Thuật toán lặp Jacobi

# Thuật toán kiểm tra tính chéo trội hàng

* B1: i = 0
* B2: sum = 0;
* B3: Cho j chạy từ 0 đến n-1, tính sum = sum + abs(a[i][j])
* B4: kiểm tra 2\*abs(a[i][i]) <= sum, nếu đúng thì báo ma trận không chéo trội hàng và kết thúc

Nếu không đúng thì i = i + 1 và quay lại b2

# Thuật toán kiểm tra tính chéo trội cột

* B1: i = 0;
* B2: sum = 0
* B3: cho j chạy từ 0 đến n-1, tính sum = sum + abs(a[j][i])
* B4: kiểm tra 2\*abs(a[i][i]) <= sum, nếu đúng thì báo ma trận không chéo trội cột và kết thúc

Nếu không đúng thì i = i + 1 và quay lại b2

# Thuật toán lặp khi chéo trội hàng

* B1: Cho i chạy từ 0 đến n-1

D[i] = a[i][n] / a[i][i];

B[i][i] = 0;

Cho j chạy từ 0 đến n-1, nếu i khác j thì B[i][j] = -a[i][j]/a[i][i];

* B2: Tính hệ số co q = chuẩn vô cùng của B
* B3: Thực hiện tính x\_(k+1) = B \* x\_k + D trong khi (chuẩn vô cùng) ||x\_(k+1) – x\_k|| > ep\*(1-q)/q

Với ep là sai số cho trước trong đề bài

* B4: in ra x\_(k+1) là nghiệm của hệ phương trình và kết thúc

# Thuật toán lặp khi chéo trội cột

* B1: cho i chạy từ 0 đến n-1

D[i] = a[i][n]

B[i][i] = 0

Cho j chạy từ 0 đến n-1, nếu i khác j B[j][i] = -a[j][i] / a[i][i]

* B2: tính hệ số co q = chuẩn một của B
* B3: thực hiện tính x\_(k+1) = B \* x\_k + D trong khi (chuẩn một) ||x\_(k+1) – x\_k || > ep\*(1-q)/q
* B4: in ra x\_(k+1)

Cho i chạy từ 0 đến n-1, tính nghiem[i] = x\_(k+1)[i] / a[i][i]

* B5: khi đó nghiem là nghiệm của hệ phương trình và kết thúc

# Chương trình chính

* B1: Nhận input A, ep, x0
* B2: kiểm tra tính chéo trội của ma trận A

Nếu chéo trội hàng thì thực hiện lặp jacobi hàng

Nếu chéo trội cột thì thực hiện lặp jacobi cột

Nếu chéo trội cả hàng và cột thì thực hiện lặp jacobi hàng

Nếu không chéo trội kết thúc