**EXPORTING DATA FROM MATLAB TO EXCEL (** xlswrite)

**Syntax**

xlswrite(filename, a)

xlswrite(filename, a, sheet)

xlswrite(filename, a, sheet, range)

**Description**

Writes data to a Microsoft Excel spreadsheet file.

**Inputs**

|  |  |
| --- | --- |
| **Name** | **Description** |
| **filename** | Specifies the Excel file to which you want to write. **file** is a string. |
| **a** | Specifies the data you want to write. **a** can be string or a numeric value in a scalar or a matrix. |
| **sheet** | Specifies the spreadsheet in **file** to which you want to write. **sheet** can be an integer that specifies the index of a spreadsheet or a string that specifies the name of a spreadsheet(i.e.’sheet1’). If you do not specify **sheet**, **MATLAB** writes **a** to the first spreadsheet. |
| **range** | Specifies the range of data in **sheet** you want to write. **range** is a string. The default is 'A1'. |

**Details**

To specify the **range**, you can use a cell number, such as 'A1', that specifies the top left cell at which to begin writing, or you can use a colon as a range operator. For example, you can specify 'A1:B3', where A1 and B3 specify the top left and bottom right cells to which you want to write data. If you use a colon as a range operator to specify the **range** and if the size of **a** is larger than the **range**, **MATLAB** ignores the elements outside of the **range**.

NB: filename should be a valid .xls or xlsx Excel file name.

Summary of syntax

**xlswrite ('FileName.xls', VariableName)**  
**xlswrite ('FileName.xls', VariableName, sheet)**

**xlswrite ('FileName.xls', VariableName, sheet,range)**

NB:

xls →97-2003 excel format

xlsx →2007-2010 excel format

If **VariableName is not already defined in matlab ,then it must be in string(i.e.’variablename’).**

## Examples

**Writing Numeric Data to the Default Worksheet**

1.a = magic(3);  
 xlswrite('test', a);  
pause(1)  
b = xlsread('test')

2. Write a 7-element vector to Microsoft Excel file testdata.xls. By default, the data is written to cells A1 through G1 in the first worksheet in the file:

* xlswrite('testdata', [12.7 5.02 -98 63.9 0 -.2 56]);

b = xlsread('testdata')

EXERCISE

In this exercise, we are going to reduce bearings from angles given initial bearing and export the results to an excel file assuming that you know how to

1.import an excel file into matlab

2.index into matrices

3.reduce H.C.R to horizontal angles

4.compute bearing and distances from coordinates.

Follow the steps carefully.

1.import excel file ‘TRY1.xlsx’ into the variable fn.

A=xlsread(fn,2);

B=xlsread(fn,1);

d=A(:,1); %assigning 1st column to HCR in degrees

m=A(:,2); %assigning 2nd column to HCR in minutes

s=A(:,3); %assigning 3rd column to HCR in seconds

dist=A(:,4); %assigning 4th column to distances

% getting starting coordinates loop traverse

fnc=B(1,1); %assigning 1st row,1st column initial northing coordinates

fec=B(1,2); %assigning 1st row,1st column initial easting coordinates

tnc=B(2,1); %assigning 2nd row,1st column final northing coordinates

tec=B(2,2); %assigning 2nd row,2nd column final easting coordinates

% computing initial bearing and distance from coordinates

dn=tnc-fnc; % change in northing coordinates

de=tec-fec; % change in easting coordinates

if dn==0 && de>0

bearing=90;

end

if dn<0 && de==0

bearing=180;

end

if dn==0 && de<0

bearing=270;

end

if dn>0 && de>0

bearing=atand(de/dn);

end

if dn<0 && de>0

bearing=(atand(de/dn))+180;

end

if dn<0 && de<0

bearing=(atand(de/dn))+180;

end

if dn>0 && de<0

bearing=(atand(de/dn))+360;

end

dms=[d m s];

deg=dms2degrees(dms);

n=(size(deg,1))/4 ;%size of file

% REDUCTION OF ANGLES FROM FIELD BOOK

% face left reductions

for i=1:n(:,1)

FL(i,1)=deg((i+i+i+i)-2)-deg((i+i+i+i)-3);

if find(FL(i,1)<0)

FL(i,1)=FL(i,1)+360 ;

end

end

% face right reduction

for j=1:n(:,1)

FR(j,1)=deg((j+j+j+j)-1)-deg((j+j+j+j));

if find(FR(j,1)<0)

FR(j,1)=FR(j,1)+360 ;

end

end

%COMPUTING DISTANCE FROM FIELD BOOK

% Back sight mean distance(Check)

for i=1:n(:,1)

S(i,1)=(dist((4\*i)-3)+dist((4\*i)))/2;

end

BS=S(1,1); % distance between control points

S(1,:)=[]; % deleting 1st row,1st column of the matrice

s=[S;BS] ;% rearranging back sight distances

% Fore sight mean distance(original)

for j=1:n(:,1)

D(j,1)=(dist((4\*j)-2)+dist((4\*j)-1))/2;

end

% mean distance

mean\_dist=(D+s)/2;

%mean included angle

mean\_angle=(FR+FL)/2;

% sum of measured angles

sum\_angles1=sum(mean\_angle);

dms\_sum\_angles1=round(degrees2dms(sum\_angles1));

% mathematical check for external angles

sum\_angles2=((2\*n(:,1))+4)\*90;

dms\_sum\_angles2=degrees2dms(sum\_angles2);

% misclose

misclose=dms2degrees(dms\_sum\_angles1)-dms2degrees(dms\_sum\_angles2);

misclose=degrees2dms(misclose);

%REDUCTION OF UNADJUSTED FORWARD BEARING

FB=zeros(n,1);

FB=[bearing;FB];

mean\_angle1=[0;mean\_angle];

[r c]=size(mean\_angle1);

for i=1:r

if (FB(i)+mean\_angle1(i+1))<180

FB((i+1),1)=(FB(i)+mean\_angle1(i+1))+180;

else

FB((i+1),1)=(FB(i)+mean\_angle1(i+1))-180;

if find(FB((i+1),1)>360)

FB((i+1),1)=FB((i+1),1)-360;

else

FB((i+1),1)=FB((i+1),1);

i=i+1;

if i==r

break

end

end

end

end

FB(1,:)=[];

FB\_DMS=degrees2dms(FB);

% CORRECTION PER STATION

cps\_1=dms2degrees(bearing\_DMS)-dms2degrees(FB\_DMS(end,:));

cps=cps\_1/n;

% CORRECTION FACTOR

%....................

CF=mean\_angle;

for i=1:n(:,1)

CF(i,1)=i;

end

% ADJUSTMENT

CF=CF.\*cps;

% ADJUSTED FORWARD BEARING

RFB=CF+FB ;% adding adjustment to unadjusted for bearing

RFB\_DMS=round(degrees2dms(RFB)); % converting adjusted forward bearing degrees to degree,minute,seconds

%COORDINATES COMPUTATIONS

dN=mean\_dist.\*cosd(RFB); % latitude computation

dE=mean\_dist.\*sind(RFB); % departure computation

e\_x=sum(dN); % sum of latitudes

e\_y=sum(dE); % sum of departures

distance=sum(mean\_dist); % sum of distances

LM=sqrt((e\_x)^2+(e\_y)^2); % linear misclose

FM=roundn(distance/LM,3); % fractional misclose

c\_e\_x=(mean\_dist./distance)\*(-1\*e\_x); % correction to latitudes

c\_e\_y=(mean\_dist./distance)\*(-1\*e\_y); % correction to departures

% ADJUSTMENT TO LATITUDES AND DEPARTURES

adjusted\_dN=c\_e\_x+dN;

adjusted\_dE=c\_e\_y+dE;

format long g

N=tnc+cumsum(adjusted\_dN); % northing coordinates

E=tec+cumsum(adjusted\_dE); % easting coordinates

N(end-1:end,:)=[fnc;tnc];

E(end-1:end,:)=[fec;tec];

a4=n;

a5=(1:a4)';

[num,raw,txt]=xlsread(fn,2);

txt1=txt(:,1);

txt1(1:2,:)=[];

txt2=txt(:,3);

txt2(1:2,:)=[];

[r c]=size(txt1);

for i=1:r

txt3(i,1)=txt1((4\*i)-3); % instrument stations

txt4(i,1)=txt2((4\*i)-3); % forward stations

if i==n(:,1)

break

end

end

txt5=txt3(end,1);

txt6=txt3(1,1);

% WRITING RESULT TO AN EXCEL FILE

% writing to reduction of forward bearing

xlswrite('TRAVERSE COMPUTATIONS.xls',txt3,'REDUCTION OF FWD BEARING','B7')

xlswrite('TRAVERSE COMPUTATIONS.xls',txt4,'REDUCTION OF FWD BEARING','C7')

xlswrite('TRAVERSE COMPUTATIONS.xls',D\_M\_S,'REDUCTION OF FWD BEARING','D7')

xlswrite('TRAVERSE COMPUTATIONS.xls',FB\_DMS,'REDUCTION OF FWD BEARING','G7')

xlswrite('TRAVERSE COMPUTATIONS.xls',RFB\_DMS,'REDUCTION OF FWD BEARING','J7')

xlswrite('TRAVERSE COMPUTATIONS.xls',a5,'REDUCTION OF FWD BEARING','A7')

% writing to traverse computation

xlswrite('TRAVERSE COMPUTATIONS.xls',txt5,'TRAVERSE COMPUTATIONS','B14')

xlswrite('TRAVERSE COMPUTATIONS.xls',txt6,'TRAVERSE COMPUTATIONS','P14')

xlswrite('TRAVERSE COMPUTATIONS.xls',txt3,'TRAVERSE COMPUTATIONS','B15')

xlswrite('TRAVERSE COMPUTATIONS.xls',txt4,'TRAVERSE COMPUTATIONS','P15')

xlswrite('TRAVERSE COMPUTATIONS.xls',D\_M\_S,'TRAVERSE COMPUTATIONS','C15')

xlswrite('TRAVERSE COMPUTATIONS.xls',bearing\_DMS,'TRAVERSE COMPUTATIONS','F14')

xlswrite('TRAVERSE COMPUTATIONS.xls',RFB\_DMS,'TRAVERSE COMPUTATIONS','F15')

xlswrite('TRAVERSE COMPUTATIONS.xls',mean\_dist,'TRAVERSE COMPUTATIONS','I15')

xlswrite('TRAVERSE COMPUTATIONS.xls',dN,'TRAVERSE COMPUTATIONS','J15')

xlswrite('TRAVERSE COMPUTATIONS.xls',dE,'TRAVERSE COMPUTATIONS','k15')

xlswrite('TRAVERSE COMPUTATIONS.xls',c\_e\_x,'TRAVERSE COMPUTATIONS','L15')

xlswrite('TRAVERSE COMPUTATIONS.xls',c\_e\_y,'TRAVERSE COMPUTATIONS','M15')

xlswrite('TRAVERSE COMPUTATIONS.xls',tnc,'TRAVERSE COMPUTATIONS','N14')

xlswrite('TRAVERSE COMPUTATIONS.xls',tec,'TRAVERSE COMPUTATIONS','O14')

xlswrite('TRAVERSE COMPUTATIONS.xls',N,'TRAVERSE COMPUTATIONS','N15')

xlswrite('TRAVERSE COMPUTATIONS.xls',E,'TRAVERSE COMPUTATIONS','O15')

xlswrite('TRAVERSE COMPUTATIONS.xls',LM,'TRAVERSE COMPUTATIONS','H5')

xlswrite('TRAVERSE COMPUTATIONS.xls',FM,'TRAVERSE COMPUTATIONS','J8')

xlswrite('TRAVERSE COMPUTATIONS.xls',e\_x,'TRAVERSE COMPUTATIONS','L5')

xlswrite('TRAVERSE COMPUTATIONS.xls',e\_y,'TRAVERSE COMPUTATIONS','L8')

xlswrite('TRAVERSE COMPUTATIONS.xls',a5,'TRAVERSE COMPUTATIONS','A15')