

MeteoLab

一个用于气象数据诊断分析的 Matlab 工具箱

Dy

2022 年 4 月

目 录

一、MeteoLab 简介	4
二、安装步骤	4
三、函数使用说明	4
3.1 基本常数	4
3.2 基础物理量	4
科氏参数	4
经纬度网格距	5
空气密度	5
散度	5
垂直相对涡度	5
绝对涡度	6
水平平流项	6
3.3 差分/梯度/平滑	6
水平梯度	6
垂直梯度	6
拉普拉斯项（二维）	7
拉普拉斯项（垂直）	7
五点平滑	7
九点平滑	7
3.4 动力相关量	8
地转风	8
Q 矢量	8
正压模式下的位势涡度	8
Ertel 位涡	9
湿位涡	9
流函数	9
势函数	10
3.5 热力相关量	11
位温	11
相当位温	11
虚温	11
虚位温	11

干绝热递减率	11
湿绝热递减率	12
非绝热加热率	12
锋生函数	12
3.6 水汽相关量	13
露点温度	13
抬升凝结温度	13
单层水汽通量/散度	13
饱和水汽压	13
饱和混合比	13
混合比	14
饱和比湿	14
3.7 大气稳定性	14
静力稳定度	14
四、高级功能	14
1. Shuman-Shapiro 滤波	14
2. 涡度收支诊断	14
3. 台风诊断	15
二维直角坐标插值到二维极坐标系	15
三维直角坐标插值到柱坐标系	15
极坐标系插值到二维直角坐标系	16
柱坐标系插值到三维直角坐标系	16
切向风和径向风	16
绝对角动量	17

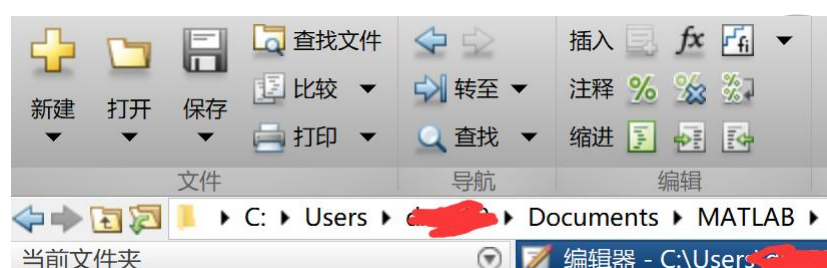
MeteoLab 使用说明

一、MeteoLab 简介

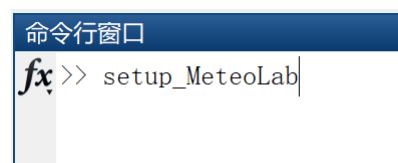
MeteoLab 是基于 Matlab 开发的，专门用于气象数据诊断和分析的工具箱，包含气象物理量诊断相关的基本函数，可快速计算出所需的气象物理量。可用于气象科研、数据可视化等领域。需要在 Matlab2016a 以上版本运行。

二、安装步骤

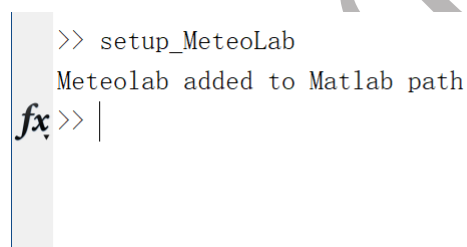
1. 需要先将当前文件夹设置为工具箱安装路径



2. 在命令行输入 setup_MeteoLab



3. 出现 Meteolab added to Matlab path, 即为启动成功



三、函数使用说明

3.1 基本常数

3.2 基础物理量

科氏参数

$$f = 2\Omega \sin\varphi$$

$$\beta = 2\Omega \cos\varphi$$

`function [f,b]=coriolis_parameter(lat,lon)`

输入：

纬度 **lat**(n)、
经度 **lon**(m) (可选)

输出:

科氏参数 **f**(n,m) (单位: /s)、
 β 项(科氏参数随纬度的导数)**b**(n,m) (单位: /s)
如果不输入经度, 则为: **f**(n)、**b**(n)

经纬度网格距

```
function [dx,dy]=latlon2delta(lat,lon)
```

输入:

纬度 **lat**(n)、
经度 **lon**(m)

输出:

纬向、经向网格距(目前仅适用于等经纬度和墨卡托投影网格)
dx,dy(n,m) (单位: m)

空气密度

$$\rho = \frac{p}{RT}$$

```
function rho=density(T,pressure)
```

输入:

温度 **T**(p,n,m) (单位: K)、
气压 **pressure**(p) (单位: Pa)

输出:

空气密度 **rho**(p,n,m) (单位: kg/m^3)

散度

$$Div = \nabla_h \cdot \vec{V}_h$$

```
function dv=divergence_2d(U,V, dx,dy,lat)
```

输入:

经向风、纬向风 **U,V**(n,m) (单位: m/s)、
纬向、经向网格距 **dx,dy**(n,m) 或 **dx,dy**(1) (单位: m)、
纬度 **lat**(n) (可选)

输出:

水平散度 **dv**(n,m) (单位: /s)

垂直相对涡度

$$\zeta = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$$

```
function rv=vorticity_2d(U,V,dx,dy,lat)
```

输入:

经向风、纬向风 $U, V(n, m)$ (单位: m/s)、
纬向、经向网格距 $dx, dy(n, m)$ 或 $dx, dy(1)$ (单位: m)、
纬度 $lat(n)$ (可选)

输出:

垂直相对涡度 $rv(n, m)$ (单位: /s)

绝对涡度

$$\zeta_a = f + \zeta$$

`function Ca = absolute_vorticity(U,V,lat,dx,dy)`

输入:

经向风、纬向风 $U, V(n, m)$ (单位: m/s)、
纬向、经向网格距 $dx, dy(n, m)$ 或 $dx, dy(1)$ (单位: m)、
纬度 $lat(n)$

输出:

绝对涡度 $Ca(n, m)$ (单位: /s)

水平平流项

$$adv = \vec{V} \cdot \nabla F$$

`function adv=advection_2d(F,U,V,dx,dy)`

输入:

物理量 $F(n, m)$ 、
经向风、纬向风 $U, V(n, m)$ (单位: m/s)、
纬向、经向网格距 $dx, dy(n, m)$ 或 $dx, dy(1)$ (单位: m)

输出:

水平平流项 $adv(n, m)$

3.3 差分/梯度/平滑

水平梯度

`function [gradx,grady]=gradient_2d(F,dx,dy)`

输入:

物理量 $F(n, m)$ 、
纬向、经向网格距 $dx, dy(n, m)$ 或 $dx, dy(1)$ (单位: m)

输出:

等压面/等位势面上的水平梯度 $gradx, grady(n, m)$

垂直梯度

`function [gradp]=gradient_vert(F,vertical)`

输入:

物理量 $F(p,n,m)$ 、

垂直方向的坐标（气压或高度） $vertical(p)$ （单位：Pa）

输出：

物理量的垂直梯度 $gradp(p,n,m)$

拉普拉斯项（二维）

$$\Delta F = \frac{\partial^2 F}{\partial x^2} + \frac{\partial^2 F}{\partial y^2}$$

`function Lap=Laplacian_2d(F,dx,dy)`

输入：

物理量 $F(n,m)$ 、

纬向、经向网格距 $dx,dy(n,m)$ （单位：m）

输出：

等压面/等位势面上的拉普拉斯项 $Lap(n,m)$

拉普拉斯项（垂直）

`function Lap_vert=Laplacian_vert(F,vertical)`

输入：

物理量 $F(p,n,m)$ 、

垂直方向的坐标（气压或高度） $vertical(p)$ （单位：Pa）

输出：

物理量垂直方向的拉普拉斯项 $Lap_vert(p,n,m)$

五点平滑

$$\widetilde{F}_{i,j} = F_{i,j} + \frac{s}{4}(F_{i+1,j} + F_{i-1,j} + F_{i,j+1} + F_{i,j-1} - 4F_{i,j})$$

`function Fl=smth5(F,s)`

输入：

物理量 $F(n,m)$ 、

平滑系数 s

输出：

五点平滑后的数据 $Fl(n,m)$

九点平滑

$$\begin{aligned}\widetilde{F}_{i,j} = F_{i,j} + \frac{s(1-s)}{2}(F_{i+1,j} + F_{i-1,j} + F_{i,j+1} + F_{i,j-1} - 4F_{i,j}) \\ + \frac{s^2}{4}(F_{i+1,j+1} + F_{i-1,j-1} + F_{i-1,j+1} + F_{i+1,j-1} - 4F_{i,j})\end{aligned}$$

`function Fl=smth9(F,s)`

输入：

物理量 $\mathbf{F}(\mathbf{n},\mathbf{m})$ 、

平滑系数 s

输出:

九点平滑后的数据 $\mathbf{F1}(\mathbf{n},\mathbf{m})$

3.4 动力相关量

地转风

$$U_g = -\frac{g}{f} \frac{\partial H}{\partial y}$$

$$V_g = \frac{g}{f} \frac{\partial H}{\partial x}$$

`function [Ug,Vg]=geostrophic_wind(H,dx,dy,f)`

输入:

位势高度 $\mathbf{H}(\mathbf{n},\mathbf{m})$ (单位: gpm)、

纬向、经向网格距 $\mathbf{dx},\mathbf{dy}(\mathbf{n},\mathbf{m})$ 或 $\mathbf{dx},\mathbf{dy}(1)$ (单位: m)、

科氏参数 $\mathbf{f}(\mathbf{n},\mathbf{m})$ (单位: /s)

输出:

地转风场 $\mathbf{Ug},\mathbf{Vg}(\mathbf{n},\mathbf{m})$ (单位: m/s)

Q 矢量

$$Q_x = -\frac{R}{p} \left(\frac{\partial \vec{V}_g}{\partial x} \cdot \nabla T \right)$$

$$Q_y = -\frac{R}{p} \left(\frac{\partial \vec{V}_g}{\partial y} \cdot \nabla T \right)$$

`function [Qx,Qy]=qvector_isobaric(T,Ug,Vg,pressure,dx,dy)`

输入:

温度 $\mathbf{T}(\mathbf{n},\mathbf{m})$ (单位: K)、

地转风场 $\mathbf{Ug},\mathbf{Vg}(\mathbf{n},\mathbf{m})$ (单位: m/s)、

气压(Pa) $\mathbf{pressure}(1)$ (单位: Pa)、

纬向、经向网格距 $\mathbf{dx},\mathbf{dy}(\mathbf{n},\mathbf{m})$ (单位: m)

输出:

Q 矢量 $\mathbf{Q}(\mathbf{n},\mathbf{m})$ (单位: $m^2 \cdot kg^{-1}s^{-1}$)

正压模式下的位势涡度

$$PV = \frac{f + \zeta}{H}$$

`function pv = potential_vorticity_barotropic(U,V,H,lat,dx,dy)`

输入:

风场 $\mathbf{U},\mathbf{V}(\mathbf{n},\mathbf{m})$ (单位: m/s)、

位势高度场 $\mathbf{H}(\mathbf{n},\mathbf{m})$ (单位: gpm)、

纬度 **lat**(n)、

纬向、经向网格距 **dx,dy**(n,m) （单位： m）

输出：

位涡 **pv**(n,m) （单位： PVU）

Ertel 位涡

$$pv = (\zeta_\theta + f) \left(-g \frac{\theta}{p} \right)$$

function **pv** = **Ertel_potential_vorticity**(theta,U,V,dx,dy,lat,pressure)

输入：

位温 **theta**(p,n,m) （单位： K）、

风场 **U,V**(p,n,m) （单位： m/s）、

纬向、经向网格距 **dx,dy**(n,m) （单位： m） 、

纬度 **lat**(n)、

气压 **pressure**(p) （单位： Pa）

输出：

Ertel 位涡 **pv**(p,n,m) （单位： PVU）

湿位涡

$$MPV = -g\zeta_\theta \frac{\partial \theta_e}{\partial p} = MPV1 + MPV2$$

$$MPV1 = -g(\zeta + f) \frac{\partial \theta_e}{\partial p}$$

$$MPV2 = -g\zeta \frac{\partial \theta_e}{\partial p} + g \left(\frac{\partial v}{\partial p} \frac{\partial \theta_e}{\partial x} - \frac{\partial u}{\partial p} \frac{\partial \theta_e}{\partial y} \right)$$

function [**MPV**,**MPV1**,**MPV2**] = ...

moist_potential_vorticity(RH,T,U,V,lat,dx,dy,pressure)

输入：

相对湿度 **RH**(p,n,m) （单位： 小数）、

温度 **T**(p,n,m) （单位： K）、

风场 **U,V**(p,n,m) （单位： m/s）、

纬度 **lat**(n)、

纬向、经向网格距 **dx,dy**(n,m) （单位： m）、

气压 **pressure**(p) （单位： Pa）

输出：

湿位涡 **pv**(p,n,m)

流函数

$$\Delta\psi = \zeta$$

function [**psi**,**u**,**v**] = **stream_function**(C,dx,dy,eps)

输入:

相对涡度 $C(n,m)$ (单位: $/s$)、

纬向、经向网格距 $dx, dy(n,m)$ (单位: m)、

收敛精度 eps : 默认为 10^{-7} , 可以不输入

输出:

流函数 $psi(n,m)$

无辐散风场 $u, v(n,m)$ (单位: m/s)

势函数

$$\Delta\phi = -D$$

`function [phi,u,v] = potential_function(D,dx,dy,eps)`

输入:

散度 $D(n,m)$ (单位: $/s$)、

纬向、经向网格距 $dx, dy(n,m)$ (单位: m)、

收敛精度 eps : 默认为 10^{-7} , 可以不输入

输出:

势函数 $phi(n,m)$

无旋风场 $u, v(n,m)$ (单位: m/s)

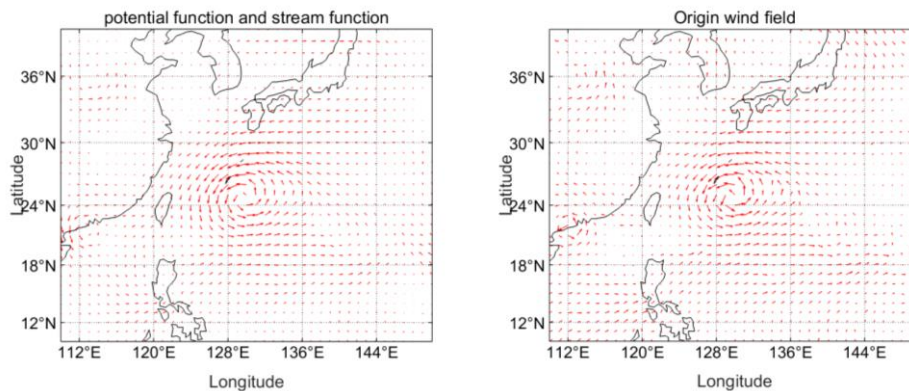


图 3.1 使用流函数和势函数程序计算得到的风场（左）与实际风场（右）的对比

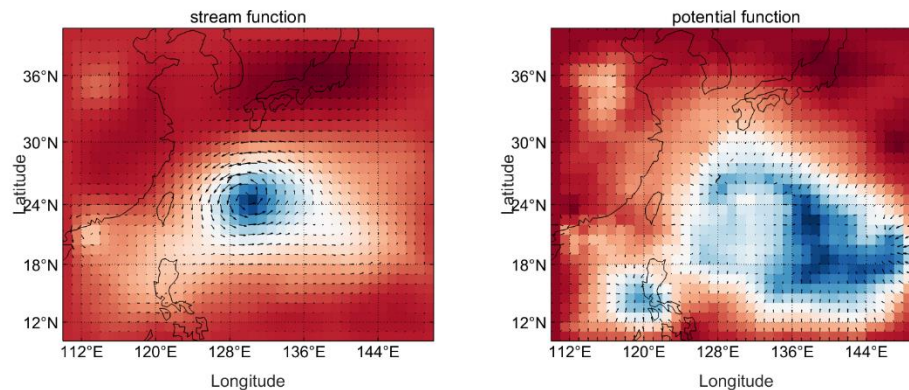


图 3.2 计算得到的流函数场、无辐散风（左）和势函数场、无旋风场

3.5 热力相关量

位温

`function theta=potential_temperature(T,pressure)`

输入:

温度 `T(p,n,m)` (单位: K)、

气压 `pressure(p)` (单位: Pa)

输出:

位温 `theta(p,n,m)` (单位: K)

相当位温

`function theta_e=equivalent_potential_temperature(RH,T,pressure)`

输入:

相对湿度 `RH(p,n,m)` (单位: 小数)、

温度 `T(p,n,m)` (单位: K)、

气压 `pressure(p)` (单位: Pa)

输出:

相当位温 `theta_e(p,n,m)` (单位: K)

虚温

`function tv = virtual_temperature(T,q)`

输入:

温度 `T` (单位: K)、

比湿 `q` (单位: kg/kg)

输出:

虚温 `tv` (单位: K)

虚位温

`function theta_v = virtual_potential_temperature(theta,q)`

输入:

位温 `theta` (单位: K)、

比湿 `q` (单位: kg/kg)

输出:

虚位温 `theta_v` (单位: K)

干绝热递减率

$$\gamma_d = \frac{dT}{dp} = \frac{RT}{c_p p}$$

`function gamma_d = dry_lapse(T,pressure)`

输入:

温度 `T(p,n,m)` 或 `T(n,m)`或 `T(p)` (单位: K)、

气压 **pressure(p)** 或 **pressure(1)** (单位: Pa)

输出:

干绝热递减率 **gamma_d(p,n,m)**或 **gamma_d(n,m)**或 **gamma_d(p)** (单位: K/Pa)

湿绝热递减率

$$\gamma_s = \frac{dT}{dp} = \frac{1}{p} \frac{R_d T + L_v r_s}{c_{pd} + \frac{L_v^2 r_s \epsilon}{R_d T^2}}$$

function gamma_s = moist_lapse(T,pressure)

输入:

温度 **T(p,n,m)** 或 **T(n,m)**或 **T(p)** (单位: K)、

气压 **pressure(p)** 或 **pressure(1)** (单位: Pa)

输出:

湿绝热递减率 **gamma_s(p,n,m)**或 **gamma_s(n,m)**或 **gamma_s(p)**(单位: K/Pa)

非绝热加热率

$$\frac{d\theta}{dt} = \omega \left(\frac{\partial \theta}{\partial p} - \frac{\gamma_m}{\gamma_d} \frac{\theta}{\theta_e} \frac{\partial \theta_e}{\partial p} \right)$$

function H = diabatic_heating(T,Omega,pressure,RH)

输入:

温度 **T(p,n,m)** (单位: K)、

P 坐标垂直速度 **Omega(p,n,m)** (单位: Pa/s)、

气压 **pressure(p)** (单位: Pa)

相对湿度 **RH(p,n,m)** (单位: 小数)、

输出:

非绝热加热率 **H(p,n,m)** (单位: K/s)

锋生函数

$$F \approx -\frac{1}{|\nabla_h \theta|} \left\{ \left[\left(\frac{\partial \theta}{\partial x} \right)^2 \frac{\partial u}{\partial x} + \left(\frac{\partial \theta}{\partial y} \right)^2 \frac{\partial v}{\partial y} \right] + \left[\frac{\partial \theta}{\partial x} \frac{\partial \theta}{\partial y} \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) \right] \right\}$$

function F = frontogenesis(theta,U,V,dx,dy)

输入:

温度 **T(n,m)** (单位: K)、

风场 **U,V(n,m)** (单位: m/s)、

纬向、经向网格距 **dx,dy(n,m)** (单位: m)

输出:

水平锋生函数 **F(n,m)** (单位: K/(m.s))

3.6 水汽相关量

露点温度

```
function Td=Dewpoint(RH,T)
```

输入:

相对湿度 **RH** (单位: 小数)、

温度 **T** (单位: K)

输出:

露点温度 **Td** (单位: K)

抬升凝结温度

```
function Tlcl=T_lcl(T,Td)
```

输入:

温度 **T** (单位: K)、

露点温度 **Td** (单位: K)

输出:

抬升凝结温度 **Tlcl** (单位: K)

单层水汽通量/散度

```
function [qu,qv,qd]=vapor_flux(Q,U,V,dx,dy,lat)
```

输入:

比湿 **Q(n,m)** (单位: kg/kg)、

风场 **U,V(n,m)** (单位: m/s)、

纬向、经向网格距 **dx,dy(n,m)** (单位: m)

纬度 **lat(n)** (可选)

输出:

水汽通量 **qu,qv(n,m)** (单位: $kg \cdot m^{-1}s^{-1}$)

水汽通量散度 **qd(n,m)** (单位: $kg \cdot m^{-2}s^{-1}$)

饱和水汽压

```
function es=saturation_vapor_pressure(T)
```

输入:

温度 **T** (单位: K)

输出:

饱和水汽压 **es** (单位: Pa)

饱和混合比

```
function rs=saturation_mixing_ratio(T,pressure)
```

输入:

温度 **T(p,n,m)** 或 **T(n,m)**或 **T(p)** (单位: K)、

气压 `pressure(p)` 或 `pressure(1)` (单位: Pa)

输出:

饱和混合比 `rs(p,n,m)`或 `rs(n,m)`或 `rs(p)` (单位: kg/kg)

混合比

```
function r = mixing_ratio(e,pressure)
```

输入:

水汽压 `e` (单位: Pa 或 hPa)、

气压 `pressure` (单位: Pa 或 hPa)

输出:

混合比 `r` (单位: kg/kg)

饱和比湿

```
function qs = saturation_specific_humidity(es,pressure)
```

输入:

饱和水汽压 `es` (单位: Pa 或 hPa)、

气压 `pressure` (单位: Pa 或 hPa)

输出:

饱和比湿 `qs` (单位: kg/kg)

3.7 大气稳定性

静力稳定度

```
function [ss]=static_stability(T,pressure)
```

输入:

温度 `T(p,n,m)`、

气压 `pressure(p)`

输出:

静力稳定度 `ss(p,n,m)`

四、高级功能

1. Shuman-Shapiro 滤波

```
function [synoptic_scale,meso_scale]=  
shuman_shapiro_filter(F,s,option)
```

2. 涡度收支诊断

```
function [H_adv,V_adv,Tilt,Dive]=  
vorticity_equation(U,V,W,vertical,dx,dy,lat)
```

%输入三维的风场、垂直坐标 (气压或高度)、水平格点间距、纬度

%!!!!!!!!!!!!!! 注意

%输入气象要素信息必须为 F(vertical_dim,y_dim,x_dim)

%如果不是请用 matlab 自带的 permute 函数调整维度
%如果是在等经纬度网格上计算，请输入一维的纬度数组
%如果是在非经纬度网格上计算，请用一个具体的纬度代替
%如用 45N 代替，则 lat 变量输入 45
%输出信息为涡度方程右侧的
%水平涡度平流项、垂直涡度平流项、扭转项、辐散项

3. 台风诊断

二维直角坐标插值到二维极坐标系

```
function [F,theta,r,lat_n,lon_n] = hurricane_cart2pol ...  
    (F0,ctr_lat,ctr_lon,lat,lon,Radius,Nr,Ntheta,option)  
%% 功能：以涡旋中心为原点，将二维直角坐标插值到极坐标  
%使用方法：  
%输入变量：F0：气象要素，ctr_lat,ctr_lon：涡旋中心经纬度  
%lat,lon：一维经纬度向量，R：需要插值的极坐标半径范围（km 或度）  
%Nr：极坐标径向格点个数，Ntheta：方位格点个数  
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
%!!! option：径向方向的单位是距离还是度，是距离则输入 0，度则输入 1  
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
%输出变量：F：插值后的气象要素，theta：方位坐标，r：径向坐标  
%lat_n,lon_n：插值后气象要素的经纬度坐标
```

三维直角坐标插值到柱坐标系

```
function [F,theta,r,lat_n,lon_n] = hurricane_cart2cyl ...  
    (F0,ctr_lat,ctr_lon,lat,lon,vertical,Radius,Nr,Ntheta,option)  
%% 功能：以涡旋中心为原点，将三维直角坐标插值到柱坐标  
%使用方法：  
%输入变量：F0：气象要素，ctr_lat,ctr_lon：涡旋中心经纬度  
%lat,lon：一维经纬度向量，vertical：一维垂直坐标（气压或高度）  
%R：需要插值的极坐标半径范围（km 或度）  
%Nr：极坐标径向格点个数，Ntheta：方位格点个数  
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
%!!! option：径向方向的单位是距离还是度，是距离则输入 0，度则输入 1  
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  
%输出变量：F：插值后的气象要素，theta：方位坐标，r：径向坐标  
%lat_n,lon_n：插值后气象要素的经纬度坐标
```


极坐标系插值到二维直角坐标系

```
function F = hurricane_pol2cart
    (F0,F_base,theta,r,ctr_lat,ctr_lon,lat,lon,option)

%% 功能：以涡旋中心为原点，将极坐标插值到二维直角坐标
%使用方法：
%输入变量：F0：气象要素，theta：方位坐标，r：径向坐标
%ctr_lat,ctr_lon：涡旋中心经纬度
%lat,lon：一维经纬度向量
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%!!! option：径向方向的单位是距离还是度，是距离则输入0，度则输入1
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%输出变量：F：插值后的气象要素，
```

柱坐标系插值到三维直角坐标系

```
function F = hurricane_cyl2cart
    (F0,F_base,vertical,theta,r,ctr_lat,ctr_lon,lat,lon,option)

%使用方法：
%输入变量：F0：气象要素，F_base：等经纬度网格上的数据
%vertical：垂直坐标，theta：方位坐标，r：径向坐标
%ctr_lat,ctr_lon：涡旋中心经纬度
%lat,lon：一维经纬度向量
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%!!! option：径向方向的单位是距离还是度，是距离则输入0，度则输入1
%!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%输出变量：F：插值后的气象要素
```

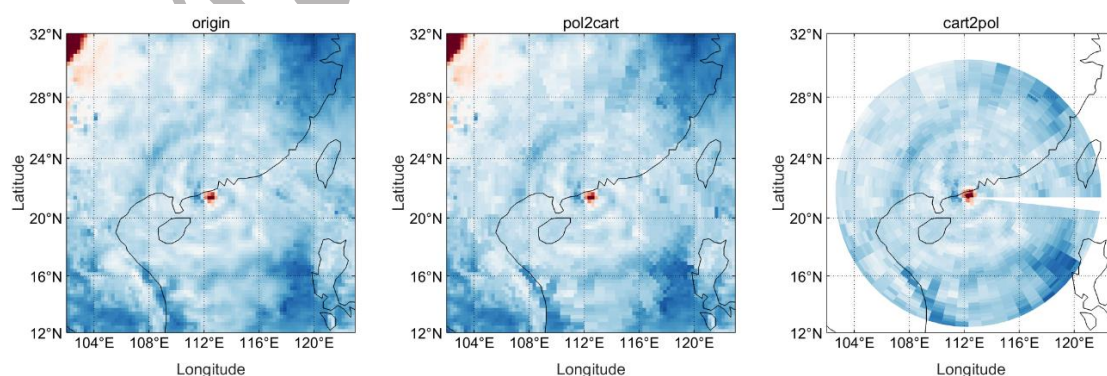


图 4.1 原始场（左）、从极坐标插值到直角坐标后（中）、直角坐标插值到极坐标后（右）

切向风和径向风

```
function [v_theta,v_r] = hurricane_uv(U,V,theta,r)
```


输入:

纬向、经向风场 $U, V(r, \theta)$ (单位: m/s)、

方位角 $\theta(\theta)$ (单位: 度)

径向距离 $r(r)$ (单位: 度或 km)

输出:

切向风、径向风 $v_\theta, v_r(r, \theta)$ (单位: m/s)

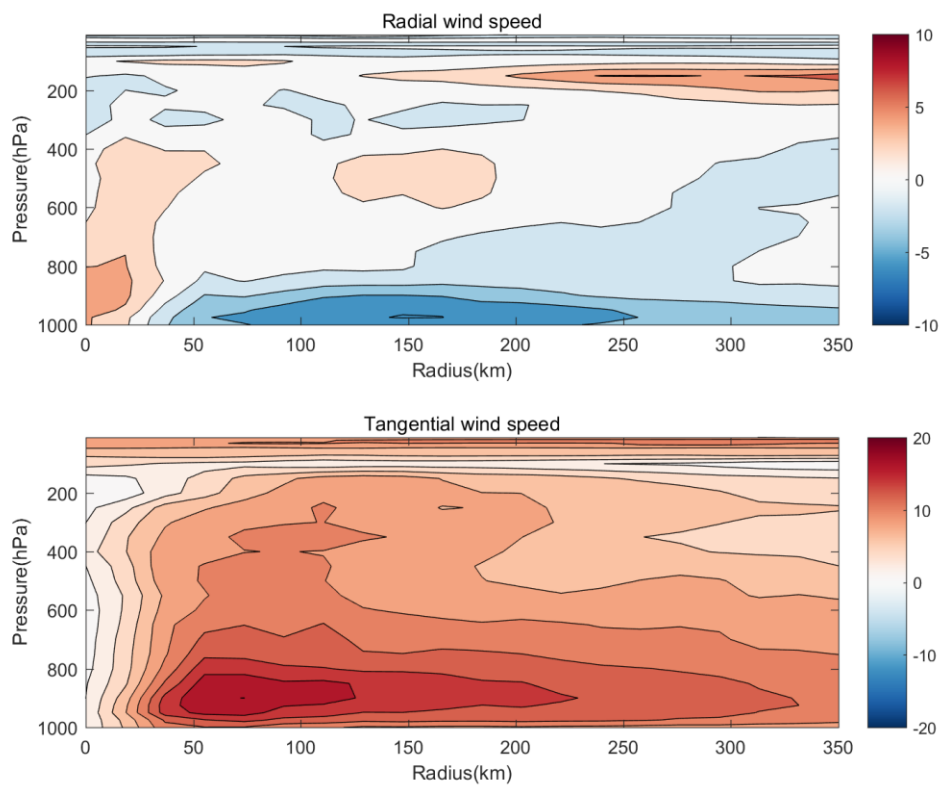


图 4.2 计算得到的径向风速(上)和切向风速(下)的分布

绝对角动量

```
function M = hurricane_absolute_angular_momentum  
(v_theta, v_r, r, ctr_lat)
```

输入:

切向风、径向风 $v_\theta, v_r(r, \theta)$ (单位: m/s)、

径向距离 $r(r)$ (单位: km)、

TC 中心所在纬度 ctr_lat

输出:

绝对角动量 $M(r, \theta)$