

用 C 语言开发的气象常用参数 和物理量计算函数库(三)

李社宏

(铜川市气象局 铜川·727000)

用有限元插值法求三角形内的实测风涡度
(scfwd __xy) dat [3] 结构中为三角形三个顶点的
经度、纬度、风向、风速,其中前三项为百进位角
度,风速为米/秒,涡度单位: $1\text{E}-5/\text{秒}$

```
double scfwd __xy (struct jwdf __type dat [3])
{
    float x1, x2, x3, y1, y2, y3;
    float u1, u2, u3, v1, v2, v3;
    double PI=3.14159265;
    double R=6.371229 * 1e6;
    double out;
    x1=0; y1=0;
    x2=(dat[1].jd-dat[0].jd) * PI * R * cos
        (((dat[1].wd+dat[0].wd)/2) * PI/
        180)/180;
    y2=(dat[1].wd-dat[0].wd) * PI * R/180;
    x3=(dat[2].jd-dat[0].jd) * PI * R * cos
        (((dat[2].wd+dat[0].wd)/2) * PI/
        180)/180;
    y3=(dat[2].wd-dat[0].wd) * PI * R/180;
    fsfj(dat[0].fd,dat[0].ff,&u1,&v1);
    fsfj(dat[1].fd,dat[1].ff,&u2,&v2);
    fsfj(dat[2].fd,dat[2].ff,&u3,&v3);
    out=((v1-v2) * (y2-y3)-(v2-v3) * (y1-y2)-
        (x1-x2) * (u2-u3)+(x2-x3) * (u1-u2))/
        ((x1-x2) * (y2-y3)-(x2-x3) * (y1-y2));
    return out * 1E5;
}
```

根据 T8,Td8,T5 计算 showalter 稳定度指数

showalter>0:大气层结稳定

showalter<0:大气层结不稳定

```
double showalter(double t8,double td8,double t5)
{
    double w,m1,m2,Qc,Q5,P5=500;
    double Etd8,Eout,ETa,T8,P8=850;
    double Ta,Pa;
    double Cpd=0.2403;
    double Cpv=0.445;
    double Rd=6.85578 * 0.01;
    double Rw=11.017874 * 0.01;
    double T0=273.16;
    double L0=597.40;
    double C=1.002;
    double C1=0.57;
    double out,step=10;
    T8=T0+t8;
    Etd8=E __ WATER(t8);
    w=(Rd/Rw) * Etd8/(P8-Etd8);
    m1=(Cpd * (1+Cpv * w/Cpd))/(Rd * (1+
    w/R(Rd/Rw)));
    Ta=Tc(P8,t8,td8)+T0;
    Pa=P8 * (pow(Ta/T8,m1));
    m2=(Cpd/Rd) * (1+C * w/Cpd);
    ETa=E __ WATER(Ta-T0);
    Qc=log((Pa-ETa)/pow(Ta,m2))-(0.622/
    Rd) * (((L0+C1 * (T0-Ta))/Ta) *
    (ETa/(Pa-ETa)));
    out=T8;
    Eout=E __ WATER(out-T0);
    Q5=log((P5-Eout)/pow(out,m2))-(0.622/
    Rd) *
```

```

(((L0 + C1 * (T0-out))/out) * (Eout/(P5-
Eout))),
while(fabs(Qc-Q5)>0.0001){
    if(Qc>Q5){
        out=out-step;
    }
    else{
        out=out+step;
        step=step/5;
        out=out-step;
    }
    Eout=E __ WATER(out-T0);
    Q5=log((P5-Eout)/pow(out,m2))-(0.622/
    Rd) * (((L0 + C1 * (T0-out))/out) *
    (Eout/(P5-Eout)));
}
return t5-(out-T0);
}

```

根据 P1,T1,FD1,FF1,

P2,T2,FD2,FF2 计算

richardson 乱流指数

$Ri \leq Ric$, 乱流发展 $Ri > Ric$, 乱流减弱 $Ric \sim 0.1$

double richardson (double pdn, double tdn, double
fddn, double ffdn, double pup, double tup, double
fdup, double ffup)

```

{
    double Rd=2.8704 * le6;
    double Cpd=0.2403;
    double A=2.38844 * le-8;
    float udn,vdn,uup,vup;
    double out;
    double pjp,pjT,cha __ p, cha __ t, cha __ u,
    cha __ v;
    fsfj(fdup,ffup,&uup,&vup);
    fsfj(fddn,ffdn,&udn,&vdn);
    pjp=(pup+pdn)/2;
    pjT=273.16+(tup+tdn)/2;
    cha __ p=pup-pdn;
    cha __ t=tup-tdn;
    cha __ u=uup-udn;
    cha __ v=vup-vdn;

```

```

    out=-le-4 * (Rd * cha __ p/pjp) * (cha __ t-
    (A * Rd * pjT/Cpd) * (cha __
    p/pjp));
    out=out/(cha __ u * cha __ u + cha __ v * cha
    __ v);
    return out;
}

```

根据区站号求出该站的经度(getjd)

```

float getjd(long int zh)
{
    FILE * fp;
    long int zhr;
    char * zm=" ";
    float jdr,wdr,zr;
    register int i;
    if(! (fp=fopen("da.dat","r"))){
        printf("da.dat can not open\n");
        exit(0);
    }
    while(! feof(fp)){
        fscanf(fp,"%ld",&zhr);
        for(i=0;i<8;i++) fscanf(fp,"%c",
        &zm[i]);
        fscanf(fp,"%f,%f,%f",&wdr,&jdr,
        &zr);
        if(zh==zhr) break;
    }
    fclose(fp);
    return(int)jdr+(jdr-(int)jdr)/0.60;
}

```

根据区站号求出该站的纬度(getwd)

```

float getwd(long int zh)
{
    FILE * fp;
    long int zhr;
    char * zm=" ";
    float jdr,wdr,zr;
    register int i;
    if(! (fp=fopen("da.dat","r"))){printf("da.
    dat can not open\n");exit(0);
    }
    while(! feof(fp)){ fscanf(fp,"%ld",&zhr); for

```

```

(i=0;i<8;i++) fscanf(fp,"%c",&zm[i]);
fscanf(fp,"%f,%f,%f,%f",&wdr,&jdr,&zr); if
(zh==zhr) break;
}
fclose(fp);
return(int)wdr+(wdr-(int)wdr)/0.60;
}

```

根据区站号求出该站名(getzm)

```

char * getzm(long int zh)
{
    FILE * fp;
    long int zhr;
    char * zm="";
    float jdr,wdr,zr;
    register int i;
    if(! (fp=fopen("da.dat","r"))){printf("da.dat
can not open\n");exit(0);
}
while(! feof(fp)){fscanf(fp,"%ld",&zhr);
    for(i=0;i<8;i++) fscanf(fp,"%c",
        &zm[i]); fscanf(fp,"%f,%f,%f",
        &wdr,&jdr,&zr);if(zh==zhr) break;
}
fclose(fp);
return zm;
}

```

根据区站号求出该站的海拔高度(getz)

```

float getz(long int zh)
{
    FILE * fp;
    long int zhr;
    char * zm="";
    float jdt,wdr,zr;
    registet int i;
    if(! (fp=fopen("da.dat","r"))){printf("da.dat
can not open\n");exit(0);
}
while(! feof(fp)){fscanf(fp,"%ld",&zhr);
    for(i=0;i<8;i++) fscanf(fp,"%c",
        &zm[i]); fscanf(fp,"%f,%f,%f",
        &wdr,&jdr,&zr);if(zh==zhr) break;
}
}

```

```

fclose(fp);
return zr;
}

```

用正方形网格法求

高空地转风 u 分量(dzfug __zfwg)

N __H,S __H: 地势米 ,d: 网格距

(公里) wd: 百进位角度

```

double dzfug __zfwg(double N __H,double S __H,
double d,double wd)
{
    double out,f,w=7.2921152*1e-5;
    f=2*w*sin(wd*3.14159265/180);
    out=-9.80665*(N __H-S __H)/(2*f*d*
1000);
    return out;
}

```

用正方形网格法求高空地转风

v 分量(dzfvg __zfwg)

E __H,W __H: 地势米 d: 网格距(公里) wd: 百进位角度

```

double dzfvg __zfwg(double E __H,double W __
H,double d,double wd)
{
    double out,f,w=7.2921152*1e-5;
    f=2*w*sin(wd*3.14159265/180);
    out=9.80665*(E __H-W __H)/(2*f*d*
1000);
    return out;
}

```

用经纬网格法求高空

地转风 u 分量(dzfug __jwwg)

N __H,S __H: 地势米 wgj: 网格距(百进位角度)

wd: 纬度(百进位角度)

```

double dzfug __jwwg(double N __H,double S
__H,double wgj,double wd)
{
    double out,f,d,w=7.2921152*1e-5;
    d=wgj*111.137;
    f=2*w*sin(wd*3.14159265/180);
    out=-9.80665*(N __H-S __H)/(2*f*d*
1000);
    return out;
}

```

用经纬网格法求高空地转风

v 分量(dzfvg __ jwwg)

E __ H, W __ H: 位势米 wgj: 网格距(百进位角度)

wd: 纬度(百进位角度)

double dzfvg __ jwwg(double E __ H, double W __ H, double wgj, double wd)

```
{
    double out, w=7.2921152 * 1e-5;
    double R=6.371229 * 1e6;
    double PI=3.14159265;
    double B;
    B=wgj;
    out=441.0 * (E __ H-W __ H)/(PI * B * R *
w * sin(wd * PI/180) * cos(wd * PI/180));
    return out;
}
```

用正方形网格法求高空地

转风涡度(dzfwd __ zfwg)

H0: 中心网格点高度(位势米)

H1--H4: 四周网格点高度(位势米)

d: 网格距(公里) wd: 纬度(百进位角度)

涡度单位: 1e-5/秒

double dzfwd __ zfwg(double H1, double H2, double H3, double H4, double H0, double d, double wd)

```
{
    double Hp, out;
    d=d * 1000;
    Hp=(H1+H2+H3+H4)/4;
    out=19.6 * (Hp-H0)/(7.29 * 1e-5 * d * d *
sin(wd * 3.14159265/180));
    return out * 1e5;
}
```

根据任意三点的要素

值用有限元插值法求三角形内任

一点的要素值(用于散点资料网

格化和缺测记录的插补)

double wgh __ yxy(double jd0, double wd0, struct jwdf __ type dat[3])

```
{
    double x1, x2, x3, y1, y2, y3, z1, z2, z3;
    double PI=3.14159265;
```

double R=6.371229 * 1e6;

double out;

x1=(dat[0].jd-jd0) * PI * R * cos(((dat[0].wd+wd0)/2) * PI/180)/180;

x2=(dat[1].jd-jd0) * PI * R * cos(((dat[1].wd+wd0)/2) * PI/180)/180;

x3=(dat[2].jd-jd0) * PI * R * cos(((dat[2].wd+wd0)/2) * PI/180)/180;

y1=(dat[0].wd-wd0) * PI * R/180;

y2=(dat[1].wd-wd0) * PI * R/180;

y3=(dat[2].wd-wd0) * PI * R/180;

z1=dat[0].fd;

z2=dat[1].fd;

z3=dat[2].fd;

out=z1 * x2 * y3+x1 * y2 * z3+y1 * z2 * x3-y1 * x2 * z3-x1 * z2 * y3-z1 * y2 * x3;

out=out/((x1-x2) * (y2-y3)-(x2-x3) * (y1-y2));

return out;

}

根据 T8, Td8, T7, Td7, T5 计算 K 指数(K)

double K(double T8, double Td8, double T7, double Td7, double T5)

```
{
    return (T8-T5)+Td8-(T7-Td7);
}
```

根据该层及低一层的散度

DK, DK __ 1 和两层之间的气压差计算

下一层到该层的垂直速度增量(czsdl)

DK __ 1, DK: 1e-5/秒

P __ cha: 百帕

垂直速度增量单位: 1e-3 百帕/秒

double czsdl(double DK __ L, double DK, double P __ cha)

```
{
    return(1e-5 * (DK __ 1+DK) * P __ cha/2.0)
* 1e3;
```

根据两站点的经纬度 wd1, jd1,

wd2, jd2 求两站点间的距离

double D __ jw(double wd1, double jd1, double wd2, double jd2)

```
{
    (下转 36 页)
```

波的传播也有干扰作用。

雷雨天气下的冰雹也会产生次声,周期约在 20—40 s 间,声压约一微巴左右,而且它产生的次声波主周期与冰雹直径大小有关,冰雹大时,其所产生的次声波的主周期也比较长。

风暴(台风、热带气旋、雷雨前、龙卷风、寒潮、飚线等都可引起大风)也能辐射次声波,其幅度峰值在 0.2—1.5 微巴左右,周期在 12—60 s 之间,信号的持续时间可达 6—7 h,有些风暴云如积雨云、台风来临前的飞云,都会辐射出次声波。

锋面抵达前夕,次声自动记录系统可记录到波动曲线,当然这儿也夹杂有风速曲线,这就需要进行分析对比才行。

研究探讨有关灾害性天气产生的次声波的特性及其与恶劣天气的内在关系,有助于人们及早地发现和预测出灾害性天气,达到趋利避害,减少灾损之目的。

次声为何能够探测恶劣天气?次声是我们人耳感觉不出来的,但能用次声接受系统来发觉它,一般可用次声传声器,将次声波的信号的声能转换为可供放大传输或记录的电信号;在接收仪器中,采取一系列抗干扰技术措施,提高信噪比;将接收到信号经模拟转换器转为数学信息,直接输入计算机进行各种计算,识别、分析;从而知道次声波

的方位角度和次声源的中心位置,次声信号的特性,显示在屏幕显示器上。经过识别分析,就可判断出是什么恶劣天气的次声波。

为什么次声波能传播得这样遥远呢?原来声波的吸收效应和频率有关。频率高,衰减吸收大;频率低,吸收就小,衰减就越少。因此次声在大气介质中传播时衰减特别小。此外,声波传播快慢与温度有关,温度高时,空气密度小,声波传播得快,温度低时传播得慢(如温度为 0℃ 时声速为 $331.4 \text{ m} \cdot \text{s}^{-1}$, 温度为 30℃ 时声速为 $349.6 \text{ m} \cdot \text{s}^{-1}$, 快 $18.2 \text{ m} \cdot \text{s}^{-1}$)。在空间大气层中的上层和下层都有一个温度极小值的气层如对流层顶,次声波沿着温度极小值所形成的通道向前传播,不至于逃逸到平流层去,这也是次声波能够象“马拉松”似的传播得遥远的原因之一。鉴于次声波身怀上述两种“绝技”所以能传到天涯海角。人们只要装上“顺风耳”——次声接收系统,就能及时地捕捉到它,并应用于探测恶劣天气如台风、龙卷风、晴空湍流、风暴、雷雨、冰雹、锋面等。

开展对次声科学的研究,探索灾害性天气所产生的次声波的特性及其机制,这对于做好灾害性天气的预测和监测预报工作无疑会有很大益处。

(谢在永)

(上接 31 页)

```
double x,y,out;double PI=3.14159265;double R=6.371229*1e6;
x=(jd-z-jd1)*PI*R*cos(((wd1+wd2)/2)
```

```
*PI/180)/180;y=(wd2-wd1)*PI*R/180;
out=hypot(x,y);
return out/1000;
```