矩阵对角化与奇异值分解：静态 SageMath 与动态 ChatGPT 的对照

Natanael Karjanto

成均馆大学自然科学校区大学学院数学系，韩国大韩民国水原16419

更新于2023年3月31日

arXiv:2303.17163v1 [math.HO] 30 Mar 2023

摘要

我们研究了本科阶段学习线性代数时学生经常面临的一些困难，并确定了他们在处理需要算法思维技能的主题（如矩阵分解）时经常出现的一些常见错误和困难。具体而言，我们聚焦于（正交）对角化和奇异值分解（SVD）。我们还提供了使用 SageMath 探索这些主题的可能性。SageMath 是一种基于 Python 的免费开源计算机代数系统（CAS），尽管其输出本质上是静态的，但已被证实有助于许多学生完成计算过程。随后，我们通过询问聊天机器人相关主题来探索动态 ChatGPT，例如要求其提供一个示例或解决一个问题，即根据特定矩阵构造（正交）对角化或 SVD。通过巩固线性代数的基本概念，并通过有效练习提升计算技能，掌握这些主题将变得更加容易，错误也可以得到最小化。尤其是静态 SageMath，它是确认计算结果和处理繁琐计算的有力辅助工具。尽管动态 ChatGPT 在解决线性代数问题方面相对不可靠，但它所产生的错误可以成为提升批判性思维技能的宝贵工具。

1 引言

线性代数是数学的一个分支，研究线性方程、线性函数及其在向量和矩阵中的表示。它涉及向量空间、线性变换、矩阵、行列式、特征值和特征向量的研究。线性代数是数学许多领域中的基础工具，包括几何学、微积分、优化和数值分析。它在物理学、工程学、计算机科学和经济学等多个领域也有实际应用 [1–20]。

与数学的其他领域类似，在线性代数中，算法思维是理解和掌握课程内容的一项关键技能。算法思维通常也称为计算思维，是将复杂数学问题分解为更小、更易处理的部分，然后按照一系列步骤或算法系统地求解的过程。这一过程包括识别模式、构建算法，以及培养解决这些问题所需的逻辑和分析能力 [21–24]。面向数学专业和非数学专业学生的线性代数教学与学习，为培养和运用问题解决、逻辑推理和计算思维能力提供了机会；这些技能对于运筹学、计算机科学、数据科学和机器学习等需要扎实线性代数知识的科学与工程领域至关重要 [3,20,25–33]。加强未来教师的计算思维能力，也有助于他们日后培养新一代学生 [34,35]。

除了能够以代数方式处理矩阵并执行矩阵运算之外，线性代数中的一些基本概念还包括但不限于理解向量空间和子空间、应用线性变换、处理范数和内积、求解最小二乘解，以及执行矩阵分解的能力。在本研究中，我们重点关注后者。在线性代数中，分解有时也称为因式分解，指的是将一个矩阵分解为若干个通常更简单的矩阵的过程，这些矩阵更易于分析和处理。这些更简单的矩阵

可能具有特殊的性质或结构，从而更易于分析或进行计算。对于每一位线性代数学习者而言，掌握执行矩阵分解时的算法思维至关重要，因为这一主题不仅要求掌握线性代数中的其他基础概念，也是理解线性代数中其他更复杂主题的重要基础。

矩阵分解有多种类型，例如 LU分解、QR分解、对角化、正交对角化、奇异值分解（奇异值分解）和特征分解。后者是指将矩阵分解为一种标准形式，其中矩阵由其特征值和特征向量表示。每种分解都有各自特定的性质和应用。例如，LU分解适用于求解线性方程组，QR分解用于解决最小二乘问题，奇异值分解用于数据压缩和特征提取，而特征分解则用于分析线性算子的行为。矩阵对角化仅适用于方阵，正交对角化仅适用于一类特殊的对称方阵，而奇异值分解可用于任意大小的矩形矩阵，这类矩阵不一定是方阵。

简而言之，奇异值分解是线性代数中的一种分解方法，可将实值或复值矩阵分解为三个部分：由奇异值构成的对角矩阵，以及两个酉矩阵（正交矩阵）。奇异值分解在数学上的应用包括计算（Moore-Penrose）伪逆、近似矩阵，以及确定矩阵的秩、值域和零空间。奇异值分解广泛应用于数据分析和机器学习，用于降低数据维度并提取重要特征。奇异值分解还应用于科学和工程学的多个领域，包括信号与图像处理、文本挖掘、数据最小二乘拟合、过程控制和推荐系统 [36–46]。奇异值分解得到的奇异值表示原始数据中各特征的重要性，可用于以不同精度重建数据。

线性代数中的常见课程安排是，在向学生介绍奇异值分解（SVD）之前，学生应当掌握矩阵对角化和正交对角化的计算方法。然而，我们观察到，学生在进行这些分解时常常遇到许多困难。对于正交对角化，一些学生经常忘记将相关特征向量归一化为单位向量。在其他情况下，当一个特征值对应两个特征向量时，一些学生也会忘记将特征向量集转换为正交归一集；利用 Gram-Schmidt过程进行正交投影即可轻松完成这一转换，即保留其中一个向量，同时对另一个向量进行投影。

对于奇异值分解（SVD），一个常见错误是求矩阵的奇异值。设 A 是一个矩阵，学生不去求 A^TA 的特征值，而是求 A 的特征值，取其平方根，并将它们指定为奇异值。当矩阵不满秩时，还会出现另一个困难；此时，学生需要利用正交性质或叉积运算，在其中一个矩阵中求出缺失的一个或多个正交归一特征向量。如果缺少这些特征向量，所得分解就称为“约化奇异值分解”，而不是（完整的）奇异值分解。

据我们所知，已发表的文献尚未充分探讨学习者在理解（正交）对角化和奇异值分解时常犯的错误及所面临的困难。与本研究略有关联的最接近文献，是 Yildiz Ulus 于 2013年发表的论文。作者在文中探讨了如何借助高级计算器教授对角化，并指出，这类技术工具有助于学习者掌握这一线性代数主题中的算法性数学知识 [47]。虽然 Lazar 在其 2012年硕士论文中指出，扎实理解线性代数的基本概念对于掌握奇异值分解至关重要，但作者并未询问研究参与者在学习这一主题时遇到的困难或常见错误 [48]。

本文试图通过考察学生学习线性代数，尤其是（正交）对角化和奇异值分解时常见的错误与挑战，来填补这一空白。此外，由于当代数学的教学与学习无法通过使用技术工具来割裂，我们也希望研究计算机软件或其他新近出现的技术是否能够帮助并促进学生的学习，而不是对其造成干扰和损害。对于计算机代数系统（CAS），我们的研究范围是具有静态性质的 SageMath；同时，我们将 ChatGPT 视为一种人工智-

能（AI）辅助工具，具备动态交互功能。下文将简要介绍静态计算机代数系统 SageMath 和动态的 ChatGPT。

什么是计算机代数系统？计算机代数系统是一种软件程序，可用于处理和计算数学表达式与符号，包括代数方程、微积分及其他数学函数。它旨在执行符号运算、数值计算和图形绘制，并提供求解方程、处理矩阵和向量以及执行其他数学运算的工具。计算机代数系统常用于科学研究、工程学和教育，可用于解决复杂的数学问题，这些问题可能难以手工求解，或需要耗费大量时间 [49–63]。常见的计算机代数系统包括 (wx)Maxima [64–68], WolframMathematica [69–73],Maplesoft Maple [74–78], 和 Mathworks MATLAB [79–83]。

什么是 SageMath？SageMath（又称 Sage）是一款免费的开源数学软件系统，以 Python 作为主要编程语言。它旨在成为商业数学软件系统的替代方案，例如 Wolfram Mathematica 和 Maplesoft Maple，同时也提供与其他流行数学软件系统的接口，例如 Mathworks MATLAB 和用于计算离散代数的系统 GAP。*SageMath* 功能广泛，包括代数和数值计算、图形、符号运算及组合数学。它由社区驱动开发，并依据 GNU通用公共许可证 发布 [84–89]。

有趣的是，SageMath 不仅可用于线性代数，而且对这一领域提供了非常强大的支持，内置了许多用于处理矩阵、向量和线性变换的函数与功能。SageMath 提供了多种用于解决线性代数问题的算法和方法，例如执行矩阵运算、求解线性方程组、寻找特征值和特征向量，以及计算矩阵分解（例如 LU分解、QR分解和奇异值分解）。SageMath 是一种功能强大的符号线性代数工具，支持对变量和表达式进行精确计算。尽管 SageMath 本质上是静态的，我们仍然相信它可用于线性代数的教学与学习，不仅能够协助完成计算任务，还能帮助检查我们的手算结果是否正确。此外，节省下来的计算时间还可用于其他目的，例如深入理解线性代数中的数学概念，或探索该主题中的各种问题。

什么是 ChatGPT？ChatGPT（聊天生成式预训练变换器）是由 OpenAI 开发的一种语言模型。OpenAI 是一家美国人工智能研究实验室，由一群致力于创造安全且有益的人工智能的研究人员和工程师组成。尽管 OpenAI 成立于 2015年，ChatGPT 作为其产品之一，于 2022年11月以原型形式发布。令人惊叹的是，它在发布后的五天内就获得了 100 万名用户。ChatGPT 是一种构建于无监督变换器语言模型 GPT-3 之上的人工智能工具，经过海量文本数据集的训练，能够回答问题、撰写连贯的段落，甚至与用户进行对话。它旨在协助完成各种任务，例如回答问题、提供信息，或以对话方式生成文本。

有趣的是，作为一种人工智能语言模型，ChatGPT 接受了包括数学概念和数学问题在内的大量文本语料训练。因此，它能够解决数学问题，至少它自己是这样声称的。ChatGPT 承认，其解决数学问题的能力可能会受到训练数据的限制，不过它当然可以在各种数学主题上提供帮助和指导。具体而言，当被问及是否能够解决线性代数问题时，ChatGPT 声称“掌握了线性代数概念的知识与理解，并且能够为这一领域的问题提供解决方案”。在过去几个月中，与 ChatGPT 及其能力相关的已发表文章数量持续增加，其中许多文章发表于各种预印本服务器。下面的示例仅涵盖少数几篇与 ChatGPT 相关的数学文章。

最近，Frieder 等人（2023）通过提出各种各样的问题，研究了 ChatGPT 的数学能力。尽管它能理解问题，却常常无法给出正确答案。他们得出结论，ChatGPT 的数学能力显著低于普通数学研究生 [90]。Shakarian 等人（2023）也评估了 ChatGPT 解答数学文字题的表现，并发现其表现会因是否要求展示解题过程而大幅变化。当不要求聊天机器人透露完整解答时，它有 84% 的几率答错；而要求展示详细解答时，答错率仅为 20% [91]。Azaria（2022）测试了

ChatGPT 的数字素养，并注意到在使用数字时，ChatGPT 存在明显偏向：机器生成频率最高的数字是 7，而这也恰好是许多人最喜欢的数字 [92]。Borji（2023）重点研究了 ChatGPT 在数学方面的不足，包括算术、逻辑和推理等 [93]。

在输出生成方面，SageMath 和其他计算机代数系统往往是静态的，而 ChatGPT 及其竞争者则是动态的。后者可以逐步解释某个特定问题的解法。对于线性代数中的矩阵分解，学习者不仅需要理解计算的技术细节，还必须掌握特定问题任务背后的算法。当然，矩阵分解本身是一个广泛的主题，任何讨论其他类型矩阵分解的尝试都应在其他地方另行展开。目前，我们对“矩阵分解”这一主题的关注点仅涵盖对角化、正交对角化和奇异值分解。基于上述文献，我们提出以下研究问题：

- 学生在线性代数中学习矩阵分解时会遇到哪些常见错误和困难?
- 静态计算机代数系统（例如 SageMath）如何帮助学生学习线性代数，尤其是（正交）对角化和奇异值分解?
- 我们能否依靠动态的 ChatGPT，更好地理解（正交）对角化和奇异值分解?

本文其余部分安排如下。引言之后，第 2 节介绍学生学习（正交）对角化和奇异值分解时常犯的一些错误。第 3 节接着探讨 SageMath 及其辅助学生学习（正交）对角化和奇异值分解的能力。第 4 节举例说明我们曾要求聊天机器人解决与（正交）对角化和奇异值分解相关的问题。我们还讨论了 ChatGPT 会在哪些方面出错，以及在完成指定任务时会遇到哪些困难。最后，第 5 节对本研究进行讨论并作出总结。

2 矩阵分解中的常见错误

2.1 矩阵对角化

本节讨论学生学习对非对称矩阵和对称矩阵进行对角化时常遇到的一些错误，也探讨奇异值分解中的类似问题。对于前者，若 A 是一个 $n \times n$ 方阵，那么只要 A 拥有 n 个线性无关特征向量，它就是可对角化的。通常，只要学生理解求解步骤，在对非对称的可对角化矩阵进行对角化时就不会遇到困难。典型的算法首先求出特征值及其对应的特征向量，然后构造对角矩阵 D，再构造可逆矩阵 P，求出其逆矩阵 P^{-1}，并写出对角化形式，即 $A = PDP^{-1}$。下面的例子展示了这一算法。

给定如下 3×3 矩阵A：

$$A = \begin{bmatrix} 1 & 2 & 2 \\ 0 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}.$$

要将矩阵对角化，我们必须确定其特征值及其对应的特征向量。前者可以通过求解特征方程 $\det(A - \lambda I) = 0$ 或 $(\lambda - 1)(\lambda - 3)(\lambda - 1) = 0$得到，该方程给出 $\lambda_1 = 1$、$\lambda_2 = 3$ 和 $\lambda_3 = 1$。需要注意的是，一些学生可能会尝试将特征方程表示为三次形式，即 $\lambda^3 - 5\lambda^2 + 7\lambda - 3 = 0$，并据此求解 λ。尽管在其他情况下，这可能是必要的步骤，甚至不可避免，但在这个具体示例中，我们不必执行这一步。到目前为止，我们可以通过将特征值置于对角线上来构造对角矩阵 D：

$$D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

然而，这并不是唯一的选择。根据指定为第一个、第二个、第三个特征值等的选择不同，我们可能会得到 D 的不同表达式。巧合的是，在本例的矩阵中，特征值 $\lambda = 1$ 的代数重数和几何重数相等，均为 2。在这种情况下，我们可以确保得到三个线性无关特征向量，因此该矩阵是可对角化的。

令$\mathbf{p}_1$，$\mathbf{p}_2$，和$\mathbf{p}_3$ 为相应的特征向量，我们发现

$$\mathbf{p}_1 = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}, \qquad \mathbf{p}_2 = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}, \qquad \mathbf{p}_3 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

A随后，可以通过将这些特征向量作为列堆叠起来，形成一个可逆矩阵P：

$$P = \begin{bmatrix} 0 & 2 & 1 \\ 1 & 1 & 0 \\ -1 & 1 & 0 \end{bmatrix}.$$

其逆矩阵P^{-1} 为

$$P^{-1} = \begin{bmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 1 & -1 & -1 \end{bmatrix}.$$

最后，我们可以将A 表示为PDP^{-1} ，从而写出A 的对角化形式，并验证乘积还原为原矩阵A：

$$\underbrace{\begin{bmatrix} 1 & 2 & 2 \\ 0 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 0 & 2 & 1 \\ 1 & 1 & 0 \\ -1 & 1 & 0 \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 1 & -1 & -1 \end{bmatrix}}_{P^{-1}}.$$

2.2 正交对角化

以下示例展示了对称矩阵的正交对角化，其中一个特征值的代数重数和几何重数均为二，因此所得的特征向量并不正交。考虑以下 3×3 矩阵B：

$$B = \begin{bmatrix} 3 & -4 & -4 \\ -4 & 3 & -4 \\ -4 & -4 & 3 \end{bmatrix}.$$

该矩阵的特征多项式为 $\lambda^3 - 9\lambda^2 - 21\lambda + 245$，解特征方程后可得三个实特征值： $\lambda_1 = -5$ 和 $\lambda_2 = 7 = \lambda_3$。对角矩阵 D 为

$$D = \begin{bmatrix} -5 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 7 \end{bmatrix}.$$

设 $\mathbf{q}_1$，$\mathbf{q}_2$，和 $\mathbf{q}_3$ 为对应于这些特征值的特征向量，则有

$$\mathbf{q}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \qquad \mathbf{q}_2 = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}, \qquad \mathbf{q}_3 = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}.$$

我们经常遇到的一个常见错误是，有些学生不对这些特征向量进行归一化，就直接用它们构造可逆矩阵P ，然后仅通过转置来求它的逆矩阵，也就是

$P^T = P^{-1}$。然而，正确的做法是构造一组正交归一特征向量，并将它们作为P的列。我们观察到，$\mathbf{q}_1$ 与 $\mathbf{q}_2$ 和$\mathbf{q}_3$都正交。然而，$\mathbf{q}_2$ 与$\mathbf{q}_3$ 并不正交，因为它们来自同一个特征子空间。只有与不同特征值对应的特征向量才彼此正交。对集合$\{\mathbf{q}_2,\mathbf{q}_3\}$应用Gram-Schmidt过程，我们可以得到一组新的正交特征向量 $\{\mathbf{q}_2,\mathbf{q}'_3\}$，其中

$$\mathbf{q}'_3 = \mathbf{q}_3 - \frac{\langle \mathbf{q}_2, \mathbf{q}_3 \rangle}{\langle \mathbf{q}_2, \mathbf{q}_2 \rangle}\mathbf{q}_2 = \frac{1}{2}\begin{bmatrix} -1 \\ 2 \\ -1 \end{bmatrix}.$$

尖括号表示通常的内积（点积）。通过对这些向量进行归一化，我们得到一个正交归一特征向量组，并据此构造矩阵 P 。由于 P 现在是一个正交矩阵，其逆矩阵就是它的转置，如下所示：

$$P = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & -\frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \end{bmatrix}, \qquad P^{-1} = P^T = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{6}} & -\frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \end{bmatrix}.$$

现在，我们可以表示 B 的正交对角化，并再次确认，右侧经过乘法和化简后将还原为原矩阵：

$$\underbrace{\begin{bmatrix} 3 & -4 & -4 \\ -4 & 3 & -4 \\ -4 & -4 & 3 \end{bmatrix}}_{B} = \underbrace{\begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & -\frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} -5 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & 7 \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{6}} & -\frac{2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \end{bmatrix}}_{P^{-T}}.$$

2.3 奇异值分解

矩阵对角化和正交对角化只能应用于方阵，而奇异值分解同时适用于方阵和矩形矩阵。设 M 为一个 $m \times n$ 实值矩阵，其秩为 r，则 M 的奇异值分解为 $M = U\Sigma V^T$，其中 U 是一个 $m \times m$ 正交矩阵， Σ 是一个 $m \times n$ 矩形“对角”矩阵，其对角线上为非负实数，而 V 是一个 $n \times n$ 正交矩阵。这种分解中 U 和 V 的列分别称为 M的左、右奇异向量。对于复值矩阵，U和 V 是复酉矩阵，此时不再使用 V^T，而使用 V的共轭转置，即 V^*。一般而言， Σ由矩阵块组成，并具有以下形式：

$$\Sigma = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix},$$

其中D 是一个$r \times r$ 对角矩阵，其中$r \leq \min\{m,n\}$。该分块矩阵的第二行和第二列中（零）行和列的数量分别为 $(m-r)$ 和 $(n-r)$。当$r = m$、$r = n$或$r = m = n$时，部分或全部零块会消失。此外，D 的对角元素是M的前r 个奇异值，即 $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$。

与求得矩阵对角化和正交对角化类似，构造矩阵的奇异值分解需要按步骤执行算法，并理解（正交）对角化。对于满秩方阵，许多学习者通常在构造矩阵的奇异值分解时不会遇到困难，但计算对称矩阵$M^T M$的特征值及相应特征向量时，必须格外谨慎，因为这里计算的不是原矩阵M的特征值及特征向量。然而，对于矩形矩阵或秩亏矩阵，许多学生往往无法完成奇异值分解的构造，部分原因是一个或多个奇异值可能为零。下面的例子说明如何构造秩亏方阵的奇异值分解。设

$$M = \begin{bmatrix} 3 & -1 \\ -6 & 2 \end{bmatrix}$$

为一个秩为$r = 1$的矩阵，则$A = M^T M$的特征值为 $\lambda_1 = 50$和 $\lambda_2 = 0$，因此只有一个奇异值 $\sigma_1 = 5\sqrt{2}$，因为第二个特征值为零。因此，$D = \sigma_1^-$，矩阵Σ 在D的右侧和下方的每一行、每一列中都包含一个零块，如下所示：

$$\Sigma = \begin{bmatrix} 5\sqrt{2} & 0 \\ 0 & 0 \end{bmatrix}.$$

通过求出 λ_1 和 λ_2对应的特征向量，我们得到$\mathbf{v}_1$ 和$\mathbf{v}_2$，并构造正交矩阵V:

$$V = \begin{bmatrix} 3/\sqrt{10} & 1/\sqrt{10} \\ -1/\sqrt{10} & 3/\sqrt{10} \end{bmatrix}.$$

矩阵U的第一列，即$\mathbf{u}_1$，可以通过一个公式计算得出

$$\mathbf{u}_1 = \frac{1}{\sigma_1} M\mathbf{v}_1 = \frac{1}{5\sqrt{2}} \begin{bmatrix} 3 & -1 \\ -6 & 2 \end{bmatrix} \frac{1}{\sqrt{10}} \begin{bmatrix} 3 \\ -1 \end{bmatrix} = \frac{1}{\sqrt{5}} \begin{bmatrix} 1 \\ -2 \end{bmatrix}.$$

矩阵U的第二列，即$\mathbf{u}_2$，必须与$\mathbf{u}_1$正交：

$$\mathbf{u}_2 = \frac{1}{\sqrt{5}} \begin{bmatrix} 2 \\ 1 \end{bmatrix}.$$

M 的奇异值分解如下：

$$\underbrace{\begin{bmatrix} 3 & -1 \\ -6 & 2 \end{bmatrix}}_{M} = \underbrace{\begin{bmatrix} 2/\sqrt{5} & 1/\sqrt{5} \\ 1/\sqrt{5} & -2/\sqrt{5} \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} 5\sqrt{2} & 0 \\ 0 & 0 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} 3/\sqrt{10} & -1/\sqrt{10} \\ 1/\sqrt{10} & 3/\sqrt{10} \end{bmatrix}}_{V^T}.$$

3 使用 SageMath 进行（正交）对角化和奇异值分解

要在线使用 SageMath，我们可以使用 Sage Cell Server 或 SageMathCell，可通过网址 https://sagecell.sagemath.org/在线访问。除了提供开源、可扩展且易于使用的 SageMath Web 界面外，这个单元服务器还允许将 Sage 计算嵌入任何网页。无需加载任何程序，只要有互联网连接，这就是一种使用 SageMath 执行临时计算的方式，其目标是尽可能简单地通过云端访问计算结果。

3.1 使用 SageMath 进行矩阵对角化

以下　　给出了第2.1小节中讨论的矩阵对角化 SageMath 命令。

```
A=矩阵([[1,2,2],[0,2,1],[0,1,2]]) print("矩阵A = ") print(A)
print() print("A 的特征多项式：p(x) =",A.charpoly()) print() print("
A 的特征值 =", A.eigenvalues()) print() print("特征值、特征向量和几
何重数：", ) print(A.eigenvectors_right()) print() lambda1 =
A.eigenvalues()[0]lambda2 = A.eigenvalues()[1]lambda3 =
A.eigenvalues()[2]
```

```
print("提取特征值：") print("l
ambda1 =", lambda1) print("
lambda2 =", lambda2) print("
lambda3 =", lambda3) print() D=对角
_矩阵(A.eigenvalues()) print("对角矩
阵 D = ") print(D) print() r1=
A.eigenvectors_right()[0][1]p1=2*r
1[0]r2=A.eigenvectors_right()[1][1]
p2=r2[0]r3=A.eigenvectors_right()
[1][1]p3=r3[1]print("提取特征向量：")
print("p1 =", p1) print("p2 =", p2)
print("p3 =", p3) print() P=矩阵([
p1,p2,p3]).转置() print("可逆矩阵 P =
") print(P) print() print("P 的逆矩阵,
P^(-1) = ") print(P.inverse())
print() print("计算 PDP^(-1) =")
print(P*D*P.inverse()) print("= A")
```

知道特征值及其对应的特征向量后，对角矩阵*D* 和可逆矩阵*P* 的构造便可立即完成。事实上，这一过程必须手动进行，但并不困难。此外，使用 Sage 的命令“`P.inverse()`”即可自然地求出*P* 的逆矩阵。前面代码行的相应输出如下：

```
矩阵 A =
[1 2 2]
[0 2 1]
[0 1 2]
```

A 的特征多项式：`p(x) = x^3 - 5*x^2 + 7*x - 3`

```
Eigenvalues of A = [3, 1, 1]
```

特征值、特征向量和几何重数：
`[(3, [(1, 1/2, 1/2)], 1), (1, [(1, 0, 0),(0, 1, -1)], 2)]`

提取特征值：lambda
`1 = 3`lambda`2 = 1`
lambda`3 = 1`

对角矩阵 D =
[3 0 0]
[0 1 0]
[0 0 1]

提取特征向量:
p1 = (2, 1, 1) p
2 = (1, 0, 0) p3 =
(0, 1, -1)

可逆矩阵 P =
[2 1 0]
[1 0 1]
[1 0-1]

P 的逆矩阵, P^(-1) =
[0 1/2 1/2]
[1 -1 -1]
[0 1/2-1/2]

计算 PDP^(-1) =
[1 2 2][0 2 1][0 1 2]
= A

最后一步中，我们通过验证乘积恢复为原矩阵*A*，确认对角化是正确的。

3.2 使用 SageMath 进行正交对角化

以下命令展示了针对第 2.2 小节中所考察矩阵的一个正交对角化示例的 SageMath 代码。

A=矩阵([[3,-4,-4], [-4,3,-4], [-4,-4,3]]) print("矩阵 A = ") print(A)
print() print("A 的特征多项式：p(x) =",A.charpoly()) print() print("
A 的特征值 =", A.eigenvalues()) print() lambda1 =A.eigenvalues()
[0]lambda2 =A.eigenvalues()[1]lambda3 =A.eigenvalues()[2]
print("提取特征值：") print("lambda1 =", lambda1) print("l
ambda2 =", lambda2) print("lambda3 =", lambda3) print() D=
diagonal_矩阵(A.eigenvalues()) print("对角矩阵 D = ") print(D)

```
print() print("特征值、特征向量和几何重数：", ) print(A.eigenvectors_right()) print() q1=A.eigenvectors_right()[0][1][0]q2=A.eigenvectors_right()[1][1][0]q3=A.eigenvectors_right()[1][1][1]print("提取特征向量：") print("q1 =", q1) print("q2 =", q2) print("q3 =", q3) print() print("检查正交性：") print("<q1,q2> =", q1.inner_product(q2)) print("<q1,q3> =", q1.inner_product(q3)) print("<q2,q3> =", q2.inner_product(q3)) print() print("q2 和 q3 不正交；应用 Gram-Schmidt过程：") print() q31 = q3 - q2.inner_product(q3)/q2.inner_product(q2)*q2 print("q3’ =", q31) print() print("检查现在 q2 和 q3’ 是否正交： <q2,q3’> =", q2.inner_product(q31)) print() print("归一化所有正交特征向量：") print() p1 = q1/sqrt(q1.inner_product(q1)) print("p1 =", p1) print() p2 =q2/sqrt(q2.inner_product(q2)) print("p2 =", p2) print() q32 = 2*q31 p3 =-q32/sqrt(q32.inner_product(q32)) print("p3 =", p3) print() print("构造一个可逆且正交的矩阵 P：") P =矩阵([p1,p2,p3]).transpose() print("P = ") print(P) print() print("P 的逆矩阵，P^(-1) = ") print(P.inverse()) print() print("计算 PDP^(-1) =") print(P*D*P.inverse()) print("= A") print() print("P 的转置，P^T = ") print(P.transpose()) print()
```

```
print("计算 PDP^T = ")
print(P*D*P.transpose())
print("= A")
```

我们获得的输出如下:

```
矩阵 A =
[ 3 -4-4I-4
3-4I-4 -4  3]
```

A 的特征多项式: p(x) = x^3 - 9*x^2 - 21*x + 245

A 的特征值 = [-5, 7, 7]

提取特征值: lambda1 =-
5 lambda2 = 7lambda
3 = 7

对角矩阵 D =

```
[-5  0  0]
[ 0  7  0]
[ 0  0  7]
```

特征值、特征向量和几何重数: [(-5, [(1, 1, 1)], 1), (7, [(1, 0, -
1),(0, 1, -1)], 2)]

提取特征向量: q1 = (1,
1, 1) q2 = (1, 0, -1) q3 =
(0, 1, -1)

检查正交性:

```
<q1,q2> = 0<
q1,q3> = 0<
q2,q3> = 1
```

q2 和 q3 不正交; 应用 Gram-Schmidt过程:

```
q3' = (-1/2, 1, -1/2)
```

检查现在 q2 和 q3' 是否正交: <q2,q3'> = 0

将所有正交特征向量归一化:

p1 = (1/3*sqrt(3), 1/3*sqrt(3), 1/3*sqrt(3))

p2 =(1/2*sqrt(2), 0, -1/2*sqrt(2))

p3 =(1/6*sqrt(6), -1/3*sqrt(6), 1/6*sqrt(6))

构造一个可逆且正交的矩阵 P:

P =
[1/3*sqrt(3) 1/2*sqrt(2) 1/6*sqrt(6)I 1/3*sqrt(3) 0 -
1/3*sqrt(6)I 1/3*sqrt(3) -1/2*sqrt(2) 1/6*sqrt(6)]

P 的逆矩阵，P^(-1) =
[1/3*sqrt(3) 1/3*sqrt(3) 1/3*sqrt(3)I 1/2*sqrt(2) 0 -
1/2*sqrt(2)I 1/6*sqrt(6) -1/3*sqrt(6) 1/6*sqrt(6)]

计算 PDP^(-1) =[3 -
4-4I-4 3-4I-4 -4 3]=
A

P 的转置，P^T =
[1/3*sqrt(3) 1/3*sqrt(3) 1/3*sqrt(3)I 1/2*sqrt(2) 0 -
1/2*sqrt(2)I 1/6*sqrt(6) -1/3*sqrt(6) 1/6*sqrt(6)]

计算 PDP^T =[3 -
4-4I-4 3-4I-4 -
4 3]= A

由于 $P^{-1}=P^{T}$，因此 PDP^{-1}和 PDP_{T}的计算都归约为原矩阵 A并不令人意外。然而，在本例中，尽管我们可以直接构造对角矩阵D，但仍应应用 Gram-Schmidt过程来构造正交且可逆的矩阵 P。

3.3 使用 SageMath 进行奇异值分解

以下给出了用于构造第 2.3 小节中所讨论矩阵的奇异值分解的 SageMath 命令。

M =矩阵([[2,-2], [-3,-4], [-4,-3]]) print("矩阵M = ") print(M) print() A =M.转置()*M print("矩阵M^T M = ") print(A) print() print("M^T M =的特征值", A.eigenvalues()) print() print("特征值、特征向量和几何重数:",) print(A.eigenvectors_right()) print() sigma1 = sqrt(A.eigenvalues()[0]) sigma2 = sqrt(A.eigenvalues()[1]) print("M的奇异值： ") print("sigma1 =", sigma1)

print("sigma2 =", sigma2) print() Sigma = 矩阵([[sigma1, 0], [0,sigma2], [0,0]]) print("矩阵 Sigma =") print(Sigma) print() w1 =A.eigenvectors_right()[0][1][0]v1 =w1/w1.norm() w2 = A.eigenvectors_right()[1][1][0]v2 = w2/w2.norm() print("矩阵 M^T M 的特征向量：") print("v1 =", v1) print("v2 = ", v2) print() print("矩阵 V =") V = matrix([v1, v2]) print(V) print() u1 = 1/sigma1*M*v1 u2 = 1/ sigma2*M*v2 u3 =u1.cross_product(u2) print("M 的左奇异向量：") print("u1 =", u1) print("u2 =", u2) print("u3 =", u3) print() print("检查正交性") print("u1.u2 =", u1.dot_product(u2)) print("u1.u3 =", u1.dot_product(u3)) print("u2.u3 =", u3.dot_product(u2)) print() UT =matrix([u1, u2, u3]) U =UT.transpose() print("矩阵 U =") print(U) print() print("M 的奇异值分解 = U*Sigma*V^T =") print(U*Sigma*V.transpose()) print("= M") 输出如下。 矩阵 M =[2-2][-3 -4][-4 -3]矩阵 M^T M =[29 20][20 29]M^T M 的特征值 = [49, 9]

特征值、特征向量和几何重数：[(49，[(1, 1)]，1)、(9，[(1, -1)
]，1)]

M 的奇异值：sigma
1 = 7sigma2 = 3

矩阵 Sigma =
[7 0]
[0 3]
[0 0]

M~T M 的特征向量：
v1 =(1/2*sqrt(2), 1/2*sqrt(2)) v2 =
(1/2*sqrt(2), -1/2*sqrt(2))

矩阵 V =
[1/2*sqrt(2) 1/2*sqrt(2)I 1/
2*sqrt(2) -1/2*sqrt(2)]

M 的左奇异向量：u1 = (0,-1/2*sqrt(2), -
1/2*sqrt(2)) u2 = (2/3*sqrt(2), 1/6*sqrt(2), -
1/6*sqrt(2)) u3 =(1/3, -2/3, 2/3)

检查正交性
u1.u2 = 0
u1.u3 = 0
u2.u3 = 0

矩阵 U =
[0 2/3*sqrt(2) 1/3I-1/2*sqrt(2)
1/6*sqrt(2) -2/3I-1/2*sqrt(2)-1/6*sqrt(2) 2/3]

M 的奇异值分解 =
U*Sigma*V~T =[2-2I-3 -
4I-4 -3]= M

我们确认所构造的奇异值分解是正确的，因为各矩阵相乘后会还原为原矩阵 M，即 $M = U\Sigma V^T$。在此例中，向量 $\mathbf{u}_1$ 和 $\mathbf{u}_2$ 均通过涉及奇异值的公式计算得出，也就是说，

$$\mathbf{u}_i = \frac{1}{\sigma_i} M\mathbf{v}_i, \qquad \text{for} \quad i = 1, 2.$$

然而，由于第三个奇异值 σ_3 不存在，我们无法使用这个公式，而是通过点积或叉积来计算 $\mathbf{u}_3$，以确保其与 $\mathbf{u}_1$ 和 $\mathbf{u}_2$正交。本例中我们采用了后者。

我们发现，有些学生不知道如何求出第三个向量 $\mathbf{u}_3$。很多时候，他们干脆完全忽略并放弃它，并使用矩阵 U 中的前两个向量构造分解

而不是使用 $\mathbf{u}_1$ 和$\mathbf{u}_2$。在这种情况下，他们混淆了“约化奇异值分解”和“（完整的、非约化的）奇异值分解”。前者取决于 M 的秩，即 r；它不使用 Σ，而是采用对角矩阵 D 。因此，本小节所考虑示例中 M 的约化奇异值分解为

$$\underbrace{\begin{bmatrix} 2 & -2 \\ -3 & -4 \\ -4 & -3 \end{bmatrix}}_{M} = \underbrace{\begin{bmatrix} 0 & 2\sqrt{2}/3 \\ -1/\sqrt{2} & \sqrt{2}/6 \\ -1/\sqrt{2} & -\sqrt{2}/6 \end{bmatrix}}_{U_2} \underbrace{\begin{bmatrix} 7 & 0 \\ 0 & 3 \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}}_{V_2^T}.$$

这种约化奇异值分解有助于计算矩阵 M 的伪逆 M^+（也称为摩尔－彭若斯逆）。对于本小节所考虑的特定矩阵 M ，其伪逆为

$$\underbrace{\frac{1}{63}\begin{bmatrix} 14 & -1 & -8 \\ -14 & -8 & -1 \end{bmatrix}}_{M^+} = \underbrace{\begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}}_{V_2} \underbrace{\begin{bmatrix} 1/7 & 0 \\ 0 & 1/3 \end{bmatrix}}_{D^{-1}} \underbrace{\begin{bmatrix} 0 & -1/\sqrt{2} & -1/\sqrt{2} \\ 2\sqrt{2}/3 & \sqrt{2}/6 & -\sqrt{2}/6 \end{bmatrix}}_{U_2^T}.$$

SageMath 当然可以使用命令`print(M.pseudoinverse())`在几秒钟内计算矩阵的伪逆。不过，应谨慎使用这一步骤，仅将其用于检查最终计算结果，而不是替代用于求得伪逆的适当算法。

4 ChatGPT 中的（正交）对角化与奇异值分解

我们使用d ChatGPT 在本研究中生成的任何回答均采用 2023年3月14日的版本 y.

4.1 ChatGPT 中的矩阵对角化

在没有指定矩阵大小的情况下，要求聊天机器人给出一个矩阵对角化的例子时，ChatGPT 选择了一个简单的 2×2 矩阵来作答，如下例所示。我们问：“请给出一个矩阵对角化的例子。”ChatGPT 回答：

当然，下面是一个矩阵对角化的例子：假设我们有以下矩阵A：

$$\begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix}.$$

我们要对A进行对角化，也就是求一个对角矩阵D 和一个可逆矩阵P，使得$A = PDP^{-1}$，其中D 的对角线元素是A 的特征值。要对A进行对角化，我们首先求A的特征值。A的特征多项式为：

$$\begin{aligned} \det(A-\lambda I) &= \det\begin{pmatrix} 3-\lambda & 1 \\ 0 & 2-\lambda \end{pmatrix} \\ &= (3-\lambda)(2-\lambda) = 6-5\lambda+\lambda^2. \end{aligned}$$

解特征多项式的根，得到 $\lambda = 3$ 和 $\lambda = 2$。它们是A的特征值。

接下来，我们求出与每个特征值对应的特征向量。对于 $\lambda = 3$，我们解方程 $(A-3I)\mathbf{x} = \mathbf{0}$：

$$\begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

由此得到方程 $x_2 = 0$，而 x_1 可以是任意非零值。因此，与 $\lambda = 3$ 对应的特征向量是 $[1 \quad 0]^T$。

对于 $\lambda = 2$，我们求解方程 $(A - 2I)\mathbf{x} = \mathbf{0}$：

$$\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

这得到方程 $x_1 + x_2 = 0$，或 $x_1 = -x_2$。因此，对应于 $\lambda = 2$ 的一个特征向量是 $[-1 \quad 1]^T$。

然后，我们可以将这些特征向量归一化，得到：

$$\mathbf{v}_1 = \frac{[1 \quad 0]^T}{\sqrt{1^2 + 0^2}} = [1 \quad 0]^T,$$
$$\mathbf{v}_2 = \frac{[-1 \quad 1]^T}{\sqrt{1^2 + 1^2}} = \left[-1/\sqrt{2} \quad 1/\sqrt{2}\right]^T.$$

We 随后，可以将特征向量作为列堆叠起来，形成矩阵 P ：

$$P = \begin{bmatrix} 1 & -1/\sqrt{2} \\ 0 & 1/\sqrt{2} \end{bmatrix}.$$

我们还可以将特征值置于对角线上，形成对角矩阵 D ：

$$D = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}.$$

最后，我们可以验证 $A = PDP^{-1}$：

$$PDP^{-1} = \begin{bmatrix} 1 & -1/\sqrt{2} \\ 0 & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 0 & 2 \end{bmatrix},$$

这等于 A。因此，我们成功地将 A 对角化了。

从这个例子可以看出，ChatGPT 给出了一个很好的对角化示例，并附有循序渐进的讲解，有助于读者更好地理解正确的算法。虽然没有展示求出 p^{-1} 的计算过程，但机器假定读者对此已有一定的掌握，尤其是对于一个计算相对简单、容易处理的 2 阶矩阵。正如下一小节中的一些例子所示，在对较大阶的矩阵（如 3×3）进行（正交）对角化时，ChatGPT 遇到了一些困难。

4.2 ChatGPT 中的正交对角化

当我们要求 ChatGPT 给出一个 3×3 对称矩阵的对角化示例时，它给出了以下回答：

好，我们来考虑这个对称矩阵：

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 6 \\ 3 & 6 & 9 \end{bmatrix}.$$

要对角化这个矩阵，我们需要找到一个矩阵 P 和一个对角矩阵 D ，使得：

$$A = PDP^T,$$

其中 P^T 表示 P 的转置。首先，我们需要通过求解特征方程来找出 A 的特征值：

$$\det(A - \lambda I) = 0,$$

其中 λ 是特征值，I 是单位矩阵。由此可得：

$$\det\begin{bmatrix}1-\lambda & 2 & 3\\ 2 & 5-\lambda & 6\\ 3 & 6 & 9-\lambda\end{bmatrix}=0$$
$$=(1-\lambda)[(5-\lambda)(9-\lambda)-6*6]-2[2*(9-\lambda)-6*3]+3[2*6-5*3]$$
$$=(1-\lambda)\left(\lambda^2-15\lambda+36\right)+2(3\lambda-12)-3(6-15)$$
$$=\lambda^3-15\lambda^2+54\lambda-36$$

我们注意到，ChatGPT 从第二个表达式开始就出现了计算错误：它把 $5-\lambda$ 写成了 5。后续各项，包括最后给出的特征方程表达式，无疑都是错误的。错误部分以红色标出。正确的特征方程是 $\lambda^3-15\lambda^2+10\lambda=0$。接下来，我们继续看 ChatGPT 在这一步计算之后给出的内容。

这个多项式在 $\lambda=0$ 处有一个重根，在 $\lambda=3$ 和 $\lambda=12$ 处有不同根。因此，A 的特征值为 $\lambda_1=0$、$\lambda_2=3$ 和 $\lambda_3=12$。接下来，我们需要求出 A 对应每个特征值的特征向量。对于 $\lambda_1=0$，我们有：

$$A-\lambda_1 I=A=\begin{bmatrix}1 & 2 & 3\\ 2 & 5 & 6\\ 3 & 6 & 9\end{bmatrix}.$$

将这个矩阵化为行阶梯形后，我们得到：

$$\begin{bmatrix}1 & 2 & 3\\ 0 & 1 & 1\\ 0 & 0 & 0\end{bmatrix},$$

这给出了特征向量 $v_1=[-2,\ 1,\ 0]$。对于 $\lambda_2=3$，我们有 ：

$$A-\lambda_2 I=\begin{bmatrix}-2 & 2 & 3\\ 2 & 2 & 6\\ 3 & 6 & 6\end{bmatrix}.$$

将这个矩阵化为行阶梯形后，我们得到：

$$\begin{bmatrix}1 & 0 & 1\\ 0 & 1 & 2\\ 0 & 0 & 0\end{bmatrix},$$

w这给出了特征向量 $v_2=[-1,\ -2,\ 1]$。对于 $\lambda_3=12$，我们有 ：

$$A-\lambda_3 I=\begin{bmatrix}-11 & 2 & 3\\ 2 & -7 & 6\\ 3 & 6 & -3\end{bmatrix}.$$

将这个矩阵化为行阶梯形，我们得到：

$$\begin{bmatrix}1 & 0 & -1\\ 0 & 1 & -2\\ 0 & 0 & 0\end{bmatrix},$$

由此得到特征向量 $v_3=[1,\ 2,\ 1]$。

ChatGPT 在此阶段停止了，甚至没有完成第三个特征向量的最后一个分量（如红色所示）。在其他情况下，ChatGPT 可以继续构造对角矩阵 D 和可逆矩阵 P，并以“最后，我们可以验证$A = PDP^{-1}$。”结束答案，但本例中并未如此。

然而，有趣的是，ChatGPT 得到了一个正确的特征值，即 $\lambda_1 = 0$，尽管行阶梯形矩阵中的一个元素不正确，即$a_{23} = 0$ 而不是 1。相应的正确特征向量为$\mathbf{v}_1 = (3,\ 0,\ -1)$。此外，与 ChatGPT 得到的其他正整数特征值不同，矩阵A在本例中的 λ_2 和 λ_3 不是整数，而是无理数，尽管二者仍然为正数。按升序排列，它们为

$$\lambda_2 = \frac{1}{2}\left(15 - \sqrt{185}\right), \qquad \text{and} \qquad \lambda_3 = \frac{1}{2}\left(15 + \sqrt{185}\right).$$

相应特征向量的一种可能选择为

$$\mathbf{v}_2 = \left(\frac{1}{3},\ \frac{2\left(-15 + \sqrt{185}\right)}{3\left(-13 + \sqrt{185}\right)},\ 1\right), \qquad \text{and} \qquad \mathbf{v}_3 = \left(\frac{1}{3},\ \frac{2\left(15 + \sqrt{185}\right)}{3\left(13 + \sqrt{185}\right)},\ 1\right).$$

如果我们使用 ChatGPT 给出的特征值及其对应的特征向量来构造原矩阵，则所得矩阵既不对称，也不包含全部整数元素。相反，我们得到如下矩阵：

$$PDP^{-1} = \frac{1}{10}\begin{bmatrix} 15 & 30 & 45 \\ 30 & 60 & 90 \\ 9 & 18 & 75 \end{bmatrix} = \begin{bmatrix} \frac{3}{2} & 3 & \frac{9}{2} \\ 3 & 6 & 9 \\ \frac{9}{10} & \frac{9}{5} & \frac{15}{2} \end{bmatrix}.$$

为了进一步测试其能力，我们要求 ChatGPT 对一个特定的 3 阶对称矩阵进行正交对角化。该矩阵就是我们在小节 B 中研究的同一个矩阵 2.2：

$$B = \begin{bmatrix} 3 & -4 & -4 \\ -4 & 3 & -4 \\ -4 & -4 & 3 \end{bmatrix}.$$

遗憾的是，ChatGPT 对这个矩阵对角化问题给出了错误的答案。它对特征多项式的计算是错误的：$\lambda^3 - 9\lambda^2 + 24\lambda$，由此得到了一组错误的特征值：$\lambda_1 = 0$、$\lambda_2 = 3$ 和 $\lambda_3 = 6$。随后继续计算相应的特征向量$\mathbf{v}_1$ 和$\mathbf{v}_2$，但未能完成$\mathbf{v}_3$的计算。计算过程在矩阵方程中出现一些缺失项后突然停止。通过进一步分析每一步，我们发现，针对前两个特征值的增广矩阵行化简过程以及所得的特征向量都不正确。要求 ChatGPT“ 重新生成回答 ” 也没有真正起到帮助，因为它再次给出了错误且不完整的答案。从这些例子中，我们观察到，当前版本的 ChatGPT 似乎还无法胜任超出 2×2 的简单矩阵（正交）对角化。

4.3 ChatGPT 中的 SVD

我们要求 ChatGPT“提供一个奇异值分解的示例”。有趣的是，ChatGPT 提供了一个相对高级的 3×3 矩阵示例，并在 Python 中使用了矩阵代数库 NumPy，SageMath 也将其作为主要的语言求解器使用。所选矩阵为

$$A = \begin{bmatrix} 4 & 11 & 14 \\ -1 & 1 & 17 \\ 0 & 12 & 5 \end{bmatrix}.$$

在前两种情况下，ChatGPT 解释了执行矩阵（正交）对角化的算法。而在这种情况下，它直接提供了 Python 代码及其相应的输出。此外，

尽管原矩阵包含整数元素，但其奇异值分解中的矩阵由实数组成，手动检查可能较为困难。

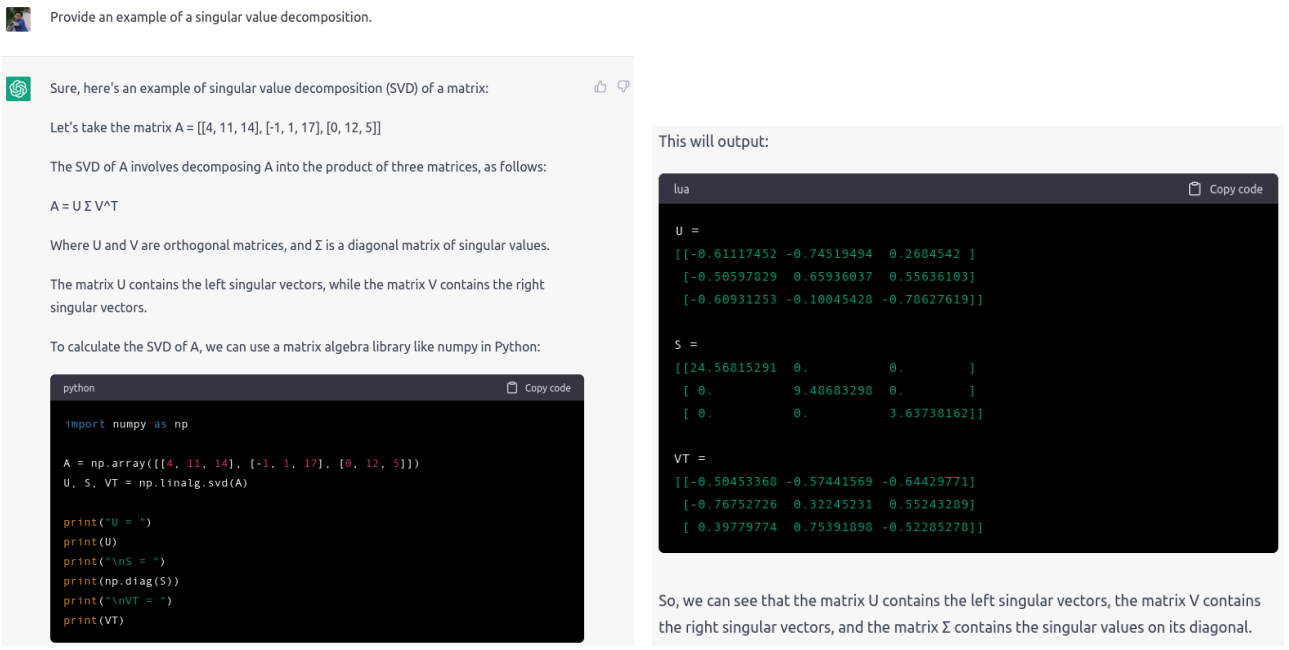

图1：ChatGPT 对“提供一个奇异值分解示例”这一询问的输出

遗憾的是，正如图1所示，ChatGPT 给出的示例本身就是错误的。它给出的奇异值为 $\sigma_1 = 24.57$、$\sigma_2 = 9.49$ 和 $\sigma_3 = 3.63$，而正确的奇异值为 $\sigma_1 = 25.58$、$\sigma_2 = 11.36$ 和 $\sigma_3 = 3.13$。虽然第一个奇异值的误差小于 4%，但第二个和第三个奇异值的误差约为 16%，这太大了，不能接受。使用 SageMath 命令 `A=matrix(RDF,[[4,11,14],[-1,1,17],[0,12,5]])` 和 `A.SVD()`，我们得到以下输出

([-0.7020788202506024 0.21713691817311598-0.6781864706124737I -0.5829773893610252 -0.7221727564849129 0.3722954113130941I -0.4089287146304667 0.6567481012970376 0.6336081105805151],

[25.575254913906992 0.0 0.0I 0.0 11.363233131054077 0.0I 0.0 0.03.1278217497152005],

[-0.08701136700816689 0.13998836518016763 -0.9863226042353253I -0.5166317611363659 0.8401931429966347 0.16482446980522636I -0.8517749969308255 -0.5239071865088518 0.0007839318281394777]),

其中，第一个、第二个和第三个矩阵分别对应 U、Σ 和 V^T。这里，RDF 指实双精度域，即一种使用双精度浮点数近似实数的域。

我们进一步要求 ChatGPT“给出一个奇异值分解示例，其中矩阵的奇异值为整数”。ChatGPT 用相同的措辞作答，但选用了另一个矩阵，这次它是一个4×2 矩阵，如下所示：

$$A = \begin{bmatrix} 2 & 2 \\ 1 & 3 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

然而，结果仍然不正确，而且回答几乎到结尾时便中断，并显示“正文流错误”，同时提示“重新生成回答”。见图 2。多次要求重新生成回答也没有得到任何结果。对于给定的矩阵，其正确的奇异值并非整数，不过可以用解析式表示，即 $\sigma_{1,2} = \sqrt{9 \pm \sqrt{65}} \approx 4$，分别约为 13 和 0.97。

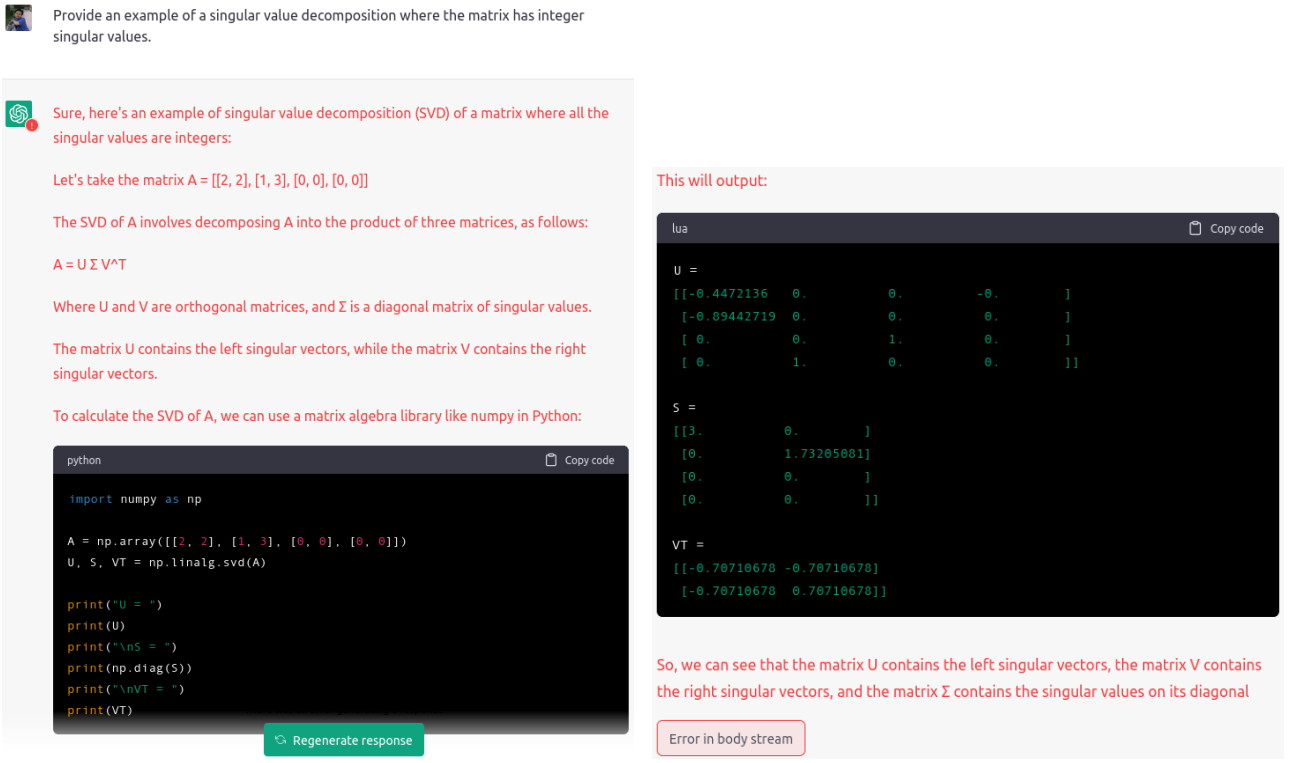

图2：ChatGPT 对“请给出一个矩阵具有整数奇异值的奇异值分解示例”这一问题的回答。

使用一个新的聊天机器人并提出相同的问题后，我们得到了更好的结果，尽管回答中仍有一些概念性错误。参见图3。我们选择了另一个 3×3 矩阵，但这次它是对称的：

$$A = \begin{bmatrix} 3 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 3 \end{bmatrix}.$$

ChatGPT 没有计算奇异值，而是计算了矩阵A的特征值。巧合的是，在这个特定示例中，两者相同，即 $\lambda_1 = 5 = \sigma_1, \lambda_2 = \lambda_3 = 2 = \sigma_2 = \sigma_3$。这种情况相当罕见，因为只有正定对称矩阵（或复值情况下的厄米矩阵）才会出现。尽管矩阵U 和V 看起来正确，但矩阵 Σ 并不正确，因为 ChatGPT 显示的是特征值的平方根，而没有正确计算奇异值。

5 讨论与结论

本研究存在一些局限性。首先，我们仅关注（正交）对角化和奇异值分解，而矩阵分解中的算法思维涵盖更广泛的主题。例如，我们没有考虑与求解线性方程组相关的分解，如 LU分解、QR分解和乔列斯基分解等。其他基于特征值的分解也与我们的研究相关，但本文未涵盖的还包括 Jordan 分解、Schur 分解、Takagi 分解和 QZ分解等。其次，为了更好地理解矩阵分解，我们特别选用了 SageMath

而不是其他计算机代数系统。主要原因在于，它是一个免费且开源的计算机代数系统，并且正如其网站所述，其使命是“创建一个可行的免费开源替代方案，以替代 Magma、Maple、Mathematica 和 Matlab”[94]。此外，只要具备互联网连接，我们就可以直接在服务器上执行计算，而无需费心在自己的计算机上安装该程序。

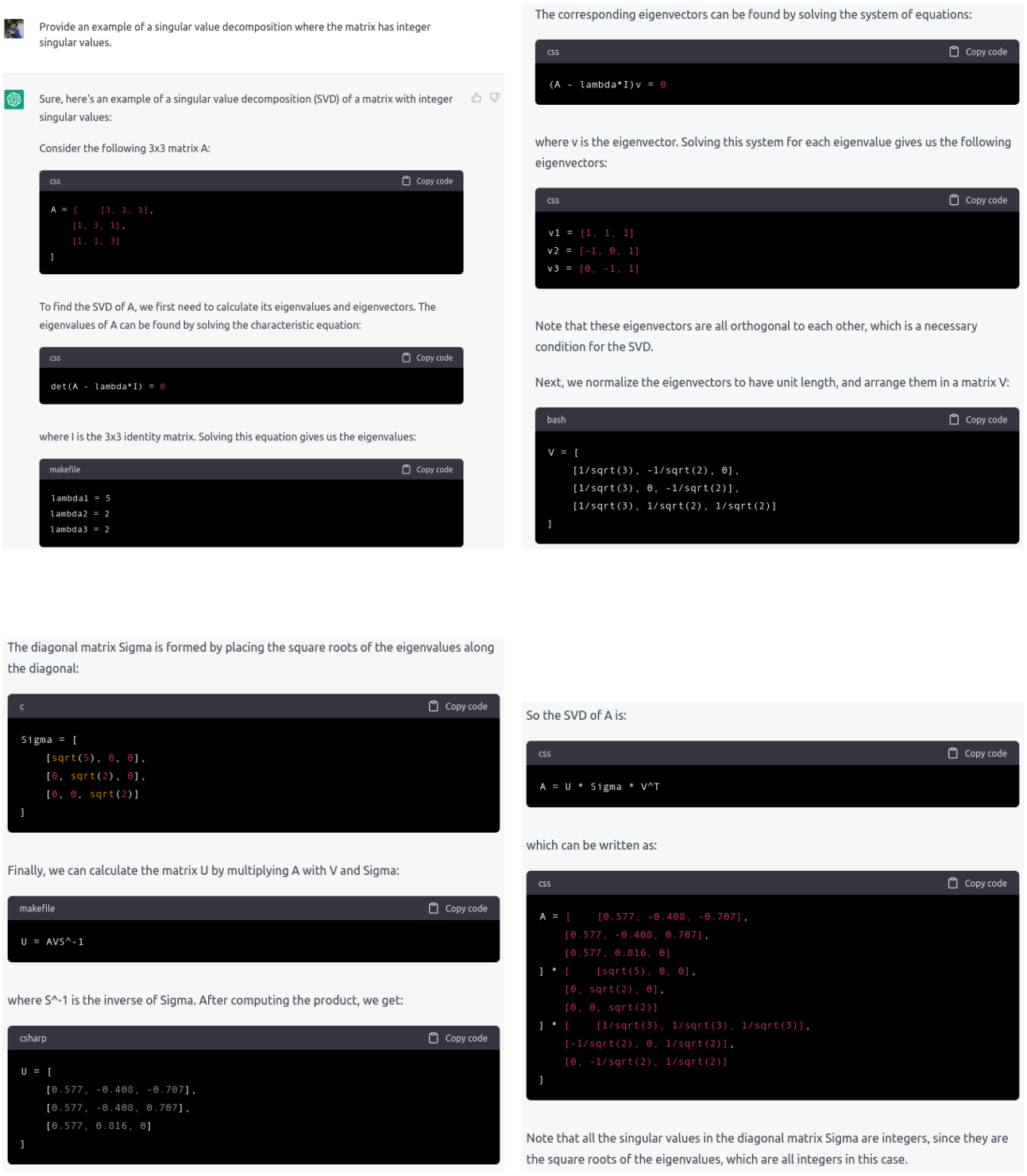

图3：ChatGPT 对与图中相同问题的回答2。阅读顺序从左上方开始，接着是右上方，然后是左下方，最后是右下方。

第三，令人失望的是，ChatGPT 给出的结果包含许多错误。一方面，这可能会让许多学业基础薄弱、只会接受聊天机器人提供的内容的学生感到困惑。另一方面，这也为进一步探讨如何培养学习者的批判性思维提供了绝佳机会。此外，由于我们的研究是在 ChatGPT 开发的初期阶段开展的，本讨论中考察的回答在未来可能不再具有参考价值，因此其他研究人员可能难以复制类似的研究。不过，我们知道未来会推出更新版本，回答也会相应改进。（事实上，在撰写本文时，ChatGPT Plus 订阅用户已经可以使用更先进的聊天机器人版本，即 GPT-4 模型。）因此，我们可以确信，关于矩阵分解或其他数学问题的错误答案最终都会逐渐减少 [90]。

对于许多数学教育者和线性代数教师而言，有些人会发现，学习这门学科似乎顺畅而轻松，而另一些人则似乎会迷失方向，无法将一个概念与另一个概念联系起来。这一特点使得向本科生教授线性代数颇具挑战性。为了探究**第一个研究问题**，我们指出了**学生在特定线性代数主题中遇到的一些常见错误和困难，即（正交）对角化和奇异值分解**。事实上，这些主题对许多学习者而言可能颇具挑战性，因为它们要求学习者具备更基础的线性代数知识，例如进行初等行变换和求正交投影。同时，如果不理解矩阵对角化，在后续学习中也会遇到更多困难，例如学习正交对角化、奇异值分解，以及其他与对角化相关的主题。这些主题通常不会在标准的线性代数课程中讲授，但在数据科学或机器学习中有一些应用。

线性代数的学习者应当认识到，并非所有方阵都能对角化。误以为每个矩阵都能对角化，可能会导致后续学习中产生困惑。要掌握解决对角化问题所需的计算思维，除了理解对特定矩阵进行对角化的算法步骤，学生还应对特征值及其对应的特征向量等较为抽象的对象形成直观认识。这也意味着，他们应当知道如何求矩阵的特征方程、计算特征值，并通过初等行变换求出对应的特征向量。难以进行行化简的学习者，通常也难以进一步掌握如何简洁地表示特征向量。当然，若学习者在这些更基础的矩阵代数内容上遇到困难，就应当充分练习，直到熟练掌握计算过程。

学生学习对称矩阵的正交对角化时，还会遇到另一种困难。尽管其步骤与先前学过的矩阵对角化相似，但有些学生常常忘记将求得的特征向量归一化，并检查这些特征向量彼此是否标准正交。若不确定这一点，只是简单地构造 $P^{-1} = P^T$，就无法得到正确的正交对角化；也就是说，PDP^T 无法还原为原矩阵 A。当需要构造一组正交特征向量时，还会出现进一步的困难：如果这些特征向量来自同一个特征子空间，它们可能并不正交。要将向量投影和 Gram-Schmidt 正交化过程这一概念联系起来，除了熟练计算正交投影本身，还需要扎实理解相关概念。掌握这些计算思维技能后，任何线性代数学习者都能够解决（正交）对角化问题。

学习奇异值分解时，一个常见的错误是混淆奇异值和特征值。即使学习者知道奇异值的定义是特征值的平方根，他们中的许多人仍常常忘记，应该计算对称矩阵$M^T M$的特征值，而不是原矩阵M的特征值。算法早期步骤中的一个小错误，就会导致无法得到正确的奇异值分解，尽管他们理解如何计算矩阵的奇异值分解，也能正确执行这一算法。另一个难点是构造矩阵 Σ，尤其是在M 既不是方阵也不满秩的情况下。事实上，我们必须正确确定应在对角矩阵D的对角线上嵌入多少个零块，才能构造出 Σ。一个容易记住的规则是，Σ 必须与M大小相同。如果M 的行数多于列数，那么方正交矩阵U 将大于方正交矩阵V，反之亦然。构造U 时的一个难点是，M 秩亏，或$m > n$，其中m 和

n 分别表示行数和列数。当我们尝试在缺少特征向量的情况下写出分解时，得到的是约化奇异值分解，而非完整的非约化形式。扎实掌握内积（点积）和叉积，并且不要忘记将向量归一化，就足以掌握求矩阵奇异值分解所需的算法思维技能。

这引出了对**第二个研究问题**的讨论，即**静态计算机代数系统（如 Sage- Math）如何帮助学习者掌握需要算法思维技能的线性代数主题，尤其是（正交）对角化和奇异值分解?** 使用计算机代数系统辅助计算过程，尤其是线性代数中的计算，对我们许多人而言都大有裨益，其中包括学习者和教师。SageMath 尤其如此：它的功能比图形计算器更强大，而且对于拥有互联网连接的用户来说，通过网页浏览器单元即可使用，因此更为方便。与其他符号计算型计算机代数系统类似，SageMath 可以帮助我们解决许多数学问题、检查用纸笔完成的计算是否正确，并节省大量时间，免去手动完成那些冗长复杂且难以进行或极为耗时的计算。

正如我们在第3节中所见，尽管 SageMath 具有静态特性，它仍能帮助我们求出给定矩阵的（正交）对角化和奇异值分解。计算特征多项式、特征值和特征向量既直接又快速。对于较大的矩阵，这一点尤为有用，因为求出特征值和特征向量可能要花费大量时间。无需为这一步操心，线性代数学习者便可以直接专注于核心任务：为（正交）对角化和奇异值分解构造矩阵。得到所有矩阵后，SageMath 可以通过计算（正交）对角化或奇异值分解中的矩阵乘积，帮助我们迅速判断这些矩阵是否正确。任何正确的分解（即使并非唯一）都将还原出原矩阵。SageMath 的静态特性也要求学习者仔细查看输出。例如，命令 `A.eigenvectors_right()` 按特征值、特征向量及其几何重数的顺序给出输出。构造对角矩阵D 和可逆矩阵P 也需要人工干预，这要求用户发挥一定的主动性。静态的 SageMath 与动态的学习者之间的这种互动，使计算机代数系统成为学习数学的有力工具。此外，SageMath 的交互功能允许用户尝试不同的矩阵，或尝试尺寸、特征各异的矩阵。尽管本研究并未真正着重介绍 SageMath 的可视化工具，但它呈现绘图或图的能力将进一步帮助学习者解释数学中遇到的更难理解的概念。

随着知识不断进步、技术持续发展，在静态计算机代数系统出现的同时，利用人工智能强大能力的动态聊天机器人也应运而生，例如 ChatGPT。尽管 ChatGPT 的主要设计目的并非回答数学问题，但本研究中的一些尝试以及其他研究人员的研究表明，它具备处理数学问题的能力。它的回答得到改进、变得越来越好，只是时间问题。这促使我们探讨**第三个也是最后一个研究问题**，即**尽管 ChatGPT 具有动态特性，我们能否依赖它来理解线性代数，尤其是（正交）对角化和奇异值分解?** 简短的回答是否定的。理由简单明了：ChatGPT 经常给出错误答案。在我们的研究中，除第一个问题外，ChatGPT 对其他所有问题的回答都或多或少存在错误。如果确实如此，那么在学习数学或理解线性代数时，我们是否应该直接放弃使用 ChatGPT 及其他类似的人工智能生成的聊天机器人，甚至彻底禁止学生在学习期间使用它们? 我们该如何在利用 ChatGPT 等人工智能生成的聊天机器人辅助数学学习和线性代数理解的同时，确保学生深入、全面地掌握相关内容，而不是完全依赖这项技术?

尽管许多人不会认同这一观点，但我们对于放弃或禁止使用 Chat- GPT 仍有所保留；这不仅适用于数学学习，也适用于其他学科。无论我们是否喜欢，这辆“技术列车”都已驶出站场，此后一路加速，抵达地球的各个角落，人人都能访问互联网。它的动态特性吸引了许多研究人员。而这仅仅是个开始。自去年年底发布以来，它已颠覆了许多行业，而学术界

和教育也不例外。因此，与其放弃或禁止使用它，我们不如拥抱它，并将其融入教学与学习之中。计算器或计算机代数系统进入传统的粉笔与黑板教室时，也有类似的情况。起初，人们对将技术融入数学教学与学习有所抵触；如今，这已成为惯例，甚至鼓励有志成为教育工作者的人这样做 [95——98]。毕竟，这是弥合数字原住民与数字移民之间鸿沟的一种方式 [99——102]。

ChatGPT 开发的这一早期阶段为培养学习者的算法思维技能提供了巨大的机会，尤其是在输出包含许多错误的情况下。当被询问时，ChatGPT 并不是一次性给出答案，而是逐字母地输出答案，实际上就像有人在电脑另一端逐字输入答案一样。这一动态特性使 ChatGPT 优于静态计算机代数系统，也吸引了许多新用户。在我们关于矩阵分解的研究中，ChatGPT 展示了如何对给定矩阵进行对角化或求其奇异值分解的正确分步算法。在未指定特定矩阵的情况下，它甚至可以自行选择一个矩阵示例，通常是规模相对较小的矩阵。

正如第 4 节关于正交对角化的讨论所示，ChatGPT 展示了如何计算矩阵的特征多项式，并求解相应的特征方程以找到特征值，尽管其中存在计算错误。这为学习者提供了批判性审查所给答案并找出错误所在的机会。在寻找相关特征向量时出现的其他错误，以及 ChatGPT 突然停止计算而未完成计算、导致结果不完整的情况，也同样适用于这一情形。对于奇异值分解，前两次尝试提供了详细算法似乎被有意回避的示例。相反，ChatGPT 使用 Python 中的矩阵代数库来完成该任务。随后，它给出了分解中涉及的三个矩阵的最终答案，即 U、 Σ 和 V。我们的第三次尝试似乎有所改进：ChatGPT 展示了计算奇异值、寻找特征向量、构造矩阵 V（其各列包含右奇异向量）、构成矩阵 Σ，以及计算矩阵 U（其各列包含左奇异向量）的分步算法。ChatGPT 将特征值与奇异值混淆这一事实，使我们能够从该输出中获得一条有价值的经验，因为正如第 2节所述，在线性代数学习者中也经常发现这一问题。

总之，我们探讨了线性代数中的算法思维技能与学习者在学习（正交）对角化和奇异值分解等特定主题时常犯错误之间的关系，以及静态计算机代数系统 SageMath 和动态人工智能 ChatGPT 等技术工具如何帮助提升对这些主题的算法理解。理解矩阵分解的步骤并准确执行计算过程，需要扎实掌握线性代数中的其他主题，包括但不限于使用初等行变换求解线性方程组、求特征值及其对应的特征向量、正交性与向量投影（Gram-Schmidt过程），以及点积（内积）和叉积。

我们发现，通过更多练习并强化线性代数的基本概念及其相互关系，可以消除学生常犯的一些错误。尽管计算机代数系统 SageMath 本质上是静态的，但它在核验手工计算结果、处理更大尺寸的矩阵、检查所得因式分解是否正确，以及完成理解材料所需的其他计算活动方面，都能提供极大帮助。与 Maple、Matlab 和 Mathematica 等价格昂贵的计算机代数系统相比，SageMath 的主要优势在于免费、开源，并提供基于网页的界面，支持使用单元格和 Python。尽管 ChatGPT 现阶段具有动态且引人注目的特点，可用于辅助学习，但由于其输出错误众多，既不应将其作为主要工具，也不应依赖它。这一发展前景令人振奋，尤其是在教育领域；未来 ChatGPT 的更新版本将得到改进并陆续发布。如何有效利用聊天机器人开展数学教学与学习，同时不损害学术诚信，仍是一个有待解决的问题。对此，我们也期待其他研究人员相应地开展本研究的后续研究。

参考文献

[1] Andrilli, S., & Hecker, D.（2022年）。《初等线性代数》，第六版。美国马萨诸塞州剑桥：Academic Press。[2] Anton, H., & Rorres, C.（2013年）。《初等线性代数：应用版》，第12版。美国纽约：John Wiley & Sons。[3] Axler, S.（2015 年）。《线性代数：正确地做》，第三版。德国柏林海德堡：Springer。[4] Baker, R. & Kuttler, K. L.（2021年）。《应用线性代数》。新加坡：世界科学出版社。[5] Blyth, T. S., & Robertson, E. F.（2002年）。《基础线性代数》。德国柏林：Springer Science & Business Media。[6] Bretscher, O.（2018年）。《应用初等线性代数》，第五版。英国伦敦：培生教育。[7] Boyd, S., & Vandenberghe, L.（2018年）。《应用线性代数导论：向量、矩阵与最小二乘法》。英国剑桥：剑桥大学出版社。[8] Friedberg, S. H., Insel, A. J., & Spence, L. E.（2013年）。《线性代数》，第五版。英国伦敦：培生教育。[9] Hill, R. O.（2014年）。《初等线性代数》。美国马萨诸塞州剑桥：Academic Press。[10] Hoffman, K. & Kunze, R.（2015年）。《线性代数》，第二版。印度：培生教育。[11] Johnson, L., Riess, D., & Arnold, J.（2017年）。《线性代数导论》，第五版。英国伦敦：培生教育。[12] Kolman, B. & Hill, D.（2017年）。《应用初等线性代数》，第九版。英国伦敦：培生教育。[13] Kuttler, K. L.（2021年）。《初等线性代数》。独立出版。[14] Lang, S.（1997年）。《线性代数导论》，第二版。德国柏林海德堡：Springer Science & Business Media。[15] Larson, R.（2016年）。《初等线性代数》，第八版。美国马萨诸塞州波士顿：圣智学习。[16] Lax, P. D.（2007年）。《线性代数及其应用》，第二版。美国纽约：John Wiley & Sons。[17] Lay, D. C., Lay, S. R., & McDonald, J.（2021年）。《线性代数及其应用》，第六版。英国伦敦：培生教育。[18] Mirsky, L.（2013年）。《线性代数导论》。美国纽约州米尼奥拉：多佛出版社。[19] Spence, L., Insel, A., & Friedberg, S.（2017年）。《初等线性代数》，第二版。英国伦敦：培生教育。[20] Strang, G.（2006年）。《线性代数及其应用》，第四版。美国加利福尼亚州贝尔蒙特：Thomson, Brooks/Cole。美国马萨诸塞州波士顿：圣智学习。[21] Barcelos, T. S., Muñoz-Soto, R., Villarroel, R., Merino, E., & Silveira, I. F.（2018年）。通过计算思维活动开展数学学习：系统性文献综述。《通用计算机科学杂志》，24(7)，815–845。[22] Stephens, M., & Kadijevich, D. M.（2020年）。计算/算法思维。载于 Lerman, S.（编者），《数学教育百科全书》，117–123。瑞士沙姆：Springer。[23] Sung, W., Ahn, J., & Black, J. B.（2017年）。将计算思维引入年轻学习者：通过数学教育中的具身化实践计算视角。《技术、知识与学习》，22，443–463。[24] Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U.（2016年）。为数学与科学课堂界定计算思维。《科学教育与技术杂志》，25，127–147。[25] Boyd, S., & Vandenberghe, L.（2004年）。《凸优化》。英国剑桥：剑桥大学出版社。

[26] Trefethen, L. N., & Bau, D. (1997). 数值线性代数。美国宾夕法尼亚州费城：工业与应用数学学会（SIAM）。[27] Bennett, S. R. (2021). 数据科学线性代数：R语言实例。美国加利福尼亚州旧金山：GitHub。可在线访问：https://shainarace.github.io/LinearAlgebra/。检索日期：2023年3月31日。[28] Cohen, M. (2022). 数据科学实用线性代数：使用 Python 从核心概念到应用。美国加利福尼亚州塞巴斯托波尔：O'Reilly Media。[29] Golub, G. H., & Van Loan, C. F. (2013). 矩阵计算。美国马里兰州巴尔的摩查尔斯村：约翰·霍普金斯大学出版社。[30] Klein, P. N. (2013). 矩阵编程：通过计算机科学应用学习线性代数。美国马萨诸塞州牛顿：Newtonian Press。[31] Strang, G. (2019). 线性代数与从数据中学习。美国马萨诸塞州韦尔斯利：Wellesley-Cambridge Press。[32] Aggarwal, C. C. (2020). 机器学习的线性代数与优化：教科书。瑞士沙姆：Springer。[33] Yoshida, R. (2021). 线性代数及其在R语言中的应用。美国佛罗里达州博卡拉顿：CRC Press。[34] Gadanidis, G., Cendros, R., Floyd, L., & Namukasa, I. (2017). 数学教师教育中的计算思维。《技术与教师教育当代问题》，17(4)，458–477。[35] Yadav, A., Stephenson, C., & Hong, H. (2017). 面向教师教育的计算思维。《ACM通讯》，60(4)，55–62。[36] Abdi, H. (2007). 奇异值分解（SVD）与广义奇异值分解（GSVD）。载于 N. J. Salkind（编），《测量与统计百科全书》，第907–912页。美国加利福尼亚州千橡市：Sage出版社。[37] Akritas, A. G., & Malaschonok, G. I. (2004). 奇异值分解（SVD）的应用。《数学与计算机在模拟中的应用》，67(1-2)，15–31。[38] Andrews, H., & Patterson, C. (1976). 奇异值分解与数字图像处理。《IEEE声学、语音与信号处理汇刊》，24(1)，26–53。[39] Biglieri, E., & Yao, K. (1989). 奇异值分解的若干性质及其在数字信号处理中的应用。《信号处理》，18(3)，277–289。[40] Bisgard, J. (2021). 分析与线性代数：奇异值分解及其应用。美国罗德岛州普罗维登斯：美国数学学会。[41] Freire, S. L., & Ulrych, T. J. (1988). 奇异值分解在垂直地震剖面中的应用。《地球物理学》，53(6)，778–785。[42] Henry, E. R., & Hofrichter, J. (1992). 奇异值分解：在实验数据分析中的应用。载于 L. Brand & M. L. Jonhson （编），《基本数值计算机方法》，《酶学方法》第 210 卷，第 129–192 页。美国马萨诸塞州伯灵顿：Academic Press。[43] Klema, V., & Laub, A. (1980). 奇异值分解：计算及若干应用。《IEEE自动控制汇刊》，25(2)，164–176。[44] Lange, K. (2010). 奇异值分解。载于 Lange, K.（作者），《统计学家的数值分析》，第129–142页。《统计与计算》。美国纽约：Springer。[45] Maciejewski, A. A., & Klein, C. A. (1989). 奇异值分解：计算及其在机器人学中的应用。《国际机器人研究杂志》，8(6)，63–79。[46] Mandel, J. (1982). 奇异值分解在回归分析中的应用。《美国统计学家》，36(1)，15–24。[47] Yildiz Ulus, A. (2013). 借助技术讲授线性代数中的"对角化概念"：加拉塔萨雷大学的一项案例研究。《土耳其在线教育技术杂志》，12(1)，119–130。[48] Lazar, Z. (2012). 矩阵奇异值分解教学：一种计算方法。硕士论文。加拿大魁北克省蒙特利尔：康考迪亚大学。

[49] Buchberger, B., Collins, G. E., Loos, R., & Albrecht, R.（编）。（1983）。《计算机代数：符号与代数计算》，第二版。德国柏林海德堡：Springer Science & Business Media。[50] Chudnovsky, V., & Jenks, R. D.（编）（1990）。《数学中的计算机》。美国佛罗里达州博卡拉顿：CRC Press。[51] Cohen, J. S.（2002）。《计算机代数与符号计算：初等算法》。美国佛罗里达州博卡拉顿：CRC Press。[52] Davenport, J. H., Siret, Y., & Tournier, É（1993）。《计算机代数：代数计算系统与算法》，第二版。美国马萨诸塞州剑桥：Academic Press。[53] Fey, J. T.（编）。（2003）。《中学数学教育中的计算机代数系统》。美国弗吉尼亚州雷斯顿：美国全国数学教师理事会/美国全国数学教师理事会。[54] Fuchs, K. J.（2003）。《数学教育中的计算机代数系统：教师培训项目、挑战与新目标》。《数学教学中心期刊》，35，20–23。[55] Geddes, K. O., Czapor, S. R., & Labahn, G.（1992）。《计算机代数算法》。德国柏林海德堡：Springer Science & Business Media。[56] Grabmeier, J., Kaltofen, E., & Weispfenning, V.（编）（2003）。《计算机代数手册：基础、应用与系统》。德国柏林海德堡：Springer。[57] Harper, D., Wooff, C., & Hodgkinson, D.（1991）。《计算机代数系统指南》。美国纽约：John Wiley & Sons。[58] Koepf, W.（2021）。《计算机代数：面向算法的入门》。德国柏林海德堡：Springer Nature。[59] Lamagna, E. A.（2019）。《计算机代数：概念与技术》。美国佛罗里达州博卡拉顿：CRC Press。[60] Simon, G.（1996）。《计算机代数系统间的互操作性》。德国图宾根：Wilhelm-Schickard-Institut für Informatik（WSI）。[61] Soiffer, N. M.（1992）。《计算机代数系统用户界面的设计》。博士论文。美国加利福尼亚州伯克利：加利福尼亚大学。[62] Von Zur Gathen, J., & Gerhard, J.（2013）。《现代计算机代数》，第三版。英国英格兰剑桥：剑桥大学出版社。[63] Wester, M. J.（1999）。《计算机代数系统：实用指南》。美国纽约：John Wiley & Sons。[64] Hannan, Z.（2015）《微积分 I 的 wxMaxima》。《微积分 II 的 wxMaxima》。美国加利福尼亚州费尔菲尔德：Solano Community College。可在线访问 https://wxmaximafor.wordpress.com/。最后访问日期：2023年3月31日。[65] Kanagasabapathy, M.（2018）。《科学计算 wxMaxima 入门》。印度新德里：BPB Publications。[66] Kadry, S., & Awad, P.（2019）。《工程师与科学实验室数学：使用 Maxima》。美国佛罗里达州博卡拉顿：CRC Press。[67] Senese, F.（2019）。《化学家的符号数学：Maxima 用户指南》。美国新泽西州霍博肯：John Wiley & Sons。[68] Timberlake, T. K., & Mixon, J. W.（2016）。《使用 Maxima 学习经典力学》。美国纽约：Springer。[69] Abell, M. L., & Braselton, J. P.（2022）。《Mathematica 实例教程》，第六版。英国英格兰伦敦与美国马萨诸塞州剑桥：Academic Press。[70] Maeder, R.（1991）。《Mathematica 编程》，第二版。美国马萨诸塞州波士顿：Addison-Wesley Longman Publishing。[71] Trott, M.（2007）。《Mathematica 符号计算指南》。美国纽约：Springer Science & Business Media。[72] Wagon, S.（1999）。《Mathematica 实战》，第二版。美国纽约：Springer-Verlag。[73] Wolfram, S.（2003）。《MATHEMATICA® 之书》，第五版。美国伊利诺伊州尚佩恩：Wolfram Media。[74] Abell, M. L., & Braselton, J. P.（2005）。《Maple 实例教程》，第三版。美国马萨诸塞州伯灵顿：爱思唯尔。

[75] Fox, W. P. 与 Bauldry, W.（2019）。《使用 Maple 进行高级问题求解：第一门课程》。美国佛罗里达州博卡拉顿：Chapman and Hall/CRC Press。[76] Fox, W. P. 与 Bauldry, W.（2020）。《使用 Maple 进行高级问题求解：应用数学、运筹学、商业分析与决策分析》。美国佛罗里达州博卡拉顿：Chapman and Hall/CRC Press。[77] Garvan, F.（2001）。《Maple 指南》。美国佛罗里达州博卡拉顿：Chapman and Hall/CRC Press。[78] Heck, A.（2003）。《Maple 入门》，第三版。美国纽约：Springer。[79] Attaway, S.（2013）。《工程师与科学家必备 MATLAB：编程与问题求解实用入门》，第六版。英国英格兰牛津：Butterworth-Heinemann。[80] Davis, T. A.（2010）。《MATLAB 入门指南》，第八版。美国佛罗里达州博卡拉顿：CRC Press。[81] Etter, D. M.（2017）。《MATLAB 入门》，第四版。美国纽约：培生。[82] Higham, D. J. 与 Higham, N. J.（2017）。《MATLAB 指南》，第三版。美国宾夕法尼亚州费城：SIAM。[83] Valentine, D. T. 与 Hahn, B.（2022）。《工程师与科学家必备 MATLAB》，第八版。美国马萨诸塞州剑桥：Academic Press。[84] Bard, G. V.（2015）。《本科生 Sage》。美国罗德岛州普罗维登斯：美国数学学会。[85] Finch, C.（2011）。《Sage 初学者指南》。英国英格兰伯明翰：Packt Publishing。[86] Joyner, D. 与 Stein, W.（2008）。《Sage 教程》。美国加利福尼亚州斯科茨谷：CreateSpace 独立出版平台。[87] Kumar, V.（2022）。《SageMath 基础：数学（实用篇）》。美国华盛顿州西雅图：亚马逊 Kindle 直接出版。[88] Szabó, P. 与 Galanda, J.（2017）。《用于教育与研究的 Sage 数学软件》。载于《2017年第15届新兴电子学习技术与应用国际会议》（ICETA）（第 1–4 页）。美国纽约州曼哈顿：电气电子工程师学会（IEEE）。[89] Zimmermann, P.、Casamayou, A.、Cohen, N.、Connan, G.、Dumont, T.、Fousse, L.、……与 Thomas, H.（2018）。《使用 SageMath 进行计算数学》。美国宾夕法尼亚州费城：SIAM。[90] Frieder, S.、Pinchetti, L.、Griffiths, R. R.、Salvatori, T.、Lukasiewicz, T.、Petersen, P. C.、……与 Berner, J.（2023）。《ChatGPT 的数学能力》。arXiv 预印本 arXiv:2301.13867。[91] Shakarian, P.、Koyyalamudi, A.、Ngu, N. 与 Mareedu, L.（2023）。《对 ChatGPT 解答数学文字题能力的独立评估》（MWP）。arXiv 预印本 arXiv:2302.13814。[92] Azaria, A.（2022）。《ChatGPT 的使用与局限》。hal-03913837。[93] Borji, A.（2023）。《ChatGPT 失败案例分类档案》。arXiv 预印本 arXiv:2302.03494。[94] https://www.sagemath.org/[95] D’Souza, S. M. 与 Wood, L. N.（2003）。《中学生抵制将计算机技术纳入数学学习》。数学与计算机教育，37(3)：284–295。[96] Niess, M. L.（2006）。《特邀社论：培养教师运用技术教授数学》。《技术与教师教育当代问题》，6(2)，195–203。[97] Li, Q.（2007）。《学生与教师对技术的看法：双城记？》。《教育技术研究杂志》，39(4)，377–397。[98] Crompton, H.（2011）。《技术时代的数学：技术在数学课堂中有一席之地》。《教育技术研究中心杂志》，7(1)，54–66。[99] Prensky, M.（2001）。《数字原住民、数字移民：第 1 部分》。《展望》，9(5)，1–6。[100] Prensky, M.（2001）。《数字原住民、数字移民：第 2 部分：他们真的思维不同吗？》。《展望》，9(6)，2–6。[101] Prensky, M.（2009）。《数字智人：从数字移民和数字原住民到数字智慧》。《创新：在线教育杂志》，5(3)，1–9。[102] Prensky, M.（2010）。《教授数字原住民：携手实现真正的学习》。美国加利福尼亚州千橡市：Corwin Press。