Problem Sheet 14

Problem 14.1 ODEs for Matrix-Valued Functions

Let the matrix-valued function $\mathbf{Y} : \mathbb{R} \to \mathbb{R}^{d\times d}$ be a solution of the (matrix) differential equation

$$\dot{\mathbf{Y}} = \mathbf{A}\mathbf{Y} \quad \text{with} \quad \mathbf{A} \in \mathbb{R}^{d\times d}. \tag{14.1.1}$$

(14.1a) Assume $\mathbf{A}^\top\mathbf{H} = -\mathbf{H}\mathbf{A}$. Show that $\mathbf{Y}(t)^\top\mathbf{H}\mathbf{Y}(t) = \mathbf{H}$ for all $t > 0$ provided $\mathbf{Y}(0)^\top\mathbf{H}\mathbf{Y}(0) = \mathbf{H}$.

HINT: You might want to compute $\frac{\mathrm{d}}{\mathrm{d}t}$ of $\mathbf{Y}^\top\mathbf{H}\mathbf{Y}$.

Solution: We will show that $\mathbf{Y}^\top\mathbf{H}\mathbf{Y}$ is constant. From this, it follows that the orthogonality is preserved, as claimed.

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}(\mathbf{Y}^\top\mathbf{H}\mathbf{Y}) &= \dot{\mathbf{Y}}^\top\mathbf{H}\mathbf{Y} + \mathbf{Y}^\top\mathbf{H}\dot{\mathbf{Y}} \\
&\overset{\text{IVP}}{=} (\mathbf{A}\mathbf{Y})^\top\mathbf{H}\mathbf{Y} + \mathbf{Y}^\top\mathbf{H}\mathbf{A}\mathbf{Y} \\
&= \mathbf{Y}^\top(\mathbf{A}^\top\mathbf{H} + \mathbf{H}\mathbf{A})\mathbf{Y} \\
&= 0
\end{aligned}$$

Therefore,

$$\mathbf{Y}(t)^\top\mathbf{H}\mathbf{Y}(t) = \mathit{const} = \mathbf{Y}(0)^\top\mathbf{H}\mathbf{Y}(0) = \mathbf{H}.$$

(14.1b) Implement the following functions in MATLAB

(i) `function Y = ExplEulStep(A, Y0, h),`

(ii) `function Y = ImplEulStep(A, Y0, h),`

(iii) `function Y = ImplMidpStep(A, Y0, h),`

which, for a given initial value $\mathbf{Y}(t_0) = \mathbf{Y}_0$ and for a given step size h, compute approximations to $\mathbf{Y}(t_0 + h)$ for the solution of (14.1.1) using a (*single*) step of

(i) the explicit Euler method,

(ii) the implicit Euler method,

(iii) the implicit mid-point method.

Solution: The explicit Euler method is given by

$$\mathbf{Y}_{k+1} = \mathbf{Y}_k + h\mathbf{f}(t_k, \mathbf{Y}_k).$$

For the given differential equation we therefore obtain

$$\mathbf{Y}_{k+1} = \mathbf{Y}_k + h\mathbf{A}\mathbf{Y}_k.$$

The implicit Euler method is given by

$$\mathbf{Y}_{k+1} = \mathbf{Y}_k + h\mathbf{f}(t_{k+1}, \mathbf{Y}_{k+1}).$$

For the given differential equation this yields

$$\mathbf{Y}_{k+1} = \mathbf{Y}_k + h\mathbf{A}\mathbf{Y}_{k+1} \quad \Longrightarrow \quad \mathbf{Y}_{k+1} = (\mathbf{I} - h\mathbf{A})^{-1}\mathbf{Y}_k$$

Finally, the implicit mid-point method is given by

$$\mathbf{Y}_{k+1} = \mathbf{Y}_k + h\mathbf{f}\big(\tfrac{1}{2}(t_k + t_{k+1}), \tfrac{1}{2}(\mathbf{Y}_k + \mathbf{Y}_{k+1})\big),$$

hence

$$\mathbf{Y}_{k+1} = \mathbf{Y}_k + \tfrac{h}{2}\mathbf{A}(\mathbf{Y}_k + \mathbf{Y}_{k+1}) \quad \Longrightarrow \quad \mathbf{Y}_{k+1} = \big(\mathbf{I} - \tfrac{h}{2}\mathbf{A}\big)^{-1}\big(\mathbf{Y}_k + \tfrac{h}{2}\mathbf{A}\mathbf{Y}_k\big).$$

For details concerning the implementation see `ExpEulStep.m`, `ImplEulStep.m` and `ImplMidpStep.m` (implementation Listing 14.1, Listing 14.2, Listing 14.3).

Listing 14.1: Explicit Euler method

```
function Y = ExplEulStep(A, Y0, h)
% explicit Euler step
% Input:
%    A - matrix
%    Y0 - initial value
%    h - step size

Y = Y0 + h*A*Y0;
```

Listing 14.2: Implicit Euler Method

```
function Y = ImplEulStep(A, Y0, h)
% implicit Euler step
% Input:
%    A - matrix
%    Y0 - initial value
%    h - step size

Y = (eye(size(A)) - h*A) \ Y0;
```

Listing 14.3: Midpoint method

```
function Y = ImplMidpStep(A, Y0, h)
% step of implicit midpoint rule
% Input:
%     A - matrix
%     Y0 - initial value
%     h - step size

Y = (eye(size(A)) - 0.5*h*A) \ (Y0 + 0.5*h*A*Y0);
```

(14.1c) Take now $\mathbf{A} = \begin{pmatrix} -3 & -6 \\ 6 & 3 \end{pmatrix}$, $\mathbf{Y}(0) = \frac{1}{\sqrt{3}}\begin{pmatrix} 2 & 1 \\ -1 & -2 \end{pmatrix}$, and $\mathbf{H} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$. Complete the template `CompareNorms.m` where, using the functions from subproblem (14.1b), you should compute discrete approximations $\mathbf{Y}_k$ of $\mathbf{Y}(kh)$, for $k = 1, \ldots, 20$ with $h = 1/20$. Compare the norms $\left\| \mathbf{Y}_k^\top \mathbf{H} \mathbf{Y}_k - \mathbf{H} \right\|_F$, for $k = 1, \ldots, 20$ and all three methods, and comment on your observations with regards to the invariant from subproblem (14.1a).

HINT: The Frobenius norm $\|\cdot\|_F$ of a matrix can be computed using the command `norm(A,'fro')`.

Solution: For the implementation, see `comparenorms.m`, Listing 14.4.

Listing 14.4: comparenorms.m

```
% set the matrix A, the intital condition Y(0) and step size h

d = 3;
A = [-d, -2*d; 2*d, d];
Y0 = [2, 1; -1, -2] / sqrt(3);
H = [2 1; 1 2];
h = 1/20;

% starting values for the iteration
Yexpleul = Y0;
Yimpleul = Y0;
Yimplmidp = Y0;

% allocate memory for results (||Y_k'Y_k-I||_F)
normexpleul = zeros(20, 1);
normimpleul = zeros(20, 1);
normimplmidp = zeros(20, 1);

for k = 1:20

    % compute the steps using the explicit, implicit Euler
      and implicit
    % midpoint rule
     Yexpleul = ExplEulStep(A, Yexpleul, h);
     Yimpleul = ImplEulStep(A, Yimpleul, h);
     Yimplmidp = ImplMidpStep(A, Yimplmidp, h);
```

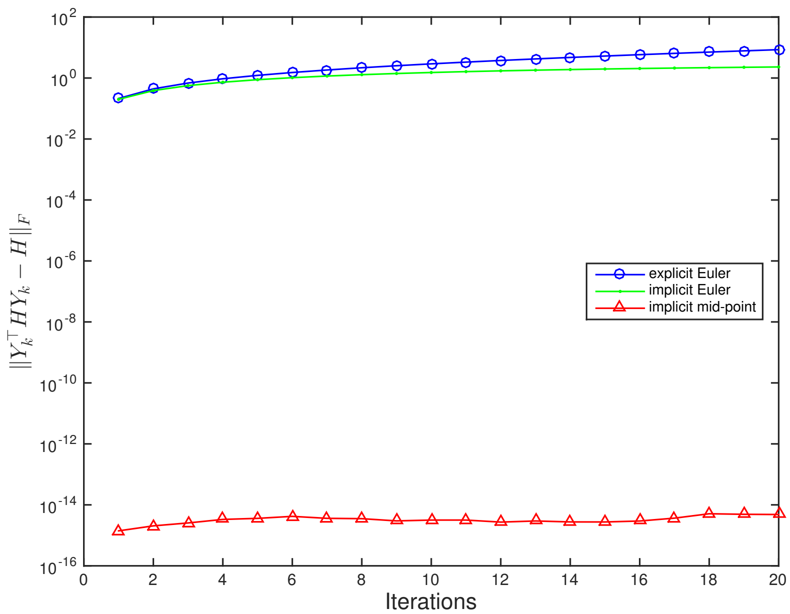

Figure 14.1: Deviation from $\mathbf{H}$ for the three methods considered.

```
    % compute the norms
    normexpleul(k) = norm(Yexpleul'*H*Yexpleul - H, 'fro');
    normimpleul(k) = norm(Yimpleul'*H*Yimpleul - H, 'fro');
    normimplmidp(k) = norm(Yimplmidp'*H*Yimplmidp - H,
       'fro');
end

% plot the results

fprintf('Explicit Euler: %.2e\n', normexpleul(20));
fprintf('Implicit Euler: %.2e\n', normimpleul(20));
fprintf('Implicit mid-point: %.2e\n', normimplmidp(20));

figure
semilogy(1:20, normexpleul, 'bo-', 1:20, normimpleul, 'g.-',
   1:20, normimplmidp, 'r^-')
xlabel('Iterations', 'FontSize', 14)
ylabel('$\Vert Y_{k}^\top H Y_{k}-H\Vert_F$', 'Interpreter',
   'latex', 'FontSize', 14)
legend('explicit Euler', 'implicit Euler', 'implicit
   mid-point', 'Location', 'East')
```

The results are shown in Figure 14.1. We can see that the implicit mid-point method is the only method preserving orthogonality at machine precision.

(14.1d) Show that the solution $\mathbf{Y}_k$ computed via the implicit mid-point rule satisfies:

$$\text{if} \quad \mathbf{Y}_0^\top \mathbf{H} \mathbf{Y}_0 = \mathbf{H} \quad \text{then} \quad \mathbf{Y}_k^\top \mathbf{H} \mathbf{Y}_k = \mathbf{H} \quad \text{for all } k \geq 1.$$

HINT: You might find the identity $\mathbf{Y}_1 - \mathbf{Y}_0 = \frac{h}{2}\mathbf{A}(\mathbf{Y}_0 + \mathbf{Y}_1)$ useful.

Solution: It suffices to carry out the proof for a single step ($k = 1$) of the implicit mid-point method. The statement for general k follows by induction. In analogy to the proof of subproblem (14.1a) we consider the expression $(\mathbf{Y}_1 - \mathbf{Y}_0)^\top \mathbf{H}(\mathbf{Y}_0 + \mathbf{Y}_1) + (\mathbf{Y}_0 + \mathbf{Y}_1)^\top \mathbf{H}(\mathbf{Y}_1 - \mathbf{Y}_0)$. On one hand, we have

$$(\mathbf{Y}_1 - \mathbf{Y}_0)^\top \mathbf{H}(\mathbf{Y}_0 + \mathbf{Y}_1) + (\mathbf{Y}_0 + \mathbf{Y}_1)^\top \mathbf{H}(\mathbf{Y}_1 - \mathbf{Y}_0) = 2\mathbf{Y}_1^\top \mathbf{H} \mathbf{Y}_1 - 2\mathbf{Y}_0^\top \mathbf{H} \mathbf{Y}_0,$$

on the other hand, plugging in $\mathbf{Y}_1 - \mathbf{Y}_0 = \frac{h}{2}\mathbf{A}(\mathbf{Y}_0 + \mathbf{Y}_1)$ from the implicit mid-poin rule:

$$\begin{aligned}
&(\mathbf{Y}_1 - \mathbf{Y}_0)^\top \mathbf{H}(\mathbf{Y}_0 + \mathbf{Y}_1) + (\mathbf{Y}_0 + \mathbf{Y}_1)^\top \mathbf{H}(\mathbf{Y}_1 - \mathbf{Y}_0) \\
&= \tfrac{h}{2}(\mathbf{Y}_0 + \mathbf{Y}_1)^\top \mathbf{A}^\top \mathbf{H}(\mathbf{Y}_0 + \mathbf{Y}_1) + \tfrac{h}{2}(\mathbf{Y}_0 + \mathbf{Y}_1)^\top \mathbf{H}\mathbf{A}(\mathbf{Y}_0 + \mathbf{Y}_1) \\
&= 0.
\end{aligned}$$

Combining the two statements yields

$$\mathbf{Y}_1^\top \mathbf{H} \mathbf{Y}_1 = \mathbf{Y}_0^\top \mathbf{H} \mathbf{Y}_0,$$

as desired.

Problem 14.2 Projection Method for Hamiltonian equations

When an ODE is defined on a manifold it is important that the approximate solution is also contained in the manifold. A natural ansatz consists in projecting the approximate solution to this manifold. In the present exercise we will apply this ansatz to Hamilton's equations in such a way that important first integrals are preserved. We will also investigate how many such projections are necessary in order to obtain a qualitatively correct solution.

As our example we consider the perturbed Kepler problem whose Hamiltonian is given by

$$H(\mathbf{p}, \mathbf{q}) = \frac{1}{2}\left(p_1^2 + p_2^2\right) - \frac{1}{\sqrt{q_1^2 + q_2^2}} - \frac{0.005}{2\sqrt{(q_1^2 + q_2^2)^3}}. \tag{14.2.1}$$

(14.2a) Derive the Hamilton's equations for (14.2.1).

Solution:

$$\begin{cases} \dot{\mathbf{q}} = \frac{\partial H}{\partial \mathbf{p}} = \mathbf{p}, \\ \dot{\mathbf{p}} = -\frac{\partial H}{\partial \mathbf{q}} = \left(-\frac{1}{\sqrt{(q_1^2+q_2^2)^3}} - \frac{3 \cdot 0.005}{2\sqrt{(q_1^2+q_2^2)^5}}\right)\mathbf{q}. \end{cases}$$

(14.2b) Write down a first integral for the ODE derived in subproblem (14.2a) and show that the angular momentum

$$L(\mathbf{p}, \mathbf{q}) = q_1 p_2 - q_2 p_1 \tag{14.2.2}$$

defines an additional first integral.

Solution: The Hamiltonian is a first integral and

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} L(\mathbf{p}, \mathbf{q}) &= \dot{q}_1 p_2 + q_1 \dot{p}_2 - \dot{q}_2 p_1 - q_2 \dot{p}_1 \,, \\
&= p_1 p_2 + q_1 \left(-\frac{1}{\sqrt{(q_1^2+q_2^2)^3}} - \frac{0.015}{2\sqrt{(q_1^2+q_2^2)^5}} \right) q_2 - p_2 p_1 \\
&\quad - q_2 \left(-\frac{1}{\sqrt{(q_1^2+q_2^2)^3}} - \frac{0.015}{2\sqrt{(q_1^2+q_2^2)^5}} \right) q_1 \\
&= 0 \,.
\end{aligned}
$$

(14.2c) Fill in the template `proHam1.m` which solves the ODE 14.2.1 using the explicit Euler method, the Störmer–Verlet method and the symplectic Euler method and plots the trajectory of the position coordinate $\binom{q_1(t)}{q_2(t)}$ for each of the three aforementioned methods .

HINT: The symplectic Euler method is a Lie–Trotter splitting method, see [NUMODE, Eq. (2.5.11)].

Solution:

Listing 14.5: Implementation of subproblem (14.2c)

```
function proHam1
close all

% y=[q1;q2;p1;p2]
e=0.6;
y0=[1-e;0;0;sqrt((1+e)/(1-e))];

% T=200;

h=0.03;

N=8000; % N=ceil(T/h);

g=@(y) -1/sqrt((y(1)^2+y(2)^2)^3) -
   3*0.005/sqrt((y(1)^2+y(2)^2)^5)/2;

% explicit Euler
f=@(y) [y(3);y(4);y(1)*g(y(1:2));y(2)*g(y(1:2))];
clear y
clear v

q=zeros(2,N+1);
q(:,1)=y0(1:2);
y=y0;
for ii = 1:N

    y = y + h*f(y);

```

```
    q(:,ii+1) = y(1:2);

end
figure;
plot(q(1,:),q(2,:),'*-');

% Stoermer-Verlet
f=@(y) [y(1)*g(y(1:2));y(2)*g(y(1:2))];
clear y
clear v

q=zeros(2,N+1);
q(:,1)=y0(1:2);

y=y0(1:2);
v=y0(3:4);
for ii = 1:N

    v05=v+h/2*f(y);
    y=y+h*v05;
    v=v05+h/2*f(y);

    %y = y + h*f(y);

    q(:,ii+1) = y(1:2);

end
figure;
plot(q(1,:),q(2,:),'*-');

% Symplectic Euler
f=@(y) [y(1)*g(y(1:2));y(2)*g(y(1:2))];
clear y
clear v

q=zeros(2,N+1);
q(:,1)=y0(1:2);
y=y0(1:2);
v=y0(3:4);
for ii = 1:N

    v=v+h*f(y);
    y=y+h*v;

    q(:,ii+1) = y(1:2);

end
```

```
figure;
plot(q(1,:),q(2,:),'*-');
```

(14.2d) subproblem (14.2c) makes clear the fact that the explicit Euler method yields a qualitatively wrong solution. To obtain a better approximation, we project the numerical solution to the submanifold

$$\mathcal{M} := \left\{ \mathbf{z} = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \in \mathbb{R}^4 \,\middle|\, H(\mathbf{x}, \mathbf{y}) = H(\mathbf{p}(0), \mathbf{q}(0)) \right\} \subseteq \mathbb{R}^4$$

after each time step. Let $\mathbf{z} \in \mathbb{R}^4$ (e.g. $\mathbf{z}$ might represent the approximation computed using the explicit Euler method). A projection $\tilde{\mathbf{z}}$ of $\mathbf{z}$ onto the manifold $\mathcal{M}$ can be defined to be a solution of the following system

$$\begin{aligned} \tilde{\mathbf{z}} &= \mathbf{z} + s\,\mathbf{grad}(H(\mathbf{z})), \\ 0 &= H(\tilde{\mathbf{z}}) - H(\mathbf{p}(0), \mathbf{q}(0)), \end{aligned} \tag{14.2.3}$$

where $s \in \mathbb{R}$. Carry out one step of the Newton method for (14.2.3) and determine an approximation of the projection $\tilde{\mathbf{z}}$. Choose $s^{(0)} = 0$ as the initial value for the parameter s.

HINT: Solve (14.2.3) for s as the only unknown. Then determine an approximation for s via the Newton method and construct the projection $\tilde{\mathbf{z}}$.

Solution: The parameter s satisfies

$$0 = H(\mathbf{z} + s\,\mathbf{grad}(H(\mathbf{z}))) - H(\mathbf{p}(0), \mathbf{q}(0)),$$

where $\mathbf{z}$, $\mathbf{p}(0)$ and $\mathbf{q}(0)$ are known. We define the function

$$F(s) :\,= H(\mathbf{z} + s\,\mathbf{grad}(H(\mathbf{z}))) - H(\mathbf{p}(0), \mathbf{q}(0)).$$

An approximation of s using a single step of the Newton method with initial value $s^{(0)} = 0$ yields

$$s^{(1)} = -F(0) \,/\, \frac{\partial F}{\partial s}(0),$$

where

$$\frac{\partial F}{\partial s}(s) = \mathbf{grad}(H(\mathbf{z} + s\,\mathbf{grad}(H(\mathbf{z}))))^\top\, \mathbf{grad}(H(\mathbf{z})).$$

The approximation of the projection $\tilde{\mathbf{z}}$ is then given by

$$\tilde{\mathbf{z}} \approx \mathbf{z} + s^{(1)}\,\mathbf{grad}(H(\mathbf{z})).$$

(14.2e) Complete the template `proHam2.m`, which solves the ODE (14.2.1) using the method derived in subproblem (14.2d), and plots the trajectory of the position coordinates $\binom{p_1(t)}{p_2(t)}$ as well as the evolution of the energy.

Solution:

Listing 14.6: Implementation of subproblem (14.2e)

```
function proHam2

close all

%y=[q1;q2;p1;p2]
e=0.6;
y0=[1-e;0;0;sqrt((1+e)/(1-e))];

h=0.03;

N=8000; %N=ceil(T/h);

g=@(y) -1/sqrt((y(1)^2+y(2)^2)^3) -
   3*0.005/sqrt((y(1)^2+y(2)^2)^5)/2;
f=@(y) [y(3);y(4);y(1)*g(y(1:2));y(2)*g(y(1:2))];

q=zeros(2,N+1);
q(:,1)=y0(1:2);

H=@(y) 0.5*(y(3)^2+y(4)^2)-1/sqrt(y(1)^2+y(2)^2)-0.005/2/...
    sqrt((y(1)^2+y(2)^2)^3);
DH=@(y) [-y(1)*g(y(1:2));-y(2)*g(y(1:2));y(3);y(4)];

y=y0;

energy=zeros(1,N+1);
energy(1)=H(y0);
for ii = 1:N

    %y0=y;
    F=@(s,y,y0) H(y+s*DH(y))-H(y0);
    DF=@(s,y,y0) DH(y+s*DH(y))'*DH(y);

    %explicit Euler
    y = y + h*f(y);

    %projection to H(p,q)=H(p0,q0)
    s=-DF(0,y,y0)\F(0,y,y0);
    y=y+s*DH(y);
    energy(ii+1)=H(y);
    q(:,ii+1) = y(1:2);

end
figure;
plot(q(1,:),q(2,:),'*-');
figure;
```

```
plot(0:h:N*h,energy);
ylim([H(y0)-1 H(y0)+1])
```

(14.2f) From the subproblem (14.2e) we can infer that the projection to the manifold $\mathcal{M}$ is not sufficient to obtain a qualitatively correct solution. Let us try to improve this method by projecting onto a smaller submanifold of $\mathbb{R}^4$, given by

$$\mathcal{L} := \left\{ \mathbf{z} = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \in \mathbb{R}^4 \,\middle|\, H(\mathbf{x}, \mathbf{y}) = H(\mathbf{p}(0), \mathbf{q}(0)),\ L(\mathbf{x}, \mathbf{y}) = L(\mathbf{p}(0), \mathbf{q}(0)) \right\}.$$

The projection $\tilde{\mathbf{z}}$ of $\mathbf{z}$ to the manifold $\mathcal{L}$ can be defined as the solution to the following system

$$\begin{aligned} \tilde{\mathbf{z}} &= \mathbf{z} + s\,\mathbf{grad}(H(\mathbf{z})) + r\,\mathbf{grad}(L(\mathbf{z})), \\ 0 &= H(\tilde{\mathbf{z}}) - H(\mathbf{p}(0), \mathbf{q}(0)), \\ 0 &= L(\tilde{\mathbf{z}}) - L(\mathbf{p}(0), \mathbf{q}(0)), \end{aligned} \tag{14.2.4}$$

where s and r are real parameters. Carry out one step of the Newton method for (14.2.4) and compute an approximation of the projection $\tilde{\mathbf{z}}$. Choose $s^{(0)} = 0$ and $r^{(0)} = 0$ as starting values for s and r.

HINT: Solve (14.2.3) for s and r as the only unknowns. Then determine an approximation for s and r via the Newton method and construct the projection $\tilde{\mathbf{z}}$.

Solution: The parameters s and r satisfy

$$\begin{cases} 0 = H(\mathbf{z} + s\,\mathbf{grad}(H(\mathbf{z})) + r\,\mathbf{grad}(L(\mathbf{z}))) - H(\mathbf{p}(0), \mathbf{q}(0)), \\ 0 = L(\mathbf{z} + s\,\mathbf{grad}(H(\mathbf{z})) + r\,\mathbf{grad}(L(\mathbf{z}))) - L(\mathbf{p}(0), \mathbf{q}(0)), \end{cases}$$

where $\mathbf{z}$, $\mathbf{p}(0)$ and $\mathbf{q}(0)$ are known. We define the function

$$F(s, r) := \begin{pmatrix} H(\mathbf{z} + s\,\mathbf{grad}(H(\mathbf{z})) + r\,\mathbf{grad}(L(\mathbf{z}))) - H(\mathbf{p}(0), \mathbf{q}(0)) \\ L(\mathbf{z} + s\,\mathbf{grad}(H(\mathbf{z})) + r\,\mathbf{grad}(L(\mathbf{z}))) - L(\mathbf{p}(0), \mathbf{q}(0)) \end{pmatrix}.$$

An approximation of s and r using a single step of the Newton method with starting values $s^{(0)} = 0 = r^{(0)}$ yields

$$\begin{pmatrix} s^{(1)} \\ r^{(1)} \end{pmatrix} = -(D_{s,r}F(0,0))^{-1} F(0,0),$$

where

$$D_{s,r}F(s,r) = \begin{pmatrix} \mathbf{grad}(H(\mathbf{z} + s\,\mathbf{grad}(H(\mathbf{z})) + r\,\mathbf{grad}(L(\mathbf{z}))))^\top\, \mathbf{grad}(H(\mathbf{z})) \ldots \\ \mathbf{grad}(L(\mathbf{z} + s\,\mathbf{grad}(H(\mathbf{z})) + r\,\mathbf{grad}(L(\mathbf{z}))))^\top\, \mathbf{grad}(H(\mathbf{z})) \end{pmatrix}$$
$$\begin{pmatrix} \ldots\,\mathbf{grad}(H(\cdots))^\top\, \mathbf{grad}(L(\mathbf{z})) \\ \mathbf{grad}(L(\cdots))^\top\, \mathbf{grad}(L(\mathbf{z})) \end{pmatrix}.$$

The approximation of the projection $\tilde{\mathbf{z}}$ is then given by

$$\tilde{\mathbf{z}} \approx \mathbf{z} + s^{(1)}\,\mathbf{grad}(H(\mathbf{z})) + r^{(1)}\,\mathbf{grad}(L(\mathbf{z})).$$

(14.2g) Complete the template `proHam3.m` which solves the ODE (14.2.1), using the method described in subproblem (14.2f), and plots the trajectory of the position coordinates $\binom{p_1(t)}{p_2(t)}$ as well as the evolution of the energy and the angular momentum.

Solution:

Listing 14.7: Implementation of subproblem (14.2g)

```
function proHam3

close all

% y=[q1;q2;p1;p2]
e=0.6;
y0=[1-e;0;0;sqrt((1+e)/(1-e))];

h=0.03;

N=8000; % N=ceil(T/h);

g=@(y) -1/sqrt((y(1)^2+y(2)^2)^3) -
   3*0.005/sqrt((y(1)^2+y(2)^2)^5)/2;
f=@(y) [y(3);y(4);y(1)*g(y(1:2));y(2)*g(y(1:2))];

q=zeros(2,N+1);
q(:,1)=y0(1:2);

H=@(y) 0.5*(y(3)^2+y(4)^2)-1/sqrt(y(1)^2+y(2)^2)-0.005/2/...
    sqrt((y(1)^2+y(2)^2)^3);
DH=@(y) [-y(1)*g(y(1:2));-y(2)*g(y(1:2));y(3);y(4)];

L=@(y) y(1)*y(4)-y(2)*y(3);
DL=@(y) [y(4);-y(3);-y(2);y(1)];

y=y0;
for ii = 1:N

    % y0=y;
    F=@(s,t,y,y0)
       [H(y+s*DH(y)+t*DL(y))-H(y0);L(y+s*DH(y)+t*DL(y))-L(y0)];
    DF=@(s,t,y,y0) [DH(y+s*DH(y)+t*DL(y))'*DH(y),...
                    DH(y+s*DH(y)+t*DL(y))'*DL(y);...
                    DL(y+s*DH(y)+t*DL(y))'*DH(y), ...
                    DL(y+s*DH(y)+t*DL(y))'*DL(y)];

    % explicit Euler
    y = y + h*f(y);

    % projection to H(p,q)=H(p0,q0)
```

```
    step=-DF(0,0,y,y0)\F(0,0,y,y0);
    s=step(1);
    t=step(2);
    y=y+s*DH(y)+t*DL(y);

    q(:,ii+1) = y(1:2);

end
figure;
plot(q(1,:),q(2,:),'*-');
```

Problem 14.3 Instability of the Störmer-Verlet Method

We consider the Hamiltonian of a harmonic oscillator, with a frequency parameter $\omega \in \mathbb{R}$, defined as

$$H(p,q) = \frac{1}{2}\omega(p^2 + q^2), \quad p, q \in \mathbb{R}.$$

(14.3a) Formulate the corresponding Hamilton's equations.

Solution: Hamilton equation's are given by

$$\dot{p} = -\omega q \qquad \dot{q} = \omega p.$$

(14.3b) Formulate the Störmer-Verlet method for the ODE from subproblem (14.3a).

Solution: The Störmer-Verlet method is

$$\begin{aligned} p_{\frac{1}{2}} &= p_0 - \frac{h}{2}\omega q_0 \\ q_1 &= q_0 + h\omega p_{\frac{1}{2}} \\ p_1 &= p_{\frac{1}{2}} - \frac{h}{2}\omega q_1 \end{aligned}$$

(14.3c) The analytic solution $\mathbf{y}(t) := (p(t), q(t))^\top$ of the ODE from subproblem (14.3a) is

$$\mathbf{y}(t) = \mathbf{W}(t\omega)\mathbf{y}_0,$$

with

$$\mathbf{W}(t\omega) := \exp\left(t\begin{pmatrix} 0 & -\omega \\ \omega & 0 \end{pmatrix}\right).$$

Plot the eigenvalue loci of $\mathbf{W}(t\,\omega)$, with respect to $t\,\omega$.

Solution: The eigenvalues of $W(t\,\omega)$ are given by $\exp(\pm i\,\omega\,t)$. Hence, the locus is the unit circle in the complex plane.

(14.3d) Transform the Störmer-Verlet method obtained in subproblem (14.3b) into the form

$$\mathbf{y}_1 = \mathbf{S}(h\,\omega)\mathbf{y}_0.$$

Plot the eigenvalue loci, with respect to $h\omega$. What can you observe by comparing this curve with the curve obtained in subproblem (14.3c)?

Solution: We find

$$\begin{pmatrix} p_1 \\ q_1 \end{pmatrix} = \begin{pmatrix} 1 - \frac{1}{2}(h\omega)^2 & -h\omega + \frac{1}{4}(h\omega)^3 \\ h\omega & 1 - \frac{1}{2}(h\omega)^2 \end{pmatrix} \begin{pmatrix} p_0 \\ q_0 \end{pmatrix}.$$

Figure 14.2 shows the loci of the eigenvalues $\lambda_{1,2} = \frac{1}{2}(2 - h^2\omega^2 \pm \sqrt{h^4\omega^4 - 4h^2\omega^2})$. The top row shows the real part of the two eigenvalues while the bottom row shows the imaginary parts. For $h\omega < 1$ the spectra of the exact and approximated evolution operator lie close together.

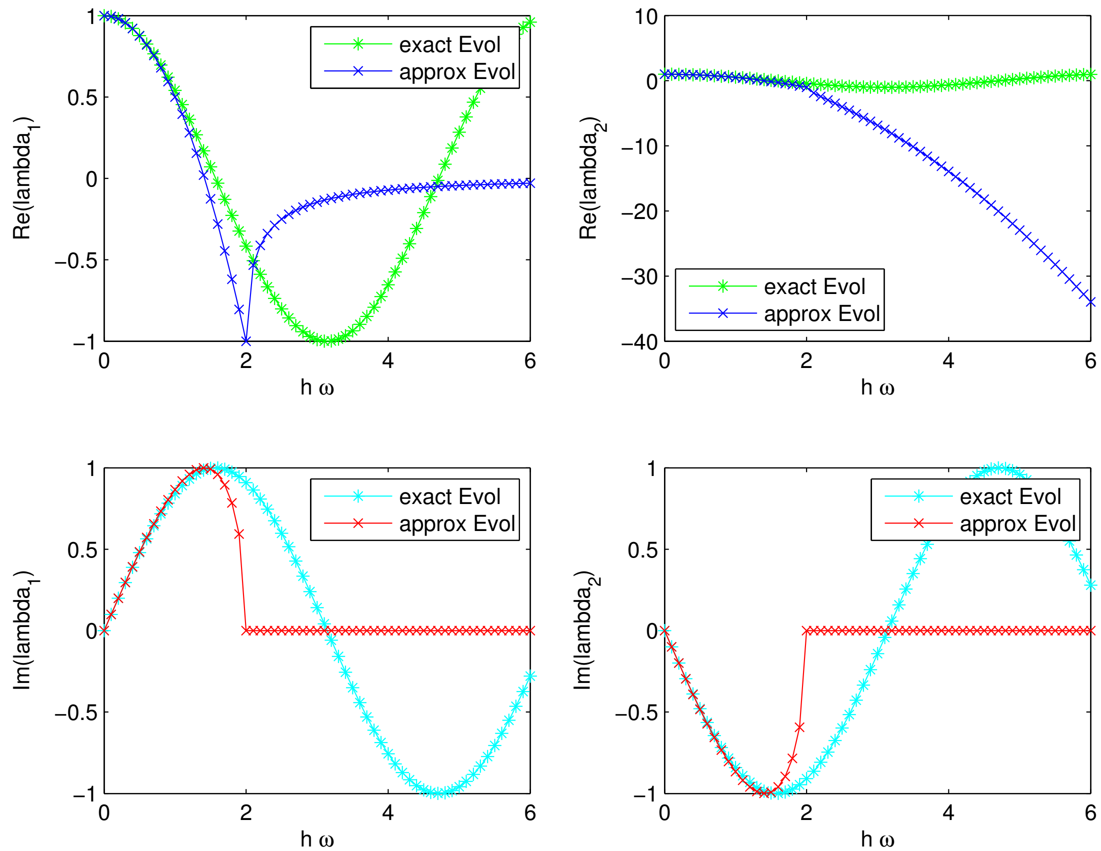

Figure 14.2: Loci of the eigenvalues of the Störmer-Verlet method.

Listing 14.8: subproblem (14.3d)

```
A=0;
B=6;
h=0.1;
I=A:h:B;
d1=zeros(2,length(I));
d2=zeros(2,length(I));

% eigenvalues of exact evolution
e1=@(s)exp(1i * s);
e2=@(s)exp(-1i * s);

for i=1:length(I)
```

```
    d1(:,i)=[e1(I(i)), e2(I(i))];;
end
figure;
title('exact evolution')
subplot(2,2,1);
plot(I,real(d1(1,:)),'bx-');
ylabel('Re(lambda_1)')
xlabel('h \omega')
subplot(2,2,3);
plot(I,imag(d1(1,:)),'rx-');
ylabel('Im(lambda_1)')
xlabel('h \omega')
subplot(2,2,2);
plot(I,real(d1(2,:)),'bx-');
ylabel('Re(lambda_2)')
xlabel('h \omega')
subplot(2,2,4);
plot(I,imag(d1(2,:)),'rx-');
ylabel('Im(lambda_2)')
xlabel('h \omega')
print -depsc fig4c.eps

% stability function
S=@(s) [1-s^2/2 -s+s^3/4 ; s 1-s^2/2];

% exact eigenvalues
e1=@(s)1/2*(2-s^2+sqrt(s^4-4*s^2));
e2=@(s)1/2*(2-s^2-sqrt(s^4-4*s^2));

% exact eigenvalues
for i=1:length(I)
    d2(:,i)=[e1(I(i)), e2(I(i))];
end
figure;
title('approximating evolution')
subplot(2,2,1);
plot(I,real(d2(1,:)),'bx-');
ylabel('Re(lambda_1)')
xlabel('h \omega')
subplot(2,2,3);
plot(I,imag(d2(1,:)),'rx-');
ylabel('Im(lambda_1)')
xlabel('h \omega')
subplot(2,2,2);
plot(I,real(d2(2,:)),'bx-');
ylabel('Re(lambda_2)')
xlabel('h \omega')
```

```
subplot(2,2,4);
plot(I,imag(d2(2,:)),'rx-');
ylabel('Im(lambda_2)')
xlabel('h \omega')
print -depsc fig4dex.eps

figure;
subplot(2,2,1);
plot(I,real(d1(1,:)),'g*-',I,real(d2(1,:)),'bx-');
ylabel('Re(lambda_1)')
xlabel('h \omega')
legend('exact Evol','approx Evol')
subplot(2,2,3);
plot(I,imag(d1(1,:)),'c*-',I,imag(d2(1,:)),'rx-');
ylabel('Im(lambda_1)')
xlabel('h \omega')
legend('exact Evol','approx Evol')
subplot(2,2,2);
plot(I,real(d1(2,:)),'g*-',I,real(d2(2,:)),'bx-');
ylabel('Re(lambda_2)')
xlabel('h \omega')
legend('exact Evol','approx Evol','Location','Southwest')
subplot(2,2,4);
plot(I,imag(d1(2,:)),'c*-',I,imag(d2(2,:)),'rx-');
ylabel('Im(lambda_2)')
xlabel('h \omega')
legend('exact Evol','approx Evol')
print -depsc fig4all.eps
```

(14.3e) What do you observe for $h\omega \to \infty$? What are the consequences on the numerical method?

Solution: We have $\lambda = \frac{1}{2}(2 - h^2\omega^2 - \sqrt{h^4\omega^4 - 4h^2\omega^2}) \to -\infty$ for $h\omega \to \infty$, i.e. the method becomes unstable.

Published on 30 May 2016.

To be submitted by 31 Dec 9999.

There's no submission deadline for this assignment, but at least having a look at all assignments before final exam might be a good suggestion.

References

[NODE] Lecture Notes for the course "Numerical Methods for Ordinary Differential Equations".

[NUMODE] Lecture Slides for the course “Numerical Methods for Ordinary Differential Equations”, SVN revision # 52913.

Last modified on May 26, 2016