



Anais do XXXIV COBENGE. Passo Fundo: Ed. Universidade de Passo Fundo, Setembro de 2006.
ISBN 85-7515-371-4

UM TUTORIAL PARA PROGRAMAÇÃO EM VISUAL FORTRAN

Ivomar B. Soares – ivomar@cos.ufrj.br

Universidade Federal do Rio de Janeiro (UFRJ), Programa de Engenharia de Sistemas e Computação (PESC/COPPE)

Centro de Tecnologia – Sala H319 – Cx. Postal 68511

21945-970 – Rio de Janeiro – RJ

Wilton P. da Silva – wiltonps@uol.com.br

Cleide M. D. P. S. e Silva – cleidedps@uol.com.br

Universidade Federal de Campina Grande (UFCG), Departamento de Física (DF/CCT)

Rua Aprígio Veloso – 882 – Bodocongó

58109-970 – Campina Grande – PB

Diogo D. P. S. e Silva – diogodme@gmail.com

Universidade Estadual de Campinas (UNICAMP), Instituto de Matemática, Estatística e Computação Científica (IME)

Rua Sérgio Buarque de Holanda – 651 – Cidade Universitária – Cx. Postal: 6065

13083-859 – Campinas – SP

Cleiton D. P. S. e Silva – cleitondiniz@directnet.com.br

Instituto Tecnológico de Aeronáutica (ITA), Divisão de Engenharia Eletrônica (IEE)

Praça Mal. Eduardo Gomes – 50 – Vila das Acácias

12228-900 – São José dos Campos – SP

Resumo: Este artigo tem o objetivo de comunicar o desenvolvimento de um tutorial, em Português e em Inglês, para programação em Visual Fortran. O tutorial é dividido em nove lições das quais três são destinadas a realizar um breve estudo dos principais recursos da linguagem, com ênfase nas características introduzidas nas revisões de 1990 e 1995, como alocação dinâmica de memória e também construção (e uso) de módulos. Quatro lições adicionais apresentam as noções básicas para a construção de uma interface gráfica mais amigável para os programas desenvolvidos. Essas noções básicas envolvem os conceitos fundamentais de programação para o ambiente Windows, o que inclui construção de menus, caixas de diálogo, animações, gráficos 2D e 3D, DLL e outros recursos mais avançados. É apresentado o código fonte de uma estrutura que possibilita compilar, com recursos gráficos, programas desenvolvidos em Fortran 77. As duas lições restantes são destinadas a recursos visuais mais avançados e em breve serão concluídas. O grande número de downloads de uma versão piloto do tutorial, bem como o número de e-mails recebidos pelos autores com questões ligadas ao Visual Fortran, e ainda, a grande demanda por cursos de extensão sobre programação visual em Fortran com o uso do tutorial são indicadores da relevância do material desenvolvido e distribuído livremente através da internet.

1. INTRODUÇÃO

Nas duas últimas décadas, com os avanços ocorridos na área da informática e com a inclusão dos computadores no cotidiano das pessoas, houve uma profunda modificação nos recursos disponíveis para ensino e pesquisa, em todas as áreas do conhecimento humano. Para acompanhar esta revolução, tanto na busca por novos conhecimentos, como no acesso às informações disponíveis, professores, alunos e profissionais de diversas áreas tiveram que se adequar às modernas ferramentas disponíveis na área da computação. Novas linguagens de programação foram desenvolvidas, e as existentes até então tiveram que ser modernizadas para não sucumbir diante deste novo cenário. Como exemplo desta modernização pode-se citar o *Basic*, que passou a se chamar *Visual Basic* ao agregar recursos visuais (ver, por exemplo, *The History of Visual Basic and BASIC on the PC*, online).

Uma das linguagens disponíveis no início dessa revolução e que teve (e ainda tem) uma grande influência no desenvolvimento da computação é o *Fortran* (*FORmula TRANslation*). Existe uma farta literatura em língua portuguesa sobre *Fortran 77* (um dos primeiros e mais conhecidos padrões da linguagem Fortran), podendo-se destacar FARRER *et al* (1992). Grande parte dos programas desenvolvidos até meados da década de 80, principalmente aqueles com aplicação em ciências exatas, era desenvolvida nessa linguagem, criada na década de 50 (ver, por exemplo, a seção 1.4 “*The History of Fortran*” em CHAPMAN, 1998). Mesmo atualmente, o acervo de programas para a área tecnológica desenvolvidos em *Fortran 77* é significativo, mas, em geral, restrito à área acadêmica. Muitos desses programas correm o risco de entrar em desuso devido ao fato de não terem uma interface gráfica mais amigável, embora apresentem uma boa performance em termos de tempo de execução de programas de cálculo numérico.

No início da década de 90, a linguagem *Fortran* foi reformulada para se adequar à nova realidade em termos de informática, havendo uma significativa reformulação da linguagem, em si, voltada para o cálculo numérico, com a introdução de muitos recursos como, por exemplo, alocação dinâmica de memória, criação (e uso) de módulos e também de DLLs (*Dinamic Link Library*). Algumas opções de programação visual para o ambiente Windows foram criadas, e, dentre elas, destacam-se o *QuickWin Application* e também o *Windows Application*. Nascia, então, o primeiro estúdio para programação em *Visual Fortran*, denominado *Microsoft Fortran PowerStation* (ver *Language Guide*, 1993). Conforme o *Language Guide* (1993), o *QuickWin Application* é uma opção de programação visual simplificada, com recursos que possibilitam: a) compilar programas criados para o ambiente DOS dentro de aplicações simples para Windows, b) minimizar e maximizar aplicativos, c) chamar rotinas gráficas, d) carregar e salvar bitmaps, e) selecionar, copiar e colar texto, gráfico ou uma mistura de ambos, f) detectar e responder a cliques do mouse (*Waitonmouseevent*), g) imprimir gráficos, h) criar menus, i) criar ícones e j) abrir múltiplas janelas. Em contraposição à excelência em termos de cálculo numérico, logo se perceberam alguns problemas com os recursos visuais dos programas desenvolvidos neste estúdio. Como exemplo, podem-se destacar a pouca fidelidade na reprodução de cores na tela, identificado em alguns tipos de computadores, e também na impressão de documentos. Havia, ainda, alguns erros em tempo de execução denominados “*QuickWin Internal Error*”, que ocorriam de forma aleatória, em geral, quando, no programa, se usavam eventos controlados pelo mouse (*Subroutine Mouse*) ou ainda a função *Waitonmouseevent*. Com isso, muitos engenheiros, professores e estudantes que trabalhavam com *Fortran* se sentiram desestimulados em estudar a nova ferramenta, bem mais complexa que a antiga, e desistiram

de incorporar recursos visuais em seus programas. Mesmo quando utilizavam o estúdio *PowerStation* para desenvolver os seus programas com o *Fortran 90*, muitos programadores preferiam trabalhar em uma opção de programação chamada *Console Application*, voltada para a criação de programas a serem executados no ambiente DOS.

Em 1997, os problemas detectados com os recursos visuais (os mencionados aqui e outros) foram resolvidos com o lançamento do *Microsoft Visual Studio (Microsoft Developer Studio 97)*. Este estúdio representou um avanço em muitos aspectos em relação ao *PowerStation*, mas ele também tinha alguns defeitos, como, por exemplo, a demora na inicialização dos programas nele compilados, quando executados concorrentemente com outros programas. A função *Playsound* foi retirada do módulo *Dfwin* (antigo *Msfwin*) de forma que a execução de recursos sonoros carecia do uso de uma solução alternativa: adicionar, ao projeto, uma biblioteca chamada *Winmm.lib*. Mais uma vez, uma boa parte dos programadores em *Fortran* preferiu continuar utilizando a opção de programação *Console Application*.

No ano de 2000, o estúdio passou a se chamar *Compaq Visual Fortran (CVF, 2000)* e sofreu várias modificações que fizeram dele uma ferramenta bem mais estável. Um indicador desta estabilidade foram os softwares que começaram a ser disponibilizados na Internet, nestes últimos anos, desenvolvidos neste estúdio. Como exemplo pode-se citar o *DynaFit (KUZMIC, 2003)*, o *EnergyPlus (2003)* e o *SimFit (BARDSLEY, 2003)*. O problema com o qual se defrontou a maioria dos usuários do novo estúdio passou a ser, então, a familiarização com as opções de programação visual, que são, em geral, muito mais complexas que a opção *Console Application*. Para dificultar ainda mais o uso dos recursos gráficos disponíveis, não havia, na época, como ainda hoje não há, muitas referências sobre o assunto. As poucas referências disponíveis não eram acessíveis aos usuários comuns, ficando destinadas aos que já tinham algum conhecimento prévio em programação visual e, mesmo assim, somente na língua inglesa (ver, por exemplo, LAURENCE, 2002). Não raro, o interessado em *Visual Fortran* dispunha apenas da Ajuda do estúdio, que não é uma ferramenta de estudo muito amigável para programadores iniciantes.

Devido ao fato de SILVA e SILVA (1999) terem desenvolvido um programa para tratamento de dados denominado *LAB Fit Curve Fitting Software (LAB Fit)*, inicialmente para o ambiente DOS, e querendo que ele tivesse uma interface mais amigável, não restou, a eles, outra alternativa, senão estudar de forma detalhada a Ajuda do CVF e procurar conhecer ferramentas que viabilizassem a migração do *LAB Fit* para o ambiente Windows. Foi tão difícil este estudo, que eles decidiram publicar, em Português e em Inglês, um tutorial com as noções básicas necessárias ao aprendizado, denominado *VFortran Tutorial (SILVA et al, 2002)*. Entre as duas principais opções disponíveis para programação em *Visual Fortran* no CVF (*Windows Application* e *QuickWin Application*) foi escolhida para o tutorial a menos complicada, mas nem por isto considerada simples: *QuickWin Application*. Além de ser menos complicada por oferecer um escopo de programação visual já pronto, esta opção possibilita compilar programas desenvolvidos para o ambiente DOS como aplicações simples para Windows, o que faz do *QuickWin* um poderoso aliado dos adeptos do *Fortran 77*. Quanto ao tutorial, a idéia básica foi a de criar um material específico para quem já tinha alguma experiência em programação em *Fortran*, mas que não tinha familiaridade com programação voltada para o ambiente Windows. Neste tutorial nem se tentaria corrigir hábitos do *Fortran 77*, decorridos da própria estruturação inicial da linguagem, como, por exemplo, escrever os comandos a partir da sétima coluna (ao invés da forma livre com o uso de indentação), ou mesmo fazer um questionamento sobre o uso de variáveis implícitas, ou ainda, o uso de características consideradas obsoletas como os comandos *Goto* e *Format* (ver *Programmer's Guide*, 1994). O tutorial foi dividido em 9 lições, das quais 7 já foram concluídas e, em breve, será complementado com mais 2 lições contendo recursos visuais

avançados. Em cada lição, além do estudo das ferramentas de programação, o programador deve desenvolver, também, parte de um projeto que, ao final, resultará no seu primeiro software para o ambiente Windows.

2. O VFORTRAN TUTORIAL

Mesmo tendo como objetivo principal o estudo de recursos visuais de programação desenvolvidos para a linguagem *Fortran*, as três primeiras lições do tutorial são destinadas a realizar uma breve revisão dos principais componentes desta linguagem. Esta revisão procura destacar algumas inovações introduzidas na linguagem em 1990 e em 1995 como, por exemplo, a construção (e uso) de módulos e alocação dinâmica de memória. As quatro lições seguintes apresentam alguns recursos visuais para a construção de uma interface gráfica mais amigável para os usuários que utilizam programas desenvolvidos nesta linguagem. As Figuras 1 e 2 mostram, respectivamente, os itens do menu “Tutorial” e as lições disponíveis quando se clica no item “Texto com as lições (Detalhado – CVF)”.

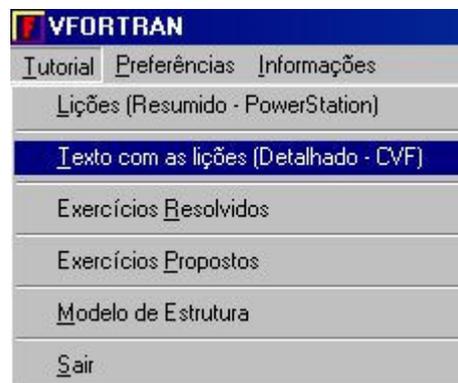


Figura 1 – Itens do menu “Tutorial”



Figura 2 – Caixa de diálogo para a escolha da lição

A seguir, um resumo com o conteúdo de cada lição será apresentado.

2.1. Lição 1: apresentação, o Visual Studio, formas fixa e livre, entrada e saída

A primeira lição faz uma apresentação resumida do *Fortran 77* e do *Fortran 90*, tentando realizar um paralelo entre as duas, e mostra como acessar o estúdio de desenvolvimento, o CVF. Há uma abordagem sobre as formas fixa (*Fortran 77*) e livre (*Fortran 90*) de codificação dos programas e também aborda a questão da entrada e saída de dados. É feita, ainda, uma revisão sobre variáveis e formatos, e também, sobre abertura de arquivos e leitura de dados. Também é abordado os temas programa fonte, programa objeto e programa executável. A lição propõe o desenvolvimento de um programa simples e termina com aquilo que será um aspecto essencial em todas as lições e na vida do usuário como programador: incentivá-lo a conhecer e utilizar a Ajuda do estúdio.

Uma informação adicional: como as lições são apresentadas em arquivos do tipo *hlp* (arquivos de ajuda do Windows), há a possibilidade de se selecionar trechos de código fonte dessas lições, copiar, e depois colar em arquivos de um projeto, o que simplifica e agiliza seu estudo.

2.2. Lição 2: tipos de dados, operadores, if, goto, do, dimension, programas simples

Esta lição faz uma revisão dos tipos básicos de dados (inteiro, real, complexo, literal e lógico) e dos principais operadores da linguagem: de atribuição, aritméticos, de concatenação, lógicos e relacionais. Há, também, a apresentação dos comandos *If*, *Goto*, *Do* e *Select Case*. A declaração *Dimension* é abordada para motivar e estimular o estudo de alocação dinâmica de memória. A lição é encerrada com o desenvolvimento de um programa simples para a resolução de equações do segundo grau que, posteriormente, será adaptado a uma estrutura com interface gráfica.

2.3. Lição 3: programas complexos, programa principal, funções e subrotinas

A terceira lição aborda o desenvolvimento de programas mais complexos, para os quais existe a possibilidade de sua divisão em um programa principal e em subprogramas (funções e subrotinas) para tornar a programação mais modular e estimular a reusabilidade de código. Há também um breve estudo sobre recursividade no uso de subrotinas. Nesta lição, o programador tem o seu primeiro contato, ainda que um tanto quanto superficial, com algumas funções e subrotinas gráficas contidas em módulos disponíveis no estúdio. É proposto o desenvolvimento de um programa para calcular o valor médio e o desvio padrão de uma série de valores. Tal programa também será adaptado a uma estrutura de programação visual que resultará no primeiro software para Windows a ser desenvolvido. A lição é encerrada com o estudo da criação e uso em programas de pacotes que contêm um conjunto de funções e subrotinas, chamados de módulos.

2.4. Lição 4: estruturação de um projeto com interface gráfica

A quarta lição apresenta a estrutura de um projeto em *QuickWin Application*. É apresentado o código fonte da estrutura que resultará em um software simples, com interface gráfica. O código fonte é subdividido em: a) um programa principal com laço infinito para que o programa não termine ao final da primeira execução, b) uma função denominada *Initialsettings* e c) várias subrotinas denominadas *Callbacks*. Normalmente, em *QuickWin Application*, o programa principal é a parte do software onde se obtém as informações gerais sobre a configuração do computador (como resolução, número de linhas e colunas da tela, etc). É também no programa principal que normalmente se faz **a modificação dessas**

configurações, e também, onde se define a aparência da tela principal do software. Além do programa principal, é enfatizado, também, a forma de se construir um menu através de uma função lógica chamada *Initialsettings*. É abordada, ainda, a forma de se executar antigos programas desenvolvidos para o ambiente DOS com subrotinas denominadas “callbacks”. Esta lição termina com a transformação dos programas desenvolvidos nas lições anteriores em subrotinas *callbacks* e com a sua inclusão na estrutura previamente estudada.

2.5. Lição 5: diálogos: tipos, construção e ativação de uma caixa de diálogo

A quinta lição aborda a construção e a ativação de caixas de diálogo, bem como alguns tipos de diálogos disponíveis em *QuickWin Application*. É explicado como incluir, no projeto, as caixas de diálogos que o programador julgar necessárias ao bom desempenho de seu software. Explicam-se, também, formas de construção e inserção de *bitmaps* e ícones em um projeto.

2.6. Lição 6: animações (idéia básica), função *Dlgsetsub*, gráficos 2D e 3D

A sexta lição enfoca a criação de animações simples e explica também como se processa a abertura de outras janelas, além da principal, chamadas “*Child Window*”. A lição aprofunda o estudo de criação de caixas de diálogos, enfocando dois novos elementos de diálogos: “*check box*” e “*radio button*”. Aborda, também, a evocação de caixas de diálogo, com o uso da função lógica *Dlgsetsub*, e ainda o assunto recursividade em caixas de diálogo. A lição ainda fala sobre o desenvolvimento de programas gráficos 2D e 3D, apresentando, inclusive, dois programas fontes preliminares que, compilados e executados, podem gerar gráficos similares aos mostrados nas Figuras 3 e 4.

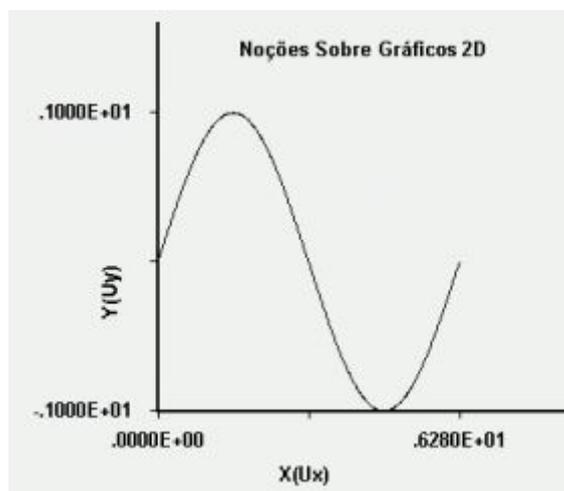


Figura. 3 – Gráfico da função $y = \sin(x)$

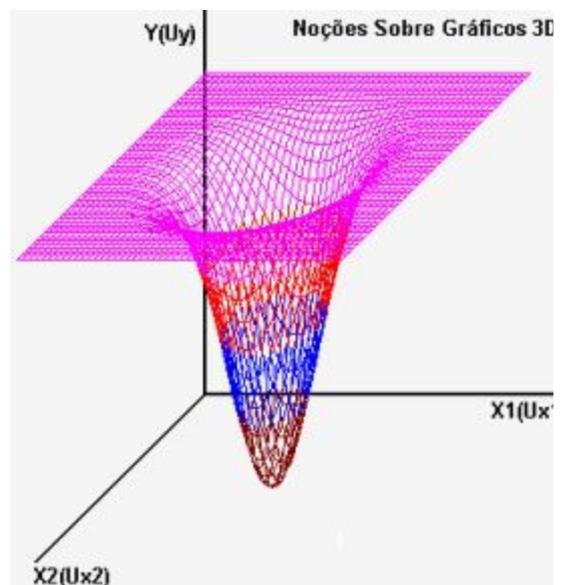


Figura. 4 – Gráfico da função
 $y = -\exp(-x_1^{**2} - x_2^{**2})$

Deve-se ressaltar que, no tutorial, o objetivo destes programas é mostrar a base da programação para a construção de gráficos, e não como fazê-los com qualidade para publicação.

2.7. Lição 7: criação de uma DLL e evocação por um executável

A lição de número 7 tem um único objetivo: mostrar de forma simples e direta como desenvolver, compilar e evocar, em *Fortran 90*, uma “*Dinamic Link Library*” (DLL). Uma DLL é um arquivo que pode ser carregado e executado por programas dinamicamente; basicamente, é um repositório externo de código pré-compilado. Um exemplo simples de desenvolvimento de uma DLL, bem como de um programa executável que a evoque é apresentado nesta lição. Já detalhes sobre suporte a diferentes linguagens, embora não seja especificamente tratado, está disponível na Ajuda do estúdio.

2.8. Lições 8 e 9

Essas duas lições visam ao estudo de: a) eventos controlados pelo mouse e b) abertura e salvamento de arquivos. Muito embora tais lições ainda não estejam concluídas, os assuntos a serem abordados já estão disponíveis na forma de programas fontes, na pasta de instalação do tutorial, em um arquivo compactado chamado “*CodigosFontes.zip*”. Neste arquivo existem vários códigos fontes exemplos, e alguns deles cobrem os assuntos destas duas lições cujo texto final ainda não está disponível.

3. UMA ESTRUTURA PARA PROGRAMADORES INICIANTES

Na lição de número 4 é apresentado o código fonte de uma estrutura inicial para programação em *Visual Fortran*. Entretanto, no final da lição há um comentário sobre as limitações dessa estrutura: recomenda-se apenas para a execução de tarefas rápidas, porque, durante sua execução, todo o software fica em modo de espera e, portanto, encontra-se indisponível. Isto acontece porque a função *Initialsettings* bloqueia todo o software enquanto uma subrotina *callback* evocada por ela estiver executando uma determinada tarefa. Outro ponto fraco é que os programas desenvolvidos com essa estrutura não dispõem de uma barra de ferramentas, o que seria uma alternativa ao uso do menu. Aliás, a opção *QuickWin Application*, até então, não dispõe deste recurso pronto (ver *Language Guide*, 1993), mas ele pode ser construído manualmente. Para contornar os problemas dessa estrutura preliminar, uma outra mais complexa, porém mais completa, está sendo disponibilizada. Nesta estrutura, além da existência de uma barra de ferramentas, uma *callback* evocada pela função *Initialsettings* não executa diretamente a tarefa correspondente ao clique de um botão (ou de um item do menu). Para evitar o bloqueio do software durante a execução, a *callback* transfere tal execução para o laço infinito do programa principal. Isto possibilita a criação de softwares mais eficientes, como o que será mencionado a seguir.

3.1. Aparência do software

A estrutura eficiente mencionada anteriormente já está disponível, podendo ser encontrada no diretório de instalação do *VFortran Tutorial* em um arquivo compactado chamado *TBM.zip*. Quando compilada e executada, a sua interface visual, que pode ser modificada à medida em que o programador for adquirindo experiência, é aquela mostrada na Figura 5.

Apesar de ser bem mais complexa quando comparada com as anteriores, envolvendo inclusive eventos controlados pelo mouse, utilizados, por exemplo, para a construção da barra de ferramentas, a estrutura é mais fácil de ser utilizada por programadores iniciantes do que aquela proposta na quarta lição. Isto porque os programas desenvolvidos pelo usuário para o

ambiente DOS são simplesmente adicionados ao projeto com mínimas alterações (às vezes, nenhuma), conforme as instruções simples que serão mencionadas a seguir.

3.2. Instruções para uso da estrutura

Naturalmente, o uso desta estrutura de programação pressupõe que o interessado já tenha alguma informação básica acerca de programação visual em *Fortran*. Mas, mesmo aqueles que estão apenas iniciando os seus estudos nessa área podem utilizar a estrutura, transformando seus antigos programas para DOS em aplicativos para o ambiente Windows. Para tal, foi elaborado um conjunto de instruções curtas, direcionadas a tal estudo.



Figura 5 – Tela principal da estrutura criada para programas em Visual Fortran

4. CONCLUSÕES

É inegável que a década de 90 foi pouco proveitosa para os programadores em *Fortran*, em termos de programação visual, quando comparado com outras linguagens. São poucos os softwares disponíveis nesta linguagem, destinados a um público diferente do acadêmico, desenvolvidos durante este período. Entretanto, o legado de programas desenvolvidos nos últimos 50 anos é enorme. Existe um número imenso de bibliotecas com subrotinas altamente otimizadas, principalmente para cálculo numérico, desenvolvidas em *Fortran* (ver, por exemplo, as “*Math Libraries*” em qualquer um dos três estúdios já referidos). Os programas desenvolvidos com essa ferramenta, em geral, são rápidos e com erros numéricos mínimos, em comparação com outras ferramentas similares. Essas características mantiveram viva, no meio acadêmico, uma linguagem que durante uma década inteira enfrentou alguns problemas em termos de avanços em recursos gráficos. Com o desenvolvimento de ferramentas visuais mais confiáveis, o que aconteceu a partir do ano de 2000, a tendência para o *Fortran 90* é tentar ocupar uma posição de mais destaque em universidades com cursos na área das ciências exatas.

Infelizmente, ainda existem poucos livros sobre a utilização de recursos visuais em *Fortran* e isso deve explicar, pelo menos em parte, o aparente sucesso e a utilidade do *VFortran Tutorial*, tanto no Brasil, quanto em alguns outros países. Uma versão piloto,

disponibilizada no início de 2002, já foi utilizada por cerca de 30.000 pessoas, o que é um número expressivo, levando-se em conta a especificidade deste material. O interesse crescente em *Visual Fortran*, tanto por estudantes quanto por professores de cursos de engenharia e afins, pode ser evidenciado pelas centenas de *e-mails* recebidos nos últimos dois anos pelos dois primeiros autores deste artigo, com questionamentos sobre temas ainda não abordados pelo tutorial. Mesmo na universidade em que tais autores trabalham, a demanda por cursos de extensão sobre programação visual através do *Fortran 90* é muito grande e crescente. Em dezenas de trabalhos de iniciação científica, mestrado e doutorado já se pode perceber a utilidade do *VFortran Tutorial*. Obviamente, a publicação deste artigo teve como objetivo levar a um número maior de pessoas a informação sobre a existência deste material, o que deve aumentar ainda mais o número de problemas solucionados com o estudo dessa ferramenta.

Por último, uma pergunta que leitores deste artigo e usuários do tutorial poderiam fazer, seria a respeito do sistema operacional: porque *Windows* e não *Linux*? Na verdade, os próprios autores já se fizeram esta pergunta e, ao invés de tentar respondê-la, já estavam realizando estudos iniciais para desenvolver uma versão para Linux do software *LAB Fit* que, em última análise, foi o elemento gerador de todo este trabalho. Entretanto, como já existem softwares como o *Winehq* (2003) que possibilitam a execução de aplicativos Windows no ambiente Linux, apesar de os autores reconhecerem que esta não é a solução ideal, essa questão, embora importante, deixou de ser uma prioridade, pelo menos por enquanto. Quanto ao tutorial, no momento, o mais adequado é concluir o trabalho iniciado, disponibilizando as duas últimas lições, freqüentemente cobradas por usuários, abordando os temas já mencionados.

5. REFERÊNCIAS BIBLIOGRÁFICAS

BARDSLEY, W. G. **Simfit V5.6.11**. Copyright © 2003, University of Manchester, online (acesso em 29/04/2006), disponível em < <http://www.simfit.man.ac.uk/default.htm> >.

CHAPMAN, S. J. **Fortran 90/95 for Scientists and Engineers**. WCB/McGraw-Hill, 1st Edition, Boston, USA, 1998.

Compaq Visual Fortran (CVF). Copyright 2000 Compaq Computer Corporation, Additional Information: Compaq Visual Fortran Professional Edition.

EnergyPlus V 1.1.1 (2003). **United States Department of Energy**. Online (acesso em 29/04/2006), disponível em < <http://www.eere.energy.gov/buildings/energyplus/> >.

FARRER, H.; Becker, C. G.; Faria, E. C.; Campos Filho, F. F.; Matos, H. F.; Santos, M. A.; Maia, M. L. **Fortran Estruturado**. Editora Guanabara Koogan S.A., 1^a Edição, Rio de Janeiro, RJ, 1992.

KUZMIC, P. **Program DynaFit**. Online (acesso em 29/04/2006), disponível em <<http://www.biokin.com>>.

Language Guide: Microsoft Fortran PowerStation. Professional Development System, Microsoft Corporation, 1993.

LAURENCE, N. **Compaq Visual Fortran – A Guide to Creating Windows Applications**. Digital Press, Woburn MA, USA, 2002.

Microsoft Developer Studio 97. Copyright 1994-97 Microsoft Corporation, Additional Information: Visual Fortran Professional Edition.

Programmer's Guide - Microsoft Developer Studio. Copyright 1994-95 Microsoft Corporation, Additional Information: Fortran PowerStation 4.0.

SILVA, W. P. e SILVA, C. M. D. P. S. **LAB Fit Curve Fitting Software** (1999-2004). Online (acesso em 29/04/2006), disponível em < <http://www.angelfire.com/rnb/labfit> >.

SILVA, W. P.; Silva, C. M. D. P. S.; Silva, C. D. P. S.; Soares, I. B. **VFortran Tutorial** (2002). Online (acesso em 29/04/2006), disponível em <http://www.extension.hpg.ig.com.br>.

The History of Visual Basic and BASIC on the PC. **Online (acesso em 29/04/2006), disponível em** < <http://dc37.dawsoncollege.qc.ca/compsci/gmack/info/VBHistory.htm> >.

Winehq (2003). Online (acesso em 29/04/2006), disponível em < <http://www.winehq.com> >.

A TUTORIAL ON PROGRAMMING IN VISUAL FORTRAN

Abstract: *This paper wants to make available to the general public the development of a tutorial, in Portuguese and in English, on programming in Visual Fortran. The tutorial is divided into nine lessons, of which the first three ones are destined to perform a quick overview of the main features of the language, with emphasis on the characteristics introduced with the 1990 and 1995 revisions, like allocatable arrays and the possibility to use modules. Additional four lessons present the basic notions for the building of a friendlier, graphical user interface by the developed programs. These basic notions involve the general concepts of programming for the Windows environment, including the development of menus, toolbars, dialog boxes, animations, 2D and 3D graphs, DLL and other more advanced resources. The source code of a structure that allows the compilation, with graphical resources, programs developed in Fortran 77 is presented. The two remaining lessons are destined to more advanced visual resources and they soon will be finished. The great number of downloads of a trial version of this tutorial, as well as the number of received emails related to Visual Fortran and also the great demand for courses using the tutorial are indicators of the usefulness of the developed material that is distributed freely at the internet.*

Key-words: GUI, Visual programming, Windows, Menu