

# 人工智能背景下线性代数与Python融合教学改革探索

吕海炜, 李 钊, 秦 营

成都大学计算机学院, 四川 成都

收稿日期: 2026年4月5日; 录用日期: 2026年5月21日; 发布日期: 2026年5月29日

## 摘 要

线性代数与Python融合教学改革是鉴于当前线性代数课程概念抽象、计算繁琐、重理论轻应用、与专业实践脱节等问题, 为满足人工智能发展与现代工程教育对创新人才的培养需求而进行的。通过教学内容重构、教学方法创新、典型案例设计与评价体系完善, 借助Numpy、Matplotlib等工具, 构建“理论讲解-代码实现-工程应用”三层次一体化的教学体系, 实现抽象概念可视化、复杂计算简便化和专业内容场景化。

## 关键词

人工智能, 线性代数, Python, 教学改革, 可视化

# Reform and Practice of Linear Algebra and Python Integrated Teaching under the Background of Artificial Intelligence

Haiwei Lyu, Zhao Li, Ying Qin

College of Computer Science, Chengdu University, Chengdu Sichuan

Received: April 5, 2026; accepted: May 21, 2026; published: May 29, 2026

## Abstract

In order to solve the problems existing in the current linear algebra course such as abstract concept, complicated calculation, too much emphasis on theory and lack of connection with professional practice, the reform of integrated teaching of linear algebra and Python is carried out. It seeks to satisfy the need for developing innovative talents against the backdrop of artificial intelligence

progress and modern engineering education. Through the restructuring of teaching content, innovation of teaching methods, design of typical cases and improvement of the evaluation system, with the aid of NumPy, Matplotlib and other tools, a three-level integrated teaching system of “theoretical explanation - code implementation - engineering application” is established, making it possible to visualize abstract ideas, simplify difficult computations, and put professional topics into context.

## Keywords

Artificial Intelligence, Linear Algebra, Python, Teaching Reform, Visualization

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

在人工智能, 大数据, 机器学习等技术飞速发展的背景之下, 线性代数已经变成计算机、电子信息、医学影像、自动化等专业的核心数学根基[1], 矩阵运算、线性方程组、特征值分解、奇异值分解等知识点, 被全面应用到图像处理、数据降维、模式识别、电路仿真等工程领域[2] [3], 传统的线性代数教学侧重于理论推导和手工计算, 内容抽象而乏味, 脱离专业应用, 学生很难领会知识的实际意义, 学习的积极性不高[4] [5]。

Python 语言简洁、开源, 且具有较为完备的科学计算生态, 比如 NumPy, SciPy, Matplotlib 等, 可以高效完成矩阵运算、方程求解、数据可视化等任务, 是将数学理论和工程应用相联系的理想工具[6] [7]。当前研究更多着眼于单一知识点的可视化实现, 缺少针对人工智能场景的系统化教学体系与多元化案例。本文凭借教学经验, 系统构建人工智能背景下线性代数与 Python 融合教学体系, 从内容重塑、方法创新、案例设计、评价改革等方面实施探索, 为同类课程改革提供应用范例。

## 2. 线性代数课程教学存在的问题

### 2.1. 概念抽象, 理解难度大

线性代数包含很多抽象的数学概念, 比如向量, 矩阵, 线性变换, 特征向量等[8], 这些概念很少有直观的几何解释, 也很难和实际问题对应起来。在以往的教学模式下, 教师只能借助板书和教材来讲解定义, 证明定理并进行逻辑推理, 学生很难深刻领悟到概念的意义与应用, 更难把数学与几何直观、实际应用联系在一起[4], 这种“会做题, 不懂原理”的学习状况既影响学习兴趣, 又影响后续专业课程对对数学工具的使用, 远远达不到人工智能时代对数学能力的要求。

### 2.2. 计算繁琐, 学习效率低

线性代数涉及高阶矩阵初等变换, 线性方程组求解以及特征值计算等内容。这些运算一旦复杂起来, 靠手工演算, 步骤繁杂, 学生很容易出错, 而且花费大量时间。课堂与课后时间分配不合理, 就有可能产生畏难情绪和厌学心理[5]。在人工智能和大数据环境下, 复杂计算应交给计算机自动完成, 教学重心应向建模、理解与实际应用转化, 传统以计算为主的教学模式已经明显跟不上时代发展的要求。

### 2.3. 重理论轻应用, 与专业脱节

当下, 过半高校的线性代数课程依旧依照传统数学教材体系, 其内容侧重于理论框架, 案例数量较

少,而且大多是纯粹的数学例题,并未同各个工科专业的实际场景相融合,对于计算机,电子信息,医学成像技术等专业的学生来说,他们更关注线性代数怎样应用于图像处理,CT重建,电路分析,数据挖掘以等方面,学生长时间陷于“学了不知该如何用,学会无法用”的境地,这样的教学模式与专业严重脱离[3],既不符合新工科创建所倡导的学科交叉理念,又难达到人工智能行业对复合型人才的需求。

## 2.4. 教学模式单一,实践能力薄弱

传统线性代数课堂以教师讲授为核心,其教学模式较为单一,大致围绕“定义-定理-例题-练习”展开,缺少实验环节与编程训练[6],学生无法获得把数学问题转为成计算机模型的机会,其应用能力和改进思维也就无法得到提升,在人工智能技术飞速发展的时代,这样只侧重于理论灌输,忽视实际操作体验的教学方法,已无法满足新时期对于人才改进能力和计算思维塑造的需求。

## 3. Python 与线性代数融合教学的优势

### 3.1. 库函数丰富,覆盖全知识点

Python 具备完善的科学计算生态,其能全方位覆盖线性代数各个知识点,NumPy 库包含高效创建矩阵,执行矩阵运算,求逆,计算行列式以及解决线性方程组的功能,是线性代数计算的关键依靠,SciPy 库依托 NumPy 扩展了高级别的数值计算能力,可以应对更为繁杂的工程问题求解任务,Matplotlib 库能够立即生成二维,三维图形,达成向量,变换及运算过程的可视化显示,这些库函数相互协作,形成从基本运算到工程应用的完整体系,从而使得线性代数教学做到“理论-计算-可视化-应用”一体化。

### 3.2. 可视化呈现,化解抽象难题

线性代数教学存在一个很大难点,即抽象性较强,不过 Python 具备很强的可视化能力,这可用来解决此难题,教师经由简短代码,便能把向量变换,矩阵乘法,线性映射等抽象过程变成动态图形或者直观图像,使得抽象概念变得“可见”,“可触”而且“可操作”,矩阵变换可直接表现成图形的旋转,放大,剪切;奇异值分解可体现为图像的压缩与复原。可视化教学把抽象符号转为直观图像,突出减小领悟的困难,唤起学生探究的兴趣,助力学生深入把握线性代数的几何意义及其数学精髓。

### 3.3. 贴合工程实际,强化应用导向

当下,Python 是人工智能,数据分析,机器学习以及计算机视觉领域中的主要开发工具,这同学生日后的专业学习及职业发展紧密相连,把 Python 和线性代数结合,并非仅仅更新教学工具,更是改变了教学导向,有益于学生从“学数学”过渡到“用数学”,在教学过程中运用图像压缩,人脸识别,数据拟合,电路仿真等实际案例,让学生在编程操作的时候领悟数学的应用价值,形成起“数学建模-代码执行-结果分析”的完整工程思维,这样的教学模式既能优化课程的实用性,又能加强学生的就业竞争力,切实达成“以学促用,以用践学”的教学目的。

## 4. 教学改革的具体实践

### 4.1. 实施对象与周期

改革实施对象为成都大学计算机学院 2025 级计算机专业本科生,共计 72 人。教学周期为 1 个学期,共 56 学时,其中理论 48 学时、上机实践 8 学时。

### 4.2. 教学方法创新

本次教学采用课前预习-课中实操-课后拓展的混合式教学方式,以此提升课堂互动效果与学生参

与程度。课前，教师通过线上平台发放预习材料和 Python 基础练习，让学生提前熟悉基本语法和相关知识，为课堂动手操作打好基础。课中以核心知识点讲解、代码演示、小组调试为主，教师演示计算过程与可视化效果，学生亲手调整参数、观察结果变化，在实际操作中加深理解。课后布置图像处理、数据建模等贴近工程实际的拓展任务，锻炼学生的综合应用能力。这套模式把学习的主动权交给学生，能有效提升自主学习意识与探索精神。

### 4.3. 典型教学案例

#### 4.3.1. 案例 1：线性方程组求解与药品配制应用

线性方程组属于线性代数的关键部分，还是工程问题里较为常见的数学模型类型，在食品调配，资源安排，药品配制，电路调和之类的情况下，都可以转变成线性方程组的求解问题。在以往的教学，多元方程组的消元步骤繁杂又容易犯错，不过利用 Python 的 NumPy 库之后，只要几行代码就能立即得出答案，如此一来不但提升了速度，还使得学生可以关注到建模这个阶段，经由具体实例，学生就能够清楚认识到从实际问题过渡到数学模型，再由数学模型转为计算机执行这样一套完整的步骤，从而领会到方程组在现实生活中的价值所在，进而加强自身的数学建模观念及其实际操作水平。

以中药特效药配制为例，某中药厂利用 9 种中草药按照不同比例配制出 7 种特效药，各种药物的用量见表 1 (单位：克)：

**Table 1.** Seven kinds of specific medicine ingredients  
**表 1.** 7 种特效药成分

| 中草药 | 1 号特效药 | 2 号特效药 | 3 号特效药 | 4 号特效药 | 5 号特效药 | 6 号特效药 | 7 号特效药 |
|-----|--------|--------|--------|--------|--------|--------|--------|
| 1   | 10     | 2      | 14     | 12     | 20     | 38     | 100    |
| 2   | 12     | 0      | 12     | 25     | 35     | 60     | 55     |
| 3   | 5      | 3      | 11     | 0      | 5      | 14     | 0      |
| 4   | 7      | 9      | 25     | 5      | 15     | 47     | 35     |
| 5   | 0      | 1      | 2      | 25     | 5      | 33     | 6      |
| 6   | 25     | 5      | 35     | 5      | 35     | 55     | 50     |
| 7   | 9      | 4      | 17     | 25     | 2      | 39     | 25     |
| 8   | 6      | 5      | 16     | 10     | 10     | 35     | 10     |
| 9   | 8      | 2      | 12     | 0      | 0      | 6      | 20     |

如果需要解决以下问题：3 号和 6 号特效药能否用其他特效药配制出来？如果用这 7 种特效药能否合成出下面三种新药？如何合成？三种新药成分见表 2。

**Table 2.** Ingredients of the three new drugs  
**表 2.** 3 种新药成分

| 中药 | 1 号新药 | 2 号新药 | 3 号新药 |
|----|-------|-------|-------|
| 1  | 40    | 162   | 88    |
| 2  | 62    | 141   | 67    |
| 3  | 14    | 27    | 8     |

续表

|   |    |     |    |
|---|----|-----|----|
| 4 | 44 | 102 | 51 |
| 5 | 53 | 60  | 7  |
| 6 | 50 | 155 | 80 |
| 7 | 71 | 118 | 38 |
| 8 | 41 | 68  | 21 |
| 9 | 14 | 52  | 30 |

7种特效药成分可以看成是一个9行7列的矩阵 $A$ ，设 $A = (\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7)$ ，三种新药成分构成一个9行3列的矩阵 $B$ ，设 $B = (\beta_1, \beta_2, \beta_3)$ 。第一个问题判断3号、6号特效药能否用其他特效药配制，其本质是求向量 $\alpha_3, \alpha_6$ 能否由 $\alpha_1, \alpha_2, \alpha_4, \alpha_5, \alpha_7$ 线性表示；第二个问题判断三种新药能否用7种特效药配制，其本质是求 $\beta_1, \beta_2, \beta_3$ 能否由 $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7$ 线性表示？如果可以，就能配制，如果不可以，就不能配制，通过python实现线性表示判定与计算。核心代码见附录1。

运行结果为：

行最简形矩阵

```
[[1 0 1 0 0 0 0 1 3 0]
 [0 1 2 0 0 3 0 3 4 0]
 [0 0 0 1 0 1 0 2 2 0]
 [0 0 0 0 1 1 0 0 0 0]
 [0 0 0 0 0 0 1 0 1 0]
 [0 0 0 0 0 0 0 0 0 1]
 [0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0]]
```

极大线性无关组所在列号

```
[1, 2, 4, 5, 7, 10]
```

问题 1: 3号、6号特效药能否由1、2、4、5、7号配制

3号特效药：可以配制

6号特效药：可以配制

运行结果表明，3、6号特效药可以由1、2、4、5、7号特效药配制而成，且 $\alpha_3 = \alpha_1 + 2\alpha_2$ ， $\alpha_6 = 3\alpha_2 + \alpha_4 + \alpha_5$ 。又 $\beta_1 = \alpha_1 + 3\alpha_2 + 2\alpha_4$ ， $\beta_2 = 3\alpha_1 + 4\alpha_2 + 2\alpha_4 + \alpha_7$ ，但 $\beta_3$ 不能由 $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7$ 线性表示，从而1、2号新药可以由7种特效药配制而成，但是3号新药不能配制。

通过Python实现线性表示判定与计算，直观体现线性方程组的实际价值。

#### 4.3.2. 案例 2：矩阵运算与线性回归

矩阵运算和线性拟合属于数据分析的基本工具，以合金强度与碳含量的关系作为场景，经由矩阵运算来做到线性回归建模，然后用Matplotlib对拟合结果执行可视化处理。

现从某产品中收集到合金强度和其中的碳含量数据见表3：

**Table 3.** Relationship between Alloy strength and Carbon content  
**表 3.** 合金强度与碳含量的关系

|      |      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|------|
| 碳含量  | 0.10 | 0.11 | 0.12 | 0.13 | 0.14 | 0.15 | 0.16 | 0.17 | 0.18 |
| 合金强度 | 42.0 | 41.5 | 45.0 | 45.5 | 45.0 | 47.5 | 49.0 | 55.0 | 50.0 |

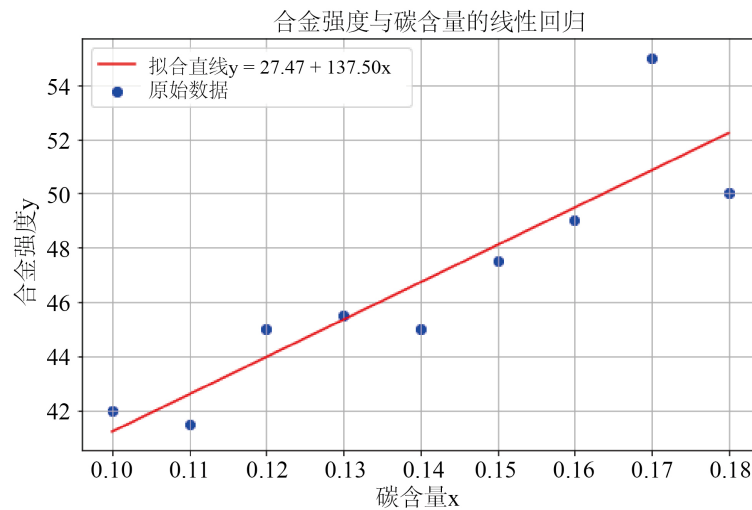
设碳含量为  $x$ ，合金强度为  $y$ ，试对  $x$  与  $y$  的关系进行线性回归分析  $y=ax+b$ ，确定  $a$  与  $b$  的值，并用图形表示出来。核心代码见附录 2，运行结果见图 1。

运行结果：

线性回归方程： $y = 27.47 + 137.50x$

截距  $a = 27.4722$

斜率  $b = 137.5000$



**Figure 1.** Linear regression of alloy strength vs carbon content  
**图 1.** 合金强度与碳含量的线性回归

该案例可供学生了解线性回归的矩阵原理及其达成途径，从而领会数据拟合在工程中的实际意义。

### 4.3.3. 案例 3：线性变换

线性变换表现了向量在矩阵作用下的映射关系，它是关联代数与几何的重要知识点，利用矩阵运算来做到图形的平移，旋转，缩放以及错切，既可以直观地表现出线性变换的几何意义，又可以让学生感受到数学工具在计算机图形处理方面所具有的实际价值，把 Python 的矩阵运算和绘图功能联系起来，就能够即时改变字符，图形等对象，从而有效地加强教学的直观性和实用性。

如：大写字母 H 由 12 个点确定，这些点的坐标可以储存在一个数据矩阵  $A$  中：

$$\begin{pmatrix} 0 & 0 & 1 & 1 & 4 & 4 & 5 & 5 & 4 & 4 & 1 & 1 \\ 0 & 9 & 9 & 5 & 5 & 9 & 9 & 0 & 0 & 4 & 4 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

用矩阵  $\begin{pmatrix} 1 & 0.5 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$  作用于  $H$  可得斜体  $H$ ，用矩阵  $\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$  作用于  $H$  可将  $H$  放大 2 倍，用矩阵

$$\begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

作用于 H 可将 H 逆时针旋转 45°。核心代码见附录 3，运行结果见图 2。

运行结果：

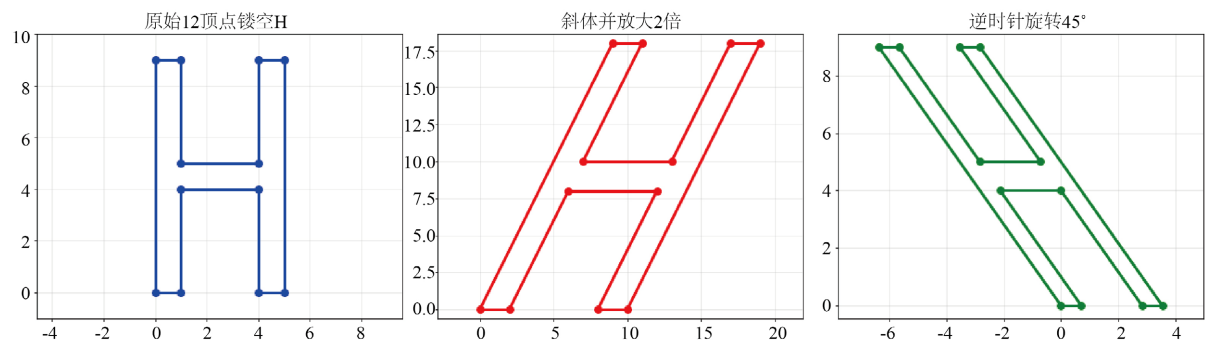


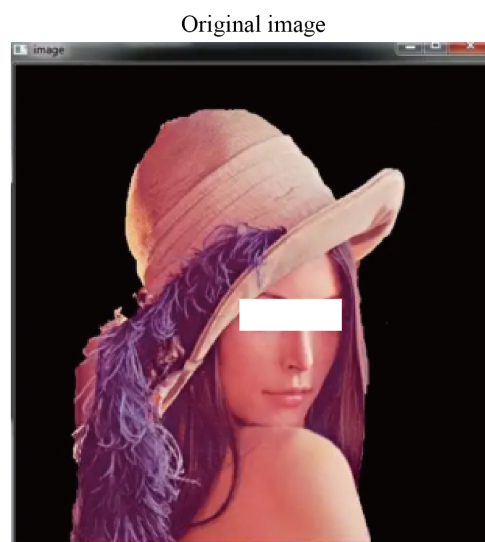
Figure 2. Linear transformations of letter H: italic, scaling and rotation

图 2. 字母 H 的线性变换：斜体、缩放与旋转

#### 4.3.4. 案例 4：奇异值分解(SVD)与图像压缩

奇异值分解(SVD)是 Python 里很常用的一种图像压缩方法。我们可以把一张图像看成由像素构成的矩阵，再用 SVD 把它分解为三个矩阵的乘积。实际压缩时，只保留数值较大的奇异值，忽略较小的奇异值，再用这些关键信息重构图像。这种做法能保留图像主要特征，明显减小数据量，在保证显示效果的同时提升资源利用率。压缩时会用到压缩比这一关键参数，可根据实际使用场景灵活调整，在文件大小与视觉效果之间取得平衡。相关实现代码见附录 4，运行结果见图 3。

运行结果：





**Figure 3.** Comparison of SVD-based on Lena image compression with different singular values  
**图 3.** 基于 SVD 的 Lena 图像不同奇异值压缩效果对比

由运行结果可以看出，SVD 图像压缩技术清晰展示了奇异值数量  $k$  与图像质量、压缩率的关系：当  $k$  从 5 逐步增加到 100，图像质量明显提升。 $k=5$  时，只能看到模糊的图像轮廓； $k=20$  时，可分辨出大致结构，但细节模糊； $k$  增大到 50 时，已能看清楚主体，只是细小的地方有一点点变形； $k$  继续增加到 100 时，画质基本和原图一致，肉眼很难看出差别。这一结论表明，图像的关键信息往往存在于几个较大的奇异值当中，取前 50 个奇异值就能做到用较小的储存空间实现高质量图像压缩的效果，而  $k$  值超过一定数值后，再继续增加，画质也不会有明显提升。

## 5. 评价体系改革

为全面考查学生对知识的掌握程度、实操能力与综合素养，我们对原有考核方式进行调整，搭建过程性评价 + 实践任务评价 + 期末综合评价的多元评价模式，具体流程如下：

1. 过程性评价(50%)：全程跟踪记录学生表现。课前通过线上平台布置预习内容和 Python 基础练习，由系统自动记录完成情况；课上通过提问、现场调试代码、小组汇报等方式进行实时考查；课后用常规作业、公式推导、代码纠错巩固学习。教师根据学生课堂态度、参与情况、代码熟练程度逐项打分，最后加权计算。

2. 实践任务评价(30%): 以任务带动学习, 安排编程实验和综合项目, 重点考查学生建模、编程与创新能力, 依据完成质量给出成绩。

3. 期末综合评价占(20%): 采用理论笔试加机考结合的方式, 侧重概念理解、建模思路与代码阅读, 考查学生对线性代数核心思想和 Python 实现逻辑的掌握情况。

这套评价体系能真实反映学生的学习成果, 引导学生主动学习与动手实践。

## 6. 挑战与反思

在教学实践中我们发现, 学生的 Python 基础差距比较大, 课堂时间有限, 部分案例和真实应用场景结合得不够紧密。针对这些情况, 我们会在课前做基础辅导, 按学生水平分层教学, 组织小组互助学习, 提前给大家准备好代码模板, 同时建设贴合不同专业的人工智能相关案例, 让整个教学过程更顺畅、更稳定。

## 7. 教学效果与分析

### 7.1. 学习兴趣显著提升

融合教学把抽象数学变成可视化, 可操作又可应用的内容, 课堂趣味性和直观性得到很大加强, 学生由以前被动听讲, 机械做题, 变为积极编写代码, 调试程序并观察效果, 学习积极性大幅上升, 课堂提问, 互动交流以及课后探究参与人数明显增多, 不少学生自发去尝试拓展案例, 把学到的知识用在处理自己图片, 数据等实际材料上, 学习体验得以改善, 这会直接促使学习兴趣优化, 学生对于线性代数的固有看法发生很大改变, 惧怕困难的情绪也大幅减轻。

### 7.2. 知识理解更加深入

学生利用 Python 进行可视化操作并执行相关代码之后, 可以全方位地从几何意义, 运算逻辑以及应用场景等方面去领会知识点, 不会再仅仅局限于记忆公式这个阶段, 同传统的课堂相比, 学生对于矩阵变换, 特征值, 奇异值分解这些重点难点内容的领悟会更为精准, 透彻, 而且能够用自身的话来阐述概念的内涵, 并讲述它在工程当中所起到的作用, 在做作业和参加考试的时候, 学生应对概念类题目, 应用类题目的答题水平得到了很大的改善, 思路变得越发明晰, 表述也越发恰当, 这显示出其知识的吸收消化程度以及认识深度确实得到了有效的加强。

### 7.3. 综合能力同步增强

学生修完相关课程之后, 不但可以收获线性代数方面的知识, 而且还能具备基本的 Python 编程能力, 数学建模能力以及问题解决能力, 过半学生能够独自去完成一些较为简单的数据处理, 图像转换, 方程求解之类的程序, 已经具有把实际问题转为成数学模型, 并在计算机上去执行的初步能力, 这样一种数学素养, 编程能力与工程思维一同得到加强的情况, 正符合新工科以及人工智能时代对于人才塑造的关键需求, 学生在后面的专业课程学习过程中, 会更快领悟算法原理, 显示出更强的学习适应能力和更大的发展潜力。

## 8. 结论

人工智能技术极速发展, 现代工程教育创建不断深入, 在此大背景之下, 线性代数这门课程需由传统的理论讲解朝着实用化、工程化以及跨学科融合的方向去转变, Python 语言具有简洁的语法, 众多的库函数以及很强的可视化功能, 是把线性代数理论同工程应用联系起来的理想工具。经由重新塑造教学

内容,更新教学模式,开发典型案例并改革评价体系,形成起“理论-代码-应用”相融合的一体化教学体系,可以有效地解决传统教学中抽象概念难以领会,手工计算效率低下,知识与操作脱离实际等难题,教学应用显示,这种融合的教学模式能够明显改善学生的学习兴趣,加深他们对知识的领悟程度,加强他们的编程能力和数学应用能力,从而加强课程的深度,创新性与挑战性,进而符合新时代复合型创新人才的培养需求,以后还要不断地加深教学研究,充实案例资源和教学平台,以进一步优化教学质量,给高校数学课程改革提供更为成熟的操作经验。

## 基金项目

2024年模式识别与智能信息处理高校重点实验室开放基金项目:基于分数阶微积分理论的粘弹性梁裂纹识别的算法研究,项目编号:MSSB-2024-11;天府文化数字化创新四川省文化和旅游厅重点实验室开放课题:天府文化传播的动态模拟研究,项目编号:TFWH-2024-25。

## 参考文献

- [1] 余金. Python语言在数据分析处理中的应用[J]. 电脑编程技巧与维护, 2022(6): 18-20.
- [2] 王恒, 方晓峰, 张辉. Python在线性代数知识点可视化中的应用[J]. 电脑知识与技术, 2025, 21(16): 124-128.
- [3] 韩文静, 董建鑫, 周辰, 胡玲静. 人工智能背景下医学影像技术专业线性代数课程教学改革与实践[J]. 数理医药学杂志, 2024, 37(10): 799-805.
- [4] 邵丽丽. 新工科背景下计算机类专业线性代数课程教学研究[J]. 科学咨询(科技管理), 2021(36): 285-286.
- [5] 范丽媛, 李明翔. 基于《线性代数》课程与Python相融合的教学模式探索[J]. 塑料包装, 2025, 35(3): 259-261, 120.
- [6] 邵松涛. Python在线性代数教学中的应用案例研究[J]. 科技资讯, 2025, 23(11): 189-192.
- [7] 康颢, 沈瑶, 王博文, 赵彦珍. 基于Python的线性动态电路可视化分析软件设计与实现[J]. 实验室研究与探索, 2022, 41(2): 116-120.
- [8] 李清华, 王宝娟. 线性代数知识点的可视化教学设计探索与实践[J]. 大学数学, 2022, 38(2): 112-119.

## 附 录

### 1. Python 在线性方程组求解与药品配制中的应用

核心代码:

```
import numpy as np
import sympy as sp

A = [
    [10, 2, 14, 12, 20, 38, 100, 40, 162, 88],
    [12, 0, 12, 25, 35, 60, 55, 62, 141, 67],
    [5, 3, 11, 0, 5, 14, 0, 14, 27, 8],
    [7, 9, 25, 5, 15, 47, 35, 44, 102, 51],
    [0, 1, 2, 25, 5, 33, 6, 53, 60, 7],
    [25, 5, 35, 5, 35, 55, 50, 50, 155, 80],
    [9, 4, 17, 25, 2, 39, 25, 71, 118, 38],
    [6, 5, 16, 10, 10, 35, 10, 41, 68, 21],
    [8, 2, 12, 0, 0, 6, 20, 14, 52, 30]
]

mat = sp.Matrix(A)
rref_mat, pivot_cols = mat.rref()

rref_np = np.array(rref_mat).astype(float)
rref_final = np.round(rref_np).astype(int)

print("行最简形矩阵")
print(rref_final)

print("极大线性无关组所在列号")
pivot_1 = [x+1 for x in pivot_cols]
print(pivot_1)

print("问题 1: 3 号、6 号特效药能否由 1、2、4、5、7 号配制")

M = np.array(A, dtype=float)
available = M[:, [0,1,3,4,6]]
drug3 = M[:, 2]
drug6 = M[:, 5]

x3, *_ = np.linalg.lstsq(available, drug3, rcond=None)
```

```
x6, *_ = np.linalg.lstsq(available, drug6, rcond=None)

can3 = all(x >= -1e-6 for x in x3)
can6 = all(x >= -1e-6 for x in x6)

print(f"3号特效药: {'可以配制' if can3 else '不能配制'}")
print(f"6号特效药: {'可以配制' if can6 else '不能配制'}")
```

## 2. Python 在矩阵运算与线性回归中的应用

核心代码:

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.font_manager import FontProperties
import warnings

x = np.array([0.10, 0.11, 0.12, 0.13, 0.14, 0.15, 0.16, 0.17, 0.18])
y = np.array([42.0, 41.5, 45.0, 45.5, 45.0, 47.5, 49.0, 55.0, 50.0])
n = len(x)

sum_x = np.sum(x)
sum_y = np.sum(y)
sum_xy = np.sum(x * y)
sum_x2 = np.sum(x ** 2)

b = (n * sum_xy - sum_x * sum_y) / (n * sum_x2 - sum_x ** 2)
a = np.mean(y) - b * np.mean(x)
print(f"线性回归方程: y = {a:.2f} + {b:.2f}x")
print(f"截距 a = {a:.4f}")
print(f"斜率 b = {b:.4f}")

font = FontProperties(fname=r"C:\Windows\Fonts\simhei.ttf", size=12)
plt.figure(figsize=(8, 5))
ax = plt.gca()
ax.set_title('合金强度与碳含量的线性回归', fontproperties=font)
ax.set_xlabel('碳含量 x', fontproperties=font)
ax.set_ylabel('合金强度 y', fontproperties=font)
ax.tick_params(axis='both', labelsize=12)

ax.scatter(x, y, color='blue', label='原始数据')
```

```
ax.plot(x, a + b * x, color='red', label=f'拟合直线 y = {a:.2f} + {b:.2f}x')
ax.legend(prop=font)
ax.grid(True)
plt.show()
```

### 3. Python 在线性变换中的应用

核心代码:

```
import numpy as np
import matplotlib.pyplot as plt

D = np.array([
    [0, 0, 1, 1, 4, 4, 5, 5, 4, 4, 1, 1],
    [0, 9, 9, 5, 5, 9, 9, 0, 0, 4, 4, 0],
    [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
])
edges = [
    [0, 1], [1, 2], [2, 3], [3, 4], [4, 5],
    [5, 6], [6, 7], [7, 8], [8, 9], [9, 10],
    [10,11],[11,0]
]

Shear = np.array([[1, 0.5, 0],
                  [0, 1, 0],
                  [0, 0, 1]])
Scale = np.array([[2, 0, 0],
                  [0, 2, 0],
                  [0, 0, 1]])
T_shear_scale = Scale @ Shear
D1 = T_shear_scale @ D
theta = np.pi / 4
R = np.array([
    [np.cos(theta), -np.sin(theta), 0],
    [np.sin(theta), np.cos(theta), 0],
    [0, 0, 1]
])
D2 = R @ D

plt.figure(figsize=(18, 5))
plt.subplot(1, 3, 1)
x0, y0 = D[0], D[1]
```

```
for e in edges:
    plt.plot(x0[e], y0[e], 'b-', linewidth=3)
plt.scatter(x0, y0, c='blue', s=60, zorder=5)
plt.title('原始 12 顶点镂空 H', fontsize=14)
plt.grid(True, alpha=0.3)
plt.axis('equal')
plt.xlim(-1, 6)
plt.ylim(-1, 10)
```

```
plt.subplot(1, 3, 2)
x1, y1 = D1[0], D1[1]
for e in edges:
    plt.plot(x1[e], y1[e], 'r-', linewidth=3)
plt.scatter(x1, y1, c='red', s=60, zorder=5)
plt.title('斜体并放大 2 倍', fontsize=14)
plt.grid(True, alpha=0.3)
plt.axis('equal')
```

```
plt.subplot(1, 3, 3)
x2, y2 = D2[0], D[1]
for e in edges:
    plt.plot(x2[e], y2[e], 'g-', linewidth=3)
plt.scatter(x2, y2, c='green', s=60, zorder=5)
plt.title('逆时针旋转 45° ', fontsize=14)
plt.grid(True, alpha=0.3)
plt.axis('equal')
plt.tight_layout()
plt.show()
```

#### 4. 奇异值分解(SVD)与图像压缩

核心代码:

```
import numpy as np
from scipy.linalg import svd
from PIL import Image
import matplotlib.pyplot as plt
import os
import warnings
#SVD 图像压缩函数
def svd_image_compression(image_array, k):
```

```
if len(image_array.shape) == 3:
    compressed = []
    for i in range(3):
        U, S, Vt = svd(image_array[:, :, i], full_matrices=False)
        S_k = np.diag(S[:k])
        channel = U[:, :k] @ S_k @ Vt[:, :]
        channel = np.clip(channel, 0, 255).astype(np.uint8)
        compressed.append(channel)
    return np.stack(compressed, axis=2)
else:
    U, S, Vt = svd(image_array, full_matrices=False)
    S_k = np.diag(S[:k])
    img_compressed = U[:, :k] @ S_k @ Vt[:, :]
    return np.clip(img_compressed, 0, 255).astype(np.uint8)
#主程序
if __name__ == "__main__":
    image_path = r"C:\Users\win10\Desktop\lena.png"
    #加载图片
    img = Image.open(image_path)
    img_array = np.array(img)
    print(f"✅ 成功加载图片，尺寸：{img_array.shape}")
    #显示原图
    plt.figure(figsize=(6, 6))
    plt.title("Original Image")
    plt.imshow(img_array)
    plt.axis("off")
    plt.show()
    #不同压缩程度演示
    k_values = [5, 20, 50, 100]
    plt.figure(figsize=(12, 10))
    for idx, k in enumerate(k_values):
        compressed_img = svd_image_compression(img_array, k)
        plt.subplot(2, 2, idx + 1)
        plt.title(f"Compressed with k={k}")
        plt.imshow(compressed_img)
        plt.axis("off")
    plt.tight_layout()
    plt.show()
```