

Fractals from the Lorenz Equations

My Solutions >

Determine the fractal that arises from using Newton's method to compute the fixed-point solutions of the Lorenz equations. Use the parameter values $r = 28$, $\sigma = 10$ and $\beta = 8/3$. Initial values (x_0, z_0) are taken on a grid in the x - z plane with always $y_0 = 3\sqrt{2}$. For assessment purposes, the computational grid and the graphics code will be given in the Learner Template. To pass the assessment, every pixel in your figure needs to be colored correctly.

(Hint: Some grid points may require as many as 33 Newton iterations to converge while others may require as few as three. Unfortunately, if you uniformly use 33 Newton iterations at every grid point, the MATLAB Grader may time out. You can accelerate your code by using a **while** loop instead of a **for** loop.)

Script ?

Save Reset MATLAB Documentation (https://www.mathworks.com/help/)

```
1 r=28; sigma=10; beta=8/3;
2 x1=0; y1=0; z1=0;
3 x2=sqrt(beta*(r-1)); y2=sqrt(beta*(r-1)); z2=r-1;
4 x3=-sqrt(beta*(r-1)); y3=-sqrt(beta*(r-1)); z3=r-1;
5 nx=500; nz=500;
6 xmin=-40; xmax=40; zmin=-40; zmax=40;
7 x_grid=linspace(xmin,xmax,nx); z_grid=linspace(zmin,zmax,nz);
8 [X,Z]=meshgrid(x_grid,z_grid);
9
10 % Write Newton's method using every grid point as the initial condition
11 % Perform enough iterations that every initial condition converges to a root
12 % Save the x-values of the converged roots in the matrix X
13 % To pass the assessment, every pixel in the figure must be correctly colored
14
15 %!!!!!!!!! Set initial value y=3*sqrt(2) for all values (x,z) on the grid !!!!!!!!!!
16
17 %size(X)
18 maxiter = 0;
19 for kk = 1:nz
20     for nn = 1:nx
21         x = X(kk,nn);
22         z = Z(kk,nn);
23         y=3*sqrt(2);
24         RelTol=1.e-06; AbsTol=1.e-09;
25         error=Inf;
26         iter = 1;
27         while error > AbsTol %max(RelTol*max(abs([x,y,z])),AbsTol)
28             J= [-sigma sigma 0; r-z -1 -x; y x -beta]; % DEFINE THE JACOBIAN MATRIX
29             rhs = -[sigma*(y-x); x*(r-z)-y; x*y-beta*z]; % DEFINE THE RIGHT-HAND SIDE
30             delta_xyz=J\rhs;
31             x = x + delta_xyz(1);
32             y = y + delta_xyz(2);
33             z = z + delta_xyz(3);
34             error=max(abs(delta_xyz));
35             iter = iter + 1;
36         end
37         if iter > maxiter
38             maxiter = iter;
39         end
40         X(kk, nn) = x;
41     end
42 end
43
44 maxiter
45
46 eps=1.e-03;
47 X1 = abs(X-x1) < eps; X2 = abs(X-x2) < eps; X3 = abs(X-x3) < eps;
48 X4 = ~(X1+X2+X3);
49 figure;
50 map = [1 0 0; 0 1 0; 0 0 1; 0 0 0]; colormap(map); %[red;green;blue;black]
51 X=(X1+2*X2+3*X3+4*X4);
52 image([xmin xmax], [zmin zmax], X); set(gca,'YDir','normal');
53 xlabel('$x$', 'Interpreter', 'latex', 'FontSize',14);
54 ylabel('$z$', 'Interpreter', 'latex', 'FontSize',14);
55 title('Fractal from the Lorenz Equations', 'Interpreter', 'latex','FontSize', 16)
56
```

Run Script ?

Assessment: All Tests Passed

Submit ?

- Test red pixel values
- Test green pixel values
- Test blue pixel values

Output

maxiter =

34

