EMPREGO DE LINGUAGEM DE PROGRAMAÇÃO ORIENTADA A OBJETOS NA INTEGRAÇÃO NUMÉRICA DE FUNÇÃO COMPLEXA DE DOMÍNIO REAL PELO MÉTODO DOS TRAPÉZIOS.

Edson de Almeida Rego Barros – prof_edson@mackenzie.com.br
Lincoln César Zamboni – lincoln.zamboni@mackenzie.com.br
Sérgio Vicente Denser Pamboukian – sergiop@mackenzie.com.br
Universidade Presbiteriana Mackenzie – Escola de Engenharia – Depto. Propedêutica de Eng.
Rua da Consolação, 930
CEP: 01302-907 – São Paulo - SP

Resumo: *Um bom exemplo do uso de disciplinas de computação e programação, nos cursos de engenharia, dando suporte ao desenvolvimento da multidisciplinaridade, é a interação destas com o cálculo numérico. O problema considerado neste artigo é a geração de um algoritmo que seja capaz de resolver através de métodos numéricos a integral de uma função complexa de domínio real (R→C) com uma abordagem didática e pedagógica. Para tanto, revisamos as teorias relativas à integração numérica de funções reais pelo Método dos Trapézios e também números complexos, desenvolvemos e implementamos um algoritmo, orientado a objetos, nas linguagens C++ Builder e Delphi. Este trabalho foi resultado de um projeto de pesquisa desenvolvido pela Escola de Engenharia da UPM (Universidade Presbiteriana Mackenzie) e patrocinado pelo fundo do Mackpesquisa da mesma universidade. A idéia da apresentação do trabalho para o COBENGE 2005 é compartilhar com a comunidade acadêmica o conjunto de experiências acumuladas com o projeto.*

Palavras-chave: Método dos trapézios, Números complexos, Computação, Programação, Cálculo Numérico

1. INTRODUÇÃO

Nos cursos de engenharia existem várias disciplinas que se complementam na formação do aluno. Um caso típico e bem freqüente ocorre entre as disciplinas que abordam o desenvolvimento de programas de computador de cunho científico e as que trabalham com os mais variados métodos matemáticos. Na Escola de Engenharia da Universidade Presbiteriana Mackenzie (EEUPM), pode-se referenciar as disciplinas de Computação Básica e Programação I e II e a disciplina de Cálculo Numérico como exemplos desta afirmação.

Para o atual artigo, optou-se em abordar o desenvolvimento de um algoritmo que use um método numérico de cálculo da integral de uma função complexa de domínio real.

O presente estudo foi realizado durante um projeto de pesquisa para o curso de Engenharia Elétrica da EEUPM, e foi patrocinado pelo fundo Mackpesquisa da Universidade Presbiteriana Mackenzie (UPM), no ano de 2002.

Para maior fluência, no decorrer do texto omitiremos algumas citações referentes à Programação (BARROS ET AL. (2003) e BARROS ET AL. (2000)), referentes a Números Complexos (ROSA NETO (1981), SONNINO E MIRSHAWKA (1965) e VIGGIANI (1968)) e referente ao Cálculo Numérico (ZAMBONI E MONEZZI JÚNIOR (2002)).

2. CONCEITOS TEÓRICOS NECESSÁRIOS

Para o desenvolvimento de uma solução satisfatória foram necessários conhecimentos básicos em três segmentos distintos: o método numérico para o cálculo da integral; os conceitos referentes às operações com números complexos; a familiaridade com uma linguagem de programação orientada a objetos.

2.1 O método dos trapézios de integração numérica

Na disciplina de Cálculo Numérico, há a oportunidade de se ensinar muitos métodos de cálculo passíveis de serem programados em computadores e calculadoras e, em especial, com orientação a objetos.

Um tópico particularmente interessante é o assunto que trata da integração numérica, pois se pode apresentar aos alunos uma visão comparativa entre métodos como, por exemplo, Método dos Trapézios, Método de Simpson 1/3, Método de Simpson 3/8 e Método de Weddle.

Devido a sua maior abrangência, bem como sua simplicidade, optou-se em desenvolver os estudos fazendo uso do Método dos Trapézios. Este trata do somatório da área de vários trapézios de altura constante e de bases calculadas pela função em análise. Desta forma, conforme ilustrado na "Figura 1", uma função real qualquer e contínua em um intervalo pode ser aproximada por pequenos segmentos de reta e estes, por sua vez, podem aproximar a integral da função entre os limites dados.

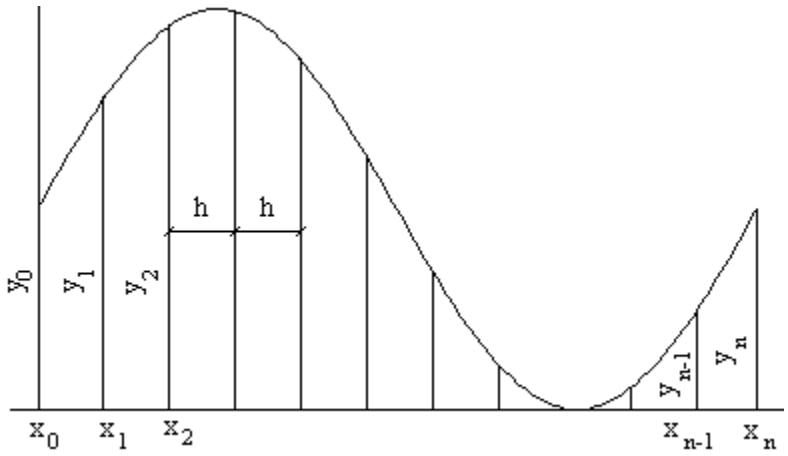

Figura 1 – Integração numérica

Algumas considerações importantes sobre a "Figura 1" são:

$$x_0 = a \tag{1}$$

$$x_n = b \tag{2}$$

$$h = (b - a) / n \tag{3}$$

$$x_i = x_0 + i \cdot h \tag{4}$$

$$y_i = f(x_i) \text{ (uma função qualquer)} \tag{5}$$

$$Integral = h \cdot (y_0 / 2 + y_n / 2 + y_1 + y_2 + \ldots + y_{n-1}) \tag{6}$$

Onde a "equação (1)" e a "equação (2)" são, respectivamente, os limites inferior e superior da integral, a "equação (3)" o passo de integração que representa a altura constante de cada um dos trapézios, a "equação (4)" é, em progressão aritmética, o conjunto dos pontos do eixo *x* usados para cálculo das bases dos trapézios, a "equação (5)" é a função integrando e a "equação (6)" é a fórmula de integração obtida pelo método dos trapézios.

Convém ressaltar que, neste método, como a precisão do resultado é proporcional à quantidade de trapézios, o computador se torna uma ferramenta de cálculo imprescindível, pois podemos dividir a área da função em milhares de trapézios (o que seria praticamente impossível de se calcular manualmente), aumentando assim a qualidade do resultado.

2.2 Operações básicas com números complexos

Uma das formas de representação geométrica de um número complexo ($z = x + y \cdot i$) é pelo posicionamento, em um plano cartesiano, de sua parte real (x) e sua parte imaginária (y). Isto pode ser visto na "Figura 2". A parte imaginária é multiplicada, algebricamente, pelo número $\sqrt{-1}$ para o qual se convencionou utilizar a letra latina *i*, conforme a "equação (7)".

$$i = \sqrt{-1} \tag{7}$$

Outra forma de representação geométrica de um número complexo ($z = r \cdot e^{i \cdot \theta} = r \angle \theta$) é pela utilização de um raio (r) e um ângulo horizontal (θ). A "Figura 2" também ilustra tal representação.

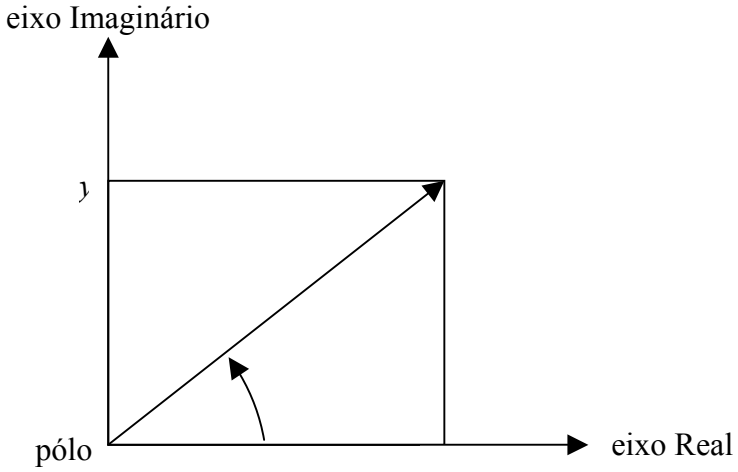

Figura 2 – Representações geométricas de um número complexo

A expressão *"$z=x+y \cdot i$"* é chamada de forma cartesiana do número complexo, *"$z=r \angle \theta$"* de forma polar e *"$z = r \cdot e^{i \cdot \theta}$"* de forma exponencial. Sendo *"x"* a componente do eixo Real (R), *"y"* a componente do eixo Imaginário (C), *"r"* a distância do valor complexo até a origem dos eixos e *"θ"* o ângulo formado entre o eixo Real (R) e a reta que liga o valor complexo à origem dos eixos.

As operações matemáticas de adição e subtração são desenvolvidas mais facilmente na forma cartesiana conforme "Tabela 1", enquanto que as operações de multiplicação e divisão são mais fáceis na forma exponencial ou também na forma polar conforme "Tabela 2".

Tabela 1 – Operações na forma cartesiana

Dados z_1 e z_2, onde: $z_1=x_1+y_1i$ e $z_2=x_2+y_2i$	
Adição: $z_{Resp} = z_1 + z_2$	Subtração: $z_{Resp} = z_1 - z_2$
$z_{Resp} = (x_1 + x_2) + (y_1 + y_2)\cdot i$	$z_{Resp} = (x_1 - x_2) + (y_1 - y_2)\cdot i$

Tabela 2 – Operações na forma exponencial e polar

Dados z_1 e z_2, onde: $z_1 = r_1 \cdot e^{i\cdot\theta_1} = r_1 < \theta_1$ e $z_2 = r_2 \cdot e^{i\cdot\theta_2} = r_2 < \theta_2$	
Multiplicação: $z_{Resp} = z_1 \cdot z_2$	Divisão: $z_{Resp} = z_1 / z_2$
$z_{Resp} = (r_1 \cdot r_2) \cdot e^{i\cdot(\theta_1+\theta_2)} = (r_1 \cdot r_2) < (\theta_1+\theta_2)$	$z_{Resp} = (r_1/r_2) \cdot e^{i\cdot(\theta_1-\theta_2)} = (r_1/r_2) < (\theta_1-\theta_2)$

A partir destes conceitos pode-se aplicar este tipo de número em vários segmentos do conhecimento humano, desde que a interpretação dos resultados seja válida.

Em alguns casos há a necessidade do cálculo de funções que, a partir de um número real (domínio), retornem um número complexo (contradomínio). Nestes casos uma interpretação geométrica e didática se faz conveniente: a função se comportaria como um ponto movendo-se em um plano formado pelos Eixos Real e Imaginário (contradomínio), e este plano transladar-se-ia ao longo do um Eixo do domínio conforme a "Figura 3".

Na "Figura 3" ilustra-se o traçado do valor obtido de uma função qualquer f: R→C, onde $z=f(p)$, R é o conjunto dos números reais (domínio) e C é o conjunto dos números complexos (contradomínio).

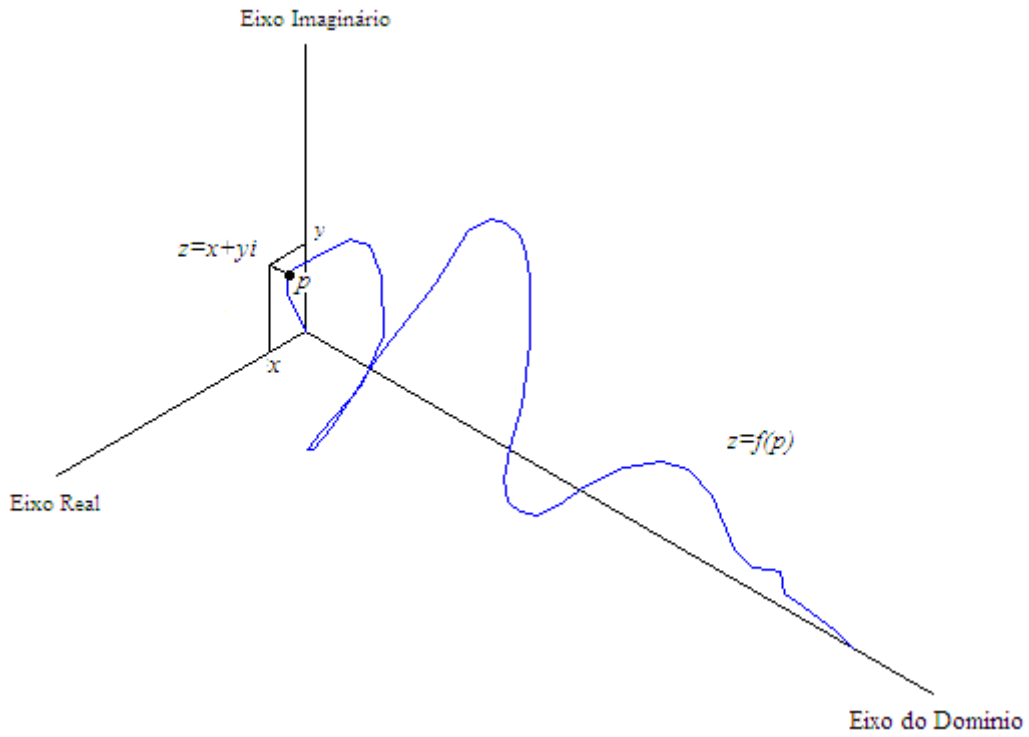

Figura 3 – Representação de f: R→C

O traçado da função pode ser mais facilmente compreendido se forem observadas as suas projeções no plano formado pelo eixo Real (R) e pelo eixo do domínio (*p*) e o plano formado pelo eixo imaginário e o eixo do domínio (*p*). Este traçado tridimensional projeta uma "sombra" em cada um dos dois planos.

A "Figura 4" evidencia didaticamente tais "sombras".

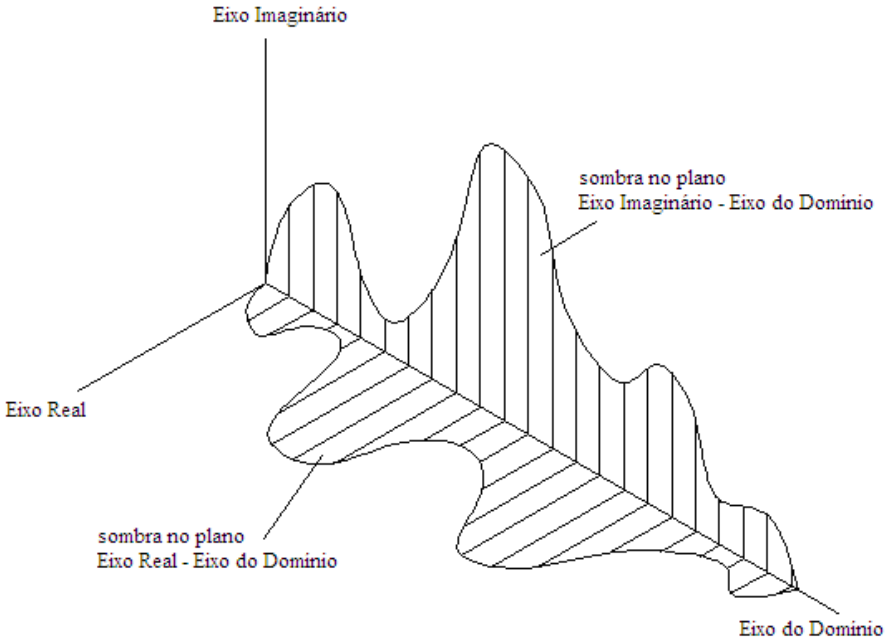

Figura 4 – "Sombras" da função

O interessante é que com essa abordagem também fica fácil entender o significado de uma integração da função que retorna valores complexos, conforme a "equação (8)".

$$I = \int_m^n f(p)dp = \int_m^n \left[f_x(p) + f_y(p) \cdot i \right] dp = \left[\int_m^n f_x(p)dp \right] + \left[\int_m^n f_y(p)dp \right] \cdot i = I_x + I_y \cdot i \quad (8)$$

Onde *I* é um valor complexo que representa o resultado da integração entre *m* e *n* na variável *p* do domínio.

Por aproximação pode-se aplicar qualquer um dos métodos numéricos de integração em cada uma das "sombras" para se obter o resultado *I* esperado.

A integração "sombra real" (plano Eixo Real – Eixo do Domínio) resultará no valor real de *I* (I_x) e a integração da "sombra imaginária" (plano Eixo Imaginário – Eixo do Domínio) resultará no valor imaginário de *I* (I_y).

A "Figura 5" mostra de forma didática e geométrica o significado da "equação (8)", uma vez que se trata de uma integração que relaciona as partes real e imaginária de uma função complexa de domínio real.

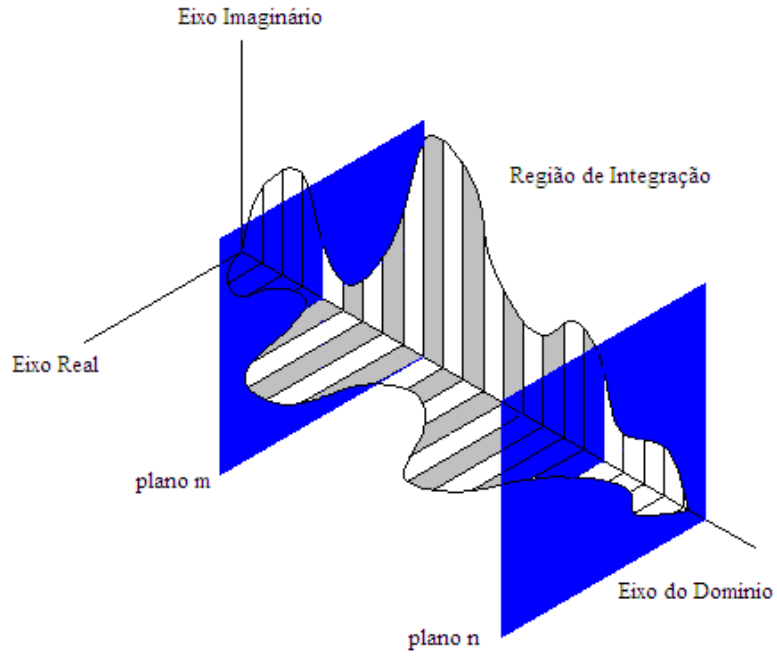

Figura 5 – Região de integração

2.3 Familiaridade com os conceitos computacionais

Para o problema em questão julgou-se interessante a criação de: uma classe específica para tratar da representação e manuseio do conceito "ponto", a classe TPonto; uma outra classe herdeira da classe TPonto específica para registro e operação do conceito de "número complexo", a classe TComplexo; e uma função para teste que retorne valores do tipo TComplexo.

A estrutura da classe do tipo TComplexo pode ser observada na "Figura 6", onde os métodos descritos destinam-se a efetuar as quatro operações básicas com um número complexo, além de permitir que um número complexo seja multiplicado por um número real.

```
class TComplex : public TPonto          //Herda características de TPonto
{
private:
protected:
public:
 TComplex CSoma (TComplex P);           //Método para calcular a soma
 TComplex CSubt (TComplex P);           //Método para calcular a subtração
 TComplex CMult (TComplex P);           //Método para calcular o produto
 TComplex CDivi (TComplex P);           //Método para calcular a divisão
 TComplex ProdNum ( double n);          //Produto de complexo por número
};
```

Figura 6 – Cabeçalho da classe TComplex, desenvolvido em C++

A computação das operações aritméticas com números complexos só é possíveis graças ao tratamento de valores cartesianos e polares desenvolvidos na classe básica TPonto, de onde se herda a segurança do encapsulamento dos dados nos dois formatos, conforme a "Figura 7".

```
class TPonto
{
 private:
 protected:
  double x,y;                                  //Coordenadas cartesianas
  double r,a;                                  //Coordenadas polares
  void CarToPol();                             //Converte de cartesiana para polar
  void PolToCar();                             //Converte de polar para cartesiana
  void Verifica();                             //Verifica 0 < a < 2*Pi
 public:
  TPonto (double xp=0.0, double yp=0.0);       //Construtor
  void SetCar(double xp, double yp);           //Captura ponto no formato cartesiano
  void SetPol(double rp, double ap);           //Captura ponto no formato polar
  double GetX();                               //Retorna X (formato cartesiano)
  double GetY();                               //Retorna Y (formato cartesiano)
  double GetR();                               //Retorna Raio (formato polar)
  double GetA();                               //Retorna Ângulo (formato polar)
  double Dist(TPonto P);                       //Distância até outro ponto
  TPonto PtoMed(TPonto P);                     //Ponto médio até outro ponto
};
```

Figura 7 – Cabeçalho da classe TPonto, desenvolvido em C++

Finalmente, para efeito de teste, torna-se necessário a programação de uma função do tipo R→C que, em termos de linguagem de programação, significa uma função onde se introduza um valor do tipo ponto flutuante e se retorna uma instância da classe TComplexo, conforme a "Figura 8".

```
//---------------------------------------------------------------------------
//Função que retorna valor complexo
//---------------------------------------------------------------------------
TComplex Funcao(double p)
{
 TComplex Aux;
 Aux.SetCar(sqrt(fabs(p)),pow(p,2));    //Exemplo de cálculo de Aux=f(p)
 return Aux;
}
```

Figura 8 – Função desenvolvida em C++

3. PROGRAMAÇÃO DO ALGORÍTMO

Uma vez dominados os conceitos teóricos básicos sobre integração numérica e números complexos, além de se ter desenvolvido o ferramental computacional necessário para a boa programação do problema, torna-se fácil codificar o algoritmo em busca do reúso das classes propostas e também da função.

Utilizando a "equação (8)", na qual os dados de entrada são os limites de integração (inferior *m* e superior *n*), e a função de interesse *f(p)*, que já foi devidamente programada, o algoritmo de integração poderá ser confinado em uma função como a que pode ser vista na "Figura 9".

```
/*-------------------------------------------------------------------------
Integração pelo método dos trapézios
Dados:
m    -> Limite inferior -> função f(p)
n    -> Limite superior -> função f(p)
d    -> Número de divisões da integral
i    -> Variável de controle da Integração Numérica
p    -> Passo de cálculo no intervalo [m,n]
Aux -> Aux = Função(m+i*p), para o cálculo dos trapézios
S   -> Somatória das áreas trapezoidais
-------------------------------------------------------------------------*/
TComplex Integral( double m,  double n)
{
 TComplex Aux,S;                  //Cria variáveis complexas
 int d=1000;                      //Define o número de divisões da integral
 double p=(double)(n-m)/d;        //Passo do cálculo
 S.SetCar(0.0,0.0);               //Zera o valor da somatória
 for(int i=0;i<=d;i++)            //Controle da rotina
 {
  Aux=Funcao(m+i*p);              //Cálculo de cada aresta f(p)
  if((i==0)||(i==d))              //Se for a 1a ou a última aresta
   Aux=Aux.ProdNum(0.5);          //usa só a metade do valor
  S=S.CSoma(Aux);                 //Acumula o valor calculado
 }
 S=S.ProdNum(p);                  //Multiplica a somatória pelo passo
 return S;
}
```

Figura 9 – Função desenvolvida em C++Builder

4. APLICAÇÃO DO ALGORÍTMO

Uma boa forma para se ter certeza de que o assunto estudado foi bem entendido pelo discente é aplicá-lo em um caso real.

Em 2002, a Profª Drª Yara Maria Botti Mendes de Oliveira liderou um projeto de pesquisa, patrocinado pelo fundo do Mackpesquisa da UPM, cujo título era: *"Distribuição de Campos Elétrico e Magnético no Corpo Humano Devido a Linhas de Transmissão de Energia (60 Hz)*".

A íntegra do resumo, do relatório interno, da pesquisa desenvolvida é apresentado abaixo:

"As linhas de transmissão de energia elétrica, na faixa de 60 Hz geram campos elétrico e magnético no ambiente. A propagação desses campos no espaço se constitui em um tipo de radiação, denominada não ionizante. A expansão das aplicações de equipamentos elétricos e eletrônicos, que produzem campos elétricos e magnéticos, nas mais variadas atividades humanas, aumenta o grau de exposição do homem a esse tipo de radiação e conseqüentemente os riscos às sua saúde. Os campos elétricos e magnéticos estão presentes em qualquer lugar e por essa razão a população encontra-se atualmente sujeita a uma maior exposição e seus possíveis efeitos. A associação entre a exposição do ser humano a esse tipo de radiação não ionizante e possíveis riscos à saúde vem sendo objeto de estudos por cientistas e pesquisadores. Alguns estudos mostraram que crianças que viviam próximo de linhas de transmissão de energia apresentaram maior incidência de um tipo de câncer (leucemia). Outros estudos, porém, não confirmaram essa relação de causa e efeito.Trata-se de um assunto bastante polêmico, que pela sua relevância, é tema de um projeto da Organização Mundial de Saúde, o EMF Project - Projeto Internacional de Campos Eletromagnéticos, iniciado em 1998 envolvendo pesquisadores do mundo inteiro, com o objetivo de verificar cientificamente se a radiação não ionizante é ou não a causa de doenças como câncer".

Os cálculos da referida pesquisa necessitavam de um aprofundado conhecimento dos conceitos pertinentes a números complexos.

Após a devida interpretação dos dados coletados na pesquisa, solicitou-se aos professores da disciplina de Computação Básica e Programação I e II que desenvolvessem um aplicativo capaz de calcular diversos casos do problema em estudo.

O software foi desenvolvido em ambiente visual de programação, Delphi da Borland, fazendo uso dos mesmos algoritmos apresentados no presente artigo.

Uma de suas tela pode ser vista na "Figura 10".

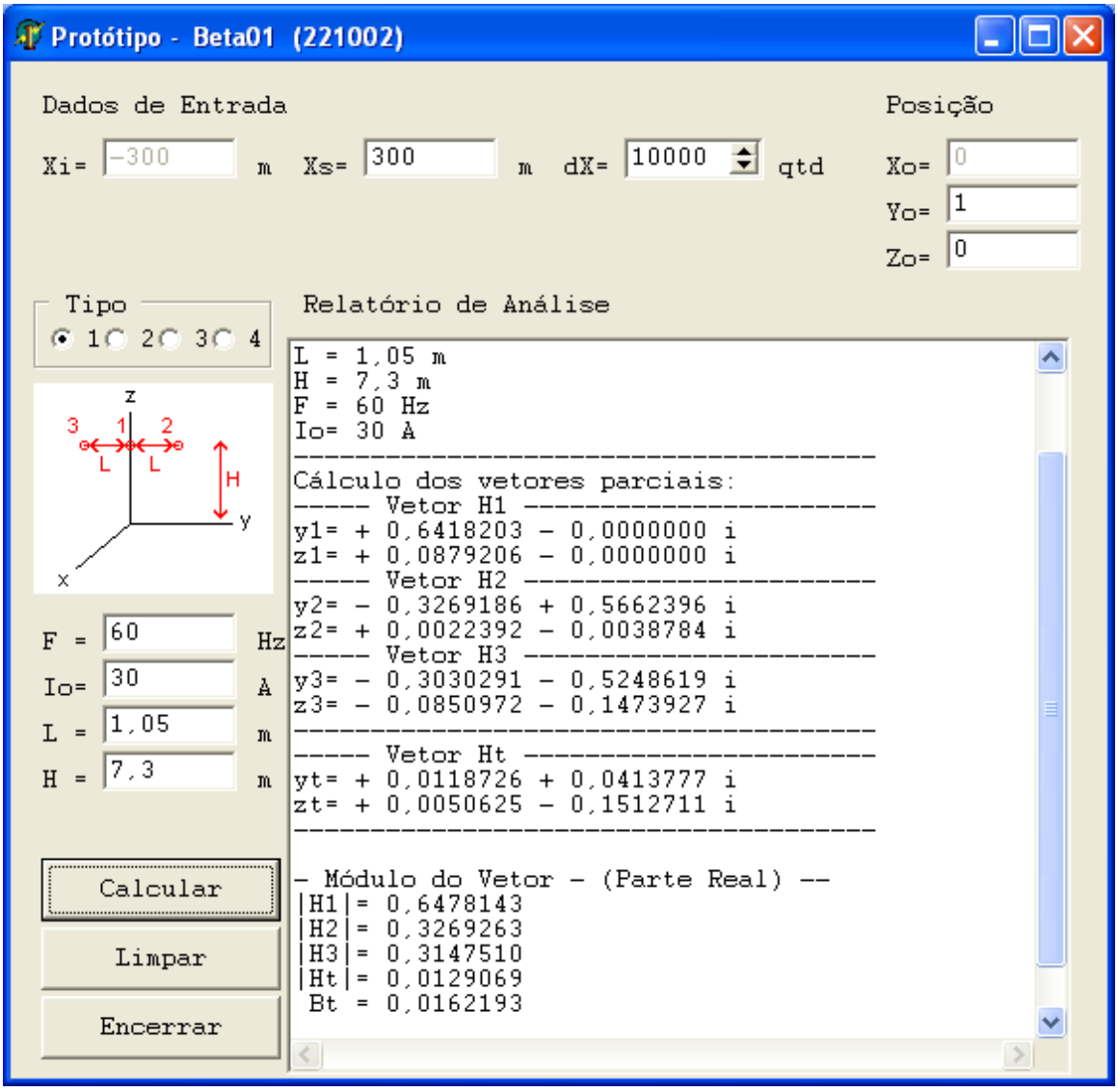

Figura 10 – Tela de aplicativo desenvolvido em Delphi

5. CONSIDERAÇÕES FINAIS

Como já foi mencionado no artigo BARROS ET AL. (2004), a nossa maior preocupação é que o atual conhecimento abordado nas disciplinas de computação e programação, que permite calcular as mais variadas funções, sejam elas complexas ou não, não se perca com a sedução de usar "aplicativos enlatados" que cada vez mais aparentemente resolvem tudo.

Quando o conhecimento se perde o seu resgate pode se tornar custoso, passar a ser domínio de umas poucas empresas que o confinam em chips de computadores que serão vendidos a elevados preços.

No presente trabalho houve a possibilidade da criação de uma verdadeira ferramenta de cálculo que pode proporcionar assistência a uma pesquisa séria em outra área de conhecimento humano.

Recomenda-se que cada colega professor busque registrar as pequenas pérolas do saber de suas disciplinas, antes que seja tarde.

Agradecimentos

Gostaríamos de agradecer ao patrocínio do fundo Mackpesquisa da UPM no projeto citado no presente artigo, bem como ao convite da Profª Drª Yara Maria Botti Mendes de Oliveira, que liderou a pesquisa, de participarmos dos trabalhos do mesmo, sem o qual não teríamos desenvolvido o presente estudo.

REFERÊNCIAS BIBLIOGRÁFICAS

BARROS, E. A. R.; PAMBOUKIAN, S. V. D.; ZAMBONI, L. C. **C++ Builder para Universitários**. São Paulo: Páginas & Letras, 2ª ed., 2003.

BARROS, E. A. R.; GRINKRAUT, M. L.; ZAMBONI, L. C.; PAMBOUKIAN, S. V. D. **Delphi para Universitários**. São Paulo: Páginas & Letras, 2ª ed., 2000.

BARROS, Edson de Almeida Rego; PAMBOUKIAN, Sergio Vicente D.; ZAMBONI, Lincoln César. **Ensino de Computação para estudantes de Engenharia**. COBENGE2004 (Congresso Brasileiro de Ensino de Engenharia), Brasília – DF, Unb 14 a 17 set 2004.

ROSA NETO, Ernesto. **Números complexos**. São Paulo: PAED – Pesquisa e Assessoria em Educação, 2ª ed., 1981.

SONNINO, Sérgio; MIRSHAWKA, Victor. **Números complexos**. São Paulo: Nobel, 3ª ed., 1965.

VIGGIANI, Domingos. **Alguns aspectos da aplicação dos números complexos e a geometria**. Marília: Faculdade de Filosofia, Ciências e Letras de Marília, 1968.

ZAMBONI, Lincoln César; MONEZZI JÚNIOR, Orlando. **Cálculo Numérico para Universitários**. São Paulo: Páginas & Letras Editora e Gráfica, 2002.

***Abstract**: A good example of the use of disciplines of computation and programming, in the courses of engineering, giving support to the development of multidisciplinarity, is its iteration with numerical calculus. The considered problem in this paper is the generation of an algorithm that is capable to solve by numerical methods the integral of a complex function with a real domain (R→C) in a didactic and pedagogic point of view. In such a way, we review the theories about numerical integration of real functions using Trapeze's Rule and also complex numbers, develop and implement an algorithm, with objects orientation in C++ Builder and in Delphi languages. This works had been resulted of a project of research developed by the Escola de Engenharia da UPM (Universidade Presbiteriana Mackenzie) and sponsored by the fund Mackpesquisa of the same university. The idea of the presentation of the work for COBENGE 2005 is to share with the academic community the set of experiences accumulated with the project.*

Key-words: Trapeze's Rule, Complex number, Computation, Programming, Numerical Calculus