Notes for MATH 4665/4875/7140/7300, Fall 2019, HKBU

Location for classes: LMC 514, HKBU

Time for classes: Wednesdays 15:30–16:20; Thursdays 13:30–15:20

Instructor: Prof. Tim Sheng

FUNDAMENTALS FOR FINANCE

C. Numerical Solution of PDEs.

C.06 Systems of Differential Equations.

......

Suppose that we want to solve the following linear financial system

$$u'(t) = Au(t) + g, \tag{2.22}$$

together with an initial condition

$$u(0) = u_0,$$

where

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & & \ddots & \vdots \\ a_{n-1,1} & a_{n-1,2} & & a_{n-1,n} \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix} \in \mathbb{R}^{n\times n}, \quad u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-1} \\ u_n \end{bmatrix} \in \mathbb{R}^n, \quad n \gg 1.$$

Vectors and matrices are not numbers. They are arrays of numbers. They are rich in many properties for applications. Herewith, let us recall some of the basics.

In addition to vector and matrix norms we have studied, there are a number of important issues we must keep in mind.

1. Frequently used special matrices in financial engineering
   - (a) Identity matrices
   - (b) Tridiagonal matrices
   - (c) Triangular matrices
   - (d) Symmetric matrices
   - (e) Toeplitz matrices (can be full matrices)
   - (f) TST (Tridiagonal + Symmetric + Toeplitz) matrices
   - (g) Positive-definite matrices

     Definition 2.2 (positive definite matrices).
     A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is said to be positive definite if the scalar

     $$\mu = v^\intercal Av$$

     is strictly positive for every nonzero vector $v \in \mathbb{R}^n$.

(h) Normal matrices

Definition 2.3 (normal matrices).
We say that $A \in \mathbb{R}^{n\times n}$ is normal if and only if when

$$A^\intercal A = AA^\intercal.$$

We may rewrite the above to

$$[A, A^\intercal] \; = \; A^\intercal A - AA^\intercal \; = \; \Phi,$$

where $\Phi \in \mathbb{R}^{n\times n}$ is a zero matrix.

We refer the operation $[A, B]$ as the *commutator* of A and B, or the *Lie bracket* of A and B. Here A, B can be $n \times n$ matrices, or any other linear operators. Lie brackets are used frequently in the field of differential topology.

We also say that A, B commute if $[A, B] = \Phi$. So for a normal matrix A, it commutes with its transpose.

(i) Band matrices

(j) Sparse matrices

(k) Matrix functions

$$B = f(A) = \sum_{i=0}^{\infty} \frac{1}{i!} f^{(i)}(\Phi) A^i \in \mathbb{R}^{n\times n},$$

where Φ is a zero matrix.

2. Singularity

Definition 2.4 (matrix singularity).
We say that $A \in \mathbb{R}^{n\times n}$ is singular if and only if when

$$\det(A) = 0.$$

3. Eigenvalues and eigenvectors

A nonzero vector $v \in \mathbb{C}^n$ of is an eigenvector of $A \in \mathbb{R}^{n\times n}$ if it satisfies the linear equation

$$Av = \lambda v, \tag{2.23}$$

where $\lambda \in \mathbb{C}$ is a scalar, and is referred as the eigenvalue of A corresponding to v.

Rewrite (2.23) to

$$(A - \lambda I)v = 0.$$

Thus, the condition to ensure that the above linear system to have nonzero solution v is that

$$\det(M) = \det(A - \lambda I) = 0. \tag{2.24}$$

But it can be shown that

$$\det(A - \lambda I) = p_n(\lambda)$$

is a degree n real coefficient polynomial of λ. Therefore to solve (2.24) is equivalent to find all roots, some of them with certain multiplicities, from the degree n polynomial

$$p_n(\lambda) = a_n\lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0\,.$$

This is apparently the same as the solution procedure for

$$a_n(\lambda - \lambda_1)^{n_1}(\lambda - \lambda_2)^{n_2}\cdots(\lambda - \lambda_m)^{n_m} = 0,$$

where

$$\begin{aligned} 1 \;\le\; m \;&\le\; n, \\ n_1 + n_2 + \cdots + n_m \;&=\; n. \end{aligned}$$

Therefore we are guaranteed to have eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$ with multiplicities $n_1, n_2, \ldots, n_m$. Again, some of the eigenvalues can be complex, although A is a real matrix.

For each eigenvalue λ_i, we have a specific eigen-equation

$$(A - \lambda_i I)v = 0, \quad i = 1, 2, \ldots, m.$$

There will be m_i, $1 \le m_i \le n_i$, linearly independent solutions to each eigen-equation. The linear combinations of such m_i solutions are the eigenvectors associated with the eigenvalue λ_i. We frequently call m_i as the geometric multiplicity of λ_i.

Theorem 2.2 (matrix singularity).
Matrix $A \in \mathbb{R}^{n\times n}$ is singular if and only if zero is one of its eigenvalues.

Theorem 2.3 (matrix singularity).
A matrix $A \in \mathbb{R}^{n\times n}$ is nonsingular if and only if there exists a matrix $B \in \mathbb{R}^{n\times n}$ such that

$$AB = BA = I,$$

where I denotes the $n\times n$ identity matrix and the multiplication used is ordinary matrix multiplication.

In the above case, the matrix B is uniquely determined by A and is called the inverse of A, denoted by A^{-1}.

Singular matrix is not invertible by definition.

Theorem 2.4 (matrix eigenvalues).
If $\lambda \ne 0$ is an eigenvalue of a nonsingular matrix A, then $1/\lambda$ is an eigenvalue of the inverse matrix A^{-1}.

4. Stiffness

Definition 2.5 (matrix condition number).
Given any $A \in \mathbb{R}^{n\times n}$. The p-condition number of A is defined as

$$\kappa_p(A) = \|A\|_p\|A^{-1}\|_p, \quad p \ge 1.$$

Theorem 2.5 (matrix condition numbers).
For any $p \geq 1$, we have

$$\kappa_p(A) \geq 1.$$

Apparently, $\kappa_p(I) = 1,\ p \geq 1$.

Condition number is a rough reference index. A matrix does not suddenly become ill-conditioned, falling off the edge of the world. Ill-conditioning is an entire spectrum, going from good to bad to worse.

The condition number tells us how much solving a system of linear equations, or a system of differential equations, will magnify any noise in your data. Think of it as a measure of amplification, a gain.

So, let us assume that our condition number is 10^8, then when we solve a linear system, that the solution procedure may magnify small noise by roughly a factor of 10^8. This might not be a problem if our data is in double precision, and our noise level is down in the least significant bits of the data (that is, at the level of $\mathcal{O}(10^{-16})$. But if our data comes from real world experiments, then your noise might be on the order of, say, as big as 1% of the solution size.

If we then magnify that noise by a factor of 10^8, then our noise will dominate the solution, even though the matrix problem is still solvable with no warnings issued!

So we must understand that the condition number of a matrix is a relative thing in terms of how bad it can be. A solve will not issue warning messages until the condition number goes above some specified threshold. The result will not turn to NaNs until things get really bad. But even for some larger condition numbers, we can still get garbage out if the noise is sufficiently large.

Therefore a good numerical method is important.

Of course, a condition number can never fall below 1. For example, an *orthogonal matrix* would have a condition number of 1. It will not amplify any noise in your data. So a condition number that is small is ideal. The bigger it is, our troubles may get worse as that condition number increases. A condition number of 10^{16} is truly big enough that may expect garbage for results almost always (at least when working in double precision).

The following example is due to readers' discussions in Matlab URL.

Example. Let

$$A(\delta) = \begin{bmatrix} 1 & 1-\delta \\ 1 & 1+\delta \end{bmatrix},$$

where $0 < \delta \ll 1$ is a small parameter.

First, we set $\delta = 10^{-8}$. For this,

$$A(10^{-8}) = \begin{bmatrix} 1 & 0.99999999 \\ 1 & 1.00000001 \end{bmatrix}.$$

Utilize Matlab, we have

$$\kappa_p(A(10^{-8})) = 199999999.137258 \approx 10^9.$$

Let us consider a system of two equations

$$A(10^{-8})x = b.$$

We may assume that

$$x = \begin{bmatrix} 1 \\ 3 \end{bmatrix} \tag{$*$}$$

and for this,

$$b = \begin{bmatrix} 3.99999997 \\ 4.00000003 \end{bmatrix}$$

which is nice!

Now, our experiment is about the question that can we recover the solution x from the following system?

$$\begin{bmatrix} 1 & 0.99999999 \\ 1 & 1.00000001 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3.99999997 \\ 4.00000003 \end{bmatrix}.$$

Using the famous backslash operation $A \setminus b$, we find that

$$x = \begin{bmatrix} 0.999999994448885 \\ 3.00000000555112 \end{bmatrix}$$

which is different from the original solution x in equation (*). This indicates that the result is corrupted by noise. The solution procedure used floating point operations in double precision in Matlab. In the end, the moderately large condition number amplified any errors in the least significant bits by roughly 10^8.

To see more the role of the condition number in above experiment, let us enlarge slightly the noise magnitude to about $\mathcal{O}(10^{-7})$. We just want to see what can be the consequence. To this end, we consider

$$\tilde{b} = b + \text{randn(size(}b\text{)*1.e-7)}$$

and then solve

$$A(10^{-8})\tilde{x} = \tilde{b}$$

by using the same backslash operation in Matlab. It turns out that

$$\tilde{x} = \begin{bmatrix} 4.94229551969727 \\ -0.942295486579804 \end{bmatrix}.$$

This is definitely out-of-question!

In our computations, we would call A as ill-conditioned if

$$\kappa(A) \geq 10^5.$$

Unfortunately, discretizations of black-scholes model equations always lead to ill-conditioned matrices. That is why the study of superior numerical methods becomes crucial.

$\Longleftrightarrow$

For materials related to matrix properties, you may consider several paragraphs in Wikipedia such as

https://en.wikipedia.org/wiki/Condition_number

For more information about solutions of linear systems such as (2.22), you may read *Matrix Computations* by Gene Golub and Charles Van Loan, 4th Ed., Johns Hopkins University Press, Baltimore, USA, 2013.

$\Longleftrightarrow$

Now, let us go back to our system of linear differential equation (2.22):

$$u'(t) = Au(t) + g, \tag{2.22}$$

for which $A \in \mathbb{R}^{n\times n},\ u, g, \phi \in \mathbb{R}^n$, together with an initial condition

$$u(0) = \phi.$$

Note that n can be a very big integer, say, $n \approx 10^8$, in typical financial option computations.

Theoretically, we can solve the system by use different methods. However, keep in mind that our matrix A is big in size. Therefore most classical methods are either difficult, or inefficient, to use practically.

In fact, the exact solution of (2.22) can be written as

$$u(t) = e^{tA}\phi + \int_0^t e^{(t-\xi)A} g(\xi)d\xi, \quad t \geq 0. \tag{**}$$

The matrix exponential $e^{\tau A}$ is defined through the Taylor expansion

$$e^{\tau A} = I + \tau A + \frac{1}{2!}(\tau A)^2 + \frac{1}{3!}(\tau A)^3 + \cdots + \frac{1}{k!}(\tau A)^k + \cdots \in \mathbb{R}^{n\times n}.$$

Apparently, on a mesh $\Omega = \{t_0, t_1, t_2, \ldots, t_M, t_{M+1}\}$, solution formula (**) can be written as

$$u(t_{i+1}) = e^{t_{i+1}A}u(t_i) + \int_{t_i}^{t_{i+1}} e^{(t_{i+1}-\xi)A} g(\xi)d\xi, \quad i = 0, 1, \ldots \tag{***}$$

Discretizations of (2.22) or (***) lead to different numerical methods for solving the initial value problem. For instance, a forward finite difference approximation of the derivative offers the following:

$$\begin{aligned} u(t_{i+1}) &= (I + \tau_{i+1}A)u(t_i) + g(t_i) + \mathcal{O}(h^2), \\ u(t_0) &= u_0, \end{aligned}$$

in which

$$A \in \mathbb{R}^{n\times n},\ u, g, u_0 \in \mathbb{R}^n, \text{ and } \tau_{i+1} = t_{i+1} - t_i,\ t_\ell \in \Omega.$$

Remove the truncation error term, we obtain from above the forward Euler method,

$$u_{i+1} = (I + \tau_{i+1}A)u_i + g_i, \quad i = 0, 1, 2, \ldots \tag{2.25}$$

$$u_0 = u_0, \tag{2.26}$$

where $g_i = g(t_i)$ and u_i is an approximation of $u(t_i),\ i = 0, 1, 2, \ldots$

Example. Consider

$$y' = \Lambda y, \quad t > 0, \tag{2.27}$$

$$y(0) = \phi, \tag{2.28}$$

where

$$\Lambda = \begin{bmatrix} -100 & 1 \\ 0 & -1/10 \end{bmatrix}, \ y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \ \phi = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}.$$

We notice that (2.27) can be written in a classical form

$$\begin{aligned} y_1' &= -100y_1 + y_2, \quad t > 0, \\ y_2' &= -(1/10)y_2, \quad t > 0. \end{aligned}$$

Solve the second equation first, we acquire that

$$y_2 = \exp\left\{-\frac{1}{10}t\right\}\phi_2, \quad t \geq 0.$$

Substituting into the first equation, we have

$$y_1' = -100y_1 + \exp\left\{-\frac{1}{10}t\right\}\phi_2, \quad t > 0.$$

Use (**), we have

$$\begin{aligned} y_1(t) &= e^{-100t}\phi_1 + \int_0^t e^{-100(t-\xi)}\exp\left\{-\frac{1}{10}\xi\right\}\phi_2 d\xi \\ &= e^{-100t}\phi_1 + e^{-100t}\int_0^t \exp\left\{\left(100 - \frac{1}{10}\right)\xi\right\} d\xi\phi_2 \\ &= e^{-100t}\phi_1 + e^{-100t}\int_0^t \exp\left\{\frac{999}{10}\xi\right\} d\xi\phi_2 \\ &= e^{-100t}\phi_1 + \frac{10}{999}e^{-100t}\left(\exp\left\{\frac{999}{10}t\right\} - 1\right)\phi_2, \quad t \geq 0. \end{aligned}$$

The above shows that there exist $x_1, \ x_2 \in \mathbb{R}^2$ depending only on ϕ_1, ϕ_2 (can you prove this?) such that

$$y(t) = e^{-100t}x_1 + e^{-t/10}x_2, \quad t \geq 0.$$

We notice that the first term decays much faster than the second term in the above solution as $t \to +\infty$.

♡♡

Of course, we may use an operator form solution for solving (2.27), (2.28). To this end, integrating both sides of (2.27) yields

$$y = e^{t\Lambda}\phi, \quad t \geq t.$$

Note that a similar transform leads Λ to

$$\begin{aligned} \Lambda &= \begin{bmatrix} -100 & 1 \\ 0 & -1/10 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 999/10 \end{bmatrix}\begin{bmatrix} -100 & 0 \\ 0 & -1/10 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 0 & 999/10 \end{bmatrix}^{-1} \\ &= VDV^{-1}. \end{aligned}$$

Thus, matrices Λ and D are similar. Therefore

$$\lambda_1 = -100,\ \ \lambda_2 = -\frac{1}{10}$$

are eigenvalues of Λ. It follows that

$$\begin{aligned}
e^{t\Lambda} &= I + t\Lambda + \frac{1}{2!}(t\Lambda)^2 + \frac{1}{3!}(t\Lambda)^3 + \cdots + \frac{1}{k!}(t\Lambda)^k + \cdots \\
&= I + t\Lambda + \frac{t^2}{2!}\Lambda^2 + \frac{t^3}{3!}\Lambda^3 + \cdots + \frac{t^k}{k!}\Lambda^k + \cdots \\
&= I + tVDV^{-1} + \frac{t^2}{2!}(VDV^{-1})^2 + \frac{t^3}{3!}(VDV^{-1})^3 + \cdots + \frac{t^k}{k!}(VDV^{-1})^k + \cdots \\
&= I + tVDV^{-1} + \frac{t^2}{2!}VDV^{-1}VDV^{-1} + \frac{t^3}{3!}VDV^{-1}VDV^{-1}VDV^{-1} + \cdots \\
&\quad + \frac{t^k}{k!}VDV^{-1}VDV^{-1}\cdots VDV^{-1} + \cdots \\
&= I + tVDV^{-1} + \frac{t^2}{2!}VD^2V^{-1} + \frac{t^3}{3!}VD^3V^{-1} + \cdots + \frac{t^k}{k!}VD^kV^{-1} + \cdots \\
&= V\left(I + tD + \frac{t^2}{2!}D^2 + \frac{t^3}{3!}D^3 + \cdots + \frac{t^k}{k!}D^k + \cdots\right)V^{-1} \\
&= V\begin{bmatrix} e^{-100t} & 0 \\ 0 & e^{-t/10} \end{bmatrix} V^{-1} \in \mathbb{R}^{2\times 2}.
\end{aligned}$$

Recall the solution of (2.27), (2.28). We have

$$y = V\begin{bmatrix} e^{-100t} & 0 \\ 0 & e^{-t/10} \end{bmatrix} V^{-1}\phi, \quad t \geq 0.$$

The above leads to the same conclusion that there exist $x_1,\ x_2 \in \mathbb{R}^2$ depending only on ϕ_1, ϕ_2 such that

$$y(t) = e^{-100t}x_1 + e^{-t/10}x_2, \quad t \geq 0.$$

Can you mathematically prove it?

References

[1] K. in 't Hout, *Numerical Partial Differential Equations in Finance Explained,* Springer, Antwerp, Belgium, 2017.

[2] R. Pratap, *Getting Started with Matlab: A Quick Introduction for Scientists and Engineers,* 7th Ed., Oxford Univ. Press, 2016.

[3] K. Atkinson and W. Han, *Elementary Numerical Analysis,* 3rd Ed., John Wiley & Sons, Somerset, NJ, 2004.

[4] A. Iserles, *A First Course in the Numerical Analysis of Differential Equations,* Cambridge University Press, Cambridge and New York, 2004.

[5] B. Jain and A. Sheng, An exploration of the approximation of derivative functions via finite differences, *Rose-Hulman Undergrd. Math J.,* 8:172-188, 2007.

[6] Khan Academy, *Intro to Matrices,* https://www.khanacademy.org/math/precalculus/x9e81a4f98389efdf:matrices/x9e81a4f98389efdf:mat-intro/a/intro-to-matrices

[7] R. J. Spiteri, *Solution of Nonlinear Equations,* https://www.cs.usask.ca/~spiteri/M211/notes/chapter4.pdf

[8] G. Golub and C. Van Loan, *Matrix Computations,* 4th Ed., Johns Hopkins University Press, Baltimore, USA, 2013.