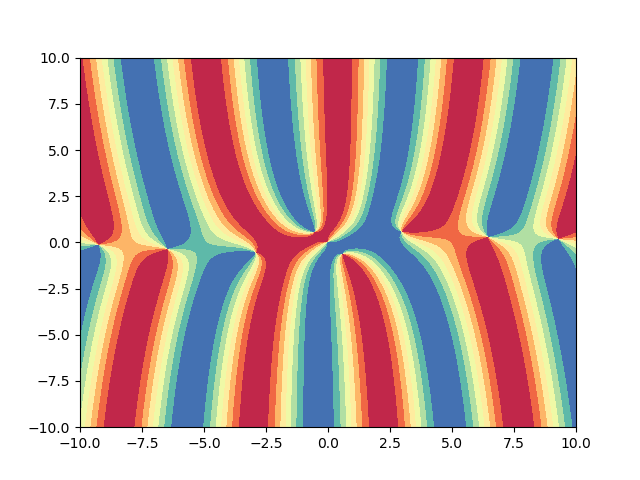
1. Finding zeroes of delta(lambda)
   1. Using contour plots of the argument of delta (zeroes are where the contour changes rapidly): Difficult with contour plots like this



* 1. Find zeroes manually: TBD (read the paper “On the zeros of exponential sums and integrals” (good when |lambda| is large, i.e., asymptotically, could make a good initial starting point for some other root-finding algorithms)?)
     1. First idea was to approximate delta using Chebyshev polynomials, but finding zeroes of Chebyshev approximations requires complex root-finding
     2. Separate real and imaginary: lambda = x+iy and solve for Real(x,y)=0 and Imaginary(x,y)=0 simultaneously <https://docs.sympy.org/latest/_modules/sympy/functions/elementary/complexes.html>

<https://stackoverflow.com/questions/25927501/separating-real-and-imaginary-parts-using-sympy>

https://stackoverflow.com/questions/46103490/getting-real-and-imaginary-parts-of-complex-function-in-matlab

<https://math.stackexchange.com/questions/1033430/complex-root-finding> (Newton Raphson requires very good initial guesses)

* + 1. To be proved: Zeroes of an analytic function are exactly discontinuities in the sine of its argument (maybe stronger)
       1. Algorithms to find discontinuities of functions in two variables?
       2. Zero is just the pole of the reciprocal of the function (read abt multiplicity of poles/zeros): Go around a zero, if it crosses the red and blue and yellow once, then it’s first order; if twice, then it’s second order. Statements about zeros may be proved using statements about poles.
    2. Assuming we can find the order of the pole as above: Go around a contour to see if it crosses 2\*n pieces of yellow – 0, n pieces of red - -1, and n pieces of blue – 1 (assuming sine is used); if it does, shrink the contour so that it still crosses the above regions; if not, then the contour does not enclose a zero. If the contour cannot be shrunk anymore, then it encloses n zeroes of first order; if it can be shrunk all the way, then it encloses one zero of the nth order.

1. The “f” in (2.15a) and (3.12) is as in (2.12b), right?
   1. (2.12b) and (2.12c) mean that the spatial q(x,0) = f and temporal q(, t) both satisfy the homogeneous boundary conditions. To test the transform pairs, do I just choose an f (Chebyshev approximated) that satisfies the homogeneous boundary conditions Uf=0 and set it to be q(x,0)? (q, f are IBVP parameters)

Yes, and more generally, need to find a way to generate well-posed IBVP to test the implementation.

1. In (2.15a), to characterize “\lambda \in \Gamma”, I need to find out how to check if a given point is on the contour characterized by a list of points (it may not be one of the points), right?
   1. **But the contour is approximated, how would this affect the determination of whether \lambda is on the contour? (mention that I had to ask this question)**
      1. No need, see equation (2.18), integrating over the contours. The interpretation of (3.12) should change accordingly.
2. Documentation
   1. Platform & format: IJulia (explanation in markdown and examples in code)
   2. Content: Not just how to use but also what exactly the code is doing
3. Semester plan:
   1. Comments about last semester? (later)
   2. Independent reading?
      1. Contribute to capstone: separate, independent chapter (2nd version of the method)
   3. March 22nd initial thesis, April 23rd final thesis. Package & documentation should be ready before March 22nd.
   4. Integrator?
   5. Focus on the implementation (without proof)