

CLAPACK 在 vc++6.0 中成功调用

Kevin Wong

2010-3-24

序:

CLAPACK 是 LAPACK 的 C 语言接口。LAPACK 的全称是 Linear Algebra Package, 是非常著名的线性代数库。原版的 LAPACK 是用 Fortran 写的, 为了方便 C/C++ 程序的使用, 就有了 LAPACK 的 C 接口 CLAPACK。

网上虽然给出了一些关于 CLAPACK 在 windows 下的调用讲解, 但用起来总会有这样那样的问题。对于 Visual Studio C++ 6.0 环境如何实现对 CLAPACK 的调用, 基本上就没有相关资料了。

本人青睐于 vc++6.0 的运行速度, 研究了一下再此环境中调用 CLAPACK 的攻略。终于功夫不负苦心人, 在此将基本步骤梳理一下, 希望对大家有所帮助。

一、你需要准备什么?

(1) MS vc++6.0. (有些费话了, 呵呵)

(2) 下载 [clapack-3.2.1-CMAKE.tgz](#) 并解压缩。

(3) 下载 [CMAKE](#) 并安装。

有了(2)和(3)的准备可以提供 vc 各个版本需要的项目和工程文件了。

二、编译库文件

1) 使用 CMAKE

- ☒ 打开 CMAKE-gui
- ☒ 将 Source 目录指向你的 CLAPACK-3.2.1-CMAKE 文件夹;
- ☒ 将 Bulid 目录指向你准备存放生成文件的文件夹(最好不同于 Source 文件夹)
- ☒ 点击 configure, 选择 vc++6.0 作为你的编译器.(如果你的电脑上没有其他编译器, 默认就可以了)
- ☒ 你需要再次点击 configure 直到所有选项都变成白色.
- ☒ 点击 generate, 将会生成 Visual Studio 需要的项目和工程文件, 这就完成了.
- ☒ 关闭 CMAKE

2) 查看"build"指向的文件夹, 打开 CLAPACK.dsw.

3) 激活"ALL_BUILD"工程, 就可以组建(bulid)出所需要的所有的库文件了。

编译完成后, 找到 libf2c.lib blas.lib lapack.lib tmglb.lib。这些都是你自己编写程序时需要调用的库。另外, 在 lapack-3.2.1-CMAKE.tgz 解压的文件夹\INCLUDE 中的包含了这些库的头文件。

三、如何调用库文件。

1) 头文件

头文件就是.h 文件。存放在\INCLUDE 中。在自己的工程里加入这个目录就行了。程序中主要调用的头文件是 f2c.h 和 clapack.h。

2) 库文件

库文件就是我们前面编译生成的那些 lib 文件了。

3) vc 工程本身的一些设置:

(这个很重要, 很多时候就是因为这里设置不对, 才会出问题的)

//开始设置//

1 在设置中选择所有设置。

2 在 VC++IDE 中选择 Projects→settings→C/C++。

1)在 category 中选择 Code Generation, 在 Use run-time library 中选择 Debug Multithread DLL。(必须的)

2)在 category 中选择 Precompiled Headers, 中选择 Automatic use of precompiled headers, 并且在 Through header 下面添加 stdafx.h。(可省去)

3)在 Category 中选择 Preprocessor, 在 Preprocessor definitions 中添加 MSVC,MSWIND (可以省略)

3 在 VC++IDE 中选择 Projects→settings→Link

1)在 Categories 中选择 Input, 在 Ignore libraries 中填入: msvcrt.lib(必须的)

2)在 Object/Library Modules 框中, 添加库文件名: libf2c.lib blas.lib lapack.lib tmglb.lib (必须的)

//设置完成//

注意: 因为 BLAS 库通常比 LAPACK 库提供的子程序的版本要新, 所以 BLAS 库要列在 LAPACK 库之前。

注意: 如果想在调试时能对库函数进行源码级调试。那么需要在 VS 的 工具--选项--项目和解决方案--VC++目录 中添加\SRC 的目录。

四、vc++6.0 调用实例

////////// 以下代码例子来自 Kaien 的博文 //////////

本文, 我们将调用 CLAPACK 的一个函数 dgesvd_()来学习使用的方法。

注意: 包括此函数在内的所有的 CLAPACK 函数可以在\SRC 下找到源代码, 并在代码中有函数参数的说明信息。dgesvd_的代码文件就是 dgesvd.c。

dgesvd_的函数声明:

```
int dgesvd_(char *jobu, char *jobvt, integer *m, integer *n,
double *a, integer *lda, double *s, double *u, integer *
ldu, double *vt, integer *ldvt, double *work, integer *lwork,
integer *info)
```

dgesvd_的功能是对一个实矩阵 A 进行 SVD 分解(singular value decomposition)。

即 $A = U * SIGMA * transpose(V)$

dgesvd.c 文件里有详细地函数说明和参数说明。

SIGMA is an M-by-N matrix which is zero except for its min(m,n) diagonal elements, U is an M-by-M orthogonal matrix, and V is an N-by-N orthogonal matrix. The diagonal elements of SIGMA are the singular values of A; they are real and non-negative, and are returned in descending order. The first min(m,n) columns of U and V are the left and right singular vectors of A.

Note that the routine returns V^{*T} , not V.

```
.....

//test.cpp 代码//////////////////////////////////////
#include <stdio.h>
#include <process.h>
#include <f2c.h>
//因为程序是 C++, 而 CLAPACK 是 f2c 程序转换的 C 语言版本, 所以在此处用 extern 关键字调用
extern "C"
{
#include <clapack.h>
}

#define SIZE 4
int main()
{
    char JOBU;
    char JOBVT;
    int i;
    //数据类型 integer 是 fortran 里的。这里在 C++下可以使用的原因是 f2c.h 文件中已经作了定义
    integer M = SIZE;
    integer N = SIZE;
    integer LDA = M;
    integer LDU = M;
    integer LDVT = N;
    integer LWORK;
    integer INFO;
    integer mn = min( M, N );
    integer MN = max( M, N );
    double a[SIZE*SIZE] = { 16.0, 5.0, 9.0 , 4.0, 2.0, 11.0, 7.0 , 14.0, 3.0, 10.0, 6.0, 15.0, 13.0,
8.0, 12.0, 1.0};
    double s[SIZE];
    double wk[201];
    double uu[SIZE*SIZE];
    double vt[SIZE*SIZE];

    JOBU = 'A';
    JOBVT = 'A';
    LWORK = 201;

/* Subroutine int dgesvd_(char *jobu, char *jobvt, integer *m, integer *n,
    doublereal *a, integer *lda, doublereal *s, doublereal *u, integer *
    ldu, doublereal *vt, integer *ldvt, doublereal *work, integer *lwork,
    integer *info)
*/
```

```
    dgesvd_( &JOBU, &JOBVT, &M, &N, a, &LDA, s, uu, &LDU, vt, &LDVT, wk,
    &LWORK, &INFO);
```

```
    printf("INFO=%d \n", INFO );
    for ( i= 0; i< SIZE; i++ ) {
        printf("s[ %d ] = %f\n", i, s[ i ]);
    }
    system("pause");
    return 0;
}
```

```
//代码结束////////////////////////////////////////////////////////////////
```

```
////////// 以上代码例子来自 Kaien 的博文 //////////
```

运算结果输出:

INFO=0

s[0] = 34.000000

s[1] = 17.888544

s[2] = 4.472136

s[3] = 0.000000

调试成功!!! 庆祝一下啦!!! 🍷

五、致谢:

(1) 感谢 Kaien 在“如何在 VC 中调用 CLAPACK”一文中的详细讲解。

网址: http://hi.baidu.com/kaien_space/blog/item/dcb84b8b96347bd4fd1f1011.html

(2) 感谢 The Innovative Computing Laboratory (ICL) at the University of Tennessee 提供了英文版 CLAPACK for Windows.

而且他们一直热情地回答用户使用 CLAPACK 遇到的各类问题。

网址: <http://icl.cs.utk.edu/lapack-for-windows/clapack/index.html>

论坛网址: <http://icl.cs.utk.edu/lapack-forum/index.php>

(3) 最重要的是要感谢 CLAPACK 团队和 CMAKE 团队提供的开源软件包。

六、作者信息:

Kevin Wong

Email: janpo@sohu.com

2010.03.24