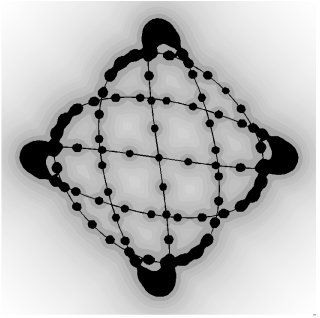
AIM
Network



Non-linear adjoint looping, a brief tutorial

Matthew Juniper, with lots of help from Peter Schmid, together with Patrick Huerre's notes on optimal control

1 Lagrange optimization

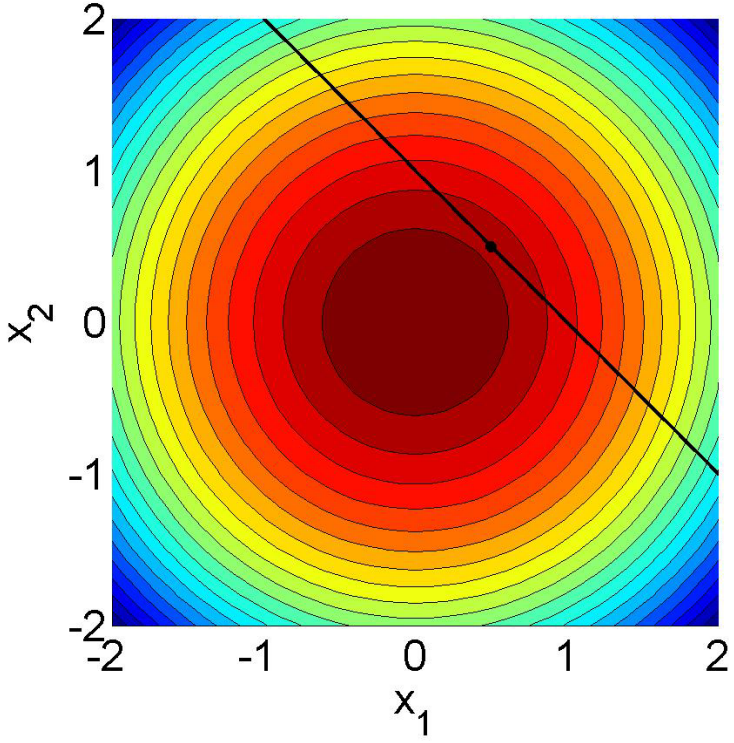
We want to optimize some cost function, $\mathcal{J}(x_1, x_2)$, subject to some constraint, $F(x_1, x_2) = 0$. We do this by defining a new function, $\mathcal{L}(x_1, x_2) = \mathcal{J} - aF$, where a is an unknown constant called a Lagrange multiplier. At optimality, $\partial\mathcal{L}/\partial x_1 = 0$, $\partial\mathcal{L}/\partial x_2 = 0$ and $\partial\mathcal{L}/\partial a = 0$.

e.g. maximize

$$\mathcal{J} \equiv 1 - x_1^2 - x_2^2$$

subject to the constraint

$$F \equiv 1 - x_1 - x_2 = 0.$$



We start with a simple example of Lagrange optimization

e.g. maximize

$$\mathcal{J} \equiv 1 - x_1^2 - x_2^2$$

subject to the constraint

$$F \equiv 1 - x_1 - x_2 = 0.$$

$$\begin{aligned}
\mathcal{L} &\equiv \mathcal{J} - aF = 1 - x_1^2 - x_2^2 - a + ax_1 + ax_2, && (1)\\
\frac{\partial \mathcal{L}}{\partial x_1} &= -2x_1 + a = 0, && (2)\\
\frac{\partial \mathcal{L}}{\partial x_2} &= -2x_2 + a = 0, && (3)\\
\frac{\partial \mathcal{L}}{\partial a} &= 1 - x_1 - x_2 = 0. && (4)
\end{aligned}$$

From (2) and (3), we have $x_1 = x_2$ and from (4), which is just the original constraint, we have $x_1 = x_2 = 1/2$. We do not need to calculate a.

Exercise 1: Use Lagrange optimization to calculate the minimum surface area of a closed cylindrical can whose volume is V. Compare the result with one calculated via a different method.

We start with a simple example of Lagrange optimization

e.g. maximize

$$\mathcal{J} \equiv 1 - x_1^2 - x_2^2$$

subject to the constraint

$$F \equiv 1 - x_1 - x_2 = 0.$$

$$\begin{aligned}
\mathcal{L} &\equiv \mathcal{J} - aF = 1 - x_1^2 - x_2^2 - a + ax_1 + ax_2, && (1)\\
\frac{\partial \mathcal{L}}{\partial x_1} &= -2x_1 + a = 0, && (2)\\
\frac{\partial \mathcal{L}}{\partial x_2} &= -2x_2 + a = 0, && (3)\\
\frac{\partial \mathcal{L}}{\partial a} &= 1 - x_1 - x_2 = 0. && (4)
\end{aligned}$$

From (2) and (3), we have $x_1 = x_2$ and from (4), which is just the original constraint, we have $x_1 = x_2 = 1/2$. We do not need to calculate a.

Now let us consider Lagrange optimization when x is a function. In a moment, we will try to find the initial state that gives the highest growth after a given time, T.

2 Lagrange optimization of linear ODEs

Let us consider a state vector $\mathbf{x} \equiv (x_1, x_2)^T$ whose time evolution is governed by the matrix $\mathbf{M}$:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{M}\mathbf{x}, \tag{5}$$

$$\mathbf{M} \equiv \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}. \tag{6}$$

Exercise 2: (a) For

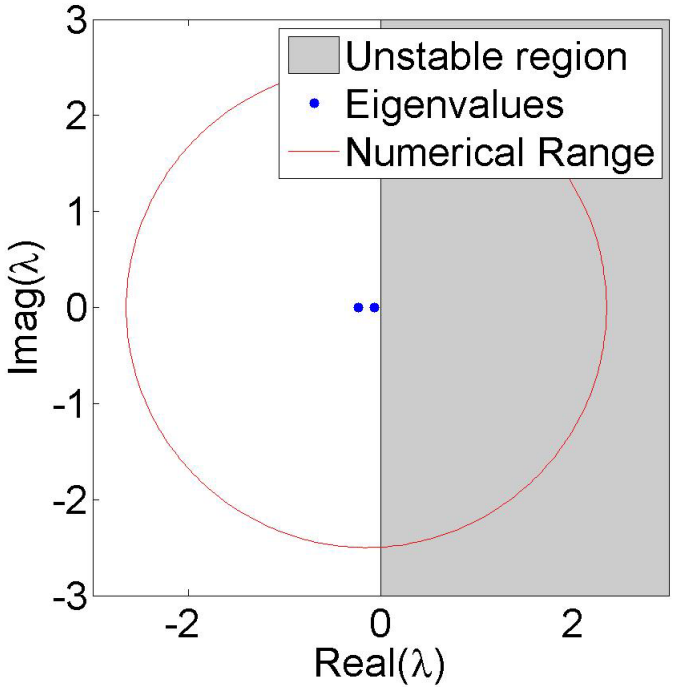
$$\mathbf{M} = \begin{pmatrix} -0.1 & 0 \\ 0 & -0.2 \end{pmatrix},$$

write a `Matlab` script that evaluates $\mathbf{x}(t)$ given $\mathbf{x_0}$, using the matrix exponential (`expm`). Check your answer against one evaluated with a simple exponential (`exp`). (b) Repeat for $m_{12} = n$, for general n. (n represents the degree of non-normality.) (c) Setting $n = 5$, write a script to plot the norm of $\mathbf{x}(t)$ as a function of t, for general $\mathbf{x_0}$. (The norm of $\mathbf{x}$ is defined as $||\mathbf{x}|| \equiv \sqrt{x_1^2 + x_2^2}$.) Repeat for different values of n. (d) Setting $n = 5$, plot $||\mathbf{x}(t)||$ as a function of t for different $\mathbf{x_0}$, keeping $||\mathbf{x_0}||$ constant. (e) Use the SVD to find the initial state, $\mathbf{x_0}$, that maximizes $||\mathbf{x}(T)||$ for $T = 5$. The growth $||\mathbf{x}(T)||/||\mathbf{x_0}||$ at this $\mathbf{x_0}$ is given the label $G(T)$. (f) Plot $G(T)$ for varying T and find its maximum value. This is given the label $G_{max}(T)$ and is the maximum possible transient growth for this matrix $\mathbf{M}$.

First, we will examine the behaviour of a simple non-normal matrix.

$$\mathbf{M} \equiv \begin{pmatrix} -0.1 & 5 \\ 0.001 & -0.2 \end{pmatrix}$$

```
M = [-0.1 5 ; 0.001 -0.2];
lam = eig(M);
plot(real(lam),imag(lam))
NumR = NumRange(M,300);
plot(real(NumR),imag(NumR),'r-')
```



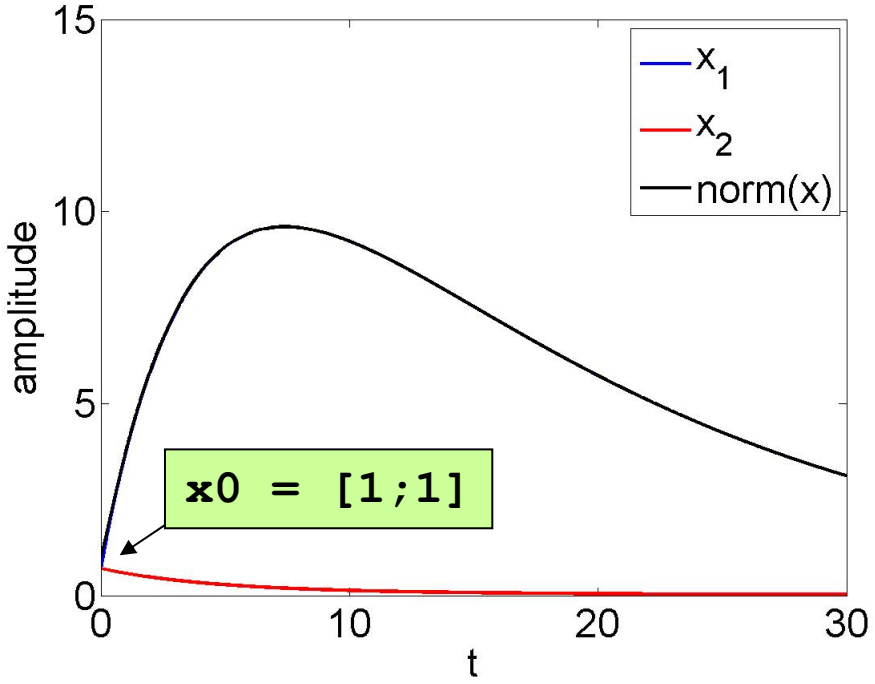
If $\mathbf{M}$ has negative eigenvalues then $||\mathbf{x}|| \to 0$ as $t \to \infty$ but if $\mathbf{M}$ is also non-normal then transient growth is possible at intermediate times. For this linear system we can find the initial state $\mathbf{x_0}$ that gives maximum growth of $||\mathbf{x}||$ after time T with the Singular Value Decomposition (SVD). In Matlab we would type

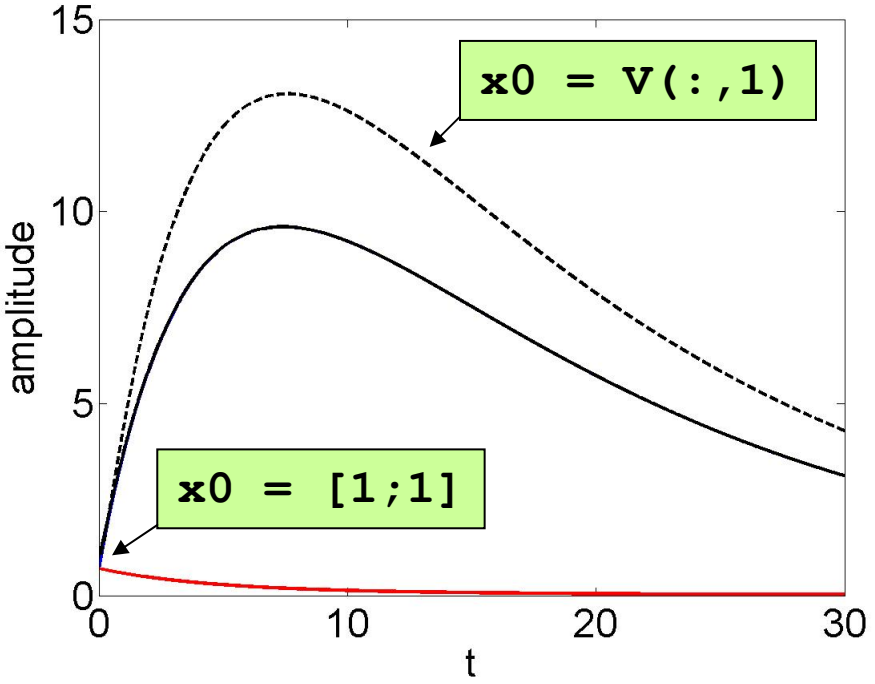
```
[U,S,V] = svd(expm(M*T)); x0 = V(:,1).
```

We can find the optimal initial condition with the Singular Value Decomposition.

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{M}\mathbf{x},$$

```
% First order Euler time-march
for n = 1:floor(T/dt)-1
    dXdt = M * X(:,n);
    X(:,n+1) = X(:,n) + dt * dXdt;
end
[U,S,V] = svd(expm(M*T));
```





We can also find the optimal initial condition via adjoint looping and an optimization routine. The adjoint looping algorithm finds the gradient of L w.r.t. initial conditions, $\mathbf{x_0}$.

$$\frac{\mathrm{d}x_1}{\mathrm{d}t} = m_{11}x_1 + m_{12}x_2, \tag{7}$$

$$\frac{\mathrm{d}x_2}{\mathrm{d}t} = m_{21}x_1 + m_{22}x_2, \tag{8}$$

which can be written as two constraints:

$$F_1 \equiv \frac{\mathrm{d}x_1}{\mathrm{d}t} - m_{11}x_1 - m_{12}x_2 = 0, \quad t \in [0, T], \tag{9}$$

$$F_2 \equiv \frac{\mathrm{d}x_2}{\mathrm{d}t} - m_{21}x_1 - m_{22}x_2 = 0, \quad t \in [0, T]. \tag{10}$$

Similarly, we define the initial condition, $\mathbf{x_0} \equiv (x_{10}, x_{20})$, formally with two constraints:

$$G_1 \equiv x_1(0) - x_{10} = 0, \quad t = 0, \tag{11}$$

$$G_2 \equiv x_2(0) - x_{20} = 0, \quad t = 0. \tag{12}$$

optional

We can also find the optimal initial condition via adjoint looping and an optimization routine. The adjoint looping algorithm finds the gradient of L w.r.t. initial conditions, $\mathbf{x_0}$.

Also, we define an inner product:

$$\langle p(t), q(t) \rangle \equiv \frac{1}{T} \int_0^T pq \, \mathrm{d}t. \tag{13}$$

We will keep the same cost functional that we had for the SVD:

$$\mathcal{J} \equiv \frac{||\mathbf{x}(T)||^2}{||\mathbf{x_0}||^2} = \frac{x_1^2(T) + x_2^2(T)}{x_{10}^2 + x_{20}^2}. \tag{14}$$

We can still use Lagrange optimization but this time the Lagrange multipliers a_1 and a_2 must be functions rather than variables:

$$\mathcal{L} \equiv \mathcal{J} - \langle a_1(t), F_1(t) \rangle - \langle a_2(t), F_2(t) \rangle - b_1 G_1 - b_2 G_2, \tag{15}$$

functions of t

We can also find the optimal initial condition via adjoint looping and an optimization routine. The adjoint looping algorithm finds the gradient of L w.r.t. initial conditions, $\mathbf{x}_0$.

$$\langle p(t), q(t) \rangle \equiv \frac{1}{T} \int_0^T pq \, \mathrm{d}t. \tag{13}$$

$$\mathcal{L} \equiv \mathcal{J} - \langle a_1(t), F_1(t) \rangle - \langle a_2(t), F_2(t) \rangle - b_1 G_1 - b_2 G_2, \tag{15}$$

reminder

and we must also define what we mean by $\partial \mathcal{L} / \partial x$ when $x(t)$ is a function:

$$\left\langle \frac{\partial \mathcal{L}}{\partial x}, \delta x \right\rangle \equiv \epsilon \underset{\rightarrow}{\lim} 0 \, \frac{\mathcal{L}(x + \epsilon \delta x) - \mathcal{L}(x)}{\epsilon}, \tag{16}$$

where $\delta x(t)$ is an arbitrary perturbation to $x(t)$. We see from (13,15) that all variations with respect to the Lagrange multipliers are zero if $F_1 = F_2 = 0$ for $t \in [0, T]$ and if $G_1 = G_2 = 0$ at $t = 0$.

We can also find the optimal initial condition via adjoint looping and an optimization routine. The adjoint looping algorithm finds the gradient of L w.r.t. initial conditions, $\mathbf{x_0}$.

$$F_1 \equiv \frac{\mathrm{d}x_1}{\mathrm{d}t} - m_{11}x_1 - m_{12}x_2 = 0, \quad t \in [0, T], \qquad (9)$$

} reminder

In order to evaluate $\partial\mathcal{L}/\partial x_1$, $\partial\mathcal{L}/\partial x_2$, $\partial\mathcal{L}/\partial x_{10}$ and $\partial\mathcal{L}/\partial x_{20}$, it helps to re-arrange (15) so that x_1, x_2, x_{10} and x_{20} appear explicitly. We do this term by term:

$$\langle a_1, F_1 \rangle = \langle a_1, \frac{\mathrm{d}x_1}{\mathrm{d}t} \rangle - \langle a_1, m_{11}x_1 \rangle - \langle a_1, m_{12}x_2 \rangle. \qquad (17)$$

The last two terms are easy to re-arrange:

$$-\langle a_1, m_{11}x_1 \rangle - \langle a_1, m_{12}x_2 \rangle = -\langle x_1, m_{11}a_1 \rangle - \langle x_2, m_{12}a_1 \rangle. \quad (18)$$

The first term can be integrated by parts:

$$\langle a_1, \frac{\mathrm{d}x_1}{\mathrm{d}t} \rangle = \frac{1}{T} \int_0^T a_1 \frac{\mathrm{d}x_1}{\mathrm{d}t} \mathrm{d}t \qquad (19)$$

$$= \frac{1}{T} \left(a_1(T)x_1(T) - a_1(0)x_1(0) \right) - \frac{1}{T} \int_0^T x_1 \frac{\mathrm{d}a_1}{\mathrm{d}t} \mathrm{d}t. \quad (20)$$

We can also find the optimal initial condition via adjoint looping and an optimization routine. The adjoint looping algorithm finds the gradient of L w.r.t. initial conditions, $\mathbf{x}_0$.

We re-arrange $\langle a_2, F_2 \rangle$ in the same way. Putting these together we have:

$$\begin{aligned}
\mathcal{L} &= \frac{x_1^2(T) + x_2^2(T)}{x_{10}^2 + x_{20}^2} - \langle x_1, F_1^+ \rangle - \langle x_2, F_2^+ \rangle \dots \\
&\dots \quad -b_1(x_1(0) - x_{10}) - b_2(x_2(0) - x_{20}) \dots \\
&\dots \quad -\frac{1}{T}\left(a_1(T)x_1(T) - a_1(0)x_1(0)\right) \dots \\
&\dots \quad -\frac{1}{T}\left(a_2(T)x_2(T) - a_2(0)x_2(0)\right), \qquad (21)
\end{aligned}$$

where

$$F_1^+ \equiv -\frac{\mathrm{d}a_1}{\mathrm{d}t} - m_{11}a_1 - m_{21}a_2, \qquad (22)$$

$$F_2^+ \equiv -\frac{\mathrm{d}a_2}{\mathrm{d}t} - m_{12}a_1 - m_{22}a_2. \qquad (23)$$

We can also find the optimal initial condition via adjoint looping and an optimization routine. The adjoint looping algorithm finds the gradient of L w.r.t. initial conditions, $\mathbf{x_0}$.

Now we can evaluate $\partial\mathcal{L}/\partial x_1$, $\partial\mathcal{L}/\partial x_2$, $\partial\mathcal{L}/\partial x_{10}$ and $\partial\mathcal{L}/\partial x_{20}$ more easily:

$$\begin{aligned}
\langle\frac{\partial\mathcal{L}}{\partial x_1},\delta x_1\rangle &= \frac{2x_1(T)\delta x_1(T)}{x_{10}^2+x_{20}^2}-\langle\delta x_1,F_1^+\rangle-b_1\delta x_1(0)\ldots \\
&\ldots\quad -\frac{1}{T}a_1(T)\delta x_1(T)+\frac{1}{T}a_1(0)\delta x_1(0) \qquad (24)\\
\langle\frac{\partial\mathcal{L}}{\partial x_2},\delta x_2\rangle &= \frac{2x_2(T)\delta x_2(T)}{x_{10}^2+x_{20}^2}-\langle\delta x_2,F_2^+\rangle-b_2\delta x_2(0)\ldots \\
&\ldots\quad -\frac{1}{T}a_2(T)\delta x_2(T)+\frac{1}{T}a_2(0)\delta x_2(0) \qquad (25)\\
\frac{\partial\mathcal{L}}{\partial x_{10}}\delta x_{10} &= -2x_{10}\frac{x_1^2(T)+x_2^2(T)}{(x_{10}^2+x_{20}^2)^2}\delta x_{10}+b_1\delta x_{10} \qquad (26)\\
\frac{\partial\mathcal{L}}{\partial x_{20}}\delta x_{20} &= -2x_{20}\frac{x_1^2(T)+x_2^2(T)}{(x_{10}^2+x_{20}^2)^2}\delta x_{20}+b_2\delta x_{20} \qquad (27)
\end{aligned}$$

We can also find the optimal initial condition via adjoint looping and an optimization routine. The adjoint looping algorithm finds the gradient of L w.r.t. initial conditions, $\mathbf{x_0}$.

Now we can evaluate $\partial\mathcal{L}/\partial x_1$, $\partial\mathcal{L}/\partial x_2$, $\partial\mathcal{L}/\partial x_{10}$ and $\partial\mathcal{L}/\partial x_{20}$ more easily:

$$\langle \frac{\partial \mathcal{L}}{\partial x_1}, \delta x_1 \rangle = \frac{2x_1(T)\delta x_1(T)}{x_{10}^2 + x_{20}^2} - \langle \delta x_1, F_1^+ \rangle - b_1 \delta x_1(0) \ldots$$

$$\ldots \quad -\frac{1}{T} a_1(T)\delta x_1(T) + \frac{1}{T} a_1(0)\delta x_1(0) \qquad (24)$$

$$\langle \frac{\partial \mathcal{L}}{\partial x_2}, \delta x_2 \rangle = \frac{2x_2(T)\delta x_2(T)}{x_{10}^2 + x_{20}^2} - \langle \delta x_2, F_2^+ \rangle - b_2 \delta x_2(0) \ldots$$

$$\ldots \quad -\frac{1}{T} a_2(T)\delta x_2(T) + \frac{1}{T} a_2(0)\delta x_2(0) \qquad (25)$$

$$F_1^+ = 0, \quad t \in (0, T), \qquad (28)$$

$$F_2^+ = 0, \quad t \in (0, T). \qquad (29)$$

We can also find the optimal initial condition via adjoint looping and an optimization routine. The adjoint looping algorithm finds the gradient of L w.r.t. initial conditions, $\mathbf{x}_0$.

Now we can evaluate $\partial\mathcal{L}/\partial x_1$, $\partial\mathcal{L}/\partial x_2$, $\partial\mathcal{L}/\partial x_{10}$ and $\partial\mathcal{L}/\partial x_{20}$ more easily:

$$\langle \frac{\partial \mathcal{L}}{\partial x_1}, \delta x_1 \rangle = \frac{2x_1(T)\delta x_1(T)}{x_{10}^2 + x_{20}^2} - \langle \delta x_1, F_1^+ \rangle - b_1 \delta x_1(0) \ldots$$

$$\ldots - \frac{1}{T} a_1(T)\delta x_1(T) + \frac{1}{T} a_1(0)\delta x_1(0) \qquad (24)$$

$$\langle \frac{\partial \mathcal{L}}{\partial x_2}, \delta x_2 \rangle = \frac{2x_2(T)\delta x_2(T)}{x_{10}^2 + x_{20}^2} - \langle \delta x_2, F_2^+ \rangle - b_2 \delta x_2(0) \ldots$$

$$\ldots - \frac{1}{T} a_2(T)\delta x_2(T) + \frac{1}{T} a_2(0)\delta x_2(0) \qquad (25)$$

$$\frac{2x_1(T)}{x_{10}^2 + x_{20}^2} - \frac{1}{T} a_1(T) = 0, \quad t = T, \qquad (30)$$

$$\frac{2x_2(T)}{x_{10}^2 + x_{20}^2} - \frac{1}{T} a_2(T) = 0, \quad t = T. \qquad (31)$$

We can also find the optimal initial condition via adjoint looping and an optimization routine. The adjoint looping algorithm finds the gradient of L w.r.t. initial conditions, $\mathbf{x}_0$.

Now we can evaluate $\partial\mathcal{L}/\partial x_1$, $\partial\mathcal{L}/\partial x_2$, $\partial\mathcal{L}/\partial x_{10}$ and $\partial\mathcal{L}/\partial x_{20}$ more easily:

$$\langle \frac{\partial \mathcal{L}}{\partial x_1}, \delta x_1 \rangle = \frac{2x_1(T)\delta x_1(T)}{x_{10}^2 + x_{20}^2} - \langle \delta x_1, F_1^+ \rangle - b_1 \delta x_1(0) \ldots$$

$$\ldots - \frac{1}{T} a_1(T)\delta x_1(T) + \frac{1}{T} a_1(0)\delta x_1(0) \quad (24)$$

$$\langle \frac{\partial \mathcal{L}}{\partial x_2}, \delta x_2 \rangle = \frac{2x_2(T)\delta x_2(T)}{x_{10}^2 + x_{20}^2} - \langle \delta x_2, F_2^+ \rangle - b_2 \delta x_2(0) \ldots$$

$$\ldots - \frac{1}{T} a_2(T)\delta x_2(T) + \frac{1}{T} a_2(0)\delta x_2(0) \quad (25)$$

$$-b_1 + \frac{1}{T} a_1(0) = 0, \quad t = 0, \quad (32)$$

$$-b_2 + \frac{1}{T} a_2(0) = 0, \quad t = 0. \quad (33)$$

We can also find the optimal initial condition via adjoint looping and an optimization routine. The adjoint looping algorithm finds the gradient of L w.r.t. initial conditions, $\mathbf{x_0}$.

Now we can evaluate $\partial\mathcal{L}/\partial x_1$, $\partial\mathcal{L}/\partial x_2$, $\partial\mathcal{L}/\partial x_{10}$ and $\partial\mathcal{L}/\partial x_{20}$ more easily:

$$\langle\frac{\partial\mathcal{L}}{\partial x_1},\delta x_1\rangle = \frac{2x_1(T)\delta x_1(T)}{x_{10}^2+x_{20}^2} - \langle\delta x_1, F_1^+\rangle - b_1\delta x_1(0)\ldots$$

$$\ldots -\frac{1}{T}a_1(T)\delta x_1(T) + \frac{1}{T}a_1(0)\delta x_1(0) \quad (24)$$

$$\langle\frac{\partial\mathcal{L}}{\partial x_2},\delta x_2\rangle = \frac{2x_2(T)\delta x_2(T)}{x_{10}^2+x_{20}^2} - \langle\delta x_2, F_2^+\rangle - b_2\delta x_2(0)\ldots$$

$$\ldots -\frac{1}{T}a_2(T)\delta x_2(T) + \frac{1}{T}a_2(0)\delta x_2(0) \quad (25)$$

$$\frac{\partial\mathcal{L}}{\partial x_{10}}\delta x_{10} = -2x_{10}\frac{x_1^2(T)+x_2^2(T)}{(x_{10}^2+x_{20}^2)^2}\delta x_{10} + b_1\delta x_{10} \quad (26)$$

$$\frac{\partial\mathcal{L}}{\partial x_{20}}\delta x_{20} = -2x_{20}\frac{x_1^2(T)+x_2^2(T)}{(x_{10}^2+x_{20}^2)^2}\delta x_{20} + b_2\delta x_{20} \quad (27)$$

We can also find the optimal initial condition via adjoint looping and an optimization routine. The adjoint looping algorithm finds the gradient of L w.r.t. initial conditions, $\mathbf{x}_0$.

$$\frac{\partial \mathcal{L}}{\partial x_{10}}\delta x_{10} = -2x_{10}\frac{x_1^2(T)+x_2^2(T)}{(x_{10}^2+x_{20}^2)^2}\delta x_{10} + b_1\delta x_{10} \quad (26)$$

$$\frac{\partial \mathcal{L}}{\partial x_{20}}\delta x_{20} = -2x_{20}\frac{x_1^2(T)+x_2^2(T)}{(x_{10}^2+x_{20}^2)^2}\delta x_{20} + b_2\delta x_{20} \quad (27)$$

$$-b_1 + \frac{1}{T}a_1(0) = 0, \quad t=0, \quad (32)$$

$$-b_2 + \frac{1}{T}a_2(0) = 0, \quad t=0. \quad (33)$$

reminder

Substituting (32,33) into (26,27) gives the gradient of $\mathcal{L}$ with respect to the initial conditions, which must also be zero at optimality:

$$\frac{\partial \mathcal{L}}{\partial x_{10}} = -2x_{10}\frac{x_1^2(T)+x_2^2(T)}{(x_{10}^2+x_{20}^2)^2} + \frac{1}{T}a_1(0) \quad (34)$$

$$\frac{\partial \mathcal{L}}{\partial x_{20}} = -2x_{20}\frac{x_1^2(T)+x_2^2(T)}{(x_{10}^2+x_{20}^2)^2} + \frac{1}{T}a_2(0) \quad (35)$$

We can also find the optimal initial condition via adjoint looping and an optimization routine. The adjoint looping algorithm finds the gradient of L w.r.t. initial conditions, $\mathbf{x_0}$.

Step 1 Start from an initial state $\mathbf{x_0} \equiv (x_{10}, x_{20})^T$.

Step 2 Set $x_1(0)$ and $x_2(0)$ with (11,12).

Step 3 Evolve x_1 and x_2 forwards to $t = T$ with (9,10).

Step 4 Initialize the adjoint variables a_1 and a_2 at $t = T$ with (30,31):

$$a_1(T) = \frac{2Tx_1(T)}{x_{10}^2 + x_{20}^2} \quad (36)$$

$$a_2(T) = \frac{2Tx_2(T)}{x_{10}^2 + x_{20}^2} \quad (37)$$

Step 5 Evolve a_1 and a_2 backwards to $t = 0$ with (22,23).

Step 6 Evaluate the gradient $\nabla_{\mathbf{x_0}}\mathcal{L} \equiv (\partial\mathcal{L}/\partial x_{10}, \partial\mathcal{L}/\partial x_{20})$ with (34,35). When this gradient is zero, all variations of $\mathcal{L}$ are zero and an optimal value of $\mathcal{J}$ has been found.

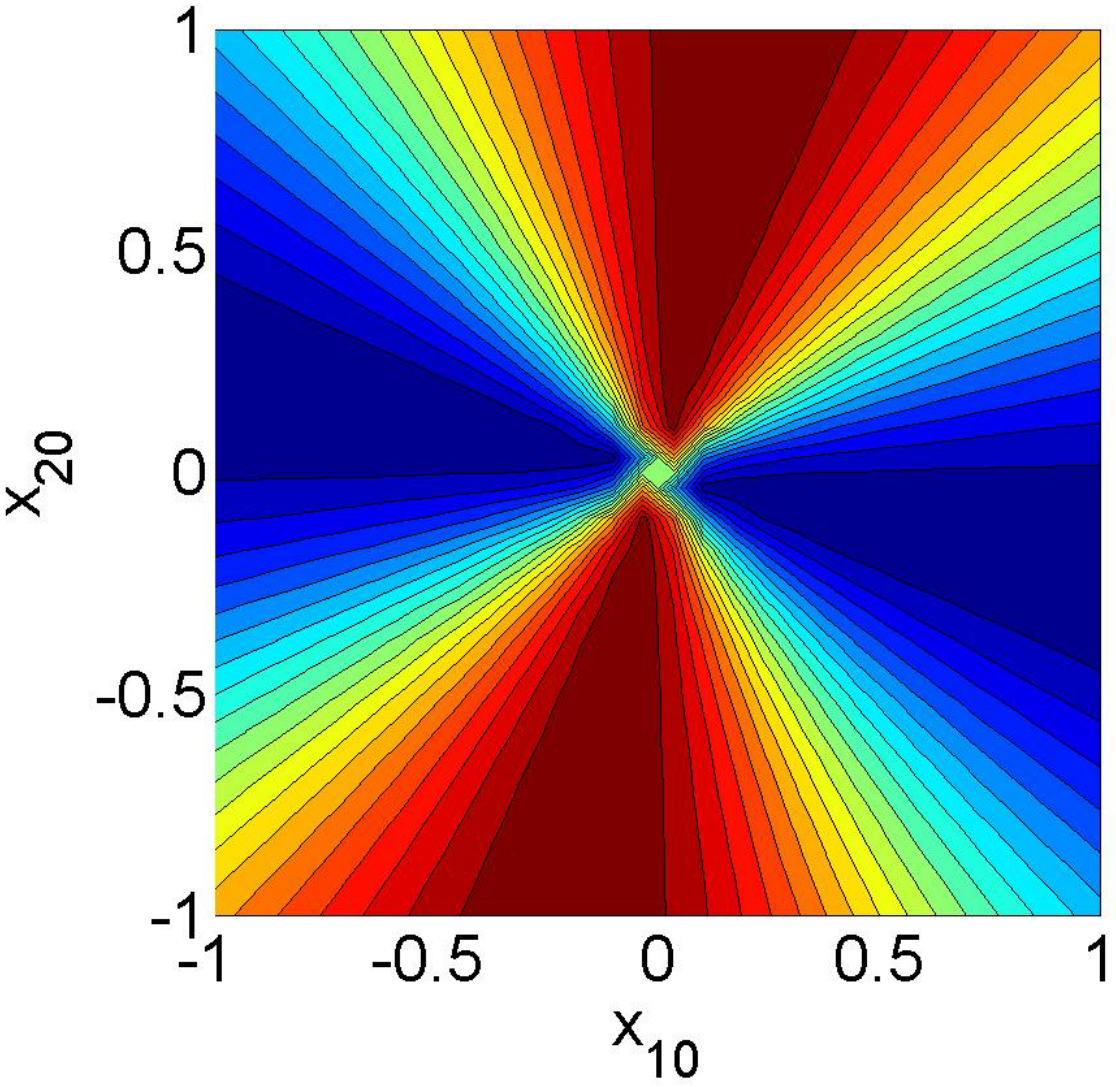
Exercise 3: (a) Write a first order Euler timemarch algorithm in `Matlab` to evolve $\mathbf{x_0}$ from $t = 0$ to $t = T$. (i.e. `x(n+1) = x(n) + dt * dxdt(n)`.) (b) Compare your results with those calculated with the matrix exponential in exercise 2. Test the dependence on `dt`. (c) The cost functional $\mathcal{J}$ is defined as $||\mathbf{x}(T)||/||\mathbf{x_0}||$. Write your script as a `Matlab` function that returns $\mathcal{J}$ as a function of $\mathbf{x_0} \equiv (x_{10}, x_{20})$ for given T and dt. (d) Write a script that plots contours of $\mathcal{J}(x_{10}, x_{20})$ for $-1 < x_{10} < 1$ and $-1 < x_{20} < 1$. `Matlab`'s `contour` function will be useful. (e) Implement the adjoint looping algorithm to evaluate $\partial\mathcal{L}/\partial x_{10}$ and $\partial\mathcal{L}/\partial x_{20}$, which we shall label `gradL`. Add this to your function so that it returns `gradL` as well as $\mathcal{J}$. (f) Write a script that plots the directions of `gradL` on top of the contours of $\mathcal{J}$. `Matlab`'s `quiver` function will be useful.

Exercise 3b In the linear adjoint loop, the quantity $\langle \mathbf{x}(t), \mathbf{a}(t) \rangle$ is conserved , which gives us a useful tool for de-bugging the algorithm. It also gives us a relationship between the direct matrix, $\mathbf{M}$ and the adjoint matrix, A. These are defined such that $\mathrm{d}\mathbf{x}/\mathrm{d}t = \mathbf{M}\mathbf{x}$ and $\mathrm{d}\mathbf{a}/\mathrm{d}t = \mathbf{A}\mathbf{a}$. By considering $\langle \mathbf{x}(0), \mathbf{a}(0) \rangle$ and $\langle \mathbf{x}(T), \mathbf{a}(T) \rangle$, or otherwise, find the relationship between matrix $\mathbf{M}$ and matrix $\mathbf{A}$.

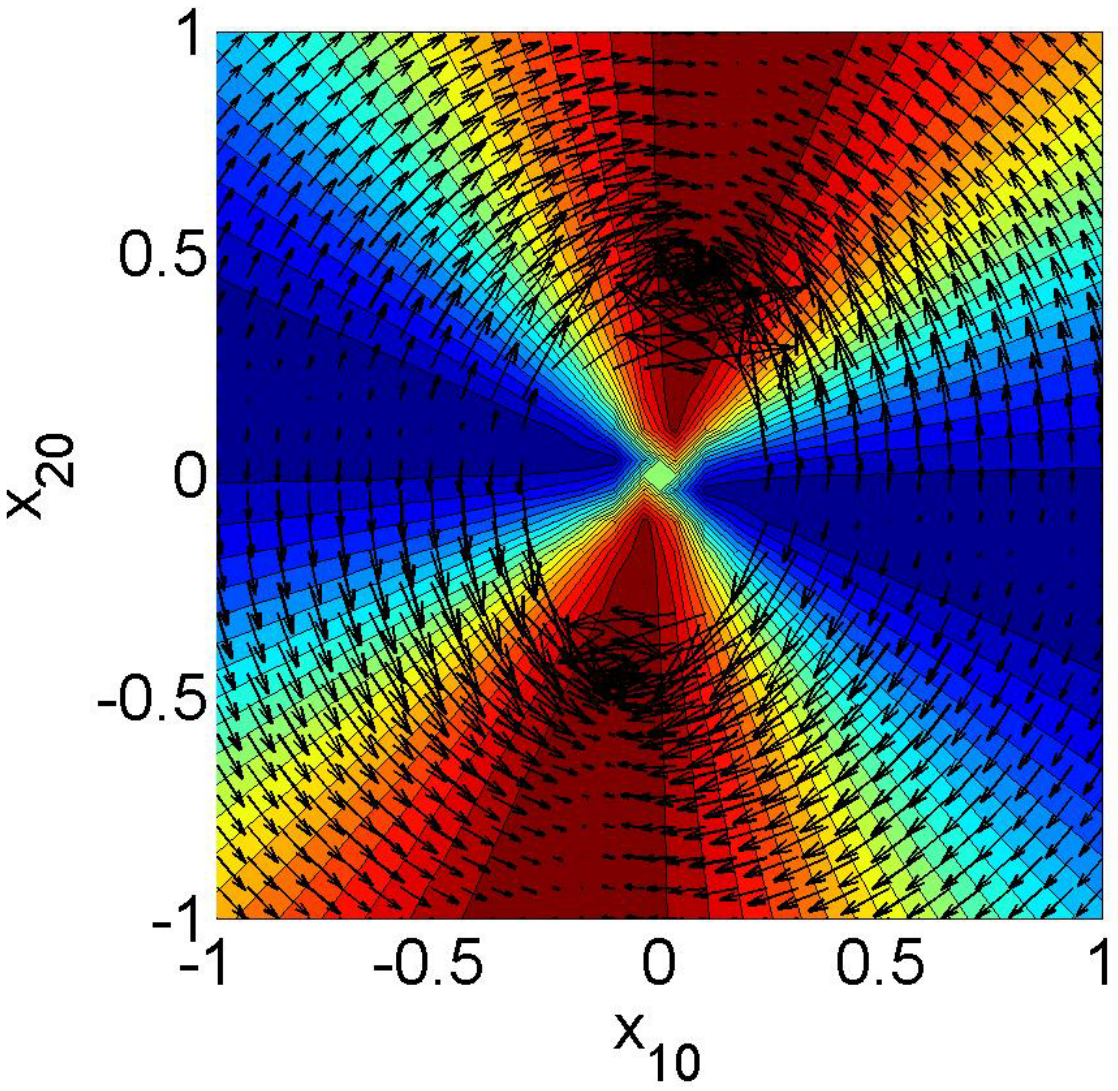
We can also find the optimal initial condition via adjoint looping and an optimization routine. The adjoint looping algorithm finds the gradient of L w.r.t. initial conditions, $\mathbf{x}_0$.

```
function [J,dL] = fun_JdL_lin_RK1(T,dt,X0)

t = 0:dt:T; L = length(t);

X(:,1) = X0;
for n = 1:L-1
    dXdt = M * X(:,n); X(:,n+1) = X(:,n) + dt * dXdt;
end
E = sum(X.^2); ET = E(L);
J = ET/E0;

Ma = -M';
A(:,L) = X(:,L) * 2*T/E0;
for n = L:-1:2
    dAdt = Ma * A(:,n); A(:,n-1) = A(:,n) - dt * dAdt;
end

dL = A(:,1)/T  -  X0 * 2*ET/E0^2;
```

The contours show cost functional, $J(\mathbf{x}_0)$

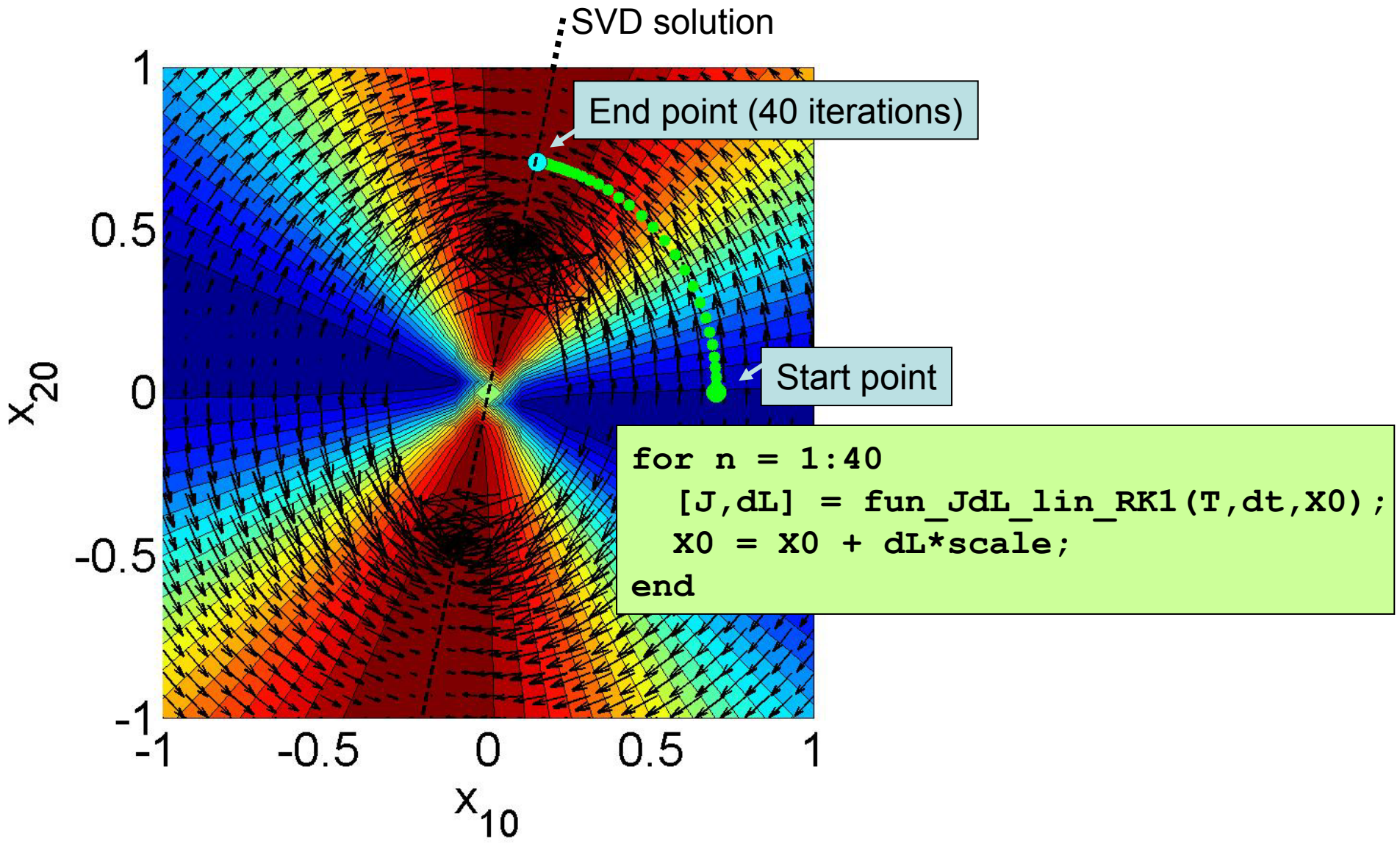


The contours show cost functional, $J(\mathbf{x}_0)$. The arrows show the gradient information calculated with the adjoint looping algorithm.



Exercise 4: (a) A simple optimization algorithm, which works for linear systems, is to set $(x_{10}, x_{20}) \leftarrow (a_1(0), a2(0))$ between loops. Because the system is linear, $\mathbf{x_0}$ can be renormalized between loops. Implement this algorithm and investigate its rate of convergence. (b) Another simple algorithm is to move incrementally in the direction of steepest descent or ascent of $\mathcal{J}$, which is given by $\pm$`gradL`. Implement this algorithm and investigate its convergence.

The contours show cost functional, $J(\mathbf{x}_0)$. The arrows show the gradient information calculated with the adjoint looping algorithm. Green dots show an optimization algorithm.




```
for n = 1:40
   [J,dL] = fun_JdL_lin_RK1(T,dt,X0);
   X0 = X0 + dL*scale;
end
```

Now let us find gradient information from non-linear governing equations.

3 Lagrange optimization of nonlinear ODEs

Let us add some simple non-linear terms to (5):

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{M}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \mathbf{N}\begin{pmatrix} x_1^2 \\ x_2^2 \end{pmatrix} \tag{38}$$

$$\mathbf{N} \equiv \begin{pmatrix} n_{11} & n_{12} \\ n_{21} & n_{22} \end{pmatrix}. \tag{39}$$

Equations (9,10) become:

$$F_1 \equiv \frac{\mathrm{d}x_1}{\mathrm{d}t} - m_{11}x_1 - m_{12}x_2 - n_{11}x_1^2 - n_{12}x_2^2 = 0, \tag{40}$$

$$F_2 \equiv \frac{\mathrm{d}x_2}{\mathrm{d}t} - m_{21}x_1 - m_{22}x_2 - n_{21}x_1^2 - n_{22}x_2^2 = 0. \tag{41}$$

Equation (17) now has two more terms:

$$\langle a_1, F_2 \rangle = \ldots - \langle a_1, n_{11} x_1^2 \rangle - \langle a_1, n_{12} x_2^2 \rangle \quad (42)$$

$$= \ldots - \frac{1}{T} \int_0^T a_1 n_{11} x_1^2 \mathrm{d}t - \frac{1}{T} \int_0^T a_1 n_{12} x_2^2 \mathrm{d}t \quad (43)$$

If we denote these terms by $\mathcal{N}$, their contributions to $\partial \mathcal{L} / \partial x_1$, $\partial \mathcal{L} / \partial x_2$ are:

$$\langle \frac{\partial \mathcal{N}}{\partial x_1}, \delta x_1 \rangle \equiv \frac{\mathcal{N}(x + \epsilon \delta x_1) - \mathcal{N}(x_1)}{\epsilon} \quad (44)$$

$$= \frac{1}{T} \int_0^T 2 x_1 n_{11} a_1 \delta x_1 \mathrm{d}t \quad (45)$$

$$= \langle \delta x_1, 2 x_1 n_{11} a_1 \rangle, \quad (46)$$

$$\text{and} \quad \langle \frac{\partial \mathcal{N}}{\partial x_2}, \delta x_2 \rangle = \langle \delta x_2, 2 x_2 n_{12} a_1 \rangle \quad (47)$$

When the variations with respect to δx_1 and δx_2 are set to zero, these terms contribute four more terms to the adjoint equations (22,23):

$$F_1^+ \equiv -\frac{\mathrm{d}a_1}{\mathrm{d}t} - m_{11}a_1 - m_{21}a_2 - 2x_1 n_{11} a_1 - 2x_1 n_{21} a_2 \quad (48)$$

$$F_2^+ \equiv -\frac{\mathrm{d}a_2}{\mathrm{d}t} - m_{12}a_1 - m_{22}a_2 - 2x_2 n_{12} a_1 - 2x_2 n_{22} a_2 \quad (49)$$

Equations (28 - 35) remain the same. The algorithm remains the same but $\mathbf{x}$ must be stored during the forward evolution in **Stage 3** so that it can be used during the backward evolution in **Stage 4**.

Exercise 5: Repeat exercises 2 and 3 for the non-linear system with the same $\mathbf{M}$ and with

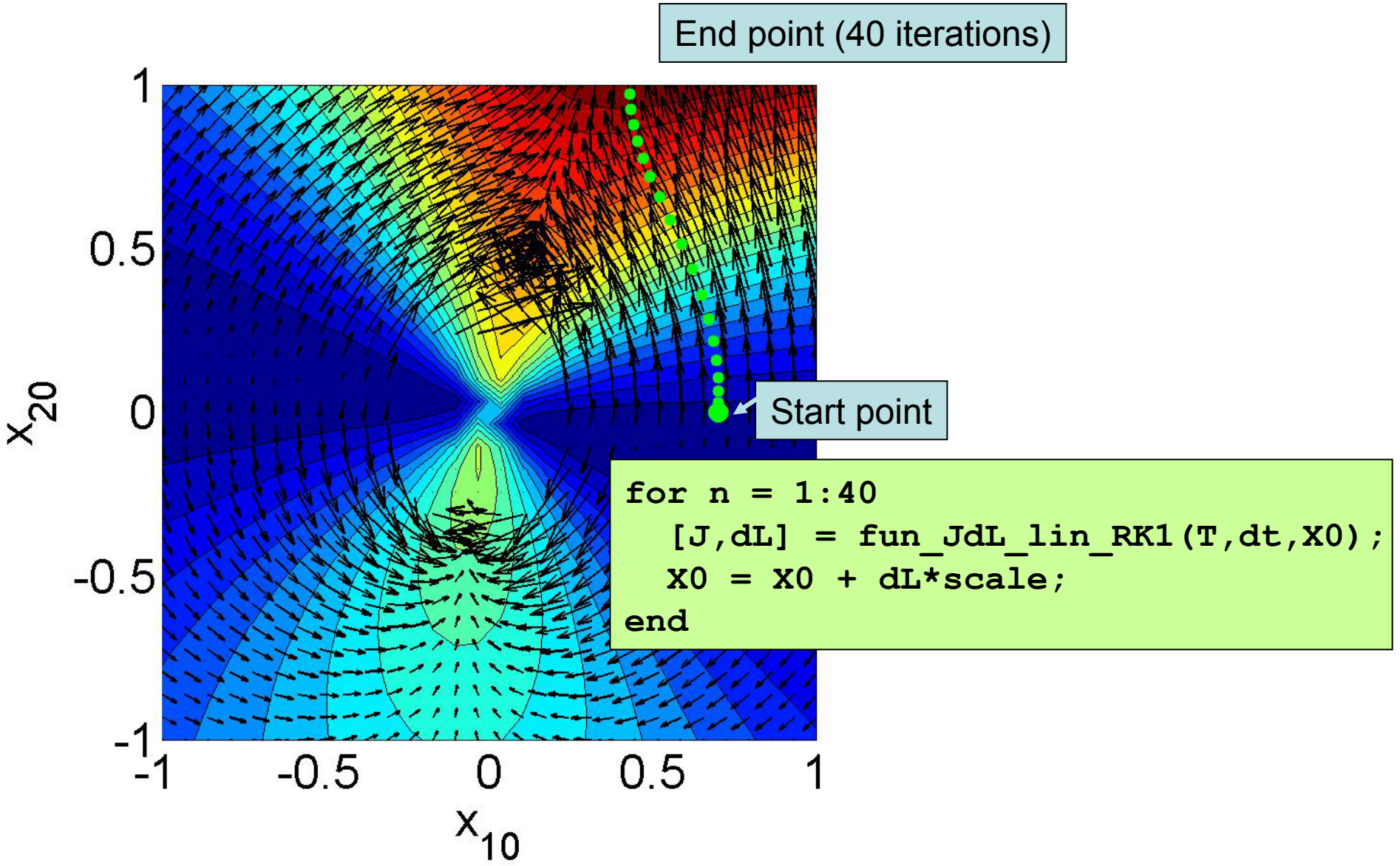
$$\mathbf{N} = \begin{pmatrix} 0.1 & -0.03 \\ 0.05 & -0.1 \end{pmatrix}.$$

An example is shown in Fig. 1 for these values of M and N. This particular example does not have isolated maxima but demonstrates the principle nonetheless. Examples with isolated maxima are shown in §4.

The algorithm is very similar to that for the linear governing equations. The extra terms are highlighted.

```
function [J,dL] = fun_JdL_nonlin_RK1(T,dt,X0)

t = 0:dt:T; L = length(t);

X(:,1) = X0;
for n = 1:L-1
    dXdt = M * X(:,n) + N * X(:,n).^2;
    X(:,n+1) = X(:,n) + dt * dXdt;
end
E = sum(X.^2); ET = E(L);
J = ET/E0;

Ma = -M'; Na = -N';
A(:,L) = X(:,L) * 2*T/E0;
for n = L:-1:2
    dAdt = Ma * A(:,n) + 2*X(:,n) .* (Na*A(:,n));
    A(:,n-1) = A(:,n) - dt * dAdt;
end

dL = A(:,1)/T  -  X0 * 2*ET/E0^2;
```

The contours show cost functional, $J(\mathbf{x}_0)$. The arrows show the gradient information calculated with the adjoint looping algorithm. Green dots show an optimization algorithm.



Exercise 6: (a) Repeat exercises 2 and 2 for a simple model of a thermoacoustic system:

$$\frac{\mathrm{d}x_1}{\mathrm{d}t} = x_2 \tag{50}$$

$$\frac{\mathrm{d}x_2}{\mathrm{d}t} = -x_1 - \zeta x_2 + \beta \left[((1+x_1)^2 + \epsilon^2)^{1/4} - 1 \right], \tag{51}$$

where ζ and β are constants of order 0.1 to 1 and ϵ is a small constant of order 10^{-6}. (b) Repeat this exercise when x_1 in the β-term is replaced with $x_1(t-\tau)$, where τ is a time delay (difficult).