadas aqui e que envolvem o uso de programas destinados a complementar o trabalho feito em sala de aula através da apresentação de conceitos, exploração da informação, exercícios práticos e outros. A resolução de problemas cuja solução envolve o uso de conceitos matemáticos e também assuntos abordados em outras disciplinas, auxilia a compreensão da relação entre a matemática com as outras áreas e disciplinas, além de auxiliar o entendimento da importância da matemática na formação profissional do aluno.

Cada tipo de abordagem tem suas características e vantagens contribuindo de diferentes formas na aprendizagem de Matemática, sendo que sua utilização também está condicionada aos objetivos de cada Curso. Este artigo apresenta uma breve descrição sobre o uso do computador no ensino e concentra-se em fazer um estudo comparativo entre as duas abordagens, onde o aluno desenvolve seus próprios aplicativos. A partir destes é possível executar testes a fim de verificar se o trabalho desenvolvido atende ao problema inicialmente proposto.

2. O USO DO COMPUTADOR E O ENSINO DE MATEMÁTICA

2.1 Conceituação

O uso do computador na educação pode ser feito de várias formas, através do uso de programas originalmente construídos para fins educacionais e também de programas de propósitos gerais tais como os ambientes integrados de programação e os aplicativos.

O professor é responsável por conduzir a prática educativa, devendo para isto se atualizar constantemente a fim de conhecer as possibilidades e dominar os recursos computacionais existentes [Kampff, 2003].

Segundo [Estevan, 1990], os principais objetivos do uso do computador na área educacional são preparar os estudantes para trabalharem com os computadores em suas atividades profissionais e utilizá-lo nas diferentes disciplinas como ferramenta poderosa para incentivar o aprendizado de forma criativa. Existem várias classificações ou taxonomias para programas educacionais e também para a forma de uso do computadores no processo de ensino-aprendizagem.

Segundo [Lee, 1998] existem duas abordagens. Na primeira o computador é visto como tutor, desempenhando o papel de professor, isto é lançando perguntas, analisando respostas e a partir destas, determinando o que deve ser apresentado na seqüência. Recursos gráficos, animações e diretrizes diversas podem ser usados para facilitar a aprendizagem. Na abordagem do computador como ferramenta, este é utilizado para adquirir e manipular informações, realizando tarefas diversas como elaboração de texto, manipulação de banco de dados e planilhas, construção e manipulação de gráficos, sistemas de autoria e construção de outros aplicativos.

Em [Valente, 1993] são apresentadas outras duas abordagens: Na primeira, o computador é visto como máquina de ensinar através de programas tutoriais baseados no conceito de instrução programada, programas de exercício e prática, jogos educacionais com exploração dirigida e simulações envolvendo a criação de modelos dinâmicos, permitindo a exploração de situações fictícias, de risco, experimentos complicados, caros ou demorados. Na segunda abordagem, o computador é visto como ferramenta, abrangendo programas aplicativos destinados a automatizar tarefas como edição de textos, planilhas, banco de dados, construção de gráficos e sistemas de autoria. Esta abordagem envolve a resolução de problemas e a elaboração de projetos através de linguagens de programação, onde o aluno faz a descrição formal de sua resolução, observando a execução do programa e analisando seus resultados, incluindo também os programas de controle de processo, onde o aluno analisa determinados processos, a forma de controlá-los, testa situações e verifica o resultado de suas decisões.

Uma outra abordagem existente corresponde aos Sistemas Tutoriais Inteligentes (STIs) que utilizam técnicas de inteligência artificial, teoria de agentes, aspectos pedagógicos, dentre outros, permitindo que seja feito acompanhamento individualizado e tomada de decisões sobre estratégias de ensino.

Algumas das taxonomias citadas, embora antigas ainda são muito utilizadas. Alguns programas podem se encaixar em mais de uma categoria.

2.2 O Ensino de Matemática e o uso do Computador

A maior parte do conteúdo de matemática está concentrado nos primeiros anos dos cursos de Engenharia e Computação. Alguns problemas existentes no ensino dessas disciplinas poderiam ser contornados com o uso da tecnologia [D'Ambrosio, 1999].

O ensino de matemática com conceitos apresentados como verdades absolutas, não favorecem o processo de ensino/aprendizagem e nem estimulam a criatividade, tornando a aprendizagem de Matemática uma atividade difícil e desinteressante causando baixo

rendimento por parte dos alunos. Segundo [Lima, 1995], o ensino da Matemática deve abranger os seguintes componentes: conceituação (formulação de definições e conexões entre conceitos); manipulação (prática de exercícios) e aplicações (uso de teorias para resolução de problemas diversos). Estes estão detalhados em [Preto, 2004].

O uso do computador como ferramenta de apoio ao ensino tem auxiliado na melhoria da qualidade do ensino e contornado alguns dos problemas relacionados ao ensino de Matemática e aquisição de conceitos. Existem muitas ferramentas computacionais para apoiar o ensino de matemática e pode-se estabelecer uma rápida relação entre alguns tipos de ferramentas de acordo com as taxonomias citadas e as componentes do ensino de Matemática: (1) Ferramentas de apoio à conceituação e sistemas de representação: programas tutoriais, jogos e simulações. (2) Ferramentas de apoio à manipulação: programas de exercício e prática, jogos e simulações também podem atender à finalidade de manipulação. (3) Ferramentas de apoio às aplicações e situações de uso: resolução de problemas através do uso de linguagens de programação e programas de controle de processo. Os sistemas tutoriais inteligentes podem atender a todas as componentes, de acordo com a sua concepção, que pode ter grandes variações.

Na abordagem da pedagogia construtivista, defende-se a necessidade de atividades que propiciem o desenvolvimento da exploração informal e investigação reflexiva, deixando o aluno tomar iniciativas e controlar a situação. Cabe ao professor propor situações que estimulem o questionamento, a colocação de problemas e a busca de solução. Desafios bem projetados e estruturados, que visam a exploração e investigação auxiliam os alunos na tarefa de se tornar aprendizes ativos.

A construção de modelos corretos sobre um assunto também pode ser feita através de exploração de expressões quantitativas (funções, taxas de variação, equações diferenciais) e possibilidade de manipulação e compreensão das relações entre as variáveis. No recurso de simulação, o programa oferece o modelo que geralmente é complexo e os alunos exploram as relações matemáticas através do dinamismo visual da representação, sem a preocupação com a dedução de relações matemáticas analiticamente. Neste recurso evidenciam-se as relações entre conceitos matemáticos e fenômenos físicos [Santarosa, 1998].

De forma geral, na perspectiva construtivista, são utilizados ambientes onde o aluno aprende através de explorações, análises e construção do conhecimento. Tais ambientes oferecem oportunidades para o aluno modelar problemas, analisar simulações, fazer experimentos, conjecturar. Em uma abordagem os alunos expressam, confrontam e refinam suas idéias, e programam o computador, sem precisar usar recursos avançados de linguagem de programação. Em outra abordagem, são necessários conhecimentos avançados de programação, favorecendo o desenvolvimento de habilidades referentes a manipulação de elementos de programação e estruturas de dados destinadas a solucionar na forma algorítmica os problemas propostos. Nesta segunda abordagem também incentiva-se a exploração, análise e construção do conhecimento, no entanto os recursos usados são diferentes da primeira abordagem.

3. O USO DE PROGRAMAÇÃO NO ENSINO DE MATEMÁTICA

O uso de programação no ensino de Matemática pode ser feito de diferentes formas, sendo que este trabalho aborda dois tipos de ferramentas (programas) para essa finalidade. A primeira abordagem trata de programas destinado a fins matemáticos, ou seja, possuem diversas funcionalidades acopladas e não exigem conhecimentos profundos de programação, apresentam facilidades para construção de gráficos e aplicações específicas de Matemática e Engenharia. Estes programas geralmente são simples, mas envolvem algum tipo de treinamento para a sua utilização. Alguns exemplos são o Maple, Matlab e Scilab.

A segunda abordagem baseia-se no uso de linguagens de programação de propósitos gerais que exigem do aluno conhecimentos mais profundos sobre conceitos de programação, algoritmos e estruturas de dados. Estas linguagens também apresentam algumas funcionalidades para aplicações matemáticas, porém em menor quantidade em relação a abordagem anterior. Vale ressaltar que o uso dessas linguagens propicia a realização de atividades interdisciplinares envolvendo matemática e de programação. Além de proporcionar a prática de conceitos relacionados às disciplinas de programação, algoritmos e estruturas de dados. Algumas linguagens deste tipo são a linguagem C, Pascal, Fortran, Visual Basic, Java e outras.

Para analisar essas abordagens será feita uma comparação entre a programação com Matlab e a programação linguagem C, apresentando características principais de cada linguagem e alguns exemplos de aplicações feitas em cada uma, ressaltando as contribuições de cada forma de implementação.

3.1 Matlab

O uso de linguagens voltadas para aplicações específicas representa uma opção muito interessante para apoiar o ensino de Matemática. Neste tipo de abordagem, são propostos trabalhos aos alunos, onde a elaboração de soluções é feita através de seqüências lógicas de comandos, obedecendo um conjunto de regras, através de uma linguagem simples e de fácil utilização.

Para o uso destas linguagens, se faz necessário um treinamento inicial, porém não são necessários conhecimentos profundos sobre conceitos de programação de computadores, algoritmos e estruturas de dados. A linguagem apresentam diversas funcionalidades prontas para serem usadas, sendo que em sua construção estão já embutidos diversos algoritmos e construções complexas. Na maior parte dos casos, o aluno deve conhecer o conjunto de funcionalidades entendendo basicamente como usar e quais tipos de informações devem ser fornecidas.

A utilização desta abordagem, proporciona uma opção de trabalho interessante pois o aluno pode resolver diversos tipos de problemas através do uso de diversas funcionalidades que são de fácil aprendizagem. O esforço do aluno concentra-se na elaboração da solução matemática e na análise dos resultados obtidos, proporcionando uma grande economia de tempo em relação a utilização de outras opções para programação como as linguagens de propósitos gerais, por exemplo, C, Pascal, Java, Fortran. O aluno tem a oportunidade de usar sua criatividade na construção de soluções através da utilização das várias funcionalidades, que auxiliam também na forma de fazer a interação do sistema com o usuário, além de proporcionar diversas opções para a manipulação de dados e sua visualização através de recursos gráficos.

O Matlab foi desenvolvido inicialmente para fazer cálculos com matrizes (*MATrix LABORatory*), porém suas funcionalidades abrangem diversas aplicações. Os comandos do Matlab são muito próximos da forma como são escritas as expressões algébricas, tornando seu uso muito simples, além de permitir que novas rotinas (funções) sejam elaboradas e também permite a utilização de comandos para controle de iteração (repetição) e comparação entre elementos (decisão) [Santos, 1998], [Hanselman, 1999].

3.2 A Linguagem C de Programação

A programação em linguagem C exige que o aluno possua conhecimentos em programação, pois não oferece um grande conjunto de funcionalidades prontas como o Matlab e as outras linguagens destinadas a aplicações matemáticas. No entanto, a linguagem C permite que uma grande quantidade de cálculos e algoritmos sejam feitos e também oferece a vantagem de gerar programas executáveis que são mais eficientes em termos de tempo de execução quando comparados aos aplicativos gerados em Matlab.

A linguagem C é uma linguagem de programação de alto nível, inicialmente projetada para o sistema operacional UNIX, sendo atualmente usada em diversas plataformas. C é de propósitos gerais pois atende a várias aplicações como sistemas comerciais, problemas que exigem visualização de dados (gráficos, imagens), problemas que exigem muitos cálculos, construção de sistemas operacionais e compiladores. A linguagem C suporta diferentes níveis de programação pois por um lado possui todas as características de uma linguagem de alto nível: fluxos de controle, estruturas de dados, procedimentos e por outro lado, permite o acesso às características do *hardware* tais como manipulação de *bits*, acesso a endereços e palavras de memória. Permite extensibilidade, ou seja, através das bibliotecas é possível “aumentar a linguagem”, implementando novas características e também permite portabilidade, ou seja, possibilita o transporte de um sistema para outro [Schildt, 1997].

Nesta abordagem, o esforço do aluno concentra-se na elaboração de algoritmos que implementam uma determinada solução matemática para um problema proposto. Para este propósito será necessário usar conceitos relacionados a construção de estruturas de dados, algoritmos e técnicas avançadas de programação.

3.3. Programação em Matlab x Linguagem C

A seguir são exibidas as características e a resolução de vários problemas nas linguagem Matlab e C [Hanselman, 1999], [Schildt, 1997], enfatizando suas semelhanças e diferenças, além de citar algumas possibilidades para integração.

Existem diversas semelhanças entre a linguagem de programação do Matlab e C, a última tabela (tabela 7) mostra um algoritmo nas duas linguagens onde pode-se perceber semelhanças na atribuição de valores a variáveis e nos comandos de controle de repetição.

Tabela 1 - Números Complexos

| A) Tarefa: Manuseio de Números Complexos. | | |
|--|-----------------------------------|--|
| Comentário: Matlab apresenta facilidades, na linguagem C é necessário criar estrutura e funções para contemplar os complexos. | | |
| Integração: Programarão de computadores (construção de funções e uso de estruturas ou registros) e Cálculos Matemáticos elementares | | |
| MatLab | Ação | C |
| <code>c1 = 1 -2i;</code> | Atribuição de imaginário | <code>struct imaginario {float re, im; }; struct imaginario c1, c2; c1.re = 1; c1.im = -2; //atribuição</code> |
| <code>z = real(c1);</code> | Z recebe a parte real de c1 | <code>float real(struct imaginario c) //funcao { return c.re; } real z, w; z = real(c1);</code> |
| <code>w = imag(c1);</code> | w recebe a parte imaginária de c1 | <code>float imag(struct imaginario c) //funcao { return c.im; } w = imag(c1);</code> |
| <code>c2 = sqrt(-3);</code> | interpretado como $0 + 1.4142 i$ | <code>struct imaginario sqrt_i(float x) //funcao sqrt_i { struct imaginario y if (x<0) { y.im = - sqrt(x); y.re = 0} else { y.re = sqrt(x); y.im = 0}; return y; } c2= sqrt_i(-3);</code> |

Tabela 2 - Vetores e Matrizes

| B) Tarefa: Criação de Vetores e Matrizes e acesso aos seus elementos: |
|---|
| Comentário: MatLab oferece uma maneira simples e intuitiva para a sua manipulação. C necessita de |

| comandos explícitos através do uso de estruturas de controle. | | |
|---|---|--|
| Integração: Programação de computadores (uso de estruturas de controle de repetição , acesso e manipulação de vetores), Cálculos Matemáticos elementares, Algebra Linear e Métodos Numéricos | | |
| MatLab | Ação | C |
| x = [0 0.1*pi 0.2*pi 0.3*pi 0.4*pi 0.5*pi] | Criação e inicialização de vetor | float x[TAM] = { 0, 0.3142,0.6283,0.9425, 1.2566, 1.5708 } ou for (i=0, inc= 0.1; i<TAM; i++, in+=0.1) x[i] = pi*inc; |
| x = -3: 3 | gera x = -3 -2 -1 0 1 2 3 | for (i=0, k= -3; i<7; i++, k++) x[i] = k; |
| x(3) | acessa o 3º elemento da lista | X[2] |
| x(1:3) | acessa do 1º ao 3º elemento de X | for (i=0; i<3; i++) printf(“%f”, x[i]); |
| x(4:end) | acessa do 4º elemento ao final | for (i=3; i<TAM; i++) printf(“%f”, x[i]); |
| x(1:2:5) | acessa do 1º ao 5º elemento com incremento de 2 em 2 | for (i=0; i<5; i+=2) printf(“%f”, x[i]); |
| a = 1:5 | cria um vetor com elementos de 1 a 5: a = [1 2 3 4 5] | for (i=0; i<5; i++) a[i] = i+1; |
| c = [b a] | C recebe a concatenação de b com a, logo c = [1 3 5 7 9 1 2 3 4 5] | for (i=0, k = 0; i < 9; i++, k++) c[k] = b[i]; for (i=0 ; i<5; i++, k++) c[k] = a[i]; |
| d = [a(1:3:5) 1 0 1] | Concatenação de parte de a com os valores restantes: 1 0 1 | for (i=0, k = 0; i < 5; i+=2, k++) d[k] = a[i]; d[k++] = 1; d[k++] = 0; d[k++] = 1; |
| g = [1 2 3 4; 5 6 7 8; 9 10 11 12] cada linha é separada por ponto e vírgula | Criação da matriz e atribuição de valores por linhas | int g[3][4] = {1,2 ,3, 4, 5, 6, 7, 8, 9, 10, 11, 12}; ou for (i=0, k=1; i<3; i++) for(j=0; j< 4; j++, k++) g[i][j] = k; |
| A(2,6) = 1 | atribuição | A[1][5] = 1; |

Tabela 3 - Operações sobre Vetores e Matrizes

| C) Tarefa: Executar operações sobre os elementos da matriz ou vetor | | |
|---|--|--|
| Comentário: MatLab oferece uma maneira simples e intuitiva para a sua manipulação. C necessita de comandos explícitos através do uso de estruturas de controle. | | |
| Integração: Programação de Computadores (manipulação de matrizes e vetores), Algebra Linear e Métodos Numéricos (utilização de matrizes para fins diversos). | | |
| MatLab | Ação | C |
| x = [0 0.3142 0.6283 0.9425 1.2566 1.5708] y = sin(x); | y é um novo vetor onde cada elemento corresponde à seno de x | for (i=0 ; i<6; i++) y[i] = sin(x[i]); |
| 2*g -1 | Multiplica todos os elementos por 2 e subtrai 1. | for (i=0, k=1; i<3; i++) for(j=0; j< 4; j++, k++) g[i] = 2*g[i] -1; |
| Soma: + Ex.: s = g + h | Somar duas matrizes de dimensões M X N | for (i=0; i<M; i++) for(j=0; j< N; j++) s[i] = g[i] + h[i]; |
| Subtração: - Ex.: s = g - h | Subtrair duas matrizes de dimensões M X N | for (i=0; i<M; i++) for(j=0; j< N; j++) s[i] = g[i] - h[i]; |
| Multiplicação elemento a elemento mul = g.*h | Multiplica cada elemento de uma pelo seu correspondente na outra. Pode-se fazer: g./h e h.\g | for (i=0; i<M; i++) for(j=0; j< N; j++) mul[i] = g[i]* h[i]; |
| Multiplicação entre matrizes: m = p * q com p(MXP),q(PXN) | Multiplicação tradicional entre matrizes de dimensões (MXP) e (PXN), gerando uma nova matriz MXN | for (i=0; i<M; i++) for(j=0; j< N; j++) for(k=0,m[i][j]=0; k< P; k++) { m[i][j] = m[i][j] + p[i][k]*q[k][j]; } |

| | | |
|------------|---|---|
| $r = g.^h$ | calcula cada elemento de g elevado a cada elemento de h | <pre> for (i=0; i<M; i++) for(j=0; j< N; j++) { r[i][j] = 1; for (k=1; k< =h[i][j]; k++) r[i][j] = r[i][j]*g[i][j]; } </pre> |
|------------|---|---|

Tabela 4 - Funcionalidades e operações diversas sobre matrizes

| D) Tarefa: Funcionalidades diversas para manipulação de matrizes | | |
|--|---|---|
| Comentário: MatLab oferece uma maneira simples e intuitiva para a sua manipulação. C necessita de comandos explícitos através do uso de estruturas de controle e funções | | |
| Integração: Programação de Computadores (manipulação de matrizes e funções). Estruturas de Dados (o problema de expansão de matrizes e a obtenção de matrizes a partir de vetores pode ser resolvido através de matrizes representadas por listas encadeadas). | | |
| Álgebra Linear e Métodos Numéricos (operações básicas com matrizes). | | |
| MatLab | Ação | C |
| Ones(3) | cria matrizes 3X3 com elementos iguais a 0 | <pre> void ones(int k) { int i, j; for (i=0; i<k; i++) for(j=0; j< k; j++) m[i][j] = 1; } </pre> |
| Expansão natural de matrizes. Seja a= [1 2 3; 4 5 6; 7 8 9] e a(2, 6)=1 | Como a coluna 6 não existe, é feita a expansão da matriz até aparecer a posição (2,6), os outros elementos são preenchidos com zeros. | <pre> for (i=0; i<3; i++) for(j=3; j< 6; j++) a[i][j] = 0; a[1][5] = 1; </pre> |
| A(:, 4) = 5 | Preenche uma coluna (4ª) com único valor (5). | <pre> for (i=0; i< M; i++) m[i][3] = 5; </pre> |
| Uma matriz pode receber partes de outra B = A(1:2, 2:3) | B recebe as duas primeiras linhas e as 2ª e 3ª colunas de A | <pre> for (i=0; i<2; i++) for(j=1; j<3; j++) for(i=0; i< M; i++) b[i][j] = a[i][j]; </pre> |
| Obtenção de vetores a partir de matrizes C = D(:) | Armazena as colunas de D no vetor C da esquerda para a direita | <pre> for (j=0, k=0; j<N; j++) for(i=0; i< M; i++) C[k++] = D[i][j]; </pre> |
| D(:, 2) | Eliminação de colunas (ex.:2ª) | <pre> for (j=1; j<N; j++) for(i=0; i< M; i++) D[i][j] = D[i][j+1]; </pre> |

Tabela 5. Operações Lógicas sobre vetores e busca de submatrizes

| E) Tarefa: Operações Lógicas sobre vetores e busca de submatrizes | | |
|---|---|--|
| Comentário: MatLab oferece uma maneira simples e intuitiva. C necessita de implementação para as mesmas tarefas. Operações lógicas são realizadas analisando valores dos vetores e geram resultados verdadeiros ou falsos (1 ou 0). | | |
| Integração: Programação de Computadores (manipulação avançada de matrizes e vetores) | | |
| Estruturas de Dados (manipulação avançada de vetores, representação de seqüências de dados e matrizes através de listas encadeadas, desenvolvimento de funções recursivas e grande exploração da locação dinâmica). | | |
| Álgebra Linear e Métodos Numéricos (operações básicas com matrizes e conjunto de valores). | | |
| MatLab | Ação | C |
| x = -3 -2 -1 0 1 2 3 abs(x) > 1 gera: 1 1 0 0 0 1 1 | Para cada elemento de x, verifica se seu módulo é maior que 1 e gerando 1(V) ou 0 (F) | <pre> for (i= 0; i<7; i++) if (abs(x[i] >1) r[i] = 1; else r[i] = 0; </pre> |
| k = find(abs(x) > 1) gera k = 1 2 6 7 | Busca de sub-matrizes: fornece índices das expressões V. | <pre> for (i= 0, k=0; i<7; i++) if (abs(x[i] >1) r[k++] = i; </pre> |
| igual = is_equal(A, C) | compara 2 matrizes ou matriz com vetor, retorna 0 ou 1 | <pre> igual = 1; for (i=0; i<M; i++) for(j=0; j< N; j++) if (A[i][j] != C[i][j]) igual = 0; </pre> |

| | | |
|---------------------------------|--|---|
| <code>C= is_member(A, B)</code> | retorna 1 para elementos que são idênticos | <pre>for (i=0; i<M; i++) for(j=0; j< N; j++) if (A[i][j] == B[i][j]) C[i][j] = 1; else C[i][j] = 0;</pre> |
| <code>R= union(A, B)</code> | Fornece a união em ordem crescente | <pre>for (i=0, k= 0; i<M; i++) R[k++]= A[i]; for (i=0; i<N; i++) R[k++]= B[i]; quicksort(R);</pre> |
| <code>C= Intersect(A, B)</code> | Fornece a interseção | <pre>for (i=0, k= 0; i<M; i++) for (j=0; j<N; j++) if (A[i] == B[j]) C[k++]= A[i];</pre> |

Tabela 6. Operações Matriciais Avançadas, resolução de Sistemas Lineares

| F) Tarefa: Operações matriciais avançadas, resolução de sistemas lineares | | |
|--|---|---|
| Comentário: MatLab oferece uma maneira simples e intuitiva para a sua manipulação. C necessita da elaboração de algoritmos que envolve. estruturas de controle, respresentação de dados e funções. | | |
| Integração: Programação de Computadores (manipulação avançada de matrizes). | | |
| Estruturas de Dados (a obtenção de vetores a partir de matrizes e de eliminação de linhas ou colunas de matrizes pode ser resolvido usando-se listas encadeadas). | | |
| Álgebra Linear e Métodos Numéricos (operações básicas com matrizes). | | |
| MatLab | Ação | C |
| <code>x = inv(A) * b</code> | Resolução de Sistema Linear através da função inversa de A usando o conceito $Ax=b$ e $x = A^{-1}$, correspondendo à <code>inv(A)</code> . | <pre>Ia= inv(A); for (i=0, k= 0; i<M; i++) for (j=0, x[i]=0; j<N; j++) x[i]+= Ia[i][j]* x[j];</pre> |
| <code>norm(A, 1)</code> | calcula norma 1 que corresponde a soma dos módulos dos elementos de A | <pre>for (i=0, N=0; i<M; i++) for (j=0; j<N; j++) if (A[i][j] >= 0) N+= A[i][j]; else N-= A[i][j];</pre> |
| <code>norm(A, 2)</code> | calcula norma 2 que corresponde a raiz da soma dos quadrados de todos os elementos | <pre>for (i=0, N=0; i<M; i++) for (j=0; j<N; j++) N+= A[i][j] * A[i][j]; N= sqrt(N);</pre> |
| Outras operações: <code>chol(A)</code> <code>cond(A)</code> <code>det(A)</code> <code>inv(A)</code> | decomposição de Cholesky numero de condição de A (verifica se uma matriz é ou não estabilizada), determinante de A inversa de A | |

Tabela 7. Algoritmo da Bissecção

| G) Tarefa: Implementação do algoritmo da Bissecção |
|--|
| Comentário: a implementação é semelhante, ambas as linguagem apresentam estruturas para controle, no entanto em Matlab algumas funcionalidades já estão prontas ou são mais fáceis, por exemplo inserção e cálculos referentes a funções matemática. |
| Integração: Programação de Computadores (estruturas de controle, organização de algoritmos e funções) e Métodos Numéricos (zeros de funções) |

| MatLab | C |
|---|--|
| <pre>function[x,y]=bisecao(funcao1,a,b,tol,max) a(1)=a; b(1)=b; ya(1)=feval(funcao1,a(1)); yb(1)=feval(funcao1,b(1)); for i=1:max x(i)=(a(i)+b(i))/2; y(i)=feval(funcao1,x(i)); if (x(i)-a(i))<tol disp(' convergiu'); break; end if y(i)==0 disp('Zero exato encontrado'); break; elseif (y(i)*ya(i)<0 a(i+1)=a(i); ya(i+1)=y(i); b(i+1)=x(i); yb(i+1)=y(i); else a(i+1)=x(i); ya(i+1)=y(i); b(i+1)=b(i); yb(i+1)=yb(i); end iter=i; end if (iter>=max) disp('Zero não encontrado') end function t=funcao1(x) t=x^2-5*x+2;</pre> | <pre>void bisecao(float a, float b, float tol, int max) { float xa[MAX], xb[MAX], x[MAX], ya{MAX}, yb[MAX], y[MAX]; char achou = 'F'; xa[0] = a; xb[0] = b; ya[0] = feval(xa[0]); yb[0] = feval(xb[0]); for (i=0; i< max && achou == 'F', i++) { x[i]=(xa[i]+xb[i])/2; y[i]=feval(xa[i]); if ((x[i]-xa[i])<tol) { printf(" convergiu"); achou = 'T'; } if (y[i] ==0) { printf(" Zero Exato Encontrado"); achou = 'T'; } if (achou == 'F') if (y[i]*ya[i] <0) { xa[i+1]=xa[i]; ya[i+1]=ya[i]; xb[i+1]=x[i]; yb[i+1]=y[i]; } else { xa[i+1]=x[i]; ya[i+1]=y[i]; xb[i+1]=xb[i]; yb[i+1]=yb[i]; } iter=i; } if (iter>=max) printf("Zero não encontrado"); } float eval(float x); { return x*x -5*x+2; }</pre> |

4. CONSIDERAÇÕES FINAIS

Observando-se no item anterior, a maior parte das rotinas matemáticas já estão prontas no Matlab, basta conhecer seu nome e sua forma de usar (parâmetros). Como vantagem do uso de linguagens como o Matlab, tem-se a possibilidade de poder concentrar esforços em aspectos como a conceituação do problema e as aplicações, contemplando os aspectos citados em LIMA (1995). Uma facilidade apresentada por tais linguagens que não foi apresentada neste trabalho, diz respeito a construção de gráficos bidimensionais e tridimensionais. Existem diversas rotinas prontas para a criação e exibição de gráficos, valendo ressaltar que a exploração de gráficos também fortalece a aquisição de conceitos matemáticos além de servir de elemento motivador.

Com o uso da linguagem C, se faz necessário a implementação das rotinas usando diversos conceitos relacionados a estruturas de dados, algoritmos, construção de funções e técnicas avançadas de programação. Como vantagem, tem-se a possibilidade de realizar a integração de disciplinas da área de matemática (álgebra linear, métodos numéricos, calculo integral) com a área de programação (programação, algoritmos e estruturas de dados). Esta integração traz benefícios como o entendimento da relação entre as disciplinas, utilização de conceitos aprendidos em uma disciplina para resolver problemas de outras, por sua vez também serve de elemento motivador a medida que o aluno enxerga a ligação entre os diversos assuntos aprendidos,

As duas abordagens apresentadas são importantes, acredita-se que a situação ideal seria que ao longo do curso ambas fossem exploradas, levando também em consideração o curso no qual o aluno está inserido, pois cada uma acrescenta formas diferentes de conhecimento, porém igualmente contribuindo para a boa qualidade da aprendizagem.

REFERENCIAS BIBLIOGRAFICAS

- CNE/CES (2001). “Diretrizes Curriculares Nacionais dos Cursos de Engenharia”. MEC. CR99 “Currículo de Referência da SBC para Cursos de Graduação em Computação e Informática”, <http://www.sbc.org.br>, 2003.
- D’AMBRÓSIO, U. “Etnomatemática, Informática, Ciências e Matemática”. Disponível em <http://vello.sites.uol.com.br/tve.html>. 1999.
- ESTEVAN, R. C. O. e SEGRE, L. M. “Aplicação de Técnicas CAIO e SUSI no contexto de desenvolvimento e avaliação de Software Educacional”. In: I SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO- I SBIE, **Anais**, 1990.
- KAMPPF, A. J.; MACHADO, J. C. e CAVEDINI, P. “Novas Tecnologias e Educação Matemática”, WEI- Workshop de Educação em Informática – Congresso da SBC, 2003.
- HANSELMAN, D. e LITTLEFIELD, B.. “Matlab 5 - Guia do Usuário”, Makron Books, 1999.
- LEE, H.D.. "Taxonomias para Software Educacionais". Monografia. Curso de Especialização em Engenharia de Software. DINF – UFPR, 1998.
- LIMA, E. L.. “Conceituação, Manipulação e Aplicações”. Revista do Professor de Matemática. No. 41. Sociedade Brasileira de Matemática. Disponível em <http://cenp.edunet.sp.gov.br/RPM>, 1995.
- PRETO, T.M.; SCHEER, S. “Integração entre Disciplinas de Matemática e Programação Através de Trabalhos Interdisciplinares”. In.: CONGRESSO BRASILEIRO DE ENSINO DE ENGENHARIA - XXXI COBENGE, **Anais**, 2004.
- SANTAROSA, L. e GRAVINA, M.A. “A Aprendizagem da Matemática em Ambientes Informatizados”. IV Congresso RIBIE, **Anais**, 1998.
- SANTOS, R.. “Introdução ao MATLAB”. Apostila on-line. Disponível em <http://www.mat.ufmg.br/~regi> (11/05/04). UFMG, 1998.
- SCHILDT, H. “**C Completo e Total**” - MAKRON Books, 1997.
- VALENTE, J.A., “Diferentes Usos do Computador na Educação”. **Em Aberto**, Brasília, ano 12, n.57, 1993.

Abstract: *This article presents an analysis of two types of programming languages in order to improve the learning of Mathematics in the Courses of Engineering and Computation. In the first type, the language is focused on mathematical applications and the students does not need to use advanced programming techniques; in the second type, the students needs a great knowledge of programming, beyond algorithms and data structures, thus favoring the interdisciplinary works involving mathematics and programming. This article analyzes the two types of programs and its contributions in the learning process.*