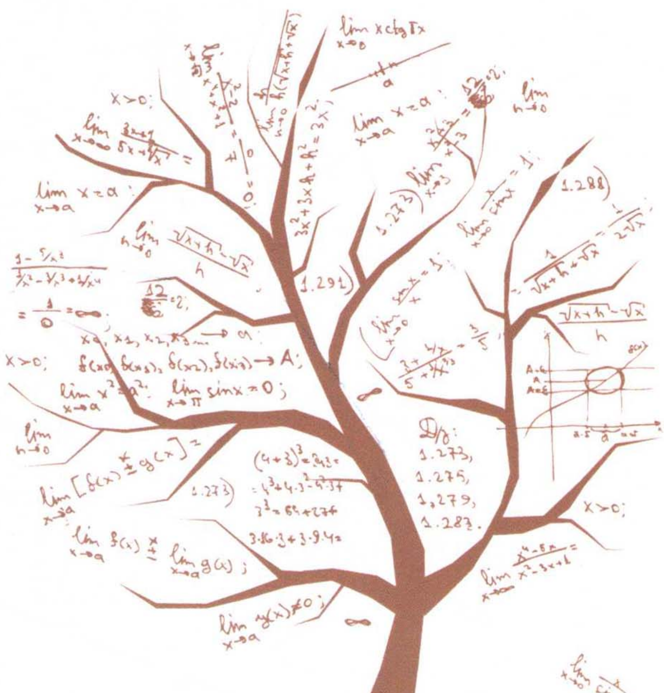
TURING

图灵程序设计丛书

程序员的数学3

线性代数

[日] 平冈和幸　堀玄 / 著　卢晓南 / 译



从入门到应用，透彻理解线性代数的本质

最易上手的图解教程

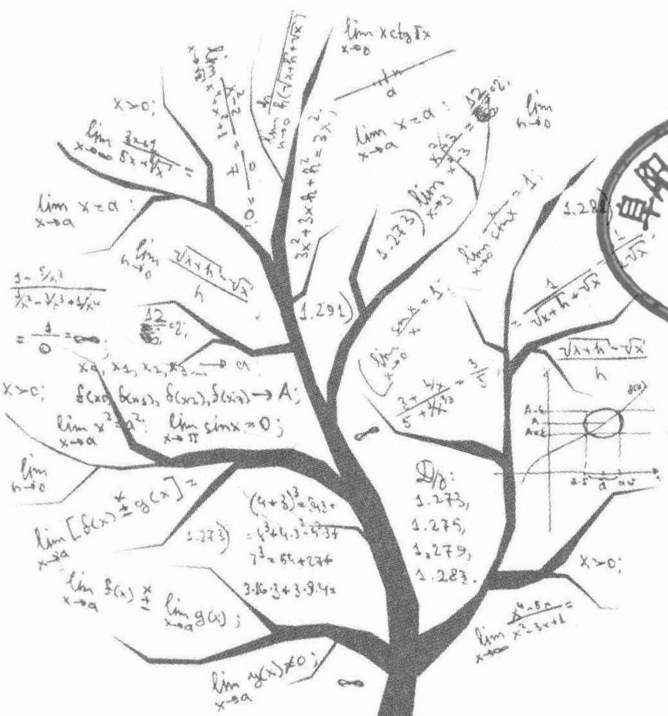
中国工信出版集团　人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序
设计丛书

程序员的数学③
线性代数

[日] 平冈和幸 堀玄 / 著 卢晓南 / 译



阜阳师范学院图书馆
藏书

人民邮电出版社
北京

图书在版编目（CIP）数据

程序员的数学. 3，线性代数 /（日）平冈和幸，（日）堀玄著；卢晓南译. — 北京：人民邮电出版社，2016.3

（图灵程序设计丛书）

ISBN 978-7-115-41774-9

Ⅰ. ①程… Ⅱ. ①平… ②堀… ③卢… Ⅲ. ①电子计算机—数学基础②线性代数 Ⅳ. ①TP301.6②O151.2

中国版本图书馆CIP数据核字（2016）第023197号

内 容 提 要

本书沿袭“程序员的数学”系列平易近人的风格，用通俗的语言和具象的图表深入讲解了编程中所需的线性代数知识，内容包括向量、矩阵、行列式、秩、逆矩阵、线性方程、LU分解、特征值、对角化、Jordan标准型、特征值算法等。

本书适合所有与计算机相关的专业和非专业人士，以及学习线性代数的学生阅读。

◆ 著　[日] 平冈和幸　堀玄
　译　卢晓南
　责任编辑　乐　馨
　执行编辑　杜晓静
　责任印刷　杨林杰

◆ 人民邮电出版社出版发行　北京市丰台区成寿寺路11号
　邮编　100164　电子邮件　315@ptpress.com.cn
　网址　http://www.ptpress.com.cn
　北京天宇星印刷厂印刷

◆ 开本：800×1000　1/16
　印张：24
　字数：510千字　2016年3月第1版
　印数：1-4 000册　2016年3月北京第1次印刷

著作权合同登记号　图字：01-2013-8253号

定价：79.00元

读者服务热线：(010)51095186转600　印装质量热线：(010)81055316

反盗版热线：(010)81055315

广告经营许可证：京东工商广字第8052号

译者序

相信手捧本书的读者，对“线性代数”这个词一定不陌生。不管当你回想起这四个字时心里是什么滋味，我相信通过本书，你一定会对这门数学（同时可能也是你的工具）产生新的认识。

其实，译者在最初学习线性代数的时候，也经历过一段曲折的过程。在没有问题导向的学习方式下，学习的过程既枯燥又效率低下。但是后来在数学专业课的学习过程中，无时无刻不在使用微积分和线性代数作为基本工具，经过这种“居高临下”的训练，才悟出线性代数的本质是什么。当时很多数学系的同学都有这样的感悟：“学过数值代数，才知道线性代数是多简单，多纯粹。”（本书就会带领大家快速进入数值代数领域！）工科的读者可能没有这样的经历，也就渐渐忘记了大一时学过的数学。然而，在信息科学、计算机科学的研究开发领域，向量、矩阵已经是“基本单位”了。如果你正因为“向量、矩阵”而在研究中放不开手脚，或者因为“数值算法的实现”而在工作中畏首畏尾，那么本书正是为你准备的。关于本书的内容和风格，这里就不多说了，请看后面的前言吧！

说起来，译者也是偶然与本书结缘的。在“脱离”数学系后，写了不少年的程序，直到译者赴日本继续数学学习和研究，无意间见到本书原版，这时才发现，原来可以通过这样的方式来讲解、学习线性代数。本书不仅直接从本质意义出发对所有核心概念都给予了直观的解释，还能带领读者“快速直达”数值代数领域！

同样也是机缘巧合，在发现图灵公司打算引进本书时，译者几乎是一瞬间就决定了要尝试翻译本书。在投身翻译工作之后，才意识到仅有数学、计算机知识以及日语阅读能力是远远不够的。因此，在翻译中可能会有表达不到位的情况出现，欢迎大家批评指正。这里也要特别感谢图灵的编辑在翻译过程中给予的帮助和支持。

正文中原作者多次提到了参考文献和扩展阅读书目，但是据译者所知，原书的参考文献

目前都没有中译本。为此，译者在这里斗胆推荐一些中文的参考文献。

关于第 2 章中提到的“张量”“外积”等概念，建议有兴趣的读者参考柯斯特利金的《代数学引论（第二卷）：线性代数》的第 6 章“张量”。另外，对于在数学的抽象性和严密性上有较高要求的“数学派”的读者，特别推荐龚昇先生的《线性代数五讲》。这本很薄的小册子，从现代数学的观点（模理论）出发，对线性空间、线性变换进行了全新的诠释。由于龚先生书写得非常精炼，如果阅读起来感觉吃力的话，不妨看看戈德门特的《代数学教程》。通过我们这本书，读者可以对线性变换（包括坐标变换）有直观感受，而通过更高阶的阅读，则能从一般线性群等更抽象的角度去理解“变换”，这对从事信息科学、数据科学等研究工作的读者来说是颇有裨益的。从 [1] 和 [2] 中，读者也可以略微感受到老牌数学强国俄罗斯和法国的数学风格。如果不打算深究“纯数学”的话，读者可以在本书的基础上，根据自己的需要，参考《数值分析》《矩阵论》等教材。

最后，由衷希望大家能从本书中有所收获，喜欢本书，并推荐给亲友。谢谢！

卢晓南

2015 年 11 月于名古屋

参考文献与扩展阅读书目

[1] А. И. 柯斯特利金 [著], 牛凤山 [译]. 代数学引论（第二卷）线性代数（第 3 版）[M]. 北京: 高等教育出版社, 2008.

[2] R. 戈德门特 [著], 王耀东 [译]. 代数学教程 [M]. 北京: 高等教育出版社, 2013.

[3] 龚昇. 线性代数五讲 [M]. 北京: 科学出版社, 2005.

[4] 丘维声. 高等代数（上）（第 2 版）[M]. 北京: 高等教育出版社, 2002.

[5] 丘维声. 高等代数（下）（第 2 版）[M]. 北京: 高等教育出版社, 2003.

[6] 徐树方, 高立, 张平文. 数值线性代数 [M]. 北京: 北京大学出版社, 2000.

试读结束：需要全本请在线购买： www.ertongbook.com

前言

看到本书的标题《程序员的数学 3：线性代数》时，不同的读者应该会有不同的印象吧。我们根据读者可能会产生的第一印象，为读者设定了“快捷方式”。

- “又是程序员的数学系列啊”→(a)
- “肯定有好多公式，推理也会很烦，念起来应该很吃力吧”→(b)
- “想必会解释得细致入微，但讲解的深度应该很有限吧”→(c)
- “这作者是干嘛的”→(d)
- “我也不编程啊”→(e)

(a) 致想到“又是程序员的数学系列啊”的读者

本书面向的主要读者群体包括与计算机相关的所有专业与非专业人士。作为一本线性代数的参考书，本书的一大特色是，针对以上这些读者，在讲解时使用了易于他们理解的表述方式，并运用了大量的示例和比喻。我们的目标是向非数学专业的读者讲述线性代数的本质。正因为如此，这不是一本单单讲解“如何进行线性代数相关的编程”的书。读者只要阅读一下前言的(c)部分，就可以对本书的风格有个大概的了解了。我们把本书特别推荐给以下读者。

- 想要从事信号处理、数据分析等方面的工作或研究，在阅读相关专业书籍的过程中遇到了线性代数，面对这些问题怎么也搞不明白，因此希望学习（或者补充）一下相关知识。但是能找到的参考书中，不是充斥着数学证明的教科书，就是看过之后似懂非懂的入门书

- 正在学校学习线性代数，而且不仅仅满足于通过考试，而是希望切切实实地掌握相关知识，以便在以后的工作中熟练使用

因为本书面向的读者主要是非数学专业的，所以我们不会为了数学而讲数学，而是更加强调“这些知识在哪里会有用”。虽然理工科中有众多不同的专业，每个专业所研究的对象也是各种各样，但是其中涉及的数学问题却总有着这样那样的共通之处。在本书中，我们首先会提炼出这类问题，接着在挑战这些问题的过程中导入线性代数的概念。这就是本书的风格。这样做不仅是为了讲解数学理论，更是为了使读者学会线性代数的“用法”。

(b) 致想到“肯定有好多公式，推理也会很烦，念起来应该很吃力吧”的读者

为了让读者尽可能透彻地理解线性代数的本质含义，本书中在说明时穿插了很多直观的示意图。试想，如果学完了线性代数，却只懂得行列式的计算，而对行列式的意义一无所知，那这种学习有什么用呢? 无论是笔算还是用计算机算，如果只是求出了“迷一般”的行列式的值，那没有任何意义。为了避免这种徒劳无功的学习，本书会着重对原理、推导过程进行细致的解释。

但是，就算再完美、再严密的理论体系，(对非数学专业的人来说) 也总会在少数地方存在一些麻烦的东西。在学习比入门书难度稍大一点的参考书时，想必很多读者都在一些无关痛痒的难点处栽了跟头。在本书中，我们会重点关注真正重要的地方，为读者讲解思路和方法，使读者不只是学会计算步骤，还能达到更高的层次。关于数学公式，在必要的时候当然会使用，但是为了避免读者产生恐惧心理，我们尽量避免了那些往往会吓到外行的一本正经的表述①。

另外，针对读者不同层次的需求，本书采用了可以进行跳跃式阅读的结构 (更加细致的章节结构请参考目录)。

第一层次

在阅读那些以线性代数为工具的资料时，比如信号处理、数据分析等领域的参考书，希望能够明白其中的数学公式等的意义

→ 阅读第 1 章 (跳过标有 $\triangledown$ 和 $\triangledown\triangledown$ 的章节②)

---

① 比如，不写成 $\sum_{i=1}^{10} a_i$ 的形式，而是采用 $a_1 + \cdots + a_{10}$ 的记法。对于涉及变量、下标等的地方，我们会采用更加具体直观的写法。

② 标有 $\triangledown$ 的章节的主要内容是如何进行笔算 (以及相关知识)，标有 $\triangledown\triangledown$ 的章节的主要内容是如何使用计算机进行计算以及相关的算法分析。那些只是草草学过一遍线性代数的读者，在阅读过第 1 章之后，也会有不少新的发现和启发。

第二层次

在阅读以线性代数为工具的参考书时，希望理解书中的内容
→ 阅读全书（跳过标有 ▽ 和 ▽▽ 的章节）

第三层次

希望能够自己进行计算
→ 阅读全书（跳过标有 ▽▽ 的章节）

第四层次

希望踏入大规模矩阵计算的世界
→ 阅读全书，包括标有 ▽▽ 的章节

那些不期望自己成为专家的读者，把目标定在第二层次如何？如果有时间学习“逆矩阵的笔算法”的话，记住“如果映射带来了压缩扁平化变换，则不存在逆矩阵”这一本质特性是非常有意义的。哪怕是为了计算而学，比起死记硬背笔算法的步骤，“能够区分 $\boldsymbol{x}\boldsymbol{x}^T$ 是矩阵，$\boldsymbol{x}^T\boldsymbol{x}$ 是数”“能够用分块矩阵来表示 $A\boldsymbol{x}+\boldsymbol{b}$”这些技能①在后续的学习和工作中会更有用。好了，以上是笔者对读者的一些建议。

(c) 致想到“想必会解释得细致入微，但讲解的深度应该很有限吧”的读者

生硬的数学书就好比是缺少注释的源代码。虽然有些程序执行效率非常高，也很优雅，但是从可读性的角度来讲，要想理解代码的含义，就需要付出一定的努力、具有一定的素养，以及对程序有感觉（夸张地讲，可能还需要一些逆向工程的知识）。另一方面，入门书一不小心就会出现类似于“只有注释而没有代码”“虽然有代码段，但是没办法将其作为完整的程序来执行”的情况。本书属于“既给出能够执行的完整代码②，又附带充足的注释”那种类型。我们的注释，不会是如下这种风格。

```
# 给 p 增加 1
p = p + 1
```

这种注释有没有都一样。我们的风格是这样的。

```
# 前奏已经足够了，转到下一页
```

① 这些将在 1.2.13 节和 1.2.9 节中进行说明。
② 对于无论如何都无法避免繁杂操作的部分，在少数地方我们会利用现成的工具包、类库进行封装处理。

```
p = p + 1
```

注释只有说明了指令的意图，才是有意义的。我们一方面通过注释帮助代码把“内心的真实想法”传达给读者，同时也写出了本身就“不乏风味”的代码。这也是本书的一大亮点。

另外，“不乏风味”还有一层含义，那就是不单单让读者学会解题，还要学到一种看待问题的新方式。比如，“学了秩的概念之后，一下子就能判断出能否由结果推出原因了”“学了特征值、特征向量的概念之后，很快就能了解系统是否有失控危险了”，像这样，如果读者能够感到自己的思路被打开，最后体会到了“一览众山小”的快乐，那么我们的目的就达到了。为此，如果仅仅是简单介绍秩的概念，或者简单讲解计算方法，那是远远不够的。只有充分理解了其本质含义，才能从问题的根源出发，从而轻而易举地理解“不满秩则矩阵的逆不存在”等问题。出于这种考虑，本书遵循以下原则。

- 要讲就要讲到本质，否则就没有意义了
- 用浅显的语言逐步解释，让读者打心底里认为“推出这样的结果是理所当然的”

最后要达到的层次绝对不算低。本书涉及了很多线性代数教科书中通常不会提及的数值分析的内容。这一点大家看看目录就会发现。

另外，本书作为“基础篇”，基本上不考虑带有度量的问题，而是以无度量的问题为中心。以后如果有机会的话，我们也许会以某种方式发布“应用篇”，届时再讨论带有度量的实际问题。

(d) 致想到“这作者是干嘛的”的读者

笔者是从事应用数学、工程数学研究的科研工作者。在诸如模式识别、神经网络、非线性动力系统、统计数据分析等领域，本书涉及的内容作为“常识”，每天都在发挥着关键作用。就算是进入了非线性理论的世界，线性代数作为基本的工具也是必不可少的。无论是在理论上还是在应用上，笔者都将尽可能地展现出数学真正有用的一面。另外，本书中也用心地选取了合适的题材和恰当的切入点，使得在实际问题中线性代数的“使用”不显得突兀。

(e) 致想到“我也不编程啊”的读者

读过前言的 (a) 部分就会了解到，本书并不以“如何进行线性代数的相关编程”为主要内容。现在不管在任何领域，多多少少都要与计算机打交道，而本书就是着眼于在众多领域中都发挥作用的线性代数，对其内涵进行剖析。与计算机打交道的核心就是“(编) 程序”，而本书就是为编程序的人，即程序员而写的，这就是本书书名的由来。

在第 3 章“计算机上的计算”中，我们也提供了可在实际应用中用于矩阵计算的例程。

另外，为了让读者切身体会到矩阵是如何表示映射的，我们也准备了简单的程序供读者在自己的计算机上使用。读者可以从图灵社区的网站上下载源代码[①]，请一定要体验一下！

还有一点要说明，本书中给出的源代码全部采用 Ruby 语言编写。之所以选择 Ruby，首先，我们希望选择一种高级语言，从而在代码中避开那些与算法本身关系不大的繁杂部分。其次，我们希望程序语言的语法比较接近自然语言，从而减少不必要的理解错误。最后一个非常重要的原因是，相比伪代码，我们更希望呈现给大家能实际运行的源代码。这也正是我们采用 Ruby 的初衷。

– 致认为本书是讲解如何用 Ruby 进行线性代数编程的读者

不好意思，真不是这样的。书中的代码不会有浓浓的"Ruby 风"，为了使人人都可以轻松理解，笔者在写代码时参照了伪代码的写法。

– 致一听到 Ruby 就想把书合起来的读者

如前所述，只要读者接触过其他主流的程序设计语言，就可以在没有任何 Ruby 基础的情况下顺利地阅读本书中的源代码。最后，有一点请读者千万不要误解，真正的 Ruby 语言，可不是只能做做这些简单死板的工作而已，敬请广大读者周知。

① 打开 http://www.ituring.com.cn/book/1239，点击"随书下载"。

致谢

感谢埼玉大学的重原孝臣教授在本书编写过程中提出了诸多想法、建议，和笔者进行了有益的讨论，指正了其中的错误，并给予笔者很多鼓励。同时，感谢欧姆社开发部的诸位，一直给予我们创作的动力，并帮助我们完善稿件。正因为有了他们的努力，本书才得以与广大读者见面。非常感谢。

本书在处理插图以及验算时，用到了 Ruby（程序设计语言）、Gnuplot （绘图工具）、Maxima（公式推导和计算工具）、xlispstat（统计计算处理环境）等软件。这里向开发、发布这些优秀作品的诸位表示感谢。

试读结束：需要全本请在线购买： www.ertongbook.com

综述
—— 通过动画学习线性代数

—— 比起计算方法，更重要的是掌握本质含义。

在本书的综述部分，我们将对主要内容进行简要总结。建议读者每阅读完一部分内容，就回过头来看看这里，整理一下思路。

矩阵不仅仅是数字排列而成的表而已。比如 $m \times n$ 矩阵 A，它表示了从 n 维空间到 m 维空间的“映射”。具体来讲，就是把 n 维空间中的点 $\boldsymbol{x}$（n 维列向量）变换到 m 维空间中的点（m 维列向量）$A\boldsymbol{x}$ 的映射。为了便于读者观察这个映射的行为，了解秩、行列式、特征值、对角化等概念，我们在这里汇总了动画演示程序的执行结果。关于动画演示程序的写法、读法以及用法，请参考附录 F。

小试牛刀：观察对角矩阵

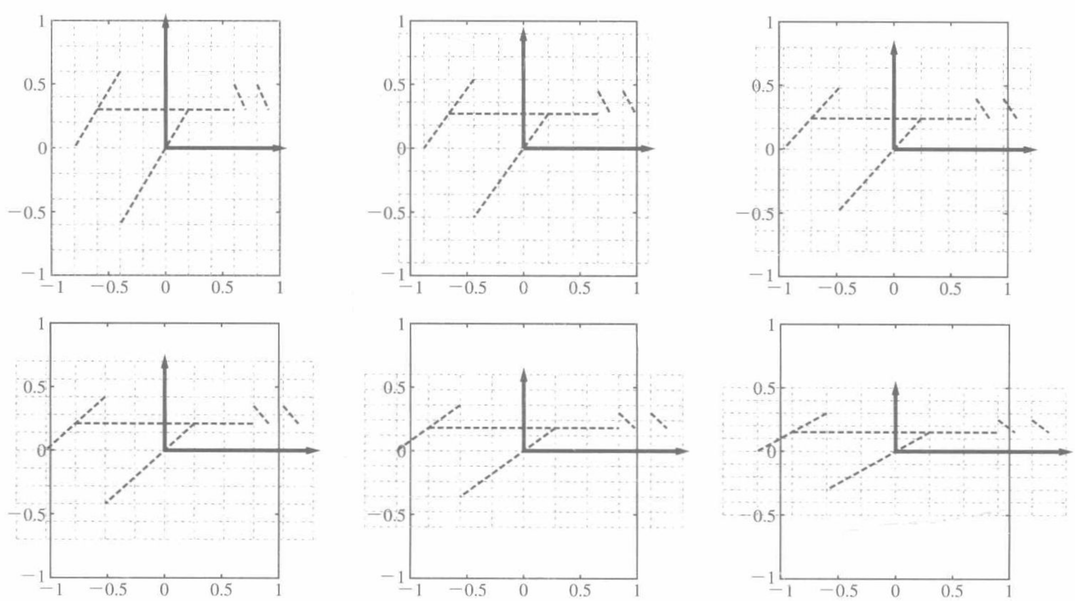
■ 首先是典型的对角矩阵

在下列矩阵 A 的作用下，空间会产生怎样的变形呢？让我们通过动画来看一看。

$$A = \begin{pmatrix} 1.5 & 0 \\ 0 & 0.5 \end{pmatrix}$$

执行以下命令后，就可以通过动画观察到矩阵 A 给空间带来的一系列变化。

```
ruby mat_anim.rb -s=0 | gnuplot
```



观察要点：

- 水平和垂直方向上的伸缩
- 水平方向上扩大（1.5 倍），垂直方向上缩小（0.5 倍）
- 各小方格的面积变成了原来的 $1.5 \times 0.5 = 0.75$ 倍。这里的面积扩大率 0.75 就是 $\det A$。因此，对角矩阵的**行列式** = 对角元素的乘积

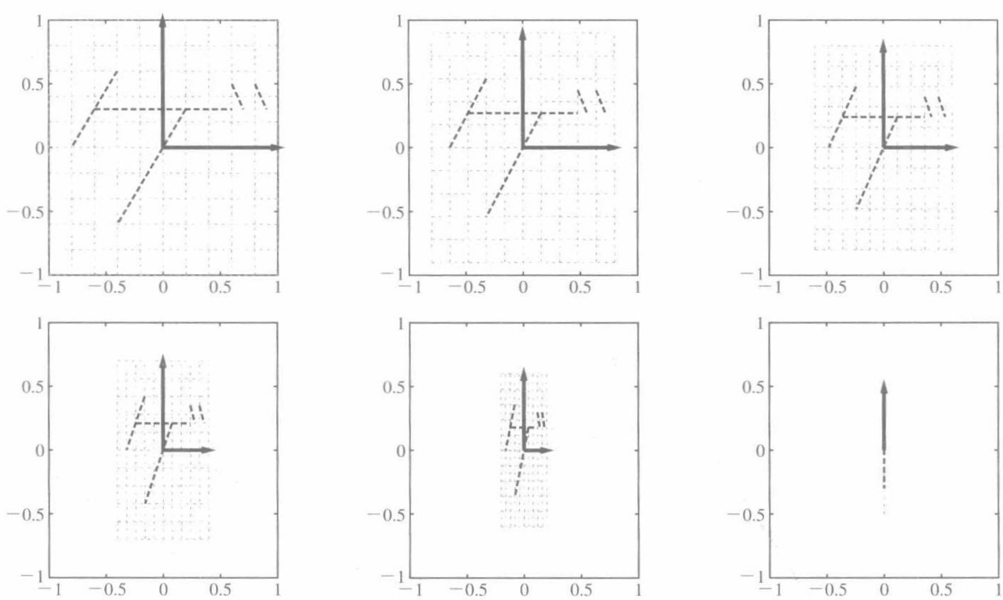
■ 如果对角元素中有 0 的话 ……

像下面的矩阵 A 一样，如果对角元素中有 0，那么会带来什么样的变形呢?

$$A = \begin{pmatrix} 0 & 0 \\ 0 & 0.5 \end{pmatrix}$$

执行以下命令后，就可以通过动画观察到矩阵 A 给空间带来的一系列变化。

```
ruby mat_anim.rb -s=1 | gnuplot
```



观察要点：

- 水平方向上变成原来的 0 倍 → 压缩扁平化
- 面积扩大率 $\det A = 0$

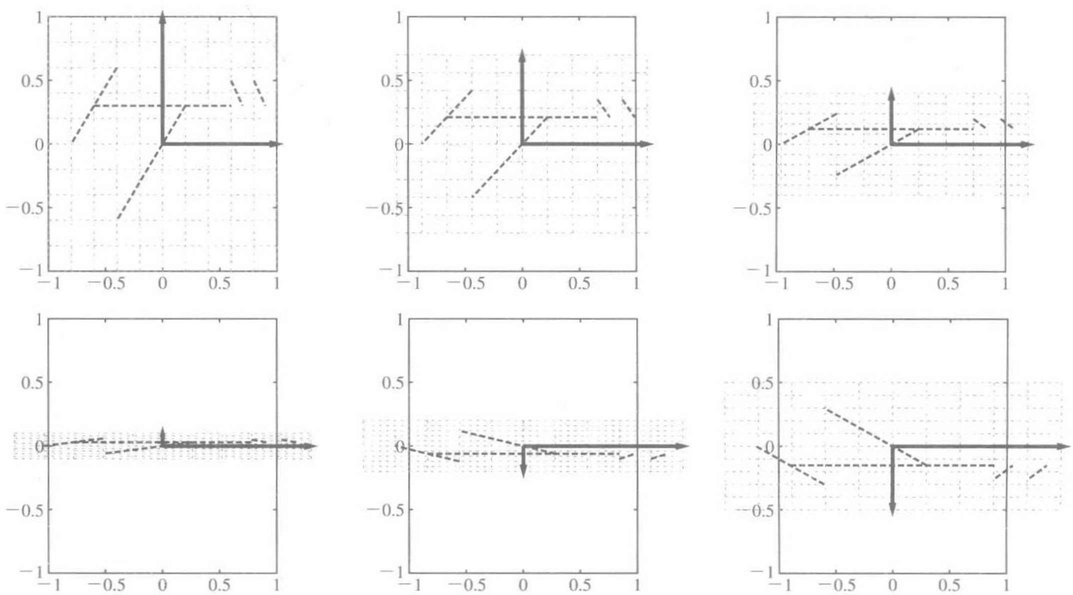
■ 如果对角元素中有负数的话……

下面的矩阵 A 中，对角元素中出现了负数。

$$A = \begin{pmatrix} 1.5 & 0 \\ 0 & -0.5 \end{pmatrix}$$

执行以下命令后，就可以通过动画观察到矩阵 A 给空间带来的一系列变化。

```
ruby mat_anim.rb -s=2 | gnuplot
```



观察要点：

- 垂直方向上变成原来的 -0.5 倍 → 上下颠倒
- 这时有 $\det A < 0$

观察特征值、特征向量与对角化

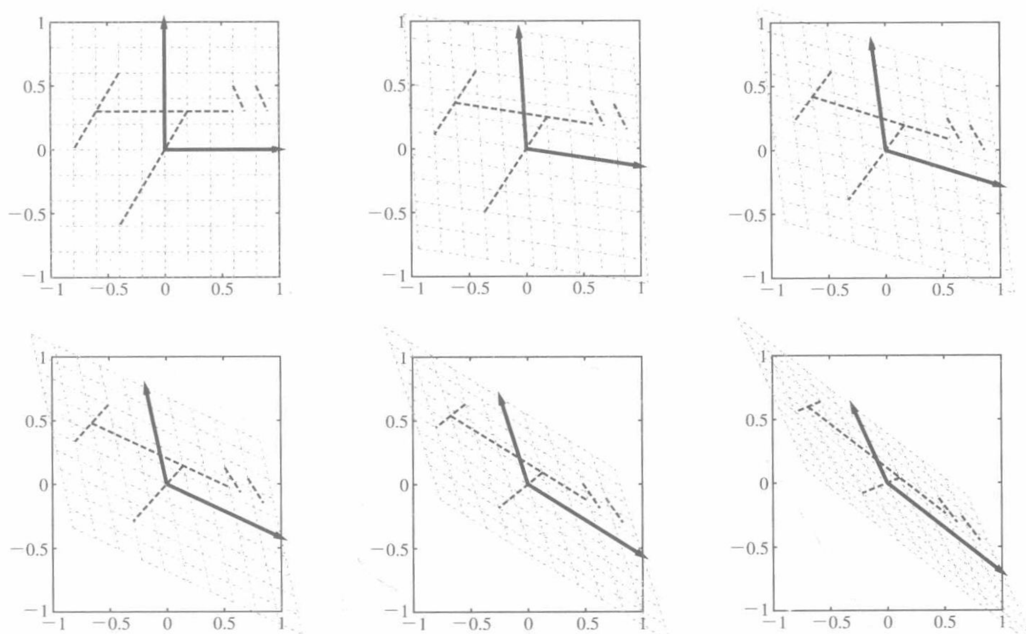
■ 非对角的一般矩阵的情况下，会发生倾斜

我们来观察如下所示的非对角矩阵 A 的情况。

$$A = \begin{pmatrix} 1 & -0.3 \\ -0.7 & 0.6 \end{pmatrix}$$

在 A 的作用下，空间发生了如下倾斜变形。

```
ruby mat_anim.rb -s=3 | gnuplot
```



观察要点：

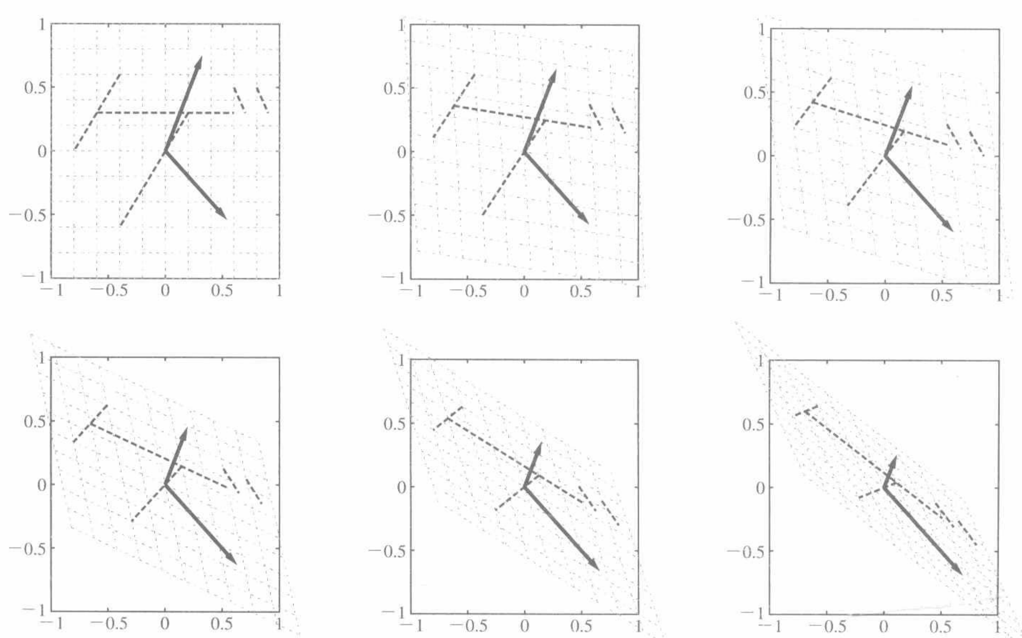
- 倾斜
- 并非“扭曲”，直线依然是直线，平行的依然保持平行
- A 的第 1 列 $\binom{1}{-0.7}$ 为 $\binom{1}{0}$ 的像（目标点），A 的第 2 列 $\binom{-0.3}{0.6}$ 为 $\binom{0}{1}$ 的像（目标点）
- 只要知道了以上两点的像，就可以对整个空间的变化情况进行推断了

■ 如果画出特征向量的话……

随着空间的变化，特征向量会发生什么样的变化呢？下面我们就来看看。矩阵还是上例中的矩阵，所以空间的变化情况和刚才一样。

$$A = \begin{pmatrix} 1 & -0.3 \\ -0.7 & 0.6 \end{pmatrix}$$

```
ruby mat_anim.rb -s=4 | gnuplot
```



观察要点：

- 有向线段只发生了伸缩，而方向没有变化，这便是**特征向量**
- 伸缩率等于**特征值**。拉伸的那条对应的特征值是 1.3，收缩的那条对应的特征值是 0.3

试读结束：需要全本请在线购买： www.ertongbook.com