

```

1 module subprogs
2   implicit none
3   contains
4     subroutine gauss_jordan(a0, x, b, n)
5       ! -- ガウス・ジョルダン法(部分pivot選択なし) --
6       integer, intent(in) :: n ! 配列の寸法
7       real(8), intent(in) :: a0(n, n), b(n) ! ay = b
8       real(8), intent(out) :: x(n)
9       integer i, k
10      real(8) ar, a(n, n)
11      a(:, :) = a0(:, :) ! a0をaにcopy
12      x(:) = b(:) ! bをxにcopy
13      do k = 1, n
14        if (a(k, k) == 0.0d0) stop 'pivot = 0' ! pivotが0なら停止する
15        ar = 1.0d0 / a(k, k)
16        a(k, k) = 1.0d0
17        a(k, k+1:n) = ar * a(k, k+1:n) ! k行のk+1列からn列にarをかける
18        x(k) = ar * x(k)
19        do i = 1, n
20          if (i /= k) then
21            a(i, k+1:n) = a(i, k+1:n) - a(i, k) * a(k, k+1:n)
22            x(i) = x(i) - a(i, k) * x(k)
23            a(i, k) = 0.0d0
24          endif
25        enddo
26      enddo
27    end subroutine gauss_jordan

```

You, 56 分前 • 6\_1\_3

```

29 subroutine set_random_ad(a, b, x, n)
30   ! nを取得 a,b,xを割付け aとbに乱数を設定
31   real(8), allocatable, intent(out) :: a(:, :), b(:, :), x(:)
32   integer n
33   write(*, '(a)', advance = 'no') ' input n : '
34   read(*, *) n
35   if (n < 1 .or. 100 < n) stop 'n must be 0 < n < 101'
36   allocate (a(n, n), b(n), x(n))
37   call random_number(a)
38   call random_number(b)
39   print *, a
40   print *, b
41 end subroutine set_random_ad
42 end module subprogs

```

```

44 program main
45   use subprogs
46   implicit none
47   real(8), allocatable :: a(:, :), b(:, :), x(:), r(:)
48   integer n
49   call set_random_ad(a, b, x, n)
50   call gauss_jordan(a, x, b, n)
51   allocate (r(n)) ! 残差ベクトルの内積を出力
52   r(:) = b(:) - matmul(a, x)
53   write(*, *) 'Gauss-Jordan error = ', dot_product(r, r)
54   deallocate(a, b, x)
55 end program main

```

Gauss-Jordan法

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned} \quad (6.4)$$

$$\begin{aligned} 1 \cdot x_1 + a'_{12}x_2 + a'_{13}x_3 &= b'_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned} \quad (6.5)$$

$$\begin{aligned} x_1 + a'_{12}x_2 + a'_{13}x_3 &= b'_1 \\ 0 + a'_{22}x_2 + a'_{23}x_3 &= b'_2 \\ 0 + a'_{32}x_2 + a'_{33}x_3 &= b'_3 \end{aligned} \quad (6.6)$$

$$\begin{aligned} x_1 + 0 + a''_{13}x_3 &= b''_1 \\ 0 + x_2 + a''_{23}x_3 &= b''_2 \\ 0 + 0 + a''_{33}x_3 &= b''_3 \end{aligned} \quad (6.7)$$

1行目 →  $a_{11}$ で割る2行目 → 1行目に  $a_{21}$  をかけลบ。  
3行目 → 1行目に  $a_{31}$  をかけลบ。2行目 →  $a'_{22}$  で割る  
3行目 → 2行目に  $a'_{32}$  をかけลบ。作成したプログラム

$$Ax = b$$

係数行列  $A$  と 右辺ベクトル  $b$  は 乱数が設定される。  
次元  $n$  を入力すると  $x$  が求まる。

$A \in \mathbb{R}^{n \times n}$   
係数を判別する  
と = 3

残差の出力。