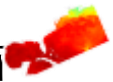
Nctoolbox 使用说明中文版

1. nctoolbox—— 一个可以读取常见数据集模型的Matlab 工具箱 

（原文网址：http://nctoolbox.github.io/nctoolbox/ ）

摘要

Nctoolbox 是一个可以读取常见数据集模型的 Matlab 工具箱。在后台，nctoolbox 使用 NetCDF-Java 作为数据访问层。这使得 nctoolbox 可以使用相同的 API 访问 NetCDF,OPeNDAP,HDF5，GRIB，GRIB2，HDF4 和许多其他（15+）文件格式和服务。它适用于 Matlab2008a 及更高版本。

开始

旧版说明和文档请点击：http://code.google.com/p/nctoolbox/（译者注：没打开+_+||| ）
我们正在将项目移植到这里托管：https://github.com/nctoolbox/nctoolbox
新的文档将在这里：http://nctoolbox.github.io/nctoolbox/

为什么使用 nctoolbox？

俩字：简单！

安装简单！
下载，运行 setup_nctoolbox。就酱紫！

使用简单！
所有类型数据集(NetCDF,HDF5,OpenDAP)的读取方式完全相同。想从 OpenDAP URL 读取数据？

```
ds=ncgeodataset('http://geoport.whoi.edu/thredds/dodsC/coawst_2_2/fmrc/coawst_2_2
_best.ncd')
time_run=ds.data('time_run');
```

想调用子集数据，按照 NetCDF 风格（例如开始，结束，间隔）？

```
lwrad=ds.data('lwrad',[10000 100 400],[10100 105 400]) % 没有间隔
lwrad=ds.data('lwrad',[10000 100 400],[10100 105 400],[1 1 1]) % 有间隔
```

想调用子集数据，按照 Matlab 风格？

```
lwrad=ds{'lwrad'}(10000:10100,100:105,400)
```

前提

MatlabR2008a +。可以通过键入以下内容来验证 Matlab 的版本：

```
version
```

Java 6+。可以通过键入以下内容来验证 Matlab 使用的 Java 版本：

```
version('-java').
```

返回的版本应以“Java 1.6”开头。如果以“Java 1.5”开头，可以尝试更新 MatlabJVM：http://www.mathworks.com/support/solutions/en/data/1-1812J/或使用本工具箱的较早版本：nctoolbox-20091112。

安装

在 Matlab 中，转到 nctoolbox 目录。例如，

```
cd ~/Documents/MATLAB/nctoolbox
```

运行 setup_nctoolbox.m 函数

```
setup_nctoolbox
```

这只会为当前的 Matlab 会话启动 nctoolbox。如果希望在启动 Matlab 时 nctoolbox 自动可用，则需要将以下几行添加到 startup.m 文件中：

```
% 更正'/Path/to/nctoolbox'到正确的 nctoolbox 目录
addpath('/Path/To/nctoolbox');
setup_nctoolbox;
```

演示

展示基本功能的演示在'demos'子目录中。如果本机无法访问远程数据 URL，演示就不能用。这些演示的集合在 http://nctoolbox.github.io/nctoolbox/demos.html（译者注：本说明后面也附上了相

关演示程序）。
显示附加功能或特殊功能的演示位于 demos/contrib 目录中。其中一些依赖于访问远程站点来获取数据，这些数据可能不如“demos”目录中的数据 URL 可靠。

文档

我们正在将文档从 GoogleCode nctoolbox documentation 迁移到 Github nctoolbox documentation。
nctoolbox 类的描述：NctoolboxClasses。
这个库包括许多有用的实用功能（译者注：本说明下一部分包含此内容）。
wiki（译者注：本说明下一部分即为此内容）中包含有 using ncdataset 和 using ncgeodataset 的文档。

引用

如果您使用了 `nctoolbox` 进行研究，请考虑在您的工作/论文等中添加一个参考文献。此项目的正式引文根据您的出版商而有所不同，但它将类似于：

```
B. Schlining,R. Signell,A. Crosby,nctoolbox (2009),Github repository,
    https://github.com/nctoolbox/nctoolbox
```

（译者注，引用部分译自 https://github.com/nctoolbox/nctoolbox ）

2. nctoolbox 百科

（原文网址：https://github.com/nctoolbox/nctoolbox/wiki）

2.1 主页

欢迎来到 nctoolbox 百科！

2.2 cfdataset

##CFDATASET 类
cfdataset 从 ncdataset 继承方法，所有这些方法均被高级类 ncgeodataset 继承。
####详情
使用 opendap url 或本地文件路径调用构造函数：

```
>>nc=cfdataset('http://testbedapps.sura.org/thredds/dodsC/estuarine_hypoxia/cbofs
2-1tdo/agg.nc')

nc=

 cfdataset handle

Properties:
  location: 'http://testbedapps.sura.org/thredds/dodsC/estuarine_hypoxia/cbofs2-
1tdo/agg.nc'
    netcdf: [1x1 ucar.nc2.dataset.NetcdfDataset]
 variables: {84x1 cell}

Methods,Events,Superclasses
```

获取坐标信息（在 `cfdataset` 中有时会遗漏某些轴...）：

```
>>grid=nc.grid('temp',[1 1 1 1],[1 1 50 50],[1 1 1 1])

grid=

lat_rho: [50x50 double]
lon_rho: [50x50 double]
```

获取数据集中 `temp` 变量的变量对象（请注意，它返回的是 ncvariable 而非 ncgeovariable）：

```
>>var=nc.variable('temp')

var=

 ncvariable handle

 Properties:
    dataset: [1x1 cfdataset]
       name: 'temp'
       axes: {2x1 cell}
 attributes: {7x2 cell}

Methods,Events,Superclasses

>>nc.geovariable('temp')

??? No appropriate method,property,or field geovariable for class cfdataset.
```

获取给定子集中的坐标信息和变量的值：

```
>>grid=nc.struct('temp',[1 1 1 1],[1 1 50 50],[1 1 1 1])

grid=

lat_rho: [291x332 double]
lon_rho: [291x332 double]
   temp: [4-D single]
```

2. 3 文档

##快速开始

- 运行并检查 nctoolbox/demos 目录下的演示程序或者看看这里：
  http://nctoolbox.github.com/nctoolbox/demos.html
- 使用[关于使用 NCGEODATASET 类在 NCTOOLBOX 中读取数据的教程]
  （UsingNcgeodataset）。由于关于下面的原因，这是大多数人想要读取数据的方式。

##剩下的故事

读取数据可以使用好几种数据集类，每种都提供不同类型的功能。所有数据集类都提供对变量数据、变量名列表和 netcdf 属性以及基础 netcdf-java 数据集对象的访问。大多数用户将希望使用 NCGEODATASET 类，也就是最新的类，因为只有它可以访问 geovariable 类。这种类允许按纬度

/经度值和时间窗口进行子集设置，并能够返回对每个数据集变量使用时间、经度、纬度、垂直方向标准名称的坐标（甚至可以执行坐标转换）。这种类还具有一种方法，可以将属性从 netcdf 文件提取到方便的 matlab 结构中，并可以提取坐标的范围。保留较旧的 NCDATASET 和 CFDATASET 类主要是为了后向兼容。

- nctoolbox Matlab 类
  - NCDATASET - 通过开始/停止索引提供对数据的基本访问。
    - 教程
    - 方法和功能概述
  - CFDATASET - 提供从 CF 和 COARDS 兼容数据集中获取数据的附加功能
    - 尚未完成的教程
    - 方法和功能概述
  - NCGEODATASET -提供各种高级功能来利用 Unidata 通用数据模型实现互操作性
    - 教程
    - 方法和功能概述

2.4 常见问答

问：为什么用 nctoolbox？
答：可以访问很多常见数据集模型…

```
nc=ncdataset('localfile.nc')   % NetCDF 文件
nc=ncdataset('localfile.grb')  % GRIB 文件
nc=ncdataset('localfile.grb2') % GRIB2 文件
nc=ncdataset('http://geoport.whoi.edu/thredds/dodsC/coawst_2_2/fmrc/coawst_2_2_b
est.ncd') % DAP URL
nc=ncdataset('http://www.mbari.org/staff/brian/pub/m1_pco2.nc') % NetCDF URL
```

…而且用起来也很简单

```
nc=ncdataset('http://www.mbari.org/staff/brian/pub/m1_pco2.nc')
p=nc.data('pco2')
t=nc.data('time_d')
plot(t,p)
datetick('x')
set(gca,'XLim',datenum(['2005-01-01'; '2008-01-01']))
```

###问：为什么 nctoolbox **只能读**?
###答：MatlabAPI 是可以写入的。
简而言之，Matlab 已经内置了对写入 HDF，HDF5 和 NetCDF 文件的支持。为了进行读取，尽管 Matlab 现在支持 OPeNDAP，但如果存在 CF，它就不会利用 CF 规范，这使得访问标准化数据集

变得更加困难。*nctoolbox* 通过提供非常简单直观的 API 大大简化了数据访问。下面是差异的简要：

	NCTOOLBOX	Matlab 内置 API
读取本地文件	是	是
读取网上服务器文件	是	否
读取 OpenDAP URLs	是	是
读取不同格式数据是否使用统一的 API?	是	否
易于理解	是	http://www.mathworks.com/help/techdoc/ref/netcdf.getatt.html
写入 HDF、HDF5 和 NetCDF 文件	否	是

代码比较：从一个本地 NetCDF 文件中读取变量名

Matlab 内置 NetCDF API

```
ncid=netcdf.open('m1_pco2.nc','NC_NOWRITE');
[numdims,numvars,numglobalatts,unlimdimID]=netcdf.inq(ncid);
varnames={};
for i=0:(numvars - 1)
    [varnames{i + 1,1},xtype,dimids,numatts]=netcdf.inqVar(ncid,i);
end
```

nctoolbox

```
ds=ncdataset('m1_pco2.nc');
varnames=ds.variables;
```

（译者注：个人感觉这么比较不太厚道，毕竟现在用 ncinfo 也挺简单的...）

2.5 开发者信息

分支

为了发展，我们有以下分支:

- `master`: 发布版分支（*希望如此*），其中包含我们的发布版。
- `integration`: 通常是稳定的分支，用于在将更改合并到 master 之前对其进行测试。该分支在发行版中会合并到 master 中。
- `develop`: 最新的突破性改进将在这里。

Git Hook

如果您是一名正在贡献代码的开发人员，我们使用 git hook 在 nctoolbox 中包含发行信息。请执行以下操作使用 hook（在 Mac 或 Linux 上）:

- 创建文件 `nctoolbox/.git/hooks/pre-commit`
- 加入以下：

```
#!/usr/bin/env bash

NC_COMMIT=$(git rev-parse --short HEAD)
NC_BRANCH=$(git rev-parse --abbrev-ref HEAD)
NC_MODDATE=$(git --no-pager log -1 --format=%cd --date=short)
NC_DESCRIBE=$(git describe --tags)

cat > "cdm/nctoolbox-version.txt" << EOL
GIT COMMIT:   $NC_COMMIT
GIT BRANCH:   $NC_BRANCH
GIT DATE:     $NC_MODDATE
GIT DESCRIBE: $NC_DESCRIBE
EOL

git add "cdm/nctoolbox-version.txt"
```

- 别忘了用 chmod u+x pre-commit 让它运行一下。

2.6 安装

下载 NCTOOLBOX

####大多数用户:（稳定版）

下载最近的稳定版

解压下载的压缩文件并安装 NCTOOLBOX（方法见下）。

####冒险者:（开发版）

下载最近的开发版
解压下载的压缩文件并安装 NCTOOLBOX（方法见下），或者如果想要确认是最新版的，请这么做：
克隆源代码库：

```
git clone https://github.com/nctoolbox/nctoolbox.git
```

第一次克隆之后，可以通过如下方法更新到最新版：

```
git pull origin master
```

现在，安装 NCTOOLBOX（见下）
####安装 NCTOOLBOX
在 Matlab 中，将目录调整到 nctoolbox，例如：

```
cd ~/Documents/Matlab/nctoolbox
```

在 Matlab 中，运行 setup_nctoolbox 函数

```
setup_nctoolbox
```

如果希望在每次启动 Matlab 时 nctoolbox 都可用，编辑 startup.m：

```
edit startup.m
```

添加如下行：

```
addpath('c:/change-to-your-path/nctoolbox')
setup_nctoolbox
```

2.7 ncdataset

#NCDATASET 类
用于处理 *nctoolbox* 中的 netcdf 文件的基本数据集类。很少显现 netcdf 文件的规范或结构。cfdataset 是此类的子类，并添加了几种利用 CF 规范元数据的方法。

```
>>nc=ncdataset('http://testbedapps.sura.org/thredds/dodsC/estuarine_hypoxia/cbof
s2-1tdo/agg.nc')
```

```
nc=

 ncdataset handle

 Properties:
   location:'http://testbedapps.sura.org/thredds/dodsC/estuarine_hypoxia/cbofs2-
1tdo/agg.nc'
     netcdf: [1x1 ucar.nc2.dataset.NetcdfDataset]
  variables: {84x1 cell}

 Methods,Events,Superclasses
```

```
>>temp=nc.data('temp',[1 1 1 1],[1 1 50 50],[1 1 1 1]);
>>size(temp)
```

```
ans=

 1     1    50    50



>>nc.axes('temp')

ans=

'ocean_time'
's_rho'
[]
[]
```

```
>>nc.attributes
```

```
ans=

'type'               'ROMS/TOMS history file'
'Conventions'         'CF-1.0'
'title'              'CBOFS2-1TDO (NOAA) - ROMS-3.0'
```

```
'var_info'           '../../../ROMS/External/varinfo.dat'
'rst_file'           'ocean_rst_synoptic_seg38.nc'
'his_file'           'ocean_his_synoptic_seg38.nc'
'sta_file'           'ocean_sta_synoptic_seg38.nc'
'grd_file'           'CBOFS2_grid_2mcut_CnD.nc'
'ini_file'           'ocean_rst_synoptic_seg37.nc'
...
...
'wms-link'           'http://testbedapps.sura.org/ncWMS/wms'
'wms-layer-prefix'    'cbofs2-1tdo'
'id'                 'eh.noaa.cbofs2-1tdo.2004_only'
'naming_authority'    'noaa.ioos.testbed'
'summary'                                      [1x61  char]
'creator_name'        'Lyon Lanerolle'
'creator_email'       ''
'creator_url'                                   [1x53  char]
'cdm_data_type'       'Grid'
```

```
>>nc.attributes('temp')
```

```
ans=

'long_name'        'potential temperature'
'units'           'Celsius'
'time'            'ocean_time'
'coordinates'      'lat_rho lon_rho'
'field'           'temperature,scalar,series'
'wms-layer'        'cbofs2-1tdo/temp'
'standard_name'    'sea_water_temperature'
```

```
>>nc.attribute('units','temp')
```

```
ans=

 Celsius
```

```
>>nc.size('temp')
```

```
ans=

    400         20        291        332
```

2.8 ncgeodataset

#NCGEODATASET 类

ncgeodataset 类是用于处理与本地或远程 CF 兼容地理 netcdf 数据集的高级类。使用文件系统路径或 OPeNDAP 链接作为参数调用构造函数。此类是 Nctoolbox 类中处理数据最复杂的类，它旨在通过展示以一致的方式帮助数据访问和比较的方法来实现最大的互操作性。有关使用此类的更多示例，请参见 UsingNcgeodataset。

Ncgeodataset 引用了 cfdataset 类实现的 `variable`、`attributes`、`variables`、`data`、`grid` 等方法，但还添加了自己的方法，如 `metadata` 和 `geovariable`。geovariable 方法返回 ncgeovariable，提供比 ncvariable 类更高级别和更可互操作的版本。

##ncgeodataset

是 cfdataset 的子类

```
>>nc=ncgeodataset('http://testbedapps.sura.org/thredds/dodsC/estuarine_hypoxia/c
hesroms/agg.nc')
```

```
nc=

 ncgeodataset handle

 Properties:
   location:
'http://testbedapps.sura.org/thredds/dodsC/estuarine_hypoxia/chesroms/agg.nc'
     netcdf: [1x1 ucar.nc2.dataset.NetcdfDataset]
  variables: {92x1 cell}
 Methods,Events,Superclasses
```

获取数据集的元数据：

```
>>nc.metadata
```

```
ans=

 global_attributes: {32x2 cell}
```

```
           ntimes: {'long_name'  'number of long time-steps'}
          ndtfast: {'long_name'  'number of short time-steps'}
              dt: {2x2 cell}
                    ...
               u: {7x2 cell}
               v: {7x2 cell}
               w: {5x2 cell}
            temp: {8x2 cell}
            salt: {8x2 cell}
             rho: {5x2 cell}
                    ...
```

只获取数据集的全局变量：

```
>>nc.attributes
```

```
ans=

'type'               'ROMS/TOMS history file'
'Conventions'         'CF-1.0'
'title'              'ChesROMS (UMCES) - ROMS-2.2'
'var_info'                                  [1x54  char]
'rst_file'           'chesroms_rst_Nz20.nc'
'his_base'           'chesroms_his_Nz20'
'avg_base'           'chesroms_avg_Nz20'
               ...
'creator_name'        'Wen Long'
'creator_email'       'wenlong@umces.edu'
'creator_url'         'http://www.umces.edu/'
```

创建变量对象：

```
var=nc.variable('salt');
```

创建 ncgeovariable 对象的实例：

```
>>gvar=nc.geovariable('salt')
```

```
gvar=

```

```
 ncgeovariable handle

 Properties:
    dataset: [1x1 ncgeodataset]
       name: 'salt'
       axes: {4x1 cell}
 attributes: {8x2 cell}

 Methods,Events,Superclasses
```

从特定变量/关键字（按此顺序）或全局（仅带有 1 个参数）中获取属性值：

```
>>saltunits=nc.attribute('salt','units')
```

```
saltunits=

 PSU

>>modeltitle=nc.attribute('title')

modeltitle=

 ChesROMS (UMCES) - ROMS-2.2
```

查找指定变量的边界：

```
>>nc.extent('salt')
```

```
ans=

lon: [-77.1732 -75.0920]
lat: [36.0202 39.9742]
```

这在返回一个给定地理变量的时间轴时很有用。
要找到为地理变量提供时间轴的 netcdf 变量的名称，请使用 `gettimename`。

```
>>name=nc.gettimename('u_water')
```

要返回表示给定地理变量的时间轴的地理变量对象，请使用 `gettimevar`。

```
>>t_gvar=nc.gettimevar('u_water')
```

##ncgeovariable

是 `ncvariable` 的子类，用于打开数据集，然后创建一个变量对象：

```
>>nc=ncgeodataset('http://testbedapps.sura.org/thredds/dodsC/estuarine_hypoxia/c
hesroms/agg.nc');


>>gvar=nc.geovariable('salt')
```

```
gvar=

ncgeovariable handle

Properties:
   dataset: [1x1 ncgeodataset]
      name: 'salt'
      axes: {4x1 cell}
attributes: {8x2 cell}

Methods,Events,Superclasses
```

获取坐标轴（注意，这将返回所有变量名，与 ncvariable/ncdataset/cfdataset 方法和属性相反，不带空格）：

```
>>gvar.axes
```

```
ans=

'lat_rho'
'lon_rho'
'ocean_time'
's_rho'
```

从指定关键字中获取变量属性值：

```
>>gvar.attributes
```

```
ans=
'long_name'       'salinity'
'units'           'PSU'
'time'            'ocean_time'
'coordinates'     'lat_rho lon_rho'
'field'           'salinity,scalar,series'
'missing_value'   [                    0]
'wms-layer'       'chesroms/salt'
'standard_name'   'sea_water_salinity'

>>longname=gvar.attribute('long_name')
```

```
longname=

 potential temperature

>>var.attribute('long_name')
```

```
ans=

 salinity
```

返回坐标网格，在适当坐标网格情况使用标准化的字段名称（`grid_interop` 也会进行垂直坐标转换和时间到 Matlabdatenum 的转换）：

```
>>grid=gvar.grid_interop(1:end,:,1:50,1:50)
```

```
grid=

 lat: [50x50 double]
 lon: [50x50 double]
time: [5479x1 double]
   z: [20x50x50 double]

>>oldgrid=gvar.grid(1:end,:,1:50,1:50)
```

```
oldgrid=
```

```
   lat_rho: [50x50 double]
   lon_rho: [50x50 double]
ocean_time: [5479x1 double]
     s_rho: [20x1 double]
```

基于经纬度和时间值或输入结构体字段中索引的子集：

```
>>s.time={'01-Jan-2002','01-Feb-2002'}; % Can be omitted or s.t_index=[i1 i2];
or s.time=[datenum1,datenum2];
>>s.t_stride=2; % Can be omitted
>>s.lat=[36.1 37.0];
>>s.lon=[-76.0 -75.0];
>>s.h_stride=[2 2]; % Can be omitted
>>s.z_index=[1 3]; % Can be omitted
>>s.v_stride=2; % Can be omitted
>>sub=gvar.geosubset(s); % Subset method
```

在窗口中获取时间（以 datenum 形式返回）或时间索引：

```
>>time=gvar.timewindow(731944,732272) %args in datevec or datenum form.
```

```
time=

  731944
  731945
  731946
  731947
  731948
  731949
  731950
  731951
   ...


>>timewindow=gvar.timewindowij(731944,732272);
```

```
timewindow=

index: [350x1 double]
 time: [350x1 double]
```

也可以是这样的：

```
time=timewindow(gvar,start,stop);
time2=timewindowij(gvar,start,stop);
```

获取子集边界的经纬度索引（如果经纬度是矢量，输出值是这样的：`[minlat_ind maxlat_ind minlon_ind maxlon_ind]`）：

```
[rowmin rowmax colmin colmax]=geoij(gvar,subsetstruct);
```

2.9 ncgeovariable

#ncgeovariable
是具有更复杂功能的 ncvariable 子类。
有关这些增强功能的一些工具，可以参见 NctoolboxUtilityFunctions。*ncgeodataset* 类引用 ncgeovariable。

```
>>nc=ncgeodataset('http://geoport.whoi.edu/thredds/dodsC/examples/bora_feb.nc');
>>nc.variables     % list the variables
>>gvar=nc.geovariable('salt')
```

```
gvar=

  4-D ncgeovariable array with properties:

      dataset: [1x1 ncgeodataset]
         name: 'salt'
         axes: {4x1 cell}
    attributes: {6x2 cell}
```

获取坐标轴（注意，这将返回所有变量名，与 ncvariable/ncdataset/cfdataset 方法和属性相反，不带空格）：

```
>>gvar.axes
```

```
ans=

'lat_rho'
'lon_rho'
```

```
'ocean_time'
's_rho'
```

从指定关键字中获取变量属性：

```
>>gvar.attributes
```

```
ans=

    'long_name'      'averaged salinity'
    'units'          '1'
    'time'           'ocean_time'
    'coordinates'    'lat_rho lon_rho'
    'field'          'salinity,scalar,series'
    '_FillValue'     [                    0]

>>longname=gvar.attribute('long_name')
```

```
ans=
averaged salinity
```

可以看到变量的尺寸：

```
>>size(gvar)
```

```
ans=
     8         20         60        160
```

以下使用 matlab 索引符号从地理变量对象中返回数据值：

```
>>salt=gvar.data(1:end,:,1:50,1:50);
```

以下会在合适的情况下返回使用标准情况下字段名称的坐标网格（`grid_interop` 也会进行垂直坐标转换和时间到 Matlabdatenum 的转换）：

```
>>grid=gvar.grid_interop(1:end,:,1:50,1:50)
```

```
grid=

```

```
 lat: [50x50 double]
 lon: [50x50 double]
time: [8x1 double]
   z: [4-D double]
```

现在就有需要分析或画图的信息了，例如：

```
>>pcolorjw(grid.lon,grid.lat,double(squeeze(salt(end,end,:,:))));
```

您可能会想使用 `grid_interop` 方法，但是如果出于某些原因想要查看原始变量名或查看未转换的时间值时，也可以使用 `grid` 方法：

```
>>rawgrid=gvar.grid(1:end,:,1:50,1:50)
```

```
rawgrid=

   lat_rho: [50x50 double]
   lon_rho: [50x50 double]
ocean_time: [8x1 double]
     s_rho: [20x1 double]
```

基于经纬度和时间值或输入结构体字段中索引的子集：

```
>>s.time=['12-Feb-2003' '16-Feb-2003']; % Can be omitted or s.t_index=[i1 i2];
can be used for subsetting time using indices
>>s.t_stride=2; % Can be omitted
>>s.lat=[43.7551   45.8778];
>>s.lon=[11.4735   14.4367];
>>s.h_stride=[2 2]; % Can be omitted
>>s.z_index=[1 3]; % Can be omitted
>>s.v_stride=2; % Can be omitted
>>sub=gvar.geosubset(s); % Subset method
```

在窗口中获取时间（以 datenum 形式返回）或时间索引：

```
>>time=gvar.timewindow(731623.5,731626.5) %args in datevec or datenum form.
```

```
 time=

```

```
    index: [4x1 double]
     time: [4x1 double]
```

获取子集边界的经纬度索引（如果经纬度是矢量，输出值是这样的：`[minlat_ind maxlat_ind minlon_ind maxlon_ind]`）：

```
[rowmin rowmax colmin colmax]=geoij(gvar,subsetstruct);
```

这在返回一个给定地理变量的时间轴时很有用。
要找到为地理变量提供时间轴的 netcdf 变量的名称，请使用 `gettimename`。

```
>>name=gvar.gettimename()
```

要返回表示给定地理变量的时间轴的地理变量对象，请使用 `gettimevar`。

```
>>t_gvar=gvar.gettimevar()
```

2.10 Nctoolbox 类

##大图
对于那些没有读取 UML 的来讲，意味着 cfdataset 是 ncdataset.的*子类*。（一分钟将了解子类的含义）。反过来，ncgeodataset 是 cfdataset 的子类。另外，cfdataset 引用 *ncvariable*,，而 ncgeodataset 引用 ncgeovariable。（*译者注：此处应该有 UML 图，但是图挂了……*）
因此，您可能想知道这意味着什么。简而言之，子类可以完成父类的所有工作，但它也可以做更多的事情和/或与父类做的事情略有不同。（晓得了？）。
ncdataset、 cfdataset 和 ncgeodataset 共有的方法
所有 3 个类（ncdataset、 cfdataset 和 ncgeodataset）都有以下作用*完全*相同的方法：

- *attribute* - 返回值，该值是由其关键字指定的全局属性或由关键字和变量指定的变量属性。
- *attributes* - 返回值，该值是由其关键字指定的全局属性或由关键字和变量指定的变量属性。
- *axes* - 返回一个元胞数组，其中包含给定变量坐标轴的变量名。
- *data* - 获取给定变量的数据。此方法还允许通过指定第一个和最后一个点的索引以及跨度（或间隔）来获取数据的子集。
- *mdata* -（仅限于开发版）获取给定变量的数据。此方法还允许通过使用类似 matlab 的语法指定 hyperslab 来获取数据的子集。
- *save* - 将数据保存到本地 netcdf 文件。
- *size* – 在不获取数据的前提下返回永久存储中变量的尺寸，帮助您了解自己正在干什么。 ;-)
- *time* - 尝试将数据转换为 Matlab 的本地时间格式。

ncgeodataset、cfdataset 具有的相同方法。（但是 ncdataset 不是）===

cfdataset 添加了一些 CF 和 COARDS 规范的方法；ncgeodataset 继承了这些，因此它也具有相同的方法。这些方法使得运用遵循这些规范的数据集更加方便。这些新方法是：

- *grid* - 检索变量的全部或部分坐标数据。返回值是包含一个有数据每个维度的变量的结构体数据。不返回变量的数据，仅返回变量的坐标数据！！！！
- *standard_name* - 返回变量的 *standard_name*（如果存在）（通常这是变量的“ standard_name”属性）。
- *struct* - 检索给定变量的全部或部分数据。数据以结构体形式返回，包含数据以及数据的每个维度的变量。

####问：UML 显示 cfdataset 引用 *ncvariable* 对象。这些是做什么用的？
####答：您可以直接使用它们，但老实说，我不会。*cfdataset* 在内部使用 ncvariables 将 *grid* 和 *struct* 的数据分组。

ncgeodataset 独有的方法

ncgeodataset 添加了一些功能，这些功能使得使用基于地理坐标系的数据集变得更加容易。*它还支持 Matlab 风格的矩阵索引，而 ncdataset 和 cfdataset 没有*。它支持的其他方法是：

- *metadata* - 一次获取所有属性（全局和变量）的功能。
- *timextent* - 用于计算变量的开始和结束时间的函数。

请原谅，我们正在努力更新文档

2.11 Nctoolbox 实用程序功能

##简介

nctoolbox 具有一组和基本的 matlab 风格一样有用的函数，用来处理地理空间数据的不同方面。这些 函数 包括（未特定排序）：

- `interptime`
    - 用于将最多 4 个数据矩阵（包括时间）内插到特定时间或重新采样的时间数组。

像这样：

```
>>data=interptime(data,datatime,[2005 1 1 0 0 0]);
```

- `zsliceg`
    - 特定垂直层面的水平切片。
- 像这样：

```
>>varz=zsliceg(data,z,depth);
```

- `getinterp`
    - 这为输入数据矩阵的内插提供了 Matlab 内插方式。
  - 像这样：

```
>>int=getinterp(data,lon,lat,method);
```

- `interptoxy`
    - 将 2 或 3 阶数据矩阵插值到一个 xy 点或点阵。（3 阶矩阵会在该点返回一个剖面）
  - 像这样：

```
>>data=interptoxy(data,datalon,datalat,-77,45,'linear');
```

- `vsliceg`
    - 沿着一条 xy 坐标直线的垂直切面。
  - 像这样：

```
>>[X,Z,VDATA]=vsliceg(DATA,GRD,CX,CY);
```

- `nc_genslice`
    - 沿 xy 和时间获取垂直切片，用于将轨迹（像滑翔机飞过一样）建模。
  - 像这样：

```
[Tvar,Tdis,Tzed,Tlon,Tlat]=nc_genslice(file,varname,lonTrk,latTrk,timTrk)
```

基本上像 GSLICE 一样工作。在由 lon、lat 和 time 的向量指定的坐标处（例如，船舶横断面或滑翔机轨迹）从 ROMS 输出文件中获取“通用”垂直切片。
输入：file - netcdf 或 opendap url（跨时间聚集）；varname – 想追踪的变量名；lonTrk,latTrk – 轨迹的经纬度坐标；timTrk – 沿着轨迹的时间变量，以 Matlabdatenum 格式给出。其必须和经纬度维度一致，如果假设轨道是单一时间点，可以是单变量。输出：Tvar – 沿着轨迹的内插二维数据；Tdis – 沿着轨迹的二维距离或范围（单位：km）；Tzed – 沿着轨迹的二维深度；Tlon/Tlat – 沿着轨迹的二维经纬度。（译者注：此处原文较混乱，译者根据自己理解翻译的。）
gslice 原版由 John Evans 制作，由 Weifeng Gordon Zhang 改写。此版本由 John Wilkin 于 2011-01-26 和 Alexander Crosby 于 2011 年 4 月改写。

- `ncunits`
    - 将数据矩阵从一组已知单位转换为另一组单位（指定为字符串）时非常有用的功能。
  - 像这样：

```
>>converted_data=ncunits(data,'m/s','cm/s');
```

- `ModelLook`
    - 这是一个便于模型可视化和探索的框架。它主要是提供了几种不同的方式来了解 CF 兼容的 netcdf 文件中的 2、3、4 维变量。目前，这仅适用于规则网格化模型数据，但不久之后它将也可用于非结构化网格。调用 openDap 链接作为第一个参数，并使用 Matlabdatevec 或 datenum 作为您要查看的时间（在模型中查找到最近的时间）。
    - 对于 3 维和 4 维网格变量，将绘制索引 1 处的图层。有时这和模型的表面层并不对应，以下例子就是这种情况。
  - 像这样：

```
>>modellook('uri','http://testbedapps.sura.org/thredds/dodsC/estuarine_hypoxia/ch
3d/agg.nc','date',[2004 1 10 21 0 0],'level',1);
```

2. 12 发行说明

#发行说明

###nctoolbox-20130131

此版本是正式托管在 github 上的第一个稳定版本。它包含一些错误修复，但与 2012 年以前的版本没有太大不同：
http://code.google.com/p/nctoolbox/downloads/list?can=1&q=&colspec=Filename+Summary+Uploaded+ReleaseDate+Size+DownloadCount

---

###nctoolbox-20110616-alpha2

- 添加了 *ncgeodataset* 类和 *ncgeovariable* 类。
- 使用 ncgeodataset 可使 NCTOOLBOX 用户使用 NetCDF-java 工具箱（njTBX）的大部分功能。.
- 修复了内部报告的问题：带有特殊字符的变量名称会导致异常。现在可以*根据需要*对 `ncdataset` 中的变量名进行转义。
- 添加了对 matlab 风格索引和 mexcdf 风格变量访问的支持。.
- 指定数据集位置的 *location* 属性已添加至 ncdataset。
- 为处理 Netcdf4 组所做的更改。
- 添加了“save”方法，该方法允许 nctoolbox 将数据从 OpeNDAP 服务器的远程文件或数据集中保存到 NetCDF 本地文件。

###nctoolbox-20101101

- 修正了[http://code.google.com/p/nctoolbox/source/browse/cdm/ncdataset.m ncdataset.m]中导致 [http://code.google.com/p/nctoolbox/source/browse/demos/demo5.m demo5.m]尝试将时间转换为 Matlab 格式时浏览失败的问题。
- 修正了[http://code.google.com/p/nctoolbox/issues/detail?id=8&can=1 issue] 中 [http://code.google.com/p/nctoolbox/source/browse/cdm/ncdataset.m ncdataset.m]在 MatlabR2008a 对“删除”方法访问权限不正确的问题。
- 更新到最新的 NetCDF-Java 4.2。

###nctoolbox-20100305

- 增加了对从兼容 CF/COARDS 的数据集中获取坐标数据的支持（不是变量的数据，只有变量的坐标数据！对于子集非常有用）。新方法是 *cfdataset.grid*，与 *cfdataset.struct* 类似（但 *struct* 同时返回坐标数据和变量数据）。参见演示程序 [http://code.google.com/p/nctoolbox/source/browse/demos/demo9.m demo9.m]和 [http://code.google.com/p/nctoolbox/source/browse/demos/demo9a.m demo9a.m]。
- *grid* 和 *struct* 对比的例子：

```
>>ds=cfdataset('http://dods.mbari.org/cgi-bin/nph-
nc/data/ssdsdata/deployments/m1/200810/OS_M1_20081008_TS.nc');
>>g=ds.grid('TEMP')
```

```
g=

    TIME: [9043x1 double]
   DEPTH: [11x1 double]
 LATITUDE: 36.7640
LONGITUDE: -122.0460
```

```
>>s=ds.struct('TEMP')
```

```
s=

    TEMP: [9043x11 single]
    TIME: [9043x1 double]
   DEPTH: [11x1 double]
 LATITUDE: 36.7640
LONGITUDE: -122.0460
```

###nctoolbox-20100223

- 添加了对配置网络代理的支持。请参见 [http://code.google.com/p/nctoolbox/source/browse/cdm/setproxy.m setproxy.m]
- 修正了[http://code.google.com/p/nctoolbox/issues/detail?id=6 issue]转换时间了时其单位超出公历的时间的问题。

###nctoolbox-20100128

重要说明 *– 从 nctoolbox-20100128 版本开始，需要 Java 6+。您可以使用{{{version（'-java'）}}}来验证 matlab 正在运行的 Java 版本。返回的版本应以 Java 1.6 开头。如果运行的是 Java 5，则可以改用 nctoobox-20091112。*

- 在[http://code.google.com/p/nctoolbox/source/browse/cdm/cfdataset.m cfdataset.m]中添加了一个 *struct* 方法。此方法返回一个由变量的数据及其关联的坐标变量数据组成的结构体。它取代了将在下一版本中将变为私有的 *cfdataset.variable* 方法。
    - 参见[http://code.google.com/p/nctoolbox/source/browse/demos/demo9.m demo9.m]中有关 *struct* 用法的一个例子。
- 修正了[http://code.google.com/p/nctoolbox/source/browse/cdm/cfdataset.m cfdataset.m]和[http://code.google.com/p/nctoolbox/source/browse/cdm/ncvariable.m ncvariable.m]中的错误文档。
- 添加了私有 *delete* 方法，当 Matlabncdataset 对象被清除或超出范围时，它将关闭基础 Java *Netcdf* 对象。这样会正确关闭 Java 资源，而不需要更改用户代码。
- 从 netcdf-java 4.0 升级到 4.1。此更改要求 Matlab 安装并使用 Java 6。此更新解决了我们在 MBARI 内部遇到的一些访问错误。

###nctoolbox-20091112

- 解决了一个[http://code.google.com/p/nctoolbox/issues/detail?id=1&can=1 bug]，该错误导致 *cfdataset.variable* 无法使用与 CF 兼容的数据集。
- 更改 *cfdataset.variable* 的输出：
    - nctoolbox-20091022 中的**旧方法**:*t=ds.variable* 产生包含 *metadata* 和 *data* 结构体的嵌套结构
    - nctooblox-20091112 中的**新方法**：*t=ds.variable* 仅生成包含数据的结构。您可以使用 *ds.name* 和 *ds.axes* 查询 t 中的元数据。
        - 请参阅 [http://code.google.com/p/nctoolbox/source/browse/demos/demo7.m demos7.m]和 [http://code.google.com/p/nctoolbox/source/browse/demos/demo8.m demos8.m] 有关 *cfdataset.variable* 用法的示例。

###nctoolbox-20091022

- 增加了使用 *cfdataset.variable* CF 或 COARDS 智能兼容数据集的功能。下设一个变量也将返回正确下设的坐标轴。有关其用法的示例，请参阅 [http://code.google.com/p/nctoolbox/source/browse/demos/demo7.m demos7.m]。

2.13 UsingCfdataset

##简介

请原谅，我们正在努力更新文档

2.14 使用 Ncdataset

##打开一个数据集

```
>>ds=ncdataset('http://dods.mbari.org/opendap/data/ssdsdata/deployments/m0/200706
/OS_MBARI-M0_20070621_R_TS.nc')
```

```
ds=

 ncdataset handle

 Properties:
    netcdf: [1x1 ucar.nc2.dataset.NetcdfDataset]
  variables: {12x1 cell}

 Methods,Events,Superclasses
```

####问:什么是 netcdf 属性？
####答：
是用来做 IO 的 Java 对象。如果您熟悉 Java，则可以通过 [http://www.unidata.ucar.edu/software/netcdf-java/v4.1/javadoc/index.html Netcdf Java API]使用此对象来做各种疯狂的事情。如果您不懂 Java，则可以忽略 netcdf 属性。一件好事是，它可以用于转储[http://www.unidata.ucar.edu/software/netcdf/docs/netcdf/CDL-Syntax.html CDL]并显示有关数据集的所有信息。例如：

```
>>ds=ncdataset('http://dods.mbari.org/opendap/data/ssdsdata/deployments/m0/200706
/OS_MBARI-M0_20070621_R_TS.nc');
>>ds.netcdf
```

```
ans=

 netcdf dods://dods.mbari.org/cgi-bin/nph-
nc/data/ssdsdata/deployments/m0/200706/OS_M0_20070621_TS.nc {
 dimensions:
   LONGITUDE=1;
   LATITUDE=1;
   DEPTH=5;
   bnds=2;
   TIME=UNLIMITED;   // (23510 currently)
 variables:
   double DEPTH_bnds(DEPTH=5,bnds=2);
   float PSAL(TIME=23510,DEPTH=5,LATITUDE=1,LONGITUDE=1);
   :_CoordinateAxes="TIME DEPTH LATITUDE LONGITUDE ";
   :missing_value=-1.0E34f; // float
   :_FillValue=-1.0E34f; // float
   :long_name="Hourly sea_water_salinity";
   :units=" ";
   :standard_name="sea_water_salinity";
   :uncertainty="0.02";
   :valid_min="30";
   :valid_max="36";
   :cell_methods="time: mean";
   :history="From m0_ctd0001_20070621_original";
 <blah blah blah blah ... >
```

####问：什么是变量属性?
####答：是一种字符串的的元胞数组。每个元胞都包含 `ncdataset` 对象中找到的变量名称。您需要这些名称才能获取数据（稍后会详细介绍）。要快速查看数据集包含哪些变量，可以执行以下操作：

```
>>ds=ncdataset('http://dods.mbari.org/opendap/data/ssdsdata/deployments/m0/200706
/OS_MBARI-M0_20070621_R_TS.nc');
>>ds.variables
```

```
ans=

'DEPTH_bnds'
'PSAL'
```

```
'PSAL_QC'
'TEMP'
'TEMP_QC'
'TIME_QC'
'POSITION_QC'
'DEPTH_QC'
'LONGITUDE'
'LATITUDE'
'DEPTH'
'TIME'
```

##读取数据
要从数据集中读取数据，需要为 *data* 方法提供一个变量名，无论是显式（即键入）名称，还是通过访问 `variables` 属性中的值（如果您不知道什么是 *variables 属性*，请返回之前相应部分，然后从那里开始阅读）。这是一个例子：

```
>>ds=ncdataset('http://dods.mbari.org/opendap/data/ssdsdata/deployments/m0/200706
/OS_MBARI-M0_20070621_R_TS.nc')
>>temp=ds.data('TEMP'); % Explicit
>>psal=ds.data(ds.variables{2}) %通过 variables 属性访问'PSAL'
```

####问：如果我不想读取变量中的所有数据怎么办？（即我如何获取数据的子集）
####答：
这是一个很好的问题！有时数据太大无法一次全部提取，或者您可能只需要 10 年时间序列的最后一周。幸运的是，检索数据的子集很容易。首先，您需要知道变量的大小（NetCDF API 中称为 *shape*）。这是获取尺寸的示例：

```
>>ds=ncdataset('http://geoport.whoi.edu/thredds/dodsC/coawst/fmrc/coawst_2_best.n
cd');
>>ds.size('temp')

ans=

   4672        16       336       896
```

调用 `size` *不会*获取数据；它会从已获取的 CDL 中读取信息。因此，无论数据集多大，调用 `size` 都是很快的。
一旦知道形状后，就可以使用该信息指定要检索的*第一个*数据点或切片。如果需要，可以提供一个 *last* 参数（将从 *first* 到 *last* 的数据获取）。还可以指定一个 *stride* 参数，告诉 *ncdataset* 跳过数据，仅沿特定轴获取第 n 个点。这里有几个例子：

```
>>firstIdx=[4672 16 1 1];
>>lastIdx=[4672 16 336 896];
>>t=ds.data('temp',firstIdx, lastIdx); %这里默认间隔是[1 1 1 1]
>>size(t)
```

```
ans=

 1    1   336   896
```

注意到上例中 `t` 的尺寸是 `1 1 336 896`，Matlab 讨厌这些单元素维度，它们很容易剔除：

```
>>t2=squeeze(t);
>>size(t2)

ans=

 336   896
```

####问：我像您的示例一样获取数据，但是 Matlab 提示“警告：CData 必须为 double 或 uint8”。
####答：
Matlab 希望数据具有 *double* 精度，如果您尝试使用其他类型，例如 uint16，unit32，unit64，int8，int16，int32，int64 和 single，大多数 Matlab 函数都会提示。但是，*ncdataset* 基本上按照存储在数据集中的格式来读取。如果数据集将数据存储为 *float*（32 位精度，或 Matlab 称为 *single*），则 ncdataset 会将数据读取为 *float*。使用 Matlab 函数 `double` 可以很容易地将其转换为 *double* 精度数据。这是一个例子：

```
>>ds=ncdataset('http://geoport.whoi.edu/thredds/dodsC/coawst/fmrc/coawst_2_best.n
cd');
>>tSingle=ds.data('temp',[4672 16 1 1], [4672 16 336 896]);
>>tDouble=double(tSingle); %这可以将任何变量都转化为 double
```

##探测数据集的元数据

###属性
大多数数据集都附有元数据。在 NetCDF 术语中，这些称为*属性*（*attributes*）。数据集可能既具有描述整个数据集的*全局属性*（*global attributes*），又具有描述单个变量的*变量属性*（*variable attributes*）。可以使用 *ncdataset.attribute* 方法读取它们。此方法返回一个 2xN（表示 2 列和任意数量的行）大小的元胞数组。第一列包含属性的名称，第二列包含该属性的相应值。这是获取*全局属性*的示例：

```
>>ds=ncdataset('http://www.esrl.noaa.gov/psd/thredds/dodsC/Datasets/ncep.marine/s
st.mean.nc');
>>ds.attributes
```

```
ans=

'platform'      'Ship Observations'
'title'         'NCEP Real-time Marine'
'history'               [1x223 char]
'Conventions'    'COARDS'
```

同样，如果要获取*变量属性*，只需使用变量名称作为参数即可：

```
>>ds.attributes('sst')
```

```
ans=

'_CoordinateAxes'    'time lat lon '
'dataset'                                        [1x23 char  ]
'var_desc'                                       [1x25 char  ]
'level_desc'                                      [1x9  char  ]
'statistic'                                      [1x6  char  ]
'parent_stat'                                     [1x16 char  ]
'actual_range'                                    [2x1  single]
'precision'         [                                    2]
'units'            'degC'
'long_name'         'Sea Surface Temperature Monthly Mean at Surface
```

如果要搜索特定的属性（例如单位），则可以使用 `value4key` 函数来获取值：

```
>>a=ds.attributes('sst');
>>value4key(a,'units')
```

```
ans=

 degC
```

###轴

数据集中的许多变量都有定义数据坐标的轴。轴对于了解诸如何时何地特定数据点测量数据之类的信息非常有用。例如，温度变量通常会具有时间变量，并且可能具有经度和纬度轴。`ncdataset`提供对遵循[http://ferret.wrc.noaa.gov/noaa_coop/coop_cdf_profile.html COARDS]规范的数据文件中轴名称的基本访问。`ncdataset.axes`方法返回的名称是*ncdataset*中代表变量的轴的变量（搞定）。这是搞清楚问题的一个示例：

```
>>ds=ncdataset('http://www.esrl.noaa.gov/psd/thredds/dodsC/Datasets/ncep.marine/s
st.mean.nc');
>>ds.axes('sst')
```

```
ans=

'time'
'lat'
'lon'
```

注意到上述轴的名称都是 ncdataset 中的变量。只要刷新您的记忆，这是我们数据集中的变量。

```
>>ds.variables
```

```
ans=

'sst'
'lat'
'lon'
'time'
```

轴的返回达到了变量的维度。注意“sst”的尺寸：

```
>>ds.axes('sst')
```

```
ans=

'time'
'lat'
'lon'

>>ds.size('sst')

ans=
```

```
    230         90        180
```

在此示例中，长度为 230 的 sst 的第一维具有一个由'time'变量定义的轴。长度为 90 的第二个轴为'lat'。最后，其中有 180 个数据点的第三个轴为“lon”。让我们放在一起：

```
>>sst=ds.data('sst',[230 1 1]); % Grab one day of data
>>sst=squeeze(double(sst)); % Make it a datatype Matlabfunctions like
>>t=ds.time('time',ds.data('time',[230],[230])));
>>lat=ds.data('lat');
>>lon=ds.data('lon');
>>surf(lon,lat,sst); shading('interp');view(2);
>>title(datestr(t));
>>xlabel(value4key(ds.attributes('lon'),'units'))
>>ylabel(value4key(ds.attributes('lat'),'units'))
```

##处理时间变量
许多数据集将至少包含一个时间变量。不幸的是，您使用的数据集的创建者经常使用的时间单位是由于字段特定的原因而~~任意选择[http://code.google.com/p/nctoolbox/issues/detail?id=6&can=1 evil]~~。理想情况下，由于您在 Matlab 中，因此您希望所有时间单位都为[http://www.mathworks.com/access/helpdesk/help/techdoc/ref/datenum.html Matlab 本地时间格式]。`ncdataset` 提供了一种将时间转换为 Matlab 时间的便捷方法。它通过解析变量的 *units* 属性来实现。让我们看一下带有时间变量的数据集：

```
>>ds=ncdataset('http://www.marine.csiro.au/dods/nph-dods/dods-
data/ocean_colour/seawifs/global/8_day/netcdf/f_L3m_8D_T865_9_2007.nc')
>>ds.variables
```

```
ans=

'flag'
't865'
'lon'
'lat'
'time'

>>value4key(ds.attributes('time'),'units')
```

```
ans=

```

```
 seconds since 1980-01-01 00:00:0.0
```

啊哈，让我们获取 *time* 变量，以 1980 年 1 月 1 日 00：00：0.0 为起点，单位为秒作为 Matlab 时间…

```
>>t=ds.time('time');
>>datestr(t(1)) % Using Matlabs build in time handling functions
```

```
ans=

 05-Jan-2007 00:20:11

>>datestr(t(end))

ans=

 29-Dec-2007 12:07:42
```

####问：如果我只想获取并转换时间数据的子集怎么办？
####答：
可以获取一个子集，但需要两步过程，您可以:

- 提取数据子集
- 转换获取的数据

像这样：

```
>>t2=ds.data('time',[1],[4]); % Fetch subset
>>t3=ds.time('time',t2); % convert fetched data. the variable id 'time' is needed
to find the conversion units
>>datestr(t3)
```

```
ans=

 05-Jan-2007 00:20:11
 13-Jan-2007 00:48:46
 20-Jan-2007 23:36:19
 29-Jan-2007 00:03:26
```

####后续更多

2.15 使用 Ncgeodataset

向 NCTOOLBOX 类添加 `ncgeodataset` 对象可提供用于描述地理数据基于 Netcdf-Java 通用数据模型的数据访问和浏览方法。有关本教程之外的其他信息，请参见 ncgeodataset 参考页面。

##打开数据集

数据集可以是本地文件（NetCDF、HDF、Grib）或远程 OPeNDAP 数据 URL。在这里，我们打开一个远程 OPeNDAP 数据 URL。

```
>>ds=ncgeodataset('http://testbedapps.sura.org/thredds/dodsC/estuarine_hypoxia/ch
3d/agg.nc')
```

```
ds=

  ncgeodataset handle

  Properties:
  location:
'http://testbedapps.sura.org/thredds/dodsC/estuarine_hypoxia/ch3d/agg.nc'
  netcdf: [1x1 ucar.nc2.dataset.NetcdfDataset]
  variables: {15x1 cell}

  Methods,Events,Superclasses
```

##读取数据

要从数据集中读取数据，您需要为 `data` 方法提供一个变量名，无论是显式的（即您键入它）还是通过访问 `variables` 属性中的值。这是一个例子：

```
>>ds=ncgeodataset('http://dods.mbari.org/opendap/data/ssdsdata/deployments/m0/200
706/OS_MBARI-M0_20070621_R_TS.nc')
>>temp=ds.data('TEMP'); % Explicit
>>psal=ds.data(ds.variables{2}) % Accessing 'PSAL' using variables property
```

#子集数据：可以对[使用 Ncdataset 数据集对象使用开始，停止和间隔]索引向量进行子集设置，也可以创建变量/地理变量对象，以更类似于 Matlab 的语法访问数据。

```
>>ds=ncgeodataset('http://geoport.whoi.edu/thredds/dodsC/coawst/fmrc/coawst_2_bes
t.ncd');
>>ds.size('temp')
```

```
ans=

   4672        16       336       896
```

调用 `size` *不会*获取数据；它会从已获取的 CDL 中读取信息。因此，无论数据集多大，调用 `size` 都是很快的。

```
>>temp=ds.geovariable('temp');
>>t=temp.data(1,1,:,:); % Can take stride and end index arugments as well.
>>size(t)
```

```
ans=

 1    1   336   896
```

注意到上例中 `t` 的尺寸是 `1 1 336 896`，虽然这是您明确要求的，但是您可能需要剔除它们，以使其他功能按预期运行：

```
>>t2=squeeze(t);
>>size(t2)
```

```
ans=

 336   896
```

##读取全局和变量属性

您可以使用 `metadata` 方法在 netcdf 文件中获取所有属性。这将返回一个有每个变量的属性以及一个名为 `global_attributes` 额外字段的结构体。

```
>>ds=ncgeodataset('http://testbedapps.sura.org/thredds/dodsC/estuarine_hypoxia/ch
3d/agg.nc');
>>ds.metadata
```

```
ans=

global_attributes: {12x2 cell}
          depth: {5x2 cell}
           elev: {5x2 cell}
              u: {7x2 cell}
```

```
              v: {7x2 cell}
       salinity: {7x2 cell}
    temperature: {7x2 cell}
           time: {2x2 cell}
          level: {6x2 cell}
            lon: {5x2 cell}
            lat: {5x2 cell}
```

如果要查找特定变量或数据集的属性，则可以在数据集对象上使用 `attributes` 方法（在这种情况下，您需要使用变量名称作为参数，否则它将返回全局属性）或在变量/地理变量上使用（不需要参数，因为我们已经知道要使用该对象引用什么变量）。

```
>>temp=ds.geovariable('temperature');
>>temp.attributes
```

```
ans=

'_Netcdf4Dimid'    [                3]
'long_name'       'Water temperature'
'units'           'degC'
'coordinates'      'level lat lon'
'_FillValue'      [               NaN]
'wms-layer'       'ch3d/temperature'
'standard_name'    'sea_water_temperature'

>>ds.attributes('temperature')

ans=

'_Netcdf4Dimid'    [                3]
'long_name'       'Water temperature'
'units'           'degC'
'coordinates'      'level lat lon'
'_FillValue'      [               NaN]
'wms-layer'       'ch3d/temperature'
'standard_name'    'sea_water_temperature'
```

如果您知道想要查找的特定属性，则可以对任何数据集或变量/地理变量对象使用 `attribute` 方法。

```
>>temp=ds.geovariable('temperature');
>>temp.attribute('units')
```

```
ans=

 degC

>>ds.attribute('units','temperature')

ans=

 degC
```

###例子：使用不同语法风格读取表面盐度数据
NCTOOLBOX 基本上基于数据集和变量对象，各自有独自的方法。您可以使用这些对象和方法编写脚本和函数，从而可以访问所有工具箱功能，也可以使用更高级的函数。这些函数隐藏了某些面向对象的语法，并且可以使其更紧凑。您可以选择自己喜欢的语法，根据熟悉的数组索引的样式以及所需的功能来做出决定。您还可以在函数脚本中组合不同的方法-无需只选择一种。
以下各节将演示如何使用每种语法样式从 OPeNDAP 数据集中读取表面盐度数据。
####打开数据集

```
>>url='http://geoport.whoi.edu/thredds/dodsC/examples/bora_feb.nc';
>>nc=ncgeodataset(url);
>>nc
```

```
nc=

 ncgeodataset handle

 Properties:
   location:
'http://geoport.whoi.edu/thredds/dodsC/usgs/vault0/models/examples/bora_feb.nc'
     netcdf: [1x1 ucar.nc2.dataset.NetcdfDataset]
  variables: {80x1 cell}

 Methods,Events,Superclasses
```

要访问属性，我们可以使用典型的点符号，类似于普通的 Matlab 结构。在这里，我们要获取正在查看的数据集中的变量列表。

```
>>nc.variables
```

```
ans=

'AKs'
'AKt'
'AKv'
'Akk_bak'
'Akp_bak'
'Akt_bak'
'Akv_bak' ... %输出被截断以便显示在文档中
```

`size` 方法是 ncgeodataset 的方法，该方法返回数据集中给定变量的每个维度的长度。这很像 Matlab 的内部的 size 命令，但是在这种情况下，我们还没有将任何数据加载到内存中。

```
>>nc.size('salt')
```

```
ans=

     8        20        60       160
```

####语法 1
直接使用对象和方法是最灵活的语法，并能提供对更多功能的访问。
在此示例中，我们从该数据集中的 salt 变量创建了一个地理变量对象。这可以通过使用我们感兴趣的 netcdf 变量名称作为参数调用地理变量来完成。

```
>>salt=nc.geovariable('salt')
```

```
salt=

 ncgeovariable handle

 Properties:
   dataset: [1x1 ncgeodataset]
      name: 'salt'
      axes: {4x1 cell}
 attributes: {5x2 cell}

 Methods,Events,Superclasses
```

现在我们可以使用 Matlab 风格数组索引来按索引将 salt 变量子集化。在这里，我们假设维度的排列遵循时间、垂直尺度、水平坐标的顺序。(1,end,:,:)的子集意味着我们正在获取数据集的第一个时间阶、最后一个层级以及整个空间域。
请注意，工具箱返回的值通常与 netcdf 文件中存储的类型相同，因此可能需要将其转换为 Matlab 的 *double* 类型。
同样，可能有必要删除 Matlab 命令的单元素维度，例如使用函数 `squeeze` 进行绘图。

```
>>salinity=salt.data(1,end,:,:);
>>size(salinity)
```

```
ans=

 1    1   60   160

>>class(salinity)
```

```
ans=

 single

>>salinity=squeeze(double(salinity));
```

为了在数据集中绘制第一个时间阶/最后一个层级 salt 值的 Matlab `pcolor` 图，我们需要与 salt 值关联的空间坐标。我们可以像处理 salt 变量一样获取经纬度坐标，如果数据是 CF/COARDS 规范，我们可以利用 netcdf-java 通用数据模型。
地理变量对象的 grid 方法旨在获取给定索引与地理变量关联的所有坐标。用法与 data 方法一样，不同的是结果是一个 Matlab 结构体，其中包含每个地理变量维度的字段，并且其值包含在所请求索引的子集中。

```
>>salinity_coords=salt.grid(1,end,:,:)
```

```
salinity_coords=

   lat_rho: [60x160 double]
   lon_rho: [60x160 double]
ocean_time: 1.0958e+009
     s_rho: -0.9750
```

更高级的选项是使用 `grid_interop` 方法，该方法使用 lat，lon，time 和 z 的标准化名称（而不是坐标尺寸的原始 netcdf 名称）返回地理变量的各维度。
除了使用 `grid_interop` （interop 意为互操作性）返回的更具编程性/标准化的结构外，坐标数据还转换为更标准化的形式：
由 netcdf-java cdm 重新识别的坐标为 time 坐标，将被转换为 Matlab 的 datenum（这可以从最后一个代码块的结果与下面的代码块之间的差异中看出）。
使用 0-360 度方案的经度坐标将转换为-180 至 180 值。
投影的 x 和 y 值将被转换为地理坐标经纬度。
将边界拟合的垂直 sigma 坐标方案转换为每个网格元素的实际垂直深度/高程值。（这可以从最后一个代码块的结果与以下代码块的结果之间的差异中看出。）
像这样：

```
>>salinity_coords=salt.grid_interop(1,end,:,:)
```

```
salinity_coords=

 lat: [60x160 double]
 lon: [60x160 double]
time: 7.3162e+005
   z: [4-D double]

>>size(salinity_coords.z)
```

```
ans=

 1    1   60   160
```

用 `pcolor` 画图很简单：

```
>>pcolor(salinity_coords.lon,salinity_coords.lat,salinity)
>>caxis([35 39])
>>shading flat
>>colorbar
```

现在让我们在图上添加一个标题，其中包括数据集的全局属性 *title* 和子集数据的日期。

```
>>title({nc.attribute('title'); datestr(salinity_coords.time)})
```

####语法 2

该语法使用具有基于单一索引的开始、结束和间隔数组。我们在 ncgeodataset 对象上使用 `data` 方法，其变量名称为变量，每个维的起始索引，每个维的结束索引以及每个维的可选间隔向量。因为我们需要最后一个 salt 级别，并且需要显式指定它，所以我们需要首先确定 salt 数组的尺寸。

```
>>salt_size=size(nc.salt)
```

```
ans=

 8    20    60   160

>>nz=salt_size(2);

>>salinity=nc.data('salt',[1 nz 1 1],[1 nz 60 160],[1 1 1 1]);

>>salinity=squeeze(double(salinity));
```

为了访问 salt 变量的坐标信息，使用与 `data` 相同的参数调用 ncgeodataset 对象上的 `grid` 方法。注意到以这种方式访问坐标数据取决于 netcdf 变量中的坐标属性以定义变量坐标。在这种情况下，即使应该包括时间和 z 坐标，在*坐标*属性中也仅定义了 lat_rho 和 lon_rho，因此 `grid` 方法仅返回它们。

```
>>salinity_coords=nc.grid('salt',[1 nz 1 1],[1 nz 60 160],[1 1 1 1])

salinity_coords=

 lat_rho: [60x160 double]
 lon_rho: [60x160 double]
```

获取所需子集的日期，以便将其添加到图形标题中，因为它不包含在 `grid` 命令结果中。我们可以在 ncgeodataset 对象上使用 `time` 方法来完成从模型的 *ocean_time* 到 Matlab 的 *datenum* 的转换。第一个参数应该是时间变量的名称，第二个参数是您尝试转换的时间值或值数组。第二个参数是可选的，如果省略，则将时间变量的整个长度转换为 *datenum*。

```
>>time=nc.time('ocean_time',nc.data('ocean_time',1,1))
```

```
time=

 7.3162e+005

>>time=datestr(time)
```

```
time=

 11-Feb-2003 12:00:00
```

使用 `pcolor` 画图。

```
>>pcolor(salinity_coords.lon_rho,salinity_coords.lat_rho,salinity)
>>caxis([35 39])
>>shading flat
>>colorbar
```

在图上添加一个标题，其中包括数据集的全局属性 *title* 和子集数据的日期。

```
>>title({nc.attribute('title'); time})
```

####语法 3
对于曾经看过 njTBX 或较旧的 Chuck Denham 的 Matlabnetcdf 工具箱的用户，应该熟悉此语法。我们在花括号内指定变量名称，然后使用 Matlab 风格索引对其进行子集化。

```
>>salinity=nc{'salt'}(1,end,:,:);
>>size(salinity)
```

```
ans=

 1    1   60   160

>>salinity=squeeze(double(salinity));
```

此语法中的 `grid` 方法与 *语法 1* 中的方法中的 `grid_interop` 方法是完全相同的命令。要访问数据的坐标（我们只是子集），我们可以发出相同的命令，但在末尾添加一个`.grid`。

```
>>salinity_coords=nc{'salt'}(1,end,:,:).grid
```

```
salinity_coords=

 time: 7.3162e+005
 lon: [60x160 double]
 lat: [60x160 double]
```

```
   z: [4-D double]
```

使用 `pcolor` 画图。

```
>>pcolor(salinity_coords.lon,salinity_coords.lat,salinity)
>>caxis([35 39])
>>shading flat
>>colorbar
```

现在让我们在图上添加一个标题，其中包括数据集的全局属性 *title* 和子集数据的日期。

```
>>title({nc.attribute('title'); datestr(salinity_coords.time)})
```

####语法 4

因为在特定时间提取体积或垂直层是一项常用功能，而这可能是我们要做的全部工作，所以有一个名为 `nj_tslice` 的高级便捷函数也可以完成我们的工作。函数 `nj_tslice` 旨在复制同名 njTBX 函数中的功能。如果工作路径上两个工具箱函数同时存在，它们将发生冲突。

```
>>[s,g]=nj_tslice(url,'salt',1,-1); % -1 for last level
>>pcolor(g.lon,g.lat,double(s))
>>caxis([35 39])
>>shading flat
>>colorbar
```

####结果：无论我们使用哪种语法，我们都应该得到下面的图片：
（译者注：不好意思，图挂了……）

[http://nctoolbox.googlecode.com/hg/cdm/utilities/misc/bora.png]

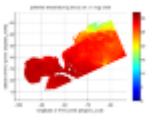
有关处理时间，坐标变量，坐标转换等的更多信息，请参见[ncgeodataset]和[ncgeovariable]的参考文档。

2.16 使用 Ncgeovariable

简介
抱歉，我们正在尽快添加相关文档。

3. 演示文件

3.1 DEMO9



--演示使用“结构”语法对 CF 规范数据集子集进行访问变量及其坐标轴数据。像 demo8 一样，使用'struct'语法访问变量及其坐标轴数据。

开始 DEMO9 ----------
演示 CF 规范数据集的子集化

```
echo('on')
url='http://geoport.whoi.edu/thredds/dodsC/coawst_2_2/fmrc/coawst_2_2_best.ncd';
ds=cfdataset(url);、
```

获取感兴趣变量的尺寸，现在还没有开始读取数据

```
sz=ds.size('temp');
```

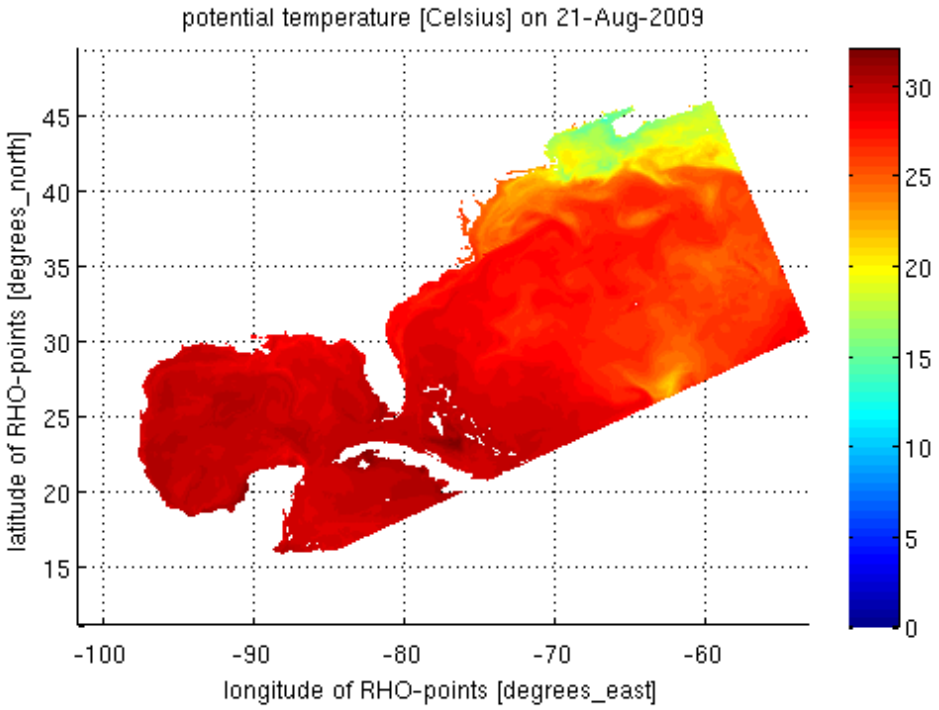
获取数据的子集，现在数据通过互联网在推送。

```
t=ds.struct('temp',[sz(1) sz(2) 1 1],[sz(1) sz(2) sz(3) sz(4)]);
```

画一个漂亮的图，注意要调用'squeeze'函数去除单元素维度

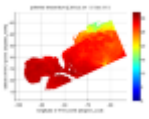
```
figure;
surf(t.lon_rho,t.lat_rho,double(squeeze(t.temp)))
shading('interp');
view(2)
axis('equal')

xatt=ds.attributes('lon_rho');
xname=value4key(xatt,'long_name');
xunits=value4key(xatt,'units');
xlabel([xname ' [' xunits ']'],'interpreter','none');

yatt=ds.attributes('lat_rho');
yname=value4key(yatt,'long_name');
```

```
yunits=value4key(yatt,'units');
ylabel([yname ' [' yunits ']'],'interpreter','none');

zatt=ds.attributes('temp');
zname=value4key(zatt,'long_name');
zunits=value4key(zatt,'units');
ztime=ds.time('time1',t.time1);
title([zname ' [' zunits '] on ' datestr(ztime(1))],'interpreter','none');

colorbar
shg
echo('off') % 结束 DEMO9 ------------------------------------------------
```



MATLAB® 7.13 出品

3.2 DEMO9A



--类似于 demo9，但是分别通过 2 步使用'grid'和'data' 语法去访问变量和坐标轴数据，代替了 demo9 中使用'struct'单一调用的方法。如果需要在提取大量数据之前基于变量的坐标进行子集化，则可能需要执行此操作。

演示 CF 规范数据集的子集化

```
echo('on')
url='http://geoport.whoi.edu/thredds/dodsC/coawst_2_2/fmrc/coawst_2_2_best.ncd';
ds=cfdataset(url);
```

获取感兴趣变量的尺寸，现在还没有开始读取数据

```
sz=ds.size('temp');

% !!在 demo9 内我们使用'ds.struct'来获取啦数据和其坐标轴。
% 在这里我们分别使用'ds.grid'和'ds.data'来分别从数据中获取坐标轴和数据。
```

仅获取坐标数据子集（即坐标“网格”）获取数据的子集
现在数据通过互联网在推送。

```
t=ds.grid('temp',[sz(1) sz(2) 1 1],[sz(1) sz(2) sz(3) sz(4)]);
```

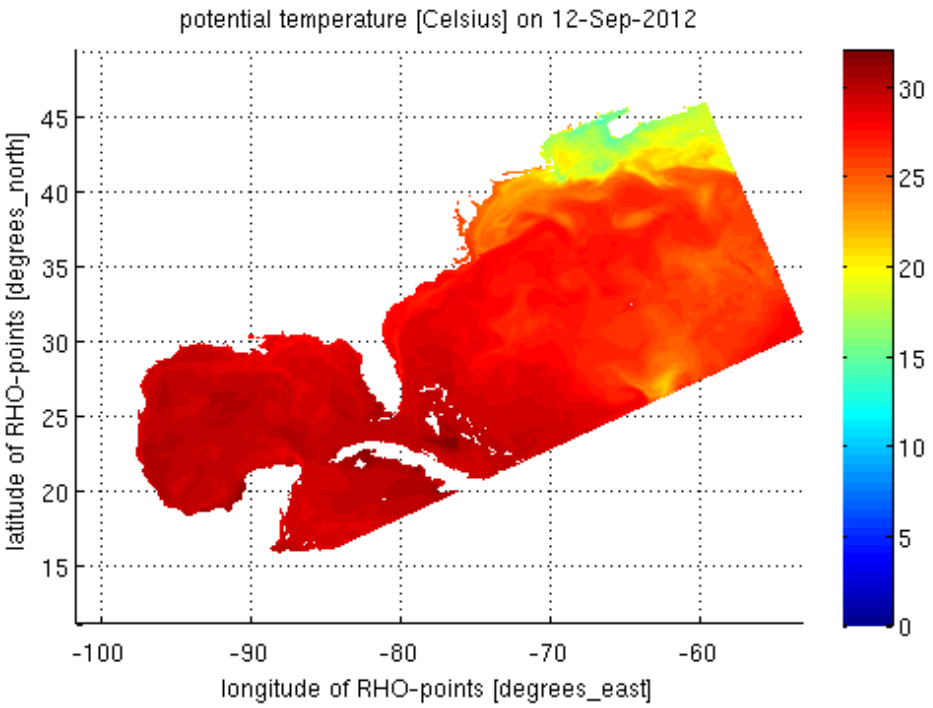
获取真实的温度数据

```
d=ds.data('temp',[sz(1) sz(2) 1 1],[sz(1) sz(2) sz(3) sz(4)]);
```

画一个漂亮的图，注意要调用“squeeze”函数去除单元素维度

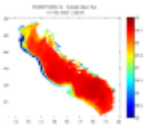
```
figure;
surf(t.lon_rho,t.lat_rho,double(squeeze(d)))
shading('interp');
view(2)
axis('equal')

xatt=ds.attributes('lon_rho');
```

```
xname=value4key(xatt,'long_name');
xunits=value4key(xatt,'units');
xlabel([xname ' [' xunits ']'],'interpreter','none');

yatt=ds.attributes('lat_rho');
yname=value4key(yatt,'long_name');
yunits=value4key(yatt,'units');
ylabel([yname ' [' yunits ']'],'interpreter','none');

zatt=ds.attributes('temp');
zname=value4key(zatt,'long_name');
zunits=value4key(zatt,'units');
ztime=ds.time('time1',t.time1);
title([zname ' [' zunits '] on ' datestr(ztime)],'interpreter','none');

colorbar
shg
echo('off')
```



MATLAB® 7.13 出品

3.3 GEODEMO 1A



--方法 A：使用地理变量语法读取表面盐度。由于在从地理变量对象提取数据之前先行进行了创建，因此采取了一些额外的步骤，但是随后可以使用所有地理变量方法。

打开符合 CF 的曲线 ROMS 模型数据集的 OPeNDAP 数据 URL

```
url='http://geoport.whoi.edu/thredds/dodsC/examples/bora_feb.nc';
nc=ncgeodataset(url)
```

```
nc=

  ncgeodataset handle

  Properties:
     location: [1x58 char]
       netcdf: [1x1 ucar.nc2.dataset.NetcdfDataset]
    variables: {80x1 cell}
```

看一下数据集中可用的变量
要访问属性，我们可以类似普通 Matlab 结构体使用典型的点符号。在这里，我们要获取正在查看的数据集中的变量列表。

```
 nc.variables
```

```
ans=

    'AKs'
    'AKt'
    'AKv'
    'Akk_bak'
    'Akp_bak'
    'Akt_bak'
    'Akv_bak'
    'Cs_r'
    'Cs_w'
```

```
    'Falpha'
    'Fbeta'
    'Fgamma'
    'M2nudg'
    'M3nudg'
    'Tcline'
    'Tnudg'
    'Znudg'
    'Zob'
    'Zos'
    'angle'
    'dstart'
    'dt'
    'dtfast'
    'el'
    'f'
    'gamma2'
    'gls_Kmin'
    'gls_Pmin'
    'gls_c1'
    'gls_c2'
    'gls_c3m'
    'gls_c3p'
    'gls_cmu0'
    'gls_m'
    'gls_n'
    'gls_p'
    'gls_sigk'
    'gls_sigp'
    'h'
    'hc'
    'mask_psi'
    'mask_rho'
    'mask_u'
    'mask_v'
    'nAVG'
    'nHIS'
    'nRST'
    'nSTA'
    'ndefHIS'
    'ndtfast'
```

```
    'ntimes'
    'ntsAVG'
    'pm'
    'pn'
    'rdrg'
    'rdrg2'
    'rho0'
    'salt'
    'spherical'
    'temp'
    'theta_b'
    'theta_s'
    'tnu2'
    'u'
    'ubar'
    'v'
    'vbar'
    'xl'
    'zeta'
    'lat_psi'
    'lat_rho'
    'lat_u'
    'lat_v'
    'lon_psi'
    'lon_rho'
    'lon_u'
    'lon_v'
    'ocean_time'
    's_rho'
    's_w'
```

确定所选变量的形状

size 方法是 ncgeodataset 的方法，该方法返回数据集中给定变量每个维度的长度。这很像 Matlab 的内部 size 命令，但是在这种情况下，我们还没有将任何数据加载到内存中。所有这些信息都来自 netcdf-java cdm。

```
 nc.size('salt')
```

```
ans=

```

```
          8          20          60         160
```

创建地理变量对象以访问 MATLAB 个个索引中的数据
在此示例中，我们从该数据集中的 salt 变量创建了一个地理变量对象。这可以通过使用我们感兴趣的 netcdf 变量名称作为参数来调用 geovariable 来完成。

```
 salt=nc.geovariable('salt')

% 现在我们可以使用 Matlab 风格的数组索引来按其索引将 salt 变量子集化。
% 我们可以使用"dimensions"方法查看维度名称：

  nc.dimensions('salt')

% 我们可以看到维度的排列遵循时间、垂直尺度、水平坐标的顺序。
% (1,end,:,:)的子集意味着我们正在获取数据集的第一个时间阶、最后一个层级以及整个空间域。

% 请注意，工具箱返回的值通常与 netcdf 文件中存储的类型相同，
% 因此可能需要将其转换为 Matlab 的 double 类型。

 salinity=salt.data(1,end,:,:);
 size(salinity)
 class(salinity)

% 同样，可能有必要删除 Matlab 命令的单元素维度，例如使用函数 squeeze 进行绘图。
 salinity=squeeze(double(salinity));
```

```
salt=

  ncgeovariable handle

  Properties:
      dataset: [1x1 ncgeodataset]
         name: 'salt'
         axes: {4x1 cell}
    attributes: {6x2 cell}



ans=
```

```
    'ocean_time'
    's_rho'
    'eta_rho'
    'xi_rho'
```

```
ans=

     1     1    60   160


ans=

single
```

使用地理变量对象通过 Matlab 风格索引去访问坐标数据
为了在数据集中绘制第一个时间阶/最后一个层级 salt 值的 Matlabpcolor 图，我们需要与 salt 值关联的空间坐标。我们可以像处理 salt 变量一样获取经纬度坐标，如果数据是 CF/COARDS 规范，我们可以利用 netcdf-java 通用数据模型。

```
% 地理变量对象的 grid 方法旨在获取给定索引与地理变量关联的所有坐标。
% 用法与 data 方法一样，不同的是结果是一个 Matlab 结构体，其中包含每个地理变量维度的字段，
% 并且其值包含在所请求索引的子集中。

 salinity_coords=salt.grid(1,end,:,:)
```

```
salinity_coords=

      lat_rho: [60x160 double]
      lon_rho: [60x160 double]
   ocean_time: 1.0958e+09
        s_rho: -0.0250
```

使用 grid_interop 方法返回带有标准、可互操作名称的坐标轴
更高级的选项是使用 grid_interop 方法，该方法使用 lat，lon，time 和 z 的标准化名称（而不是坐标尺寸的原始 netcdf 名称）返回地理变量的各维度。

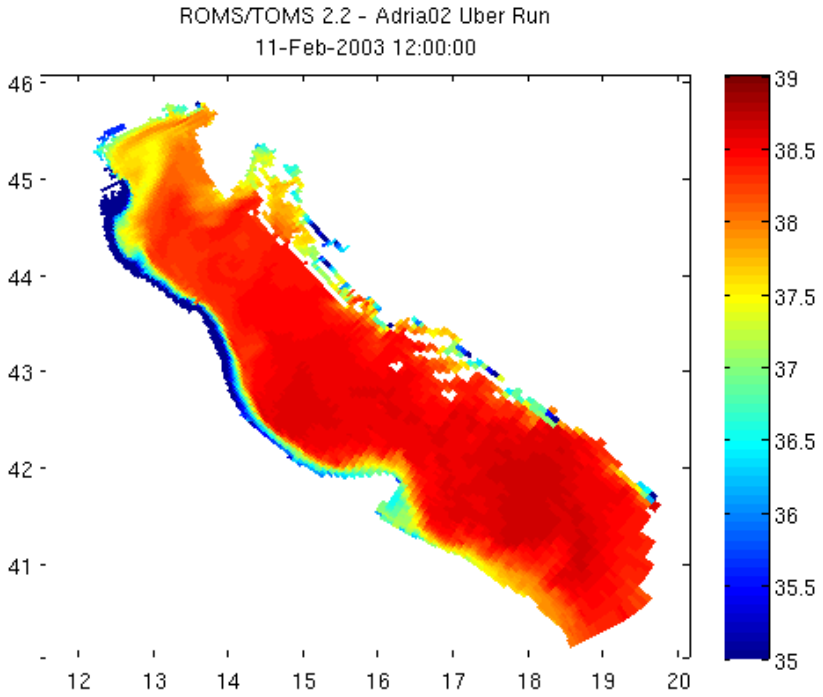
```
% 除了使用 grid_interop（interop 意为互操作性）返回的更具编程性/标准化的结构外，
% 坐标数据还转换为更标准化的形式：
%
```

```
% - 时间坐标被转化为 Matlabdatenum 形式。
% - 使用 0-360 的经度坐标被转化到[-180,180]的范围内。
% - 投影的 x 和 y 值将被转换为地理坐标经纬度。
% - 垂直坐标被转化为 z 值。

 salinity_coords=salt.grid_interop(1,end,:,:)
```

```
salinity_coords=

    lat: [60x160 double]
    lon: [60x160 double]
   time: 7.3162e+05
      z: [4-D double]
```

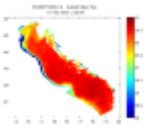
使用 MATLABpcolor 画图

```
pcolor(salinity_coords.lon,salinity_coords.lat,salinity)
shading flat; colorbar; caxis([35 39]);

% 在图上添加一个标题，其中包括数据集的全局属性 title 和子集数据的日期。
 title({nc.attribute('title'); datestr(salinity_coords.time)})
```



MATLAB® 7.13 出品

3.4 GEODEMO 1B



--方法 B：使用角/边/间隔语法读取表面盐度。如果比起 Matlab，您更习惯于在 Fortran 中使用 NetCDF，那么这可能是核实的语法。

打开符合 CF 的曲线 ROMS 模型数据集的 OPeNDAP 数据 URL

```
url='http://geoport.whoi.edu/thredds/dodsC/examples/bora_feb.nc';
nc=ncgeodataset(url)
```

```
nc=

  ncgeodataset handle

  Properties:
     location: [1x58 char]
       netcdf: [1x1 ucar.nc2.dataset.NetcdfDataset]
    variables: {80x1 cell}
```

看一下数据集中可用的变量
要访问属性，我们可以类似普通 Matlab 结构体使用典型的点符号。在这里，我们要获取正在查看的数据集中的变量列表。

```
nc.variables
```

```
ans=

    'AKs'
    'AKt'
    'AKv'
    'Akk_bak'
    'Akp_bak'
    'Akt_bak'
    'Akv_bak'
    'Cs_r'
    'Cs_w'
    'Falpha'
```

```
    'Fbeta'
    'Fgamma'
    'M2nudg'
    'M3nudg'
    'Tcline'
    'Tnudg'
    'Znudg'
    'Zob'
    'Zos'
    'angle'
    'dstart'
    'dt'
    'dtfast'
    'el'
    'f'
    'gamma2'
    'gls_Kmin'
    'gls_Pmin'
    'gls_c1'
    'gls_c2'
    'gls_c3m'
    'gls_c3p'
    'gls_cmu0'
    'gls_m'
    'gls_n'
    'gls_p'
    'gls_sigk'
    'gls_sigp'
    'h'
    'hc'
    'mask_psi'
    'mask_rho'
    'mask_u'
    'mask_v'
    'nAVG'
    'nHIS'
    'nRST'
    'nSTA'
    'ndefHIS'
    'ndtfast'
    'ntimes'
```

```
    'ntsAVG'
    'pm'
    'pn'
    'rdrg'
    'rdrg2'
    'rho0'
    'salt'
    'spherical'
    'temp'
    'theta_b'
    'theta_s'
    'tnu2'
    'u'
    'ubar'
    'v'
    'vbar'
    'xl'
    'zeta'
    'lat_psi'
    'lat_rho'
    'lat_u'
    'lat_v'
    'lon_psi'
    'lon_rho'
    'lon_u'
    'lon_v'
    'ocean_time'
    's_rho'
    's_w'
```

确定所选变量的形状

size 方法是 ncgeodataset 的方法，该方法返回数据集中给定变量每个维度的长度。这很像 Matlab 的内部 size 命令，但是在这种情况下，我们还没有将任何数据加载到内存中。所有这些信息都来自 netcdf-java cdm。

```
sz=nc.size('salt')
```

```
sz=

         8        20        60       160
```

在 NCTOOLBOX 中使用开始、结束、间隔数组风格索引
这种子集数组的语法更像是 C 或者 Java（不过 Matlab 从 1 开始索引）

```
% To access data this way,use the data method on the ncgeodataset object
% representing your local or remote netcdf dataset. This is a lower level
% data access method. Call data with arguments of variable name,start
% indices for each dimension,end indices for each dimension,and an
% optional stride vector for each dimension.

% let take a look at the dimension names first,which will inform how
% we need to specify the indices

 nc.dimensions('salt')

% We see that we have time,z,y,x dimensions,in that order

 nz=sz(2);
 salinity=nc.data('salt',[1 nz 1 1],[1 nz 60 160],[1 1 1 1]);
 size(salinity)

 salinity=squeeze(double(salinity));
```

```
ans=

    'ocean_time'
    's_rho'
    'eta_rho'
    'xi_rho'


ans=

     1     1    60   160
```

开始、结束、间隔数组风格索引返回坐标轴
为了访问 salt 变量的坐标信息，使用与数据相同的参数调用 ncgeodataset 对象上的 grid 方法。

```
% 注意到以这种方式访问坐标数据取决于 netcdf 变量中的坐标属性以定义变量坐标。
% 在这种情况下，即使应该包括时间和 z 坐标，在坐标属性中也仅定义了 lat_rho 和 lon_rho，
```

```
% 因此 grid 方法仅返回它们。

 salinity_coords=nc.grid('salt',[1 nz 1 1],[1 nz 60 160],[1 1 1 1])
```

```
salinity_coords=

    lat_rho: [60x160 double]
    lon_rho: [60x160 double]
```

使用 time 方法将模式时间转化为 Matlab 的 datenum
获取所需子集的日期，以便将其添加到图形标题中，因为它不包含在 grid 命令结果中。我们可以在 ncgeodataset 对象上使用 time 方法来完成从模型的 ocean_time 到 Matlab 的 datenum 的转换。第一个参数应该是时间变量的名称，第二个参数是您尝试转换的时间值或值数组。第二个参数是可选的，如果省略，则将时间变量的整个长度转换为 datenum。

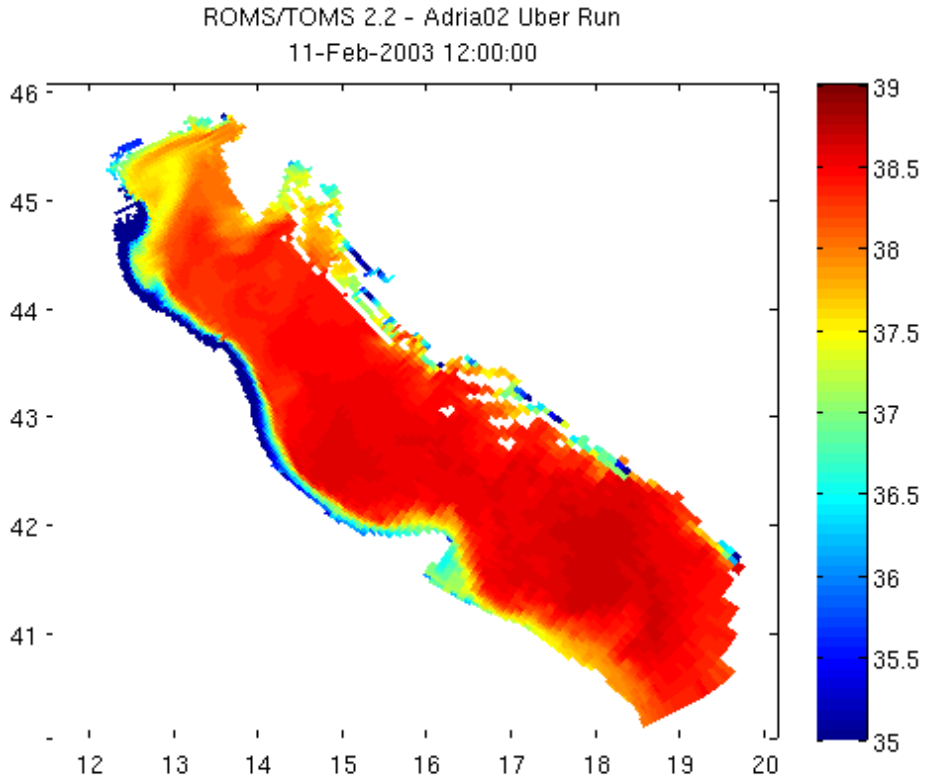
```
 time=nc.time('ocean_time',nc.data('ocean_time',1,1))

 time=datestr(time)
```

```
time=

   7.3162e+05


time=

11-Feb-2003 12:00:00
```

使用 Matlab 的 pcolor 命令画图
使用 pcolor 进行绘图与下面的代码一样简单。有时，坐标作为矢量存储在 netcdf 数据集中（与这些经纬坐标所在的二维数组相对）。在这种情况下，请参见 Matlab 的 meshgrid 函数从矢量创建二维格子网格。

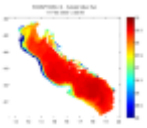
```
pcolor(salinity_coords.lon_rho,salinity_coords.lat_rho,salinity)
 shading flat; colorbar; caxis([35 39]);

% Now let's add a title to the figure that includes the dataset's global
% attribute title and the date of the data that we subset.
```

```
title({nc.attribute('title'); time})
```



MATLAB® 7.13 出品

3.5 GEODEMO 1C



--方法 C：使用大括号语法读取表面盐度：s=nc{'salt'}(1,end,:,:);最紧凑的通用方法，但功能仅限于读取数据，获得可互操作的网格结构并获取属性。
打开符合 CF 的曲线 ROMS 模型数据集的 OPeNDAP 数据 URL

```
url='http://geoport.whoi.edu/thredds/dodsC/examples/bora_feb.nc';
nc=ncgeodataset(url)
```

```
nc=

  ncgeodataset handle

  Properties:
     location: [1x58 char]
       netcdf: [1x1 ucar.nc2.dataset.NetcdfDataset]
    variables: {80x1 cell}
```

看一下数据集中可用的变量
要访问属性，我们可以类似普通 Matlab 结构体使用典型的点符号。在这里，我们要获取正在查看的数据集中的变量列表。

```
 nc.variables
```

```
ans=

    'AKs'
    'AKt'
    'AKv'
    'Akk_bak'
    'Akp_bak'
    'Akt_bak'
    'Akv_bak'
    'Cs_r'
    'Cs_w'
    'Falpha'
    'Fbeta'
```

```
    'Fgamma'
    'M2nudg'
    'M3nudg'
    'Tcline'
    'Tnudg'
    'Znudg'
    'Zob'
    'Zos'
    'angle'
    'dstart'
    'dt'
    'dtfast'
    'el'
    'f'
    'gamma2'
    'gls_Kmin'
    'gls_Pmin'
    'gls_c1'
    'gls_c2'
    'gls_c3m'
    'gls_c3p'
    'gls_cmu0'
    'gls_m'
    'gls_n'
    'gls_p'
    'gls_sigk'
    'gls_sigp'
    'h'
    'hc'
    'mask_psi'
    'mask_rho'
    'mask_u'
    'mask_v'
    'nAVG'
    'nHIS'
    'nRST'
    'nSTA'
    'ndefHIS'
    'ndtfast'
    'ntimes'
    'ntsAVG'
```

```
    'pm'
    'pn'
    'rdrg'
    'rdrg2'
    'rho0'
    'salt'
    'spherical'
    'temp'
    'theta_b'
    'theta_s'
    'tnu2'
    'u'
    'ubar'
    'v'
    'vbar'
    'xl'
    'zeta'
    'lat_psi'
    'lat_rho'
    'lat_u'
    'lat_v'
    'lon_psi'
    'lon_rho'
    'lon_u'
    'lon_v'
    'ocean_time'
    's_rho'
    's_w'
```

确定所选变量的形状

size 方法是 ncgeodataset 的方法，该方法返回数据集中给定变量每个维度的长度。这很像 Matlab 的内部 size 命令，但是在这种情况下，我们还没有将任何数据加载到内存中。所有这些信息都来自 netcdf-java cdm。

```
 nc.size('salt')
```

```
ans=

         8        20        60       160
```

使用传统语法在 NCTOOLBOX 中进行数据访问
对于曾经看过 njTBX 或较旧的 Chuck Denham 的 Matlabnetcdf 工具箱的用户，应该熟悉此语法。将大括号放在包含变量名的字符串两边，然后使用常规的 matlab 矩阵索引。如果不包括索引，则返回地理变量对象。(:)可用于整个变量。

```
% 首先看看所有维度的名称，这将告诉我们如何指定索引

 nc.dimensions('salt')

% 可以看到我们有 time,z,y,x 等维度及其顺序
 salinity=nc{'salt'}(1,end,:,:);
 size(salinity)

 salinity=squeeze(double(salinity)); % 这可能通过{}已经搞定了
```

```
ans=

    'ocean_time'
    's_rho'
    'eta_rho'
    'xi_rho'


ans=

     1     1    60   160
```

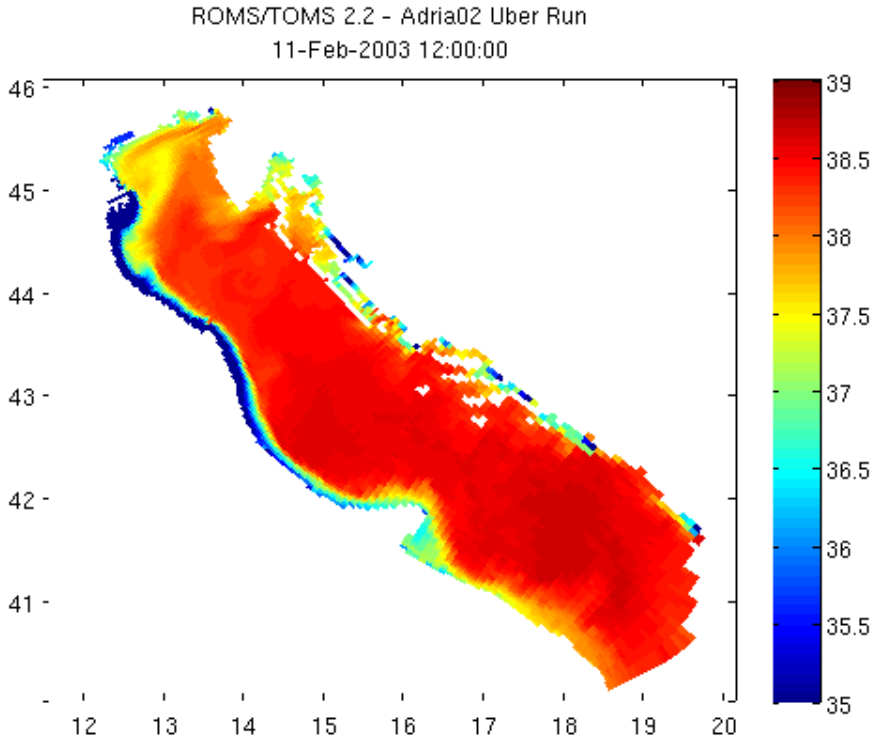
使用传统语法在 NCTOOLBOX 中进行坐标轴访问
此语法中的 grid 方法与语法 1 中的方法中的 grid_interop 方法是完全相同的命令。要访问数据的坐标（我们只是子集），我们可以发出相同的命令，但在末尾添加一个.grid。

```
% 坐标轴数据已经转化为一个更加标准化的形式
%
%     由 netcdf-java cdm 重新识别的坐标为 time 坐标，将被转换为 Matlab 的 datenum、。
%     使用 0-360 度方案的经度坐标将转换为-180 至 180 值。
%     投影的 x 和 y 值将被转换为地理坐标经纬度。
%     将边界拟合的垂直 sigma 坐标方案转换为每个网格元素的实际垂直深度/高程值。

 salinity_coords=nc{'salt'}(1,end,:,:).grid
```

```
salinity_coords=

    time: 7.3162e+05
     lon: [60x160 double]
     lat: [60x160 double]
       z: [60x160 double]
```

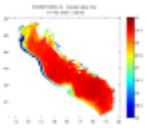
使用 MATLABpcolor 画图
使用 pcolor 进行绘图与下面的代码一样简单。有时，坐标作为矢量存储在 netcdf 数据集中（与这些经纬坐标所在的二维数组相对）。在这种情况下，请参见 Matlab 的 meshgrid 函数从矢量创建二维格子网格。

```
figure
pcolor(salinity_coords.lon,salinity_coords.lat,salinity)
shading flat; colorbar; caxis([35 39]);

% 在图上添加一个标题，其中包括数据集的全局属性 title 和子集数据的日期。

title({nc.attribute('title'); datestr(salinity_coords.time)})
```



MATLAB® 7.13 出品

3.6 GEODEMO 1D



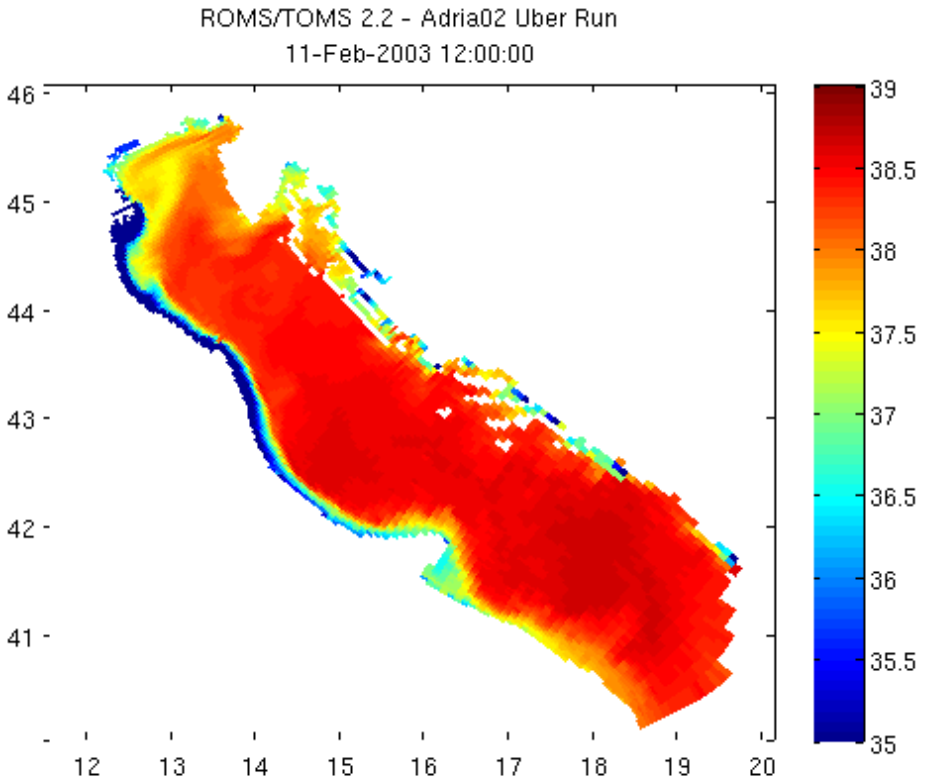
--方法 D:使用 NJ_TSLICE 读取表面盐度。读取数据和网格最方便的方法，但是受限于特定时间阶中的整个切片或三维数据量

打开符合 CF 的曲线 ROMS 模型数据集的 OPeNDAP 数据 URL

```
url='http://geoport.whoi.edu/thredds/dodsC/examples/bora_feb.nc';
itime=1;
ilevel=-1; % nj_tslice uses "-1" instead of "end" to indicate last level
[salinity,grd]=nj_tslice(url,'salt',itime,ilevel);
```

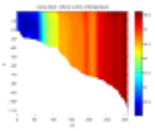
用 pcolor 画图
使用 pcolor 进行绘图与下面的代码一样简单。有时，坐标作为矢量存储在 netcdf 数据集中（与这些经纬坐标所在的二维数组相对）。在这种情况下，请参见 Matlab 的 meshgrid 函数从矢量创建二维格子网格。

```
% Remember that pcolor expects a double as input
pcolor(grd.lon,grd.lat,double(salinity))
shading flat; colorbar; caxis([35 39]);

% Now let's add a title to the figure that includes the dataset's global
% attribute title and the date of the data that we subset.
nc=ncdataset(url);
title({nc.attribute('title'); datestr(grd.time)})
```



MATLAB® 7.13 出品

3.7 GEODEMO 2

 -- 使用 VSLICEG 从三维场创建垂直剖面。

OpenDAP 数据集：

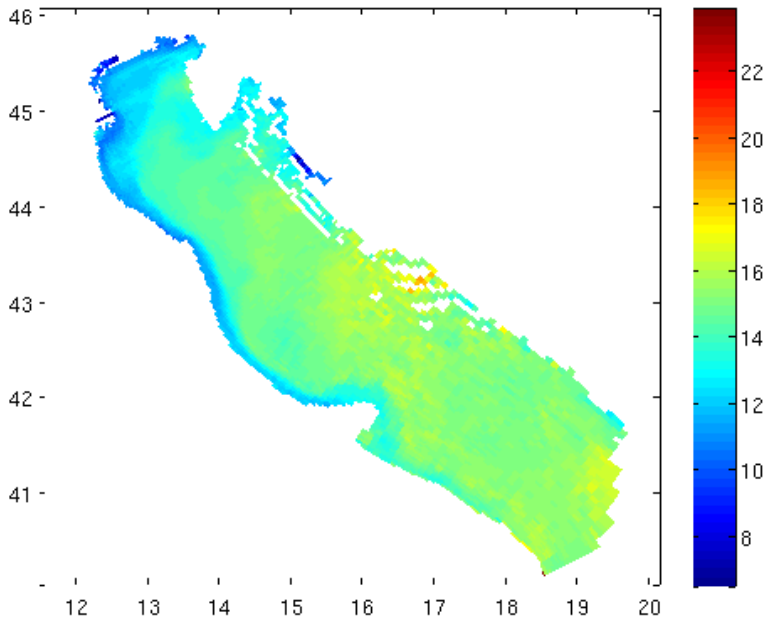
```
url='http://geoport.whoi.edu/thredds/dodsC/usgs/vault0/models/examples/bora_feb.nc';
```

使用 NCTOOLBOX 中的 nj_tbx 函数
在这里，我们将使用方便的 nj_tslice 函数，该函数在特定时间步长返回完整的三维场和网格(t,z,y,x)

```
itime=1;
[data,grd]=nj_tslice(url,'temp',itime);
```

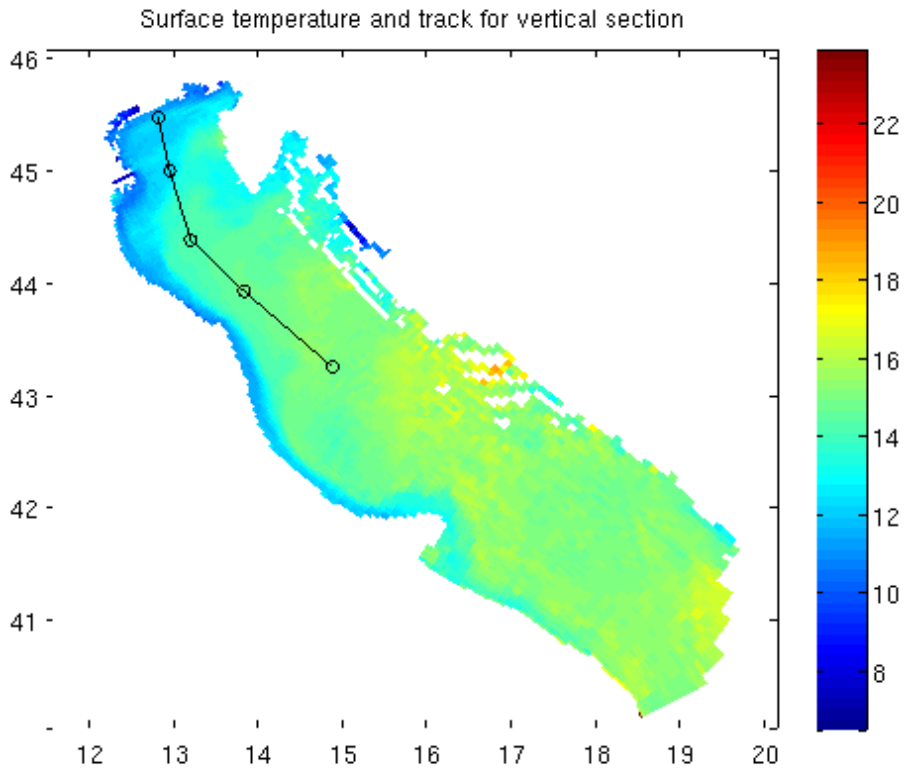
从这个数据中画出表面温度
现在我们使用 ROMS 输出值，这里最后一层(end)是表面

```
figure;
pcolorjw(grd.lon,grd.lat,double(data(end,:,:)));
colorbar
```



在格点数据集中创建垂直截面
指定一条需要计算垂直剖面的经纬度轨迹

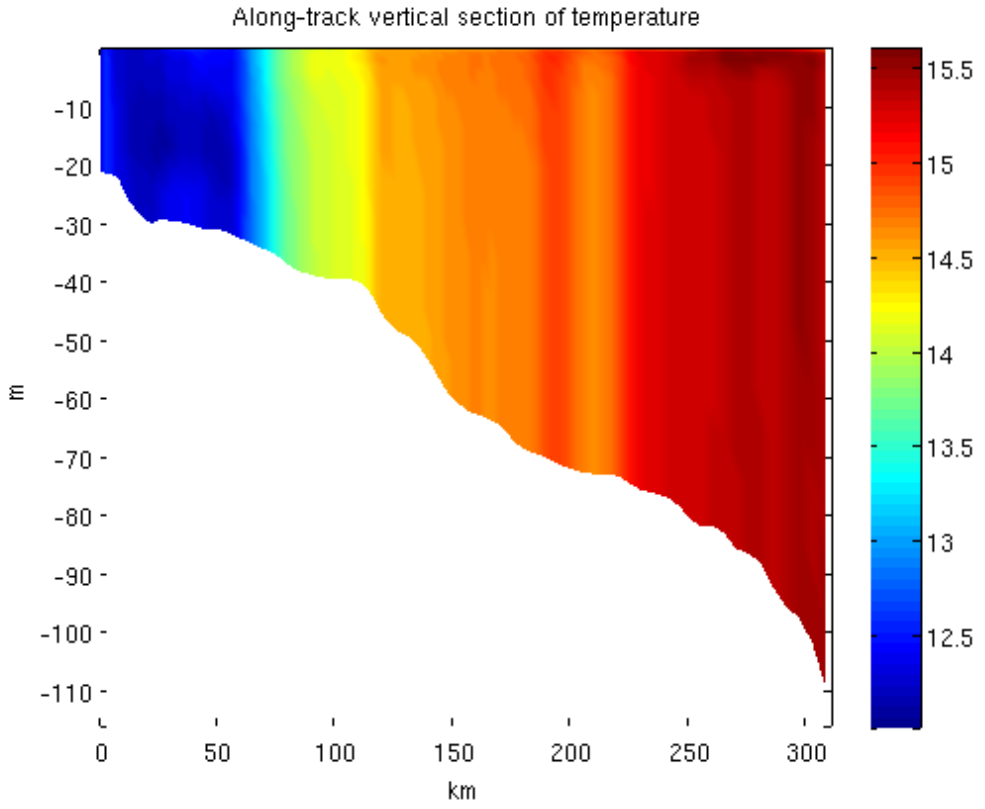
```
track=[   12.7950   45.4588
   12.9280   44.9895
   13.1773   44.3736
   13.8090   43.9190
   14.8562   43.2444];
lon=track(:,1);
lat=track(:,2);
hold on;
plot(lon,lat,'k-o')
hold off;
title('Surface temperature and track for vertical section');
```



使用 VSLICEG 计算垂直剖面

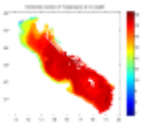
```
[x,y,vdata]=vsliceg(data,grd,lon,lat);
figure;
```

```
pcolorjw(x,y,vdata);ylabel('m');xlabel('km');
colorbar
title('Along-track vertical section of temperature');
shading interp
```



MATLAB® 7.13 出品

3.8 GEODEMO 3

 -- 使用 ZSLICEG 从三维场中创建水平截面

OpenDAP 数据集:

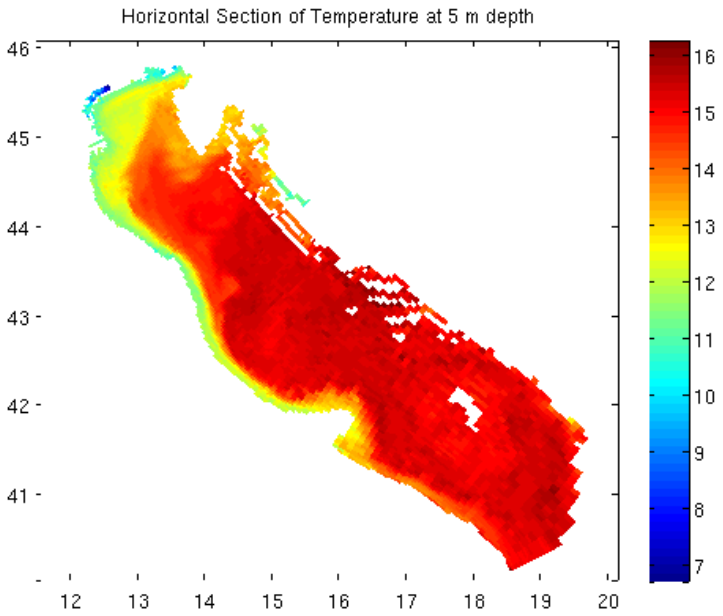
```
uri='http://geoport.whoi.edu/thredds/dodsC/examples/bora_feb.nc'
```

使用 NCTOOLBOX 中的 nj_tbx 函数
首先获取三维场，然后内插水平场到 5 米深度

```
[t,g]=nj_tslice(uri,'temp',1);% 首先获取三维场中的'temp'
tz=zsliceg(t,g.z,-5);  % 返回 5 米深度的温度切面
```

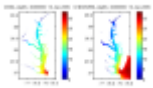
绘制内插的水平场
使用其坐标绘制内插的水平场，添加图标题和颜色条

```
figure
pcolorjw(g.lon,g.lat,double(tz));  % 绘制 5 米深度的温度
title('Horizontal Section of Temperature at 5 m depth');
colorbar
```



MATLAB® 7.13 出品

3.9 GEODEMO 4



-- 在特定的时间步长和深度下，比较来自两个不同 CF 兼容结构化网格模型（CH3D 和 ROMS）的水平切片。

CH3D

```
url{1}='http://testbedapps.sura.org/thredds/dodsC/estuarine_hypoxia/ch3d/agg.nc';
var{1}='salinity';
titl{1}='CH3D';
```

ROMS

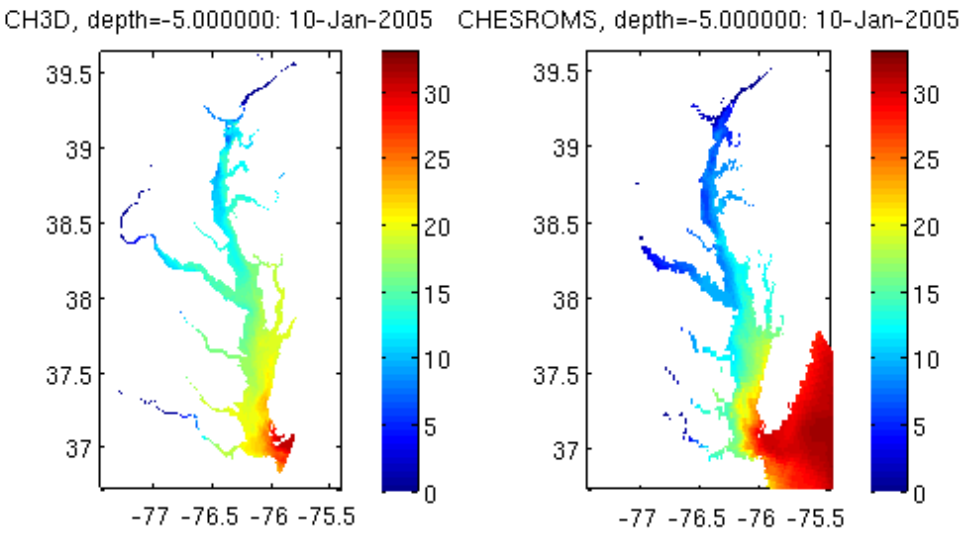
```
url{2}='http://testbedapps.sura.org/thredds/dodsC/estuarine_hypoxia/chesroms_1tdo
/agg.nc';
var{2}='salt';
titl{2}='CHESROMS';

dat=[2005 1 10 0 0 0];  % Jan 10,2005 00:00 UTC
depth=-5;  % 5 米水平切面
ax=[  -77.4572  -75.4157   36.7224   39.6242]; %经纬度范围
cax=[0 33];  %颜色范围
lat_mid=38; % 用于调节比例
```

不使用数据集附带代码就可以执行分析
访问数据集，在给定的时间获取三维场数据，将数据插值为常数 z，在 z 深度绘制结果

```
figure;
for i=1:length(url);
  nc{i}=ncgeodataset(url{i});
  jd{i}=nj_time(nc{i},var{i});  % 使用 nj_time
  itime=date_index(jd{i},dat);
  disp(['reading data from ' titl{i} '...'])

  % 这里使用 nj_tslice 不允许子集化或者有间隔：获得了一个特定时间阶的整个三维场：
  [s{i},g{i}]=nj_tslice(nc{i},var{i},itime);
  sz{i}=zsliceg(s{i},g{i}.z,depth);
```

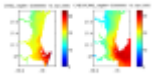
```
  a{i}=subplot(1,length(url),i);
  pcolorjw(g{i}.lon,g{i}.lat,double(sz{i}));colorbar
  axis(ax);
  caxis(cax);
  title(sprintf('%s,depth=%f: %s',titl{i},depth,datestr(g{i}.time)));
  set (a{i},'DataAspectRatio',[1 cos(lat_mid*pi/180) 1000] );

end
```

```
reading data from CH3D...
reading data from CHESROMS...
```



MATLAB® 7.13 出品

3.10 GEODEMO 4B



--在特定的时间步长和深度下，比较来自两个不同 CF 兼容结构化网格模型（CH3D 和 ROMS）的水平切片，使用地理子集获取数据子集。

CH3D

```
url{1}='http://testbedapps.sura.org/thredds/dodsC/estuarine_hypoxia/ch3d/agg.nc';
var{1}='salinity';
titl{1}='CH3D';
```

ROMS

```
url{2}='http://testbedapps.sura.org/thredds/dodsC/estuarine_hypoxia/chesroms_1tdo
/agg.nc';
var{2}='salt';
titl{2}='CHESROMS';
```

创建地理子集对象

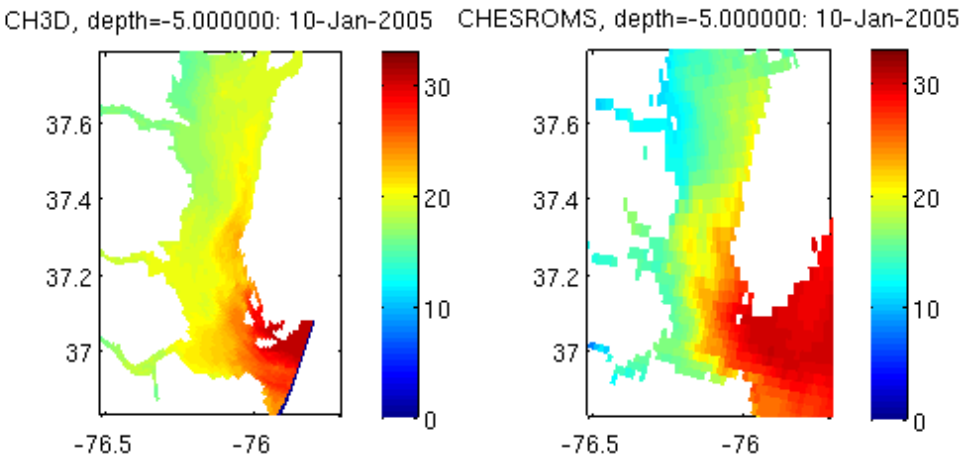
```
dat=[2005 1 10 0 0 0];  % Jan 10,2005 00:00 UTC
depth=-5;  % 5 米水平切面
ax=[ -76.5220  -75.7105   36.8248   37.7850]; %经纬度范围
cax=[0 33];  %颜色范围
lat_mid=38; % 用于调节比例

s.time=dat;
s.lon=ax(1:2);
s.lat=ax(3:4);
```

不使用数据集附带代码就可以执行分析
访问数据集，在给定的时间获取三维场数据，将数据插值为常数 z，在 z 深度绘制结果

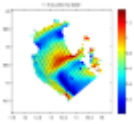
```
figure;
for i=1:length(url);
  nc{i}=ncgeodataset(url{i});
  % create a salinity geovariable object.  No data read yet.
```

```
  svar{i}=geovariable(nc{i},var{i});
  disp(['reading data from ' titl{i} '...'])
  % 这里使用地理子集，允许子集化或者有间隔。多次读取时间步骤，仅对特定的 zlevels 有效
  sub{i}=svar{i}.geosubset(s);
  sz{i}=zsliceg(squeeze(sub{i}.data),squeeze(sub{i}.grid.z),depth);
  a{i}=subplot(1,length(url),i);
  pcolorjw(sub{i}.grid.lon,sub{i}.grid.lat,double(sz{i}));colorbar
  axis(ax);
  caxis(cax);
  title(sprintf('%s,depth=%f: %s',titl{i},depth,datestr(sub{1}.grid.time)));
  set (a{i},'DataAspectRatio',[1 cos(lat_mid*pi/180) 1000] );
end
```

```
reading data from CH3D...
reading data from CHESROMS...
```



3.11 GEODEMO 6



-- 从 ROMS 等 C-GRID 模型中提取一层速度。

```
url='http://geoport.whoi.edu/thredds/dodsC/examples/bora_feb.nc';
hname='h';
uname='u';
vname='v';
aname='angle';
```

使用 NCTOOLBOX 访问数据
创建一个表示来自 opendap 端点的地理数据集对象

```
nc=ncgeodataset(url);

% 创建表示数据集中感兴趣变量的变量对象
uvar=nc.geovariable(uname);
vvar=nc.geovariable(vname);
hvar=nc.geovariable(hname);
avar=nc.geovariable(aname);
```

将 ROMS u 和 v 向量插值到与 hvar（rho 变量）相同的网格上
使用 uvar 和 vvar 对象以及 avar（角度）变量对象将速度矢量插值到 rho 网格点上，并根据角度变量旋转它们

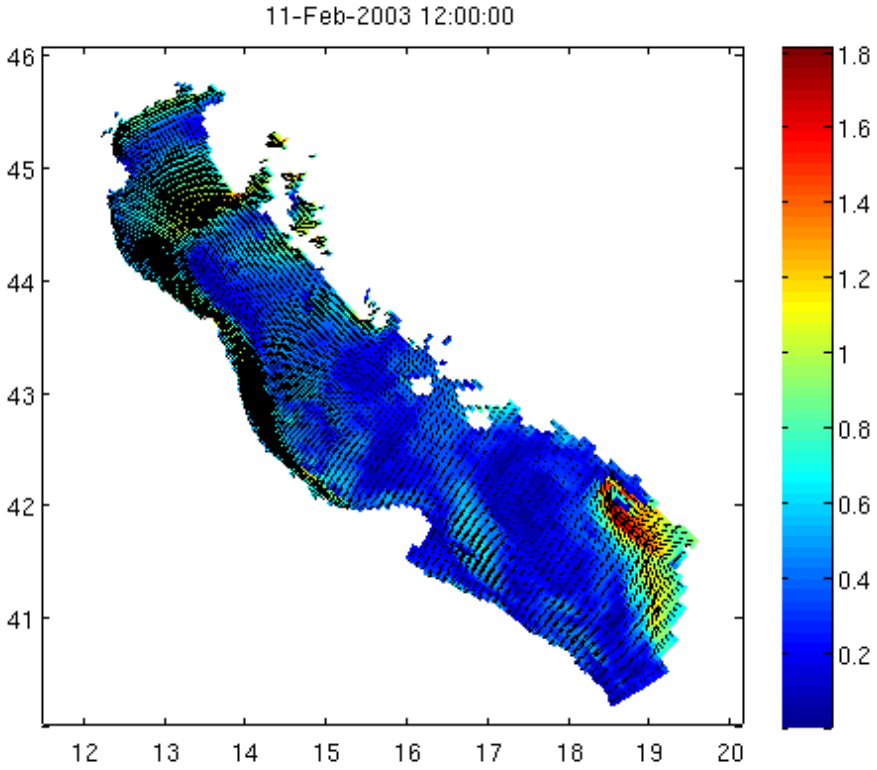
```
Uobj=hvar.getvectors(uvar,vvar,avar);
```

获取感兴趣索引上的数据

```
itime=3; % 第 3 时间阶
klev=-1; % 最后一层（顶层）
% 获取我们索引上的坐标信息
g=Uobj.grid(itime,klev,:,:);
```

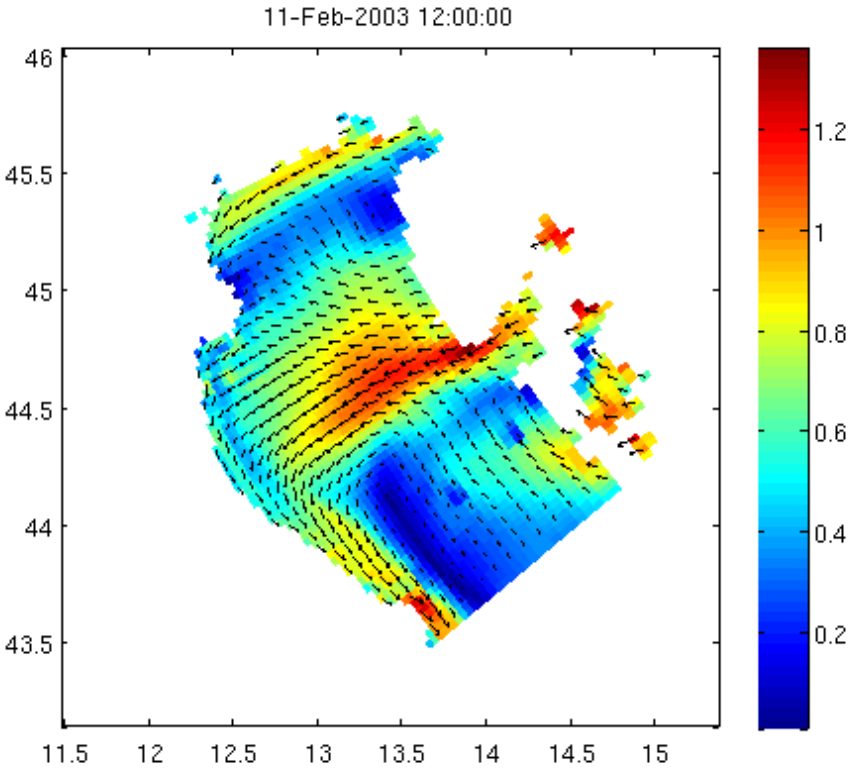
绘制矢量
在我们的索引和图上获得新矢量的大小，并覆盖新速度的矢量表示

```
figure;
pcolorjw(g.lon,g.lat,Uobj.magnitude(itime,klev,:,:));
colorbar;
arrows(g.lon(1:end,1:end),g.lat(1:end,1:end),...
   Uobj.vectors(itime,klev,1:end,1:end),0.08,'black');
title(datestr(g.time));
dasp(44);
```



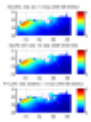
绘制矢量子集

```
figure;
pcolorjw(g.lon(1:58,1:70),g.lat(1:58,1:70),...
   Uobj.magnitude(itime,klev,1:58,1:70));
colorbar;
arrows(g.lon(1:2:58,1:2:70),g.lat(1:2:58,1:2:70),...
   Uobj.vectors(itime,klev,1:2:58,1:2:70),0.08,'black');
title(datestr(g.time));
dasp(44);
```



MATLAB® 7.13 出品

3.12 TEST CF UGRID3



-- 比较使用 UGRID 规范（http://bit.ly/cf_ugrid）的 3 种不同的非结构化网格模型中的水位，无需使用模型专用代码即可进行比较。

数据集
ADCIRC

```
titl{1}='ADCIRC';
uris{1}='http://testbedapps-
dev.sura.org/thredds/dodsC/in/und/adcirc/ike/ultralite/lr/vardrag/nowave/3d';
vars{1}='zeta';
time=[2008 9 13 06 0 0];

%%% SELFE
titl{2}='SELFE';
uris{2}='http://testbedapps.sura.org/thredds/dodsC/inundation/selfe/ike/3Dvrwoww'
;
vars{2}='elev';

%%% FVCOM
titl{3}='FVCOM';
uris{3}='http://testbedapps.sura.org/thredds/dodsC/inundation/FVCOM/ike/3Dvrwoww'
;
vars{3}='zeta';
```

建立绘图参数
框定图形外框

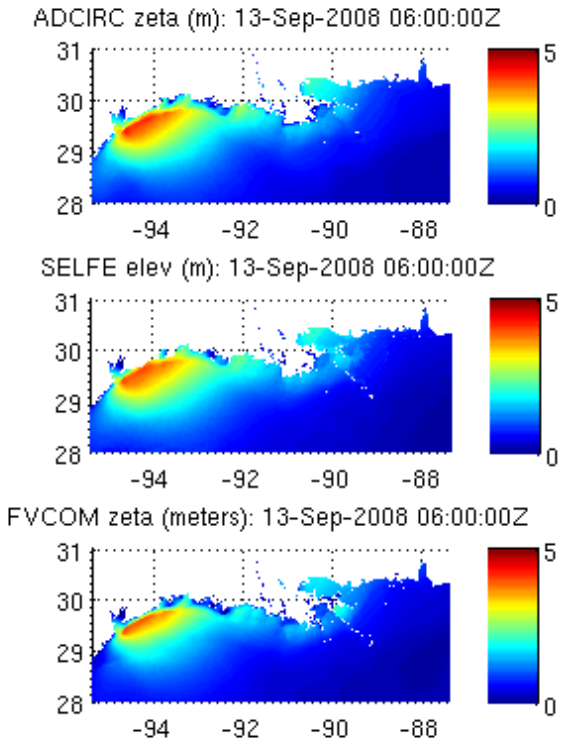
```
ax=[-95.4519  -87.3856   28.0   31.0];

%%% 图形颜色范围
cax=[0 5];
```

下面的循环中没有特定的模型！

```
for i=1:length(uris)
  tic
  % 初始化数据集对象
  nc=ncgeodataset(uris{i});
  %获取变量对象
  zvar=nc.geovariable(vars{i});
  % 找到坐标变量
  lon=zvar.getlondata(:);
  lat=zvar.getlatdata(:);
  tdat=zvar.timewindowij(time);
  itime=tdat.index;
  % 在指定的时间步读取所有节点的数据
  zeta=zvar.data(itime,:);
  % 获取网格变量名称（推理数组）
  gvar_name=zvar.attribute('mesh');
  gridvar=nc.geovariable(gvar_name);
  % 从网格变量中找到连通性数组变量
  trivar_name=gridvar.attribute('face_node_connectivity');

  % 获取连通性数组数据
  tri=nc{trivar_name}(:);
  [m,n]=size(tri);
  % 检查/修复连通性阵列的方向
  if m==3,
    tri=tri.';
  elseif n~=3
    disp('Error:Currently handling triangles only');return
  end
  subplot(length(uris),1,i)
  trisurf(tri,lon,lat,zeta);shading interp;view(2);colorbar;...
    axis(ax);caxis(cax);...
    title(sprintf('%s %s (%s): %sZ',titl{i},vars{i},...
    zvar.attribute('units'),datestr(tdat.time)));...
    % 根据平均纬度为经纬度图设置纵横比
  set (gca,'DataAspectRatio',[1 cos(mean(lat(:))*pi/180) 1] );

  toc
end
```

```
Elapsed time is 72.348191 seconds.
```

```
Elapsed time is 75.959380 seconds.
Elapsed time is 66.551909 seconds.
```



MATLAB® 7.13 出品