

Fortran 与 C#混合编程在地震属性分析中的应用

安晶¹, 安鹏²

(1. 盐城工学院 优集学院, 江苏 盐城 224051; 2. 东方地球物理公司研究院地研中心, 河北 涿州 072750)

摘要:充分利用现有的 Fortran 程序,通过混编方式,将其移植到 .NET 平台下,实现了地震属性分析中 C#可视化程序调用 Fortran 语言的混合编程。改造现有的 Fortran 源码,一方面可以达到两种语言优势互补的目的,另一方面可以利用前人留下的大量的 Fortran 程序,避免重复的编程,大大节约了科学研究时间。

关键词:地震属性分析; C#语言; Fortran 程序; 混合语言编程; .NET 平台

中图分类号: TP399 **文献标识码:** A **文章编号:** 1671-5322(2012)02-0034-04

Fortran 语言起源于 20 世纪 50 年代,其优势在于数值计算,在科学研究和工程设计领域有着广泛的用途。人们在长期的工程数值计算中积累了大量宝贵的程序资源,现有的地震属性分析程序大都是用 Fortran 语言编写的。而使用 Fortran 语言的缺陷是难以在 Windows 环境下开发高效、友好的可视化界面。C#语言是 Microsoft 推出的基于 .NET 平台的完全面向对象的开发语言,保留了 C++设计的灵活性,同时加入了 VB 快速开发特性,拥有十分强大的界面开发环境。为了提高软件的开发效率,提高代码的重用性,有必要对原有的 Fortran 程序进行改写,使原有性能优良的 Fortran 代码可以被 C#调用,最大限度的发挥两种语言的优点,以提高地震属性分析类软件的实现效率和操作便易性^[1]。

1 混合语言编程的技术规范

混合语言编程有两种具体的实现方法^[2]:

(1)利用目标文件(OBJ 文件)的方式。将 C#与 Fortran 在各自的编译环境下进行编译,产生各自的目标文件,然后再将它们进行链接,产生 C#的可执行程序。

(2)利用动态链接库的方式。首先将 Fortran 程序生成动态链接库(DLL 文件),然后利用 C#对其进行动态调用。

使用第一种方法时,常出现由于两种语言库

文件不匹配而导致的链接警告和错误。而 DLL 是一种基于 Windows 的程序模块,它的实现与编程语言和编译器无关,只要遵守 DLL 的开发规范和编程策略,编写正确的调用接口即可。因此,混合语言编程通过调用 DLL 来实现更具有通用性。

2 关键技术

在 C#语言中要实现调用 Fortran 编译的 DLL 文件,必须协调好调用程序与被调用程序之间的调用约定。主要包括以下 4 个方面:堆栈管理约定(调用约定)、命名约定、参数传递约定以及数组和字符串的传递约定^[3,4]。本文所使用的 C#编译环境是 Microsoft Visual Studio 2008, Fortran 的编译环境为 Intel Visual Fortran Compiler 10.1,操作系统平台为 Windows XP。

2.1 堆栈管理约定

决定函数参数传递时入栈和出栈的顺序,调用完成后由调用者还是被调用者来清理堆栈。C#语言在 Windows 平台上采用的与 C 语言相同,由调用方清除;而 Fortran 语言默认调用模式为 STDCaLL 模式,即被调用方清理堆栈。

在 Fortran 语言中可以通过编译器的通用指令“! DEC \$”后面的可选项“C”或“STDCALL”参数来实现:

```
! DEC $ ATTRIBUTES STDCALL :: Project
```

```
! DEC $ ATTRIBUTES C :: Project
```

第一条语句中的 STDCALL 模式指定由被调用方清除堆栈(其中 Project 为变量名或函数名),第二条语句中的 C 模式声明由主调函数清除堆栈(传递数组和字符串参数时不能用此方法指定)。

如果在 C#语言内做改动,则需要在 DllImport 属性中设置 CallingConvention 字段的值为 Cdecl(表示由调用方清理堆栈)或 STDCALL(表示由被调用方清理堆栈):

```
[DllImport("FileName.dll", CallingConvention = CallingConvention Cdecl)]
```

```
[DllImport("FileName.dll", CallingConvention = CallingConvention StdCall)]
```

2.2 命名约定

命名约定的协调是指两种语言经编译器编译后相互匹配的标识符应保持一致。C#语言是区分大小写的,Fortran 语言则一视同仁。此外,Fortran 语言经编译后对象(子例程或函数)名会发生变化,具体变化因调用约定不同而不同。

为协调两种语言间的命名约定,可以根据 Fortran 2003 标准中的规定,利用 BIND 属性来声明 Fortran 内部对象供外部程序调用时使用的名称。例如,将子例程的名字 SUBNAME 和函数的名字 FUNNAME 分别声明为外部调用时 Seis_Attr1 和 Seis_Attr2,声明方式如下:

```
SUBROUTINE SUBNAME( ) BIND (C, NAME = "Seis_Attr1")
```

```
FUNCTION FUNNAME( ) BIND (C, NAME
```

```
= "Seis_Attr2")
```

假设上述例子已经被编译到 FileName. DLL 库文件中,那么 C#语言中就要加入如下语句来声明这两个外部函数来实现对其的调用:

```
[DllImport("FileName.dll")]
```

```
Public static extern void Seis_Attr1 ( );
```

```
[DllImport("FileName.dll")]
```

```
Public static extern type Seis_Attr2 ( );
```

上述语句中 type 为函数 Seis_Attr2 的值类型。

2.3 参数传递约定

函数的参数传递有两种方式。一种是值传递(Pass - by - Value),被调用方得到参数值,向调用方传入参数的拷贝而非参数本身;另一种是引用传递(Pass - by - Reference),被调用方和调用方使用的是同一个参数。

默认情况下,Fortran 对所用参数都使用引用传递的方式,对于 C#来说,情况相对就复杂了些,分为值类型和引用类型传递两种。C#中所有数值类型都是值类型,而表示字符串的则是引用类型。参数传递约定包括两种语言之间的数据类型和数据存储结构是否符合、参数传递是通过传值还是地址来实现的。

对于 Fortran 和 C#首先要注意的是数据类型的协调,可以借助 Fortran 2003 标准提供的 ISO_C_BINDING(该模块定义了 Fortran 与 C#语言连接时必需的类型常量)内在模块定义的数据类型的种别常量来实现。Fortran 的常用数据类型通过种别常量与 C#语言的数据对照如表 1 所示。

表 1 Fortran 与 C#数据类型对照表

Table 1 Data Type Comparison of Fortran and C#

Fortran 数据类型	数据类型种别常量	C#数据类型	类型说明
Integer	C_INT	int	带符号 32 位整数
	C_SHORT	short	带符号 16 位整数
	C_LONG	long	带符号 64 位整数
	C_SIGNED_CHAR	sbyte	带符号 8 位整数
Real	C_FLOAT	float	32 位浮点值
C_DOUBLE	double	64 位浮点值	
Logical	C_BOOL	bool	布尔值
Character	C_CHAR	Char	字符型
		String	

2.4 数组和字符串传递约定

对于现有 Fortran 编制的地震属性分析源码,数组和字符串参数之间的传递是一个不容忽视的

问题。由于两种语言的混合编程在数组的存储结构和字符串的编码方式上有较大的差异,必须协调这两类参数的传递(字符串可以看作是由字符

组成的数组)。

数组在内存中都是线性连续存储的,不同的是 Fortran 中,数组采用按列优先的存储方式,在 C#语言中采用的是按行优先的存储方式。因此在用传址方式传递数组参数时,必须在传递前对数组做一个类似于矩阵转置操作变换,使行列的存储位置对调。

```
INTEGER :: Seis(1000 , * )
```

```
int Seis[ ][1000]
```

两种语言之间字符串参数的协调处理则稍显复杂。C#中字符串以 null 值来表示字符串的结束, Fortran 中则采用在字符串最右端添加空格的方式表示字符串的结束,并在其后使用一个隐藏的参数表示实际长度。此外 Fortran 2003 标准中规定的 Fortran 语言默认字符集为 ASCII 编码,因此 Fortran 语言对中文字符的支持依赖于编译器。而 C#语言中所采用的字符编码为 Unicode,对中文字符能实现语言层上的支持。因此,当字符串参数由 Fortran 传回 C#时,可利用 .NET 环境下 C#自身提供的函数来逐字符的将 ASCII 码转换成 Unicode 码。而对于字符串由 C#传入 Fortran 时,可以在 C#代码中添加一个整数型的参数表示字符串的长度。

3 地震属性分析软件的实现

本文利用 C#与 Fortran 混合编程实现地震属性分析软件的开发,C#用于开发用户界面, Fortran 用于开发算子例程和函数。篇幅有限,本文以 RS 理论为例子,该理论无需提供除问题所需处理数据之外的任何先验信息,仅根据观测数据即可删除冗余信息,比较不完整知识的程度。所以基于 RS 理论的地震属性优化方法具有对不完整数据进行分析、推理、发现数据间关系、优化属性组合的能力,结合自适应神经网络的优化方法是一种很好的地震属性优化方法。

3.1 决策算法

决策框图如图 1 所示^[5]:

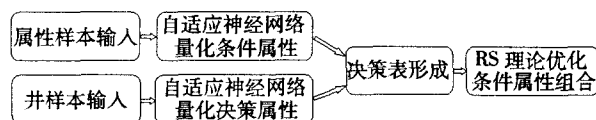


图 1 RS 理论决策图

Fig. 1 RS theory decision

其实现算法描述如下:

1) 用自组织神经网络方法对条件属性值进行分类,其分类结果为该条件属性的量化结果。

2) 对井点的储层参数(如渗透率等)进行分类,作为决策属性。

3) 选取井旁道种 N 道的地震属性与决策属性组成决策表。

4) 采用 RS 理论优选出地震属性个数最少的属性组合。

5) 用优化的地震属性组合进一步提取决策规则的值。

本文把由 Fortran 编写的自适应神经网络算法和 RS 理论优化算法编译成动态链接库,并在 C#程序中调用,从而实现了地震属性优化软件的开发。

3.2 创建 Fortran 程序动态链接库

在 Visual Studio 2008 的解决方案中选择 Intel (R) Fortran 语言创建一个空白的动态链接库工程(Dynamic - Link Library)。

```
Subroutine Attr_Class( * ,... * )
```

```
Implicit none
```

```
! DEC $ Attributes StdCall , DLLExport :: Attr_Class
```

```
USE, Intrinsic :: ISO_C_BINDING
```

对上述项目进行编译,编译成库文件“Attr_Class.dll”。

3.3 C#应用程序实现

在 C#项目所在的解决方案中添加一个 Windows 窗体应用项目,并利用窗体设计器如图 2 布置好程序界面。

在适当位置添加函数声明代码如下:

```
[DllImport(“Attr_Class.dll”, CallingConvention = CallingConvention StdCall)]
```

```
Public static extern void Attr_Class( * , * ,... , * , )
```

编写 Attr_Class(*) 操作事件的响应代码,并将上步生成的 Attr_Class.dll 和 Attr_Class.lib 文件加入到此项目中,进而编译生成可执行的软件。

最后通过此方法实现了对大庆某探区目标层段砂岩厚度的预测,如图 3 所示。

4 结论

本文利用 C#和 Fortran 混编方法实现了地震

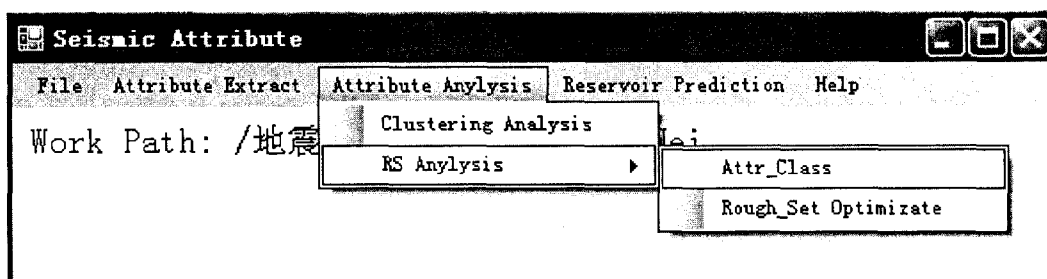


图2 地震属性分析软件界面
Fig.2 Seismic attribute analysis software interface

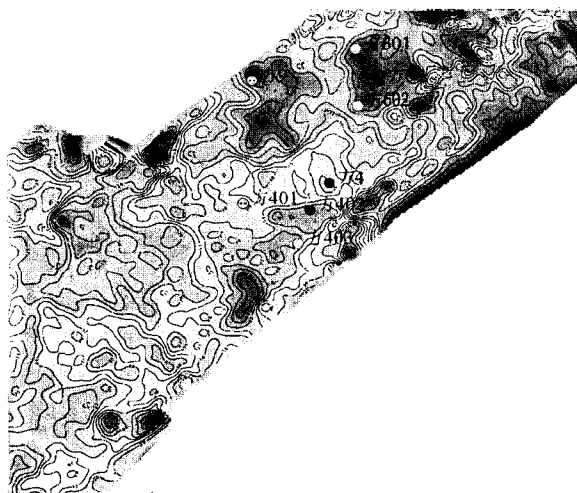


图3 目标层段砂岩厚度图
Fig.3 Target layer sandstone thickness

属性分析软件的编制,取得了初步的成果。同时,在软件实现过程中也体会到了混合编程的魅力。

1) 利用 C#语言进行界面开发,方便、高效、便于软件的维护和升级,这将使开发人员的注意力更专注于属性分析理论算法的完善和实现。

2) Fortran 计算程序被编译成动态链接库后,方便了 C#窗体程序的调用,轻松实现了对现有的经典 Fortran 程序代码的升级利用,既保护了珍贵的 Fortran 源程序代码,保留了其强大的数据计算优势,又使程序具有良好的界面,减轻了后人开发可视化软件的复杂度和工作量,大大节约了开发周期。

参考文献:

- [1] Christian Nagel, Bill Evjen, Jay Glynn. C#高级编程[M]. 4版. 李敏波,译. 黄静审校. 北京:清华大学出版社,2006.
- [2] 杨帆. 利用 Visual C#实现 FORTRAN 程序可视化的方法[J]. 计算机与数字工程,2008,36(11):170-172.
- [3] 周振红,颜国红,吴虹娟. Fortran 与 Visual C++混合编程研究[J]. 武汉大学学报,2001(5):56-59.
- [4] 陈遵德. 储层地震属性优化方法[M]. 北京:石油工业出版社,1998.
- [5] 王永刚,谢东,乐友喜,等. 地震属性分析技术在储层预测中的应用[J]. 石油大学学报:自然科学版,2003,27(3):30-32.

The Application of Mixed - Language Programming of Fortran and C# in Seismic Attribute Analysis

AN Jing¹, AN Peng²

(1. UGS College, Yancheng Institute of Technology, Yancheng Jiangsu 224051, China;
2. Institute of the BGP, Zhuozhou Hebei 072750, China)

Abstract: To make full use of the existing Fortran programs, this paper realizes Visual C# calling Fortran in seismic attribute analysis with mixed - language programming. With the transformation of the cases of the source code of Fortran, for one thing it is complementary of the advantages of each language, for another it can use those mature programs in Fortran language developed in previous days. So it can avoid the duplicate programming and save lots of time.

Keywords: seismic attribute analysis; C# Language; Fortran Program; Mixed - Language Programming; .NET Platform

(责任编辑:张英健)