- pass by reference so any changes to x inside the program would change the argument passed in

```
1 function gs(A,b,x)
2     y=copy(x)
3     for i=1:size(A)[1]
4         t=b[i]
5         for j=1:size(A)[2]
6             if i!=j
7                 t-=A[i,j]*y[j]
8             end
9         end
10        t/=A[i,i]
11        y[i]=t
12    end
13    return y
14 end
```

to prevent overwriting x, because x is an argument and that would be confusing...

subtracts this from the b_i

computes this sum $\sum_{j \neq i} A_{ij} x_j$

$$x_i = \frac{b_i - \sum_{j \neq i} A_{ij} x_j}{A_{ii}}$$

$Ax = b$

$(Ax)_i = b_i$ on components

$$(Ax)_i = \sum_{j=1}^{n} A_{ij} x_j$$

system of linear equations written using the indices...

$$\sum_{j=1}^{n} A_{ij} x_j = b_i$$

Solve for variables of the diagonal

$$A_{ii} x_i + \sum_{j \neq i} A_{ij} x_j = b_i$$

Gauss Sieded Method

$$x_i = \frac{b_i - \sum_{j \neq i} A_{ij} x_j}{A_{ii}} \quad \text{for } i = 1, \ldots, n$$

$$A = \begin{bmatrix} 5 & 2 & 0 & 0 \\ 4 & 5 & 2 & 0 \\ 0 & 3 & 5 & 1 \\ 0 & 0 & 2 & 5 \end{bmatrix}$$

$$x_2 = \frac{b_2 - 4x_1 - 2x_3}{5}$$

doesn't look like a contraction...

```
julia> A=[5 2 0 0;4 5 2 0;0 3 5 1;0 0 2 5]
4×4 Matrix{Int64}:
 5  2  0  0
 4  5  2  0
 0  3  5  1
 0  0  2  5
```

Suppose we cook a proble

so $x = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ is the answer

Let $b = A\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$

```
julia> b=A*ones(4)
4-element Vector{Float64}:
  7.0
 11.0
  9.0
  7.0
```

The answer to $Ax = b$ is $x = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$.

Does Gauss-Seidel converge? YES

```
julia> xn=zeros(4)
       for i=1:30
           xn=gs(A,b,xn)
       end

julia> xn
4-element Vector{Float64}:
 1.000000014967943
 0.9999999776555502
 1.000000015480689
 0.9999999938077245
```

← slowest where it didn't look like a contraction...

STEP 17

THE EIGENVALUE PROBLEM

The power method

Let $A \in \mathbb{R}^{n \times n}$ with distinct eigenvalues λ_i and eigenvectors ξ_i for $i = 1, \ldots, n$.

Trying to find one of the eigenvalues and one eigenvector.

How? As simple as possible...

It's always going to be approximate...

From linear algebra we define the characteristic polynomial

$$\chi_A(\lambda) = \det(A - \lambda I)$$ ← polynomial of degree n

and then solve $\chi_A(\lambda) = 0$ to find the eigenvalues.

roots of a polynomial of degree n generally get n solutions..

To solve for the roots could use Newton's method to find one root λ_1.

Then divide by it $\lambda - \lambda_1 \overline{)\ \chi_A(\lambda)}$

to get a polynomial of degree $n-1$,

Use Newton's method again to find a root λ_2. and so forth..,

Problems:

main problem → ① It's computationally expensive to find $\det(A-\lambda I)$

② Deflation propagates error rather badly..,

↑ fix it by applying Newton's iterations for $\chi_A(\lambda)$ to all the roots found to sharpen them up and make them more accurate.

Application

$$\|A\| = \max\{\|Ax\| : \|x\| \le 1\}$$

$$= \max\{\lambda_i^{1/2} : i = 1, \ldots, n\}$$

↑ here λ_i is the eigenvalues of $B = A^T A$.

For this application I only need the largest eigenvalue of B.

Let $A \in \mathbb{R}^{n \times n}$ with distinct eigenvalues λ_i and eigenvectors ξ_i for $i = 1, \ldots, n$,

Order the eigenvalues in decreasing magnitude

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$$

one more assumption this inequality is strict.

The eigenvalue of greatest magnitude is unique...

Goal: Find that eigenvalue and its eigenvector...

approximate

Suppose ξ_1^* is an approximation of ξ_1

$$A\xi_1^* \approx \lambda_1 \xi_1^*$$

Thus,

$$A\xi_1^* \cdot \xi_1^* \approx \lambda_1 \underbrace{\xi_1^* \cdot \xi_1^*}$$

just a number, so I can divide by it

or

$$\lambda_1 \approx \frac{A\xi_1^* \cdot \xi_1^*}{\xi_1^* \cdot \xi_1^*}.$$

Idea.. multiplying by A stretches a vector in each of the eigenvector directions by λ_i amount. The greatest stretching will be in the ξ_1 direction since λ_1 has greatest magnitude... Repeatedly stretch a vector until this ξ_1 direction dominates...

Specifically: Let $w^{(0)}$ be an approximation of ξ_1

↑ any vector not zero will be fine

$$w^{(1)} = A w^{(0)}$$
$$w^{(2)} = A w^{(1)}$$
$$\vdots$$
$$w^{(i+1)} = A w^{(i)}$$

Thus $\boxed{w^{(i)} \simeq A^i w^{(0)}}$ ← look at this vector... it is approximating an eigenvector for λ_1 more and more each time...,

Since eigenvectors are not unique, I don't expect $w^{(i)}$ to converge. But it will approximate some eigenvector of λ_1 as it stretches around better each time...

Thus, since

$$\lambda_1 \approx \frac{A\xi_1^* \cdot \xi_1^*}{\xi_1^* \cdot \xi_1^*}$$

we expect that $\dfrac{Aw^{(i)} \cdot w^{(i)}}{w^{(i)} \cdot w^{(i)}}$ will converge.