File Input and Output

Spring 2019

Objective

- `input` function reads values entered by a user.

Objective

- `input` function reads values entered by a user.
- `load` function read data from a file.

Objective

- `input` function reads values entered by a user.
- `load` function read data from a file.
- `save` function writes from a matrix to a data file.

Objective

- `input` function reads values entered by a user.
- `load` function read data from a file.
- `save` function writes from a matrix to a data file.
- However, `load` will only work as long as the data has the same values on each line and these values are of the same data type, so that they can be stored in a matrix.

Objective

- `input` function reads values entered by a user.
- `load` function read data from a file.
- `save` function writes from a matrix to a data file.
- However, `load` will only work as long as the data has the same values on each line and these values are of the same data type, so that they can be stored in a matrix.
- If the file has different formats we need **lower level file Input/Output** functions.

Opening and closing files

- `fopen` opens a file for reading, writing or appending.

Opening and closing files

- `fopen` opens a file for reading, writing or appending.
- `fopen` returns -1 if not successful in opening the file, or an integer value that becomes the **file identifier**.
- `fid = fopen('filename','permission string')`

Opening and closing files

- `fopen` opens a file for reading, writing or appending.
- `fopen` returns -1 if not successful in opening the file, or an integer value that becomes the **file identifier**.
- `fid = fopen('filename','permission string')`
- **permission strings** include:
  1. `r` read (default)
  2. `w` write
  3. `a` append
- `>>help fopen` for other permissions

Opening and closing files

- After calling `fopen`, value returned must be checked to make sure that the file has been successfully opened.

Opening and closing files

- After calling `fopen`, value returned must be checked to make sure that the file has been successfully opened.
- Use `fclose` to close the file after finishing reading, writing or appending them.

Opening and closing files

- After calling `fopen`, value returned must be checked to make sure that the file has been successfully opened.
- Use `fclose` to close the file after finishing reading, writing or appending them.
- `fclose` returns `0` if the file close was successful, or `-1` if not.
- One must also check that the file has been closed properly at the end of the program.

fscanf

- `fscanf` will read data from a file directly into a matrix

fscanf

- `fscanf` will read data from a file directly into a matrix
- Each line of the data file is stored as a column of the matrix

fscanf

- `fscanf` will read data from a file directly into a matrix
- Each line of the data file is stored as a column of the matrix
- The matrix may need to be reformatted to get it back into the original form from the file
- **Usage**: `mat = fscanf(fid,'format',[dimensions])`

fscanf

- `fscanf` will read data from a file directly into a matrix
- Each line of the data file is stored as a column of the matrix
- The matrix may need to be reformatted to get it back into the original form from the file
- **Usage**: `mat = fscanf(fid,'format',[dimensions])`
  1. `format` includes conversion characters we used for `fprintf`, e.g
     `%d` - integers `%f` - floats (decimals)
  2. `dimensions` specify the desired dimensions of `mat`
  3. `inf` can be used for the second dimension if this value is not known

fscanf

- `fscanf` will read data from a file directly into a matrix
- Each line of the data file is stored as a column of the matrix
- The matrix may need to be reformatted to get it back into the original form from the file
- **Usage**: `mat = fscanf(fid,'format',[dimensions])`
  1. `format` includes conversion characters we used for `fprintf`, e.g
     `%d` - integers `%f` - floats (decimals)
  2. `dimensions` specify the desired dimensions of `mat`
  3. `inf` can be used for the second dimension if this value is not known

Exercise

Download the provided `emissions.txt` file and save it in your current directory. Write a script that performs the following tasks

1. Open the file
2. Check to make sure that the file has been open successfully or otherwise and print out the appropriate message
3. Close the file and check to make sure that the file has been closed successfully.

Writing to files

- `fprintf` - write one line at a time to a file.

Writing to files

- `fprintf` - write one line at a time to a file.
- **Usage:**

```
fprintf(fid,'format',variable(s))
```

Writing to files

- `fprintf` - write one line at a time to a file.
- **Usage:**

```
fprintf(fid,'format',variable(s))
```

- `fprintf` returns the number of bytes that was written to the file

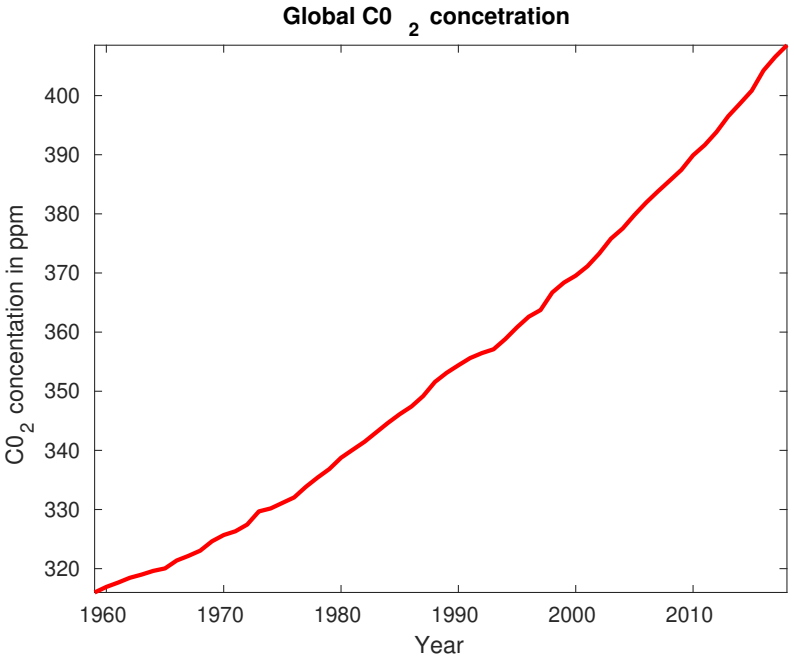
Exercise

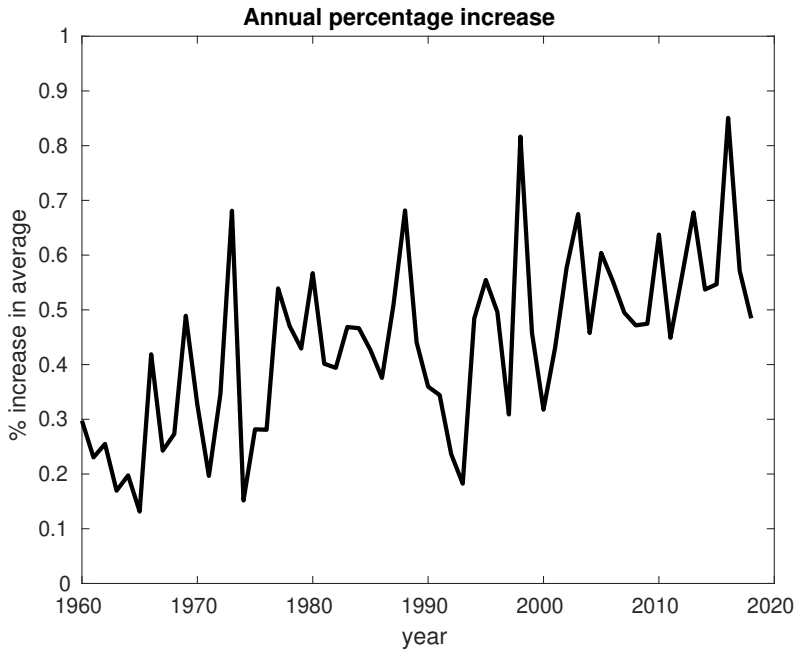
Visualizing the trend of concentration of $C0_2$ in the atmosphere over time

The file `emissions.txt` (downloaded from NOAA) contains the average $C0_2$ concentration per year.

1. Use `fscanf` to read the data in the file `emissions.txt`
2. Use the data to plot the growth of concentration of $C0_2$ over time
3. Use the data to plot the annual percentage increase in $C0_2$ concentration over time.
4. Save the annual percentage increase values in a file.

Exercise


Global C0 2 concetration
C0 2 concentation in ppm
400
390
380
370
360
350
340
330
320
1960
1970
1980
1990
2000
2010
Year


Annual percentage increase
% increase in average
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1960
1970
1980
1990
2000
2010
2020
year