文章编号:1671-6833(2003)01-0066-04

# 基于 Visual Fortran 的交互视算技术

周振红[1], 张君静[1], 陈峙峰[2], 杨国录[3]

(1. 郑州大学环境与水利学院,河南 郑州 450002; 2. 周口市公路局,河南 周口 466000; 3. 武汉大学水利水电学院,湖北 武汉 430072)

**摘 要**: 在大型非线性数模中,借助于交互视算技术,以浮力射流数值模拟为工程应用背景,在 Visual Fortran 中开发 QuickWin 型的应用程序,利用其图形库设计界面元素及绘制图形,并以创建多线程应用为突破口,使计算和绘图在主线程之外的单独线程中进行,主线程专门用来响应窗口中控件所产生的事件.研究结果表明,可以在 Visual Fortran 环境中实现交互视算技术.

**关键词**: 浮力射流; 数值模拟; 交互视算; Visual Fortran; QuickWin; 多线程

**中图分类号**: TP 311 **文献标识码**: A

## 0 引言

在计算机软、硬件飞速发展的今天,数值模拟已不满足于仅仅取得数值计算的结果,而是朝着交互视算的方向发展.科学计算可视化中的交互视算(也称为驾驭式计算)强调两方面的能力:一是用可视化图形实时跟踪计算过程的能力,即跟踪处理;二是在计算过程中随时暂停或中止计算、调整工况或模型参数后恢复计算或重新计算的能力.借助于交互视算技术,可以对数值计算的过程进行细致的洞察,进而构造出合理的数学模型,以准确模拟现实世界中的自然现象.

前些年,笔者曾综合运用了几种技术开展交互式视算[1].在 Fortran PowerStation 4.0 中,将执行计算的过程封装成动态链接库(DLL);再在 Visual Basic 6.0 中,采用混合语言编程技术,将动态链接库进一步封装成 COM 组件,利用组件的事件所发出的异步通讯机制,最终实现了交互视算技术.

现在,Fortran 语言标准已由 Fortran 77、Fortran 90发展至 Fortran 95,运行于 32 位 Windows 操作系统的 Fortran 集成开发环境(或编译器)也由 Microsoft Fortran PowerStation 4.0、Digital Visual Fortran 5.0,发展至 Compaq Visual Fortran 6.0/6.5.在 Visual Fortran 环境中,完全可以设法实现交互视算技术.本文选用 Visual Fortran 6.5 作开发环境,采取 Fortran 90/95 标准,以浮力射流数值模拟为工程背景,探讨交互视算技术的实现问题.

## 1 Visual Fortran 中的图形库

通常绘图程序不能够跨平台移植,因此在高级语言标准中,并没有专门的图形标准[2].要利用高级语言绘图,一般是调用其编译器或集成开发环境提供的图形库.Visual Fortran 中的图形库为 DFLIB,它是用模块(Module)封装的 Windows 绘图函数的一个子集,借此可以进行常规的绘图操作.

Visual Fortran 支持绘图操作的两种工程类型是 Standard Graphics 和 QuickWin.它们在绘图方面的使用方法完全相同,都是调用 DFLIB 中的过程来绘图,其主要差别在于 Standard Graphics 只能打开一个窗口绘图,而 QuickWin 可以打开多个窗口绘图;另外,QuickWin 还可以拥有菜单及对话窗的功能.因此,在 Visual Fortran 中绘图,多选取 QuickWin 工程类型,并引用 DFLIB 模块(USE DFLIB).

## 2 QuickWin 界面设计

编写交互视算程序,首先要设计用户操作界面,包括菜单、对话框、窗口等,以设置模型参数、

**收稿日期**:2002-10-21;**修订日期**:2002-12-21

**基金项目**:国家自然科学基金资助项目(50099620);河南省高校骨干青年教师资助项目

**作者简介**:周振红(1963-),男,山东省蓬莱市人,郑州大学副教授,博士后,主要从事 GIS、水利工程 DSS 的教学和科研工作.

控制计算程序的运行、展示计算的结果数据(文本和图形).在某种程度上,QuickWin 的界面设计要比 Visual C++、Visual Basic 和 Delphi 的界面设计来得容易,这得益于 Visual Fortran 对 Windows 图形设备接口(GDI)函数的封装.

在默认情况下,QuickWin 应用程序拥有一个框架主窗口和一个输出子窗口.在主窗口中,还有对子窗口及其图形和文本实施操作的菜单栏.用户可视情况需要,设计自己的菜单栏、对话框及窗口(见图 1).由于篇幅所限,本文只给出菜单、对话框及窗口设计的要点.

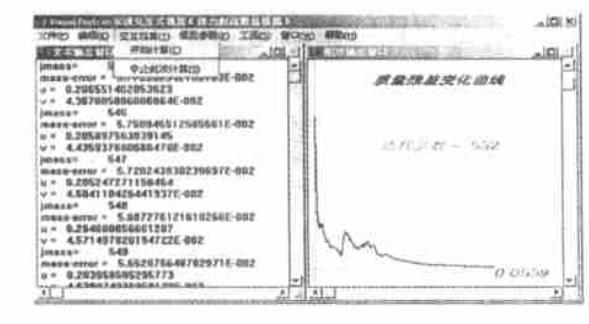

图 1　系统执行交互式视算的画面

Fig.1　The picture of system carrying out steering computation

## 2.1　菜单

通常定制主窗口菜单栏的时机应选在应用程序初始化时.QuickWin 应用程序在初始化时,会自动调用其内部的 InitialSettings 函数.因此,用户可采取"偷梁换柱"的策略,编写一个同名函数,在其中设置自己的菜单.常用的设置菜单的语句为

result = APPENDMENUQQ (menuID, flags, text, routine)

其中,result 为 4 字节长的逻辑型变量;menuID 为菜单栏中每一菜单条的顺序号,从左到右依次为 1、2、3…flags 为一标识菜单项的状态常数,如:是否有效、是否核取(核取的菜单项前有一符号"√")等;text 为菜单项名;routine 为点击菜单时执行的回调过程,若要执行复制、保存、打印等常规操作,可直接使用 QuickWin 提供的标准过程 WINCOPY、WINSAVE、WINPRINT 等.

下例为组织一个 QuickWin 应用程序中的第三组"绘图"菜单,其下拉菜单中设有两个菜单项"正弦"和"余弦":

```
i4=AppendMenuQQ(3,$MENUENABLED,'绘图(&P)'C,NUL)
i4=AppendMenuQQ(3,$MENUENABLED,'正弦(&S)'C,PlotSin)
i4=AppendMenuQQ(3,$MENUENABLED,'余弦(&C)'C,PlotCos)
```

其中,PlotSin 和 PlotCos 是用户编写的用于绘制正弦和余弦曲线的绘图过程;在菜单项名的后面有一字母 C,表明是采用 C 语言的以'\0'结尾的字符串.

## 2.2　对话框

对话框主要是提供操作界面,来输入参数或展示结果.Visual Fortran 用 DialogM 模块封装了对话框类型,以方便用户使用对话框.用户所要做的只是在 Visual Fortran 的资源编辑器中画出对话窗的外观,然后在程序中引用对话框类型定义(USE DialogM),实例化对话框对象,再设置和使用对话窗中控件的值.

需要注意的是:在保存对话框时,Visual Fortran 将其储存成 *.RC 文件,与此同时,还会自动产生一个 *.FD 文件,该文件包含了对话框中控件标识符(或 ID 号)的常量声明.使用前,除将 *.RC 和 *.FD 文件加入工程外,还需将 *.FD 文件包含(INCLUDE)进涉及对话框操作的过程中.

操作对话框的算法大致如下.

(1) 实例化对话框对象,如:

```
result =DlgInit(IDD_INPUT, dl) ! IDD_INPUT 为对话框ID,dl 为对话框类型变量
```

(2) 设置对话框中控件的值,指定命令按钮所要执行的回调过程,如:

```
result =DlgSet(dl,IDC_X_MIN,str) ! IDC_X_MIN 为文本框ID,str 为字符串变量
result =DlgSetSub(dl,IDC_RESET, ReSetRange) ! IDC_RESET 为按钮ID,ReSetRange 为点击按钮时执行的回调过程
```

(3) 显示模式对话框,如:

```
result =DlgModal(dl)
```

(4) 获取对话框中各控件的值,如:

```
if (result ==IDOK) then ! IDOK 为"确定"按钮ID
result =DlgGet(dl,IDC_X_MIN,str)
end if
```

上述代码中的 result 为长整型变量,DlgInit、DlgSet、DlgSetSub、DlgModal 和 DlgGet 为 DialogM 模块中的函数.

## 2.3　输出子窗口

在 QuickWin 中,创建一个子窗口的语法极为简单,只需在 OPEN 语句中规定 FILE 可选参数的值为'USER'即可,如:

OPEN(UNIT=12,FILE='USER',TITLE='Product Matrix'C)

该语句规定子窗口的引用单元号为12,标题为Product Matrix.

QuickWin可以打开多达40个子窗口.默认情况下,每个子窗口都是带有滚动条的文本窗口,大小为30行×80列.通过引用子窗口的单元号,就可以将输出(文本或图形)定向到单元号所代表的子窗口中.需要注意的是:

(1) 当用上述语句创建一个子窗口时,该子窗口只暂时存在于内存中,还不能成为桌面上可见的窗口.要使其成为可视的子窗口,须对其进行适当的操作.例如:

call ClearScreen($GCLEARSCREEN) ! 清屏

(2) 在有多个子窗口存在的情况下,某一时刻只能有一个窗口获得焦点,即被带到桌面的最前端.要进行绘图输出的子窗口未必要获得焦点,只要是活动窗口即可.使一个子窗口成为活动的,可通过调用SetActiveQQ函数来实现.函数FocusQQ使子窗口获得焦点,同时也使该窗口成为活动子窗口.

## 3 QuickWin绘图坐标系

绘图操作离不开所要使用的坐标系统.在其他开发环境(如Visual Basic,Delphi等)下绘图,通常采用窗口的物理坐标系(一般用像素表示坐标),且规定窗口左上角为(0,0),右下角为(窗口宽,窗口高).在这样的坐标系下绘图,不得不将绘图数据映射成像素图像数据,这给绘图工作带来不便.在Visual Fortran的QuickWin中,用户可用实际数据直接定义类似笛卡尔坐标系的逻辑窗口坐标系:

result = SETWINDOW(finvert, wx1, wy1, wx2, wy2)

其中finvert为逻辑参数,逻辑真表示$y$轴朝上为正,反之,$y$轴朝下为正;其余4个参数为双精度实型,用来规定窗口左上角和右下角的坐标,即限定窗口范围.

设置了上述逻辑窗口坐标系,就不再需要进行数据转换工作,可调用绘图过程在子窗口中直接绘图.绘图最常用的两个过程是MoveTo_W和LineTo_W(后缀W表示采用窗口逻辑坐标),其语法格式为:

CALL MOVETO_W(wx, wy, wt)

result = LINETO_W(wx, wy)

其中,参数wx,wy是双精度实数,代表位置坐标,MoveTo_W当中的坐标规定了画笔的起始位置,LineTo_W当中的坐标则规定了画笔的终止位置;参数wt代表先前的绘图位置,其数据类型为:

TYPE wxycoord
  REAL(8) wx
  REAL(8) wy
END TYPE wxycoord

## 4 实现交互式视算

本文用于浮力射流模拟的数学模型属大型、非线性,其显著特征是计算强度大,在满足精度要求前,计算需要反复进行迭代(大约几千次).在这种情况下开展交互式视算,最难实现的是驾驭计算.即在用动态图形实时跟踪计算过程时,随时暂停或中止计算,调整工况或模型参数后恢复或重新进行计算.

本文起初在交互视算程序中设计一全局逻辑变量,并通过菜单控制计算程序.但实际情况是:在计算的过程中(动态图形同时也在实时跟踪),点击菜单,系统并不做出应有的响应,要强行中止计算,不得不关闭主窗口或借助于Windows的任务管理器.经过分析和实践发现,计算一旦开始,应用程序(对应一个进程)的主线程几乎全部被计算和绘图操作所占有,这使得系统无力再来响应主窗口中菜单产生的事件.

实践证明:开发多线程应用程序是解决QuickWin中交互式视算的关键,同时也是高效的实现方案.该方案为计算和绘图操作单独设置一个线程,使进程中的主线程大部分时间处于空闲状态,以随时响应主窗口中的控件(如菜单)所产生的事件.具体的算法为:

(1) 在起动计算的菜单事件中,为计算和绘图(下例中的Compute)创建一线程.如:

ThreadHandle = CreateThread(0,0,Compute,%val(i), &
CREATE_SUSPENDED, j)

retlog = SetThreadPriority(ThreadHandle, THREAD_PRIORITY_HIGHEST)

retint = ResumeThread(ThreadHandle)

其中,线程优先级可设为次最高的,即比正常优先级高出2个优先级.

(2) 在执行迭代的循环过程中,每当要输出图形时先检察一全局控制变量的值.若为停止计算和绘图,就退出上述线程.退出线程的语句为:

call ExitThread(0)

(3) 在停止计算的菜单事件中,设置上述全局控制变量的值为"停止".另外,在上述操作线程的过程中需引用 DFMT 模块,该模块封装了 Windows 关于线程操作的函数.系统运行情况如图 1 所示.

## 参考文献:

[1] 周振红,杨国录,周洞汝.基于组件的水力数值模拟可视化系统[J].水科学进展, 2002,1(1):9～13.

[2] 彭国伦.Fortran 95 程序设计[M].北京:中国电力出版社,2002.

## Carrying out Steering Computation in Visual Fortran

ZHOU Zhen-hong[1], ZHANG Jun-jing[1], CHEN Shi-feng[2], YANG Guo-lu[3]

(1. College of Environmental & Hydraulic Engineering, Zhengzhou University, Zhengzhou 450002, China; 2. Zhoukou Highway Bureau, Zhoukou 466000, China; 3. College of Water Resources and Hydropower, Wuhan University, Wuhan 430072, China)

**Abstract**: The process of numerical computation can be presented with steering computation in large-scale and nonlinear modeling, in order to develop an accurate model. In practical context of numerical simulation on the buoyant jet, a QuickWin application is developed in Visual Fortran, interface elements and graphics being designed with graphic library of QuickWin. To carry out steering computation in Visual Fortran, the key is to create multithread application, drawing and computing in a single thread, and the system responding to the event of the window in main thread. Finally, steering computation is fully implemented.

**Key words**: buoyant jet; numerical simulation; steering computation; Visual Fortran; QuickWin; multithread

(上接第 65 页)

## Comparison of Methods to Produce Sample Points in Training ANN

WANG Shao-bo, CHAI Yan-li, LIANG Xing-pei

(Zhengzhou Research Institute of Mechanical Engineering, Zhengzhou 450052, China)

**Abstract**: Based on the theory of the NN in the biology, the ANN is a complicated, massive, nonlinear dynamic system that simulates the biologic neural structure. Lots of sample points are required to train the ANN. In this paper, three methods are presented to produce the sample points adopted in training the ANN. It is testified that under the same condition even design is the best method to produce sample points, followed by orthogonal design, and the third is ransacking. Ransacking will be better if the number of sample points is large. When the function changes little with the variables, fewer factor levels can be chosen and orthogonal design is a good method. Its advantages stand out for the occasion of multiple variables and many factor levels.

**Key words**: neural network; orthogonal design; even design