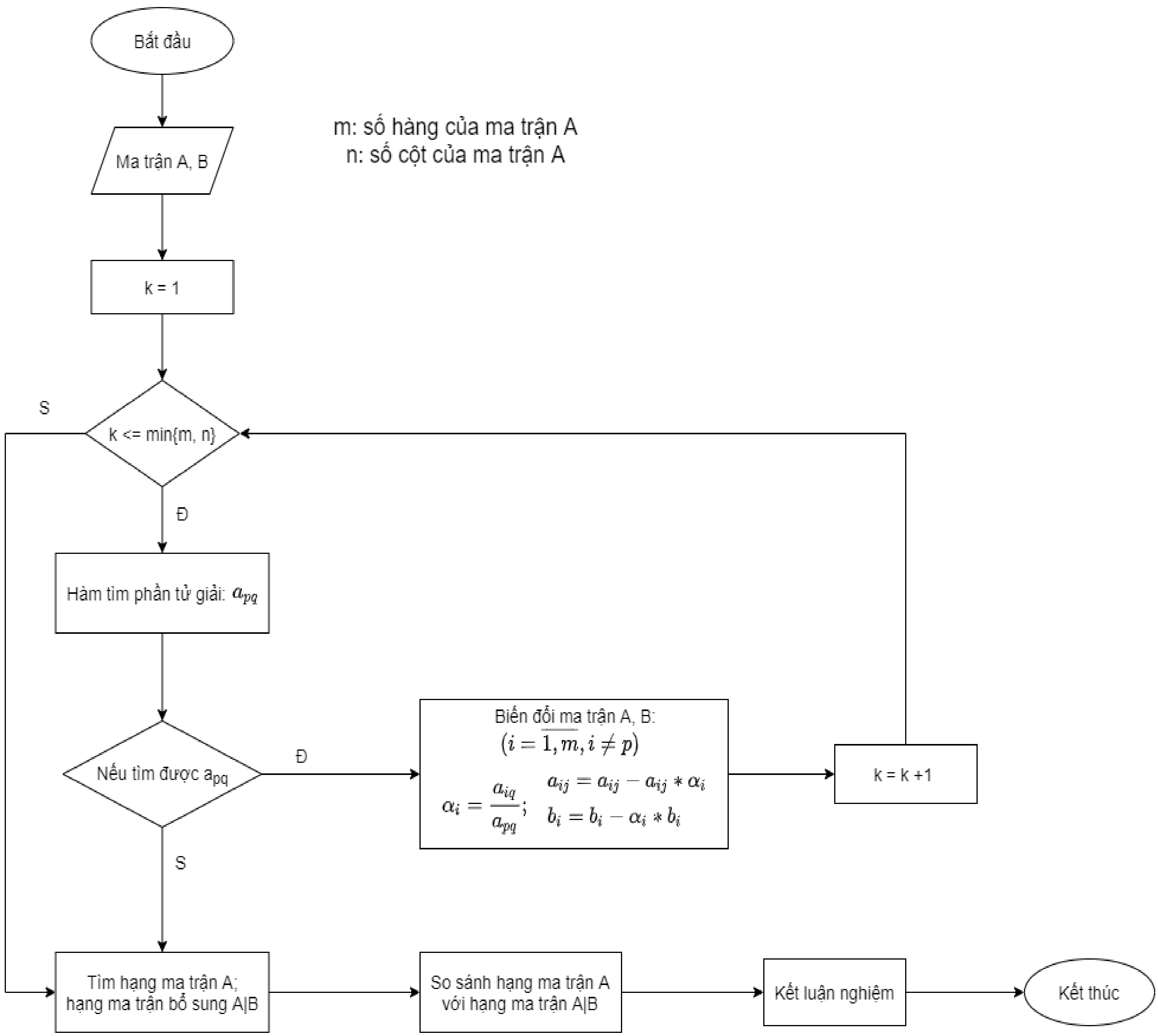
Phương pháp Gauss – Jordan giải nghiệm đúng hệ phương trình đại số tuyến tính AX = B

I. Thuật toán tổng thể



II. 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[][]

2. 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
```

3. Hàm tìm vị trí phần tử 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 |A_ij| = 1:
                         p = i
                         q = j
                         index[q] = p
                         return true
```

```
                        else:
                            if |A_ij| > max:
                                max = |A_ij|
                                p = i
                                q = j
    if max != 0:
        index[q] = p
        return true
    else:
        return false
```

4. 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 = A_iq / A_pq
                    for j = 0 to n – 1:
                        A_ij = A_ij – coeff  * A_pj

                    B_i,0 = B_i,0 – coeff * B_p,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 B_{i,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 = " B_{k,0} / A_{k,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 x_i 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 "x_i = " B_{k,0} / A_{k,i}
                               for j = 0 to count – 1:
                                     h = tham_so[j]
                           print "+" (- A_{k,h} / A_{k,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 – A_{k,h}
                              print “xi = ” (B_{k,0} + sum)/A_{k,i}
Kết thúc chương trình
```

III. Ưu và nhược điểm của phương pháp

1. Ưu điểm.

- Giảm được sai số trong quá trình tính toán khi chia cho số gần 0
- Dễ lập trình tính toán trên máy tính
- Độ chính xác của nghiệm cao
- Giải được các phương trình đại số tuyến tính khi ma trận A bất kì còn với Phương pháp lặp đơn, lặp Jacobi, pp Seidel và Gauss – Seidel điều kiện ma trận A phải là ma trận vuông
- Biểu diễn được nghiệm trong trường hợp vô số nghiệm

2. Nhược điểm

- Độ phức tạp thuật toán lớn, do mỗi lần ta đều phải chọn phần tử giải.

*Chú ý:

- Đối với những ma trận hệ số đơn giản, hoặc có thể coi là “ước”, “bội” của nhau thì kết quả nhận được sẽ hoàn toàn đúng, các phép tính đơn giản sẽ không có ảnh hưởng đến sai số
- Đối với những ma trận cỡ lớn thì nên sử dụng các phương pháp lặp để giải vì tốc độ hội tụ nó sẽ nhanh hơn rất nhiều so với phương pháp Gauss - Jordan

IV. Tóm tắt phương pháp.

1. Điều kiện của phương pháp

- Ma trận A cỡ m,n bất kì

2. Phương pháp Gauss – Jordan để giải hệ phương trình đại số tuyến tính Ax = B.

- Tìm phần tử giải trên toàn bộ ma trận, phần tử giải sau phải khác hàng, khác cột với những phần tử giải khác. Ưu tiên chọn phần tử giải = 1 hoặc – 1, sau đó ưu tiên tìm phần tử có trị tuyệt đối lớn nhất để giảm thiểu sai số khi chia cho số bé.
- Từ phương trình Ax = B => ta sẽ biến đổi trên ma trận bổ sung A|B
- Sau quá trình biến đổi ma trận A|B: những cột mà được chọn là cột giải thì sẽ chỉ còn 1 phần tử giải khác 0 còn các phần tử khác trên cột đó sẽ = 0
- So sánh hạng của ma trận A và ma trận bổ sung A|B
  - Nếu rankA < rankA|B: Kết luận hệ phương trình vô nghiệm
  - Nếu rankA = rankA|B:
    + rankA = số cột ma trận A (hạng bằng số ẩn): Hệ phương trình có nghiệm duy nhất và kết thúc
    + rankA <= min hàng và cột: Hệ phương trình vô số nghiệm, những cột i nào mà không được chọn làm cột giải thì x_i sẽ là nghiệm tham số và các nghiệm x_i còn lại sẽ được biểu diễn thông qua các nghiệm tham số.