

FreeSandal

樹莓派, 樹莓派之學習, 樹莓派之教育

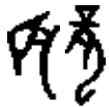
STEM 隨筆：古典力學：動力學【五·五·乙】

2018-05-18 | 懸鉤子 | 發表迴響



遯：亨，小利貞。

彖曰：遯亨，遯而亨也。剛當位而應，與時行也。小利貞，浸而長也。遯之時義大矣哉！



象曰：天下有山，遯；君子以遠小人，不惡而嚴。

☶：《遯》易曰：

六二，執之用黃牛之革，莫之勝說。

象曰：執用黃牛，固志也。

山高侵昊 天，時 厲，難與行。何德能嘉遯？志固執用黃牛革。傲來有石

猴，獨尊是佛陀，如來血脈一滴傳，因而敢登唯我峰。臨 事心踟躕，意緒無處寄

《晚晴》李商隱



深居俯夾城，春去夏猶清。
天意憐幽草，人間重晚晴。
並添高閣迴，微注小窗明。
越鳥巢乾後，歸飛體更輕。

，總是時到時擔當。

...

，真不知那吃  蟹的第一人，如何想來，又怎麼下得了口！？噗哧一笑，陡地，

 貓的理論上心頭：

通常人們都說不要『迷信』，那『科學』自己會不會也『變成』一種迷信呢？或者說所謂的科學又是『什麼』呢？因為並非一直以來科學就是『像今天』一樣，現今的科學有著這樣的『一種精神』：

- 一、事實立論
- 二、在事實的基礎上，建立用來解說的假設，然後形成它的理論
- 三、人人時時方方都可實驗，嘗試驗證或者推翻一與二之所說
- 四、保持著懷疑、設想新現象、創革舊工具；再次持續不斷的思想推翻一、二和三之所言。

不知這樣的精神能不能有『東』『西』之分？還是有『南』『北』之別呢？

大科學家牛頓養了一隻貓，他看到那貓因為進出門戶不方便而不『快樂』，所以在門上打了個洞，果然貓就快樂了起來。多年之後那貓生了小貓，牛頓很高興的在那個洞的旁邊又打了一個小洞，這樣小貓也一定會很『快樂』的了。真不知牛頓如何想出這個『貓的理論』：

大貓走大洞；小貓走小洞。

難道『小貓』就不可以走『大洞』？還是不這樣『小貓』就會不『快樂』？？

，何不效法牛頓，『大』洞『小』洞『快樂』的打，一『洞』通了是『一』洞，早晚總能『洞穿』！！♀😏

.....

 訊：宛如那一株

小草

作詞：林建助

作曲：陳輝雄

大風起 把頭搖一搖
風停了 又挺直腰
大雨來 彎著背 讓雨澆
雨停了 抬起頭 站直腳

不怕風 不怕雨 立志要長高
小草 實在是 並不小

。♀

—— 《MJO之學習筆記本《辰》組元：【言】黃牛之革》

如果前人吃過了螃蟹，後來者是否打算效法那傳說中的第一人，重新嘗試一次乎？倘不親身體驗，真能無所遺漏耶？？

探險者請注意

一人、一時、一地之『獨立事件』，通常『數理』並不處理也！

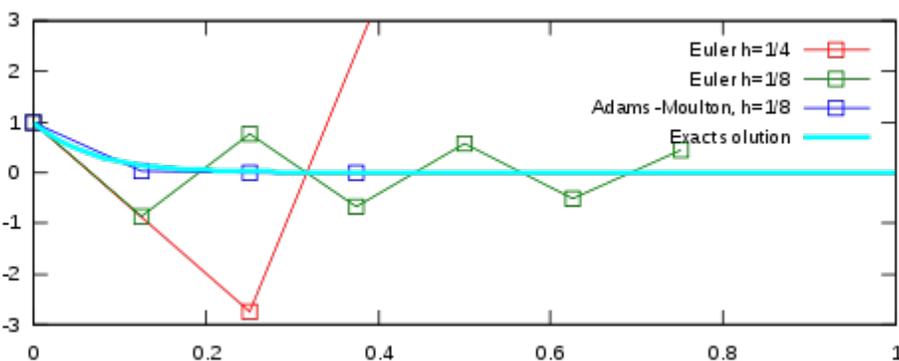
蓋無獨有偶，方可視其『現象』之所以哩！！

Stiff equation

In mathematics, a **stiff equation** is a differential equation for which certain numerical methods for solving the equation are numerically unstable, unless the step size is taken to be extremely small. It has proven difficult to formulate a precise definition of stiffness, but the main idea is that the equation includes some terms that can lead to rapid variation in the solution.

When integrating a differential equation numerically, one would expect the requisite step size to be relatively small in a region where the solution curve displays much variation and to be relatively large where the solution curve straightens out to approach a line with slope nearly zero. For some problems this is not the case. Sometimes the step size is forced down to an unacceptably small level in a region where the solution curve is very smooth. The phenomenon being exhibited here is known as **stiffness**. In some cases we may have two different problems with the same solution, yet problem one is *not* stiff and problem two *is* stiff. Clearly the phenomenon cannot be a property of the exact solution, since this is the same for both problems, and must be a property of the differential system itself. It is thus appropriate to speak of *stiff systems*.

Motivating example



Explicit numerical methods exhibiting instability when integrating a stiff ordinary differential equation

Consider the initial value problem

$$y'(t) = -15y(t), \quad t \geq 0, y(0) = 1.$$

--	--	--

(1)

The exact solution (shown in cyan) is

$$y(t) = e^{-15t} \text{ with } y(t) \rightarrow 0 \text{ as } t \rightarrow \infty.$$

--	--	--

(2)

We seek a numerical solution that exhibits the same behavior.

The figure (right) illustrates the numerical issues for various numerical integrators applied on the equation.

1. Euler's method with a step size of $h = 1/4$ oscillates wildly and quickly exits the range of the graph (shown in red).
2. Euler's method with half the step size, $h = 1/8$, produces a solution within the graph boundaries, but oscillates about zero (shown in green).
3. The trapezoidal method (that is, the two-stage Adams–Moulton method) is given by

$$y_{n+1} = y_n + \frac{1}{2}h (f(t_n, y_n) + f(t_{n+1}, y_{n+1})),$$

(3)

where $y' = f(t, y)$. Applying this method instead of Euler's method gives a much better result (blue). The numerical results decrease monotonically to zero, just as the exact solution does.

One of the most prominent examples of the stiff ODEs is a system that describes the chemical reaction of Robertson:

$$\dot{x} = -0.04x + 10^4 y \cdot z$$

$$\dot{y} = 0.04x - 10^4 y \cdot z - 3 \cdot 10^7 y^2$$

$$\dot{z} = 3 \cdot 10^7 y^2$$

--	--

If one treats this system on a short interval, for example, $t \in [0, 40]$ there is no problem in numerical integration. However, if the interval is very large (10^{11} say), then many standard codes fail to integrate it correctly.

Additional examples are the sets of ODEs resulting from the temporal integration of large chemical reaction mechanisms. Here, the stiffness arises from the coexistence of very slow and very fast reactions. To solve them, the software packages KPP and Autochem can be used.

Stiffness ratio

Consider the linear constant coefficient inhomogeneous system

$\mathbf{y}' = \mathbf{A}\mathbf{y} + \mathbf{f}(x),$	<table border="1" style="width: 100%; height: 30px;"> <tr> <td style="width: 33%;"></td> <td style="width: 33%;"></td> <td style="width: 33%;"></td> </tr> </table>				(5)

where $\mathbf{y}, \mathbf{f} \in \mathbb{R}^n$ and \mathbf{A} matrix with eigenvalues $\lambda_t \in \mathbb{C}, t = 1, 2, \dots, n$ (assumed distinct) and corresponding eigenvectors $\mathbf{c}_t \in \mathbb{C}^n, t = 1, 2, \dots, n$. The general solution of (5) takes the form

$\mathbf{y}(x) = \sum_{t=1}^n \kappa_t \exp(\lambda_t x) \mathbf{c}_t + \mathbf{g}(x),$	<table border="1" style="width: 100%; height: 30px;"> <tr> <td style="width: 33%;"></td> <td style="width: 33%;"></td> <td style="width: 33%;"></td> </tr> </table>				(6)

where the κ_t are arbitrary constants and $\mathbf{g}(x)$ is a particular integral. Now let us suppose that

$Re(\lambda_t) < 0, \quad t = 1, 2, \dots, n,$	<table border="1" style="width: 100%; height: 30px;"> <tr> <td style="width: 33%;"></td> <td style="width: 33%;"></td> <td style="width: 33%;"></td> </tr> </table>				(7)

which implies that each of the terms $\exp(\lambda_t x) \mathbf{c}_t \rightarrow 0$ as $x \rightarrow \infty$, so that the solution $\mathbf{y}(x)$ approaches $\mathbf{g}(x)$ asymptotically as $x \rightarrow \infty$; the term $\exp(\lambda_t x) \mathbf{c}_t$ will decay monotonically if λ_t is real and sinusoidally if λ_t is complex. Interpreting x to be time (as it often is in physical problems) it is appropriate to call $\sum_{t=1}^n \kappa_t \exp(\lambda_t x) \mathbf{c}_t$ the **transient solution** and $\mathbf{g}(x)$ the **steady-state solution**. If $|Re(\lambda_t)|$ is large, then the corresponding term $\kappa_t \exp(\lambda_t x) \mathbf{c}_t$ will

decay quickly as x increases and is thus called a **fast transient**; if $|Re(\lambda_t)|$ is small, the corresponding term $\kappa_t \exp(\lambda_t x) \mathbf{c}_t$ decays slowly and is called a **slow transient**. Let $\bar{\lambda}, \underline{\lambda} \in \{\lambda_t, t = 1, 2, \dots, n\}$ be defined by

$ Re(\bar{\lambda}) \geq Re(\lambda_t) \geq Re(\underline{\lambda}) , \quad t = 1, 2, \dots, n$	<table border="1" style="margin: auto;"> <tr> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> <td style="width: 20px; height: 20px;"></td> </tr> </table>				(8)

so that $\kappa_t \exp(\bar{\lambda} x) \mathbf{c}_t$ is the fastest transient and $\kappa_t \exp(\underline{\lambda} x) \mathbf{c}_t$ the slowest. We now define the **stiffness ratio** as

$\frac{ Re(\bar{\lambda}) }{ Re(\underline{\lambda}) } \cdot [1]$	<table border="1" style="margin: auto;"> <tr> <td style="width: 30px; height: 20px;"></td> <td style="width: 30px; height: 20px;"></td> <td style="width: 30px; height: 20px;"></td> </tr> </table>				(9)

Characterization of stiffness

In this section we consider various aspects of the phenomenon of stiffness. ‘Phenomenon’ is probably a more appropriate word than ‘property’, since the latter rather implies that stiffness can be defined in precise mathematical terms; it turns out not to be possible to do this in a satisfactory manner, even for the restricted class of linear constant coefficient systems. We shall also see several qualitative statements that can be (and mostly have been) made in an attempt to encapsulate the notion of stiffness, and state what is probably the most satisfactory of these as a ‘definition’ of stiffness.

J. D. Lambert defines stiffness as follows:

*If a numerical method with a finite region of absolute stability, applied to a system with any initial conditions, is forced to use in a certain interval of integration a steplength which is excessively small in relation to the smoothness of the exact solution in that interval, then the system is said to be **stiff** in that interval.*

There are other characteristics which are exhibited by many examples of stiff problems, but for each there are counterexamples, so these characteristics do not make good definitions of stiffness. Nonetheless, definitions based upon these characteristics are in common use by some authors and are good clues as to the presence of stiffness. Lambert refers to these as

'statements' rather than definitions, for the aforementioned reasons. A few of these are:

1. A linear constant coefficient system is stiff if all of its eigenvalues have negative real part and the stiffness ratio is large.
2. Stiffness occurs when stability requirements, rather than those of accuracy, constrain the steplength.
3. Stiffness occurs when some components of the solution decay much more rapidly than others.^[2]

Etymology

The origin of the term 'stiffness' seems to be somewhat of a mystery. According to Joseph Oakland Hirschfelder, the term 'stiff' is used because such systems correspond to tight coupling between the driver and driven in servomechanisms.^[3] According to Richard. L. Burden and J. Douglas Faires,

Significant difficulties can occur when standard numerical techniques are applied to approximate the solution of a differential equation when the exact solution contains terms of the form $e^{\lambda t}$, where λ is a complex number with negative real part.

...

*Problems involving rapidly decaying transient solutions occur naturally in a wide variety of applications, including the study of spring and damping systems, the analysis of control systems, and problems in chemical kinetics. These are all examples of a class of problems called **stiff** (mathematical stiffness) **systems** of differential equations, due to their application in analyzing the motion of spring and mass systems having large spring constants (physical stiffness).^[4]*

For example, the initial value problem

$$m\ddot{x} + c\dot{x} + kx = 0, \quad x(0) = x_0, \quad \dot{x}(0) = 0,$$



(10)

with $m = 1$, $c = 1001$, $k = 1000$, can be written in the form (5) with $n = 2$ and

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ -1000 & -1001 \end{pmatrix},$$

--	--	--

(11)

$$\mathbf{f}(t) = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

--	--	--

(12)

$$\mathbf{x}(0) = \begin{pmatrix} x_0 \\ 0 \end{pmatrix},$$

--	--	--

(13)

and has eigenvalues $\bar{\lambda} = -1000, \underline{\lambda} = -1$. Both eigenvalues have negative real part and the stiffness ratio is

$$\frac{|-1000|}{|-1|} = 1000,$$

--	--	--

(14)

which is fairly large. System (10) then certainly satisfies statements 1 and 3. Here the spring constant k is large and the damping constant c is even larger.^[5] (Note that 'large' is a vague, subjective term, but the larger the above quantities are, the more pronounced will be the effect of stiffness.) The exact solution to (10) is

$$x(t) = x_0 \left(-\frac{1}{999} e^{-1000t} + \frac{1000}{999} e^{-t} \right) \approx x_0 e^{-t}.$$

--	--	--

(15)

Note that (15) behaves quite nearly as a simple exponential $x_0 e^{-t}$, but the presence of the e^{-1000t} term, even with a small coefficient is enough to make the numerical computation very sensitive to step size. Stable integration of (10) requires a very small step size until well into the smooth part of the solution curve, resulting in an error much smaller than required for accuracy. Thus the system also satisfies statement 2 and Lambert's definition.

A-stability

The behaviour of numerical methods on stiff problems can be analyzed by applying these methods to the test equation $y' = ky$ subject to the initial condition $y(0) = 1$ with $k \in \mathbb{C}$. The solution of this equation is $y(t) = e^{kt}$. This solution approaches zero as $t \rightarrow \infty$ when $\operatorname{Re}(k) < 0$. If the numerical method also exhibits this behaviour (for a fixed step size), then the method is said to be A-stable.^[6] (Note that a numerical method that is L-stable (see below) has the stronger property that the solution approaches zero in a single step as the step size goes to infinity.) A-stable methods do not exhibit the instability problems as described in the motivating example.

唯因科學一再觀注當下『理論假設』，但求『證偽』耳◎

Levenberg–Marquardt algorithm

In mathematics and computing, the **Levenberg–Marquardt algorithm (LMA or just LM)**, also known as the **damped least-squares (DLS)** method, is used to solve non-linear least squares problems. These minimization problems arise especially in least squares curve fitting.

The LMA is used in many software applications for solving generic curve-fitting problems. However, as with many fitting algorithms, the LMA finds only a local minimum, which is not necessarily the global minimum. The LMA interpolates between the Gauss–Newton algorithm (GNA) and the method of gradient descent. The LMA is more robust than the GNA, which means that in many cases it finds a solution even if it starts very far off the final minimum. For well-behaved functions and reasonable starting parameters, the LMA tends to be a bit slower than the GNA. LMA can also be viewed as Gauss–Newton using a trust region approach.

The algorithm was first published in 1944 by Kenneth Levenberg,^[1] while working at the Frankford Army Arsenal. It was rediscovered in 1963 by Donald Marquardt^[2] who worked as a statistician at DuPont and independently by Girard,^[3] Wynne^[4] and Morrison.^[5]

The problem

The primary application of the Levenberg–Marquardt algorithm is in the least-squares curve

fitting problem: given a set of m empirical datum pairs (x_i, y_i) of independent and dependent variables, find the parameters $\boldsymbol{\beta}$ of the model curve $f(x, \boldsymbol{\beta})$ so that the sum of the squares of the deviations $S(\boldsymbol{\beta})$ is minimized:

$$\hat{\boldsymbol{\beta}} = \operatorname{argmin}_{\boldsymbol{\beta}} S(\boldsymbol{\beta}) \equiv \operatorname{argmin}_{\boldsymbol{\beta}} \sum_{i=1}^m [y_i - f(x_i, \boldsymbol{\beta})]^2.$$

The solution

Like other numeric minimization algorithms, the Levenberg–Marquardt algorithm is an iterative procedure. To start a minimization, the user has to provide an initial guess for the parameter vector, $\boldsymbol{\beta}$. In cases with only one minimum, an uninformed standard guess like $\boldsymbol{\beta}^T = (1, 1, \dots, 1)$ will work fine; in cases with multiple minima, the algorithm converges to the global minimum only if the initial guess is already somewhat close to the final solution.

In each iteration step, the parameter vector $\boldsymbol{\beta}$ is replaced by a new estimate $\boldsymbol{\beta} + \boldsymbol{\delta}$. To determine $\boldsymbol{\delta}$, the function $f(x_i, \boldsymbol{\beta} + \boldsymbol{\delta})$ is approximated by its linearization:

$$f(x_i, \boldsymbol{\beta} + \boldsymbol{\delta}) \approx f(x_i, \boldsymbol{\beta}) + J_i \boldsymbol{\delta},$$

where

$$J_i = \frac{\partial f(x_i, \boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$$

is the gradient (row-vector in this case) of f with respect to $\boldsymbol{\beta}$.

The sum $S(\boldsymbol{\beta})$ of square deviations has at its minimum a zero gradient with respect to $\boldsymbol{\beta}$. The above first-order approximation of $f(x_i, \boldsymbol{\beta} + \boldsymbol{\delta})$ gives

$$S(\boldsymbol{\beta} + \boldsymbol{\delta}) \approx \sum_{i=1}^m (y_i - f(x_i, \boldsymbol{\beta}) - J_i \boldsymbol{\delta})^2,$$

or in vector notation,

$$\begin{aligned}
S(\boldsymbol{\beta} + \boldsymbol{\delta}) &\approx \|\mathbf{y} - \mathbf{f}(\boldsymbol{\beta}) - \mathbf{J}\boldsymbol{\delta}\|^2 \\
&= [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta}) - \mathbf{J}\boldsymbol{\delta}]^T [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta}) - \mathbf{J}\boldsymbol{\delta}] \\
&= [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})]^T [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})] - [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})]^T \mathbf{J}\boldsymbol{\delta} - (\mathbf{J}\boldsymbol{\delta})^T [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})] + \boldsymbol{\delta}^T \mathbf{J}^T \mathbf{J}\boldsymbol{\delta} \\
&= [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})]^T [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})] - 2[\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})]^T \mathbf{J}\boldsymbol{\delta} + \boldsymbol{\delta}^T \mathbf{J}^T \mathbf{J}\boldsymbol{\delta}.
\end{aligned}$$

Taking the derivative of $S(\boldsymbol{\beta} + \boldsymbol{\delta})$ with respect to $\boldsymbol{\delta}$ and setting the result to zero gives

$$(\mathbf{J}^T \mathbf{J})\boldsymbol{\delta} = \mathbf{J}^T [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})],$$

where \mathbf{J} is the Jacobian matrix whose i -th row equals $J_i \mathbf{J}_i$, and where \mathbf{f} and \mathbf{y} are vectors with i -th component $f(x_i, \boldsymbol{\beta})$ and y_i respectively. This is a set of linear equations, which can be solved for $\boldsymbol{\delta}$.

Levenberg's contribution is to replace this equation by a "damped version",

$$(\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I})\boldsymbol{\delta} = \mathbf{J}^T [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})],$$

where \mathbf{I} is the identity matrix, giving as the increment $\boldsymbol{\delta}$ to the estimated parameter vector $\boldsymbol{\beta}$.

The (non-negative) damping factor λ is adjusted at each iteration. If reduction of S is rapid, a smaller value can be used, bringing the algorithm closer to the Gauss-Newton algorithm, whereas if an iteration gives insufficient reduction in the residual, λ can be increased, giving a step closer to the gradient-descent direction. Note that the gradient of S with respect to $\boldsymbol{\delta}$ equals $-2(\mathbf{J}^T [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})])^T$. Therefore, for large values of λ , the step will be taken approximately in the direction of the gradient. If either the length of the calculated step $\boldsymbol{\delta}$ or the reduction of sum of squares from the latest parameter vector $\boldsymbol{\beta} + \boldsymbol{\delta}$ fall below predefined limits, iteration stops, and the last parameter vector $\boldsymbol{\beta}$ is considered to be the solution.

Levenberg's algorithm has the disadvantage that if the value of damping factor λ is large, inverting $\mathbf{J}^T \mathbf{J} + \lambda \mathbf{I}$ is not used at all. Marquardt provided the insight that we can scale each component of the gradient according to the curvature, so that there is larger movement along the directions where the gradient is smaller. This avoids slow convergence in the direction of small gradient. Therefore, Marquardt replaced the identity matrix \mathbf{I} with the diagonal matrix consisting of the diagonal elements of $\mathbf{J}^T \mathbf{J}$, resulting in the Levenberg-Marquardt algorithm:

$$[\mathbf{J}^T \mathbf{J} + \lambda \text{diag}(\mathbf{J}^T \mathbf{J})]\boldsymbol{\delta} = \mathbf{J}^T [\mathbf{y} - \mathbf{f}(\boldsymbol{\beta})].$$

A similar damping factor appears in Tikhonov regularization, which is used to solve linear ill-posed problems, as well as in ridge regression, an estimation technique in statistics.

Choice of damping parameter

Various more or less heuristic arguments have been put forward for the best choice for the damping parameter λ . Theoretical arguments exist showing why some of these choices guarantee local convergence of the algorithm; however, these choices can make the global convergence of the algorithm suffer from the undesirable properties of steepest descent, in particular very slow convergence close to the optimum.

The absolute values of any choice depends on how well-scaled the initial problem is.

Marquardt recommended starting with a value λ_0 and a factor $\nu > 1$. Initially setting $\lambda = \lambda_0$ and computing the residual sum of squares $S(\boldsymbol{\beta})$ after one step from the starting point with the damping factor of $\lambda = \lambda_0$ and secondly with λ_0/ν . If both of these are worse than the initial point, then the damping is increased by successive multiplication by ν until a better point is found with a new damping factor of $\lambda_0\nu^k$ for some k .

If use of the damping factor λ/ν results in a reduction in squared residual, then this is taken as the new value of λ (and the new optimum location is taken as that obtained with this damping factor) and the process continues; if using λ/ν resulted in a worse residual, but using λ resulted in a better residual, then λ is left unchanged and the new optimum is taken as the value obtained with λ as damping factor.

