Error control in scientific modelling (MATH 500, Herbst)

Sheet 4: Matrix eigenvalue bounds

(not to be handed in)

```
# Install some packages
begin
    using LinearAlgebra
    using PlutoUI
    using PlutoTeachingTools
    using Plots
end
```

Exercise 1

We are in the setting of Theorem 3.6, i.e.:

Let $\tilde{v}$ be an approximate eigenvector to A, $\|\tilde{v}\| = 1$, $\tilde{\lambda} = \langle \tilde{v}, A\tilde{v} \rangle$ and $r = A\tilde{v} - \tilde{\lambda}\tilde{v}$. Let λ be the eigenvalue closest to $\tilde{\lambda}$, let $\delta = \min_i \{|\lambda_i - \tilde{\lambda}|, \lambda_i \neq \lambda\}$ be the distance from the spectrum and let v be an eigenvector of A associated with λ.

Prove the improved error

$$\sin \theta(\tilde{v}, v) \leq \sqrt{\frac{\|r\|_2^2 - \varepsilon^2}{\delta^2 - \varepsilon^2}},$$

where $\varepsilon = |\lambda - \tilde{\lambda}|$.

Hint: Follow the proof of Theorem 3.6.

Proof:

Following the proof of Theorem 3.6 we can see that:

$$\begin{aligned}||r||_2^2 \geq & \epsilon^2 \cos^2\theta + \delta^2 \sin^2\theta = \\ = & \epsilon^2(1-\sin^2\theta) + \delta^2\sin^2\theta = \\ = & (\delta^2-\epsilon^2)\sin^2\theta + \epsilon^2.\end{aligned}$$

From isolating $\sin\theta$:

$$\sin^2\theta \leq \frac{||r||_2^2 - \epsilon^2}{\delta^2 - \epsilon^2}$$

Therefore,

$$\sin\theta \leq \sqrt{\frac{||r||_2^2 - \epsilon^2}{\delta^2 - \epsilon^2}}$$

□

Exercise 2

Consider the near-diagonal matrix

$$\mathrm{M} = \begin{bmatrix} 1.0 & 0.001 & 0.099 & 0.099 & 0.0 \\ 0.001 & 2.0 & -0.051 & 0.0 & -0.1 \\ 0.099 & -0.051 & 3.0 & 0.1 & 0.05 \\ 0.099 & 0.0 & 0.1 & 4.0 & 0.0 \\ 0.0 & -0.1 & 0.05 & 0.0 & 5.0 \end{bmatrix}$$

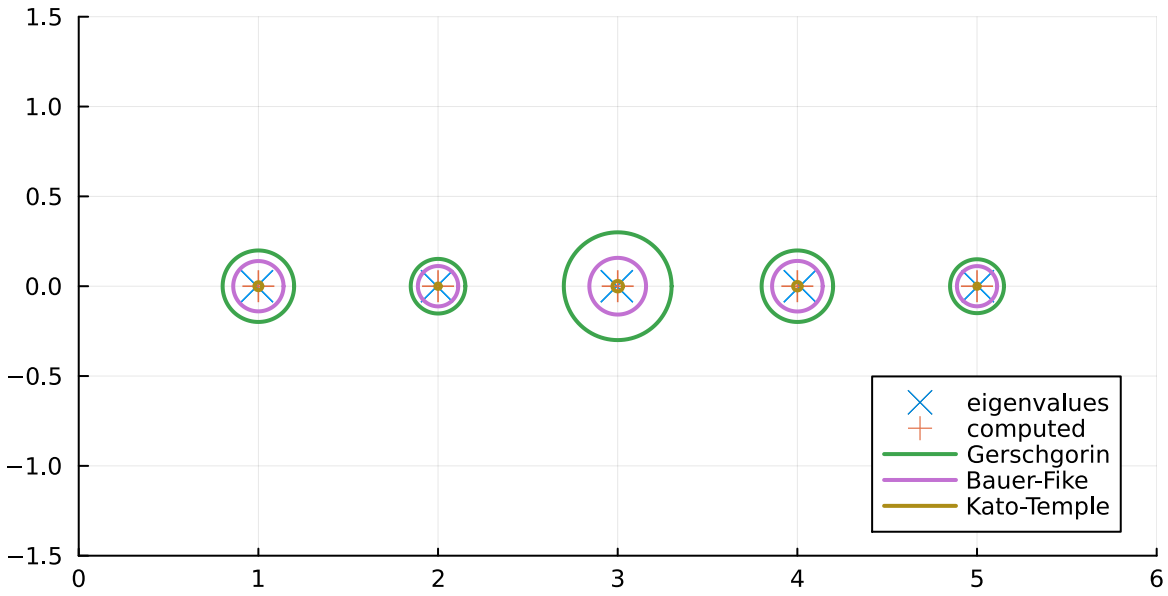
with some Sliders to tune the off-diagonal elements:

```
M = [
    1.0  M12  M13  M14  0.0;
    M12  2.0  M23  0.0  -0.10;
    M13  M23  3.0  0.1  0.05;
    M14  0.0  0.1  4.0  0.0;
    0.0  -0.1  0.05  0.0  5.0
];
```

- M12 = 0.001
- M13 = 0.099
- M14 = 0.099
- M23 = -0.051

Final plot

This section compiles a nice plot of the errors and Gerschgorin disks as we go along, see below for the tasks.



- Plot Geschgorin disks: ☑
- Plot Bauer-Fike estimate: ☑
- Plot Kato-Temple estimate: ☑

Gerschgorin disks

First we want to add a plot of the Gerschgorin disks to the plot above. Add code to compute the centres:

```
geschgorin_centres =  [1.0, 2.0, 3.0, 4.0, 5.0]
```

```
1 geschgorin_centres = diag(M)
```

Now add code to compute the radii:

```
geschgorin_radii = [0.199, 0.152, 0.3, 0.199, 0.15]
```

```
geschgorin_radii = [sum(abs(M[i, j]) for j in 1:size(M, 1) if i != j) for i in 1:size(M, 1)]
```

Approximate eigenpairs and residuals

Clearly M is nearly a diagonal matrix for most choices of the parameters. We thus take the diagonal of M as our estimate for the eigenvalues:

```
computed_eigenvalues = [1.0, 2.0, 3.0, 4.0, 5.0]
```

```
computed_eigenvalues = diag(M)
```

A natural question is: How accurate is this estimate?

For comparison we compute the exact eigenvalues. Add code to do so:

```
exact_eigenvalues = [0.992159, 1.99427, 2.99511, 4.01379, 5.00467]
```

```
exact_eigenvalues = eigen(M).values
```

Apart from this computation we will assume from here on that we know nothing about the exact spectrum and use the techniques developed in the lecture to estimate the error of `computed_eigenvalues`.

Both Bauer-Fike and Kato-Temple need the **residuals**. Thus we first compute these.

Add code to do so, keeping in mind that the "computed" eigvectors in our example are just the unit vectors.

```
computed_eigenvectors =
[[1.0, 0.0, 0.0, 0.0, 0.0], [0.0, 1.0, 0.0, 0.0, 0.0], [0.0, 0.0, 1.0, 0.0, 0.0], [0.0, 0.0
```

```
computed_eigenvectors = [Vector([i == j ? 1.0 : 0.0 for j in 1:size(M, 1)]) for i in 1:size(M, 1)]
```

```
residuals =
  [[0.0, 0.001, 0.099, 0.099, 0.0], [0.001, 0.0, -0.051, 0.0, -0.1], [0.099, -0.051, 0.0, 0.1
```

```
1 residuals = M * computed_eigenvectors .- computed_eigenvalues .* computed_eigenvectors
```

A posteriori error estimation

With the residuals in place we can proceed to compute the error estimate following the **Bauer-Fike theorem**.

The following code should produce a vector of error estimates, one for each eigenvalue

```
error_Bauer_Fike =   [0.140011, 0.112259, 0.157804, 0.140716, 0.111803]
```

```
1 error_Bauer_Fike = norm.(residuals)
```

Next we tackle the **Kato-Temple bound**. For this we need as the trickier ingredient a *lower bound* on the spectral distance δ between the approximate eigenvalue $\tilde{\lambda}_i$ we have computed and the closest exact eigenvalue λ_j with $j \neq i$ (i.e. the closest exact eigenvalue not counting the one we currently approximate).

For this we will employ the lower-bound trick mentioned in the lecture. In particular for the spectral distance of the first eigenvalue we obtain

$$\delta_1 = |\tilde{\lambda}_1 - \lambda_2| \geq |\tilde{\lambda}_1 - \tilde{\lambda}_2| - \|r_2\|.$$

For the second and higher eigenpairs we need to keep in mind that there are two neighbouring eigenvalues and we need to take the *smallest* distance, thus we have

$$\begin{aligned}
\delta_{2l} &= |\tilde{\lambda}_2 - \lambda_1| \geq |\tilde{\lambda}_2 - \tilde{\lambda}_1| - \|r_1\| \\
\delta_{2r} &= |\tilde{\lambda}_2 - \lambda_3| \geq |\tilde{\lambda}_2 - \tilde{\lambda}_3| - \|r_3\| \\
\delta_2 &= \mathbf{min}(\delta_{2l}, \delta_{2r})
\end{aligned}$$

An additional complication is that as the residuals $\|r_i\|$ get too large and δ_2 becomes negative. The interpretion in this case is that it is no longer guaranteed that our numerical scheme provides the correct ordering of the approximate eigenvalues $\tilde{\lambda}_i$ versus λ_i. The Kato-Temple bound in this case is not applicable, which we fix by explicitly setting δ_i to zero (and thus causing an infinite Kato-Temple bound). We thus need an additional step where we set

$$\delta_i \leftarrow \mathbf{max}(0, \delta_i) \qquad \forall i.$$

Code up your algorithm for computing the δ_i again as a vector of floats, one for each eigenvalue.

```
begin
    δ_list = [0. for i in 1:5]
    δ_list[1] = abs(computed_eigenvalues[1] - computed_eigenvalues[2]) -
    error_Bauer_Fike[2]

    δ_list[5] = abs(computed_eigenvalues[5] - computed_eigenvalues[4]) -
    error_Bauer_Fike[4]

    for i in 2:4
    δ_left = abs(computed_eigenvalues[i] - computed_eigenvalues[i-1]) -
    error_Bauer_Fike[i-1]
    δ_right = abs(computed_eigenvalues[i] - computed_eigenvalues[i+1]) -
    error_Bauer_Fike[i+1]

    δ_list[i] = min(δ_left, δ_right)

    end
end
```

```
δ =  [0.887741, 0.842196, 0.859284, 0.842196, 0.859284]
```

```
δ = [max(0, i) for i in δ_list]
```

With δ in place, the computation of the Kato-Temple bound is simple. Code it up here:

```
error_Kato_Temple =  [0.0220819, 0.0149633, 0.0289799, 0.0235111, 0.014547]
```

```
error_Kato_Temple = error_Bauer_Fike .^2 ./ δ
```

Playing with the parameters

Once you have everything coded up and your plot above shows the Gerschgorin disks, the Bauer-Fike bound and the Kato-Temple bound, play a bit with the sliders to make the off-diagonal elements larger and smaller.

The tighter the bounds, the better — in this case they plainly provide a better way to estimate errors. In which regime is which of the bounds the most favourable?

Answer:

- The Bauer-Fike bounds seem to always be a better approximation than the Gerschgorin disks.
- The Bauer-Fike bounds exhibit better stability compared to the Kato-Temple bounds. When we start with off-diagonal values close to zero, Kato-Temple bounds are very strict. But as we increase the absolute values of the off-diagonal elements, Bauer-Fike bounds handle these changes better, while Kato-Temple bounds can become impractical due to significant growth.