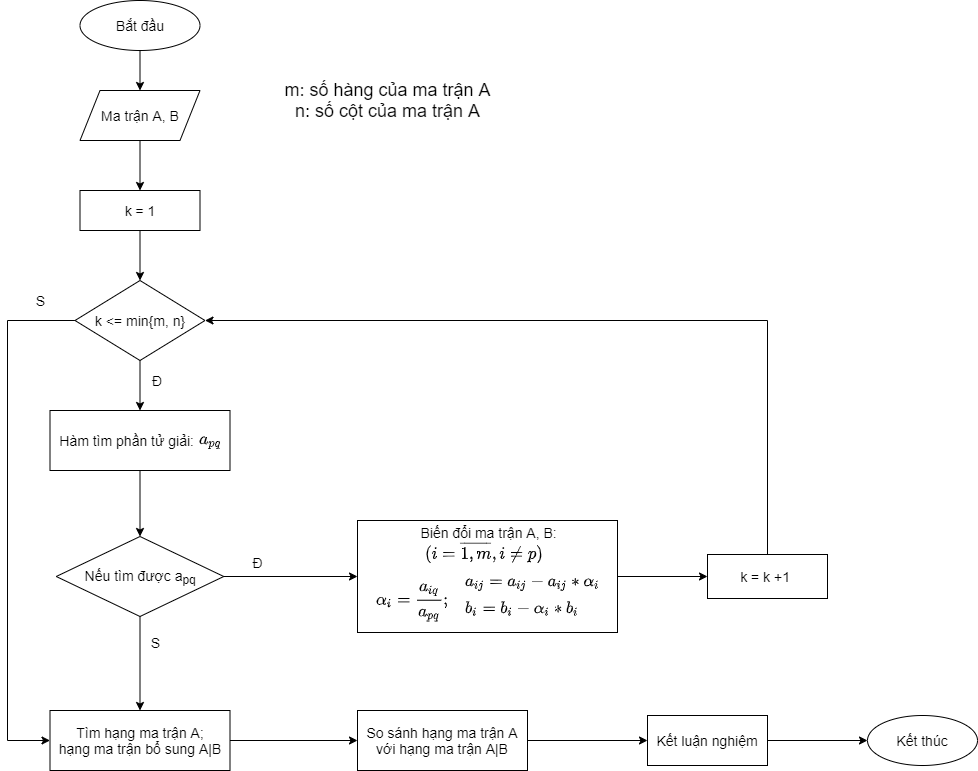
**Phương pháp Gauss – Jordan giải nghiệm đúng hệ phương trình đại số tuyến tính AX = B**

1. Thuật toán tổng thể



1. Thuật toán chi tiết. (Giả mã)

|  |
| --- |
| 1. Định nghĩa ma trận là một bản ghi gồm có:   + hàng: row  + cột: col  + dữ liệu: mảng 2 chiều a[][]   1. Hàm tìm giá trị nhỏ nhất của 2 số a, b   Input: a, b  Output: min của a, b  Function min  if a >= b:  return b  else:  return a   1. Hàm tìm vị trí phần từ giải và lưu vị trí đó vào mảng 1 chiều index[]   Input: ma trận A, mảng chứa vị trí của các phần tử giải: index[],  địa chỉ vị trí hàng giải: p, địa chỉ vị trí cột giải: q  Output: true nếu tìm được vị trí phần tử giải; false nếu không tìm được    Function index\_ele\_sol:  m = số hàng ma trận A  n = số cột ma trận A  max = 0  for i = 0 to m – 1:  check = 0  for k = 0 to n – 1:  if i = index[k]:  check = 1  End for  if check 1:  for j = 0 to n – 1:  if index[j] -1:  if |Aij| = 1:  p = i  q = j  index[q] = p  return true  else:  if |Aij| > max:  max = |Aij|  p = i  q = j  if max != 0:  index[q] = p  return true  else:  return false   1. Hàm tìm nghiệm của hệ phương trình đại số tuyến tính sử dụng pp Gauss – Jordan   Input: Ma trận đầu vào A, ma trận B  Output: kết luận nghiệm và nghiệm của hệ phương trình  Function gauss\_jordan\_method:  rankA = 0, rankAB = 0, count = 0  m = số hàng ma trận A  n = số cột ma trận A  min\_m\_n = min(m, n)  for i = 0 to n – 1:  index[i] = -1  for i = 0 to m – 1:  index\_row[i] = -1  for k = 0 to min\_m\_n:  if index\_ele\_sol(A, index, p, q) = true:  index\_row[p] = q  for i = 0 to m – 1:  if i p:  coeff = Aiq / Apq  for j = 0 to n – 1:  Aij = Aij – coeff \* Apj  Bi,0 = Bi,0 – coeff \* Bp,0  else:  End for  //Tính rank của ma trận A và ma trận bổ sung A|B  for i = 0 to m – 1:  if index\_row[i] -1:  rankA = rankA + 1  rankAB = rankAB + 1  else:  if Bi,0 0:  rankAB = rankAB + 1  //Xét các trường hợp nghiệm của hệ phương trình  if rankA < rankAB:  print “Hệ phương trình vô nghiệm”  kết thúc chương trình  //rankA = rankAB suy ra phương trình có nghiệm  else:  if rankA = n:  print “Hệ phương trình có nghiệm duy nhất. Nghiệm là”  for i = 0 to n – 1:  k = index[i]  print “x = “ Bk,0 / Ak,i  else:  if rankA <= min\_m\_n:  print “Hệ phương trình vô số nghiệm”  print “Phụ thuộc số tham số là: “ n – rankA    //Cột i nào không được chọn là cột giải thì xi là tham số  for i = 0 to n – 1:  if index[i] = -1:  tham\_so[count] = i  count = count + 1  print “Nghiệm của hệ phương trình là: “  for i = 0 to n – 1:  if index[i] = -1:  print “xi là tham số”  else:  k = index[i]  print “xi = ” Bk,0 / Ak,i  for j = 0 to count – 1:  h = tham\_so[j]  print “+“ (- Ak,h / Ak,i) “\*x“ h  print “Một nghiệm của hệ phương trình là: “  for i = 0 to n – 1:  sum = 0  if index[i] = -1:  print “xi = ” 1  else:  k = index[i]  for j = 0 to count – 1:  h = tham\_so[j]  sum = sum – Ak,h  print “xi = ” (Bk,0 + sum)/Ak,i  Kết thúc chương trình |