

MeteoLab

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 .....	5
2.1 基本常数 .....	5
2.2 基础物理量 .....	5
科氏参数 .....	5
经纬度网格距 .....	5
空气密度 .....	5
散度 .....	5
垂直相对涡度 .....	6
绝对涡度 .....	6
水平平流项 .....	6
2.3 差分/梯度/平滑 .....	7
水平梯度 .....	7
垂直梯度 .....	7
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 .....	8
2.4 动力相关量 .....	8
地转风 .....	8
Q 矢量 .....	8
正压模式下的位势涡度 .....	9
Ertel 位涡 .....	9
湿位涡 .....	9
2.5 热力相关量 .....	10
位温 .....	10
相当位温 .....	10
干绝热递减率 .....	10
湿绝热递减率 .....	11
非绝热加热率 .....	11
锋生函数 .....	11

---

2.6 水汽相关量.....	11
露点温度.....	11
抬升凝结温度.....	12
单层水汽通量/散度.....	12
饱和水汽压.....	12
饱和混合比.....	12
2.7 大气稳定性.....	13
静力稳定度.....	13
<b>四、高级功能.....</b>	<b>13</b>
1. Shuman-Shapiro 滤波.....	13
2. 涡度收支诊断.....	13
3. 台风诊断.....	13
二维直角坐标插值到二维极坐标系.....	13
三维直角坐标插值到柱坐标系.....	14

# MeteoLab 使用说明

## 一、MeteoLab 简介

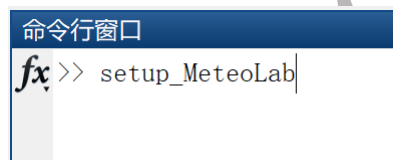
MeteoLab 是基于 Matlab 开发的，专门用于气象数据诊断和分析的工具箱，包含气象物理量诊断相关的基本函数，可快速计算出所需的气象物理量。可用于气象科研、数据可视化等领域。需要在 Matlab2016a 以上版本运行。

## 二、安装步骤

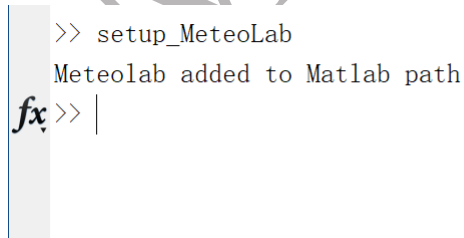
1. 需要先将当前文件夹设置为工具箱安装路径



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

#### 2.1 基本常数

#### 2.2 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)、

$\beta$ 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  $adv(n, m)$

---

## 2.3 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  $F1(n,m)$

### 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 F1=smth9(F,s)`

输入：

物理量  $F(n,m)$ 、

平滑系数  $s$

输出：

九点平滑后的数据  $F1(n,m)$

## 2.4 动力相关量

### 地转风

$$\begin{aligned}U_g &= -\frac{g}{f} \frac{\partial H}{\partial y} \\ V_g &= \frac{g}{f} \frac{\partial H}{\partial x}\end{aligned}$$

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

输入：

位势高度  $H(n,m)$  （单位：gpm）、

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

科氏参数  $f(n,m)$  （单位：/s）

输出：

地转风场  $Ug,Vg(n,m)$  （单位：m/s）

### Q 矢量

$$\begin{aligned}Q_x &= -\frac{R}{p} \left( \frac{\partial \overline{V_g}}{\partial x} \cdot \nabla T \right) \\ Q_y &= -\frac{R}{p} \left( \frac{\partial \overline{V_g}}{\partial y} \cdot \nabla T \right)\end{aligned}$$

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

输入：

温度  $T(n,m)$  （单位：K）、

地转风场  $Ug,Vg(n,m)$  （单位：m/s）、



气压(Pa) **pressure(l)** (单位: Pa)、  
纬向、经向网格距 **dx,dy(n,m)** (单位: m)

输出:

Q 矢量 **Q(n,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 **U,V(n,m)** (单位: m/s)、  
位势高度场 **H(n,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  $pv(p,n,m)$

## 2.5 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)

### 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))

## 2.6 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  $T_d$  (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  $T_d$  (单位: K)

输出:

抬升凝结温度  $T_{lcl}$  (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)

## 2.7 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