

2. matplotlib の導入

ここでは、3 章以降の作図に必要な Mac OS への matplotlib の導入について解説します。Python のバージョンは 3.8/3.9 を想定しており、このバージョンに対応したモジュールを導入します。インストールを行なった際は python3.8 ですが、python3.9 でも動作確認が終わりました。インストールには、Xcode と MacPorts を使います。既に最新版のインストールが済んでいる場合は、この章を飛ばしても問題ありません。なお、バージョン 3.7 を用いる場合は 2.3 節を、バージョン 3.6 を用いる場合は 2.4 節を参照してください。

最新版の Basemap のパッケージ (バージョン 1.2.2) には 2.5.6 節に記載した不具合があります。2.5.4 節、2.5.5 節の不具合はバージョン 1.2.2 で修正されました。2.5.6 節の不具合は残っており特定の図法で南極大陸を描く場合には問題となりますが、他の不具合については解消されたため、6 章のサンプルプログラムを動かす前に最新版に更新してください。この不具合が解消されていないため、6 章のサンプルプログラムの地図表示では、本稿の図とは南極大陸の地図表示が異なります。

本章では MacOS への導入方法を紹介していますが、python は Windows や Linux など、他の OS でも利用することが可能となっています。また、web 上で python を利用するための環境が Google 等で提供されており、例えば Google Colaboratory (グーグル・コラボレイトリー) は無料で利用することができます。とりあえず作図を試してみたい方向けに、3 章のチュートリアルを Google Colaboratory で行う方法を以下のページに載せています。

<https://yyousuke.github.io/matplotlib/matplotlib-tutorial.html>

2.1 MacPorts を使った matplotlib のインストール

2018 年 8 月最初のアップデートで、MacPorts に python バージョン 3.7 が追加され、2019 年 9 月に python バージョン 3.8 が、2020 年 10 月に python バージョン 3.9 が 2021 年 10 月に python バージョン 3.10 が追加されました。2020 年 4 月時点で python3.8 に対応したパッケージが増えていますので、3.8 に対応したものに書き換えました。また、2021 年 11 月時点で 3.9 での動作確認を行いました。バージョン 3.10 はまだリリースされたばかりで、パッケージが揃うのはこれからの段階です。

本節で紹介するインストールでは、2020 年 4 月 3 日時点で最新の macOS

Catalina (10.15.4)、Xcode11.4、MacPorts2.6.2 を用いました。2021 年 11 月時点で macOS Big Sur (11.6.1)、Xcode 13.1 と MacPorts2.7.1 での動作確認も行なっています。macOS Big Sur に更新する場合には、2.2 節の手順を参考にしてください。なお、Big Sur 11.1～11.6.1、Catalina 10.15.2～10.15.7、Xcode11.3～13.1、MacPorts2.6.2～2.7.1 でも同様に動作することを確認していますので、Catalina の場合も Big Sur の場合も最新版の導入をお勧めします。なお 2021 年 12 月時点の macOS Monterey (12.1) や Apple M1 システムオンチップでの動作確認は済んでいません。

Xcode は Mac の App Store で提供されているものです。App Store で Xcode を検索して入手します (図 2-1-1)。



図 2-1-1 Xcode の検索画面

Finder のアプリケーション一覧から Xcode.app を起動し、追加のパッケージをインストールします (図 2-1-2)。

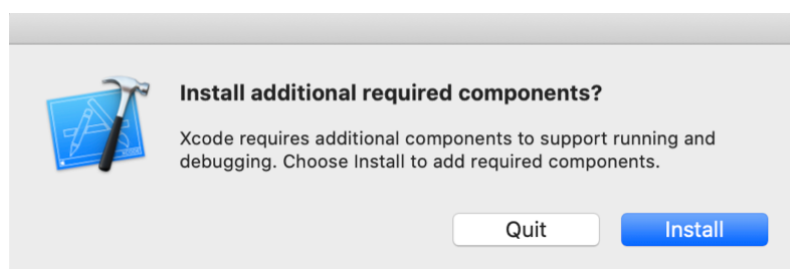


図 2-1-2 最初に Xcode を起動した時の追加のパッケージインストール画面

ターミナルを起動し、コマンドラインから続きの作業を行います。まず、

```
% sudo xcodebuild -license
```

を行い、ライセンスに同意します（最後に agree）。

```
% sudo xcode-select --install
```

でコマンドラインツールをインストールします（図 2-1-3）。



図 2-1-3 Xcode コマンドラインツールのインストール画面

次に MacPorts のインストールを行います。MacPorts の提供元 (<https://www.macports.org/install.php>) の OS X 10.15 Catalina 用のリンクから pkg ファイルをダウンロードします（図 2-1-4）。MacPorts は OS のバージョンに対応したものしか動作しないので、必ず OS のバージョンに対応する pkg ファイルをダウンロードしてください。

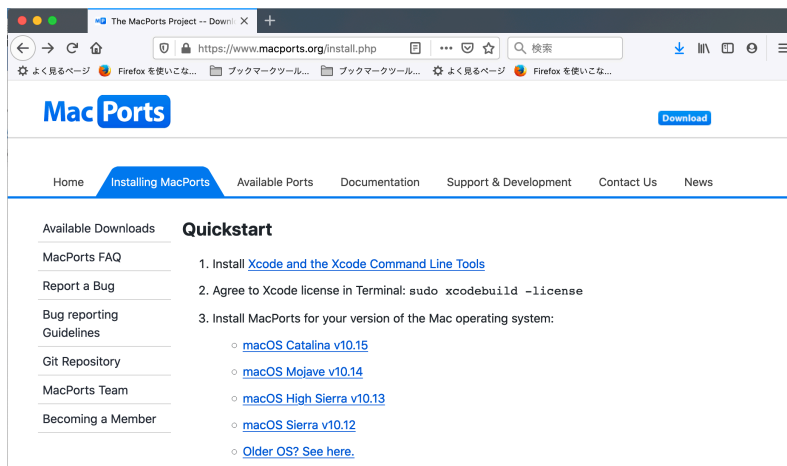


図 2-1-4 Macports のダウンロード画面

ダウンロードした pkg ファイルを Finder で開いてインストールします (図 2-1-5)。

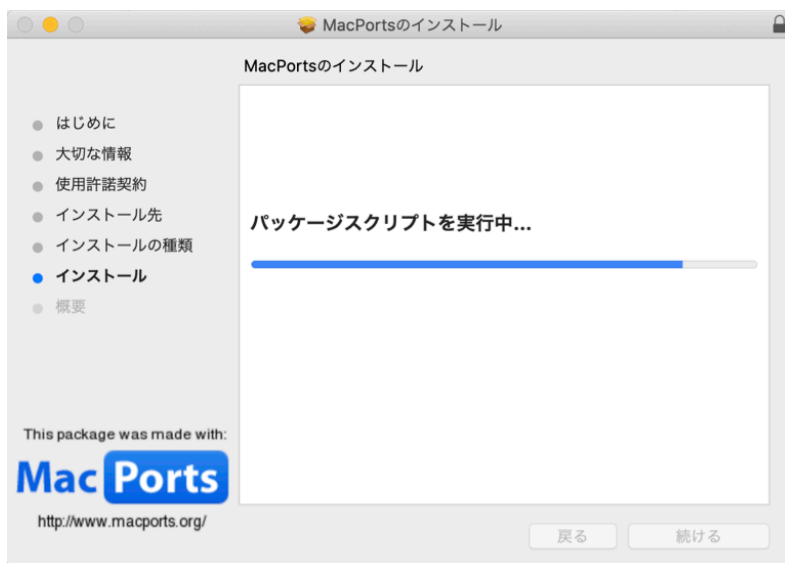


図 2-1-5 Macports のインストール画面

インストールが完了したら、ターミナルを開き、次のコマンドで MacPorts をアップデートします。MacPorts のコマンド名が port です。selfupdate では MacPorts とパッケージのリストを最新のものに更新します。

```
% sudo port selfupdate
```

MacPorts でパッケージをインストールするコマンドは、

```
% sudo port install パッケージ名
```

です。例えば python38 のパッケージをインストールするコマンドは、

```
% sudo port install python38
```

になっています。インストールは複数のパッケージを同時に行うことができ、

```
% sudo port install py38-numpy、py38-pandas、py38-scipy
```

のようにパッケージ名を並べます。

本稿のサンプルプログラムを動作させるために必要なパッケージは、python38 の他に python を対話的に実行するためのシェル py38-ipython、数値計算などを行う py38-numpy、py38-pandas、py38-scipy と、pandas で html データを解析する時に使う、py38-lxml、py38-html5lib、py38-beautifulsoup4 です。また、作図を行うために必要なパッケージは、matplotlib 本体の py38-matplotlib、Basemap 本体の py38-matplotlib-basemap、cartopy 本体の py38-cartopy の他、basemap に必要な py38-Pillow、py38-pyproj です。grib データを扱うために wgrib2、NetCDF データを扱うために netcdf、py38-netcdf4、HDF データを扱うために py38-h5py も必要です。

MacPorts のバージョン更新時にパッケージ名が変わることがありますので、もしパッケージ名でエラーが出た場合には、最新の情報を取得してください。なお、2018 年 8 月時点の MacPorts バージョン 2.5.4 から 2021 年 12 月時点の 2.7.1 までの間は、パッケージ名が変わっていないことを確認しています。

1 つのパッケージには依存関係のあるパッケージが多数あるため、次の順にインストールすることで、本稿の作図を行うサンプルプログラムを動作させるのに必要なパッケージを全て含むようになります。途中で依存関係のあるパッケージをインストールするか聞かれるので、Yes の方を選択します。

```
% sudo port install py38-pandas py38-scipy
```

```
% sudo port install py38-pip
% sudo port install py38-matplotlib-basemap py38-Pillow
% sudo port install py38-cartopy
% sudo port install py38-ipython
% sudo port install wgrib2 py38-netcdf4 py38-h5py
% sudo port install py38-beautifulsoup4
```

Python3.9 の場合には py38 を py39 に置き換えてください。パッケージのインストールが一通り終わったら、次のようなコマンドで python3、ipython3、pip3 のバージョンを 3.8 のものにしておきます。こうしておけば、python3 や ipython3、pip3 コマンドでバージョン 3.8 のものが起動します。

```
% sudo port select --set python3 python38
% sudo port select --set ipython3 py38-ipython
% sudo port select --set pip3 pip38
```

既に python2 のサポートが終了しているので、python と ipython、pip コマンドもバージョン 3.8 にしておいて良いかもしれません。

```
% sudo port select --set python python38
% sudo port select --set ipython py38-ipython
% sudo port select --set pip pip38
```

ここで紹介した MacPorts には、パッケージを検索する便利な機能が含まれています。この機能は次のように使います。

```
% port search パッケージ名の一部など
```

古いパッケージを更新する機能もあり、次のように使います。

```
% sudo port selfupdate (MacPorts とパッケージのリストを最新のものに)
% sudo port upgrade outdated (古いパッケージを更新)
```

パッケージのリストは時々更新されますが、不整合やバグを含んでいることがあるので、upgrade を実行する前に最新版で問題が起こっていないか確認したほうが良いでしょう。不整合やバグの情報は次のページにも載せています。

<https://yyousuke.github.io/matplotlib/info.html>

MacPorts でインストールした pip コマンドも、パッケージ管理に用います。MacPorts に含まれていないパッケージであっても、pip コマンドで探せることがあります。MacPorts と pip で同じパッケージが入っていると不具合が発生することがあるので、どちらか一方でインストールするようにして下さい。

pip の検索機能を使用する際には、次のように行います。

% pip3 search パッケージ名の一部など

pip コマンドによるパッケージのインストールは、次のように行います。

% sudo pip3 install パッケージ名

パッケージの詳細を表示するには、次のように行います。

% sudo pip3 show パッケージ名

インストールしたパッケージを削除する場合には、次のように行います。

% sudo pip3 uninstall パッケージ名

pip 自体のアップデートは、次のように行います。

% pip install -U pip

2.2 OS 更新時の手続き

Mac OSX は、2017 年 6 月の High Sierra (10.13)、2018 年 10 月の Mojave (11.4)、2019 年 10 月の Catalina (10.15)、2020 年 11 月の Big Sur (11.1) のように毎年更新されています。OS の更新を行った際には、MacPorts や Python も更新しておく必要があります。更新は MacOS、Xcode、MacPorts、Python の順に行います。ハードウェアによっては最新 OS に未対応のものもあるので、もし更新する場合には事前に確認してください。

High Sierra から Mojave へは「Mac App Store」で、Mojave から Catalina、Catalina から Big Sur へは、「システム環境設定」の「ソフトウェアアップデート」で更新を行うことができました（図 2-2-1）。



図 2-2-1 Catalina から Big Sur へのアップデート画面

Xcode は OS 更新時に自動更新されます。OS 更新後に App Store を開き、Xcode を検索して起動します。その際に、初期のインストール同様、追加のパッケージをインストールするか聞かれるので (図 2-1-2)、インストールを行います。次にターミナルを起動し、初期のインストール同様、ライセンスへの同意とコマンドラインツールのインストールを行います。Xcode を更新した場合にも、同じ手続きが必要となることがあります。

% sudo xcodebuild -license (最後に agree を入力しライセンスに同意)

% sudo xcode-select --install (コマンドラインツールのインストールに同意)

次に、MacPorts の提供元 (<https://www.macports.org/install.php>) から、インストールした OS に対応した pkg ファイルをダウンロードしてインストールします (図 2-2-2)。

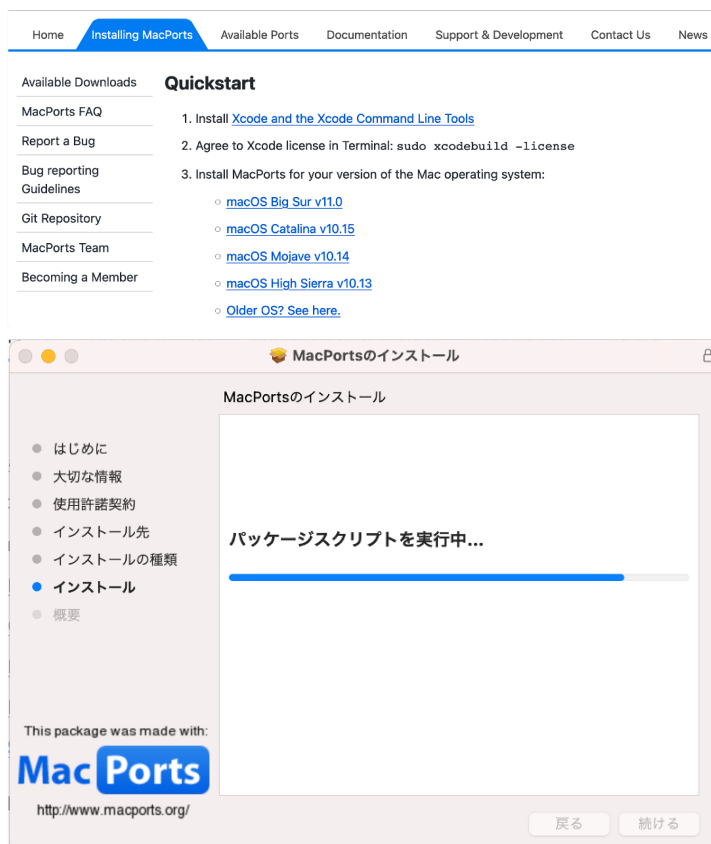


図 2-2-2 Big Sur 対応の MacPors ダウンロード画面

インストール後、ターミナルを開き MacPorts の移行を行います。

```
% port -qv installed > myports.txt (パッケージ情報を保存)
% sudo port -f uninstall installed (インストールされているものを削除)
% curl --location --remote-name https://github.com/macports/macports-contrib/raw/master/restore_ports/restore_ports.tcl
% chmod +x restore_ports.tcl
% sudo port selfupdate
% sudo ./restore_ports.tcl myports.txt
```

インストール状況によっては、途中でエラーが出て移行に失敗することがあります。その場合には、手動で python 関連のパッケージをインストールします。最初にインストールした時同様に、次の手順で確認していきます。

```
% sudo port install py38-pandas py38-scipy
% sudo port install py38-pip
% sudo port install py38-matplotlib-basemap py38-Pillow
% sudo port install py38-cartopy
% sudo port install py38-ipython
% sudo port install wgrib2 py38-netcdf4 py38-h5py
% sudo port install py38-beautifulsoup4
```

エラーが出たパッケージがあると、port clean を求めるメッセージが出力されて止まるので、その都度、

```
% sudo port clean パッケージ名
```

を行った後、再び port install を実行してインストールをやり直します。

なお、設定によっては Xcode コマンドラインツールのバージョンを認識できない場合があります。その際には、

<https://trac.macports.org/wiki/ProblemHotlist#reinstall-clt>

にアクセスして、Apple Developer site から対応するバージョンのものをダウンロードして下さい（図 2-2-3）。

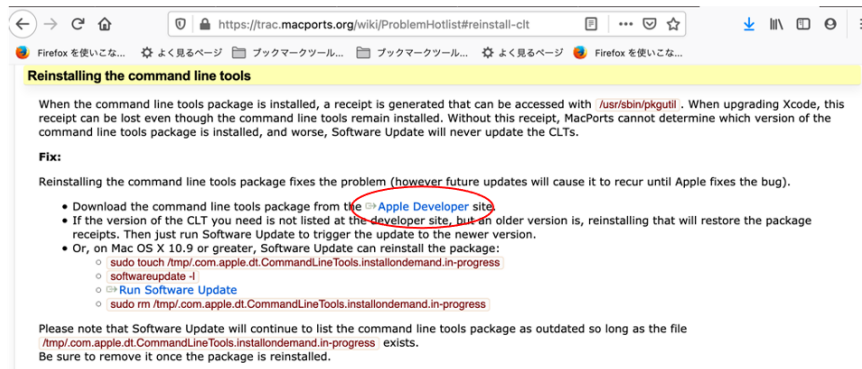


図 2-2-3 Xcode コマンドラインツールのダウンロード画面

2.3 Python3.7

2018 年 8 月最初のアップデートで、MacPorts に python37 が追加され、作図に必要なパッケージが順次揃っていきました。2019 年 12 月にはバージョン 3.7 でも動作確認を終えたため、python3.7 のドキュメントを追加しました。2020 年 4 月には最新版のバージョン 3.8 での動作確認お終えたため、デフォルトは 3.8 にしましたが、バージョン 3.7 でも動作します。MacPorts で python3.7 を導入するには、次の手順で行います。MacPorts のインストールまでは、2.1 節と同じ手順を用いてください。

```
% sudo port install py37-pandas py37-scipy
% sudo port install py37-pip
% sudo port install py37-matplotlib-basemap py37-Pillow
% sudo port install py37-cartopy
% sudo port install py37-ipython
% sudo port install wgrib2 py37-netcdf4 py37-h5py
% sudo port install py37-beautifulsoup4
```

パッケージのインストールが一通り終わったら、次のようなコマンドで python3、ipython3、pip3 のバージョンを 3.7 のものにしておきます。こうしておけば、python3 や ipython3、pip3 コマンドでバージョン 3.7 のものが起動します。

```
% sudo port select --set python3 python37
% sudo port select --set ipython3 py37-ipython
% sudo port select --set pip3 pip37
```

python3.7 のドキュメントを追加した際、2019 年 12 月 4 日時点の最新版である OSX Catalina (10.15.1)、Xcode11.2.1、MacPorts2.6.4 を用いたテストを行ないました。Catalina 10.15.1～10.15.4、Xcode11.2～11.4 でも同様に動作することを確認しています。

2.4 Python3.6

執筆開始時の 2018 年 8 月 9 日時点では、MacPorts に python37 が追加されていたものの、py37-ipython や py37-matplotlib-basemap、py37-pygrib、py37-pyproj が未整備であったため、python のバージョンは 3.6 を想定してサンプルプログラムを作成していました。2019 年 12 月にはバージョン 3.7、2020 年 4 月にはバージョン 3.8 でも動作確認を終え、新しいバージョンで解説していますが、バージョン 3.6 でも動作確認しています。なお python3.6 のパッケージは、執筆開始時点のバージョン 3.6.7 から 3.6.8 に更新されましたが、どちらも動作確認を行っています。Matplotlib は 3.0.0 から 3.1.1 で動作確認を行いました。

Python3.6 を使い本稿の作図を行うサンプルプログラムを動作させるには、以下の手順でインストールを行います。

```
% sudo port install py36-pandas
% sudo port install py36-pip
% sudo port install py36-matplotlib-basemap py36-Pillow
% sudo port install py36-ipython
% sudo port install wgrib2 py36-pygrib py36-netcdf4 py36-h5py
% sudo port install py36-beautifulsoup4
```

パッケージのインストールが一通り終わったら、次のようなコマンドで python3、ipython3、pip3 のバージョンを 3.6 のものにしておきます。

```
% sudo port select --set python3 python36
% sudo port select --set ipython3 py36-ipython
% sudo port select --set pip3 pip36
```

なお pygrib は、py36-pygrib を最後に更新されなくなっており、Python3.7 用の py37-pygrib や Python3.8 用の py38-pygrib が無いため、本稿では補足的に用いるのに留めています。

2.5 アップデートの不具合

本節では、2018 年 8 月の執筆開始以降に MacPorts で発生したパッケージの不具合について載せています。2.5.1 節から 2.5.3 節の不具合は、2019 年 12 月時点では全て解消されています。2.5.4 節のものは、2019 年 12 月以降のアップデートで、2.5.5 節のものは 2020 年 7 月のアップデートで発生するものです。

2.5.1 basemap-1.1.0 について

2018 年 10～11 月のバージョンでは、MacPorts でインストールされるバージョンが matplotlib-3.0.0 (から 3.0.2) と basemap-1.1.0 の組み合わせになっており、この組み合わせでは、basemap を使用する際にエラーが発生します。一時的な対処法としては、basemap のコードを次のように書き換えます。ディレクトリの 3.6 の部分は、使用する python のバージョンに依存します。

```
/opt/local/Library/Frameworks/Python.framework/Versions/3.6/lib/python  
3.6/site-packages/mpl_toolkits/basemap/__init__.py
```

17 行目：

```
from matplotlib.cbook import is_scalar, dedent
```

を

```
from matplotlib.cbook import is_scalar_or_string, dedent
```

4987 行目：

```
elif masked and is_scalar(masked):
```

を

```
elif masked and is_scalar_or_string(masked):
```

なお 2018 年 12 月中旬には basemap が 1.1.0 から 1.2.0 に更新され、matplotlib.cbook の is_scalar を使用しない変更が行われました。これにより上記のバグが解消されたので、バージョン 1.2.0 以降の basemap を使用する場合には問題ありません。

なお、1.1.0 から 1.2.0 の更新では、basemap の __init__.py の該当部分 (def_interp 内) が次のように変更されました。

basemap のコードの場所 (1.1.0 から 1.2.0 で変更なし)：

```
/opt/local/Library/Frameworks/Python.framework/Versions/3.6/lib/python  
3.6/site-packages/mpl_toolkits/basemap/__init__.py
```

バージョン 1.1.0 の def interp の最後 4983～4989 行目

```
if masked and isinstance(masked, bool):
    dataout = ma.masked_array(dataout)
    newmask = ma.mask_or(ma.getmask(dataout), xymask)
    dataout = ma.masked_array(dataout, mask=newmask)
elif masked and is_scalar(masked):
    dataout = np.where(xymask, masked, dataout)
return dataout
```

バージョン 1.2.0 の def_interp の最後 4992～4997 行目

```
if masked:
    newmask = ma.mask_or(ma.getmask(dataout), xymask)
    dataout = ma.masked_array(dataout, mask=newmask)
    if not isinstance(masked, bool):
        dataout = dataout.filled(masked)
return dataout
```

2.5.2 basemap-1.2.0 の 2019 年 4 月以降の更新について

2019 年 4 月頃に、MacPorts でインストールされる pyproj のバージョンが 2.1.3 になり、内部で使われる proj のバージョンが 5 から 6 に更新され、同時に、これまで参照できた pyproj.pyproj_datadir が廃止されました。Basemap-1.2.0 では、import 時に pyproj.pyproj_datadir を参照して epsg ファイルのパスを取得しているため、そのままでは basemap を読み込む際にエラーが発生します。

この時点での対処法としては、basemap のコードを変更し、2019 年 4 月の更新前に参照していた proj5 の epsg を使う方法がありました。ディレクトリの 3.6 の部分は、使用する python のバージョンに依存します。

/opt/local/Library/Frameworks/Python.framework/Versions/3.6/lib/python
3.6/site-packages/mpl_toolkits/basemap/__init__.py

152 行目：

```
epsgf = open(os.path.join(pyproj.pyproj_datadir, 'epsg'))
```

を

```
# エラー回避のため、proj5 の epsg ファイルを読み込み
pyproj_datadir = "/opt/local/lib/proj5/share/proj/"
epsgf = open(os.path.join(pyproj_datadir,'epsg'))
```

2019 年 11 月時点で basemap-1.2.1 へ更新されバグフィックスが行われたため、バージョン 1.2.1 以降を使用する場合には問題ありません。

2.5.3 matplotlib3.1.0 のログプロットについて

matplotlib-3.1.0 では、図 5-5-6、図 5-5-7 の作図のように鉛直軸をログプロットにした場合は、正しく鉛直軸の設定を変更しても 10 hPa が下、1000 hPa が上ようになってしまっています。2019 年 8 月末の matplotlib-3.1.1 への更新時にバグフィックスが行われたため、バージョン 3.1.1 以降の matplotlib を使用する場合には問題ありません。

2.5.4 正射投影図法で北緯 45 度よりも北側を指定した場合の不具合

2019 年 12 月頃から MacPorts でインストールされる Basemap-1.2.0、geos-3.8.0 の組み合わせで発生する不具合です。6.2 節で紹介している正射投影図法では、視点の中心となる緯度を設定することができ、その際に 45 度よりも北側に設定すると作図の際に python が異常終了します。Basemap の開発は既に終了していて修正コードの提供は期待できないため、次のようにエラー回避を行います。ディレクトリの 3.8 の部分は、使用する python のバージョンに依存し、3.7.x なら 3.7、3.6.x なら 3.6 です。

```
/opt/local/Library/Frameworks/Python.framework/Versions/3.8/lib/python
3.8/site-packages/mpl_toolkits/basemap/__init__.py
1428 行目と 1429 行目の間に次の 1 行を挿入する。
```

```
b = np.where(np.isposinf(np.where(np.isneginf(b),-1.e20,b)),1.e20,b)
```


2.5.5 Basemap をインポートできない不具合

2020 年 7 月 20 日頃のアップデートに伴い、MacPorts でインストールされる matplotlib がバージョン 3.2.2 から 3.3.0 になりました。それに伴い matplotlib.cbook では dedent 関数が削除されました。他方、Basemap はバージョン 1.2.1 のまま更新されないため、mpl_toolkits.basemap をインポートしようとした際に、存在しない dedent 関数を呼び出そうとしてエラーが発生します。この修正は最新の Basemap バージョン 1.2.2 で適用されているため、バージョン 1.2.2 を導入してください。

```
File"/opt/local/Library/Frameworks/Python.framework/Versions/3.8/lib/python3.8/site-packages/mpl_toolkits/basemap/__init__.py", line 50, in <module>
    from .proj import Proj
File"/opt/local/Library/Frameworks/Python.framework/Versions/3.8/lib/python3.8/site-packages/mpl_toolkits/basemap/proj.py", line 6, in <module>
    from matplotlib.cbook import dedent
ImportError: cannot import name 'dedent' from 'matplotlib.cbook'
(/opt/local/Library/Frameworks/Python.framework/Versions/3.8/lib/python3.8/site-packages/matplotlib/cbook/__init__.py)
```

エラーを回避するため、
/opt/local/Library/Frameworks/Python.framework/Versions/3.8/lib/python3.8/site-packages/mpl_toolkits/basemap/proj.py
に対して以下のような修正を行います。

修正前（6 行目）

```
from matplotlib.cbook import dedent
```

修正後（6 行目から 9 行目）

```
try:
    from inspect import cleandoc as dedent
except ImportError:
    from matplotlib.cbook import dedent
```

2.5.6 Basemap で南極大陸を描く際の不具合

Basemap1.2.2 で行われた更新後に、一部の図法で南極大陸が描かれないか東半球側にしか描かれない不具合が発生しています。現在確認しているものは、正距円筒図法、メルカトル図法、ロビンソン図法、ランベルト正積円筒図法、ミラー図法です。なお、正射投影図法、ランベルト正角円錐図法、極投影図法、正距方位図法については問題なく作図できています。