

使用 MATLAB 获取地震前兆 数据库中观测数据¹

徐沁¹⁾ 郑宁宁¹⁾ 黄经国¹⁾ 徐平²⁾

1) 中国地震台网中心, 北京 100045

2) 北京市地震局, 北京 100089

摘要 地震前兆数据库中的观测数据具有极高的科研及应用价值。MATLAB 作为具有强大功能的科学计算及分析平台是地震前兆数据分析的得力工具。虽然如此, 但 MATLAB 提供的数据库工具箱并不能完全操控基于 ORACLE 的前兆数据库。因此, 直接通过 MATLAB 获取前兆观测数据进行后续的前兆统计及分析工作受到了影响。本文首先介绍了主流的数据库访问机制以及在 MATLAB 中如何实现以这些机制对前兆数据库进行访问并指出了其中的缺陷; 然后, 给出了通过组件对象模型 (COM) 的访问方式获取前兆观测数据的方法; 最后, 给出了使用该方法进行函数封装后, 在实际应用中的实例的全部源代码。通过本文提供的方法, 台站工作人员或科研人员可以快速、方便的通过 MATLAB 获取前兆数据库中的相关信息进行科研及日常管理工作。

关键词: 前兆数据库 MATLAB 观测数据

引言

我国的地震前兆数据库是指除地震学以外的地磁、地电、地下流体、重力和定点形变等多学科固定台站观测数据的管理及共享平台 (周克昌等, 2007)。前兆观测系统所获得的观测数据通过台站、区域中心、学科中心及国家中心的四级体系进行汇集、管理、交换及应用。目前, 我国的地震前兆数据库采用 ORACLE 数据库系统 (周克昌等, 2010)。

MATLAB 是 Mathworks 公司开发的大型数学计算应用软件系统。它提供了强大的矩阵计算和可视化功能, 是科学研究及工程应用方面的主流软件 (Mathworks, 2011)。目前的最新版本是 2011 年 9 月发布的 2011b。MATLAB 包含了一系列丰富的称为工具箱 (Toolbox) 的专用函数库, 是针对不同学科和不同专业应用的得力工具。本文重点应用的是数据库工具箱 (Database Toolbox)。中国地震局地球物理研究所已经购买了 MATLAB 的行业专用版本。

通过前兆数据库获取观测数据是进行科学研究或日常管理必须要掌握的一项技术。尽管现有的前兆台网数据管理系统可以进行手工下载, 也可以采用行业专用软件或第三方开发语

1 基金项目 GNSS 系列数据产品在地震监测预报中的应用 (2012419006) 资助

[收稿日期] 2013-04-15

[作者简介] 徐沁, 女, 生于 1985 年。助理工程师。主要从事前兆数据观测和期刊编辑出版工作。E-mail: 314577461@163.com

言编程获取。但使用 MATLAB 访问数据库可以以灵活多样的方式获取高度定制化的观测数据来开展包括研究在内的各项工作，并且使得这些数据处理及研究流程能够在同一平台下无缝整合。但是，由于数据库工具箱在处理 ORACLE 数据库上的某些局限性以及涉及到对数据库访问机制缺少了解，导致使用 MATLAB 访问前兆数据库在地震前兆科研管理领域并不普及，台站工作人员及科研人员缺少一条可以快速、方便的通过 MATLAB 获取前兆数据库中的相关数据信息进行科研及日常管理工作的捷径。

本文首先介绍包括 ODBC、JDBC 以及 OLE-DB 等主流的数据库访问机制以及在 MATLAB 中如何实现以这些机制对前兆数据库进行访问，并指出其应用在基于 ORACLE 前兆数据库中的局限性；然后给出通过组件对象模型（COM）的访问方式获取前兆观测数据的方法；最后给出对该方法进行函数封装后在实际应用中的实例。通过本文提供的方法，可以为从事地震前兆数据研究及管理工作的科研及台网工作人员提供更多、更好的数据访问方式，使得利用地震前兆数据进行后续分析更加方便快捷，同时也为地震前兆数据库在防震减灾工作中发挥更好的作用提供了参考。

1 主流的数据库访问机制

目前通常使用的数据库访问机制有以下三种：

- (1) ODBC(Open DataBase Connectivity)¹
- (2) JDBC (Java Data Base Connectivity)²
- (3) OLE-DB (Object Linking and Embedding DataBase)³

除了上述三种主流的数据库访问机制外，还有 DAO (Data Access Object) 以及 RDO (Remote Data Object) 等数据库访问机制。

ODBC 是一种底层的访问技术，因此 ODBC API 可以使客户应用程序能从底层设置和控制数据库，用来完成一些高级数据库技术无法完成的功能。但不足之处在于 ODBC 只能用于关系型数据库，使得利用 ODBC 很难访问对象数据库及其它非关系数据库。

JDBC 是一种用于执行 SQL 语句的 Java API，可以为多种关系数据库提供统一访问，它由一组用 Java 语言编写的类与接口组成。JDBC 提供了一种基准，据此可以构建更高级的工具和接口，使数据库开发人员能够编写数据库应用程序。

OLE-DB 是一种底层数据访问界面接口，其依赖于组件对象模型（COM）。它不仅包括微软资助的标准数据接口开放数据库连通性（ODBC）的结构化问题语言（SQL）能力，同时还具有面向其他非 SQL 数据类型的能力。OLE-DB 是 ADO 的基础技术，同时还是 ADO.NET 的数据源。

2 MATLAB 中的数据库访问机制

在开始这部分之前，请确认已经在操作系统中安装了相应的数据库驱动程序。针对地震前兆数据库系统采用的 ORACLE 数据库，需要下载相应版本的客户端，笔者推荐使用 Oracle Client 10g Express Edition。MATLAB 需要确认安装数据库工具箱（Database Toolbox），本文

1 Microsoft <http://msdn.microsoft.com/en-us/library/ms710252%28VS.85%29.aspx>

2 ORACLE, <http://www.oracle.com/technetwork/java/javase/jdbc/index.html#corespec40>

3 Microsoft, <http://msdn.microsoft.com/en-us/library/ms722784%28VS.85%29.aspx>

使用 2011 年 9 月发行的 2011b 版本的 MATLAB。下面分别介绍三种主流访问机制在 MATLAB 下如何实现。

2.1 ODBC 模式访问

在使用 ODBC 机制访问数据库之前，首先需要配置 ODBC 源。在 WINDOWS 的控制面板->管理工具里打开数据源（ODBC），或者在运行里键入 `odbcad32` 命令直接启动 ODBC 数据源管理器。如图 1 所示，在系统 DSN 的选项卡中点击添加按钮



图 1 打开 ODBC 数据源管理界面

Fig.1 The administrator interface of opening the ODBC data source

出现创建新数据源对话框，如图 2 所示，选择 Microsoft ODBC for Oracle 驱动后点击完成。



图 2 选择数据库驱动

Fig.2 Selecting the database driver

如图 3 所示，出现 Microsoft ODBC for Oracle Setup 对话框。在服务器一栏填写数据库服务器的 IP 地址以及实例，如 255.255.255.255/PDBDC。数据源名称和描述一栏填写便于记

忆的任何符合规则的字符串。**用户名称**可以填写也可以在后续对话框中填写。

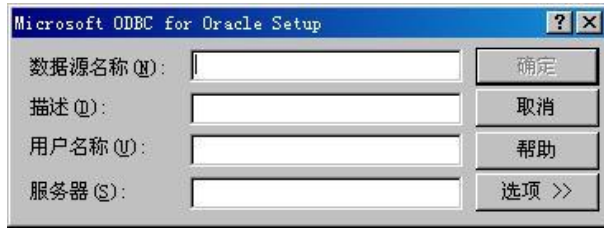


图3 ODBC 设置

Fig.3 ODBC setup

在完成配置后自动测试连接是否有效。随后在 MATLAB 的 **Command Window** 中键入如下代码：

```
conn=database('你取的数据源名称','用户名','密码');
ping(conn)
```

这里使用 **ping** 命令获得数据库连接的状态信息。如果连接成功，则打印如下信息，请注意最后一行 **AutoCommitTransactions** 的状态是 **True**，代表可以对数据库进行操作：

```
DatabaseProductName: 'Oracle'
DatabaseProductVersion: '00.00.0000 Oracle Database 10g Enterprise Edition
Release 10.2.0.3.0 - 64bi'
JDBCDriverName: 'JDBC-ODBC Bridge (msorcl32.dll)'
JDBCDriverVersion: '2.0001 (02.575.1132)'
MaxDatabaseConnections: 0
CurrentUserName: 'QZDATA'
DatabaseURL: 'jdbc:odbc:ddd'
AutoCommitTransactions: 'True'
```

2.2 JDBC 模式访问

JDBC 模式访问适合在 Linux 系统或者没有安装 ORACLE 数据库客户端的情况下访问数据库。由于 MATLAB 对 Java 的支持良好，JDBC 也是数据库工具箱推荐的访问模式，并且支持多种类型的数据库访问。在使用 JDBC 模式访问时同样需要配置 JDBC 驱动程序。配置过程如下：

(1) 找到 Oracle Client 10g Express Edition 所在的安装目录中的 **ojdbc14.jar** 文件：

一般位于...\XEClient\jdbc\lib

拷贝到如下文件目录下：

一般位于...\MATLAB\R2011b\java\jar\toolbox

(2) 打开如下目录中的 **classpath.txt** 文件

一般位于...\MATLAB\R2011b\toolbox\local

并在该文件中最后一行添加如下语句并保存：

```
$matlabroot/java/jar/toolbox/ojdbc14.jar
```

完成配置后，在 MATLAB 的 **Command Window** 中键入如下代码测试连接是否有效：

```
conn = database('数据库实例名','用户名','密码','oracle.jdbc.driver.OracleDriver','
```

```
jdbc:oracle:thin:@IP 地址:端口号:');
ping(conn)
```

如果连接成功则返回有效值。打印信息如下：

```
DatabaseProductName: 'Oracle'
DatabaseProductVersion: [1x128 char]
JDBCDriverName: 'Oracle JDBC driver'
JDBCDriverVersion: '11.2.0.1.0'
MaxDatabaseConnections: 0
CurrentUserName: 'QZDATA'
DatabaseURL: 'jdbc:oracle:thin:@10.5.67.11:1521:PDBQZ'
AutoCommitTransactions: 'True'
```

2.3 OLE-DB 模式访问

OLE-DB 模式非常适合在 WINDOWS 系统下多类型的数据库访问，通过其上层的 ADO/ADO.NET 等高级组件技术可获得很高的数据库访问效率。利用 OLE-DB 模式访问，同样需要驱动的支持。安装完 Oracle Client 10g Express Edition 后就产生 **OraOLEDB10.dll** 驱动。OLE-DB 在使用过程中需要在 MATLAB 中创建 ADO 对象，其代码如下：

```
cn = actxserver('ADODB.Connection');
```

然后针对 Oracle 数据库使用 `invoke` 函数写入连接字符串以及连接信息：

```
invoke(cn, 'Open', 'Provider=OraOLEDB.Oracle;Data Source=数据库 IP/实例数据库;
User Id=用户名;Password=密码');
```

最后设定保持连接时间和数据库操作状态：

```
set(cn, 'CommandTimeout', 60)
invoke(cn, 'BeginTrans')
```

测试并查看是否已经连接使用如下命令，如果返回值为 1，则连接成功：

```
cn.stat
```

通过 **Workspace** 可以查看 `cn` 的工作状态，如图 4 所示：

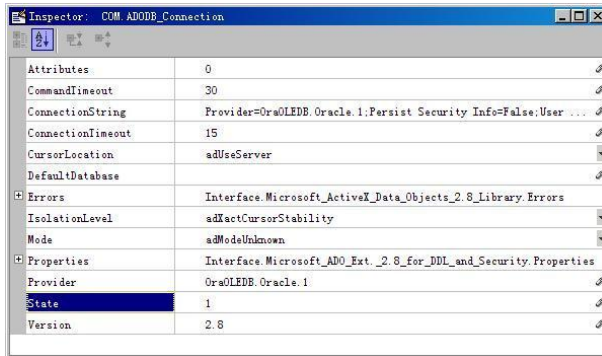


图 4 ADO 组件状态

Fig. 4 ADO module mode

本文着重介绍 ORACLE 数据库连接的方法，但是如果想要访问更多种类的数据可以通

过修改连接字符串 Provider 的方式, 以下是常用的数据库连接字符串:

连接 Access 数据库

```
Provider=Microsoft.Jet.OLEDB.4.0;Data Source=X:\mydatabase.mdb;
```

连接 Access2007 数据库

```
Provider=Microsoft.ACE.OLEDB.12.0;Data Source=X:\myAccess2007file.accdb;
Persist Security Info=False
```

连接 SQL Server 2008 数据库

```
Provider=SQLNCLI10;Server=服务器 IP;Database=数据库名称;Uid=用户名; Pwd=密码;
```

连接 MySQL 数据库

```
Provider=MySQLProv;Data Source=数据库 IP/数据库名称;User Id=用户名;Password=密码;
```

连接 IBM DB2 数据库

```
Provider=IBMDADB2;Database=数据库名称;Hostname=数据库 IP;Protocol=TCPIP;
Port=50000;Uid=用户名;Pwd=密码;
```

3 获取数据

通过上述讨论, 已经建立了 MATLAB 下访问前兆数据库的三种方法, 在成功获取连接信息后就需要获得感兴趣的数据信息。

首先介绍如何通过 ODBC 和 JDBC 方式获取数据。使用 MATLAB 的数据库工具箱操作数据库并获取数据, 通常是在建立完 ODBC 和 JDBC 连接得到连接对象后, 统一使用 `exec` 和 `fetch` 命令获取数据, 具体格式如下:

返回游标=`exec` (连接对象, sql 语句)

获取数据的结构体=`fetch` (返回的游标)

最终数据=获取数据的结构体.data

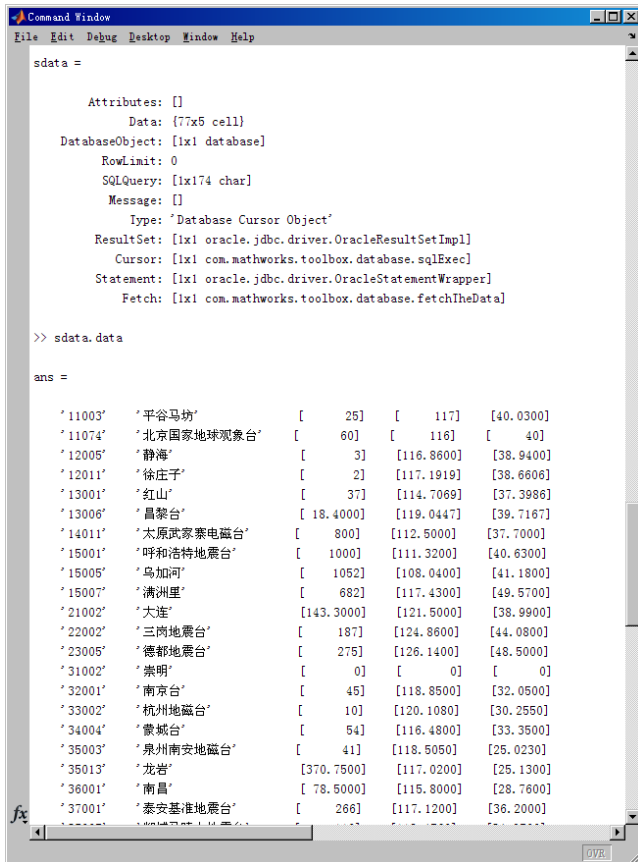
这里涉及到的 SQL 语句需要符合标准 SQL。

下例是使用该方法获取前兆数据库中所有产出秒采样数据的地磁台站的名称、经纬度等信息:

```
conn = database('数据库实例','用户名','密码','oracle.jdbc.driver.OracleDriver','
jdbc:oracle:thin:@IP地址:1521:');%获得连接
sql='select distinct(a.stationid), b.stationname, b.ALTITUDE, b.LONGITUDE,
b.LATITUDE from qz_312_dys_02 a, qz_dict_stations b where a.stationid=b.
stationid order by a.stationid';%得到台站名称及经纬度信息的 sql 语句
curs=exec(conn,sql);%执行 sql 语句返回游标
sdata=fetch(curs)%获得数据的游标结构体
data=sdata.data;
```

可以查看到 `sdata` 游标结构体中的 `data` 中有 77 行数据, 即目前全国产生秒数据的台站有 77 个, 每行 5 列, 分别是台站编号、经纬度、海拔高度等。游标结构体及最终结果如图 5 所示:

在 MATLAB 中还提供了图形交互形式的数据库查询获取方式。其命令是在 **Command Window** 中输入 `querybuilder` 启动图形界面, 在配置好的 ODBC 连接中选取特定的库和表进行交互式查询, 如图 6 所示。通过 2.1 小节中的 ODBC 配置好 `ddd` 的连接后, 选择表以及字段进行查询并指定变量存储结果。



```

Command Window
File Edit Debug Desktop Window Help

sdata =

    Attributes: []
           Data: {77x5 cell}
 DatabaseObject: [1x1 database]
      RowLimit: 0
      SQLQuery: [1x174 char]
      Message: []
      Type: 'Database Cursor Object'
      ResultSet: [1x1 oracle.jdbc.driver.OracleResultSetImpl]
      Cursor: [1x1 com.mathworks.toolbox.database.sqlExec]
      Statement: [1x1 oracle.jdbc.driver.OracleStatementWrapper]
      Fetch: [1x1 com.mathworks.toolbox.database.fetchTheData]

>> sdata.data

ans =

'11003' '平谷马坊' [ 25] [ 117] [40.0300]
'11074' '北京国家地球观察台' [ 60] [ 116] [ 40]
'12005' '静海' [ 3] [116.8600] [38.9400]
'12011' '徐庄子' [ 2] [117.1919] [38.6606]
'13001' '红山' [ 37] [114.7069] [37.3986]
'13006' '昌黎台' [ 18.4000] [119.0447] [39.7167]
'14011' '太原武家寨电磁台' [ 800] [112.5000] [37.7000]
'15001' '呼和浩特地震台' [ 1000] [111.3200] [40.6300]
'15005' '乌加河' [ 1052] [108.0400] [41.1800]
'15007' '满洲里' [ 682] [117.4300] [49.5700]
'21002' '大连' [143.3000] [121.5000] [38.9900]
'22002' '三岗地震台' [ 187] [124.8600] [44.0800]
'23005' '德都地震台' [ 275] [126.1400] [48.5000]
'31002' '崇明' [ 0] [ 0] [ 0]
'32001' '南京台' [ 45] [118.8500] [32.0500]
'33002' '杭州地震台' [ 10] [120.1080] [30.2550]
'34004' '蒙城台' [ 54] [116.4800] [33.3500]
'35003' '泉州南安地震台' [ 41] [118.5050] [25.0230]
'35013' '龙岩' [370.7500] [117.0200] [25.1300]
'36001' '南昌' [ 78.5000] [115.8000] [28.7600]
'37001' '泰安基准地震台' [ 266] [117.1200] [36.2000]

```

图 5 使用 JDBC 方式获取的台站信息数据

Fig.5 Station information data obtained using JDBC

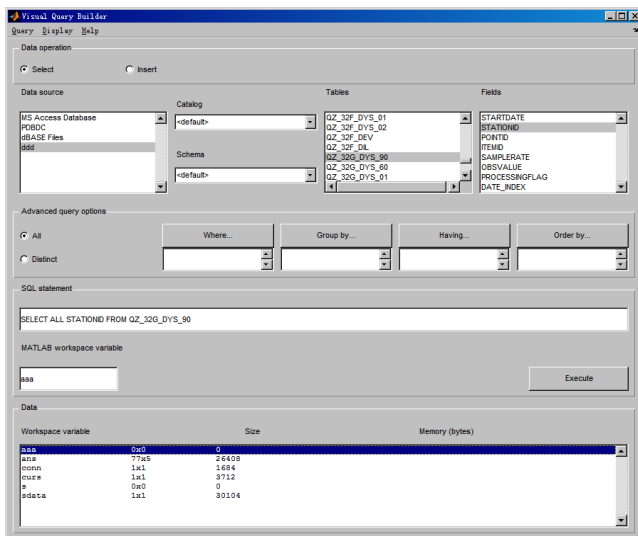


图 6 交互式方式获取数据

Fig.6 Interface of getting data

通过数据库工具箱以及 ODBC 和 JDBC 方式可以获取一部分前兆数据库中的信息,但是在获取观测数据方面却可能遇到困难。其原因主要包括:首先,观测数据在 ORACLE 数据库中使用了用于存储大对象的数据类型 (LOB) 中的 CLOB 类型字段,使得观测数据的存储空间增大至 4GB,同时在如此大对象的处理上需要进行转换处理,比如 java 中需要转化为字节流形式或者采用 oracle objects for ole 方式;其次, MATLAB 中的数据库工具箱不支持 LOB 字段的读取。在数据库工具箱中仅支持表字段数据类型: BOOLEAN、CHAR、DATE、DECIMAL、DOUBLE、FLOAT、INTEGER、LONGCHAR、NUMERIC、REAL、SMALLINT、TIME、TIMESTAMP、TINYINT、VARCHAR、NTEXT (Mathworks)。因此,采用 MATLAB 提供的数据库工具箱不能获得前兆数据库中存储观测数据的字段类型 (Mathworks, 2001)。

采用 COM 组件形式调用 ADO 组件并以 OLE-DB 方式访问数据就可以解决这样的问题,关键在于:在以 OLE-DB 方式调用 ADO 组件的过程中, CLOB 字段可以以字节流的形式进行转化,在 MATLAB 中以字符的形式展示。因此,获取 CLOB 观测数据后只需要将字符串转化为双精度数据即可。

下面以查询某台站、某个测点和某天的 GM4 原始观测数据为例予以说明。

SQL 语句如下:

```
sql= select startdate,stationid,pointid,itemid,obsvalue from qz_312_dys_02 where  
stationid='12005' and pointid='4' and startdate=to_date('2010-10-09','yyyy-mm-dd')
```

请注意,该 SQL 语句包含引号等特殊字符符号,在 MATLAB 中需要进行调整,即在单引号处转换为三个单引号,或者可以采用本文推荐的方式,即使用 SQL 语句在完成测试后写入文本文件,之后再由 MATLAB 中使用文本读取函数读入,避免由于字符串输入错误等原因导致操作失败。同时在文本文件中保存常用的 SQL 语句也有利于提高工作效率。

具体代码如下:

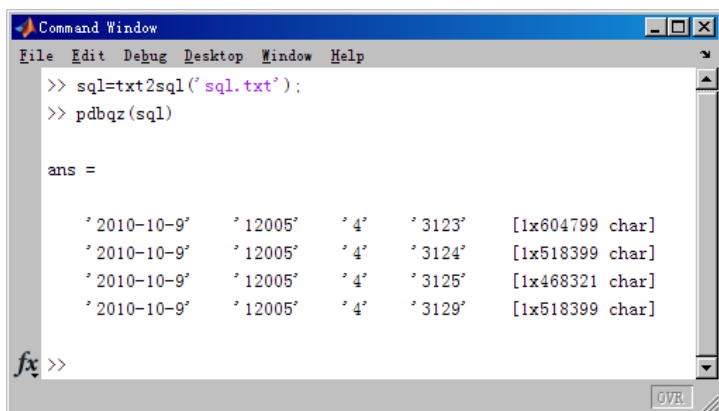
```
cn = actxserver('ADODB.Connection');  
invoke(cn,'Open','Provider=OraOLEDB.Oracle;Data Source=数据库 IP/实例数据库;  
User Id=用户名;Password=密码');  
set(cn,'CommandTimeout',60)  
invoke(cn,'BeginTrans')  
r = invoke(cn,'Execute',sql);  
invoke(cn,'CommitTrans');  
r.recordcount>0  
x=invoke(r,'getrows');  
x=x';  
invoke(r,'release');
```

通过 MATLAB 运行后得到如图 7 所示的结果:

可以看到,最后得到的结果是细胞数组五行五列,分别是日期、台站编号、测向分量以及观测数据。通过对细胞数组的操作及转化可以得到具体的观测数据值。如:

```
cell2mat(ans(1, 5))
```


得到了 H 分量的观测值。



```

Command Window
File Edit Debug Desktop Window Help
>> sql=txt2sql('sql.txt');
>> pdbqz(sql)

ans =

'2010-10-9' '12005' '4' '3123' [1x604799 char]
'2010-10-9' '12005' '4' '3124' [1x518399 char]
'2010-10-9' '12005' '4' '3125' [1x468321 char]
'2010-10-9' '12005' '4' '3129' [1x518399 char]

fx >>
  
```

图 7 运行实例

Fig.7 Instance of run

4 实际应用

本章给出包装成函数的数据库连接及操作的源代码，通过本章的实例，可以完成使用 MATLAB 读取前兆数据库中数据的全过程。

首先给出封装成函数（pdbqz.m）的数据库连接及操作的源代码。该函数可以直接在 MATLAB 中调用：

```

function x=pdbqz(sql)
cnstr='Provider=OraOLEDB.Oracle;Data Source=IP 地址/ 实例数据库;User Id=用户名;
Password=密码;';
cn = actxserver('ADODB.Connection');
set(cn,'CursorLocation',3);
invoke(cn,'Open', cnstr);
set(cn,'CommandTimeout',60);
invoke(cn,'BeginTrans');
try
    r = invoke(cn,'Execute',sql);
    invoke(cn,'CommitTrans');
    sclSuccess = 1;
catch
    invoke(cn,'RollbackTrans');
    sclSuccess = 0;
end
if r.recordcount>0
    x=invoke(r,'getrows');
    x=x';
else
    x=[];
end
invoke(r,'release');
  
```

下面使用该函数获取泰安基准台(台站编号 37001)从 2010.1—2010.6 半年内的体应变观

测数据。

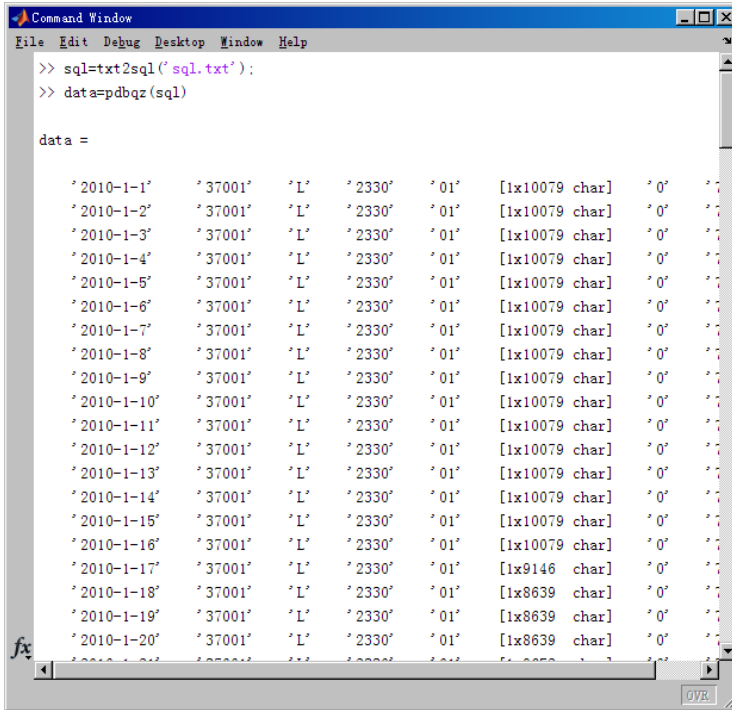
SQL 语句如下:

```
select * from qz_233_dys_01 where stationid=37001 and pointid='L' and startdate
between to_date('2010-01-01','YYYY-MM-DD') and to_date('2010-06-30','YYYY-MM-DD')
```

代码如下:

```
sql=txt2sql('sql.txt');
data=pdqbz(sql)
```

图 8 是运行结果:



```
Command Window
File Edit Debug Desktop Window Help
>> sql=txt2sql('sql.txt');
>> data=pdqbz(sql)

data =

'2010-1-1' '37001' 'L' '2330' '01' [1x10079 char] '0' '0'
'2010-1-2' '37001' 'L' '2330' '01' [1x10079 char] '0' '0'
'2010-1-3' '37001' 'L' '2330' '01' [1x10079 char] '0' '0'
'2010-1-4' '37001' 'L' '2330' '01' [1x10079 char] '0' '0'
'2010-1-5' '37001' 'L' '2330' '01' [1x10079 char] '0' '0'
'2010-1-6' '37001' 'L' '2330' '01' [1x10079 char] '0' '0'
'2010-1-7' '37001' 'L' '2330' '01' [1x10079 char] '0' '0'
'2010-1-8' '37001' 'L' '2330' '01' [1x10079 char] '0' '0'
'2010-1-9' '37001' 'L' '2330' '01' [1x10079 char] '0' '0'
'2010-1-10' '37001' 'L' '2330' '01' [1x10079 char] '0' '0'
'2010-1-11' '37001' 'L' '2330' '01' [1x10079 char] '0' '0'
'2010-1-12' '37001' 'L' '2330' '01' [1x10079 char] '0' '0'
'2010-1-13' '37001' 'L' '2330' '01' [1x10079 char] '0' '0'
'2010-1-14' '37001' 'L' '2330' '01' [1x10079 char] '0' '0'
'2010-1-15' '37001' 'L' '2330' '01' [1x10079 char] '0' '0'
'2010-1-16' '37001' 'L' '2330' '01' [1x10079 char] '0' '0'
'2010-1-17' '37001' 'L' '2330' '01' [1x9146 char] '0' '0'
'2010-1-18' '37001' 'L' '2330' '01' [1x8639 char] '0' '0'
'2010-1-19' '37001' 'L' '2330' '01' [1x8639 char] '0' '0'
'2010-1-20' '37001' 'L' '2330' '01' [1x8639 char] '0' '0'
```

图 8 运行结果

Fig.8 Result of run

5 结论

本文介绍了 ODBC、JDBC 以及 OLE-DB 等主流数据库的访问机制,以及在 MATLAB 中如何实现以这些机制对前兆数据库进行访问,指出了应用在基于 ORACLE 前兆数据库中的局限性;然后,给出了通过组件对象模型 (COM) 的访问方式获取前兆观测数据的方法;最后,给出了对该方法进行函数封装后在实际应用中的实例。通过本文提供的方法和源代码,可以快速、方便的通过 MATLAB 获取前兆数据库中的相关数据信息,是进行科研及日常管理工作的捷径,同时也为地震前兆数据库在防震减灾工作中发挥应有的作用提供了参考。

参考文献

周克昌, 蒋春花, 杨东梅等, 2007. 地震前兆数据库结构规范. 北京: 地震出版社.

周克昌, 蒋春花, 纪寿文等, 2010. 地震前兆数据库系统设计. 地震, (2): 143—151.

Mathworks, 2011. http://www.mathworks.cn/products/new_products/latest_features.html?s_cid=HP_FP_LR_2011a.

Retrieving Data from Earthquake Precursor Database Using MATLAB

Xu Qin¹⁾, Zheng Ningning¹⁾, Huang Jingguo¹⁾ and Xu Ping²⁾

1) China Earthquake Networks Center, Beijing 100045, China.

2) Earthquake Administration of Beijing Municipalities, Beijing 100089, China.

Abstract Earthquake precursor monitoring data in the database has a very high value in research and application. MATLAB, as a powerful tool, provides us scientific calculation and analysis of seismic precursory data analysis platform. However, MATLAB Toolbox can not be completely communicated with ORACLE databases on precursor data. It is incapable to take precursor monitoring data directly through the MATLAB precursor to subsequent statistics and analytical work. In this article we firstly describe the mainstream database access mechanism and realization in MATLAB how to access to the mechanisms of precursor database. Then we propose the component object model (COM) method of accessing precursor monitoring data, which is the last function encapsulating the methods used in the actual instances of all of the application source code. By using our method, the station staff or researchers can quickly and conveniently to obtain precursor database information.

Key words: Precursor database; MATLAB; Monitoring data