# Dynamical system formulation

The system is formulated in polar/spherical coordinates, thus cartesian data needs to be converted.
2D Dynamical System in polar coordinates $[\rho, \theta]$, where $\rho$ is the polar radius and $\theta$ is the polar angle:

$$\mathbf{f}(\rho,\theta) = \begin{cases} \dot{\rho} = -\sqrt{2M}(\rho - \rho_0) \\ \dot{\theta} = R\mathrm{e}^{-4M^2(\rho-\rho_0)^2} \end{cases}$$

3D Dynamical System in spherical coordinates $[\rho, \phi, \theta]$, where $\rho$ is the spherical radius, $\phi$ is the elevation angle (between $x_3$ and the $x_1$-$x_2$ plane) and $\theta$ is the azimuth angle (between $x_1$ and $x_2$):

$$\mathbf{f}(\rho,\phi,\theta) = \begin{cases} \dot{\rho} = -\sqrt{2M}(\rho - \rho_0) \\ \dot{\phi} = -\sqrt{2M}(\phi) \\ \dot{\theta} = R\mathrm{e}^{-4M^2(\rho-\rho_0)^2} \end{cases}$$

Parameters to be optimized (in red above):

- $\rho_o$ : target radius; $\rho_o > 0$ for limit cycles, $\rho_o$ = 0 for attractors. 1-dimensional.
- $M$ : "mass"; $M > 0$; relates to speed to reach the target radius (higher speed for higher values) and to how fast the system "reacts" to the presence of the cycle/focus (rotation starts tight to target for higher values, while it starts rotating further away for lower values of $M$). 1-dimensional.
- $R$ : rotation speed (only used for $\dot{\theta}$). 1-dimensional.

## Optimization

Least Squares minimizations:

$$\begin{aligned} &1. \quad \min \| -\sqrt{2M}(\rho_{data} - \rho_0) - \dot{\rho}_{data}\|^2 \\ &2. \quad \min \|R\,\mathrm{e}^{-4M^2(\rho_{data}-\rho_0)^2} - \dot{\theta}_{data}\|^2 \\ &\quad\quad u.c. \quad \rho_0 \geq 0, \quad M > 0 \end{aligned}$$

## Scaling, shifting and rotation

For limit sets not centered to the origin, rotated on the plane/in space and/or with different scaling for various dimensions, transformations need to be applied **before** conversion to polar/spherical coordinates.
We aim to retrieve the *"original"* limit cycle, i.e. the circular, 0-centered cycle on the $x_1$-$x_2$ plane. We assume the *"original"* data was first rotated by a rotation matrix $Rrot$ (NxN orthogonal matrix, where N is the number of dimensions, either 2 or 3), then scaled by $\frac{1}{a}$ ($a$ is a N-D vector) and finally shifted by $x_0$ (N-D vector). We thus plan to recover the 0-centered, unscaled and unrotated data $x_{cycle}$ by transforming the raw cartesian data $x_{data}$, as:

$$x_{cycle} = a * (Rrot' * (x_{data} + x_0))$$

*(Note that the multiplication by the inverse of $Rrot$ is a left multiplication and $x_{data}$ is expressed as a column vector. Since the matrix is orthogonal, $Rrot'$ is equal to the transpose.)*
We then convert $x_{cycle}$ to polar/spherical coordinates, instead of $x_{data}$, and we will need to invert the equation above after calculating the DS to convert the final motion to the shifted, scaled & rotated space where $x_{data}$ lives:

$$x_{data} = (Rrot * \frac{x_{cycle}}{a}) - x_0$$

*(Note that the multiplication by $Rrot$ is a left multiplication and $x_{data}$ is expressed as a column vector.)*
Added parameters to the optimization (included during conversion to polar/spherical coordinates):

- $a$ : scaling along dimensions (for ellipses); $a \geq 1$[1]. $N$-dimensional, with $N$ being the number of dimensions of the data.
- $x_0$ : shift of the origin in space. $N$-dimensional.

Not included in the optimization above, but rather found in advance by singular value decomposition (for details please refer to the paper):

- $Rrot$ : rotation matrix to bring the cycle on the $x_1$-$x_2$ plane. $N$x$N$-dimensional.

[1]The largest ellipse's axis will correspond to $\rho_0$ and the shorter axis will be $\frac{\rho_0}{a_i}$, $i = 1 : N$ being the corresponding (rotated) coordinate.