Đoạn mã mà bạn đã chia sẻ là một hàm MATLAB được gọi là "nrsolve" để giải phương trình phi tuyến tính. Dưới đây là phân tích của code:

matlab Copy Code

```matlab
function [ x ] = nrsolve( ce, res, maxLoop )
  [ceX, ceY] = size(ce);
  if (ceX > 1 && ceY == 1)
    ce = ce';
    [ceX, ceY] = size(ce);
  elseif (~(ceX == 1 && ceY > 1))
    disp(' ce is not valid ')
    return;
  end
  res = abs(res);
  % calculate derivate coefficients
  dce = zeros(ceX, ceY  - 1);
  for i=2:ceY
    if (ce(i) ~= 0)
      dce(i-1) = ce(i)*(i-1);
    end
  end
  % a rude guess, I guess :-)
  x0 = 1;
  x1 = x0 - evalf(ce, x0)/evalf(dce, x0);
  x0 = x1;
  iCnt = 0;
  while (abs(evalf(ce, x1)) > res && iCnt < maxLoop)
    x1 = x0 - evalf(ce, x0)/evalf(dce, x0);
    x0 = x1;
    iCnt = iCnt + 1;
  end
  x = x1;
end
```

Đoạn mã trên định nghĩa một hàm `nrsolve` với các tham số đầu vào là `ce`, `res`, và `maxLoop`. Hàm này sử dụng phương pháp Newton-Raphson để giải phương trình phi tuyến tính. Dưới đây là mô tả chi tiết các bước trong hàm:

1. Kiểm tra kích thước của ma trận `ce` và xử lý nếu cần thiết để đảm bảo `ce` có kích thước chính xác.
2. Tính toán ma trận `dce` chứa các hệ số đạo hàm từ ma trận `ce`.
3. Thiết lập giá trị ban đầu `x0` cho ước lượng ban đầu.
4. Sử dụng công thức Newton-Raphson để tính toán giá trị `x1` gần đúng sao cho `evalf(ce, x1)` tiến gần đến 0.
5. Lặp lại quá trình tính toán `x1` cho đến khi giá trị `evalf(ce, x1)` đạt đủ độ chính xác hoặc số vòng lặp đã đạt tối đa `maxLoop`.
6. Trả về giá trị cuối cùng của `x`.

Sau đó, code cung cấp một hàm phụ được gọi là `evalf` để tính toán giá trị của đa thức với các hệ số `cs` tại điểm `x`.

Cho hàm một biến $f(x)$, mục tiêu là tìm giá trị của $x$ thỏa phương trình $f(x) = 0$. (Tất nhiên trong trường hợp $f(x)$ là hàm bậc nhất theo x thì không còn gì phải bàn, ở đây ta muốn xét đến trường hợp tổng quát).

Phương pháp Newton xuất phát từ việc xấp xỉ giá trị của đạo hàm tại một điểm:

$$f'(x_1) = \frac{df(x_1)}{dx} \approx \frac{f(x_1) - f(x_2)}{x_1 - x_2}$$

$$\Rightarrow x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} + \frac{f(x_2)}{f'(x_1)}$$

Do ta đang muốn tìm nghiệm $x$ sao cho $f(x) = 0$, do đó với $f(x_2) = 0$ thì phương trình trên trở thành:

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

Tổng quát, phương pháp Newton lặp liên tục để tìm nghiệm ngày càng gần giá trị thực, với công thức nghiệm là:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

với giá trị khởi tạo $x_0$ nào đó.

Trong chương trình trên, $ce$ là vector chứa hệ số của đa thức. Chẳng hạn với đa thức $f(x) = x^5 + 10x^3 - \frac{1}{2}x - 4$ thì $ce = \left[-4, \frac{-1}{2}, 0, 10, 1\right]$. Tham số $res$ là độ lỗi cho phép, chương trình dừng khi sai số nhỏ hơn ngưỡng này. Trong trường hợp không tìm được nghiệm, chương trình dừng sau $maxLoop$ lần lặp.

Sử dụng hàm này, ta có thể tính nhanh nghiệm của phương trình $x^5 - 5 = 0$ (cũng chính là tìm xấp xỉ giá trị $\sqrt[5]{5}$) như sau:

```
nrsolve([-5 0 0 0 0 1], 0.00000000001, 100)
ans =
    1.3797
```