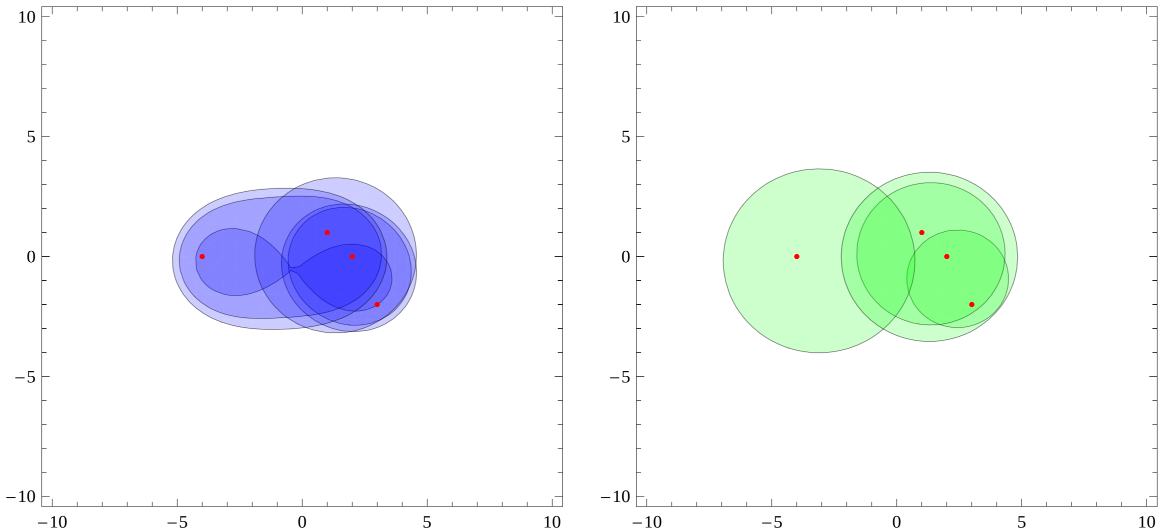
ACM 11: Brauer Ovals

This is one possible topic for the ACM11 Mathematica project. Difficulty rating: easy to medium. 100 points. 11/27/08.

In this project we will visualize Gerschgorin's circles and Brauer's ovals.



Recall Gerschgorin's theorem about the location of the eigenvalues of a matrix:

Theorem 1 (Gerschgorin's theorem) *The eigenvalues of a matrix* $A = (a_{ij}) \in \mathbb{C}^{n\times n}$, $n \geq 2$, *lie in the union* $G = \bigcup_{i=1}^{n} G_i$ *of the Gerschgorin circles*

$$G_i = \{z \in \mathbb{C} : |z - a_{ii}| \leq R_i(A)\}, \quad i = 1, \ldots, n$$

where $R_i(A) = \sum_{j\neq i}^{n} |a_{ij}|$.

We can also restrict the location of the eigenvalues by considering the $\frac{n(n-1)}{2}$ ovals of Cassini:

$$C_{ij}(A) = \{z \in \mathbb{C} : |z - a_{ii}||z - a_{jj}| \leq R_i(A)R_j(A)\}, \quad i \neq j,\ i, j = 1, \ldots, n.$$

Theorem 2 (Brauer's theorem) *The eigenvalues of a matrix* $A = (a_{ij}) \in \mathbb{C}^{n\times n}$, $n \geq 2$, *are contained in the domain*

$$C(A) = \bigcup_{\substack{i,j=1 \\ i\neq j}}^{n} C_{ij}(A).$$

It turns out that $C(A) \subseteq \bigcup_{i=1}^{n} G_i$, so in fact the Brauer ovals potentially give us more information on the location of the eigenvalues. For a more nuanced discussion, see `http://eom.springer.de/G/g130060.htm`

We will visualize the Brauer ovals and Gerschgorin circles for a few matrices. To do this, we will use `RegionPlot` to plot the individual Gershgorin circles and Brauer ovals, and `GraphicsGrid` to juxtapose them.

(a) We will need the row sums R_i, so define a function `rowSums` that takes a matrix A as its argument and returns a list of the appropriate length whose i-th entry is R_i.

(b) For our visualization we will identify $\mathbb{C}$ with the plane, so write a function `complexToPoint` that takes a complex number as its argument and returns the corresponding point in $\mathbb{R}^2$.

(c) Write a function `gerschgorin` that takes A and i as its arguments and returns a plot of the region G_i. Use `Norm`, `complexToPoint`, and `rowSums` as necessary to express the equation (x, y) has to satisfy to be in G_i. For simplicity, plot the region over the rectange $(x, y) \in [-10, 10] \times [-10, 10]$. Use the `PlotStyle` directive to ensure the region is colored green with an opacity of 0.2.

(d) Write a function `cassini` that takes A and i,j as its arguments and returns a plot of the region C_{ij}. Use `Norm`, `complexToPoint`, and `rowSums` as necessary to express the equation (x, y) has to satisfy to be in C_{ij}. For simplicity, plot the region over the rectange $(x, y) \in [-10, 10] \times [-10, 10]$. Use the `PlotStyle` directive to ensure the region is colored blue with an opacity of 0.2.

(e) Write a function `allCassini` that takes A as its argument and generates a table of all the plots of the C_{ij}, $i \neq j$ (note that $C_{ij} = C_{ji}$). Likewise, write a function `allGerschgorin` that takes A as its argument and generates a table of all the plots of the G_i.

(f) We need to be able to construct matrices with given eigenvalues, so write a function `constructMatrix` that takes a list of complex numbers as its argument and returns a matrix with those numbers as its eigenvalues. To do this, remember that if D is a diagonal matrix and P is an orthogonal matrix, then PDP^{-1} is a matrix which has the diagonal elements of D as its eigenvalues. To construct an appropriate P, use `RandomReal` to generate a matrix with entries in the range $[-1, 1]$ and then `Orthogonalize` it; to get P^{-1}, use `Inverse`. You may also find `DiagonalMatrix` useful. To ensure reproducable results, seed the random number generator with 2008 before generating P.

(g) Finally, write the function `visualize` which takes a list of complex numbers as an argument, constructs a matrix A with those numbers as its eigenvalues, and shows the Brauer ovals of A side-by-side with its Gershgorin circles. First let `cassiniG` be the graphic obtained by showing all the C_{ij} plots along with red points of `PointSize[.01]` indicating the location of the eigenvalues. Next let `gerschgorinG` be the graphic obtained by showing all the G_i plots along with red points of `PointSize[.01]` indicating the location of the eigenvalues. Use `GraphicsGrid` to show `cassiniG` and `gerschgorinG` juxtaposed, with the option `ImageSize->Large`.

(h) Put all of the above definitions in one cell. In two cells beneath, give examples of Brauer ovals and Gerschgorin circles using `visualize`. Under these two cells, give another example using the eigenvalues $\{-4, 3 - 2i, 2, 1 + i\}$.

Document your program with text cells and/or comments.