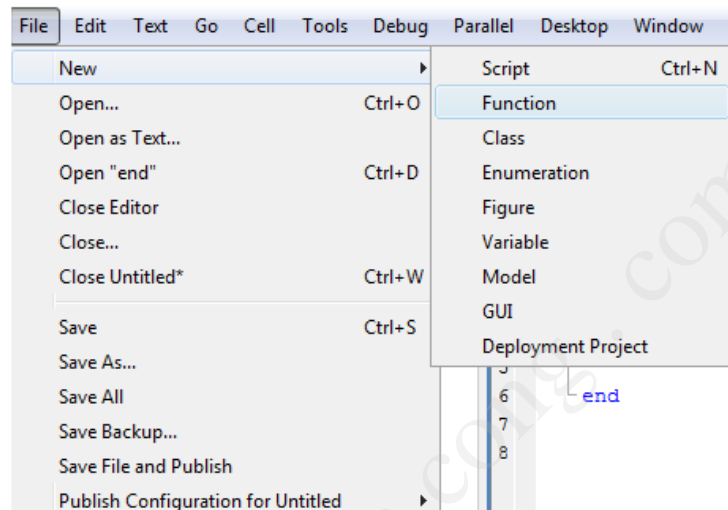


## BÀI 2A: GIẢI HỆ PHƯƠNG TRÌNH TUYẾN TÍNH

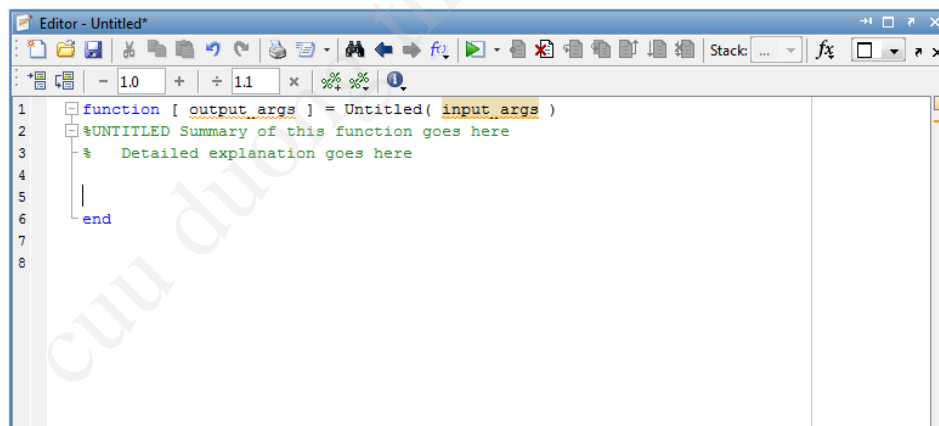
### II. PHẦN THỰC HÀNH

Phần hướng dẫn này sẽ tập trung vào cách viết và sử dụng file function cho Matlab. Code cụ thể cho từng bài các em có thể tham khảo trong sách.

Để tạo file function trong Matlab ta vào file chọn Function.



Cấu trúc của 1 file function sẽ như sau:



Mở đầu là lệnh `function`, sau đó là tên giá trị biến được xuất ra, rồi tới tên hàm và các biến được truyền vào. Chú ý, tên hàm chính và tên file phải giống nhau. Cuối cùng là lệnh `end` báo kết thúc hàm.

Ví dụ: `function x = GaussNaive(A,b)`

Tên giá trị biến được xuất ra: `x`

Tên hàm chính : `GaussNaive` => tên file được lưu là `GaussNaive.m`

Tên giá trị các biến được truyền vào: `A, b`.

Trong 1 file function ta có thể khai báo nhiều hàm, hàm chính là hàm khai báo đầu tiên. Các hàm sau có thể là hàm được sử dụng lại trong hàm chính.

Ví dụ: 2 hàm sau được viết trong cùng 1 file tên `subtract_1.m`

```
function [ x ] = subtract_1( y )
```

```

x = subtract_2(y) - 1
end

function [ x ] = subtract_2( y )
x = y - 2
end

```

Để chạy file function này, trong command window ta gọi tên file chứa hàm và truyền biến vào như sau:

```
>> ans = subtract_1(4)
```

Kết quả xuất ra sẽ là:

```

x =

    2

x =

    1

ans =

    1

```

Quá trình Matlab thực hiện câu lệnh `ans = subtract_1(4)` như sau:

1. Tìm file tên `subtract_1`, hàm đầu tiên khai báo là hàm chính. Matlab sẽ thực hiện hàm này đầu tiên.
2. Trong hàm `subtract_1`, lệnh `x = subtract_2(y) - 1` gọi tới hàm `subtract_2`. Matlab sẽ tìm hàm này và thực thi.
3. Hàm `subtract_2` được thực thi, giá trị biến truyền vào (`y`) là 4 nên kết quả trả về (`x`) là 2. Do không có dấu `;` cuối câu nên kết quả được xuất ra cả command window.
4. Hàm `subtract_1` tiếp tục được thực thi, kết quả trả về (`x`) của hàm này là 1 (`x = 2 - 1`). Do không có dấu `;` cuối câu nên kết quả được xuất ra command window.
5. Biến `ans` nhận kết quả trả về từ hàm `subtract_1` và do không có dấu `;` cuối câu nên giá trị của biến `ans` cũng được xuất ra command window.

Như vậy, đối với yêu cầu của bài 3 trong lab 2A. Các em sẽ viết một hàm tên `GaussNaive`, các giá trị truyền vào hàm này là ma trận trái `A` và ma trận phải `b`, thuật toán được sử dụng sẽ là thuật toán giải hệ phương trình: gauss không hoán đổi trụ. Sau khi đã viết hàm xong thì trong command window (hoặc tạo một file script) các em sử dụng hàm này như sau:

```

>> A = [10 2 -1; -3 -6 2; 1 1 5];
>> b=[27; -61.5; -21.5];
>> roots = GaussNaive(A,b)

```

Các câu sau cũng tương tự như vậy, viết một file function mới và sử dụng file function đó để giải hệ phương trình.