Simple tests on iterative method with arclength parameterizing

Chen Ziheng

July 24, 2017

Part I
Test on a non-singular case

1 Arclength parameterization

Consider the Hamiltonian of the following diffusion-driven system

$$H = \frac{1}{2}\langle p, ap\rangle + \langle b, p\rangle . \tag{1}$$

If we apply arclength parameterizing, we will have

$$x' = \frac{C}{|b|_a}(ap + b), \tag{2}$$

$$p' = -\frac{C}{|b|_a}(\nabla b)^T p, \tag{3}$$

where C denotes the full length of the solution x.

2 An example

To begin with, we check a trivial case: a is identity, b in the form of $-Bx$, and B is normal for the time being. Then the analytical solution is clear:

$$p(t) = exp(B^T t)p_0, \tag{4}$$

$$x(t) = (B + B^T)^{-1}p(t), \tag{5}$$

here t takes value from $[0, +\infty)$

First we give a non-singular border condition: $p(0) = (4, 1)^T$ and $x(1) = (0.9993063, 0)^T$. The iterations terminates when the difference of adjacent two $p(1)$s is smaller than a given criterion.

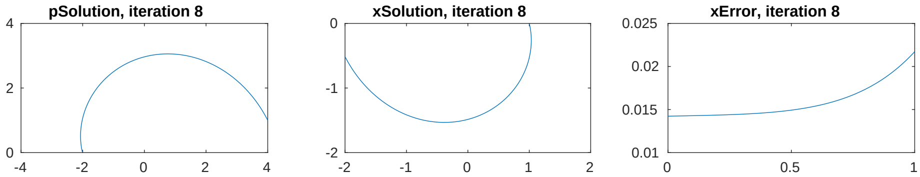


Figure 1: 8th Iteration Result

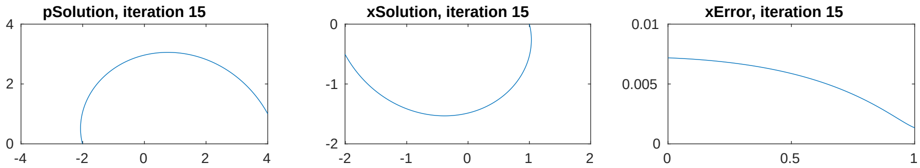


Figure 2: 15th Iteration Result

3 How to measure error

Besides, we also need to measure how the solution differs from the analytic one; take $resolution = 3, criterion = 10^{-6}$ as an exmaple (Figure 1 and 2). In the first several iterations, the error mainly distributes around $t = 1$; however, as the iterations going on, the error turns to gather around $t = 0$, which is the ending point of the iteration process. So we measure the difference where $t = 0$ to estimate the whole error.

4 Arclength-param result

The following chart shows how the error descends under arclength method; the error is indeed in first order to the resolution (so as the time needed).

Note: The precision is directly affected by the initial value. If we take $x(1) = (1, 0)$ rather than that magic number, the iterative method fails to achieve the corresponding precision at the resolution of 10^4 (for the fact that $1 - 0.9993063 > 10^{-4}$).

5 Normal method result

As for normal method, the result is quite boring: it converges after just two iterations! But that cannot prove any point, since in this case p only relies on itself

$$\frac{dp}{dt} = B^T p$$

, so the p part is precise, which can hardly happen in more complex cases.

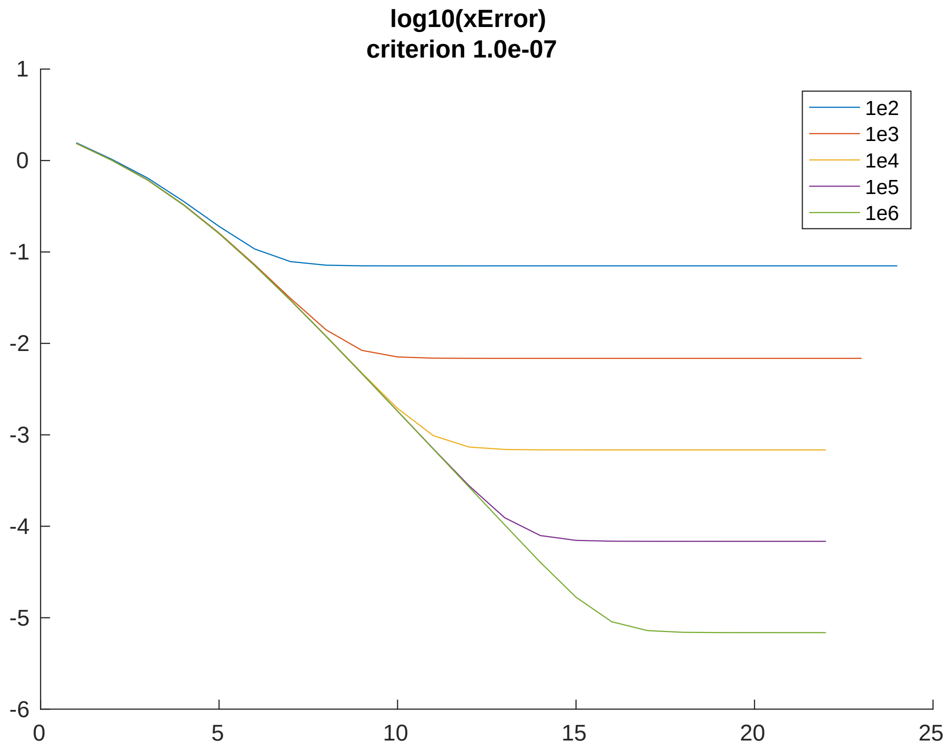


Figure 3: Error of Arclength Parameterizing Method

Part II
Failure on a singular case

6 Arclength-param result

If we change the border condition of the equations (2) to $p(0) = (4,1)^T$ and $x(1) = (0,0)^T$, the singularity of $\frac{dx}{ds}$ at $s = 1(x = 0)$ will emerge immediately. If we ignore the singulartiy for the time being, just taking any finite number as the derivative the singular point, the solution will diverge quickly, let alone converge at a first order speed. So direct use of iterative method may be too reckless to get a working effect.
The following chart shows the failure even at the first iterative round.

7 A maybe reason

The failure arises from the fact that when x starts from 0, so as $b(x)$. We can observe from the first equation

$$x' = \frac{C}{|b|_a}(ap + b)$$

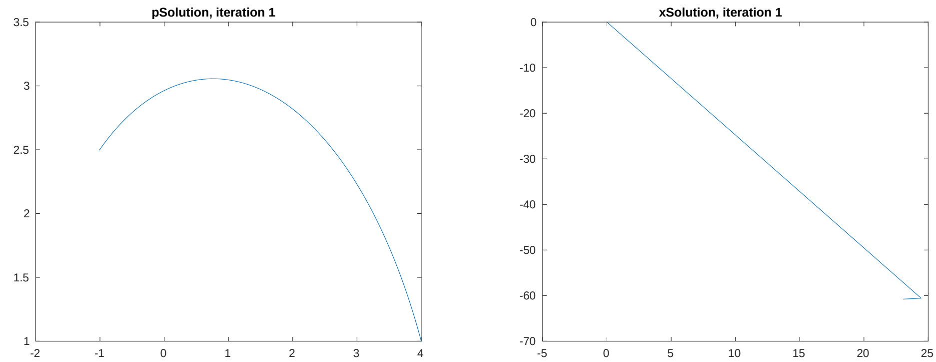


Figure 4: Failure when using iterative method directly

that $(ap + b)$ is supposed to be very near to 0, which can hardly be satisfied in the first iterative round. So $\frac{dx}{ds}$ gets very big in the first two iteration rounds, and the situation doesn't get better in the following rounds. To avoid such problem, we give an alternative way to solve the equation.

Part III
Improvement based on a naive idea

8 Result of the improved method

Since $\frac{dx}{ds}$ have singularity at $x = 0$ if p is not enough small, a simple improvement comes up very naturally: the initial condition varies in every iteration round, but it will converge to the stable point (int this case $(0, 0)^T$). Specifically, we use the following border condition:

$$p(0) = (4, 1)^T \tag{6}$$

$$x^{(k)}(1) = (0, 0)^T + (1/k, 0)^T \tag{7}$$

here k denotes the iteration round. The form of the converging initial points is not so important, since only its converging rate matters.

The following chart shows how the improved method works.

The two figures lead to some interesting conculsions:

1. As the initial condition is far from the final result, the iterative method still works (but that may largely depends on the fineness of the equations).

2. The error of the p part increases steadily as the length s, while the error of x is much larger.

3. Notice that we set the termination condition as $||p^{(k+1)} - p^{(k)}|| < criterion = 10^{-6}$, but the actual error is about 5×10^{-3}, not at the same magnitude (see figure 7). A similar situation happens at the x side.

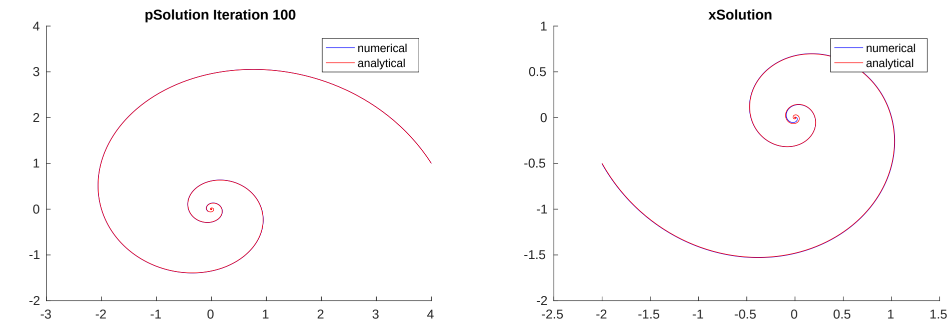


Figure 5: A demonstration of varing initial condition method

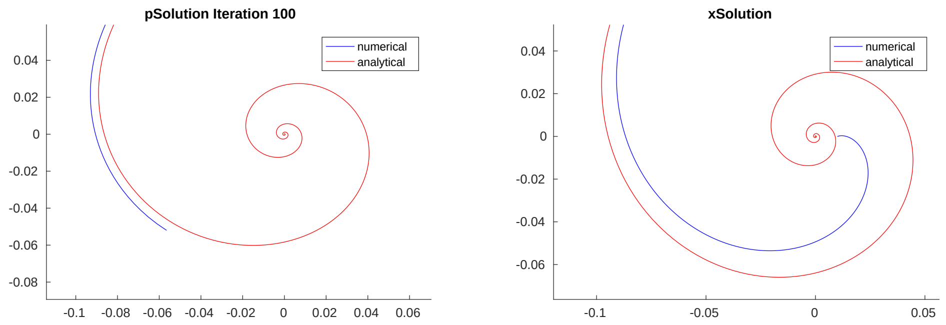


Figure 6: A close lookup near the stable point

9 An attempt on quicker convergence

We should notice that in the border condition mentioned before, the convergence rate is rather low; in fact, it will nearly stop if the iteration goes on. So a very natural idea is if we can 'speed up' the iteration by choosing a faster converging initial condition:

$$x^{(k)}(1) = max\{2^{(1-k)}, 1/resolution\}.$$

The intermediate result at 18th iteration shows in figure 8.

The main problem lies in that the solution has no time to converge before the initial condition gets too small, so we have the following improvement.

10 Criterion selection

Before we carry out any numerical experiment, the criterion of convergence should be carefully taken. Here we test over criterions with different orders. It is clear that $10/resolution$ is quite enough for precision, and $0.1/resolution$ may be a better choice.

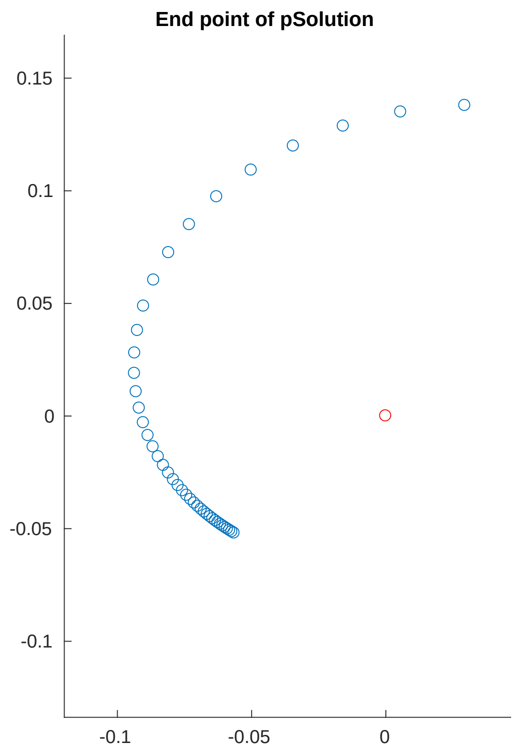


Figure 7: $p(1)$ spires in and converges very slowly

11 An improvement on initial condition

Since in section 9 we have found that the solution diverges because the initial condition converges too quicklly, we can 'wait' for the solution to converge before the initial condition get closer to the stable point. To write it explicitly, we have

$$x^{(k+1)}(1) = \begin{cases} x^{(k)}(1)/2 & if \quad ||p^{(k+1)}(1) - p^{(k)}(1)|| < criterion \\ x^{(k)}(1) & otherwise \end{cases} \tag{8}$$

Four cases are tested, where $resolution \in \{10^3, 10^4, 10^5, 10^6\}$, with $criterion = 0.1/resolution$.

As figure 11 shows, error of numerical solution (measured by $||p(1)||$) is indeed in first order. If we measure the error in phase space (I wonder if it is more important), the result is shown in the figure, similar as the previous result.

It may be quite interesting to have a look at how long these iterations take, in terms of round numbers. Results are shown in table 1. It is intriguing that every even initial condition round has just one iteration. It seems that it takes exponentially time as the resolution increases.

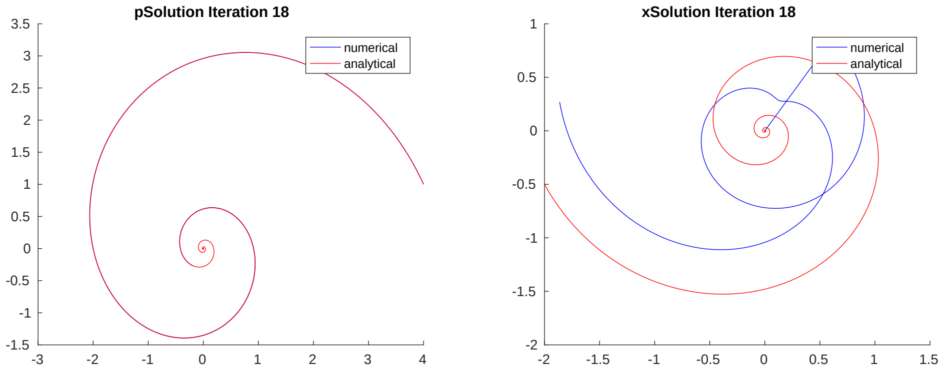


Figure 8: An unsuccessful example of speed up convergence

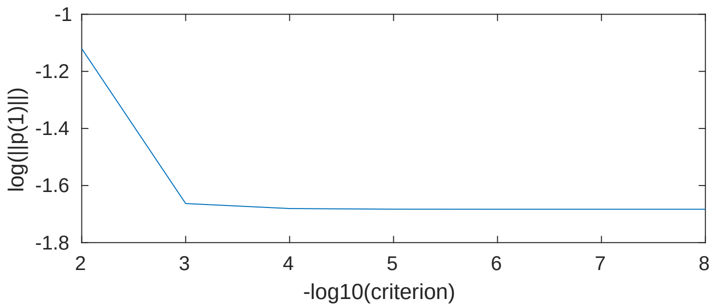


Figure 9: Error in p under different criterion conditions, $resolution = 10^4$

Part IV
Further more

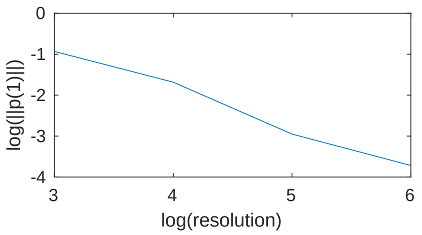
12 Using high-order schemes

Since the convergence is in first order, we wonder if the method can converge much faster with high-order schemes. However, according to result shown in figure 10, if we use improved Euler method in each iteration round, the convergence rate is still in first order.

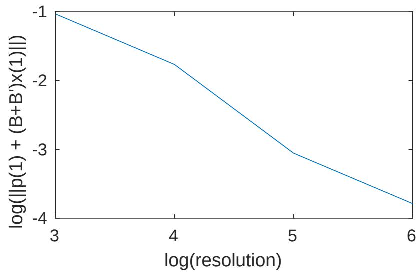
13 Tests on higher-order equations

A random normal matrix is chosen to test if the method is robust in higher dimension cases. Here, we select B to be

$$\begin{bmatrix} -0.9456 & 0.4262 & 0.2393 \\ 0.4262 & -0.9631 & -0.0686 \\ 0.2393 & -0.0686 & -1.0911 \end{bmatrix}$$



(a) Error of $p(1)$



(b) Error of $p(1)+(B+B^T)x(1)$ in phase space

Table 1: Iterations used

$\log 2(x^{(k)}(1))$	Resolution			
	10^3	10^4	10^5	10^6 *(criterion* $= 10^{-6}$*)*
0	14	17	19	19
-1	1	1	1	1
-2	15	19	23	23
-3	1	1	1	1
-4	39	45	52	52
-5	1	1	1	1
-6	74	76	87	87
-7	0	1	1	1
-8	0	105	126	126
-9	0	1	1	1
-10	0	0	154	163
-11	0	0	1	1
-12	0	0	186	202
-13	0	0	1	1
-14	0	0	0	7
-15	0	0	0	203
-16	0	0	0	1

which has three real eigenvalues. The resulting is quite confusing, because convergence emerges after at most 1300 rounds, when resolution is set to be 10^5. However, the result of $resolution = 10^3, 10^4$ is to satisfactory; see figure 13

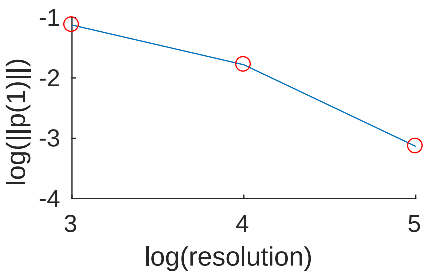
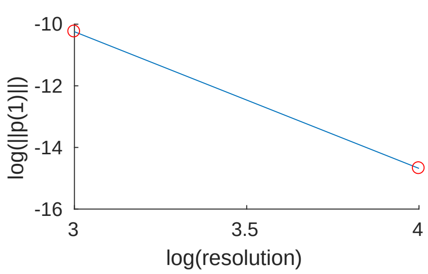
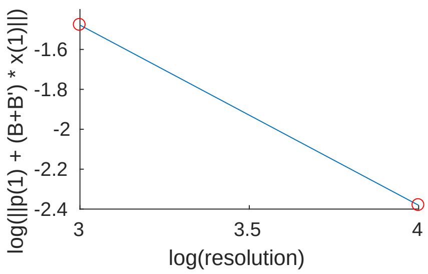


Figure 10: Error of $p(1)$, with each iteration using improved Euler method



(a) Error of $p(1)$



(b) Error of $p(1)+(B+B^T)x(1)$ in phase space