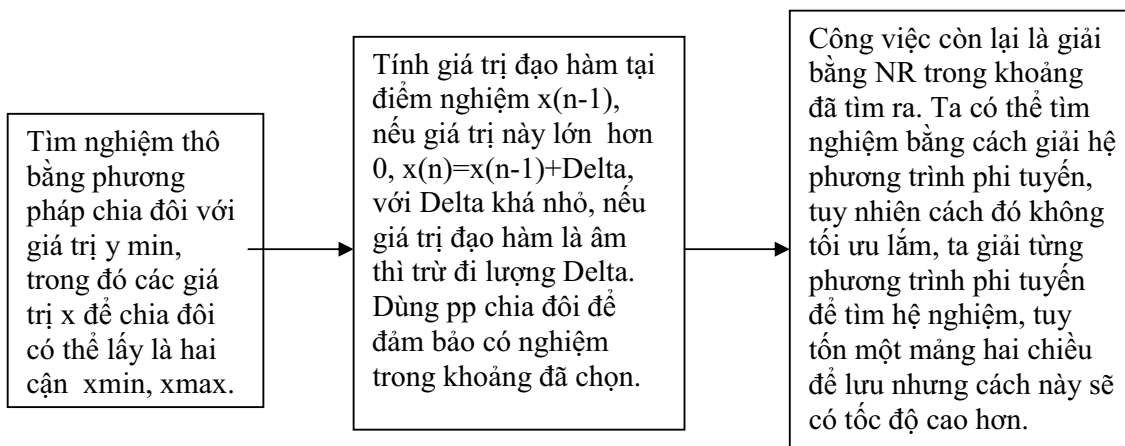


Lời nói đầu:

Mục đích chính của phương pháp Newton-Raphson là giải phương trình hoặc hệ phương trình phi tuyến có đạo hàm liên tục và đơn trị trong khoảng $[a, b]$ xác định. Với những phương trình hoặc hệ phương trình tuyến tính, nhiều khi cách giải thông thường nhờ biến đổi qua lại giữa các biến có kết quả cao hơn nếu giải bằng phương pháp NR.

Trong các phương pháp số để giải hệ phi tuyến, phương pháp chia đôi là một trong những phương pháp tìm nghiệm thô hiệu quả nhất tuy nó chỉ có tốc độ hội tụ bậc nhất. Từ nghiệm thô đó, ta có thể tìm các nghiệm khác nhờ các tính chất của hàm số.

Phương pháp sử dụng trong bài tập này là:



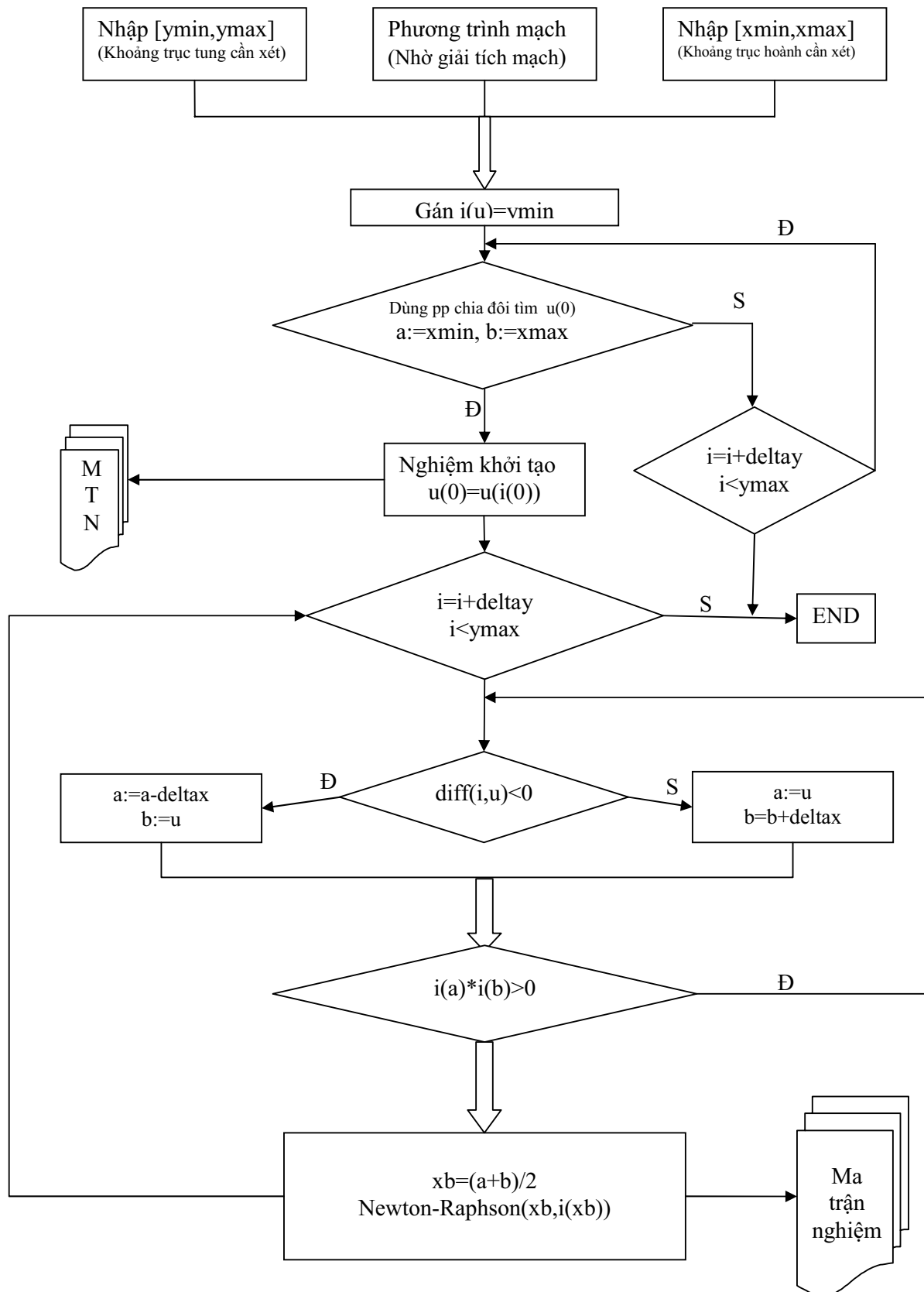
Tuy nhiên, cách trên chỉ áp dụng với những hàm đơn trị, khả vi và đơn ánh ($y \rightarrow x \rightarrow y$) thế nên còn những hạn chế rất lớn. Trong phạm vi của chương trình, hàm mũ, logarit, một nhánh của hàm bậc hai, phần đơn trị của các hàm bất kì... là những hàm thường gặp, với công cụ là phương pháp số, ta có thể vẽ được dạng các đặc tuyến của chúng.

Tìm được nghiệm của chúng với khối lượng lớn các phép tính, tốc độ có thể chậm, tuy nhiên, ta sẽ cải thiện chúng bằng các phương pháp như trong khoảng x_{min} , x_{max} của đồ thị biểu diễn, ta tìm khoảng từ 20 đến 100 giá trị, sau đó xấp xỉ các giá trị khác bằng nội suy Runga-Kutta bậc 4 chẳng hạn. Nhưng thật ra, với các hàm đưa ra, việc tìm khoảng 50 giá trị trong khoảng đã cho thật ra không khó khăn lắm, (mất khoảng 2-3 phút cho một bài toán phức tạp) nếu bài toán giải mạch đưa đến việc phải tìm ra nghiệm của hai hoặc ba, hoặc nhiều hơn các phương trình phi tuyến, ta phải hạn chế bằng cách khoảng chia Δy lấy nghiệm dài ra. Cách trên sẽ khiến đồ thị không còn mượt như trước, nhưng chủ yếu, với bài toán kĩ thuật, ta chỉ cần hình dạng của chúng nên cũng không đáng lo.

Lúc đầu, người viết định thiết kế một mô hình trực quan và đóng gói sản phẩm thành một đồ án (Project), nhưng do không đủ thời gian nên chỉ hạn chế ở phần viết code và phần chỉnh sửa hàm phi tuyến phải can thiệp trực tiếp vào code. Tuy nhiên, người viết cũng đã cố gắng đưa hàm phi tuyến đó vào một phần riêng biệt để người sử dụng có thể dễ dàng chỉnh sửa mà không ảnh hưởng đến các phần khác của chương trình.

Với cách sử dụng đặc tuyến Diode là hàm Hyperbole ở phần góc phần tư thứ hai của tọa độ Đề Các, người viết hi vọng đã giải quyết được hầu hết các bài toán cộng đồ thị trong sách mạch 2 có đặc tuyến hàm phi tuyến là của Diode lý tưởng, hoặc Diode Zenner lý tưởng, nếu người sử dụng đưa vào đúng phương trình giải tích của mạch. Với cách giải quyết bài toán phi tuyến số bằng phương pháp Newton-Raphson, người viết hi vọng đã giải quyết được hầu hết các bài toán mạch phi tuyến có hàm biểu diễn đặc tuyến i-u là hàm đơn ánh (tương ứng một u thì có một I và ngược lại), khả vi trong từng đoạn xét.

Sơ đồ thuật toán



Bài toán không ngừng tại chỗ chỉ dựng được đặc tuyến $i-u$ của mạch, với cách dùng thêm biến thời gian t , ta có thể dựng được các đáp ứng $i(t)$ khi có $u(t)$ (hoặc ngược lại) nhờ đặc tuyến $i-u$ vừa dựng (xem ví dụ ở phần cộng đồ thị ở bài 9.4c,d). Người viết hi vọng có thể giải quyết nhiều vấn đề khác, nhưng thời gian hạn chế nên chỉ dừng tại đó, có thể có nhiều ứng dụng mới tạo thành từ những chương trình đã viết sẵn.

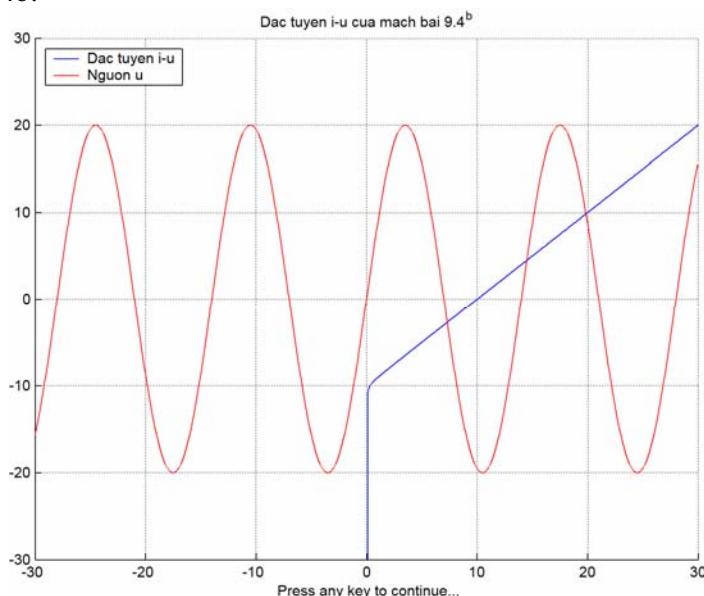
Tuy nhiên, người viết cũng nhận thấy phần hạn chế của chương trình khi xử lý số ở những khoảng chia nhỏ, đó cũng là lý do tại sao các đặc tuyến có phần hơi khác với lý thuyết.

Cuối cùng, khi được sử dụng, phần các hàm đã có của chương trình hi vọng được add ở một thư mục riêng biệt để không nhầm lẫn với các hàm khác đã có từ các chương trình trước. Phần lỗi này do người viết code chưa xây dựng được một Project cho chương trình.

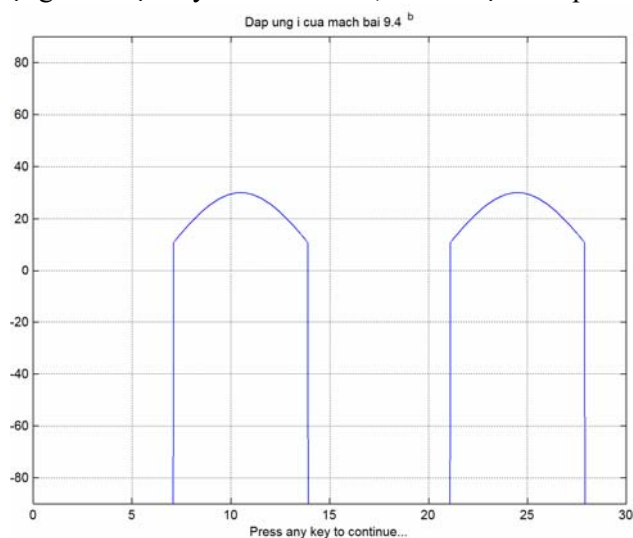
Nói về hai bài tập 9.4b và 9.4c, thực ra nó có cùng một nội dung, code dung chung cho cả hai bài ở các thuật giải, tuy nhiên, như đã nói rõ ở từng bài, chúng chưa được xây dựng hoàn hảo nên không thể dung một cách tổng quát được.

Sau khi tính toán, ta được các kết quả sau:

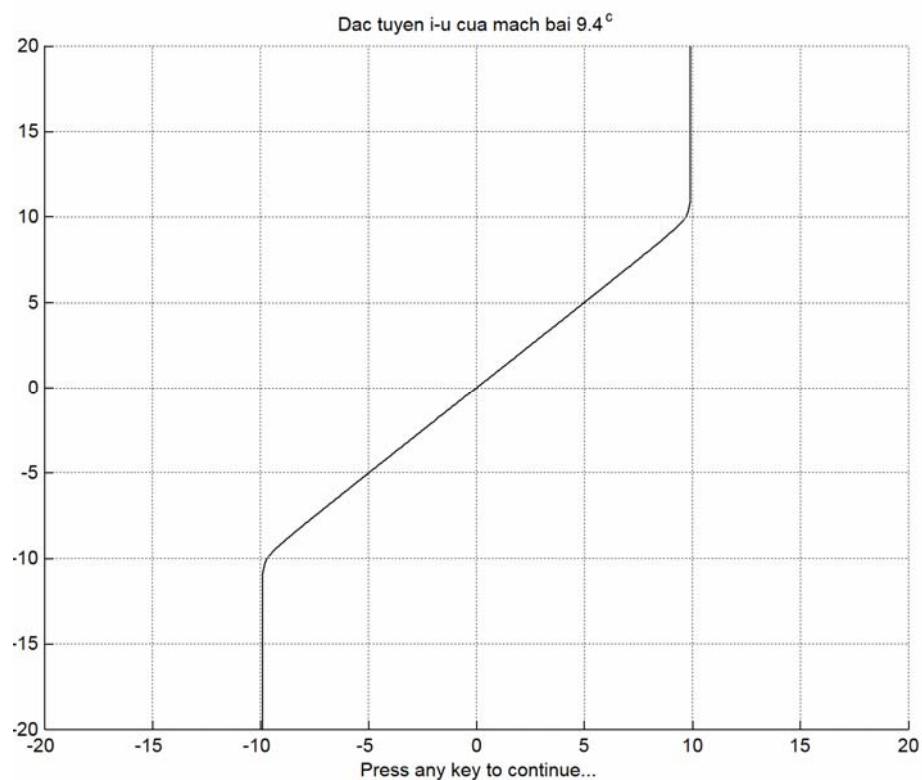
Ở bài 9.4b:



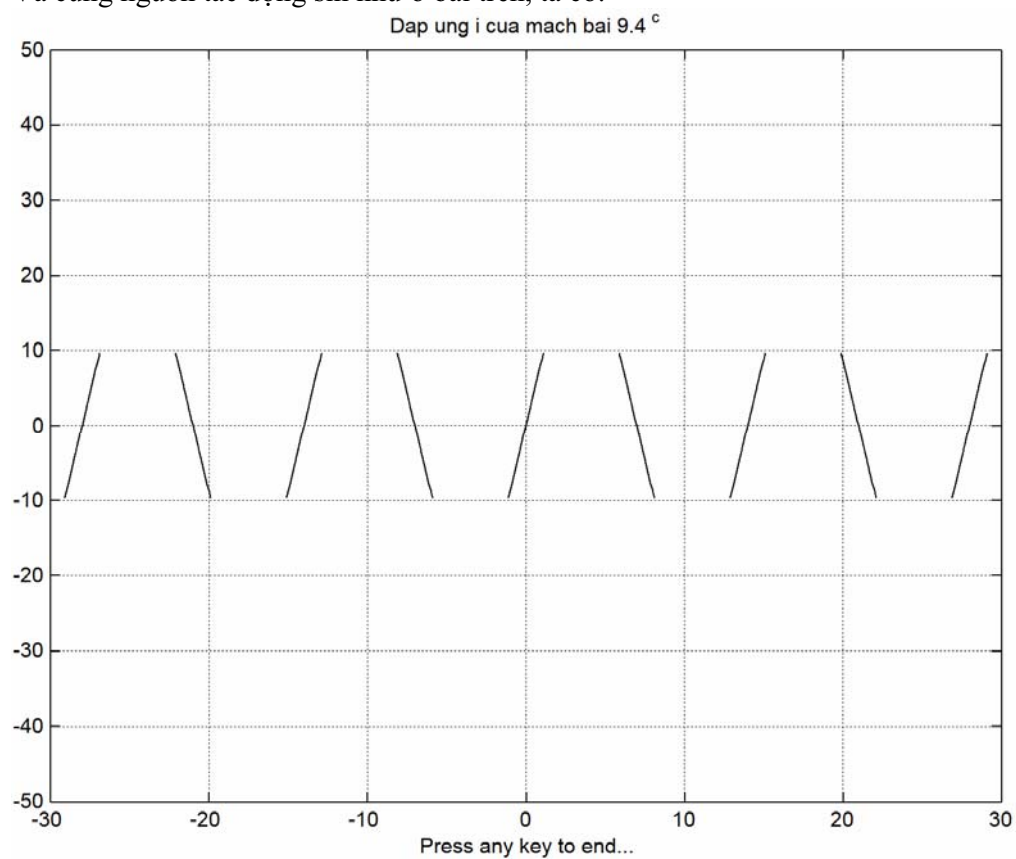
Với tác động u và đặc tuyến $i-u$ như thế, ta sẽ được kết quả:



Ở bài 9.4c:



Và cùng nguồn tác động sin như ở bài trên, ta có:



Code của chương trình được viết ở hai phương pháp: Cộng đồ thị và NRS, (kèm theo đĩa) Phần dưới đây là đoạn code viết cho NRS.

```
function[]=bai4b()
a=-30; %xmin-Co the chinh sua
b=30; %xmax-Co the chinh sua
fig=figure( ...
    'Name','Dac tuyen cua mach', ...
    'NumberTitle','off', ...
    'Visible','on', ...
    'BackingStore','on');
anh=imread('lacviet.jpg','jpeg');
image(anh);
title('Vui long doi trong 1 phut');
pause(0.00001);

k=1;
bien(1)=a;
xdamp=10^-3;          %!Chi rieng voi bai na`y!
bx(1)=nrs(eps,bien(1)); %Dung cau lenh nay toc do hoi tu se cao hon!!!Chi
                        %rieng voi bai na`y!
%bx(1)=chiadoi(a,b,bien(1)); %Dung voi bai toan tong quat tim nghiem ban dau .
%nhung hoi tu cham

anh=imread('anh2.jpg','jpeg');
image(anh);
title('Vui long doi trong 1 phut');
pause(0.00001);

while bien(k)<=b
    k=k+1;
    bien(k)=bien(k-1)+2;      %Dung NR giai ra 30 nghiem cua phuong trinh
    bx(k)=nrs(xdamp+bx(k-1),bien(k)); %?: De cho ham so dam bao hoi tu, chi .
    %ieng voi bai nay!
end
%=====
clf;
hold on;
plot(bx,bien);
axis([a b a b]);
title('Dac tuyen i-u cua cau 9.4 ^b');
xlabel('Nhan phim bat ki de ket thuc!');
grid on;
pause;
clear bx bien k anh
close(1);
%=====
function[x]= chiadoi(a,b,bieny)
syms t
epsilon=10^-6;
delta=1;
```

```

%=====Tim khoang cach li nghiêm=====
%=====Cach mo rong ve hai ben khoang cach li nghiêm bat ki nhap vao=====

while (h(a,bieny)*h(b,bieny)>0)&&b<30&&a>-30
    a=a-delta;
    b=b+delta;
end
%=====Thuat toan=====
    while(abs(a-b)>epsilon)
        if(h(a,bieny)==0)
            x=a;
            break;
        elseif(h(b,bieny)==0)
            x=b;
            break;
        else
            c=(a+b)/2;
            if( h(c,bieny)==0)
                x=c;
                break;
            elseif (h(c,bieny)*h(a,bieny)<0)
                x=c;
                b=c;
            else
                x=c;
                a=c;
            end
        end
    end
end
%=====
function[x]=nrs(xb,bieny)
N=50;
x=xb;
syms t;                                %Bien t la` bien trung gian
saio=eps;
i=1;
%//////////Dua vao ham phi tuyen//////////%
fo=f-bieny;
fd=diff(fo);                            %Tinh dao ham cua ham da cho
%.....Vong lap tim nghiêm bang Newton-Raphson.....%
while(i<N)                               %Kiem tra dieu kien i<N
    ydh=subs(fd,t,x);                    %tinh ydh=fd(x)
    yx=subs(fo,t,x);                     %Tinh yx=f(x)
    x=x-(yx/ydh);                         %Ap dung cong thuc N-R tinh x(i)
    if (abs(xb-x))<saio                   %Kiem tra dieu kien x(i)-x(i-1)<saio
        break;
    end
    xb=x;                                %Gan gia tri x(i)=xb cho vong lap moi;
    i=i+1;
end

```

```

%=====
if diot(-x)<0&&(abs(x)>0.5)
    x=nan;
end
%=====
function[idiot]=diot(udiot)
idiot=-1/(10*udiot-0.1*eps);      %Xay dung ham diot
%=====
function[f]=f()
syms t ;
f=-diot(-t)+t-10;
%=====
function[y]=h(v,bieny)
syms t;
y=subs(f-bieny,t,v);

```

%Co the mo rong tim ham truyen hoac dap ung u theo thoi gian khi biet i(t) va nguoc %lai.Tai sao nguoi viet khong su dung ha`m mu~ cho diot trong truong hop nay`?Don gian %vi` phai xu? li' cac so rat lon' cua ham mu~ la` mot chuyen rat kho' khan, neu su dung pp %chia doi thi` la`m cham chuong tri`nh, co`n su? dung pp NRS thi` se~ co' cac dao ha`m tien %toi vo cung khi i kha' lon!Moi chuyen se~ thu' vi hon nhieu voi diode Zenner, khi ma` ha`m %tan(x) trong mot chu ki`) co'the bieu dien duoc dac tuyen cua no' khi ha`m do' duoc ngan %voi' mot he so du nho!

Phần code cho bài 4c tương tự như trên, chỉ chỉnh sửa đôi chút ở hàm main (kèm theo đĩa)

Cuối cùng, do thời gian có hạn nên người viết code chỉ giới hạn ở những kết quả đơn giản, hy vọng từ những hàm đã viết, có thể mở rộng kết quả cho nhiều bài toán khác! Khi có nhiều thời gian hơn, người viết sẽ có đoạn code chung cho các chương trình dạng này với các cách xử lý chuyên nghiệp hơn và đóng gói sản phẩm hoàn chỉnh.

Tp Hồ Chí Minh, ngày 4 tháng 5 năm 2004!

Người viết:

Bùi Trung Hiếu

Đến thời điểm làm bài tập này, người viết chỉ mới biết đến sử dụng Matlab trong khoảng gần 1 tháng nên còn rất hạn chế! Phần này chỉ nên tham khảo! BTH