ภาคผนวก

1. โปรแกรมการหาค่า TEC จากแบบจำลอง IRI-95

```
C Test program for IRIT13 program. You can also get the F peak
C density and heights with the COMMON block
C       common/block1/hmf2,xnmf2,hmf1
C IRIS13 contains other COMMON blocks that may be helpful.
c Changes:   9/16/98 jf now input in IRIT13
      logical       jf(17)
      data          jf/17*.true./
c  select various options and choices for IRI-95
        open(unit=iuout,file='output.iri',status='old',form='formatted')
        jf(2)=.false.         ! no temperatures
        jf(3)=.false.         ! no ion composition
Ctest   jf(4)=.false.         ! Gulyaeva-B0
        jf(5)=.false.         ! URSI-88 for foF2
C       jf(12)=.false.        ! konsol output to file (unit=12)
        print*, 'mmdd'
        read(*,*) mmdd
        iy=1998
        hour=6.5
        jmag=0
        xla=18.8
        xlo=99.0
        print*,'hour(LT), TEC/m-2'
        hstp=0.5
        do 1 i=1,48
        hour=hour+hstp
        if (hour.gt.24) then
              hour=0.5
              mmdd=mmdd+1
        endif
        call irit13(xla,xlo,jmag,jf,iy,mmdd,hour,100.,1000.,tec,tecb,tect)
        write(6,*) hour,tec
        write(iuout,*) tec
2       continue
1       continue
        stop
        end
```

2. ฟังก์ชัน acfpacf.m

```
function [ACF,se,PACF]=acf(data)
if nargout<3
  error('[ACF,SE,PACF]=acf(data)');
end;
N=length(data);
datamean=nanmean(data);
lagtime=ceil(N/4);
in=data-datamean;
bottom=nansum(in.^2);
% to find standard error limit
limit1=2/sqrt(N);
limit2=-limit1;
for k=1:lagtime
  gam=0;
  for t=(k+1):N
    if (isnan(data(t))~=1)&(isnan(data(t-k))~=1)
       gamma=(data(t)-datamean)*(data(t-k)-datamean);
       gam=gam+gamma;
    end
  end
  gammak(k)=gam;
end
rk=(gammak./(bottom))';
ACF=rk;
% find standard error of acf:  SE(rk)
se(1)=1/sqrt(N);
for k=2:length(rk),
```

```
  sum=0;
  for j=1:(k-1),
    sum=sum+rk(j)^2;
  end
  se(k)=sqrt((1+2*sum)/N);
end
% plot ACF
figure(6);clf;set(gcf,'color',[1 1 1]);
subplot(3,1,1); plot(1:lagtime,rk,'r',1:lagtime,limit1,'b',1:lagtime,limit2,'b');
axis([0 length(rk) -1 1]); xlabel('lag time');ylabel('ACF'); legend('ACF')
% plot SE(k)
subplot(3,1,2); plot(se,'r'); axis([0 length(rk) 0 1]);
xlabel('lag time'); ylabel('Standard ERROR'); legend('SE(ACF(k))');
% find PACF
[nr,nc]=size(rk);
if (nr~=1) & (nc~=1)
  error('ARP must be vector');
end;
if nr~=1,
  rk=rk.';  nc=nr;
end;
N=nc;
PHI=zeros(N,N); PHI(1,1) = rk(1);
for K=1:N-1,
   G=(rk(K+1)-PHI(K,1:K)*rk(K:-1:1)')/(1-PHI(K,1:K)*rk(1:K)');
   PHI(K+1,1:K)=PHI(K,1:K)-G*PHI(K,K:-1:1);
   PHI(K+1,K+1)=G;
end;
PACF = diag(PHI);
% plot PACF
subplot(3,1,3); plot(1:lagtime,PACF,'r',1:lagtime,limit1,'b',1:lagtime,limit2,'b');
axis([0 length(rk) -1 1]); xlabel('lag time'); ylabel('PACF'); legend('PACF')
```

3. โปรแกรมย่อย arxsari.m

```
% Model AR(1)xSARI(1,1)
km0=1;
while km0,
  km0=menu('Input coefficients',... %title of menu
    'Constant',...%km0=1
    'AR1',...%km0=2
    'SAR1',...%km0=3
    'Process');%km0=4
   if km0==1,
    clc; disp('A default constant value is zero');
    const=input('Please input a constant value:     ');
  elseif km0==2,
    clc; disp('Please input the AR1 coefficients');
    disp('Example:  >>AR(1)=      0.2146');
    ar=input('AR(1)=      ');
  elseif km0==3,
    clc; disp('Please input the SAR1 coefficients');
    disp('Example:  >>SAR(1)=      0.6124]');
    sar=input('SAR(1)=      ');
  elseif km0==4,
    if ~exist('const','var'),
      const=0;
    end
    if exist('ar','var')&exist('sar','var')
      %process
      day_num=input('How many numbers for forecasting:      ');
      day_tot=length(avgtec)+day_num;
      tmp(1,1:day_num)=nan;
      avgtec=[avgtec tmp];x=avgtec;
      clear tmp;
      y(1:48,1)=nan;e(1:48,1)=0;
      y(49)=x(1)+const;
```

```
for t=50:96
  if isnan(x(t-1)),
    x(t-1)=y(t-1);
  end
  if isnan(x(t-48)),
    x(t-48)=y(t-48);
  end
  if isnan(x(t-49)),
    x(t-49)=y(t-49);
  end
  %predict
  y(t)=const+ar*x(t-1)+(1+sar)*x(t-48)-(ar+ar*sar)*x(t-49);
  if ~isnan(avgtec(t))
    e(t)=avgtec(t)-y(t);
  else
    e(t)=0;
  end
end
y(97)=const+ar*x(t-1)+(1+sar)*x(t-48)-(ar+ar*sar)*x(t-49)-sar*x(1);
if ~isnan(avgtec(97)),
  e(97)=avgtec(t)-y(97);
else
  e(97)=0;
end
for t=98:day_tot
  if isnan(x(t-1)),
    x(t-1)=y(t-1);
  end
  if isnan(x(t-48)),
    x(t-48)=y(t-48);
  end
  if isnan(x(t-49)),
    x(t-49)=y(t-49);
  end
  if isnan(x(t-96)),
    x(t-96)=y(t-96);
  end
  if isnan(x(t-97)),
    x(t-97)=y(t-97);
  end
  %predict
  y(t)=const+ar*x(t-1)+(1+sar)*x(t-48)-(ar+ar*sar)*x(t-49)-sar*x(t-96)+ar*sar*x(t-97);
  if ~isnan(avgtec(t))
    e(t)=avgtec(t)-y(t);
  else
    e(t)=0;
  end
end
a=find(~e);
e(a)=nan;
figure(11); clf; set(gcf,'color',[1 1 1]);
subplot(2,1,1);plot(1:length(avgtec),avgtec,'.',1:length(y),y,'r');
axis([0 day_tot 0 max(nanmax(avgtec),nanmax(y))+10]);
ylabel('TEC  [x10^1^6 electrons/m^2]');  xlabel('Time(UT)  [30 minutes]');
legend('Observed TEC','Predicted TEC');
subplot(2,1,2);plot(e); legend('error');
ylabel('error  [x10^1^6 electrons/m^2]');  xlabel('Time(UT)  [30 minutes]');
axis([0 day_tot nanmin(e)-5 nanmax(e)+5]);
else
  if ~exist('ar','var');
    disp('Please input the AR1 coefficient');
  end
  if ~exist('sma','var');
    disp('Please input the SMA1 coefficient');
  end
end
```

```
    km0=0;
  end
end
% find correlation coefficient
c=corr(y,avgtec);
clear km0 a b x1 x2 c;
```

4. **โปรแกรมย่อย arxsarima.m**

```
% Model AR(1)xSARIMA(1,1)
km1=1;
while km1,
  km1=menu('Input coefficients',... %title of menu
    'Constant',...%km1=1
    'AR1',...%km2=2
    'SAR1',...%km=3
    'SMA1',...%km=4
    'Process');%km=5
  if km1==1,
    clc; disp('A default constant value is zero');
    const=input('Please input a constant value:    ');
  elseif km1==2,
    clc; disp('Please input the AR1 coefficients');
    disp('Example:  >>AR(1)=     0.2146');
    ar=input('AR(1)=     ');
  elseif km1==3,
    clc; disp('Please input the SAR1 coefficients');
    disp('Example:  >>SAR(1)=     0.3624');
    sar=input('SAR(1)=     ');
  elseif km1==4,
    clc; disp('Please input the SMA1 coefficients');
    disp('Example:  >>SMA(1)=     0.6124]');
    sma=input('SMA(1)=     ');
  elseif km1==5,
    if ~exist('const','var'),
      const=0;
    end
    if exist('ar','var')&exist('sar','var')&exist('sma','var')
      %process
      day_num=input('How many numbers for forecasting:     ');
      day_tot=length(avgtec)+day_num; tmp(1,1:day_num)=nan;
      avgtec=[avgtec tmp];x=avgtec;
      clear tmp;
      y(1:48,1)=nan;e(1:48,1)=0;
      y(49)=x(1)+const;
      for t=50:97,
        if isnan(x(t-1)),
          x(t-1)=y(t-1);
        end
        if isnan(x(t-48)),
          x(t-48)=y(t-48);
        end
        if isnan(x(t-49)),
          x(t-49)=y(t-48);
        end
        %predict
        y(t)=const+(sar+1)*x(t-48)+ar*x(t-1)-(sar*ar+ar)*x(t-49)-sma*e(t-48);
        if ~isnan(avgtec(t)),
          e(t)=avgtec(t)-y(t);
        else
          e(t)=0;
        end
      end
      y(97)=const+(sar+1)*x(t-48)+ar*x(t-1)-(sar*ar+ar)*x(t-49)-sar*x(t-96)-sma*e(t-48);
      if ~isnan(avgtec(97)),
        e(97)=avgtec(t)-y(97);
      else
```

```
      e(97)=0;
    end
    for t=98:day_tot%length(avgtec)+day_num,
      if isnan(x(t-1)),
        x(t-1)=y(t-1);
      end
      if isnan(x(t-48)),
        x(t-48)=y(t-48);
      end
      if isnan(x(t-49)),
        x(t-49)=y(t-49);
      end
      if isnan(x(t-96)),
        x(t-96)=y(t-96);
      end
      if isnan(x(t-97)),
        x(t-97)=y(t-97);
      end
      y(t)=const+(sar+1)*x(t-48)+ar*x(t-1)-(sar*ar+ar)*x(t-49)-sar*x(t-96)+sar*ar*x(t-97)-sma*e(t-48);
      if ~isnan(avgtec(t))
        e(t)=avgtec(t)-y(t);
      else
        e(t)=0;
      end
    end
    a=find(~e);
    e(a)=nan;
    figure(10);clf;
    set(gcf,'color',[1 1 1]); subplot(2,1,1);plot(1:length(avgtec),avgtec,'.',1:length(y),y,'r');
    axis([0 day_tot 0 max(nanmax(avgtec),nanmax(y))+10]); ylabel('TEC  [x10^1^6 electrons/m^2]');
    xlabel('Time(UT)  [30 minutes]'); legend('Observed TEC','Predicted TEC');
    subplot(2,1,2);plot(e); ylabel('error  [x10^1^6 electrons/m^2]'); xlabel('Time(UT)  [30 minutes]'); legend('error');
    axis([0 day_tot nanmin(e)-5 nanmax(e)+5]);
  else
    if ~exist('ar','var');
      disp('Please input the AR1 coefficient');
    end
    if ~exist('sar','var');
      disp('Please input the SAR1 coefficient');
      if ~exist('sma','var');
        disp('Please input the SMA1 coefficient');
      end
    end
  end
  km1=0;
 end
end
c=corr(avgtec,y);
clear km2 a b x1 x2 c;
```

5. โปรแกรมย่อย arxsima.m

```
% Model AR(1)xSIMA(1,1)
km0=1;
while km0,
 km0=menu('Input coefficients',... %title of menu
  'Constant',...%km0=1
  'AR1',...%km0=2
  'SMA1',...%km0=3
  'Process');%km0=4
 if km0==1,
  clc;  disp('A default constant value is zero');
  const=input('Please input a constant value:     ');
 elseif km0==2,
  clc;  disp('Please input the AR1 coefficients');
  disp('Example:  >>AR(1)=     0.2146');
  ar=input('AR(1)=     ');
```

```
elseif km0==3,
  clc;  disp('Please input the SMA1 coefficients');
  disp('Example:  >>SMA(1)=     0.6124]');
  sma=input('SMA(1)=     ');
elseif km0==4,
  if ~exist('const','var'),
    const=0;
  end
  if exist('ar','var')&exist('sma','var')
    %process
    day_num=input('How many numbers for forecasting:     ');
    day_tot=length(avgtec)+day_num;
    tmp(1,1:day_num)=nan;
    avgtec=[avgtec tmp];
    x=avgtec; clear tmp;
    y(1:48,1)=nan;e(1:48,1)=0;
    y(49)=x(1)+const;
    for t=50:day_tot
      if isnan(x(t-1))
        x(t-1)=y(t-1);
      end
      if isnan(x(t-48)),
        x(t-48)=y(t-48);
      end
      if isnan(x(t-49)),
        x(t-49)=y(t-49);
      end
      %predict
      y(t)=const+ar*x(t-1)+x(t-48)-ar*x(t-49)-sma*e(t-48);
      if ~isnan(avgtec(t))
        e(t)=avgtec(t)-y(t);
      else
        e(t)=0;
      end
    end
    a=find(~e);
    e(a)=nan;
    figure(9);clf;
    set(gcf,'color',[1 1 1]);
    subplot(2,1,1);plot(1:length(avgtec),avgtec,'.',1:length(y),y,'r');
    axis([0 day_tot 0 max(nanmax(avgtec),nanmax(y))+10]);
    ylabel('TEC  [x10^1^6 electrons/m^2]'); xlabel('Time(UT)  [30 minutes]'); legend('Observed TEC','Predicted TEC');
    subplot(2,1,2);plot(e); legend('error'); ylabel('error  [x10^1^6 electrons/m^2]'); xlabel('Time(UT)  [30 minutes]');
    axis([0 day_tot nanmin(e)-5 nanmax(e)+5]);
  else
    if ~exist('ar','var');
      disp('Please input the AR1 coefficient');
    end
    if ~exist('sma','var');
      disp('Please input the SMA1 coefficient');
    end
  end
  km0=0;
 end
end
% calculate correlation coefficient
c=corr(avgtec,y);
clear km0 a b x1 x2 c;
```

6. ฟังก์ชัน avg30mag.m

```
function [Hav,Dav,Zav,Fav]=avg30mag(H,D,Z,F,MJD,HH,MM)
% to average magnetic data every 30 minutes
mjd0=MJD(1);hh0=HH(1);mm0=MM(1);
a=find(MM<30); ID(a)=0;
b=find(MM>=30); ID(b)=1;
id=ID(1);
```

```
Hav=[];Dav=[];Zav=[];Fav=[];
wh=0;wd=0;wz=0;wf=0;sum=0;
for p=1:length(MJD)
  if ID(p)==id
    if isnan(H(p))~=1,
      wh=wh+H(p); wd=wd+D(p); wz=wz+Z(p); wf=wf+F(p);
      sum=sum+1;
    end
  else
    if sum~=0
      Hav=[Hav wh/30]; Dav=[Dav wd/30]; Zav=[Zav wz/30]; Fav=[Fav wf/30];
    else
      Hav=[Hav nan]; Dav=[Dav nan]; Zav=[Zav nan]; Fav=[Fav nan];
    end
    id=ID(p);
    wh=0;wd=0;wz=0;wf=0;sum=0;
    if isnan(H(p))~=1,
      wh=wh+H(p); wd=wd+D(p); wz=wz+Z(p); wf=wf+F(p);
      sum=sum+1;
    end
  end
end
if sum~=0,
  Hav=[Hav wh/sum]; Dav=[Dav wd/sum]; Zav=[Zav wz/sum]; Fav=[Fav wf/sum];
else
  Hav=[Hav nan]; Dav=[Dav nan]; Zav=[Zav nan]; Fav=[Fav nan];
end
```

7. **ฟังก์ชัน avg30tec.m**

```
function [AVG,rms]=avg30tec(TEC,NUM,RMS,MJD,HH,MM)
% to average TEC data every 30 minutes
a=find(isnan(TEC)==1);
TEC(a)=0;NUM(a)=0;RMS(a)=0;
mjd0=MJD(1);hh0=HH(1);mm0=MM(1);
a=find(MM<30);
ID(a)=0;
b=find(MM>=30);
ID(b)=1;
id=ID(1);
AVG=[];rms=[];
wtec=0;sum=0;wrms=0;
for p=1:length(MJD)
  if ID(p)==id
    wtec=wtec+TEC(p)*NUM(p);
    wrms=wrms+RMS(p)*NUM(p);
    sum=sum+NUM(p);
  else
    if sum~=0
      AVG=[AVG wtec/sum];
      rms=[rms wrms/sum];
    else
      AVG=[AVG nan];
      rms=[rms nan];
    end
    id=ID(p);
    wtec=0;sum=0;wrms=0;
    wtec=wtec+TEC(p)*NUM(p);
    wrms=wrms+RMS(p)*NUM(p);
    sum=sum+NUM(p);
  end
end
if sum~=0
  AVG=[AVG wtec/sum];
  rms=[rms wrms/sum];
else
  AVG=[AVG nan];
```

```
  RMS=[RMS nan];
end
```

8. ฟังก์ชัน avgmag_min.m

```
function [H,D,Z,mjd,hh_mag,mm_mag]=avgmag_min(path,dr,sdate,h,d,z)
% to average magnetic data into 1 minutes value
a=fix(sdate/1000000);
if a>=90 %year 19xx
  d_in=fix(sdate/100)+19000000;
else
  d_in=fix(sdate/100)+20000000;
end;
[slope,sh,sm,ss]=readhdr(path,dr,sdate);
mjd=date2mjd(d_in);
sumh=0;sumd=0;sumz=0;
firstdata=60-ss;
for k=1:firstdata,
  sumh=sumh+h(k);  sumd=sumd+d(k);  sumz=sumz+z(k);
end
H(1)=sumh/firstdata; D(1)=sumd/firstdata; Z(1)=sumz/firstdata;
sumh=0;sumd=0;sumz=0;
% calculate length of complete value data
lengthdata=fix(length(h)/60);
% calculate length of residual value data
res_data=mod(length(h)-firstdata,60);
% id=data index
id=2;
for m=2:lengthdata%+1,
  for k=1:60,
    sumh=sumh+h(id); sumd=sumd+d(id); sumz=sumz+z(id);
    id=id+1;
  end
  H(m)=sumh/60; D(m)=sumd/60; Z(m)=sumz/60;
  sumh=0;sumd=0;sumz=0;
end
if res_data>0,
  for k=1:res_data,
    sumh=sumh+h(length(h)-k+1); sumd=sumd+d(length(h)-k+1); sumz=sumz+z(length(h)-k+1);
  end
  H(length(H)+1)=sumh/res_data; D(length(D)+1)=sumd/res_data; Z(length(Z)+1)=sumz/res_data;
end
% generate time for magnetic data
mm_mag(1)=sm;hh_mag(1)=sh;mjd_mag(1)=mjd;
for k=2:length(H),
  mm_mag(k)=mm_mag(k-1)+1;
  hh_mag(k)=hh_mag(k-1);
  mjd(k)=mjd(k-1);
  if mm_mag(k)>59,
    mm_mag(k)=0;
    hh_mag(k)=hh_mag(k)+1;
    mjd(k)=mjd(k-1);
    if hh_mag(k)>23
      hh_mag(k)=0;
      mjd(k)=mjd(k)+1;
    end
  end
end
```

9. โปรแกรมย่อย callacf.m

```
% callacf sub-program
clc;
kacf=1;
while kacf,
  kacf=menu('Select following button',... %title of menu
    'ACF-PACF for averaged TEC',...%kacf=1
    'ACF-PACF for difference TEC',...%kacf=2
```

```
    'ACF-PACF for seasonal diff TEC',...%kacf=3
    'Predict',...%kacf=4
    'Q-test',...%kacf=5
    'OK');%kacf=6
  if kacf==1,
    [tecacf,se,tecpacf]=acf(avgtec);
  elseif kacf==2,
    d=input('Please specify the number of difference:\n');
    z=sdiff1(avgtec,d); %z is the avgtec(t)-avgtec(t-d)
    [zacf,zse,zpacf]=acf(z);
  elseif kacf==3,
    Z=sdiff1(avgtec,48);
    [Zacf,Zse,Zpacf]=acf(Z);
  elseif kacf==4,
    callmodel;
  elseif kacf==5,
    clf;
    disp('Q-test');
    disp('Please wait.....');
    e_test=e(49:length(avgtec));
    [acf_e,se_e,pacf_e]=acf(e_test);
    Q=length(e)*nansum(acf_e.^2);
    if (exist('ar','var')&exist('sar','var'))|(exist('ar','var')&exist('sma','var')),
      parameters=2;
    else
      parameters=3;
    end
    df=length(e_test)-parameters;
    Q_x=chi2inv(0.95,df);
    disp('------------------------------');
    disp('Q-value     Critical Q-test');
    disp([Q Q_x]);
    disp('------------------------------');
    disp('CONCLUSION:');
    if Q<Q_x,
      disp('The model is fit for TEC data');
      disp('---------------------------------');
    else
      disp('The model is not fit for TEC data');
      disp('Please find the new model');
      disp('---------------------------------');
    end
  elseif kacf==6,
    kacf=0;
  end;
end;
clear kacf;
```

10. โปรแกรมย่อย callapc.m

```
% application sub-menu
% Reference:
% 1) Ajayi, G.O., Feng, S., Radicella, S.M. and Reddy,B.M.
%    "Handbook on Radiopropagation related to satellite communications
%    in tropical and subtropical countries"
%    URSI, 1996
% 2) Roddy, D. "Satellite Communication". Singapore: McGraw-Hill, 1996.
% 3) Emerson, D., "Elliptical Polarization in the Ionosphere"
%    Available: http://www.tuc.nrao.edu/~demerson/ionosphere/ionopol.html
clc;
if ~exist('avgtec','var')|~exist('Hav','var'),
  disp('---------- Warning ----------')
  disp('Please load averaged TEC data and')
  disp('averaged Magnetic databefore using this menu.');
  disp('-----------------------------------------------');
else
  [nr_tec,nc_tec]=size(avgtec);
```

```
[nr_mag,nc_mag]=size(Hav);
if nr_tec~=nr_mag,
  Hav=Hav';Dav=Dav';Zav=Zav';Fav=Fav';
end
if nc_tec~=nc_mag,
  a=min(nc_tec,nc_mag);
  avgtec=avgtec(nr_tec,1:a);
  Hav=Hav(nr_tec,1:a);
  Dav=Dav(nr_tec,1:a);
  Zav=Zav(nr_tec,1:a);
  Fav=Fav(nr_tec,1:a);
end
kapc=1;
while kapc,
  kapc=menu('Select following button',... %title of menu
    'Faraday Rotation',...%kapc=1
    'Polarization Loss (PL)',...%kapc=2
    'Cross Polarization discrimination (XPD)',...%kapc=3
    'end');%kapc=4
  if kapc==1,
    %Faraday Rotation (phi)
    Bav=nanmean([Hav;Dav;Zav])*10^(-9);
    c=2.36*10^4*Bav;
    f=(1:0.1:10)*10^8;
    if exist('phi','var'),
      clear phi;
    end
    for k=1:length(f),
      phi(k,:)=c.*avgtec*10^16/f(k)^2;   %radian unit
      phi_deg=phi*360/pi;
    end
    figure(16);clf
    mesh(phi_deg);
    ylabel('Frequency  [x10^8 Hz]'); xlabel('Time   [30 minutes]'); zlabel('Faraday Rotation   [degree]');
  elseif kapc==2,
    %Polarization Loss (PL) in dB unit
    if ~exist('phi','var'),
      Bav=nanmean([Hav;Dav;Zav])*10^(-9);
      c=2.36*10^4*Bav;
      f=(0.1:10)*10^9;
      for k=1:length(f),
        phi(k,:)=c.*avgtec*10^16/f(k)^2;   %radian unit
      end
    else
      PL=20*log(cos(phi));
      if ~isreal(PL),
        PL=real(PL);
      end
      figure(17);clf
      mesh(PL);
      ylabel('Frequency  [x10^8 Hz]'); xlabel('Time   [30 minutes]'); zlabel('Polarization loss [dB]');
    end
  elseif kapc==3,
    %Cross Polarization discrimination (XPD) in dB unit
    if ~exist('phi','var'),
      Bav=nanmean([Hav;Dav;Zav])*10^(-9);
      c=2.36*10^4*Bav;
      f=(0.1:10)*10^9;
      for k=1:length(f),
        phi(k,:)=c.*avgtec*10^16/f(k)^2;   %radian unit
      end
    else
      XPD=-20*log(cot(phi));
      if ~isreal(XPD),
        XPD=abs(XPD);
      end
```

```
        figure(18);clf
        mesh(XPD);
        ylabel('Frequency  [x10^8 Hz]'); xlabel('Time    [30 minutes]'); zlabel('XPD  [dB]');
      end
    elseif kapc==4,
      kapc=0;
    end;
  end;
end;
clear kapc;
```

11. โปรแกรมย่อย calliri.m

```
% sub-program for IRI data
kiri=1;path='c:/iri/'; clc;
disp('-------------------------------------------')
disp('        The IRI data is 30 minutes value')
disp('-------------------------------------------')
while kiri,
  kiri=menu('IRI sub-program',... %title
    'Input start date',...%k=1
    'Input end date',...%k=2
    'Input path',...%k=3
    'GET IRI DATA',...%k=4
    'End');%k=5
  if kiri==1,
    clc;
    sdate=input('please input start date:     yyyymmdd\n');
  elseif kiri==2,
    clc;
    edate=input('please input end date:       yyyymmdd\n');
  elseif kiri==3,
    clc;
    disp('Default path name is c:/iri/');
    path=input('please input the directory name for IRI data:  c:/iri/\n','s');
    if ~exist(path,'dir'),
      disp('please check the IRI directory again');
    end
  elseif kiri==4
    if exist('sdate','var')&exist('edate','var')&exist(path,'dir')
      iri=[];
      dr=num2str(fix(sdate/100));
      fp=strcat(path,dr,'/',num2str(sdate),'.iri');
      tec=readiri(fp);
      iri=[iri tec];
      while sdate~=edate,
        [dr,sdate]=checktec(dr,sdate);
        fp=strcat(path,dr,'/',num2str(sdate),'.iri');
        teciri=readiri(fp);
        iri=[iri teciri];
      end
      clear fp dr sdate edate path tec;
      figure(3);clf;set(gcf,'color',[1 1 1]); plot(iri,'b.');grid
      title('Predicted TEC by International Reference Ionosphere IRI-95 model');
      ylabel('TEC [x10^1^6 electrons/m^2 or TECU'); xlabel('Time [30 minutes]');
      kiri=0; %end
    else
      clc;  disp('Please select the option again');
    end
  elseif kiri==5
    kiri=0;
  end;
end;
clear kiri;
```

12. โปรแกรมย่อย callmag.m

```
% subfunction to manage Magnetic data
```

```
kmag=1; clc; path='c:/magne/';
while kmag,
  kmag=menu('MAG sub-program',... %title
    'Input start date',...%k=1
    'Input end date',...%k=2
    'Input path',...%k=3
    'GET MAGNETIC DATA',...%k=4
    'Averaged Magnetic data',...%k=5
    'End');%k=6
  if kmag==1,
    clc; d_in=input('please input start date:     yyyymmdd\n');
  elseif kmag==2,
    clc; d_out=input('please input end date:      yyyymmdd\n');
  elseif kmag==3,
    clc;  disp('Default path is c:/magne/');
    path=input('please input directory name for Magnetic data:   c:/magne/\n','s');
  elseif kmag==4
    if exist('d_in','var')&exist('d_out','var')
      %create default directory
      if ~exist(path,'dir')
        path='c:/magne/';
      end
      % check directory
      if ~exist(path,'dir')
        disp('Please Check the c:/magne/ directory and run again');
      end
      % get data
      H=[];D=[];Z=[];MJD=[];HH=[];MM=[];
      sdate=mod(d_in,fix(d_in/1000000)*1000000)*100;
      enddate=mod(d_out,fix(d_out/1000000)*1000000)*100;
      dr=num2str(fix(d_in/100));
      fp=strcat(path,dr,'/',num2str(sdate,8),'.mat')
      while sdate~=enddate,
        while exist(fp,'file')~=2
          [path,dr,sdate]=checkmag(path,dr,sdate,'.mat');
          fp=strcat(path,dr,'/',num2str(sdate,8),'.mat');
        end
        %process
        load (fp);
        [h,d,z,mjd,hh,mm]=avgmag_min(path,dr,sdate,h,d,z);
        MJD=[MJD mjd];HH=[HH hh];MM=[MM mm];
        H=[H h];D=[D d];Z=[Z z];
        %next data file
        [path,dr,sdate]=checkmag(path,dr,sdate,'.mat');
        fp=strcat(path,dr,'/',num2str(sdate,8),'.mat');
      end
      %process the last data file
      if exist(fp,'file')==2
        load (fp);
        [h,d,z,mjd,hh,mm]=avgmag_min(path,dr,sdate,h,d,z);
        MJD=[MJD mjd];HH=[HH hh];MM=[MM mm];
        H=[H h];D=[D d];Z=[Z z];
      else
        disp('-------------------------------------------')
        disp(' SORRY the end date has no data')
      end
      %fill data lost;
      disp('---------- process data lost ----------')
      [h,d,z,mjd_mag,hh_mag,mm_mag]=missmag(H,D,Z,MJD,HH,MM);
      % calculate total magnetic field
      f=sqrt(h.^2+d.^2+z.^2);
      clear H D Z MJD HH MM sdate d_in d_out path dr fp enddate hh mm mjd
      figure(2);clf; set(gcf,'color',[1 1 1]);
      subplot(4,1,1);plot(h,'b'); legend('Horizontal component'); axis([0 length(h) nanmin(h)-10 nanmax(h)+10]);
      title('Geomagnetic data');ylabel('H component [nT]');xlabel('Time [minute]');
      subplot(4,1,2);plot(d,'r'); legend('Declination component'); axis([0 length(d) nanmin(d)-10 nanmax(d)+10]);
```

```
      ylabel('D component [nT]');xlabel('Time [minute]');
      subplot(4,1,3);plot(z,'m'); legend('Vertical component');
      axis([0 length(z) nanmin(z)-10 nanmax(z)+10]); ylabel('Z component [nT]');xlabel('Time [minute]');
      subplot(4,1,4);plot(f,'k'); legend('Total Field');
      axis([0 length(f) nanmin(f)-10 nanmax(f)+10]); ylabel('Total Magnetic Field [nT]');xlabel('Time [minute]');
  else
      clc; disp('Please select the option again');
    end
  elseif kmag==5,
    %get averaged magnetic data
    if exist('h','var'),
      [Hav,Dav,Zav,Fav]=avg30mag(h,d,z,f,mjd_mag,hh_mag,mm_mag);
      figure(5);clf; set(gcf,'color',[1 1 1]);
      subplot(4,1,1);plot(Hav,'b'); legend('Horizontal component');
      axis([0 length(Hav) nanmin(Hav)-10 nanmax(Hav)+10]);
      title('Averaged Geomagnetic data');ylabel('H component [nT]');xlabel('Time [30 minutes]');
      subplot(4,1,2);plot(Dav,'r'); legend('Declination component');
      axis([0 length(Dav) nanmin(Dav)-10 nanmax(Dav)+10]); ylabel('D component [nT]');xlabel('Time [30 minutes]');
      subplot(4,1,3);plot(Zav,'m'); legend('Vertical component');
      axis([0 length(Zav) nanmin(Zav)-10 nanmax(Zav)+10]); ylabel('Z component [nT]');xlabel('Time [30 minutes]');
      subplot(4,1,4);plot(Fav,'k'); legend('Total Field');
      axis([0 length(Fav) nanmin(Fav)-10 nanmax(Fav)+10]);
      ylabel('Total Magnetic Field [nT]');xlabel('Time [30 minutes]');
    else
      disp('Please get the magnetic data before average');
    end
  elseif kmag==6,
    kmag=0;
  end;
end;
clear kmag;
```

13. โปรแกรมย่อย callmet.m

```
% meteorological data
clc;
km=1;
while km,
  km=menu('Meteorological Data',... %title of menu
    'Read ascii data',...%km=1
    'get TEC data',...%km=2
    'correlation',...%km=3
    'scatter diagram',...%km=4
    'regression',...%km=5
    'end');%km=6
  if km==1,
    [fname,pname]=uigetfile('c:\tec_met\*.dat','please select input file name');
    fp=strcat(pname,fname);
    fid=fopen(fp,'r');
    Temp=[];Rh=[];Dir=[];Vel=[];Press=[];
    while feof(fid)~=1,
      temp=fscanf(fid,'%6f',1);
      Temp=[Temp;temp];
      rh=fscanf(fid,'%3d',1);
      Rh=[Rh;rh];
      dir=fscanf(fid,'%4d',1);
      Dir=[Dir;dir];
      vel=fscanf(fid,'%4d',1);
      Vel=[Vel;vel];
      press=fscanf(fid,'%6f',1);
      Press=[Press;press];
    end
    fclose(fid);
    figure(7);clf;set(gcf,'color',[1 1 1]);
    subplot(3,1,1);plot(Temp);
    legend('Temperature');
    ylabel('Temperature   [^OC]');
```

```
  xlabel('Time    [30 minutes]');
  subplot(3,1,2);plot(Rh);
  legend('Relative Humidity');
  ylabel('Relative Humidity [%]');
  xlabel('Time    [30 minutes]');
  subplot(3,1,3);plot(Press);
  legend('Pressure');
  ylabel('Pressure   [hPa]');
  xlabel('Time    [30 minutes]');
elseif km==2,
  %get TEC data
  [fname,pname]=uigetfile('c:\tec\*.mat','please select input file name');
  fp=strcat(pname,fname);
  load(fp);
elseif km==3,
  % correlation
  disp('Correlation options');
  disp('Please select case:');
  disp('      1. TEC & Temperature');
  disp('      2. TEC & Relative Humidity');
  disp('      3. TEC & wind velocity');
  disp('      4. TEC & Pressure');
  s=input('case:    ');
  switch s,
  case 1,
    x1=avgtec;y1=Temp;
    c=corr(x1,y1);
  case 2,
    x1=avgtec;y1=Rh;
    c=corr(x1,y1);
  case 3,
    x1=avgtec;y1=Vel;
    c=corr(x1,y1);
  case 4,
    x1=avgtec;y1=Press;
    c=corr(x1,t1);
  end
elseif km==4,%scatter diagram
  figure(12);clf;set(gcf,'color',[1 1 1]);
  subplot(2,2,1);plot(Temp,avgtec,'x');ylabel('TEC  [x10^1^6 electron/m^2]');xlabel('Temperature [^oC]');
  subplot(2,2,2);plot(Rh,avgtec,'x');ylabel('TEC  [x10^1^6 electron/m^2]');xlabel('Relative Humidity [%]');
  subplot(2,2,3);plot(Vel,avgtec,'x');ylabel('TEC  [x10^1^6 electron/m^2]');xlabel('Wind Velocity [km/hr]');
  subplot(2,2,4);plot(Press,avgtec,'x');ylabel('TEC  [x10^1^6 electron/m^2]');xlabel('Pressure level [hPa]');
elseif km==5,
  % regression
  disp('--------------------------');
  disp('Regression method');
  disp('Please select regression equation');
  disp('      1. TEC=b0+b1*Temp+b2*Rh');
  disp('      2. TEC=b0+b1*ARMA+b2*Temp+b3*Rh');
  s1=input('');
  switch s1
  case 1,
    disp('--------------------------');
    disp('TEC=b0+b1*Temp+b2*Rh');
    disp('--------------------------');
    x1=avgtec;a1=Temp;a2=Rh;
    a=length(x1);b=length(a1);
    if a>b,
      for i=length(a1)+1:length(a1),
        a1(i)=nan;a2(i)=nan;
      end
    end
    a=find(isnan(x1));b=find(isnan(a1));
    a1(a)=nan;a2(a)=nan;x1(b)=nan;
    x2=x1(~isnan(x1));
```

```
    a3=a1(~isnan(a1));a4=a2(~isnan(a2));
    z=[ones(length(x2),1) a3 a4];
    [x2,z]=chksize(x2,z);
    [b,bint,r,rint,stats]=regress(x2,z);
    pre=[ones(length(avgtec),1) Temp Rh]*b;
    %find correlation
    c=corr(pre,avgtec);
    figure(8);clf
    set(gcf,'color',[1 1 1]);
    plot(1:length(avgtec),avgtec,'.',1:length(avgtec),pre,'r'); legend('Observed TEC','Predicted TEC by Regression');
    ylabel('TEC     [x10^1^6 electrons/m^2]'); xlabel('Time     [30 minutes]');
  case 2,
    disp('--------------------------');
    disp('TEC=b0+b1*ARMA+b2*Temp+b3*Rh');
    disp('--------------------------');
    x1=avgtec;a0=y';a1=Temp;a2=Rh;
    a=length(x1);b=length(a1);
    if a>b,
      for i=length(a1)+1:length(a1),
        a1(i)=nan;a2(i)=nan;
      end
    end
    a=find(isnan(x1));b=find(isnan(a0));
    a0(a)=nan;a1(a)=nan;a2(a)=nan;
    x1(b)=nan;a1(b)=nan;a2(b)=nan;
    x2=x1(~isnan(x1));
    a3=a0(~isnan(a0));a4=a1(~isnan(a1));
    a5=a2(~isnan(a2));
    [x2,z]=chksize(x2,z);
    z=[ones(length(x2),1) a3 a4 a5];
    [b,bint,r,rint,stats]=regress(x2',z);
    pre=[ones(length(avgtec),1) y' Temp Rh]*b;
    %find correlation
    c=corr(pre,avgtec);
    figure(8);clf
    set(gcf,'color',[1 1 1]);
    plot(1:length(avgtec),avgtec,'.',1:length(avgtec),pre,'r'); legend('Observed TEC','Predicted TEC by Regression');
    ylabel('TEC     [x10^1^6 electrons/m^2]'); xlabel('Time     [30 minutes]');
  end
 elseif km==6,
   km=0;
 end;
end;
clear fid fp a b c pname fname x1 x2 y1 y2 s km;
```

14. โปรแกรมย่อย callmodel.m

```
% callmodel sub-program
clc;
disp('------------ SUGGESTIONS ----------------');
disp('Please save the averaged TEC data before estimating the coefficients');
disp('-----------------------------------------');
avg_diff=sdiff1(avgtec,48);
diff_mean=nanmean(avg_diff);
diff_std=nanstd(avg_diff);
if diff_mean<diff_std,
  disp('This model no need the constant value');
  disp('The constant value will be set to zero');
  disp('-----------------------------------------');
  const=0;
else
  disp('This model need the constant value');
  disp('Please input the constant value from the menu options');
  disp('-----------------------------------------');
end
km=1;
while km,
```

```
km=menu('Select model',... %title of menu
  'Save',...%km=1
  'AR(1)xSARI(1,1)',...%km=2
  'AR(1)xSIMA(1,1)',...%km=3
  'AR(1)xSARIMA(1,1,1)',...%km=4
  'OK');%km=5
if km==1,
  [fname1,pname1]=uiputfile('c:\*.dat','save file name');
  filename1=strcat(pname1,fname1,'.dat');
  fid1=fopen(filename1,'w');
  for t=1:length(avgtec),
    fprintf(fid1,'%12f\n',avgtec(t));
  end
  fclose(fid1);
elseif km==2,
  arxsari;
elseif km==3,
  arxsima;
elseif km==4,
  arxsarima;
elseif km==5,
  km=0;
end;
end;
```

15. โปรแกรมย่อย callplot.m

```
% sub-program callplot.m
% MAIN program
clc;
kplot=1;
while kplot,
  kplot=menu('Select following button',... %title of menu
    'Plot TEC data',...%kplot=1
    'Plot Magnetic data',...%kplot=2
    'Plot IRI data',...%kplot=3
    'Plot TEC vs. Predicted by ARMA',...%kplot=4
    'Plot TEC vs. IRI-95 model',...%kplot=5
    'Plot TEC vs. Predicted by Magnetic data',... %kplot=6
    'end');%kplot=7
  if kplot==1,
    if exist('avgtec','var'),
      figure(1);clf; set(gcf,'color',[1 1 1]);
      subplot(2,1,1);plot(TEC,'b.'); axis([0 length(TEC) 0 nanmax(TEC)+10]); legend('Measured TEC');
      title('Measured TEC Values'); ylabel('TEC   [x10^1^6 electrons/m^2]');  xlabel('Time(UT)   [minute]');
      subplot(2,1,2);plot(avgtec,'.');  axis([0 length(avgtec) 0 nanmax(avgtec)+10]); title('Averaged TEC Values');
      ylabel('TEC   [x10^1^6 electrons/m^2]'); xlabel('Time(UT)   [30 minutes]'); legend('Averaged TEC');
    else
      disp('Please get the TEC data before ploting');
      kplot=0;
    end

  elseif kplot==2,
    if exist('h','var'),
      figure(2);clf
      set(gcf,'color',[1 1 1]);
      subplot(4,1,1);plot(h,'b.'); axis([0 length(h) 0 nanmax(h)+10]);
      title('Geomagnetic data');ylabel('Horizontal component [nT]');xlabel('Time [minute]'); legend('H');
      subplot(4,1,2);plot(d,'r.'); axis([0 length(d) 0 nanmax(d)+10]);
      ylabel('Declination component [nT]');xlabel('Time(UT) [minute]'); legend('D');
      subplot(4,1,3);plot(z,'k.'); axis([0 length(z) 0 nanmax(z)]);
      ylabel('Vertical component [nT]');xlabel('Time(UT) [minute]'); legend('Z');
      subplot(4,1,4);plot(f,'m.'); axis([0 length(f) 0 nanmax(f)]);
      ylabel('Total Magnetic Field [nT]');xlabel('Time(UT) [minute]'); legend('F');
    else
      disp('Please get the Magnetic data before ploting');
      kplot=0;
```

```
    end
  elseif kplot==3,
    if exist('iri','var'),
      figure(3);clf
      set(gcf,'color',[1 1 1]);
      plot(1:length(iri),iri,'ro',1:length(iri),iri,'b');
      axis([0 length(iri) 0 max(iri)+10]);
      legend('IRI-95');
    else
      disp('Please get the IRI data before ploting');
      kplot=0;
    end
  elseif kplot==4,
    if exist('y','var'),
      if exist('avgtec','var'),
        figure(4); set(gcf,'color',[1 1 1]);
        subplot(2,1,1); plot(1:length(avgtec),avgtec,'b.',1:length(y),y,'r');
        axis([0 length(avgtec) 0 max(nanmax(avgtec),nanmax(y))+10]);
        ylabel('TEC  [x10^1^6 electrons/m^2]'); xlabel('Time(UT)     [30 minutes]');
        legend('Observed TEC','Predicted TEC');
      else
        disp('Please get the TEC data before ploting');
        kplot=0;
      end
    else
      disp('Please get the predicted TEC data before ploting');
      kplot=0;
    end
  elseif kplot==5,
    if exist('iri','var'),
      if exist('avgtec','var'),
        figure(4); set(gcf,'color',[1 1 1]);
        subplot(2,1,2); plot(1:length(avgtec),avgtec,'b.',1:length(iri),iri,'r');
        axis([0 length(avgtec) 0 max(nanmax(avgtec),nanmax(iri))+10]);
        ylabel('TEC     [x10^1^6 electrons/m^2]'); xlabel('Time(UT)      [30 minutes]');
        legend('Observed TEC','IRI-95 model');
      else
        disp('Please get the TEC data before ploting');
        kplot=0;
      end
    else
      disp('Please get the IRI-95 model before ploting');
      kplot=0;
    end
  elseif kplot==6,
    % TEC and magne

    kplot=0;
  elseif kplot==7,
    kplot=0;
  end;
end;
clear kplot;
```

16. **โปรแกรมย่อย callregress.m**

```
% callregress.m sub-program
clc;
kr=1;
if (mjd_mag(1)>813)
  if (mjd_mag(1)<1179),
    year=1998;
    day_s=mjd_mag(1)-813;
    day_e=mjd_mag(length(mjd_mag))-813;
  elseif (mjd_mag(1)>1178)&(mjd_mag(1)<1544),
    year=1999;
    day_s=mjd_mag(1)-1178;
```

```
    day_e=mjd_mag(length(mjd_mag))-1178;
  elseif (mjd_mag(1)>1543),
    year=2000;
    day_s=mjd_mag(1)-1543;
    day_e=mjd_mag(length(mjd_mag))-1543;
  end
end
while kr,
  kr=menu('Select following button',... %title of menu
    'Input start day for prediction',...%kr=1
    'Input end day for prediction',...%kr=2
    'Process',...%kr=3
    'end');%kr=4
  if kr==1,
    sday=input('Input start day:      ');
          sday=48*(sday-1)+1;
  elseif kr==2,
    eday=input('Input end day:      ');
    eday=48*eday;
    if (length(Hav)<eday);
      eday=length(Hav);
    end
    if (length(avgtec)<eday),
      eday=length(avgtec);
    end
  elseif kr==3,
    if ~exist('sday','var')
      sday=1;
    end
    if ~exist('eday','var'),
      eday=ceil(length(h)/48);
    end
    if ~exist('Hav','var'),
      [Hav,Dav,Zav,Fav]=avg30mag(h,d,z,f,mjd_mag,hh_mag,mm_mag);
    end
    hav=Hav(sday:eday);dav=Dav(sday:eday);
    zav=Zav(sday:eday);fav=Fav(sday:eday);
    a=find(isnan(hav));
    tec=avgtec(sday:eday);
    tec(a)=nan;
    b=find(isnan(tec));
    hav(b)=nan;dav(b)=nan;zav(b)=nan;fav(b)=nan;
    hav=hav(~isnan(hav));
    dav=dav(~isnan(dav));
    zav=zav(~isnan(zav));
    fav=fav(~isnan(fav));
    y=tec(~isnan(tec));
    %scatter diagram
    figure(14);clf;
    set(gcf,'color',[1 1 1]);
    subplot(2,2,1);plot(y,hav,'x');xlabel('TEC [x10^1^6 electrons/m^2]');ylabel('H-component [nT]');
    subplot(2,2,2);plot(y,dav,'x');xlabel('TEC [x10^1^6 electrons/m^2]');ylabel('D-component [nT]');
    subplot(2,2,3);plot(y,zav,'x');xlabel('TEC [x10^1^6 electrons/m^2]');ylabel('Z-component [nT]');
    subplot(2,2,4);plot(y,fav,'x');xlabel('TEC [x10^1^6 electrons/m^2]');ylabel('Main Field [nT]');
    figure(15);
    predict0;
  elseif kr==4,
    kr=0;
  end;
end;
```

17. **โปรแกรมย่อย calltec.m**

```
% subfunction to manage TEC data
ktec=1;clc;
path='c:/tec/';
while ktec,
```

```
ktec=menu('TEC sub-program',... %title
  'Input start date',...%k=1
  'Input end date',...%k=2
  'Input path',...%k=3
  'GET TEC DATA',...%k=4
  'Averaged TEC data',...%k=5
  'End');%k=6
if ktec==1,
  clc;
  sdate=input('please input start date:     yyyymmdd\n');
elseif ktec==2,
  clc;
  edate=input('please input end date:       yyyymmdd\n');
elseif ktec==3,
  clc;
  disp('The default path is c:/tec/');
  path=input('please input directory name for TEC data:  c:/tec/\n','s');
  if ~exist(path,'dir'),
    clc;
    disp('Please check the TEC directory');
  end
elseif ktec==4
  if exist('sdate','var')&exist('edate','var')&exist(path,'dir')
    % get data
    TEC=[];MJD=[];HH=[];MM=[];NUM=[];RMS=[];
    dr=num2str(fix(sdate/100));
    mjdstart=date2mjd(sdate);
    fp=strcat(path,dr,'/',num2str(sdate,8),'.dat');
    while sdate<edate
      while exist(fp,'file')~=2
        [dr,sdate]=checktec(dr,sdate);
        fp=strcat(path,dr,'/',num2str(sdate,8),'.dat');
      end
      %process
      MJD=[MJD mjd];HH=[HH hh];MM=[MM mm];
      NUM=[NUM num];TEC=[TEC tec];RMS=[RMS rms];
      %next data file
      [dr,sdate]=checktec(dr,sdate);
      fp=strcat(path,dr,'/',num2str(sdate,8),'.dat');
    end
    %process last 2 data files
    [mjd,hh,mm,num,tec,rms]=readtec(fp);
    MJD=[MJD mjd];HH=[HH hh];MM=[MM mm];
    NUM=[NUM num];TEC=[TEC tec];RMS=[RMS rms];
    [dr,sdate]=checktec(dr,sdate);
    fp=strcat(path,dr,'/',num2str(sdate,8),'.dat');
    while exist(fp,'file')~=2
      [dr,sdate]=checktec(dr,sdate);
      fp=strcat(path,dr,'/',num2str(sdate,8),'.dat');
    end
    [mjd,hh,mm,num,tec,rms]=readtec(fp);
    MJD=[MJD mjd];HH=[HH hh];MM=[MM mm];
    NUM=[NUM num];TEC=[TEC tec];RMS=[RMS rms];
    %fill data lost;
    disp('---------- process data lost ----------')
    [tec,num,rms,mjd,hh,mm]=misstec(TEC,NUM,RMS,MJD,HH,MM);
    % select desired data
    a=find(mjdstart==mjd);
    mjdend=date2mjd(edate);
    b=find(mjdend==mjd);
    TEC=tec(a(1):b(length(b)));
    NUM=num(a(1):b(length(b)));
    RMS=rms(a(1):b(length(b)));
    mjd_tec=mjd(a(1):b(length(b)));
    hh_tec=hh(a(1):b(length(b)));
    mm_tec=mm(a(1):b(length(b)));
```

```
      clear fp tec num rms mjd hh mm HH MM MJD a b path dr sdate edate edate_real mjdend mjdstart
      figure(1);clf; set(gcf,'color',[1 1 1]);
      subplot(2,1,1);plot(TEC,'b.');grid; legend('observed TEC')
      title('Ionospheric Total Electron Content'); ylabel('TEC [x 10^1^6 electrons/m^2'); xlabel('Time [minute]');
    else
      clc;
      disp('Please select the option again');
    end
  elseif ktec==5,
    % average tec data every 30 minutes
    if exist('TEC','var'),
      [avgtec,rms]=avg30tec(TEC,NUM,RMS,mjd_tec,hh_tec,mm_tec);
      figure(1);set(gcf,'color',[1 1 1]);
      subplot(2,1,2);plot(avgtec,'.');grid; ylabel('Average TEC [x10^1^6 electrons/m^2');  xlabel('Time [30 minutes]');
      legend('Averaged TEC');
    else
      disp('Please read the TEC data before average data');
    end
  elseif ktec==6,
    ktec=0;
  end;
end;
clear ktec;
```

18. ฟังก์ชัน checkmag.m

```
function [path,dr,edate]=checkmag(path,dr,sdate,ext1)
 % to check the day of month
% cday==1:  end of month
% for magnetic data
smon=fix(sdate/10000);%start month
year=fix(smon/100);
index=smon-fix(smon/100)*100;
%check over month
if (index==1)|(index==3)|(index==5)|(index==7)|(index==8)|(index==10)|(index==12)
  smon=fix(sdate/100);
  if index<12
    if mod(smon,100)<31
      edate=sdate+1;
      fp=num2str(edate,8);
      mat1=strcat(path,dr,'/',fp,ext1);
      if exist(mat1,'file')~=2, %have no file yymmdd0x
        edate=fix(sdate/100)*100+100;
      end
    else
      dr=str2num(dr)+1;
      edate=mod(dr,10000)*10000+100;
      dr=num2str(dr);
      index=index+1;
    end
  else
    if mod(smon,100)<31
      edate=sdate+1;
      fp=num2str(edate,8);
      mat1=strcat(path,dr,'/',fp,ext1);
      if exist(mat1,'file')~=2, %have no file yymmdd0x
        edate=fix(sdate/100)*100+100;
      end
    else
      dr=str2num(dr)+89; %index=1
      edate=mod(dr,10000)*10000+100;
      dr=num2str(dr);
    end
  end;
else
  if (index==4)|(index==6)|(index==9)|(index==11)
    smon=fix(sdate/100);
```

```
  if mod(smon,100)<30
    edate=sdate+1;
    fp=num2str(edate,8);
    mat1=strcat(path,dr,'/',fp,ext1);
    if exist(mat1,'file')~=2, %have no file yymmdd0x
      edate=fix(sdate/100)*100+100;
    end
  else
    dr=str2num(dr)+1;
    edate=mod(dr,10000)*10000+100;
    dr=num2str(dr);
    index=index+1;
  end
 else
  if index==2
    if mod(year,4)~=0,%year~=2000,
      smon=fix(sdate/100);
      if mod(smon,100)<29
        edate=sdate+1;
        fp=num2str(edate,8);
        mat1=strcat(path,dr,'/',fp,ext1);
        if exist(mat1,'file')~=2, %have no file yymmdd0x
          edate=fix(sdate/100)*100+100;
        end
      else
        dr=str2num(dr)+1;
        edate=mod(dr,10000)*10000+100;
        dr=num2str(dr);
        index=index+1;
      end
    else
      smon=fix(sdate/100);
      if mod(smon,100)<30
        edate=sdate+1;
        fp=num2str(edate,8);
        mat1=strcat(path,dr,'/',fp,ext1);
        if exist(mat1,'file')~=2, %have no file yymmdd0x
          edate=fix(sdate/100)*100+100;
        end
      else
        dr=str2num(dr)+1;
        edate=mod(dr,10000)*10000+100;
        dr=num2str(dr);
        index=index+1;
      end
    end
  end
 end
end
```

19. ฟังก์ชัน checktec.m

```
function [dr,edate]=checktec(dr,sdate);
% for TEC data
% to check the day of month
% to increment day of file
% cday==1:  end of month
smon=fix(sdate/100);%smon=start month
year=fix(smon/100);
index=smon-fix(smon/100)*100;
%check over month
switch index
case {1 3 5 7 8 10}
  if (sdate-smon*100)<31
    edate=sdate+1;
  else
    dr=str2num(dr)+1;
```

```
    edate=dr*100+1;
    dr=num2str(dr);
    index=index+1;
  end
case 12
  if (sdate-smon*100)<31
    edate=sdate+1;
  else
    dr=str2num(dr)+89; %index=1
    edate=dr*100+1;
    dr=num2str(dr);
  end
case {4 6 9 11}
  if (sdate-smon*100)<30
    edate=sdate+1;
  else
    dr=str2num(dr)+1;
    edate=dr*100+1;
    dr=num2str(dr);
    index=index+1;
  end
case 2
  if mod(year,4)~=0 %year~=2000
    if (sdate-smon*100)<29
      edate=sdate+1;
    else
      dr=str2num(dr)+1;
      edate=dr*100+1;
      dr=num2str(dr);
      index=index+1;
    end
  else %year 2000
    if (sdate-smon*100)<30
      edate=sdate+1;
    else
      dr=str2num(dr)+1;
      edate=dr*100+1;
      dr=num2str(dr);
      index=index+1;
    end
  end
end
```

20. ฟังก์ชัน corr.m

```
function c=corr(x,y);
a=find(isnan(x));b=find(isnan(y));
y(a)=nan;x(b)=nan;
x=x(~isnan(x));y=y(~isnan(y));
[nrx,ncx]=size(x);
[nry,ncy]=size(y);
if nrx~=nry,
  x=x';
end
c=corrcoef(x,y);
c=c(2);
disp(['correlation coefficient is: ' num2str(c)]);
```

21. ฟังก์ชัน date2mjd.m

```
function [mjd]=date2mjd(sdate)
% to find mjd
syear=fix(sdate/10000);%start year
smon=fix(sdate/100)-syear*100;%start month
sday=sdate-syear*10000-smon*100;%start day
%check year
switch syear
case 1996
```

```
  mjd=82;
case 1997
  mjd=448;
case 1998
  mjd=813;
case 1999
  mjd=1178;
case 2000
  mjd=1543;
case 2001
  mjd=1909;
end
```

22. โปรแกรมหลัก main.m

```
% MAIN program
clc;
k=1;
while k,
  k=menu('Select following button',... %title of menu
    'Read binary TEC data',...%k=1
    'Read binary MAG data',...%k=2
    'Read IRI data',...%k=3
    'Read Meteorological data',...%k=4
    'Plot data',...%k=5
    'ARMA method',...%k=6
    'Regression',...%k=7
    'Applications',...%k=8
    'end');%k=9
  if k==1,
    calltec;
  elseif k==2,
    callmag;
  elseif k==3,
    calliri;
  elseif k==4,
    callmet;
  elseif k==5,
    callplot;
  elseif k==6,
    if ~exist('avgtec','var'),
      disp('Please load the TEC data before use this option');
    else
      callacf;
    end
  elseif k==7,
    callregress;
  elseif k==8,
    callapc;
  elseif k==9,
    k=0;
  end;
end;
clear k;
```

23. ฟังก์ชัน missmag.m

```
function [H,D,Z,mjd_mag,hh_mag,mm_mag]=missmag(h,d,z,mjd,hh,mm)
% to fill the missing value of magnetic data
hh_mag(1)=hh(1);
mm_mag(1)=mm(1);
mjd_mag(1)=mjd(1);
H(1)=h(1);
D(1)=d(1);
Z(1)=z(1);
id=1;id1=2;
%find real number of data,
hh1=hh(length(h));mm1=mm(length(h));mjd1=mjd(length(h));
```

```
hh0=hh(1);mm0=mm(1);mjd0=mjd(1);
if mm0>mm1,mm1=mm1+60;hh1=hh1-1;end
if hh0>hh, hh1=hh1+24;mjd1=mjd1-1;end
cm=mm1-mm0;ch=hh1-hh0;cmjd=mjd1-mjd0;
na=cmjd*24*60+ch*60+cm;mm2=mm0;
for k=1:na-1,
  id=id+1;
  mm_mag(id)=mm_mag(id-1)+1;
  hh_mag(id)=hh_mag(id-1);
  mjd_mag(id)=mjd_mag(id-1);
  if mm_mag(id)>59,
    mm_mag(id)=0;
    hh_mag(id)=hh_mag(id)+1;
    if hh_mag(id)>23,
      hh_mag(id)=0;
      mjd_mag(id)=mjd_mag(id)+1;
    end
  end
  if (mjd_mag(id)==mjd(id1))&(mm_mag(id)==mm(id1))&(hh_mag(id)==hh(id1)),
    H(id)=h(id1);
    D(id)=d(id1);
    Z(id)=z(id1);
    id1=id1+1;
  else
    H(id)=nan;
    D(id)=nan;
    Z(id)=nan;
  end
end
```

24. ฟังก์ชัน misstec.m

```
function [TEC,NUM,RMS,mjd_tec,hh_tec,mm_tec]=misstec(tec,num,rms,mjd,hh,mm)
% to fill the missing value of TEC data
hh_tec(1)=hh(1);
mm_tec(1)=mm(1);
mjd_tec(1)=mjd(1);
TEC(1)=tec(1);
NUM(1)=num(1);
RMS(1)=rms(1);
id=1;id1=2;
%find real number of data,
hh1=hh(length(tec));mm1=mm(length(tec));mjd1=mjd(length(tec));
hh0=hh(1);mm0=mm(1);mjd0=mjd(1);
if mm0>mm1,mm1=mm1+60;hh1=hh1-1;end
if hh0>hh, hh1=hh1+24;mjd1=mjd1-1;end
cm=mm1-mm0;ch=hh1-hh0;cmjd=mjd1-mjd0;
na=cmjd*24*60+ch*60+cm;mm2=mm0;
for k=1:na-1,
  id=id+1;
  mm_tec(id)=mm_tec(id-1)+1;
  hh_tec(id)=hh_tec(id-1);
  mjd_tec(id)=mjd_tec(id-1);
  if mm_tec(id)>59,
    mm_tec(id)=0;
    hh_tec(id)=hh_tec(id)+1;
    if hh_tec(id)>23,
      hh_tec(id)=0;
      mjd_tec(id)=mjd_tec(id)+1;
    end
  end
  if (mjd_tec(id)==mjd(id1))& (mm_tec(id)==mm(id1))&(hh_tec(id)==hh(id1)),
    TEC(id)=tec(id1);
    NUM(id)=num(id1);
    RMS(id)=rms(id1);
    id1=id1+1;
  else
```

```
    TEC(id)=nan;
    NUM(id)=0;
    RMS(id)=0;
  end
end
```

25. โปรแกรมย่อย predict0.m

```
% predict menu
k=1;
while k,
  k=menu('select independent variable',...
    'magnetic intensity H-axis',...
    'magnetic intensity D-axis',...
    'magnetic intensity Z-axis',...
    'main field',...
    'end');
  if k==1,
    x=hav;x_r=Hav(sday:eday);clc;
    predict1;
  elseif k==2,
    x=dav;x_r=Dav(sday:eday);clc;
    predict1;
  elseif k==3,
    x=zav;x_r=Zav(sday:eday);clc;
    predict1;
  elseif k==4,
    x=fav;x_r=Fav(sday:eday);clc;
    predict1;
  elseif k==5,
    k=0;clc;
  end
end
```

26. โปรแกรมย่อย predict1.m

```
% predict1 file
% to predict meantec by meanh, meand, meanz by UI-control
clf reset;fig=gcf;figure(fig) % plot on current figure
plot(avgtec(sday:eday),'b');grid
title('averaged TEC data every 30 minutes');
xlabel('time UT [30 minutes]');
ylabel('vertical TEC  [TECU]');
d=0;
buttons=buttonv([.18 .8], 0:9,d,'predict2','polynomial','return');
```

27. โปรแกรมย่อย predict2.m

```
% predict2 called from predict1
% x=independent variable
% y=dependent variable
gold = [.9 .6 .3];
tinv = [12.706 4.303 3.182 2.776 2.571 2.447 2.365 2.306 2.262];
if button == 13
  % Toggle confidence intervals.
  bounds = 1 - bounds;
else
  % Reset the color of the previous button.
  set(buttons(d+1),'back','default')
  % Get the new degree from the button.
  d = button-1;
  % Set the new degree to gold.
  set(buttons(d+1),'back',gold);
end
% Set the cursor to a watch to show computation in progress.
point = get(fig,'pointer');
set(fig,'pointer','watch');
drawnow
if d <= 9
```

```
beta=polyfit(x,y,d);
predict = polyval(beta,x_r);
disp(['curvilinear regression coefficients are: ' num2str(beta)]);
% find correlation
x1=avgtec(sday:eday);
%x1=TEC;
  y1=predict;
    a=length(x1);b=length(y1);
    if a>b,
      for i=length(y1)+1:length(x1),
        y1(i)=nan;
      end
    end
    a=find(isnan(x1));b=find(isnan(y1));
    y1(a)=nan;x1(b)=nan;
    x2=x1(~isnan(x1));y2=y1(~isnan(y1));
    c=corrcoef(x2,y2);
    r2=rsquare(x,y,beta);
  if d < 9
    plot(avgtec(sday:eday),'b');hold on;grid
    plot(predict,'r-.');hold off
    title(['Polynomial fit, degree = ' int2str(d)])
    xlabel(['time  [30 minutes]  R^2 is: ' num2str(r2) ' ,correlation coefficient is: ' num2str(c(2))]);
    ylabel('TEC [TECU]');
  else
    plot(avgtec(sday:eday),'b');hold on;grid
    plot(predict,'r-.');hold off
    title('Polynomial interpolation, degree = 9')
    xlabel(['time  [30 minutes]  R^2 is: ' num2str(r2) ' ,correlation coefficient is: ' num2str(c(2))]);
    ylabel('TEC [TECU]');
  end
else
  % Done
  set(fig,'pointer',point)
  disp('Done')
  return
end
% Update the plot
set(fig,'pointer',point)
```

28. ฟังก์ชัน readbintec.m

```
function [mjd,hh,mm,num,tec,rms]=readbintec(fp);
% to read binary TEC data
fid=fopen(fp,'r');
id=1;
while feof(fid)~=1
  mjdutc=fread(fid,2,'ushort');
  sspn=fread(fid,2,'uchar');
  data=fread(fid,9,'short');
  if (isempty(mjdutc)~=1)&(data(8)>0)&(data(1)>=10)&(fix(data(3)/10)>=30)&(data(5)>=500)&(data(6)>=500)...
     &(data(9)<100)
    tec(id)=data(8)*sqrt(1-0.899*(cos(fix(data(3)/10)*pi/180))^2)/10;
    %save
    mjd(id)=mod(mjdutc(1),10000);
    hh(id)=fix(mjdutc(2)/60);
    mm(id)=mjdutc(2)-hh(id)*60;
    num(id)=data(1);
    rms(id)=data(9);
    id=id+1;
  end
end
fclose(fid);
```

29. ฟังก์ชัน readhdr.m

```
function [SLOPE,start_h,start_m,start_s]=readhdr(path,dr,sdate)
% get magnetic parameters
```

```
fp1=fopen(strcat(path,dr,'/',num2str(sdate),'.hdr'),'r');
t1=fscanf(fp1,'%c',259);
SLOPE=fscanf(fp1,'%12f',1);
txt1=fscanf(fp1,'%c',328);
stime=fscanf(fp1,'%c',12);
stime=str2num(stime);
stime=mod(stime,fix(sdate/100));
start_h=fix(stime/10000);
start_m=fix(stime/100)-start_h*100;
start_s=mod(stime,start_h*10000+start_m*100);
st=fclose(fp1);
```

30. ฟังก์ชัน readiri.m

```
function [TEC]=readiri(fp);
% read predicted TEC by using IRI-95 model
fid=fopen(fp,'r');
for k=1:48,
  TEC(k)=fscanf(fid,'%16f',1)/10^16;
end
st=fclose(fid);
```

31. ฟังก์ชัน readkp.m

```
function [Kp,ap,Ap,Cp,C9]=readkp(year,sday,eday);
% to read kp index
if year==1998,
  fid=fopen('d:/meaw/1998kp.dat','r');
elseif year==1999
  fid=fopen('d:/meaw/1999kp.dat','r');
end
Kp=[];AP=[];Ap=[];Cp=[];C9=[];S=[];SSN=[];
for k=1:365,
  ymd=fscanf(fid,'%2d',3);
  srn=fscanf(fid,'%4d',1);
  nday=fscanf(fid,'%2d',1);
  kpid=fscanf(fid,'%2d',8);
  kp(1)=kpid(1);kp(2:6)=nan;
  kp(7)=kpid(2);kp(8:12)=nan;
  kp(13)=kpid(3);kp(14:18)=nan;
  kp(19)=kpid(4);kp(20:24)=nan;
  kp(25)=kpid(5);kp(26:30)=nan;
  kp(31)=kpid(6);kp(32:36)=nan;
  kp(37)=kpid(7);kp(38:42)=nan;
  kp(43)=kpid(8);kp(44:48)=nan;
  Kp=[Kp kp];
  eight_val=fscanf(fid,'%3d',1);
  apid=fscanf(fid,'%3d',8);
  ap(7)=apid(2);ap(8:12)=nan;
  ap(13)=apid(3);ap(14:18)=nan;
  ap(19)=apid(4);ap(20:24)=nan;
  ap(25)=apid(5);ap(26:30)=nan;
  ap(31)=apid(6);ap(32:36)=nan;
  ap(37)=apid(7);ap(38:42)=nan;
  ap(43)=apid(8);ap(44:48)=nan;
  S=[S nansum(ap)];
  AP=[AP ap];
  Ap_day=fscanf(fid,'%3d',1);
  Ap=[Ap Ap_day];
  cp=fscanf(fid,'%4f',1);
  Cp=[Cp cp];
  c9=fscanf(fid,'%1d',1);
  C9=[C9 c9];
  ssn=fscanf(fid,'%3d',1);
  SSN=[SSN ssn];
  flux=fscanf(fid,'%6f',1);
  flux_q=fscanf(fid,'%1d',1);
end
```

```
Kp=Kp(sday:eday);
ap=AP(sday:eday);
Ap=Ap(sday:eday);
Cp=Cp(sday:eday);
C9=C9(sday:eday);
fclose(fid);
```

32. ฟังก์ชัน readssn.m

```
function ssn=readssn(year,sday,eday);
% to read and plot daily ssn,
if year==1998,
  fid=fopen('d:\meaw\dailyssn1998.txt','r');
elseif year==1999,
  fid=fopen('d:\meaw\dailyssn1999.dat','r');
end
for k=1:365,
  ymd=fscanf(fid,'%8d',1);
  yf=fscanf(fid,'%10f',1);
  ssn(k)=fscanf(fid,'%4d',1);
  nor_sou=fscanf(fid,'%4d',2);
end
fclose(fid);
ssn=ssn(sday:eday);
```

33. ฟังก์ชัน rsquare.m

```
function r2=rsquare(x,y,beta)
% find R^2 coefficient
[p,n]=size(x);
nu =p-n;        % Regression degrees of freedom
yhat = polyval(beta,x);  % Predicted responses at each data point.
r = y-yhat;          % Residuals.
if nu ~= 0
  rmse = norm(r)/sqrt(nu);       % Root mean square error.
else
  rmse = Inf;
end
% Calculate R-squared.
RSS = norm(y-mean(y))^2-norm(y-yhat)^2;  % Residual sum of squares.
TSS = norm(y-mean(y))^2;     % Total sum of squares.
r2 = RSS/TSS;              % R-square statistic.
```

34. ฟังก์ชัน sdiff1.m

```
function x1=sdiff1(x,s)
% x1=difference of x by nth order
% x=input vector
% s=seasonal index; period
for k=1:length(x)-s
  x1(k)=x(k+s)-x(k);
end
x2(1:48)=nan;
x1=[x2 x1];
```

ประวัติผู้เขียน

ชื่อ	นางสาวธิดาพร เอื้อรักสกุล
วัน เดือน ปี เกิด	4 กุมภาพันธ์ 2519
ประวัติการศึกษา	สำเร็จการศึกษามัธยมศึกษาตอนปลาย โรงเรียนพะเยาพิทยาคม พะเยา ปีการศึกษา 2536 สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเชียงใหม่ ปีการศึกษา 2540
ทุนการศึกษา	ได้รับทุนอุดหนุนบัณฑิตศึกษา สถาบันพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ (ส.ว.ท.ช.)
ผลงานวิจัย	Ionospheric Total Electron Content (TEC) Modelling over Chiang Mai Province ผู้แต่งร่วม อ.ธราดล โกมลมิศร์ และรศ.ดร. เอกชัย แสงอินทร์ การประชุมเสนอผลงานวิชาการระดับประเทศ: ผลงานวิจัยระดับบัณฑิตศึกษา ครั้งที่ 1 ณ สำนักบริการวิชาการ มหาวิทยาลัยเชียงใหม่ 10-11 มิถุนายน 2543