Iteration of complex cubics

This notebook provides some functions to investigate the complex iteration of monic, depressed cubics, i.e. polynomials of the form:

$$f_{a,b}(z) = z^3 - 3\,a^2\,z + b.$$

This function is called *monic* since the coefficient of z^3 is one. It is called *depressed* since the coefficient of z^2 is zero. The constant term and the coefficient of z are arbitrary. The coefficient of z is chosen to have the form $-3\,a^2$so that the cubic has the critical points $\pm a$.

Conjugation and the arbitrary cubic

As it turns out, *any* cubic is conjugate to some $f_{a,b}$. Thus, we are essentially studying the family of *all* cubics. To see this write down the *arbitrary* cubic in the form

$$g(z) = c_3\,z^3 + c_2\,z^2 + c_1\,z + c_0.$$

We need to find a function $\varphi(z) = m\,z + d$ such that $\varphi(g(z)) = f_{a,b}(\varphi(z))$. We can set up the equations we need to solve like so:

```
phi[z_] = m * z + d;
f[a_, b_][z_] = z^3 - 3 a^2 * z + b;
g[z_]  = c3 * z^3 + c2 * z^2 + c1 * z + c0;
eqs = CoefficientList[phi[g[z]], z] == CoefficientList[f[a, b][phi[z]], z]
```

$\{d + c0\,m,\ c1\,m,\ c2\,m,\ c3\,m\} == \{b - 3\,a^2\,d + d^3,\ -3\,a^2\,m + 3\,d^2\,m,\ 3\,d\,m^2,\ m^3\}$

And, now, let’s try to solve them

```
sol = Solve[eqs, {a, b, d, m}] // Last
```

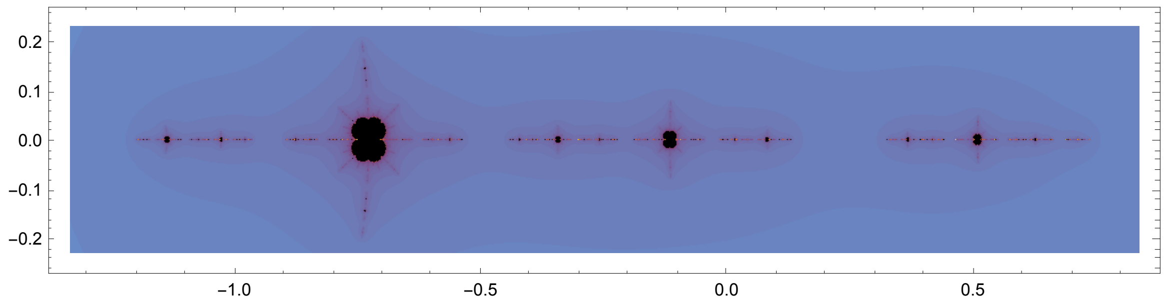
Solve: Equations may not give solutions for all "solve" variables.

$$\left\{a \to \frac{\sqrt{c2^2 - 3\,c1\,c3}}{3\,\sqrt{c3}},\ b \to \frac{2\,c2^3 + 9\,c2\,c3 - 9\,c1\,c2\,c3 + 27\,c0\,c3^2}{27\,c3^{3/2}},\ d \to \frac{c2}{3\,\sqrt{c3}},\ m \to \sqrt{c3}\right\}$$

I would interpret this to mean that $\varphi(z) = \sqrt{c_3}\,z + c_2\big/\left(3\,\sqrt{c_3}\right)$ conjugates $f_{a,b}$ to g when a and b are given by the solution above. Evidently, the correspondence is not one-to-one, as in the quadratic case for $z^2 + c$.

To illustrate, suppose that $g(z) = 4\,z^3 + 3\,z^2 - 2\,z - 1$. Here’s the Julia set:

```
g[z_] = 4 z^3 + 3 z^2 - 2 z - 1;
JuliaSetPlot[g[z], z, ImageSize → 600, ImageResolution → 1200]
```



Thus, $c_3 = 4$ and $c_2 = 3$. If we set

$$\varphi(z) = \sqrt{4}\, z + \frac{3}{3\sqrt{4}} = 2z + 1/2$$

and compute $f(z) = \varphi \circ g \circ \varphi^{-1}(z)$, we should obtain a cubic of the form $f_{a,b}$ with a geometrically similar Julia set. Let’s try!

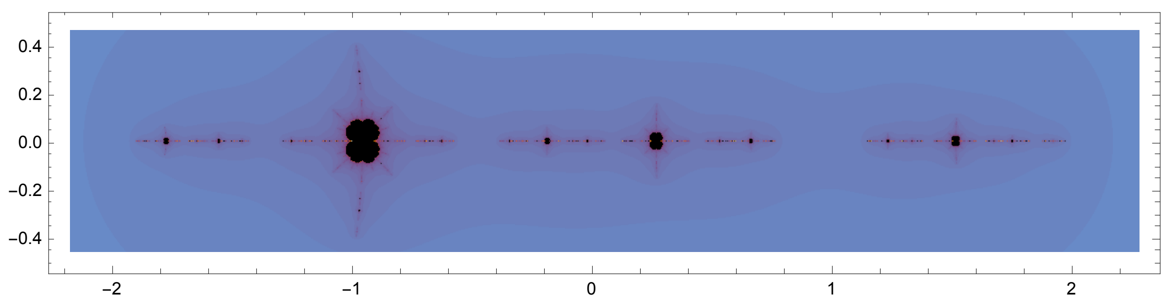
```
phi[z_] = 2 z + 1 / 2;
Phi[z_] = w /. First[Solve[phi[w] == z, w]]
```

$$\frac{1}{4}(-1 + 2 z)$$

```
f[z_] = phi[g[Phi[z]]] // Expand
```

$$-\frac{1}{4} - \frac{11 z}{4} + z^3$$

```
JuliaSetPlot[f[z], z, ImageSize → 600, ImageResolution → 1200]
```



Investigating the parameter space

The initialization cells at the bottom of the notebook defines several functions for investigating the parameter space for $f_{a,b}$.

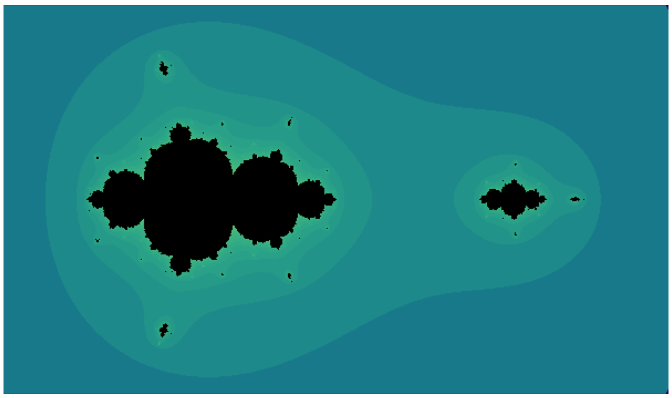
Julia sets for $f_{a,b}$

The function **cubicJuliaPic** simply generates the Julia set. It should be called like so:

```
cubicJuliaPic[a_?NumericQ, b_?NumericQ, bail_Integer, resolution_Integer]
```

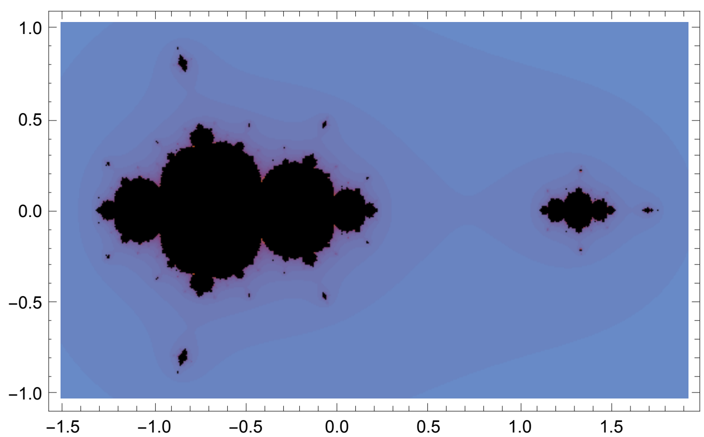
Here's the filled Julia set when $a = 1/\sqrt{2}$ and $b = -1$.

```
cubicJuliaPic[-1 / Sqrt[2], -1, 100, 600]
```



Of course that should generate the Julia set for $z^3 - 3\left(1/\sqrt{2}\right)^2 z - 1$.

```
JuliaSetPlot[z^3 - (3 / 2) z - 1, z]
```



There's a dynamic version that accepts the same arguments but includes a locator allowing you to visualize the orbit starting from where the user clicks. You can play with this, if you want:

```
dynamicCubicJuliaPic[-1 / Sqrt[2], -1, 100, 600]
```

Parameter pics for $f_{a,b}$

Recall that the critical points of $f_{a,b}(z) = z^3 - 3\,a^2\,z + b$ are $\pm a$. To study the parameter space, we should partition the four dimensional space $\mathbb{C}^2$ into the sets of points where

1. *a* and −*a* both converge to the same finite orbit (deep purple)
2. *a* and −*a* both converge but to different finite orbits (green)
3. *a* escapes to ∞ but −*a* converges to a finite orbit (red)
4. −*a* escapes to ∞ but *a* converges to a finite orbit (blue)
5. *a* and −*a* both escape to ∞ (light)

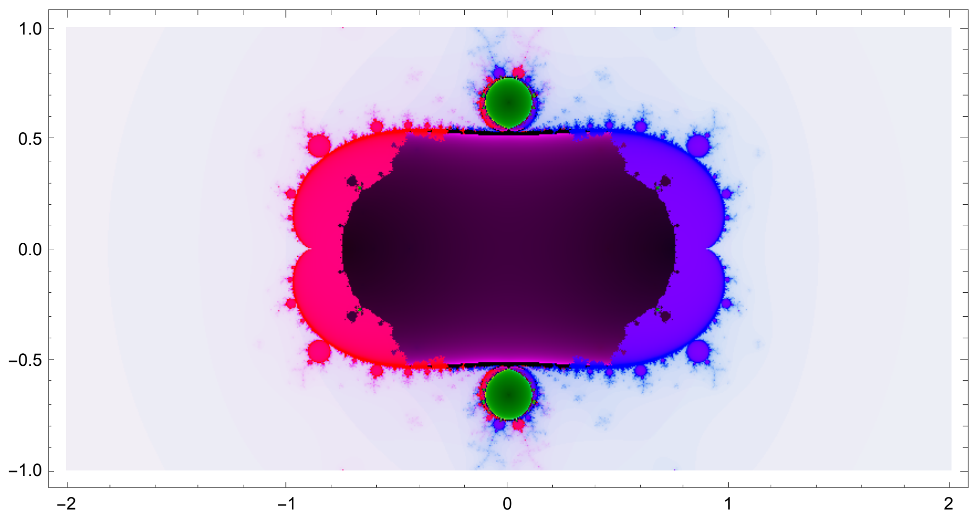
Black means that the classification was unsuccessful.

To visualize this, we can fix one of *a* or *b* and let the other vary inside a complex rectangle - effectively generating two-dimensional slice of the four dimensional object. The `parameterPic` function defined below does exactly this. It can be called in one of two ways - depending on whether you'd like to fix *a* or *b*.

```
parameterPic["a", a_?NumericQ, bMin_Complex,
 bMax_Complex, res_Real, graphicsOpts : OptionsPattern[]]
parameterPic["b", b_?NumericQ, aMin_Complex, aMax_Complex,
 res_Real, graphicsOpts : OptionsPattern[]]
```

For example:

```
parameterPic["a", 1 / 2, -2 - I, 2 + I, 0.005,
 Frame → True, ImageSize → 500]
```



I guess that's the parameter pic for $f_{1/2,b}(z) = z^3 - \frac{3}{4} z + b$, where *b* lies in the rectangle above. There's also a dynamic version that allows you to explore how Julia sets vary as the free parameter changes.

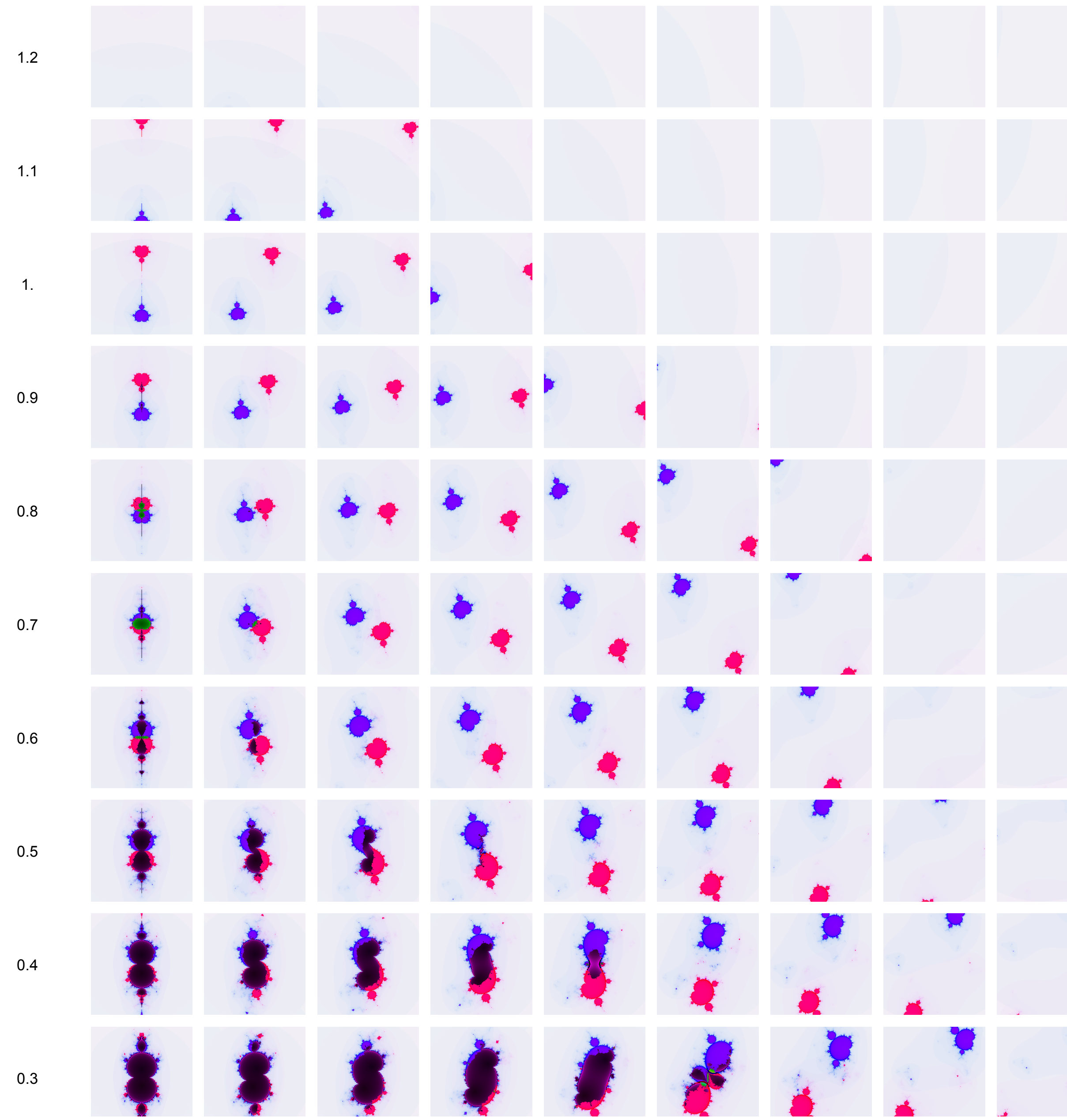
```
dynamicParameterPic["a", 1 / 2, -2 - I, 2 + I, 0.005]
```

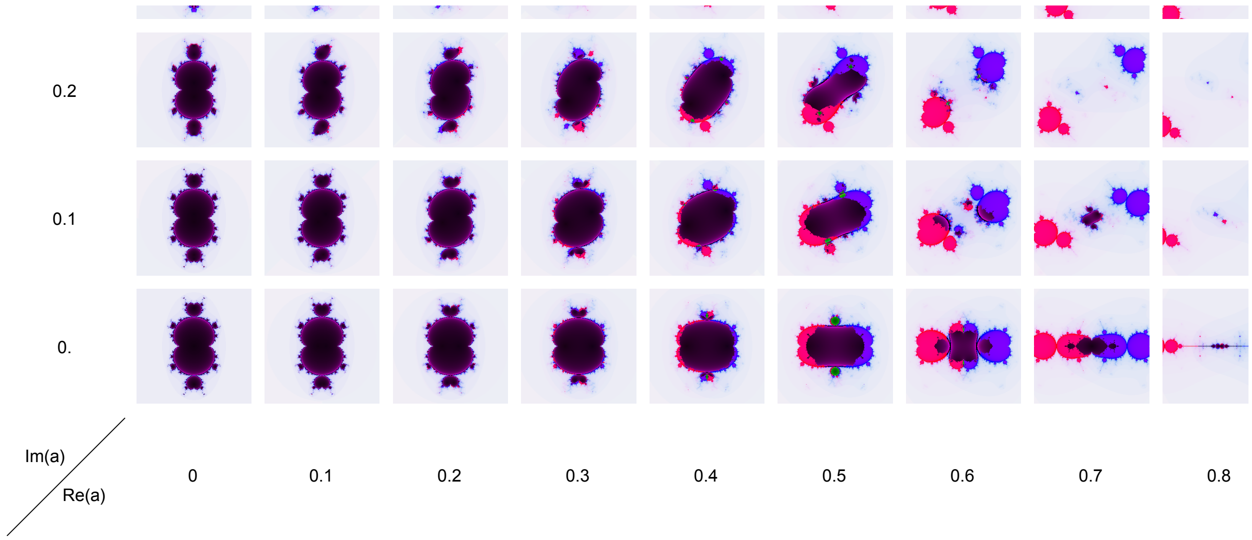
The 4D set

Here's are some pictures of grids of slices to, hopefully, illustrate the whole 4D set. These take about a half minute on my machine.

```
pics = Table[
   parameterPic["a", ax + ay * I, -1.5 - 1.5 I, 1.5 + 1.5 I, 0.01, ImageSize → 60],
   {ay, 1.2, 0, -0.1}, {ax, 0, 1, 0.1}]; // AbsoluteTiming
corner = Graphics[{Line[{{-1, -1}, {2, 2}}],
   Text["Re(a)", {1, 0}],
   Text["Im(a)", {0, 1}]},
  PlotRange → {{0, 1}, {0, 1}},
  PlotRangePadding → 1];
GraphicsGrid[Join[Table[Prepend[pics[[i]], 0.1 (13 - i)], {i, 1, Length[pics]}],
 {Flatten[{corner, Chop[0.1 (# - 1)] & /@ Range[Length[First[pics]]]}]}]]
```

{65.8843, Null}





Initialization