

**AMATH 586 SPRING 2020**  
**MIDTERM EXAM — DUE MAY 15 ON GITHUB BY 11PM**

Be sure to do a `git pull` to update your local version of the `amath-586-2020` repository.

This entire exam concerns system of ODEs called the *Toda lattice*. The system is defined using momenta  $p_j$ ,  $j = 0, \pm 1, \pm 2, \dots$  and positions  $q_j$ ,  $j = 0, \pm 1, \pm 2, \dots$ :

$$(1) \quad \begin{aligned} q'_j(t) &= p_j(t), \\ p'_j(t) &= e^{-(q_j(t)-q_{j-1}(t))} - e^{-(q_{j+1}(t)-q_j(t))}, \quad j = 0, \pm 1, \pm 2, \dots \end{aligned}$$

This is, as defined, an infinite-dimensional ODE system. We will consider finite-dimensional approximations.

---

**Problem 1:** Define new variables

$$(2) \quad \begin{aligned} a_j(t) &= \frac{1}{2}e^{-(q_{j+1}(t)-q_j(t))/2}, \\ b_j(t) &= -\frac{1}{2}p_j(t), \quad j = 0, \pm 1, \pm 2, \dots \end{aligned}$$

Using (2), first show that the Toda lattice (1) can be written as

$$(3) \quad \begin{aligned} a'_j(t) &= a_j(t)(b_{j+1}(t) - b_j(t)), \\ b'_j(t) &= 2(a_j^2(t) - a_{j-1}^2(t)), \quad j = 0, \pm 1, \pm 2, \dots \end{aligned}$$

Now consider the time-dependent tridiagonal matrices:

$$T(t) = \begin{bmatrix} \ddots & \ddots & & & \\ & b_{j-1}(t) & a_{j-1}(t) & & \\ & a_{j-1}(t) & b_j(t) & a_j(t) & \\ & & a_j(t) & b_{j+1}(t) & \ddots \\ & & & \ddots & \ddots \end{bmatrix}, \quad S(t) = \begin{bmatrix} \ddots & & \ddots & & \\ & 0 & a_{j-1}(t) & & \\ & -a_{j-1}(t) & 0 & a_j(t) & \\ & & -a_j(t) & 0 & \ddots \\ & & & \ddots & \ddots \end{bmatrix}$$

Now, show that (3) is equivalent to

$$(4) \quad T'(t) = S(t)T(t) - T(t)S(t).$$

Hint: Fix  $j$  and consider

$$e_j^T S(t) T(t) e_k \text{ and } e_j^T T(t) S(t) e_k, \quad k = j-2, j-1, j, j+1, j+2,$$

where  $e_j$  denotes the standard basis vector that is all zero except for a one in the  $j$ th entry. Despite the fact that the matrices are bi-infinite, you only need to track the  $(j-1)$ th,  $j$ th and  $(j+1)$ th entries of vectors such as  $e_j^T S(t)$  and  $T(t) e_k$ .

---

**Problem 2:** One finite-dimensional approximation of (4) is to just take a finite section (a square subblock on the diagonal) of both  $T$ ,  $S$ :

$$T_N(t) = \begin{bmatrix} b_1(t) & a_1(t) & & & \\ a_1(t) & b_2(t) & a_2(t) & & \\ & a_2(t) & b_3(t) & \ddots & \\ & & \ddots & \ddots & a_{N-1}(t) \\ & & & a_{N-1}(t) & b_N(t) \end{bmatrix},$$

$$S_N(t) = \begin{bmatrix} 0 & a_1(t) & & & \\ -a_1(t) & 0 & a_2(t) & & \\ & -a_2(t) & 0 & \ddots & \\ & & \ddots & \ddots & a_{N-1}(t) \\ & & & -a_{N-1}(t) & 0 \end{bmatrix}$$

The finite section choice can be understood by formally setting  $q_0 = -\infty$  and  $q_{N+1} = +\infty$  and then performing the change of variables (2).

With initial conditions  $b_j(0) = 0$ ,  $j = 1, 2, \dots, N$ ,  $a_j(0) = 1/2$ ,  $j = 1, 2, \dots, N-1$  and  $N = 6$ , use your favorite time-stepping method to solve

$$T'_N(t) = S_N(t)T_N(t) - T_N(t)S_N(t),$$

to  $t = 100$  and plot the solution. You should notice something striking about the solution. You might want to look at eigenvalues of  $T_N(0)$ . Comment on this. Repeat this with  $b_j(0) = -2$  and  $a_j(0) = 1$ ,  $j = 1, 2, \dots, N$  and  $N = 12$ .

**Problem 3:** The Toda lattice in the finite-section case is a Hamiltonian system with Hamiltonian

$$H(p, q) = \frac{1}{2}p_N^2 + \sum_{j=1}^{N-1} \left[ \frac{1}{2}p_j^2 + e^{-(q_{j+1}-q_j)} \right].$$

This means that the equations of motion for  $q_j(t)$  and  $p_j(t)$  can also be written as

$$(5) \quad \begin{aligned} p'_j(t) &= -\frac{\partial H}{\partial q_j}(p(t), q(t)), \\ q'_j(t) &= \frac{\partial H}{\partial p_j}(p(t), q(t)), \end{aligned}$$

where  $p(t) = (p_j(t))_{j=1}^N$  and  $q(t) = (q_j(t))_{j=1}^N$ . And, by the chain rule,  $H$  is conserved:

$$\frac{d}{dt}H(p(t), q(t)) = 0.$$

Symplectic numerical integrators for Hamiltonian systems are designed to preserve conserved quantities and geometric properties of systems they approximate. We can summarize the system (5) as

$$p'(t) = J(q(t)), \quad q'(t) = K(p(t)).$$

One symplectic method is the so-called Störmer–Verlet method and it is given by

$$\begin{aligned} P^* &= P^n + \frac{k}{2} J(Q^n), \\ Q^{n+1} &= Q^n + k K(P^*), \\ P^{n+1} &= P^* + \frac{k}{2} J(Q^{n+1}). \end{aligned}$$

Convert the initial data  $b_j(0) = 0$  and  $a_j(0) = 1/2$  for  $j = 1, 2, \dots, N$  to  $q_j(0), p_j(0)$  for  $j = 1, 2, \dots, N$  and solve the system with the Störmer–Verlet method. Perform a convergence study at  $t = 1$  for time steps  $k = 2^{-j}$ ,  $j = 1, 2, 3, 4, 5, 6$  (see <https://github.com/trogdoncourses/amath-586-2020/blob/master/notebooks/Astability.ipynb>) to determine the order of the method.

Some hints:

- Since the  $a_j, b_j$  variables depend only on a difference of the  $q_j$ . You can set one value, say  $q_1(0)$ , to be whatever value you wish.
- You also might want to write a function to convert between  $a_j, b_j$  and  $p_j, q_j$ . Here is a **Julia** implementation:

```
to_a = (p, q) -> .5 * exp .(-(q[2:end] - q[1:end-1]) / 2)
to_b = (p, q) -> -.5 * p
to_p = (a, b) -> -2 * b
function to_q(a, b) # chooses q[1] = 0
    q = fill(0., length(b))
    q[2:end] = -2 * log.(2 * a)
    cumsum(q)
end
```

- Here is a **Julia** implementation of  $J$  and  $K$ :

```
function J(q)
    out = fill(0., length(q))
    temp = exp.(q[1:end-1] - q[2:end])
    out[1:end-1] -= temp
    out[2:end] += temp
    out
end

function K(p)
    p
end
```

Just for your information: A second order method will satisfy:

$$\text{error at time } T \sim C_T k^2.$$

And the constant  $C_T$  is incredibly important as  $T$  increases. Symplectic methods can be used to keep  $C_T$  from growing too rapidly and they are very important in, say, planetary dynamics over long time scales.

---

**Problem 4:** (Extra credit) Form a finite-dimensional approximation of (1) using the boundary condition  $q_0(t) = q_{N+1}(t)$  and performing the change of variables (2). Update the matrices  $T_N$  and  $S_N$  for the periodic case.