

第 03 章 《矩阵计算基础》讲义

3.1 课程内容

全课程由八章组成。本章位于线性代数基础之后、各类矩阵问题的具体算法之前，介绍后续分析会用到的概念和工具：

1. 用浮点模型描述每一步计算的局部误差；
2. 用前向误差和后向误差评价算法；
3. 用条件数区分“问题本身敏感”与“算法放大误差”；
4. 用正交变换构造稳定的矩阵分解。

第 4--8 章的误差分析都以这些结论为基础。

 Tip

本页要点： 分析算法稳定性，要从每一步浮点误差出发，追踪它如何影响最终结果。

3.2 本章主要内容

本章分为四个部分。

- 浮点数表示与舍入误差：IEEE 格式、正规数、次正规数、单位舍入误差以及误差累积。
- 条件数与病态问题：函数求值的敏感性、线性系统条件数、前向与后向误差。
- 迭代精校与混合精度：用较低精度完成主要工作，再用较高精度残量恢复精度。
- 矩阵计算基本工具：Householder、Givens、QR、Hessenberg 化、双对角化和 Lanczos 递推。

后向稳定算法保证计算结果是某个邻近问题的精确解；最终前向误差还取决于问题条件数。这一关系贯穿全章。

 Tip

本页要点： 矩阵计算的核心问题是如何在有限精度下保持结构、精度和可扩展性。

第3.1节 浮点数与浮点误差

3.4 Sources of Error in Simulation

本页从已经确定的数学模型出发，把精确解与计算结果的差别分为三部分：

$$u(x) = U_h(x) + E_{\text{dis}} + E_{\text{alg}} + E_{\text{round}}. \tag{3.1}$$

其中 $u(x)$ 是模型的精确解， $U_h(x)$ 是计算机输出；三项分别来自空间/时间离散、代数问题未精确求解，以及浮点存储和运算中的舍入。把连续问题、精确离散问题、精确算术下的算法输出和实际输出逐级作差，就得到这一分解。模型与物理现象之间的差异不包含在本式中；软件错误也应另行排查。

准确度要与代价一起考虑。通常应让代数误差和舍入误差不显著超过离散误差；如果误差已由离散或舍入主导，再收紧求解器容差也不会增加可信数字。

 Tip

本页要点： 对给定模型，应分别控制离散、代数和舍入误差，避免无助于降低总误差的额外计算。

3.5 Why My Code Gives Different Output

结果可能随运行次数、编译器、编译选项、数学库、硬件和线程调度而改变。并行归约尤其典型：精确算术满足结合律，但浮点加法一般满足

$$fl(fl(a + b) + c) \neq fl(a + fl(b + c)). \quad (3.2)$$

不同线程树对应不同括号结构，所以最后几位可能变化。

可重复性与性能之间存在取舍。严格固定求值顺序有利于逐位复现，却可能限制向量化、融合乘加或并行树形归约；高性能模式允许更多代数重排，通常更快，但不保证位级相同，数值差异是否可接受也需要验证。工程上应根据验证目标选择模式，并记录编译器、选项、库版本和线程数。

 Tip

本页要点： 浮点加法不满足结合律，顺序和并行归约会改变最后几位。

3.6 Wake-Up Calls

以 SPE3 油藏基准中同一网格单元的压力为例。Precise 与 Strict 模式在第 0、1、5、10 年给出相同结果，而 Fast 模式从第 1 年开始偏离，例如第 10 年分别约为

$$2608.065832970213, \quad 2609.016046486710. \quad (3.3)$$

初始时刻相同并不能保证长时间模拟仍相同。每个时间步中的舍入差异会进入下一步状态；若离散动力系统对扰动敏感，这些差异会被逐步放大。单凭这张表不能断言哪一列“正确”。还要检查编译选项是否允许重排、残量是否满足容差、守恒量是否保持，以及误差是否处于离散模型允许的范围內。

 Tip

本页要点： 同一数值模型在不同精度和实现下可能产生不同轨迹，比较结果前要核对计算设置。

3.7 Floating-Point Number Storage

IEEE binary32 使用 32 位：1 位符号、8 位指数和 23 位小数字段。对指数域 $e \in \{1, \dots, 254\}$ 的正规数，数值为

$$x = (-1)^s 2^{e-127} (1.f)_2. \quad (3.4)$$

最高位的 1 不存储，称为 hidden leading bit，因此 23 位小数字段提供 24 位二进制有效数。偏置 127 使无符号指数域能够同时编码正、负指数。

字节序只决定多字节对象在内存中的排列，不改变上述数学值。指数域全 0 和全 1 留作特殊编码，因此不能把所有 256 个指数模式都解释为正规数。

 **Tip**
本页要点： binary32 的位域同时决定数值精度、范围和特殊值的编码方式。

3.8 Normal SP Floating-Point Numbers

指数域的含义如下：

e	f	含义	
0	0	± 0	
0	$f \neq 0$	次正规数	
$1 \leq e \leq 254$	任意	正规数	(3.5)
255	0	$\pm \infty$	
255	$f \neq 0$	NaN	

最小正规正数为

$$2^{-126} \approx 1.1755 \times 10^{-38}, \tag{3.6}$$

最大有限正数为

$$(2 - 2^{-23})2^{127} \approx 3.4028 \times 10^{38}. \tag{3.7}$$

超出最大有限数会溢出为无穷或触发相应异常；接近零时则进入次正规区，而不是立即变成零。

 **Tip**
本页要点： 正规数和特殊编码共同决定 binary32 的可表示范围。

3.9 Subnormal SP Floating-Point Numbers

指数域全 0、尾数非零时不再使用隐藏的首位 1，数值为

$$x = (-1)^s 2^{-126} (0.f)_2. \tag{3.8}$$

指数仍取 -126，而不是按偏置机械地取 -127。这样最大的次正规数

$$(1 - 2^{-23})2^{-126} \tag{3.9}$$

与最小正规数 2^{-126} 之间只差一个最小步长，实现 gradual underflow。最小正次正规数为

$$2^{-23}2^{-126} = 2^{-149} \approx 1.4013 \times 10^{-45}. \tag{3.10}$$

次正规区没有隐藏首位，越靠近零有效位越少。

④ Note

备注：次正规数的运算代价与平台差异。

次正规数（subnormal）没有正规数隐含的首位 1，处理时可能需要额外的移位、指数调整和渐进下溢舍入。部分处理器遇到次正规输入或结果，会转入较慢的辅助处理路径（floating-point assist）；也有硬件能直接高效处理。加载和存储次正规数通常没有特殊代价，差别主要在算术运算。

是否变慢、慢多少，取决于 CPU/GPU、具体指令、数据精度和运行时浮点模式；同一处理器上的加法、乘法、除法和 FMA 也可能表现不同。编译器会影响指令选择，`-ffast-math` 一类选项还可能改变次正规数的处理方式，具体行为须结合工具链和目标平台确认。

x86 的 SSE/AVX 浮点运算中常见两种控制：FTZ（Flush To Zero）把下溢产生的次正规结果置零；DAZ（Denormals Are Zero）把次正规输入按零处理。它们可以避开某些慢路径，但也放弃了相应的渐进下溢行为，可能改变计算结果。因此比较性能时，应同时记录浮点模式并检查精度。

💡 Tip

本页要点： 次正规数填补最小正规数与零之间的间隔，但有效位数逐渐减少。

3.10 Range of FP Representation

表中 t 包含隐藏的首位， $x_{\min}^{(s)}$ 是最小正次正规数， $x_{\min}$ 是最小正规正数， $x_{\max}$ 是最大有限正数。

格式	t	指数位	u	$x_{\min}^{(s)}$	$x_{\min}$	$x_{\max}$
bfloat16	8	8	3.91×10^{-3}	9.18×10^{-41}	1.18×10^{-38}	3.39×10^{38}
fp16	11	5	4.88×10^{-4}	5.96×10^{-8}	6.10×10^{-5}	6.55×10^4
fp32	24	8	5.96×10^{-8}	1.40×10^{-45}	1.18×10^{-38}	3.40×10^{38}
fp64	53	11	1.11×10^{-16}	4.94×10^{-324}	2.22×10^{-308}	1.80×10^{308}
fp128	113	15	9.63×10^{-35}	6.48×10^{-4966}	3.36×10^{-4932}	1.19×10^{4932}

(3.11)

④ Note

备注：为什么采用 bfloat16。

bfloat16 不是 IEEE 754 定义的标准 binary16 格式。它保留与 FP32 相同的 8 位指数，但只有 8 位二进制有效数字：用较低的精度换取接近 FP32 的数值范围，同时将存储和传输量减半。相比之下，IEEE FP16 有 11 位有效数字，却只有 5 位指数，更容易上溢或下溢。

这种取舍适合许多机器学习任务：梯度和激活值可能跨越很大的数量级，而训练往往能容忍较粗的运算精度。矩阵乘法通常采用 BF16 输入、FP32 累加，以控制累加误差。Google TPU、NVIDIA 和 AMD 的多款 GPU，以及 Intel 的部分 Xeon CPU 都支持这种格式；具体舍入和次正规数处理仍需看平台。

低精度计算前常需缩放数据，使输入和中间结果都落在合适范围。小量下溢能否忽略取决于它对结果的影响，不能仅因数值小就一律置零。

💡 Tip

本页要点： 有效位数控制精度，指数位数控制范围，低精度计算必须同时考虑这两种限制。

3.11 Precision of Floating-Point System

单位舍入误差 u 是就近舍入相对误差的统一上界。在无溢出、无严重下溢时，存储和基本运算可分别写成

$$fl(x) = x(1 + \delta), \quad |\delta| \leq u, \quad (3.12)$$

$$fl(x \circ y) = (x \circ y)(1 + \delta), \quad |\delta| \leq u, \quad \circ \in \{+, -, \times, /\}. \quad (3.13)$$

机器精度 ε 是 1 与比它大的下一个浮点数之间的间距。对单、双精度，就近舍入时分别有

$$\begin{aligned} \varepsilon_{\text{single}} &= 2^{-23} \approx 1.1921 \times 10^{-7}, & u_{\text{single}} &= 2^{-24}, \\ \varepsilon_{\text{double}} &= 2^{-52} \approx 2.2204 \times 10^{-16}, & u_{\text{double}} &= 2^{-53}. \end{aligned} \quad (3.14)$$

两者相差因子 2，是“间距”和“就近舍入最大半间距”两种定义造成的。

💡 Tip

本页要点： 单位舍入误差是分析存储和基本运算误差的统一尺度。

3.12 Rounding Errors

设浮点基数为 β ，有效数字位数为 t （包括首位）。采用就近舍入时，单位舍入误差为

$$u = \frac{1}{2}\beta^{1-t}. \quad (3.15)$$

📌 Note

为什么是 $u = \frac{1}{2}\beta^{1-t}$? 正规浮点数写成 $\pm(d_0.d_1 \cdots d_{t-1})_\beta \beta^e$ ，其中 $d_0 \neq 0$ 。最后一位增加 1，数值增加 $\beta^{e-(t-1)}$ ，所以在 $[\beta^e, \beta^{e+1})$ 内，相邻浮点数的间距为 $\Delta = \beta^{e+1-t}$ 。

对满足 $\beta^e \leq |x| < \beta^{e+1}$ 的数，就近舍入的绝对误差至多为半个间距，即 $|fl(x) - x| \leq \Delta/2$ 。再除以 $|x| \geq \beta^e$ ，得到 $\frac{|fl(x) - x|}{|x|} \leq \frac{\beta^{e+1-t}/2}{\beta^e} = \frac{1}{2}\beta^{1-t} = u$ 。指数 e 消去，因此这个界不依赖数值的量级。

这里假设没有溢出，且数值处于正规数范围。 u 是统一的相对误差上界，并非每次舍入的实际误差；次正规数区域不能直接沿用这一相对误差界。

二进制情形 $\beta = 2$ 时，

$$u = 2^{-t}, \quad \varepsilon = 2^{1-t} = 2u. \quad (3.16)$$

文献有时把 u 和 machine epsilon 混用。做一阶误差估计时常数 2 通常不改变结论。

💡 Tip

本页要点： 单位舍入误差由基数和有效位数决定，是后续误差界的基本尺度。

3.13 Examples of FP Arithmetic

浮点加法不满足结合律。取 binary32 的 $u = 2^{-24}$ ，因为 $2^{-25} < u$ ，就近舍入给出

$$fl(1 + 2^{-25}) = 1. \quad (3.17)$$

于是

$$fl(fl(1 + 2^{-25}) - 1) = 0, \quad (3.18)$$

而把同一组三项改成先计算 $1 - 1$ 的求和顺序，则

$$fl(fl(1 - 1) + 2^{-25}) = 2^{-25}. \quad (3.19)$$

再看灾难性相消。取 $\beta = 10$ 、 $t = 4$ ，按“四舍五入”（round half up）存储，1.2345 和 1.2343 分别变成 1.235 和 1.234。需要区分存储误差与最终计算结果的误差：

- **存储时：** 两个数的相对误差分别为 $0.0005/1.2345 \approx 4.05 \times 10^{-4}$ 和 $0.0003/1.2343 \approx 2.43 \times 10^{-4}$ ，均小于 10^{-3} 。
- **计算时：** 存储值之差为 $1.235 - 1.234 = 0.001$ ，而原始数之差为 $1.2345 - 1.2343 = 0.0002$ 。因此最终结果的相对误差为

$$\frac{|10^{-3} - 2 \times 10^{-4}|}{2 \times 10^{-4}} = 4 = 400\%. \quad (3.20)$$

这里的减法 $1.235 - 1.234 = 0.001$ 本身可以精确完成。问题在于两次存储误差一正一负，使差值增加了 0.0005 $- (-0.0003) = 0.0008$ ，是原始差值的 4 倍。**相近数相减时，输入中很小的相对误差，可能变成差值中很大的相对误差。**

前向差分 $f'(x) \approx [f(x+h) - f(x)]/h$ 也有这一风险：当两个函数值接近时，求值误差可能淹没它们的差，随后还要除以很小的 h 。因此步长并非越小越好，具体例子见第 3.14 页。

Tip

本页要点： 输入的存储误差很小，不保证相减后的结果准确；相对误差要相对于小差值衡量。

3.14 Comparison: Double vs Single Precisions

第一个例子计算

$$f(x) = \frac{1 - \cos x}{x^2}. \quad (3.21)$$

当 x 很小时，1 与 $\cos x$ 非常接近，分子发生灾难性相消；单精度会较早得到 0，双精度也会在更小的 x 处失效。利用恒等式

$$1 - \cos x = 2 \sin^2(x/2) \quad (3.22)$$

改写为 $2 \sin^2(x/2)/x^2$ ，可以避免两个近 1 数相减。

第二个例子用前向差分近似 $\cos'(x)$ ：

$$\cos'(x) \approx \frac{\cos(x+h) - \cos x}{h}. \quad (3.23)$$

一般位置的截断误差为 $O(h)$ ，舍入误差经除以 h 后可按 $O(u/h)$ 估计，总误差模型是

$$C_1 h + C_2 \frac{u}{h}. \quad (3.24)$$

令两项平衡得到通常的选择 $h \asymp \sqrt{u}$ 。但本页在 $x = \pi/2$ 处计算，恰好有 $\cos''(\pi/2) = 0$ ，因此

$$\frac{\cos(\pi/2 + h) - \cos(\pi/2)}{h} = -\frac{\sin h}{h} = -1 + \frac{h^2}{6} + O(h^4). \quad (3.25)$$

这里的截断误差实际为 $O(h^2)$ ；若仍采用 $O(u/h)$ 的舍入误差估计，平衡尺度变为 $h \asymp u^{1/3}$ 。无论哪种情形，步长都不是越小越好。

Note

示例把 `float32` 步长与 `np.pi` 混用，两次余弦计算可能使用不同精度的基点；很小的基点差异再除以 h ，也会造成大误差。复现实验时应统一 x 、 h 和两次函数求值的类型，并检查中间结果的精度。

Tip

本页要点： 相消会放大相对误差，改写公式通常比单纯提高精度更有效。

3.15 Effects of Floating-Point Arithmetic

编译器和硬件可能执行以下变换：

- 把 $(a + b) + c$ 重排为 $a + (b + c)$ ；
- 把除法 a/b 近似为 $a \times (1/b)$ ；
- 把 $a + 0$ 或 $a \times 0$ 化简，其中后者在 $a = \infty$ 或 NaN 时并不保持 IEEE 语义；
- 选用不同精度或不同实现的 $\sin$ 、 $\log$ 等数学函数；
- 用 fused multiply-add 一次舍入计算 $ab + c$ ；
- 对次正规数启用 flush-to-zero；
- 在不同 SIMD 指令集上采用不同归约树。

因此，算法误差分析要在实现之前完成，并明确依赖的语义。例如，证明若使用了“每次乘法和加法各舍入一次”，启用 FMA 后误差通常会更小，但原来的逐步模型已经改变。

Tip

本页要点： 重排、FMA 和快速数学会改变浮点运算顺序，性能提升必须配合误差检查。

3.16 Error Analysis for Inner Product

设 $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ ，按从左到右的顺序计算内积，每次乘法和加法分别舍入。沿用第 3.11 页的相对误差模型，记前 p 项的计算部分和为

$$s_p = fl\left(\sum_{k=1}^p x_k y_k\right), \quad p = 1, \dots, n. \quad (3.26)$$

这里 n 是向量维数, p 是当前已经累加的项数, k 是求和下标; fl 表示整个浮点计算过程, 不是先精确求和再舍入一次。最终计算结果为 $s_n = fl(\mathbf{x}^T \mathbf{y})$ 。

第一步只有乘法, 没有加法:

$$s_1 = x_1 y_1 (1 + \delta_1), \quad |\delta_1| \leq u. \quad (3.27)$$

对 $p \geq 2$, 乘法和加法分别舍入:

$$s_p = (s_{p-1} + x_p y_p (1 + \delta_p)) (1 + \varepsilon_p), \quad |\delta_p|, |\varepsilon_p| \leq u. \quad (3.28)$$

其中 δ_p 是第 p 次乘法的相对舍入误差, ε_p 是将该乘积加入部分和时的相对舍入误差。先展开前两次累加 ($n \geq 3$) :

$$\begin{aligned} s_2 &= x_1 y_1 (1 + \delta_1) (1 + \varepsilon_2) + x_2 y_2 (1 + \delta_2) (1 + \varepsilon_2), \\ s_3 &= x_1 y_1 (1 + \delta_1) (1 + \varepsilon_2) (1 + \varepsilon_3) \\ &\quad + x_2 y_2 (1 + \delta_2) (1 + \varepsilon_2) (1 + \varepsilon_3) \\ &\quad + x_3 y_3 (1 + \delta_3) (1 + \varepsilon_3). \end{aligned} \quad (3.29)$$

可见, 第 k 个乘积 $x_k y_k$ 在第 k 步进入部分和。此后每进行一次加法, 它在新部分和中的系数就再乘一个舍入因子。因此, 计算到第 p 步时, 只涉及截至 ε_p 的因子。第一项直接用于初始化, 约定 $\varepsilon_1 = 0$, 即可统一写成

$$\begin{aligned} s_p &= \sum_{k=1}^p x_k y_k (1 + \theta_{k,p}), \\ 1 + \theta_{k,p} &= (1 + \delta_k) \prod_{j=k}^p (1 + \varepsilon_j), \quad 1 \leq k \leq p \leq n. \end{aligned} \quad (3.30)$$

这个展开也可由归纳法验证: $p = 1$ 时就是初始化公式; 若对 $p - 1$ 成立, 代入递推式后, 已有各项都乘上 $1 + \varepsilon_p$, 使乘积上限从 $p - 1$ 变为 p ; 新加入的第 p 项则带有 $(1 + \delta_p)(1 + \varepsilon_p)$ 。两部分合起来便得到上式。

$\theta_{k,p}$ 表示第 k 个乘积在计算部分和 s_p 中的累积相对扰动, 不是部分和 s_k 自身的误差。已经算出的 s_p 不依赖后续运算; 后续舍入影响的是它对新部分和的贡献。

最后取 $p = n$, 得到最终内积的展开:

$$fl(\mathbf{x}^T \mathbf{y}) = s_n = \sum_{k=1}^n x_k y_k (1 + \theta_{k,n}), \quad 1 + \theta_{k,n} = (1 + \delta_k) \prod_{j=k}^n (1 + \varepsilon_j). \quad (3.31)$$

Tip

本页要点: 累积扰动要同时注明对应哪个乘积、累加到哪一步; 最终内积是部分和的特例。

3.17 Higham's Lemma

Higham 引理. 若 $|\delta_i| \leq u$ 、 $nu < 1$, 则存在 θ_n 使

$$\prod_{i=1}^n (1 + \delta_i) = 1 + \theta_n, \quad |\theta_n| \leq \gamma_n := \frac{nu}{1 - nu}. \quad (3.32)$$

这里的 n 表示引理中的因子个数。 θ_n 是随具体舍入误差变化的实际扰动，可正可负； γ_n 是只依赖 n, u 的非负上界。

证明. 先估计乘积相对 1 的偏差：

$$\left| \prod_{i=1}^n (1 + \delta_i) - 1 \right| \leq \prod_{i=1}^n (1 + |\delta_i|) - 1 \leq (1 + u)^n - 1. \quad (3.33)$$

二项式系数满足 $\binom{n}{k} \leq n^k$ ，故

$$(1 + u)^n = \sum_{k=0}^n \binom{n}{k} u^k \leq \sum_{k=0}^n (nu)^k \leq \frac{1}{1 - nu}. \quad (3.34)$$

因此

$$|\theta_n| \leq \frac{1}{1 - nu} - 1 = \frac{nu}{1 - nu} = \gamma_n. \quad (3.35)$$

对包含逆因子的形式，若 $\rho_i \in \{-1, 1\}$ ，则

$$\prod_{i=1}^n (1 + \delta_i)^{\rho_i} = 1 + \hat{\theta}_n, \quad |\hat{\theta}_n| \leq \gamma_n. \quad (3.36)$$

这是因为每个因子都落在 $[1 - u, (1 - u)^{-1}]$ 内，从而整个乘积落在 $[(1 - u)^n, (1 - u)^{-n}]$ 内。Bernoulli 不等式给出 $(1 - u)^n \geq 1 - nu$ ，故下偏差不超过 $nu \leq \gamma_n$ ，上偏差不超过 $(1 - nu)^{-1} - 1 = \gamma_n$ 。证毕。

将引理用于第 3.16 页的内积展开。第一项包含一次乘法及第 $2, \dots, n$ 步的加法舍入，共 n 个非平凡因子；第 $k \geq 2$ 项包含一次乘法及第 $k, \dots, n$ 步的加法舍入，共 $n - k + 2 \leq n$ 个因子。因此，若 $nu < 1$ ，则可统一估计 $|\theta_{k,n}| \leq \gamma_n$ 。从计算值中减去精确内积，并用三角不等式，得到

$$\begin{aligned} |fl(\mathbf{x}^T \mathbf{y}) - \mathbf{x}^T \mathbf{y}| &= \left| \sum_{k=1}^n x_k y_k \theta_{k,n} \right| \\ &\leq \sum_{k=1}^n |x_k| |y_k| |\theta_{k,n}| \\ &\leq \gamma_n \sum_{k=1}^n |x_k| |y_k| = \gamma_n |\mathbf{x}|^T |\mathbf{y}|. \end{aligned} \quad (3.37)$$

最后一个表达式中的向量绝对值按分量理解。

进一步作一阶展开，可得

$$|fl(\mathbf{x}^T \mathbf{y}) - \mathbf{x}^T \mathbf{y}| \leq \gamma_n |\mathbf{x}|^T |\mathbf{y}| = (nu + O(u^2)) |\mathbf{x}|^T |\mathbf{y}|. \quad (3.38)$$

这里的 $O(u^2)$ 是固定 n 和输入数据时的记法，高阶项的系数仍依赖 n 。若要求一个不写高阶项的简洁界，可固定 $0 < c < 1$ ，在 $nu \leq c$ 时用 $\gamma_n \leq Cnu$ ，其中 $C = (1 - c)^{-1}$ ；例如 $nu \leq 1/2$ 时可取 $C = 2$ 。

💡 Tip

本页要点： Higham 引理把一串舍入误差压缩成一个可估计的乘法因子。

3.18 Running Error Analysis

令 $s_0 = \hat{s}_0 = 0$ 、 $\hat{z}_i = fl(x_i y_i)$ ，精确部分和与计算部分和分别为

$$s_i = s_{i-1} + z_i, \quad \hat{s}_i = fl(\hat{s}_{i-1} + \hat{z}_i), \quad z_i = x_i y_i. \quad (3.39)$$

本页采用向最近数舍入，并假设没有溢出、下溢。舍入误差也可以相对于计算结果来表示：

$$\hat{s}_{i-1} + \hat{z}_i = \hat{s}_i(1 + \varepsilon_i), \quad z_i = \hat{z}_i(1 + \eta_i), \quad |\varepsilon_i|, |\eta_i| \leq u. \quad (3.40)$$

其依据是：精确结果与最近浮点数的距离不超过半个相邻间距，而这个距离不超过计算结果绝对值的 u 倍。这是向最近数舍入的另一种误差表示，不能仅由正向模型取倒数后仍沿用同一个误差参数。

记 $e_i = \hat{s}_i - s_i$ ，加法和乘法分别引入误差 $\rho_i := \hat{s}_i - (\hat{s}_{i-1} + \hat{z}_i)$ 、 $\pi_i := \hat{z}_i - z_i$ 。由上式，

$$\rho_i = -\varepsilon_i \hat{s}_i, \quad \pi_i = -\eta_i \hat{z}_i, \quad e_i = e_{i-1} + \rho_i + \pi_i. \quad (3.41)$$

因此

$$|e_i| \leq |e_{i-1}| + u|\hat{s}_i| + u|\hat{z}_i|. \quad (3.42)$$

随内积计算累积这些已经得到的量：

$$\mu_i = \mu_{i-1} + |\hat{s}_i| + |\hat{z}_i|, \quad \mu_0 = 0, \quad (3.43)$$

由 $e_0 = 0$ 逐步累加式 (3.42)，得到 $|e_n| \leq u\mu_n$ 。这种 running error bound 利用实际中间量，通常比只依赖项数的先验界更紧。

📌 Note

备注：误差界本身也需要可靠地计算。 上式的 μ_i 是精确递推值。若用普通浮点加法更新它，得到的是误差估计；要保证最终结果确为上界，累加和最后的 $u\mu_n$ 都需要向上舍入，或采用经过证明的保守放大。

💡 Tip

本页要点： running error 随实际中间量更新，通常比只看项数的先验界更贴近计算过程。

3.19 Blocked Algorithms

串行内积的误差常数约为 nu 。设 $n = kb$ ，把 n 项分成 k 个大小为 $b = n/k$ 的块，先在块内计算内积，再累加 k 个块和。假设 $bu < 1$ 、 $ku < 1$ ，块内误差系数至多为 γ_b ，块间求和的系数可用 γ_k 估计。每个块和还满足“计算值的绝对值不超过 $(1 + \gamma_b)$ 倍的块内绝对乘积和”，因而总误差满足

$$|\hat{s} - s| \leq (\gamma_{n/k} + \gamma_k + \gamma_{n/k}\gamma_k)|x|^T|y|. \quad (3.44)$$

忽略 u^2 项后，系数约为

$$\left(\frac{n}{k} + k\right)u. \quad (3.45)$$

它在 $k \approx \sqrt{n}$ 时达到约 $2\sqrt{n}u$ ，当 n 较大时比 nu 小得多。分块没有减少加法总数，改善来自缩短最长依赖链；同时，块也是缓存复用、SIMD 和多线程任务划分的自然单位。

💡 Tip

本页要点： 分块求和缩短误差累积链，改善数据局部性，也有利于提高计算性能。

3.20 Further Improvement on Inner Product

成对求和递归地把序列一分为二，最后合并两个子和。归约树高度为 $\lceil \log_2 n \rceil$ ，每个输入项只经历这一数量级的舍入，因此误差常数从 $O(nu)$ 降为

$$O(u \log n). \quad (3.46)$$

更高精度累加、补偿求和或长累加器还能进一步改善结果。

再比较普通乘加与 fused multiply-add。普通实现先乘后加：

$$fl(x + fl(yz)) = (x + yz(1 + \delta_1))(1 + \delta_2), \quad (3.47)$$

而 FMA 只舍入一次：

$$\text{fma}(y, z, x) = (x + yz)(1 + \delta), \quad |\delta| \leq u. \quad (3.48)$$

📌 Note

备注： 舍入次数不等于相对误差倍数。

FMA 避免乘积的中间舍入，尤其在相消时能改善精度。以 $|x| + |yz|$ 为尺度，一阶误差界从约 $2u$ 降为 u ：普通乘加的绝对误差不超过 $(2u + u^2)(|x| + |yz|)$ ，FMA 的绝对误差不超过 $u|x + yz| \leq u(|x| + |yz|)$ 。前者相对于 $x + yz$ 的相对误差未必小于 $2u$ 。

FABsum 算法的一阶误差系数可以不随项数 n 增长；高阶项仍可能依赖 n 。具体算法见 Blanchard、Higham 和 Mary, *A Class of Fast and Accurate Summation Algorithms*, SIAM J. Sci. Comput. 42 (2020), A1541 – A1557。

💡 Tip

本页要点： 成对求和把误差累积深度从线性降到对数级，但归约树仍影响可复现性。

3.21 Matrix-Vector Multiplication

矩阵-向量乘法可以从两种等价角度理解。按行看，若 A 的第 i 行为 $\mathbf{a}_i^T$ ，则

$$A\mathbf{x} = \begin{bmatrix} \mathbf{a}_1^T \mathbf{x} \\ \vdots \\ \mathbf{a}_m^T \mathbf{x} \end{bmatrix}; \quad (3.49)$$

每个输出分量都是一个内积。按列看，若 $A = [c_1, \dots, c_n]$ ，则

$$A\mathbf{x} = \sum_{j=1}^n x_j c_j; \quad (3.50)$$

输出是矩阵列向量的线性组合。

按行实现适合 CSR 稀疏格式，每行可以独立归约；按列实现适合按列存储或矩阵分块。两种精确公式相同，但内存访问、并行划分和浮点加法顺序不同。把内积误差界逐行应用，得到分量式

$$|fl(A\mathbf{x}) - A\mathbf{x}| \leq \gamma_n |A| |\mathbf{x}|. \quad (3.51)$$

Tip

本页要点： 矩阵-向量乘法既可按行理解为内积，也可按列理解为线性组合。

3.22 Challenges in Low-Precision FP

降低精度不仅减少有效数字，还可能缩小可表示范围。转换到低精度时，过大的数会溢出，过小的数可能下溢；矩阵条目的舍入还可能破坏正定性，例如最小特征值已经与舍入扰动同量级时。转换前可先作适当缩放，减小溢出的风险；第 3.24 页介绍用于特征值问题的矩阵平衡。

误差也不只发生在运算中：把输入存成低精度数就已引入一次舍入。对混合精度矩阵向量积，应分别考虑输入转换、乘积和累加的精度，不能只给出一个“FP16/FP32”标签。

第 3.17 页给出的误差界 $\gamma_n = nu/(1 - nu)$ 要求 $nu < 1$ 。低精度中 u 较大，当 $nu \geq 1$ 时，这个界不再适用。改进方向包括缩放与移位，以及概率误差分析：前者调节数据范围或谱的位置，后者在适当假设下分析舍入误差的抵消。

Tip

本页要点： 混合精度除了舍入误差，还要防止范围不足、溢出和下溢。

3.23 Working With Mixed Precisions

若矩阵-向量乘法在精度 u 中完成，则

$$fl_u(A\mathbf{x}) = A\mathbf{x} + \delta, \quad |\delta| \leq \gamma_n^{(u)} |A| |\mathbf{x}|, \quad (3.52)$$

其中 $\gamma_n^{(u)} = nu/(1 - nu)$ 。以下误差界假设无溢出、下溢，且相应的 γ 因子分母为正。残量计算还包含一次减法：

$$fl_u(\mathbf{b} - A\mathbf{x}) = \mathbf{b} - A\mathbf{x} + \eta, \quad |\eta| \leq \gamma_{n+1}^{(u)} (|\mathbf{b}| + |A| |\mathbf{x}|). \quad (3.53)$$

这些是最坏情形界，假定舍入误差都可能朝同一方向累积；在适当的随机误差假设下，概率误差分析可以给出更小的高概率界。

现在先用精度 $\bar{u}$ 计算残量，再用精度 u 存储。令 $\mathbf{r} = \mathbf{b} - A\mathbf{x}$ ，则

$$\tilde{\mathbf{r}} = fl_{\bar{u}}(\mathbf{b} - A\mathbf{x}) = \mathbf{r} + \mathbf{e}, \quad \hat{\mathbf{r}} = fl_u(\tilde{\mathbf{r}}). \quad (3.54)$$

记 $\bar{\gamma}_{n+1} = (n+1)\bar{u}/[1 - (n+1)\bar{u}]$ 。由式 (3.53)， $|\mathbf{e}| \leq \bar{\gamma}_{n+1}(|\mathbf{b}| + |A||\mathbf{x}|)$ 。最终存储满足 $\hat{r}_i = \tilde{r}_i(1 + \theta_i)$ 、 $|\theta_i| \leq u$ ，因此

$$\begin{aligned} \hat{r}_i - r_i &= \theta_i r_i + (1 + \theta_i)e_i, \\ |\hat{\mathbf{r}} - \mathbf{r}| &\leq u|\mathbf{r}| + (1 + u)\bar{\gamma}_{n+1}(|\mathbf{b}| + |A||\mathbf{x}|). \end{aligned} \quad (3.55)$$

当 $\bar{u} \ll u$ 时，高精度计算压低了第二项；最终存储引入的第一项与残量本身同量级。迭代接近收敛时， $|\mathbf{r}|$ 可能远小于 $|\mathbf{b}| + |A||\mathbf{x}|$ ，所以“高精度计算残量、工作精度存储”仍能明显提高残量的准确性。

Tip

本页要点： 混合精度的收益来自低精度吞吐，而可靠性依赖高精度残量和修正。

3.24 Matrix Balancing

对 $A \in \mathbb{C}^{n \times n}$ 取正对角矩阵

$$D = \text{diag}(d_1, \dots, d_n), \quad \tilde{A} = D^{-1}AD. \quad (3.56)$$

这是相似变换，所以 A 与 $\tilde{A}$ 的特征值完全相同。平衡的目标是选择 d_i ，使 $\tilde{A}$ 第 i 行与第 i 列的范数大致相等，避免某些行列量级悬殊。这里使用非对角元素的绝对值和，记为 $R_i = \sum_{j \neq i} |\tilde{a}_{ij}|$ 、 $C_i = \sum_{j \neq i} |\tilde{a}_{ji}|$ 。

把 d_i 乘以 $f_i > 0$ ，相当于将第 i 行除以 f_i 、第 i 列乘以 f_i ；对角元不变，非对角行、列和变为 R_i/f_i 和 $f_i C_i$ 。令二者相等，得到理想缩放因子

$$f_i = \sqrt{\frac{R_i}{C_i}}, \quad R_i, C_i > 0. \quad (3.57)$$

代码先跳过行、列和过小的情况，再把因子限制在 $[1/2, 2]$ ，并用 $2^{\text{round}(\log_2 f_i)}$ 量化为 2 的整数次幂，以避免一次缩放过大。在输入和结果都处于正规数范围时，这种二进制缩放是精确的。由于量化且一次更新会影响其他行列，实际算法追求逐步近似平衡，通常需要多轮扫描。

平衡常用于特征值计算的预处理，但不保证对所有矩阵都能改善精度；还需要考虑把特征向量变换回原坐标时的缩放影响。

Tip

本页要点： 矩阵平衡通过相似缩放改善行列量级，同时保持特征值不变。

3.25 Intel® Compilers

Intel oneAPI 提供编译器和数学库等开发工具。要复现数值程序的构建过程，需要记录 C/C++ 编译器版本、Fortran 编译器版本、数学库、目标指令集、浮点模型和链接方式。

Intel 当前工具链通常以 `icx/icpx` 和 `ifx` 为主，并可链接 oneMKL。验证数值结果时，应保存完整编译命令并打开向量化报告。循环是否被向量化、是否使用 FMA、是否采用快速数学库，这些信息有助于解释结果变化。

 Tip

本页要点： 复现数值结果需要完整的构建信息，包括编译器、数学库和浮点选项。

3.26 FP Accuracy and Reproducibility

浮点计算的准确性与可复现性是两件事：准确性关心结果与精确值有多接近；可复现性关心重复运行或更换计算环境后，结果是否一致。结果完全相同并不代表准确，末几位不同也不一定影响所需精度。

以 Intel 2018 年的编译器说明为例，可从以下几方面控制结果的一致性：

1. **浮点求值方式。** Windows 的 `/fp:precise /fp:source`，或 Linux 的 `-fp-model precise -fp-model source`，用于约束浮点优化及中间表达式的精度，减少编译方式造成的差异。
2. **数学函数库。** 同一体系结构的不同处理器可能选用不同的函数实现。`/Qimf-arch-consistency:true` 或 `-fimf-arch-consistency=true` 用于提高这些实现之间的结果一致性。
3. **FMA 的使用。** FMA 只舍入一次，分开的乘法和加法通常舍入两次，因此两者结果可能不同。若要比对支持与不支持 FMA 的处理器，可用 `/Qfma-` 或 `-no-fma` 禁用融合；这是统一计算路径，并不是提高精度。
4. **组合选项。** 该说明介绍的 Intel 编译器从第 17 版起提供 `/fp:consistent` 或 `-fp-model consistent`，组合设置上述选项。

 Note

备注：编译选项不能单独保证并行结果一致。 上述名称和组合关系来自特定版本，使用时应核对当前编译器的支持情况。线程数、MPI 归约树或任务调度改变后，加法顺序仍可能变化；位级复现还需要控制归约顺序或采用可复现归约算法。限制优化可能降低性能，应根据应用所需的误差容限与复现要求选择。

参考：[Intel, Consistency of Floating-Point Results using the Intel Compiler \(2018 年版本\)](#)。

 Tip

本页要点： 准确性看误差，可复现性看一致性；两者需要分别验证。

3.27 FP Model for Intel® Compilers

本页以 Intel 2014 年的选项说明为例，介绍浮点模型对优化的约束：

- `precise` 只允许保持数值语义的优化。
- `source` 控制中间表达式采用的精度，`except` 要求保留浮点异常语义。
- `strict` 包含 `precise` 和 `except` 的约束，允许程序访问浮点环境，并禁止把分开的乘、加运算融合为 FMA。
- `fast=1` 允许改变求值方式的优化，不严格保持浮点异常语义；`fast=2` 进一步允许近似。

这些选项并不是简单的一系列精度等级，部分选项控制的是不同属性。具体含义和默认值应以所用编译器版本为准；性能比较必须同时报告误差。

💡 Tip

本页要点： 编译器的浮点模型是在性能、严格语义和结果复现之间作选择。

3.28 An Example of FP Option

考虑循环

$$s \leftarrow s + x_i^2, \quad i = 0, \dots, n-1. \quad (3.58)$$

串行程序形成长度 n 的依赖链。严格或精确模式可能禁止自动向量化，因为 SIMD 归约会先在多个向量通道中累加，再合并部分和，从而改变括号结构。加入 OpenMP SIMD reduction 指令等于程序员明确授权这种重排。

树形归约的深度约为 $\log_2 n$ ，通常比串行累加的最坏误差界更好，但结果依赖线程数和归约树。若应用要求位级复现，应固定归约算法；若只要求误差界，则应验证重排后结果仍满足容差。

💡 Tip

本页要点： 串行累加形成长度为 n 的依赖链，重排归约会改变误差和可复现性。

第3.2节 条件数与病态问题

3.30 Condition of Function Evaluation

设要计算 $y = f(x)$ ，算法得到 $\hat{y}$ 。前向误差是输出空间中的差

$$\delta y = \hat{y} - y. \quad (3.59)$$

后向误差寻找一个扰动 δx ，使

$$\hat{y} = f(x + \delta x). \quad (3.60)$$

若 f 在 x 附近二阶连续可微，一阶 Taylor 展开给出

$$f(x + \delta x) = f(x) + f'(x)\delta x + O(\delta x^2), \quad (3.61)$$

所以

$$|\delta y| \approx |f'(x)| |\delta x|. \quad (3.62)$$

绝对条件数是 $|f'(x)|$ 。若 $x \neq 0$ 、 $f(x) \neq 0$ ，**相对条件数**为

$$\kappa_f(x) = \left| \frac{x f'(x)}{f(x)} \right|, \quad (3.63)$$

它把相对输入扰动放大为相对输出扰动。

💡 Tip

本页要点： 函数条件数描述输入扰动如何传到输出，是算法误差分析的起点。

3.31 Backward and Forward Error Analysis

对同一输入，精确求解与数值计算分别给出：

$$x \xrightarrow{\text{精确问题}} y = f(x), \quad x \xrightarrow{\text{数值算法}} \hat{y}. \quad (3.64)$$

前向误差衡量同一输入下 y 与 $\hat{y}$ 的差异。后向分析则在输入端寻找 $x + \Delta x$ ，使

$$\hat{y} = f(x + \Delta x), \quad (3.65)$$

并测量输入端的距离 $|\Delta x|$ 。如果不能让 $\hat{y}$ 完全等于某个邻近问题的精确输出，可使用混合前向-后向模型：

$$\hat{y} + \Delta y = f(x + \Delta x), \quad |\Delta y| \leq \varepsilon |y|, \quad |\Delta x| \leq \eta |x|. \quad (3.66)$$

它允许同时略微调整输入和输出； η 与 ε 分别控制两端的相对扰动。

💡 Tip

本页要点： 后向误差说明算出了哪个邻近问题，前向误差说明答案离目标有多远。

3.32 Backward and Forward Errors

对小扰动，一阶误差关系可概括为

$$\text{相对前向误差} \lesssim \text{条件数} \times \text{相对后向误差}. \quad (3.67)$$

这里的 $\lesssim$ 表示忽略高阶项后的上界，并不表示三个量总满足等式。后向稳定是指计算结果等于某个轻微扰动问题的精确解，扰动通常为机器精度乘以适当的维数因子。

条件数很大表示问题本身病态：即使后向误差很小，前向误差也可能很大。条件数适中时，后向稳定则能保证前向误差也小。因此，评价计算结果要同时看问题的条件数和算法的稳定性。

Wilkinson 系统发展了舍入误差与后向误差分析，并因此获得 1970 年图灵奖。

💡 Tip

本页要点： 后向稳定控制算法引入的扰动，条件数决定这些扰动会被放大多少。

3.33 Measuring Sensitivity, Revisited

当 $x \neq 0$ 、 $f(x) \neq 0$ 时，一般函数的相对条件数可写成

$$\kappa_f(x) = \lim_{\varepsilon \rightarrow 0} \sup_{0 < \|\delta x\| \leq \varepsilon \|x\|} \frac{\|f(x + \delta x) - f(x)\| / \|f(x)\|}{\|\delta x\| / \|x\|}. \quad (3.68)$$

对线性系统 $Ax = b$ ，把 A 固定，映射为 $f(b) = A^{-1}b$ 。扰动满足

$$\delta x = A^{-1}\delta b. \quad (3.69)$$

因此

$$\frac{\|\delta x\|}{\|x\|} \leq \|A^{-1}\| \frac{\|\delta b\|}{\|b\|} \frac{\|b\|}{\|x\|} \leq \|A^{-1}\| \|A\| \frac{\|\delta b\|}{\|b\|}. \quad (3.70)$$

对所有非零 b 和扰动方向取最坏情形，两个因子可分别达到各自的最大值。利用 $b = Ax$ ，得到

$$\begin{aligned} \kappa(A) &= \max_{\delta b \neq 0} \frac{\|A^{-1}\delta b\|}{\|\delta b\|} \max_{b \neq 0} \frac{\|b\|}{\|A^{-1}b\|} \\ &= \|A^{-1}\| \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \|A^{-1}\| \|A\|. \end{aligned} \quad (3.71)$$

Tip

本页要点： 敏感性分析要先明确扰动的输入量，再看它如何传到输出。

3.34 Condition Number, Revisited

对非奇异方阵， p -范数条件数定义为

$$\kappa_p(A) = \|A\|_p \|A^{-1}\|_p. \quad (3.72)$$

对实矩阵， $\|A\|_2 = \sqrt{\rho(A^T A)} = \sigma_1(A)$ ，而 $\|A^{-1}\|_2 = 1/\sigma_n(A)$ ；复矩阵将 A^T 换成 A^H 即可。因此，由 SVD Theorem 2.7 ([02线性代数基础.pdf](#) 第 2.42—2.44 页)，

$$\kappa_2(A) = \frac{\sigma_1(A)}{\sigma_n(A)}. \quad (3.73)$$

若 A 是正规矩阵，则奇异值是特征值模，因而

$$\kappa_2(A) = \frac{\max_i |\lambda_i(A)|}{\min_i |\lambda_i(A)|}. \quad (3.74)$$

酉矩阵的全部奇异值为 1，所以 $\kappa_2(Q) = 1$ 。左、右乘酉矩阵不改变二范数条件数： $\kappa_2(UAV) = \kappa_2(A)$ 。

Tip

本页要点： 条件数刻画问题本身的放大能力，不要把它和算法稳定性混为一谈。

3.35 Theorem 3.1: Backward Error of Linear System Solver

定理 3.1（线性系统的范数后向误差）。 取任意向量范数及其诱导矩阵范数，均记为 $\|\cdot\|$ 。给定近似解 y ，定义残量

$$r = b - Ay \quad (3.75)$$

以及

$$\eta(\mathbf{y}) = \min \{ \varepsilon \geq 0 : (A + \Delta A)\mathbf{y} = \mathbf{b} + \Delta \mathbf{b}, \|\Delta A\| \leq \varepsilon \|F\|, \|\Delta \mathbf{b}\| \leq \varepsilon \|\mathbf{f}\| \}. \quad (3.76)$$

这里 F 和 $\mathbf{f}$ 给出矩阵与右端的扰动尺度；取 $F = A$ 、 $\mathbf{f} = \mathbf{b}$ ，便得到通常的相对后向误差。假设

$$D := \|F\|\|\mathbf{y}\| + \|\mathbf{f}\| > 0, \quad (3.77)$$

则

$$\eta(\mathbf{y}) = \frac{\|\mathbf{r}\|}{\|F\|\|\mathbf{y}\| + \|\mathbf{f}\|}. \quad (3.78)$$

证明.

(1) **建立下界。** 对任意满足定义的扰动，

$$\mathbf{r} = \mathbf{b} - A\mathbf{y} = \Delta A\mathbf{y} - \Delta \mathbf{b}. \quad (3.79)$$

所以

$$\|\mathbf{r}\| \leq \|\Delta A\|\|\mathbf{y}\| + \|\Delta \mathbf{b}\| \leq \varepsilon D, \quad (3.80)$$

即每个可行 ε 都满足 $\varepsilon \geq \|\mathbf{r}\|/D$ 。

(2) **构造达到下界的扰动。** 先设 $\mathbf{y} \neq 0$ 。记对偶范数为 $\|\mathbf{w}\|_* = \max_{\|\mathbf{v}\|=1} |\mathbf{w}^H \mathbf{v}|$ 。由范数的对偶表示 $\|\mathbf{y}\| = \max_{\|\mathbf{w}\|_*=1} |\mathbf{w}^H \mathbf{y}|$ ，可选取 $\|\mathbf{w}\|_* = 1$ 且 $\mathbf{w}^H \mathbf{y} = \|\mathbf{y}\|$ 的向量：有限维空间中最大值可取到，再调整符号或复相位即可。令

$$\mathbf{z} = \frac{\mathbf{w}}{\|\mathbf{y}\|}, \quad \mathbf{z}^H \mathbf{y} = 1, \quad \|\mathbf{z}\|_* = \frac{1}{\|\mathbf{y}\|}. \quad (3.81)$$

取

$$\Delta A_{\min} = \frac{\|F\|\|\mathbf{y}\|}{D} \mathbf{r} \mathbf{z}^H, \quad \Delta \mathbf{b}_{\min} = -\frac{\|\mathbf{f}\|}{D} \mathbf{r}. \quad (3.82)$$

则

$$\Delta A_{\min} \mathbf{y} - \Delta \mathbf{b}_{\min} = \frac{\|F\|\|\mathbf{y}\| + \|\mathbf{f}\|}{D} \mathbf{r} = \mathbf{r}. \quad (3.83)$$

由诱导矩阵范数和对偶范数的定义，

$$\begin{aligned} \|\mathbf{r} \mathbf{z}^H\| &= \max_{\|\mathbf{v}\|=1} \|\mathbf{r}(\mathbf{z}^H \mathbf{v})\| \\ &= \|\mathbf{r}\| \|\mathbf{z}\|_* = \frac{\|\mathbf{r}\|}{\|\mathbf{y}\|}. \end{aligned} \quad (3.84)$$

于是

$$\|\Delta A_{\min}\| = \frac{\|F\|\|\mathbf{r}\|}{D}, \quad \|\Delta \mathbf{b}_{\min}\| = \frac{\|\mathbf{f}\|\|\mathbf{r}\|}{D}. \quad (3.85)$$

两项都以 $\varepsilon = \|\mathbf{r}\|/D$ 达到允许上界。若 $\mathbf{y} = \mathbf{0}$, 则 $D = \|\mathbf{f}\| > 0$, 取 $\Delta A = 0$ 、 $\Delta \mathbf{b} = -\mathbf{r}$ 即可达到相同下界。证毕。

Tip

本页要点: 后向误差回答计算结果精确求解了哪个邻近线性系统。

3.36 Theorem 3.2: Forward Error of Linear System Solver

定理 3.2. 设 A 非奇异、 $\mathbf{b} \neq \mathbf{0}$, 且

$$A\mathbf{x} = \mathbf{b}, \quad (A + \Delta A)\mathbf{y} = \mathbf{b} + \Delta \mathbf{b}. \quad (3.86)$$

若

$$\frac{\|\Delta A\|}{\|A\|} \leq \varepsilon, \quad \frac{\|\Delta \mathbf{b}\|}{\|\mathbf{b}\|} \leq \varepsilon, \quad \delta := \varepsilon \kappa(A) < 1, \quad (3.87)$$

则 $A + \Delta A$ 非奇异, 并且

$$\frac{\|\mathbf{y}\|}{\|\mathbf{x}\|} \leq \frac{1 + \delta}{1 - \delta}, \quad \frac{\|\mathbf{y} - \mathbf{x}\|}{\|\mathbf{x}\|} \leq \frac{2\varepsilon}{1 - \delta} \kappa(A). \quad (3.88)$$

考虑右端扰动的例子, 取

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 10^{-6} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 10^{-6} \end{bmatrix}, \quad \Delta \mathbf{b} = \begin{bmatrix} 0 \\ 10^{-6} \end{bmatrix}. \quad (3.89)$$

原解为 $(1, 1)^T$, 扰动后解为 $(1, 2)^T$, 故无穷范数相对解误差为 1。这里仅扰动右端, 直接由 $\mathbf{y} - \mathbf{x} = A^{-1}\Delta \mathbf{b}$ 得到第 3.33 页的界 $\|\mathbf{y} - \mathbf{x}\|_\infty / \|\mathbf{x}\|_\infty \leq \varepsilon \kappa_\infty(A) = 1$ 。它不需要 $\varepsilon \kappa_\infty(A) < 1$, 所以本例虽不满足上方定理的严格条件, 仍可用这个更直接的界解释。

Tip

本页要点: 前向误差由后向误差和问题条件数共同决定。

3.37 Proof of Sensitivity Analysis (Theorem 3.2)

(1) 证明扰动矩阵非奇异。由分解

$$A + \Delta A = A(I + A^{-1}\Delta A) \quad (3.90)$$

和

$$\|A^{-1}\Delta A\| \leq \|A^{-1}\| \|\Delta A\| \leq \varepsilon \|A^{-1}\| \|A\| = \delta < 1, \quad (3.91)$$

Neumann 级数定理 Theorem 2.11 ([02线性代数基础.pdf](#) 第 2.77 页) 说明 $I + A^{-1}\Delta A$ 非奇异, 且

$$\|(I + A^{-1}\Delta A)^{-1}\| \leq \frac{1}{1 - \delta}. \quad (3.92)$$

因此 $A + \Delta A$ 也非奇异。

(2) 估计扰动解的大小。在扰动方程左乘 A^{-1} ，有

$$(I + A^{-1}\Delta A)y = x + A^{-1}\Delta b. \quad (3.93)$$

用上面的逆矩阵界和 $\|\Delta b\| \leq \varepsilon\|b\|$ ，得到

$$\begin{aligned} \|y\| &\leq \frac{1}{1-\delta}(\|x\| + \|A^{-1}\|\|\Delta b\|) \\ &\leq \frac{1}{1-\delta}(\|x\| + \varepsilon\|A^{-1}\|\|b\|). \end{aligned} \quad (3.94)$$

由 $b = Ax$ 有 $\|b\| \leq \|A\|\|x\|$ ，所以

$$\frac{\|y\|}{\|x\|} \leq \frac{1 + \varepsilon\|A^{-1}\|\|A\|}{1 - \delta} = \frac{1 + \delta}{1 - \delta}. \quad (3.95)$$

(3) 估计解的相对误差。从两个系统相减并左乘 A^{-1} ，得到

$$y - x = A^{-1}\Delta b - A^{-1}\Delta Ay. \quad (3.96)$$

取范数，除以 $\|x\|$ ，再代入第 (2) 步的估计：

$$\begin{aligned} \frac{\|y - x\|}{\|x\|} &\leq \varepsilon\kappa(A) \left(\frac{\|b\|}{\|A\|\|x\|} + \frac{\|y\|}{\|x\|} \right) \\ &\leq \varepsilon\kappa(A) \left(1 + \frac{1 + \delta}{1 - \delta} \right) = \frac{2\varepsilon}{1 - \delta}\kappa(A). \end{aligned} \quad (3.97)$$

证毕。

Tip

本页要点： Neumann 级数把逆矩阵扰动写成几何级数，关键是假设扰动范数小于一。

3.38 Ill-Conditioned Systems

设计算解 $\hat{x}$ 是邻近系统的精确解：

$$(A + E)\hat{x} = b + e, \quad \|E\|_\infty \leq u\|A\|_\infty, \quad \|e\|_\infty \leq u\|b\|_\infty. \quad (3.98)$$

当 $u\kappa_\infty(A) < 1$ 时，把 Theorem 3.2 ([03矩阵计算基础.pdf](#) 第 3.36—3.37 页) 中的 ε 取为 u ，得到

$$\frac{\|\hat{x} - x\|_\infty}{\|x\|_\infty} \leq \frac{2u\kappa_\infty(A)}{1 - u\kappa_\infty(A)}. \quad (3.99)$$

若 $u\kappa_\infty(A) \leq 1/2$ ，则 $(1 - u\kappa_\infty(A))^{-1} \leq 2$ ，所以

$$\frac{\|\hat{x} - x\|_\infty}{\|x\|_\infty} \leq 4u\kappa_\infty(A). \quad (3.100)$$

当 $u\kappa_\infty(A)$ 接近 1 时，误差界不再小，当前精度下可能没有可保证的有效数字。此时问题不一定出在求解器，应考虑缩放、预条件、提高精度或重新建模。

Tip

本页要点： 病态系统会放大输入和舍入扰动，单纯收紧迭代容差未必能恢复有效数字。

3.39 Some Remarks on Condition Numbers

对非奇异矩阵，诱导范数满足

$$\kappa(A) = \|A\| \|A^{-1}\| \geq \|AA^{-1}\| = \|I\| = 1. \quad (3.101)$$

求逆不改变条件数；非零标量缩放中的 $|\alpha|$ 与 $|\alpha|^{-1}$ 相消，因此

$$\kappa(A) = \kappa(A^{-1}), \quad \kappa(\alpha A) = \kappa(A), \quad \alpha \neq 0. \quad (3.102)$$

除上述基于范数的条件数外，还可以定义分量敏感性量

$$\text{cond}(A) = \| |A^{-1}| |A| \|_\infty \quad (3.103)$$

以及依赖具体解的量

$$\text{cond}(A, \mathbf{x}) = \frac{\| |A^{-1}| |A| \mathbf{x} \|_\infty}{\|\mathbf{x}\|_\infty}. \quad (3.104)$$

绝对值按元素理解。由矩阵范数的次乘性，

$$\text{cond}(A, \mathbf{x}) \leq \text{cond}(A) \leq \|A^{-1}\|_\infty \|A\|_\infty = \kappa_\infty(A). \quad (3.105)$$

前两个量保留了矩阵元素之间的大小关系，可能比 $\kappa_\infty(A)$ 小得多。其中 $\text{cond}(A, \mathbf{x})$ 还考虑了具体解的各个分量，适用于按元素控制扰动的误差分析。

条件数大时，小残量不保证小误差，后向稳定的算法也可能给出不准确的解。合适的预条件可以改善迭代问题的性质、加快收敛，但具体效果还取决于所用方法和预条件后的矩阵结构。

Tip

本页要点： 条件数也可以按分量定义，用来描述不同输入分量对输出的敏感程度。

3.40 A Simple Example

考虑单位正方形上的 Poisson 方程 $-\Delta u = f$ ，并给定 Dirichlet 边界条件。每个方向取 n 个内部网格点，网格间距为 $h = 1/(n+1)$ 。五点差分格式为

$$\frac{4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1}}{h^2} = f_{i,j}. \quad (3.106)$$

边界上的 u 已知，将相应项移到右端。按一行接一行的顺序排列 n^2 个内部未知量，得到 $A\mathbf{u} = \mathbf{b}$ ，其中

这里 $\otimes$ 是 Kronecker 积： $I_n \otimes T$ 处理每行内部的左右邻接， $T \otimes I_n$ 处理相邻两行之间的上下邻接。下面的代码直接构造稀疏矩阵；当 $n = 100$ 时， A 为 10000×10000 ，每行至多五个非零元。

```
import numpy as np
from scipy.sparse import diags, eye, kron

n = 100 # 每个方向的内部网格点数
h = 1.0 / (n + 1)
T = diags([-np.ones(n-1), 2*np.ones(n), -np.ones(n-1)],
          offsets=[-1, 0, 1], format="csr")
I = eye(n, format="csr")
A = (kron(I, T, format="csr") + kron(T, I, format="csr")) / h**2
```

为什么网格越细，条件数越大？ 取 $(v_j)_i = \sin(ij\pi/(n+1))$ ，由 $\sin((i-1)\theta) + \sin((i+1)\theta) = 2\cos\theta\sin(i\theta)$ ，可直接验证

$$Tv_j = \lambda_j(T)v_j, \quad \lambda_j(T) = 2 - 2\cos\frac{j\pi}{n+1} = 4\sin^2\frac{j\pi h}{2}, \quad j = 1, \dots, n. \quad (3.108)$$

利用 Kronecker 积的乘法规则， $v_j \otimes v_k$ 是 A 的特征向量，对应特征值 $h^{-2}(\lambda_j(T) + \lambda_k(T))$ 。这 n^2 个向量构成一组基，因而给出了全部特征值。特别地，

$$\lambda_{\min}(A) = \frac{8}{h^2}\sin^2\frac{\pi h}{2}, \quad \lambda_{\max}(A) = \frac{8}{h^2}\cos^2\frac{\pi h}{2}. \quad (3.109)$$

A 实对称正定，由第 3.34 页的条件数公式，

$$\kappa_2(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} = \cot^2\frac{\pi h}{2} \sim \frac{4}{\pi^2 h^2}, \quad h \rightarrow 0. \quad (3.110)$$

因此网格间距减半时，条件数约增大为四倍。上述 $n = 100$ 的例子中， $\kappa_2(A) \approx 4133.64$ 。网格加密降低离散误差，却也使线性系统对扰动更敏感。

为什么对角预条件无效？ 本例中全部对角元相同：

$$D = \text{diag}(A) = \frac{4}{h^2}I_{n^2}, \quad D^{-1}A = \frac{h^2}{4}A, \quad \kappa_2(D^{-1}A) = \kappa_2(A). \quad (3.111)$$

对角预条件只做了标量缩放；由第 3.39 页的结论，它不改变条件数。因此它不能消除本例随网格加密而加剧的病态性；这并不是说 Jacobi 迭代不收敛。

Tip

本页要点： 网格加密会增大离散系统的条件数；本例的对角预条件只是标量缩放，无法改善条件数。

第3.3节 迭代精校方法

3.42 Iterative Refinement Methods

病态系统可能需要高精度计算才能达到要求的精度，但全程使用高精度代价较大。三精度迭代精校（iterative refinement, IR）的做法是：为残量计算、解的更新和校正求解分别选择精度，反复修正已有近似解。

给定非奇异矩阵 A 、右端 b 和初值 x_0 ，设置残量、工作和求解的单位舍入误差 $u_r \leq u \leq u_{\text{ell}}$ 。单位舍入误差越小，表示计算精度越高。第 i 次迭代执行：

(2)

若残量、校正求解和更新都精确，则 $e_i = A^{-1}r_i = x - x_i$ ，一次更新即可得到精确解。实际只能算出近似校正量，因而要重复迭代，直到满足停止条件或达到最大迭代次数。若用低精度 LU 求校正量，因子只需构造一次，之后每轮重复三角求解。

💡 Tip

本页要点： 迭代精校用高精度残量反复修正低精度求解器的误差。

3.43 Theorem 3.3: Accuracy of IR Methods

定理 3.3. 设 $Ax = b$ 、 $x \neq 0$ ，并记

$$d_i = x - x_i, \quad t_i = \frac{\|d_i\|_\infty}{\|x\|_\infty}. \tag{3.113}$$

其中 d_i 表示当前解的真实误差，与算法求得的校正量区分。这里采用 Carson—Higham 型一阶误差分析，以下推导以单步误差模型为前提：存在 $\phi_i \geq 0$ ，使计算校正 $\hat{e}_i$ 满足

$$\frac{\|\hat{e}_i - d_i\|_\infty}{\|x\|_\infty} \leq \phi_i t_i + \tau_r, \quad \tau_r = 4p \operatorname{cond}(A, x) u_r, \tag{3.114}$$

其中 p 是 A 每行非零元个数的最大值，并且

$$\operatorname{cond}(A, x) = \frac{\| |A^{-1}| |A| |x| \|_\infty}{\|x\|_\infty}. \tag{3.115}$$

该算法的收缩因子可写成

(3)

这两个系数可作如下理解。设传给求解器的计算残量为 $\tilde{r}_i$ ，令 $e_i = A^{-1}\tilde{r}_i$ 是该校正方程的精确解， $\hat{e}_i$ 是实际计算的校正量，则

$$\xi_i = \frac{\|\hat{e}_i - e_i\|_\infty}{\|e_i\|_\infty}, \quad \mu_i = \frac{\|A d_i\|_\infty}{\|A\|_\infty \|d_i\|_\infty} \leq 1. \tag{3.117}$$

这些比值用于分母非零的迭代步。 ξ_i 衡量校正方程求解得是否准确； μ_i 衡量当前误差方向经过 A 后被放大的程度，相对于最坏方向有多少。 $\mu_i \leq 1$ 直接来自诱导范数不等式。 ϕ_i 足够小，才能使前一轮误差在校正后缩小。

📌 Note

推导的前提。 下面证明单步模型 (3.114) 如何导出最终精度界；该模型中系数的具体来源还依赖残量计算、精度转换和校正求解的误差分析。这是定理的条件性推导，不是对任意低精度求解器的保证。

工作精度更新满足

$$\mathbf{x}_{i+1} = fl(\mathbf{x}_i + \hat{\mathbf{e}}_i) = \mathbf{x}_i + \hat{\mathbf{e}}_i + \boldsymbol{\delta}_i, \quad (3.118)$$

$$\|\boldsymbol{\delta}_i\|_\infty \leq u \|\mathbf{x}_i + \hat{\mathbf{e}}_i\|_\infty. \quad (3.119)$$

若

$$\bar{\phi} = (1 + u) \sup_i \phi_i < 1, \quad (3.120)$$

则

$$t_i \leq \bar{\phi}^i t_0 + \frac{1 - \bar{\phi}^i}{1 - \bar{\phi}} (u + (1 + u)\tau_r), \quad (3.121)$$

并且

$$\limsup_{i \rightarrow \infty} t_i \leq \frac{u + (1 + u)4p \operatorname{cond}(A, \mathbf{x})u_r}{1 - \bar{\phi}}. \quad (3.122)$$

在单步模型下的推导. 先忽略最后一次工作精度舍入, 令

$$\mathbf{y}_{i+1} = \mathbf{x}_i + \hat{\mathbf{e}}_i. \quad (3.123)$$

因为 $\mathbf{x}_i = \mathbf{x} - \mathbf{d}_i$, 有

$$\mathbf{x} - \mathbf{y}_{i+1} = \mathbf{d}_i - \hat{\mathbf{e}}_i. \quad (3.124)$$

单步误差模型给出

$$\frac{\|\mathbf{x} - \mathbf{y}_{i+1}\|_\infty}{\|\mathbf{x}\|_\infty} \leq \phi_i t_i + \tau_r. \quad (3.125)$$

再由三角不等式,

$$\frac{\|\mathbf{y}_{i+1}\|_\infty}{\|\mathbf{x}\|_\infty} \leq 1 + \phi_i t_i + \tau_r. \quad (3.126)$$

实际更新为 $\mathbf{x}_{i+1} = \mathbf{y}_{i+1} + \boldsymbol{\delta}_i$, 所以

$$\begin{aligned} t_{i+1} &= \frac{\|\mathbf{x} - \mathbf{x}_{i+1}\|_\infty}{\|\mathbf{x}\|_\infty} \\ &\leq \frac{\|\mathbf{x} - \mathbf{y}_{i+1}\|_\infty + \|\boldsymbol{\delta}_i\|_\infty}{\|\mathbf{x}\|_\infty} \\ &\leq \phi_i t_i + \tau_r + u(1 + \phi_i t_i + \tau_r) \\ &= (1 + u)\phi_i t_i + u + (1 + u)\tau_r \\ &\leq \bar{\phi} t_i + \tau, \end{aligned} \quad (3.127)$$

其中 $\tau = u + (1 + u)\tau_r$ 。递推展开为

$$t_i \leq \bar{\phi}^i t_0 + \tau \sum_{j=0}^{i-1} \bar{\phi}^j. \quad (3.128)$$

因为 $\bar{\phi} < 1$ ，几何级数满足

$$\sum_{j=0}^{i-1} \bar{\phi}^j = \frac{1 - \bar{\phi}^i}{1 - \bar{\phi}}, \quad (3.129)$$

这给出有限步界。令 $i \rightarrow \infty$ 即得极限上界。若 $\bar{\phi} \leq 1 - c_0$ ，其中 $c_0 > 0$ 与精度无关，再忽略 uu_r 二阶项，就得到一阶误差界

$$\frac{\|\hat{\mathbf{x}} - \mathbf{x}\|_\infty}{\|\mathbf{x}\|_\infty} \lesssim u + 4p \operatorname{cond}(A, \mathbf{x})u_r. \quad (3.130)$$

这里 $\lesssim$ 隐含的常数包含 $(1 - \bar{\phi})^{-1}$ 。在收缩因子远离 1 的前提下，最终误差的量级主要由工作精度 u 和残量精度 u_r 决定；求解精度 u_{ell} 则通过收缩因子影响能否收敛及收敛速度。

Tip

本页要点： 迭代精校的精度界依赖单步误差模型和收缩条件，不能省略这些前提。

3.44 Different Versions of IR

传统迭代精校在比因子分解更高的精度中计算残量，目标是恢复丢失的有效位。单一精度版本（mono-precision）把各阶段放在同一精度，主要改善稳定性；这里的“单一精度”不特指 binary32。采用低精度求解器的版本则着重降低时间和能耗，校正方程既可以用低精度三角求解，也可以用预条件 GMRES 求解。

选择这些方案时，需要考虑以下权衡：

- 更低的分解精度（更大的 u_{ell} ）可提高吞吐率、降低存储与能耗，但直接三角校正要求 $\kappa(A)u_{\text{ell}}$ 不能太大；
- 更低的 u_r 意味着更高的残量精度，可降低最终误差平台；
- GMRES 校正的计算成本通常高于直接三角校正，但能容忍更不准确的低精度因子。

Tip

本页要点： 分解、残量和校正求解所用的精度不同，计算成本和可达到的精度也随之变化。

3.45 Example: LU-IR Method

LU-IR 使用第 3.42 页的三种精度 $u_r \leq u \leq u_{\text{ell}}$ 。首先在低精度 u_{ell} 中计算一次 LU 分解，再用相同精度求初始解：

$$PA \approx LU, \quad LU\mathbf{x}_0 = P\mathbf{b}. \quad (3.131)$$

P 表示选主元产生的置换；未交换行时 $P = I$ 。第 i 轮在高精度 u_r 中形成残量 $\mathbf{r}_i = \mathbf{b} - A\mathbf{x}_i$ ，再用已经保存的因子，在精度 u_{ell} 中求解

$$LU\mathbf{e}_i = P\mathbf{r}_i, \quad (3.132)$$

最后在工作精度 u 中更新 $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \boldsymbol{e}_i$ 。

主要成本是一次 LU 分解；每轮只需一次矩阵-向量乘和两次三角求解。若 $\kappa(A)u_{\text{ell}}$ 太大，或分解中出现较大的主元增长，低精度因子可能无法提供足够准确的校正。允许精校修正分解误差后，有时可以采用选主元限制较少、速度较快的分解，但仍需满足第 3.43 页的收缩条件。

 **Tip**

本页要点： LU-IR 重用一次低精度分解，每轮用高精度残量求校正；因子太不准确时可能停滞。

3.46 Example: GMRES-IR Method

GMRES-IR 同样先在精度 u_{ell} 中计算 $PA \approx LU$ ，并由 $LU\boldsymbol{x}_0 = P\boldsymbol{b}$ 得到初值。随后以 $M = P^T LU \approx A$ 为预条件器，用 GMRES 近似求解每轮的校正方程

$$A\boldsymbol{e}_i = \boldsymbol{r}_i. \tag{3.133}$$

这里 $\boldsymbol{e}_i$ 是传入残量所对应的精确校正量，GMRES 实际返回的是 $\hat{\boldsymbol{e}}_i$ 。当 $\boldsymbol{e}_i \neq 0$ 时，记其相对求解误差为 $\xi_i = \|\boldsymbol{e}_i - \hat{\boldsymbol{e}}_i\|_\infty / \|\boldsymbol{e}_i\|_\infty = \theta_i u$ 。其中 θ_i 是以工作精度 u 为单位衡量的误差系数，不要求它小于一；真正控制校正质量的是乘积 $\theta_i u$ 。

由 $M^{-1} = U^{-1}L^{-1}P$ ，右预条件形式为

$$AU^{-1}L^{-1}P\boldsymbol{z}_i = \boldsymbol{r}_i, \quad \boldsymbol{e}_i = U^{-1}L^{-1}P\boldsymbol{z}_i. \tag{3.134}$$

本页的五种精度各有用途：

单位舍入误差	用途
u_{ell}	LU 分解和初始三角求解
u_r	外层残量 $\boldsymbol{b} - A\boldsymbol{x}_i$
u_g	GMRES 迭代
u_p	GMRES 内应用预条件器，即两次三角求解
u	更新、存储外层近似解

Krylov 迭代可以补偿低精度因子的误差，代价是每轮增加矩阵-向量乘、预条件应用和正交化。相应的充分小量条件是

(4)

这是相应舍入误差模型和校正求解条件下的充分条件，不能脱离这些前提单独判定收敛。它说明：分解精度较低时，可以通过提高 GMRES 和预条件应用的精度来改善收敛；收敛后的误差量级见第 3.43 页式 (3.130)，仍主要由 u 和 u_r 决定。

要求 $\xi_i \leq 1$ ，只是要求校正求解误差不超过校正量本身，单凭这一条件还不能保证收敛；还需满足第 3.43 页式 (3.120) 的严格收缩条件。忽略残量和更新中的舍入误差时， $e_i = x - x_i$ ，更新后的误差为 $x - x_{i+1} = e_i - \hat{e}_i$ ，所以 ξ_i 就是该步的误差缩小比例；等一时并未缩小。

 Tip

本页要点： GMRES-IR 用低精度三角因子作预条件器，再以高精度残量进行迭代精校。

3.47 Accuracy of IR Methods

这里采用的五精度配置为 $(u_r, u_g, u_p, u, u_{\text{ell}}) = (2^{-53}, 2^{-24}, 2^{-53}, 2^{-24}, 2^{-24})$ ：残量计算和预条件器应用使用双精度，GMRES 迭代、解的更新以及 LU 分解使用单精度。低精度 LU 因子可以在应用时转成双精度完成三角求解；这减少本次求解中的舍入误差，但不会恢复分解时已经丢失的精度。

比较 LU-IR 与 GMRES-IR 在不同条件数和精度组合下的前向误差时，要区分三个现象：

1. 初期误差按近似几何速度下降，对应 Theorem 3.3 ([03矩阵计算基础.pdf](#)，第 3.43 页) 中的 $\bar{\phi}^i t_0$ ；
2. 误差到达由 u 和 u_r 决定的平台后不再下降；
3. 当低精度因子过差时，LU-IR 可能不收敛，而 GMRES-IR 仍能通过 Krylov 校正下降。

比较曲线时还要考虑每轮成本，不能仅凭误差下降认定 GMRES-IR 的总体性能更好。对条件适中的问题，LU-IR 每轮成本低；直接校正收敛很慢或失败时，可以考虑 GMRES 校正。

GMRES-IR 用于 HPL-AI。另一种做法是用低精度因子预条件后，直接对原方程 $Ax = b$ 运行 GMRES；两者都利用 Krylov 迭代补偿因子误差，但外层精校另外显式重算高精度残量。

两个算例中的矩阵阶数均为 100，条件数分别为 10^7 和 10^8 。前者两种方法都能降低误差，后者 LU-IR 的误差不降反升，而 GMRES-IR 仍能下降。这说明校正求解是否足够准确会直接影响外层精校；不能仅由使用了哪种存储精度判断收敛。算例来源为 Carson 与 Higham, “A New Analysis of Iterative Refinement and Its Application to Accurate Solution of Ill-Conditioned Sparse Linear Systems”, *SIAM Journal on Scientific Computing*, 39(6), 2017, A2834 – A2856。

 Tip

本页要点： 比较 LU-IR 与 GMRES-IR 时，要同时看误差下降、最终精度和每轮成本。

3.48 MP Methods for Sparse Problems

混合精度对稠密分解的收益较容易预测，因为运算规则且计算密集。稀疏 LU 更复杂：

- 基于稀疏结构的重排序和符号分析主要处理图与索引数据，降低浮点精度对这些阶段几乎没有直接加速作用；
- 数值分解中的 fill-in 模式和动态选主元可能随舍入变化；
- 缩放前后的中间量可能落入次正规区；
- 稀疏三角求解常受内存带宽和依赖关系限制，降低精度未必按峰值算力比例加速。

Note

备注：缩放后仍须留意次正规数。

即使所有非零矩阵条目都已缩放到单精度正规数范围，消元更新仍可能生成很小的填充项，并在后续消元中传播，甚至进入次正规数范围。性能影响取决于实际运算精度和硬件；若硬件对次正规运算很慢，性能损失可能抵消低精度收益。因此应同时监测数值范围、fill-in、主元增长和实际硬件计数器。

相关实验见 Zounon、Higham、Lucas 与 Tisseur, “Performance impact of precision reduction in sparse linear systems solvers”, *PeerJ Computer Science*, 8, 2022, e778。

Tip

本页要点： 稀疏混合精度还要监测 fill-in、主元增长和次正规数，不能只看存储位宽。

3.49 Reading and Thinking

本页推荐 Higham 与 Mary 的综述 “Mixed precision algorithms in numerical linear algebra”, *Acta Numerica*, 31 (2022), pp. 347 – 414。结合本章，可用以下问题检查一个实际程序：

1. 目标精度由离散误差、条件数还是机器精度决定？
2. 残量在哪个精度计算，停止准则测量的是前向误差还是后向误差？
3. 编译器是否重排归约、启用 FMA 或 flush-to-zero？
4. 低精度阶段是否经过缩放，是否可能溢出或进入次正规区？
5. 正交化是否出现 $\|I - Q^H Q\|$ 增长，是否需要再正交化？
6. 混合精度节省的是算力、内存带宽还是通信量，端到端计算的主要瓶颈是否因此改变？

这些问题应通过残量历史、条件数估计、正交性度量、编译报告和不同精度对照实验回答，而不能只看一次运行的最终数值。

Tip

本页要点： 用残量、条件数、正交性和不同精度的对照实验检查程序，不能只看最终数值。

第3.4节 矩阵计算基本工具

3.51 Orthogonalization

给定线性无关向量 $\mathbf{a}_1, \dots, \mathbf{a}_n$ ，正交化要构造 $\mathbf{q}_1, \dots, \mathbf{q}_n$ ，使

$$\mathbf{q}_i^H \mathbf{q}_j = \delta_{ij}, \quad \text{span}\{\mathbf{q}_1, \dots, \mathbf{q}_n\} = \text{span}\{\mathbf{a}_1, \dots, \mathbf{a}_n\}. \quad (3.136)$$

正交基有三项直接好处：坐标由内积 $c_i = \mathbf{q}_i^H \mathbf{x}$ 得到；投影写成 $Q Q^H \mathbf{x}$ ；二范数满足 Parseval 关系 $\|Q \mathbf{c}\|_2 = \|\mathbf{c}\|_2$ 。它是 QR、最小二乘、Krylov 方法和对称特征值算法的共同基础。

若 Q 是方阵，正交变换还满足 $\|Q \mathbf{e}\|_2 = \|\mathbf{e}\|_2$ ：已有误差的二范数不会被变换放大。这是后面采用正交变换进行消元的主要理由。

Tip

本页要点： 正交化构造标准正交基并保持所张子空间，是 QR、Arnoldi 和 Lanczos 的共同基础。

3.52 QR Decomposition

对 $A \in \mathbb{R}^{m \times n}$, $m \geq n$, 完整 QR 分解为

(5)

当 $m > n$ 时, R 是上梯形矩阵。只保留 Q 的前 n 列, 就得到薄 QR 分解 $A = Q_1 R_1$; 其中 $Q_1 \in \mathbb{R}^{m \times n}$ 满足 $Q_1^T Q_1 = I_n$ 。

三种构造方法各有侧重。Gram-Schmidt 逐列减去已有方向的投影, 直接体现正交投影的构造; 经典版本在有限精度下容易丢失正交性。Householder 反射一次消去一列的尾部, 是稠密矩阵计算的常用方法。Givens 旋转一次只处理两个坐标, 适合有特殊结构的矩阵和局部更新。后面将分别说明它们的计算过程。

Tip

本页要点： QR 把矩阵分成正交方向与上三角系数, 完整和薄分解的区别在于保留多少个正交基向量。

3.53 Theorem 3.4: Properties of QR Decomposition

定理 3.4. 设 $A = [\mathbf{a}_1, \dots, \mathbf{a}_n] \in \mathbb{R}^{m \times n}$ 列满秩, 且 $A = QR$ 是 full QR 分解。则对 $k = 1, \dots, n$,

$$\text{span}\{\mathbf{a}_1, \dots, \mathbf{a}_k\} = \text{span}\{\mathbf{q}_1, \dots, \mathbf{q}_k\}. \quad (3.138)$$

若

$$Q = [Q_1, Q_2], \quad Q_1 = Q(:, 1:n), \quad Q_2 = Q(:, n+1:m), \quad (3.139)$$

则

$$\text{range}(A) = \text{range}(Q_1), \quad \text{null}(A^T) = \text{range}(Q_2). \quad (3.140)$$

证明. 因为 R 为上梯形, 第 j 列满足

$$\mathbf{a}_j = \sum_{i=1}^j r_{ij} \mathbf{q}_i. \quad (3.141)$$

所以 $\text{span}\{\mathbf{a}_1, \dots, \mathbf{a}_k\} \subseteq \text{span}\{\mathbf{q}_1, \dots, \mathbf{q}_k\}$ 。又因为 A 列满秩、 Q 的列正交, 两边的维数相同:

$$\dim \text{span}\{\mathbf{a}_1, \dots, \mathbf{a}_k\} = k = \dim \text{span}\{\mathbf{q}_1, \dots, \mathbf{q}_k\}. \quad (3.142)$$

包含关系加上维数相等, 即得式 (3.138)。

取 $k = n$ 得 $\text{range}(A) = \text{range}(Q_1)$ 。又因 Q 正交, Q_2 的列与 Q_1 的列正交, 所以

$$\text{range}(Q_2) = \text{range}(Q_1)^\perp = \text{range}(A)^\perp = \text{null}(A^T). \quad (3.143)$$

证毕。

将 R 分成上部 $R_1 = R(1:n, 1:n)$ 和下部零块, 便有 $A = [Q_1, Q_2][R_1^T, 0]^T = Q_1 R_1$ 。因此薄 QR 已包含表示 A 所需的全部列方向, Q_2 只用于补全整个空间的正交基。

Tip

本页要点: 列满秩 QR 的前几列张成原矩阵的列空间, 后几列张成左零空间。

3.54 Theorem 3.5: Reduced QR decomposition

定理 3.5. 若 $A \in \mathbb{R}^{m \times n}$ 列满秩, 则分解

$$A = Q_1 R_1, \quad Q_1^T Q_1 = I_n, \quad (3.144)$$

在要求 R_1 上三角且对角元为正时唯一。此外, R_1 是 $A^T A$ 的正对角上三角 Cholesky 因子。

证明. 依次用 Householder 反射消去各列对角线以下的元素, 即可构造 QR; 每步只作用于尚未处理的行, 不改变前面已经消出的零。取前 n 列得到薄 QR; 列满秩保证 R_1 的对角元非零。若某个对角元为负, 同时改变 Q_1 对应列与 R_1 对应行的符号, 即可使其为正而保持乘积不变。

先说明与 Cholesky 分解的关系。由 $A = Q_1 R_1$,

$$A^T A = R_1^T Q_1^T Q_1 R_1 = R_1^T R_1. \quad (3.145)$$

因为 A 列满秩, 对任意非零 x 有

$$x^T A^T A x = \|Ax\|_2^2 > 0, \quad (3.146)$$

所以 $A^T A$ 对称正定, R_1 就是其正对角上三角 Cholesky 因子, 记为 G , 满足 $G^T G = A^T A$ 。确定 R_1 后,

$$Q_1 = A R_1^{-1} \quad (3.147)$$

随之确定。

下面不借助后续章节的 Cholesky 定理, 直接证明唯一性。若另有 $A = \tilde{Q}_1 \tilde{R}_1$, 令 $W = \tilde{R}_1 R_1^{-1}$ 。由 $Q_1 = \tilde{Q}_1 W$ 得

$$I_n = Q_1^T Q_1 = W^T \tilde{Q}_1^T \tilde{Q}_1 W = W^T W. \quad (3.148)$$

W 是上三角矩阵, W^{-1} 也是上三角矩阵; 但正交性又给出 $W^{-1} = W^T$, 所以 W 必为对角矩阵。其对角元由正交性只能取 ± 1 , 而 $w_{ii} = \tilde{r}_{ii}/r_{ii} > 0$, 故 $W = I_n$ 。于是 $\tilde{R}_1 = R_1$ 、 $\tilde{Q}_1 = Q_1$, 唯一性得证。

不必先求完整 QR 才能得到薄 QR。第 3.56 - 3.57 页的 Gram-Schmidt 直接构造 n 个正交列; Householder 方法也可以只输出前 n 列或保留隐式反射。通过 $A^T A$ 求 Cholesky 因子在精确算术中同样可行, 但有限精度下会把二范数条件数平方, 不能据此认为它与 Householder QR 同样稳定。

Tip

本页要点: 正对角约定消除了 reduced QR 因符号变化产生的不唯一性。

3.55 Gram - Schmidt Orthogonalization

Gram - Schmidt 的思路是：从当前向量中减去它在已有正交方向上的投影，再将剩余部分归一化。

先看两个线性无关的向量 $\mathbf{a}_1, \mathbf{a}_2$ 。将第一个向量归一化，令 $\mathbf{q}_1 = \mathbf{a}_1 / \|\mathbf{a}_1\|_2$ ，再从第二个向量中减去沿 $\mathbf{q}_1$ 的分量：

$$\mathbf{w}_2 = \mathbf{a}_2 - r_{12}\mathbf{q}_1, \quad 0 = (\mathbf{q}_1, \mathbf{w}_2) = (\mathbf{q}_1, \mathbf{a}_2) - r_{12}. \quad (3.149)$$

因此 $r_{12} = (\mathbf{q}_1, \mathbf{a}_2) = \mathbf{q}_1^T \mathbf{a}_2$ ，即投影系数。线性无关保证 $\mathbf{w}_2 \neq \mathbf{0}$ ，于是可以令

$$r_{22} = \|\mathbf{w}_2\|_2, \quad \mathbf{q}_2 = \frac{\mathbf{w}_2}{r_{22}}, \quad \mathbf{a}_2 = r_{12}\mathbf{q}_1 + r_{22}\mathbf{q}_2. \quad (3.150)$$

这样 $\mathbf{q}_2$ 与 $\mathbf{q}_1$ 正交且长度为一，两个新向量仍张成原来的子空间。

对列满秩矩阵 $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_n] \in \mathbb{R}^{m \times n}$ ， $m \geq n$ ，从 $\mathbf{q}_1 = \mathbf{a}_1 / \|\mathbf{a}_1\|_2$ 开始，依次作同样的投影与归一化：

$$\begin{aligned} r_{ij} &= \mathbf{q}_i^T \mathbf{a}_j, \quad i < j, \\ \mathbf{w}_j &= \mathbf{a}_j - \sum_{i=1}^{j-1} r_{ij}\mathbf{q}_i, \\ r_{jj} &= \|\mathbf{w}_j\|_2, \quad \mathbf{q}_j = \mathbf{w}_j / r_{jj}. \end{aligned} \quad (3.151)$$

若 $\mathbf{w}_j = \mathbf{0}$ ，则 $\mathbf{a}_j$ 属于前 $j-1$ 列的张成空间，与列满秩矛盾，因此精确算术下每一步都能归一化。把 $\mathbf{a}_j = \sum_{i=1}^j r_{ij}\mathbf{q}_i$ 按列排列，便得到薄 QR 分解：

(6)

只需构造 n 个标准正交向量，不必扩充为 $\mathbb{R}^m$ 的整组基。下一页把这一过程写成经典 Gram - Schmidt 算法。

Tip

本页要点： 减去旧方向上的投影，再归一化，就能逐列构造薄 QR 分解。

3.56 Classical Gram - Schmidt Method

在第 j 步，Classical Gram - Schmidt 先用原始列 $\mathbf{a}_j$ 计算全部投影系数：

$$r_{ij} = \mathbf{q}_i^H \mathbf{a}_j, \quad i = 1, \dots, j-1, \quad (3.153)$$

再一次性减去这些投影：

$$\mathbf{w}_j = \mathbf{a}_j - \sum_{i=1}^{j-1} r_{ij}\mathbf{q}_i. \quad (3.154)$$

最后令

$$r_{jj} = \|\mathbf{w}_j\|_2, \quad \mathbf{q}_j = \frac{\mathbf{w}_j}{r_{jj}}. \quad (3.155)$$

精确算术中,

$$\mathbf{q}_i^H \mathbf{w}_j = \mathbf{q}_i^H \mathbf{a}_j - r_{ij} = 0. \quad (3.156)$$

列满秩保证 $\mathbf{w}_j \neq 0$, 否则不能用其范数归一化。有限精度中, 若 $\mathbf{a}_j$ 几乎落在已有子空间内, $\mathbf{w}_j$ 是多个大投影相减后的很小余量; 投影系数的舍入误差会与它同量级, 导致 $\mathbf{Q}^H \mathbf{Q} - \mathbf{I}$ 明显增大。

Tip

本页要点: Classical Gram–Schmidt 先计算全部投影, 易受舍入误差影响。

3.57 Modified Gram–Schmidt Method

Modified Gram–Schmidt 在每次投影后立即更新工作向量:

$$\begin{aligned} \mathbf{w}_j^{(0)} &= \mathbf{a}_j, \\ r_{ij} &= \mathbf{q}_i^H \mathbf{w}_j^{(i-1)}, \\ \mathbf{w}_j^{(i)} &= \mathbf{w}_j^{(i-1)} - r_{ij} \mathbf{q}_i, \quad i = 1, \dots, j-1. \end{aligned} \quad (3.157)$$

然后

$$r_{jj} = \|\mathbf{w}_j^{(j-1)}\|_2, \quad \mathbf{q}_j = \mathbf{w}_j^{(j-1)} / r_{jj}. \quad (3.158)$$

精确算术中, 已有列互相正交, 所以减去前面的投影不会改变与 $\mathbf{q}_i$ 的内积, 仍有 $r_{ij} = \mathbf{q}_i^H \mathbf{a}_j$ 。因此 CGS 与 MGS 产生同一投影 $(\mathbf{I} - \mathbf{Q}_{j-1} \mathbf{Q}_{j-1}^H) \mathbf{a}_j$ 。有限精度中, MGS 每一步都针对当前余量消去一个方向, 通常把正交性损失从与 $\kappa(A)^2 u$ 同阶改善到与 $\kappa(A)u$ 同阶。若列近线性相关, 仍应再正交化或使用 Householder QR。

Tip

本页要点: Modified Gram–Schmidt 每次投影后立即更新向量, 正交性通常更好。

3.58 QR Decomposition via Householder

对 $A \in \mathbb{R}^{m \times n}$, $m \geq n$, 第 k 步在尾部子向量 $A_{k:m,k}$ 上构造 Householder 反射 $\widetilde{H}_k$, 把第 k 列对角元以下元素消为零。嵌入后的反射为

$$H_k = \text{diag}(\mathbf{I}_{k-1}, \widetilde{H}_k). \quad (3.159)$$

连续作用得到

$$R = H_n \cdots H_2 H_1 A. \quad (3.160)$$

由于 $H_k^T = H_k = H_k^{-1}$,

$$A = QR, \quad Q = H_1 H_2 \cdots H_n, \quad Q^T = H_n \cdots H_2 H_1. \quad (3.161)$$

④ Note

备注：Householder QR 的存储与分块实现。

每一步消去当前列中对角线以下的元素。实现只存每个反射向量在 A 的下三角位置，并用块反射把多个秩一更新组织成矩阵-矩阵乘法，以提高缓存复用和并行效率。

按一次加法或乘法计一次 FLOP，形成 R 并保存反射向量的主要工作量为

$$\sum_{k=1}^n 4(m-k+1)(n-k) = 2mn^2 - \frac{2}{3}n^3 + O(mn + n^2). \quad (3.162)$$

每一步的内积和秩一更新各需约 $2(m-k+1)(n-k)$ 次运算。若显式形成 Q ，需另计工作量：形成薄 Q_1 为 $O(mn^2)$ ，形成完整 $m \times m$ 的 Q 可达 $O(m^2n)$ 。

LAPACK 的 DGEQRF 返回上三角因子和反射向量；需要显式 Q 时再调用 DORGQR，若只需 $Q^T b$ ，可直接使用反射向量作用于 b 。

💡 Tip

本页要点： Householder QR 每步只处理当前尾部列，逐步把矩阵化为上三角。

3.59 Householder Transformation, Revisited

给定实单位向量 u ，Householder 矩阵为

$$H = I - 2uu^T, \quad \|u\|_2 = 1. \quad (3.163)$$

它满足

$$H^T = H, \quad H^T H = (I - 2uu^T)^2 = I, \quad (3.164)$$

所以既对称又正交。若 $y \perp u$ ，则

$$H(y + au) = y - au. \quad (3.165)$$

这说明 H 保持超平面 $u^\perp$ 上的分量，反转沿 u 的分量，几何上是关于该超平面的反射。

要把非零 x 映到第一坐标轴，取

$$a = \begin{cases} \|x\|_2, & x_1 \leq 0, \\ -\|x\|_2, & x_1 > 0, \end{cases} \quad u = \frac{x - ae_1}{\|x - ae_1\|_2}. \quad (3.166)$$

因为 $\|x\|_2 = |a|$ ，反射交换 x 与 ae_1 ，故 $Hx = ae_1$ 。当 $x_1 \neq 0$ 时，选择与它相反符号的 a 可避免 $x_1 - a$ 中的相消； $x_1 = 0$ 时取正号，与下一页算法一致。

💡 Tip

本页要点： Householder 反射既正交又对称，可稳定地把向量或矩阵列对齐到坐标轴。

3.60 Practical Householder Method

实际计算不显式形成反射矩阵，而保存一个向量和一个标量：

$$H = I - \beta \mathbf{v} \mathbf{v}^T. \quad (3.167)$$

若 $\mathbf{x} = \mathbf{0}$ ，取 $\beta = 0$ 、 $\mathbf{v} = \mathbf{e}_1$ ，于是 $H = I$ 。对非零向量，先按最大分量缩放。为区分缩放前后的量，记

$$\sigma = \max_i |x_i|, \quad \tilde{\mathbf{x}} = \mathbf{x}/\sigma, \quad \mu = \|\tilde{\mathbf{x}}\|_2. \quad (3.168)$$

此时各分量的绝对值不超过 1，且 $1 \leq \mu \leq \sqrt{n}$ ，避免了直接对原始大分量平方造成的中间溢出。缩放只改变长度，不改变需要消去的方向。

先构造未归一化的反射向量 $\mathbf{h}$ 。取

$$\mathbf{h}_1 = \begin{cases} \tilde{x}_1 - \mu, & \tilde{x}_1 \leq 0, \\ \tilde{x}_1 + \mu, & \tilde{x}_1 > 0, \end{cases} \quad \mathbf{h} = (h_1, \tilde{x}_2, \dots, \tilde{x}_n)^T. \quad (3.169)$$

两个分支都使 $|h_1| = |\tilde{x}_1| + \mu$ ，不会发生相近数相减。令 $\mathbf{v} = \mathbf{h}/h_1$ ，便有 $v_1 = 1$ 。由 $\sum_{i=2}^n \tilde{x}_i^2 = \mu^2 - \tilde{x}_1^2$ ，得到

$$\begin{aligned} \mathbf{h}^T \mathbf{h} &= (|\tilde{x}_1| + \mu)^2 + \mu^2 - \tilde{x}_1^2 = 2\mu(\mu + |\tilde{x}_1|), \\ \beta &= \frac{2}{\mathbf{v}^T \mathbf{v}} = \frac{2h_1^2}{\mathbf{h}^T \mathbf{h}} = 1 + \frac{|\tilde{x}_1|}{\mu}. \end{aligned} \quad (3.170)$$

因此 $1 \leq \beta \leq 2$ ，且 $H\mathbf{x} = \sigma H\tilde{\mathbf{x}} = \pm \|\mathbf{x}\|_2 \mathbf{e}_1$ 。这里的 $\sigma\mu = \|\mathbf{x}\|_2$ 只用于解释结果，构造 H 时不必形成这个乘积。

将反射作用于矩阵，只需一次矩阵-向量乘和一次秩一更新：

$$\mathbf{w} = \beta A^T \mathbf{v}, \quad HA = A - \mathbf{v} \mathbf{w}^T. \quad (3.171)$$

向量首分量固定为 1，实际只需保存其尾部和 β 。

Tip

本页要点： 先缩放，再选择避免相消的符号，只存反射向量和标量即可实施 Householder 消元。

3.61 QR Decomposition via Givens

Givens QR 逐个消去 a_{ij} ($i > j$)。用 G_k 表示第 k 个旋转矩阵，每一步左乘 G_k^T 完成消元。经过 L 次旋转，得到

$$R = G_L^T \cdots G_2^T G_1^T A, \quad Q^T = G_L^T \cdots G_2^T G_1^T, \quad Q = G_1 G_2 \cdots G_L. \quad (3.172)$$

例如，对一个 4×3 矩阵，第一列依次用 $G_{3,4}^T, G_{2,3}^T, G_{1,2}^T$ 从下往上消元；第二列用 $(G'_{3,4})^T, (G'_{2,3})^T$ ，第三列用 $(G''_{3,4})^T$ 。于是 $Q = G_{3,4} G_{2,3} G_{1,2} G'_{3,4} G'_{2,3} G''_{3,4}$ 。下标表示作用的两行，撇号区分同一行对上重新计算的旋转，不表示转置。

旋转次序必须与目标稀疏结构协调：消去后不能再通过后续旋转把已为零的元素填回。对稠密矩阵，Householder 通常需要更少的运算；对上 Hessenberg 矩阵，每列只需一个 Givens；对稀疏或在线加入新行的 QR，Givens 可局部更新而不必更新整列。

一次旋转更新两行中的一个列位置，需要四次乘法和两次加法。若只计算 R ，总工作量的主项为 $3mn^2 - n^3$ 。若每次旋转还更新完整 Q 的 m 个列位置，额外工作量为 $6m \sum_{j=1}^n (m-j) = 6m^2n - 3mn^2 - 3mn$ ；累计完整正交矩阵的工作量主项为 $6m^2n - 3mn^2$ 。

Tip

本页要点： Givens QR 通过一系列局部旋转逐个消去下三角元素。

3.62 Givens Transformation, Revisited

Givens 旋转只作用于第 i 、 k 个坐标。采用

$$G_{i,k}(\Phi) = \begin{bmatrix} 1 & & & \\ & c & \cdots & s \\ & \vdots & I & \vdots \\ & -s & \cdots & c \\ & & & & 1 \end{bmatrix}, \quad c^2 + s^2 = 1, \quad (3.173)$$

其中 $c = \cos \Phi$ 、 $s = \sin \Phi$ ，二维非平凡块是

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix}. \quad (3.174)$$

按第 3.63 页的约定，消元时施加的是 $G_{i,k}(\Phi)^T$ ，所以

$$\begin{bmatrix} y_i \\ y_k \end{bmatrix} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} x_i \\ x_k \end{bmatrix}. \quad (3.175)$$

为计算 $r = \sqrt{x_i^2 + x_k^2}$ ，令 $M = \max(|x_i|, |x_k|)$ 、 $m = \min(|x_i|, |x_k|)$ ，按较大分量缩放：

$$r = \begin{cases} 0, & M = 0, \\ M\sqrt{1 + (m/M)^2}, & M > 0. \end{cases} \quad (3.176)$$

这样只对不超过 1 的比值平方，避免直接计算 x_i^2, x_k^2 引起的中间溢出。若 $r = 0$ ，取 $c = 1, s = 0$ ；否则取 $c = x_i/r$ 、 $s = -x_k/r$ 。代入可得 $y_i = r$ 、 $y_k = 0$ 。这里假设 r 本身没有超出浮点表示范围。

被动旋转保持向量不动，将坐标轴逆时针转过 Φ ；新坐标为 $x'_i = cx_i + sx_k$ 、 $x'_k = -sx_i + cx_k$ ，即 $\mathbf{x}' = G_{i,k}(\Phi)\mathbf{x}$ 。消元时使用的是其转置 $G_{i,k}(\Phi)^T$ ，因此参数中的正负号须与式 (3.175) 一致。

Tip

本页要点： Givens 旋转只改变两个坐标，适合逐元素消去并保持其余分量不变。

3.63 Representing a Givens Matrix

上一页通过缩放计算 $r = \sqrt{x_i^2 + x_k^2}$ 。本页直接用分量的比值计算旋转参数，不必先形成 r 。先处理 $x_k = 0$ ，此时已无须消元，取 $c = 1, s = 0$ 。其余情形，若 $|x_k| > |x_i|$ ，令

$$\tau = -\frac{x_i}{x_k}, \quad s = \frac{1}{\sqrt{1+\tau^2}}, \quad c = s\tau; \quad (3.177)$$

若 $|x_i| \geq |x_k|$, 令

$$\tau = -\frac{x_k}{x_i}, \quad c = \frac{1}{\sqrt{1+\tau^2}}, \quad s = c\tau. \quad (3.178)$$

两种分支都只对绝对值不超过 1 的比值平方, 从而避免中间量过大。它们都使 $sx_i + cx_k = 0$, 但保留分量的符号不一定与上一页得到的正数 r 相同; 这不影响消元。

Givens 每次只在局部消元, 其余行不变, 适合稀疏矩阵、上 Hessenberg 矩阵和动态 QR 更新; 缺点是消去一列通常需要多个旋转。

Tip

本页要点: 用较小分量与较大分量的比值计算 Givens 参数, 可避免直接平方造成的中间溢出。

3.64 Reducing Cost of Givens Transformation

传统 Givens 需要计算平方根。现代处理器虽有专用平方根指令, 延迟仍可能明显高于加法和乘法; 例如, Skylake 上部分平方根指令的延迟约为 13 - 19 个周期。使用 `-ffast-math` 等选项还可能改变浮点运算语义, 不能只比较速度。

快速 Givens 把列的方向和长度分开保存。设矩阵 M 满足

$$M^T M = D, \quad D = \text{diag}(d_1, \dots, d_n), \quad d_i > 0. \quad (3.179)$$

则

$$Q = MD^{-1/2} \quad (3.180)$$

满足 $Q^T Q = D^{-1/2} M^T M D^{-1/2} = I$ 。其中 d_i 是第 i 列的长度平方, $D^{-1/2}$ 负责列归一化。因此可用 M, D 隐式表示 Q , 把平方根推迟到确实需要标准正交矩阵时。**连续更新仍需保持这一关系**, 具体过程见第 3.66 页。

Tip

本页要点: 快速 Givens 单独维护对角缩放, 把平方根推迟到需要标准正交矩阵时再计算。

3.65 Fast Givens Transformation

设

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}. \quad (3.181)$$

考虑正对角矩阵 D 和变换矩阵 M_1 :

$$D = \begin{bmatrix} d_1 & 0 \\ 0 & d_2 \end{bmatrix}, \quad M_1 = \begin{bmatrix} \beta & 1 \\ 1 & \alpha \end{bmatrix}, \quad d_1, d_2 > 0. \quad (3.182)$$

要求第二个分量为零, 得到

$$x_1 + \alpha x_2 = 0, \quad \alpha = -\frac{x_1}{x_2} \quad (x_2 \neq 0). \quad (3.183)$$

再要求 $M_1^T D M_1$ 的非对角元为零:

$$d_1 \beta + d_2 \alpha = 0, \quad \beta = -\alpha \frac{d_2}{d_1}. \quad (3.184)$$

定义 $\gamma = -\alpha\beta = (d_2/d_1)(x_1/x_2)^2$, 便有

$$M_1^T \mathbf{x} = \begin{bmatrix} x_2(1+\gamma) \\ 0 \end{bmatrix}, \quad (3.185)$$

同时, 缩放矩阵按下式更新:

$$M_1^T D M_1 = \begin{bmatrix} d_2(1+\gamma) & 0 \\ 0 & d_1(1+\gamma) \end{bmatrix} = D_1, \quad (3.186)$$

由于 $\gamma \geq 0$, D_1 仍为正对角矩阵。

④ Note

备注: 为什么要保持正对角结构。 $M_1^T \mathbf{x}$ 的第二个分量为零, 完成了消元; $M_1^T D M_1 = D_1$ 则记录消元后的缩放, 并保证这个变换仍能对应一个正交变换。

M_1 本身一般不正交, 但令 $J = D^{1/2} M_1 D_1^{-1/2}$, 就有 $J^T J = D_1^{-1/2} (M_1^T D M_1) D_1^{-1/2} = I$ 。因此配合前后的对角缩放, 便能恢复正交变换。

因为 D_1 仍为对角矩阵, 每步只需更新两个缩放系数 $d'_1 = d_2(1+\gamma)$ 、 $d'_2 = d_1(1+\gamma)$, 下一步可继续使用相同结构。实际消元时不必形成 J , 平方根可推迟到最后需要归一化时再计算; 连续变换的具体过程见第 3.66 页。

这里 M_1 恰好对称, 因此 $M_1 \mathbf{x} = M_1^T \mathbf{x}$ 。另一种参数化通常不对称:

$$M'_1 = \begin{bmatrix} 1 & \alpha \\ \beta & 1 \end{bmatrix}, \quad \alpha = -\frac{x_2}{x_1}, \quad \beta = -\alpha \frac{d_1}{d_2} \quad (x_1 \neq 0). \quad (3.187)$$

为了与第 3.66 页的连续变换一致, 此时消元使用 $M_1'^T$ 。令 $\gamma = -\alpha\beta$, 直接相乘可得

$$M_1'^T \mathbf{x} = \begin{bmatrix} x_1(1+\gamma) \\ 0 \end{bmatrix}, \quad (3.188)$$

$$M_1'^T D M'_1 = \text{diag}(d_1(1+\gamma), d_2(1+\gamma)).$$

M_1 和 M'_1 是两种便于计算的常用形式, 并非仅有的选择; 对其各列作非零缩放, 还可得到满足相同消元与正对角结构要求的其他形式。

两种形式作用于一个二维向量都只需两次乘法和两次加法, 不需要平方根。若 $x_2 = 0$, 直接跳过消元; 否则根据分量和缩放量选择分支, 并在必要时重新缩放, 避免参数过大。

④ Note

备注：省去平方根可能导致数值不稳定。 虽然 M_1 一般不正交，但 $M_1^T D M_1 = D_1$ 保证 $J = D^{1/2} M_1 D_1^{-1/2}$ 正交。因此，Fast Givens 在精确算术下仍对应正交变换，只是用对角缩放隐式保存归一化信息。

这一等价关系不能直接保证浮点稳定。例如，取 $D = I$ 、 $\mathbf{x} = (1, 10^{-200})^T$ ，若固定采用第一种形式，则 $\alpha = -10^{200}$ 、 $\beta = 10^{200}$ ，计算 $\gamma = -\alpha\beta = 10^{400}$ 会使双精度数溢出，而对应正交变换的参数仍有界。

实现时应根据分量及缩放量选择分支，避免直接计算过大的比值或平方，并在必要时同步重新缩放数据与 D 。本页给出的是代数构造；其数值稳定性还需结合具体分支、缩放策略和舍入误差分析判断。

💡 Tip

本页要点： Fast Givens 用对角缩放推迟平方根计算；可靠实现还需安全的分支选择与缩放控制。

3.66 Scaled Transformation for QR

若一步 fast Givens 更新满足

$$M_i^T D_{i-1} M_i = D_i, \quad (3.189)$$

对应的标准正交变换为

$$J_i = D_{i-1}^{1/2} M_i D_i^{-1/2}. \quad (3.190)$$

验证正交性只需代入：

$$J_i^T J_i = D_i^{-1/2} M_i^T D_{i-1} M_i D_i^{-1/2} = I. \quad (3.191)$$

其转置作用于缩放向量时满足

$$J_i^T (D_{i-1}^{-1/2} \mathbf{x}) = D_i^{-1/2} M_i^T \mathbf{x}. \quad (3.192)$$

第 3.65 页已选择参数，使 $M_i^T \mathbf{x}$ 的目标分量为零，左侧的对角缩放不会改变这个零。

④ Note

备注：转置的位置。 第 3.65 页的第一种参数化 M_1 是对称矩阵，所以 M_1 与 M_1^T 可以互换；对一般步骤不能假定 M_i 对称，应采用式 (3.190) 的写法。

连续消元时，

$$\begin{aligned} R &= (D_L^{-1/2} M_L^T D_{L-1}^{1/2}) \cdots (D_1^{-1/2} M_1^T D_0^{1/2}) A \\ &= D_L^{-1/2} M_L^T \cdots M_1^T D_0^{1/2} A. \end{aligned} \quad (3.193)$$

若初始缩放 $D_0 = I$ ，则

$$Q = M_1 M_2 \cdots M_L D_L^{-1/2}, \quad R = Q^T A. \quad (3.194)$$

中间的 $D_i^{1/2}$ 与 $D_i^{-1/2}$ 两两抵消。也可以直接验证累计变换的正交性。令 $P_0 = I$ 、 $P_k = P_{k-1}M_k$ 。若 $P_{k-1}^T P_{k-1} = D_{k-1}$ ，则

$$P_k^T P_k = M_k^T P_{k-1}^T P_{k-1} M_k = M_k^T D_{k-1} M_k = D_k. \quad (3.195)$$

从 $P_0^T P_0 = D_0 = I$ 出发归纳，得到 $P_L^T P_L = D_L$ 。因此式 (3.194) 中 $Q = P_L D_L^{-1/2}$ 满足 $Q^T Q = I$ 。迭代中只需维护累计变换和对角缩放，平方根可以推迟到最后。

Tip

本页要点： 把平方根推迟到最后，可以在保持 QR 结构的同时减少中间计算。

3.67 Making A Matrix Upper Hessenberg

对实方阵 A ，令 $A_1 = A$ 。上 Hessenberg 形式满足 $h_{ij} = 0$ 当 $i > j + 1$ 。第 k 步对当前矩阵第 k 列的尾部 $A_{k+1:n,k}$ 构造 Householder 矩阵 $\widetilde{H}_k$ ，再嵌入

$$H_k = \text{diag}(I_k, \widetilde{H}_k). \quad (3.196)$$

为了保持特征值，必须同时从左、右作用：

$$A_{k+1} = H_k A_k H_k^T. \quad (3.197)$$

左乘把第 k 列的第 $k+2$ 行以下元素消为零；右乘恢复相似关系，虽然会更新后续列，却不会破坏已经形成的 Hessenberg 零结构。迭代 $k = 1, \dots, n-2$ 得到

$$H = Q^T A Q, \quad Q = H_1 H_2 \cdots H_{n-2}. \quad (3.198)$$

上 Hessenberg 化是非对称 QR 特征值算法的预处理，因为对 Hessenberg 矩阵做一次 QR 迭代只需 $O(n^2)$ 工作。

Note

为什么不直接消元得到 Schur 形式？ 单侧 Householder 消元可得到三角矩阵，但不保持特征值；配套的右侧变换虽恢复相似关系，却可能重新填入刚消掉的位置。以上三角 Schur 形式为例，若 $Q^H A Q = T$ ，则 $A Q = Q T$ 的第一列给出 $A q_1 = t_{11} q_1$ ，即确定第一列 Schur 向量已经涉及特征向量求解。因此通常先用 Householder 变换得到无需预知特征向量的 Hessenberg 形式，再用 QR 迭代逼近 Schur 形式。

Tip

本页要点： Householder 变换可把稠密矩阵约化为上 Hessenberg 形式，同时保持特征值不变。

3.68 Making A Matrix Bidiagonal

Hessenberg 化与双对角化都采用正交变换，先将稠密矩阵约化为便于后续迭代的形式。两者的区别在于需要保持什么量：求特征值时，采用同一个正交矩阵作相似变换 $H = Q^T A Q$ ，得到上 Hessenberg 形式；若 A 是实对称矩阵，则 H 也对称，因而进一步成为三对角矩阵。

转向奇异值问题后，矩阵可以是矩形，左右两侧也允许采用不同的正交矩阵。**变换 $B = U^T A V$ 仍保持奇异值，分别消去行、列中的非零元**，因此可以交替使用左右 Householder 变换，将矩阵约化为双对角形式。这里的右变换不再由左变换的相似性要求确定，而是针对当前行的消元目标单独构造；双对角化直接从 A 出发，无须先做 Hessenberg 化。

这就形成了两条对应的计算路径：特征值计算先做 Hessenberg 化（对称情形为三对角化），SVD 计算先做双对角化。正交变换保持二范数，不会放大已有扰动的二范数；直接双对角化还避免了显式形成 $A^T A$ 。后者在列满秩时满足 $\kappa_2(A^T A) = \kappa_2(A)^2$ ，形成过程中的舍入误差可能损害小奇异值的相对精度。

设 $A \in \mathbb{R}^{m \times n}$ 。双对角化交替使用左、右 Householder 变换。若先右乘，将第 k 行第 $k+1$ 列以后的元素消为零；再左乘，将第 k 列第 $k+2$ 行以下的元素消为零。便得到下双对角形式。

例如，对一个 4×3 矩阵，依次右乘 H_1^T 、左乘 H_2 、右乘 H_3^T 、左乘 H_4 ，得到 $B = H_4 H_2 A H_1^T H_3^T$ 。其中 H_1, H_3 是 3×3 的右侧反射， H_2, H_4 是 4×4 的左侧反射；令 $U^T = H_4 H_2$ 、 $V = H_1^T H_3^T$ ，便得到

$$B = U^T A V, \quad A = U B V^T, \quad (3.199)$$

其中 U, V 正交， B 只有主对角线和第一条次对角线可能非零。左右作用不是相似变换，因为 A 可以是矩形；它们保持奇异值，因为

$$B^T B = V^T A^T A V. \quad (3.200)$$

因此 $A^T A$ 与 $B^T B$ 正交相似，具有相同特征值。若每步先左乘消列、再右乘消行，则得到上双对角形式，两种约定都可使用。双对角结构用于 SVD 计算；LSQR 则通过 Golub - Kahan 迭代逐步构造相关的双对角小矩阵，无须显式形成这里的完整 U, V 。

Tip

本页要点： 左右正交变换保持奇异值，消元先后决定得到上双对角还是下双对角形式。

3.69 Method of Minimal Iteration

设 $A = A^T$ ，目标是近似特征对 $A v = \lambda v$ 。给定非零 v_0 ，在固定方向上使残量

$$r(\alpha) = A v_0 - \alpha v_0 \quad (3.201)$$

的二范数最小。展开平方：

$$\|r(\alpha)\|_2^2 = v_0^T A^2 v_0 - 2\alpha v_0^T A v_0 + \alpha^2 v_0^T v_0. \quad (3.202)$$

对 α 求导并令其为零。因为二次项系数 $v_0^T v_0 > 0$ ，唯一最小值点就是 Rayleigh 商：

$$\alpha_0 = \frac{v_0^T A v_0}{v_0^T v_0}. \quad (3.203)$$

令 $v_1 = A v_0 - \alpha_0 v_0$ ，则

$$v_0^T v_1 = v_0^T A v_0 - \alpha_0 v_0^T v_0 = 0. \quad (3.204)$$

这一步“残量与当前方向正交”是 Lanczos 递推的起点。若 $\mathbf{v}_1 = 0$ ，则已得到特征向量，无需继续递推。

本页方法出自 Cornelius Lanczos, “An iteration method for the solution of the eigenvalue problem of linear differential and integral operators”, *Journal of Research of the National Bureau of Standards*, 45(4), 1950, pp. 255 - 282.

Tip

本页要点： 固定非零向量后，Rayleigh 商使特征对残量的二范数最小，并使残量与该向量正交。

3.70 Lanczos Method

由上一页知 $\mathbf{v}_0 \perp \mathbf{v}_1$ 。设 $\mathbf{v}_1 \neq 0$ ，继续迭代时，既要消去 $A\mathbf{v}_1$ 在 $\mathbf{v}_1$ 上的分量，也要消去它在 $\mathbf{v}_0$ 上的分量。因此构造

$$\mathbf{v}_2 = A\mathbf{v}_1 - \alpha_1\mathbf{v}_1 - \beta_0\mathbf{v}_0. \quad (3.205)$$

同时选择 α, β ，使 $\|A\mathbf{v}_1 - \alpha\mathbf{v}_1 - \beta\mathbf{v}_0\|_2$ 最小。展开平方并利用 $\mathbf{v}_0^T\mathbf{v}_1 = 0$ ，有

$$\begin{aligned} F(\alpha, \beta) &= \|A\mathbf{v}_1 - \alpha\mathbf{v}_1 - \beta\mathbf{v}_0\|_2^2 \\ &= \|A\mathbf{v}_1\|_2^2 - 2\alpha\mathbf{v}_1^T A\mathbf{v}_1 + \alpha^2\|\mathbf{v}_1\|_2^2 \\ &\quad - 2\beta\mathbf{v}_0^T A\mathbf{v}_1 + \beta^2\|\mathbf{v}_0\|_2^2. \end{aligned} \quad (3.206)$$

对两个变量分别求导。由于两项二次系数均为正，令偏导数为零就得到唯一最小值点。关于 α 的条件等价于 $\mathbf{v}_2 \perp \mathbf{v}_1$ ：

$$0 = \mathbf{v}_1^T \mathbf{v}_2 = \mathbf{v}_1^T A\mathbf{v}_1 - \alpha_1 \mathbf{v}_1^T \mathbf{v}_1, \quad (3.207)$$

故

$$\alpha_1 = \frac{\mathbf{v}_1^T A\mathbf{v}_1}{\mathbf{v}_1^T \mathbf{v}_1}. \quad (3.208)$$

关于 β 的条件等价于 $\mathbf{v}_2 \perp \mathbf{v}_0$ ，得到

$$\beta_0 = \frac{\mathbf{v}_0^T A\mathbf{v}_1}{\mathbf{v}_0^T \mathbf{v}_0}. \quad (3.209)$$

由于 $A = A^T$ ，

$$\mathbf{v}_0^T A\mathbf{v}_1 = (A\mathbf{v}_0)^T \mathbf{v}_1 = (\mathbf{v}_1 + \alpha_0\mathbf{v}_0)^T \mathbf{v}_1 = \|\mathbf{v}_1\|_2^2, \quad (3.210)$$

所以 $\beta_0 = \|\mathbf{v}_1\|_2^2 / \|\mathbf{v}_0\|_2^2 \geq 0$ 。

Tip

本页要点： 同时减去两个旧方向上的分量，才能让新残量与这两个方向都正交。

3.71 Orthogonality Conditions

令

(7)

因为 $\mathbf{v}_0 \perp \mathbf{v}_1$, $A\mathbf{v}_1$ 在 S 上的正交投影为

$$\text{proj}_S(A\mathbf{v}_1) = \frac{\mathbf{v}_0^T A\mathbf{v}_1}{\mathbf{v}_0^T \mathbf{v}_0} \mathbf{v}_0 + \frac{\mathbf{v}_1^T A\mathbf{v}_1}{\mathbf{v}_1^T \mathbf{v}_1} \mathbf{v}_1 = \beta_0 \mathbf{v}_0 + \alpha_1 \mathbf{v}_1. \quad (3.212)$$

因此

$$\mathbf{v}_2 = A\mathbf{v}_1 - \text{proj}_S(A\mathbf{v}_1) \perp S. \quad (3.213)$$

这正是上一页最小化问题的几何意义：从 $A\mathbf{v}_1$ 中减去它在旧子空间上的正交投影。后续只保留最近两个方向仍能完成投影，则需要下一页的对称性论证。

Tip

本页要点： 残量最小化等价于减去正交投影，剩下的向量与旧子空间正交。

3.72 Short Recurrence Property

按一般的正交投影形式， $A\mathbf{v}_2$ 要减去 $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2$ 三个方向上的分量：

$$\mathbf{v}_3 = A\mathbf{v}_2 - \alpha_2 \mathbf{v}_2 - \beta_1 \mathbf{v}_1 - \gamma_0 \mathbf{v}_0. \quad (3.214)$$

但对称性使最早方向对应的系数为零。因为

$$\mathbf{v}_0^T A\mathbf{v}_2 = (A\mathbf{v}_0)^T \mathbf{v}_2 = (\mathbf{v}_1 + \alpha_0 \mathbf{v}_0)^T \mathbf{v}_2 = 0, \quad (3.215)$$

所以 $\gamma_0 = 0$ 。一般地，若已构造的非零向量 $\mathbf{v}_0, \dots, \mathbf{v}_j$ 两两正交，则对 $i \leq j-2$ ，已有递推给出 $A\mathbf{v}_i \in \text{span}\{\mathbf{v}_{i-1}, \mathbf{v}_i, \mathbf{v}_{i+1}\}$ ($i=0$ 时没有第一项)。这些向量都与 $\mathbf{v}_j$ 正交，故

$$\mathbf{v}_i^T A\mathbf{v}_j = (A\mathbf{v}_i)^T \mathbf{v}_j = 0. \quad (3.216)$$

于是每一步理论上只需最近两个旧方向，得到三项递推。类似地， $\beta_{j-1} = \|\mathbf{v}_j\|_2^2 / \|\mathbf{v}_{j-1}\|_2^2 \geq 0$ 。若在 $\mathbb{R}^n$ 中已构造出 n 个非零正交向量，下一残量只能为零，递推终止；第 3.74 页的归一化算法把这一条件写成 $\beta_n = 0$ 。

有限精度下旧向量会逐渐失去正交性，因此实际特征值计算可能需要选择性或完全再正交化。

Tip

本页要点： 理论上的三项递推只需保存少量向量，浮点正交性损失却必须额外处理。

3.73 Lanczos and Matrix Polynomials

把第 3.69 - 3.72 页生成的未归一化向量改记为 $\mathbf{z}_0, \mathbf{z}_1, \dots$ ，以便与下一页的单位向量 $\mathbf{v}_j$ 区分。本页的 α_j, β_j 仍沿用前面从 0 开始的编号，其中 $\beta_{j-1} = \|\mathbf{z}_j\|_2^2 / \|\mathbf{z}_{j-1}\|_2^2$ 。

令 $p_0(t) = 1$ 、 $p_1(t) = t - \alpha_0$ ，递推定义

$$p_{j+1}(t) = (t - \alpha_j)p_j(t) - \beta_{j-1}p_{j-1}(t), \quad j \geq 1. \quad (3.217)$$

初始有 $z_1 = (A - \alpha_0 I)z_0 = p_1(A)z_0$ 。若前两步已满足多项式表示，则

$$\begin{aligned} z_{j+1} &= (A - \alpha_j I)z_j - \beta_{j-1}z_{j-1} \\ &= [(A - \alpha_j I)p_j(A) - \beta_{j-1}p_{j-1}(A)]z_0 \\ &= p_{j+1}(A)z_0. \end{aligned} \quad (3.218)$$

由归纳法， $z_j = p_j(A)z_0$ 。这些 p_j 都是首一的 j 次多项式，因此 z_j 是 $z_0, Az_0, \dots, A^j z_0$ 的线性组合。记 Krylov 子空间为 $K_m(A, z_0) = \text{span}\{z_0, Az_0, \dots, A^{m-1}z_0\}$ 。在尚未终止时， $z_0, \dots, z_{m-1}$ 两两正交且非零，故它们构成该空间的一组基。

怎样转成单位长度的向量？ 由平方范数比逐步相乘，

$$\begin{aligned} \|z_j\|_2^2 &= \|z_0\|_2^2 \beta_0 \beta_1 \cdots \beta_{j-1}, \\ v_1 &= \frac{z_0}{\|z_0\|_2}, \quad v_{j+1} = \frac{z_j}{\|z_0\|_2 \sqrt{\beta_0 \cdots \beta_{j-1}}}. \end{aligned} \quad (3.219)$$

后一式适用于 $j \geq 1$ 且此前各 $\beta_i > 0$ 。将 $z_j = \|z_j\|_2 v_{j+1}$ 代入未归一化递推，并除以 $\|z_{j-1}\|_2$ ，得到

$$Av_j = \sqrt{\beta_{j-2}}v_{j-1} + \alpha_{j-1}v_j + \sqrt{\beta_{j-1}}v_{j+1}, \quad j \geq 2. \quad (3.220)$$

其中前一项的系数由 $\beta_{j-2}\|z_{j-2}\|_2/\|z_{j-1}\|_2 = \sqrt{\beta_{j-2}}$ 得到；首步则是 $Av_1 = \alpha_0 v_1 + \sqrt{\beta_0}v_2$ 。这解释了归一化后为什么出现平方根，也说明下一页需要重新编号系数。

一般的 p_j 不是 A 的特征多项式；若没有提前终止，最后得到的 p_n 才是，证明见第 3.77 页。

Tip

本页要点： 未归一化向量是首一矩阵多项式作用于初始向量的结果；归一化把平方范数比变成相邻向量之间的系数。

3.74 Lanczos Iteration

从本页起，使用从 1 开始编号的单位向量，并重新定义系数：新 α_j 对应上一页的 α_{j-1} ，新 β_j 对应上一页的 $\sqrt{\beta_{j-1}}$ 。因此， β_j 现在是新残量的范数，不再是未归一化向量的平方范数比。

令 $v_0 = 0$ 、 $\beta_0 = 0$ ，并取 $\|v_1\|_2 = 1$ 。对实对称矩阵 A ，第 j 步执行

$$\begin{aligned} w_j &= Av_j - \beta_{j-1}v_{j-1}, \\ \alpha_j &= v_j^T w_j, \\ w_j &\leftarrow w_j - \alpha_j v_j, \\ \beta_j &= \|w_j\|_2, \\ \beta_j &= 0 : \text{ 停止}; \quad \beta_j > 0 : v_{j+1} = w_j/\beta_j. \end{aligned} \quad (3.221)$$

精确算术下， $v_j \perp v_{j-1}$ ，所以计算 α_j 时减去前一方向不改变内积，即 $\alpha_j = v_j^T Av_j$ 。当 $\beta_j > 0$ 时，上述算法给出

$$A\mathbf{v}_j = \beta_{j-1}\mathbf{v}_{j-1} + \alpha_j\mathbf{v}_j + \beta_j\mathbf{v}_{j+1}. \quad (3.222)$$

令 $V_j = [\mathbf{v}_1, \dots, \mathbf{v}_j]$, 将 α_i 放在对角、 β_i 放在相邻对角线上, 得到对称三对角矩阵 T_j 。按列排列递推式, 便有

$$AV_j = V_jT_j + \mathbf{w}_j\mathbf{e}_j^T = V_jT_j + \beta_j\mathbf{v}_{j+1}\mathbf{e}_j^T \quad (\beta_j > 0). \quad (3.223)$$

除最后一列外, 各列的下一方向都已经包含在 V_j 中, 故余项只出现在最后一列。左乘 V_j^T , 由残量与旧基正交, 得到 $T_j = V_j^T AV_j$ 。

按第 3.73 页的定义, $\text{range}(V_j) = K_j(A, \mathbf{v}_1)$ 。若 $\beta_j = 0$, 则 $\mathbf{w}_j = 0$, 直接得到 $AV_j = V_jT_j$; 不必定义 $\mathbf{v}_{j+1}$ 。对任意 $\mathbf{z} \in \mathbb{R}^j$, 有 $A(V_j\mathbf{z}) = V_j(T_j\mathbf{z})$, 说明这个 Krylov 子空间已对 A 不变。为什么这意味着成功终止, 以及为什么至多执行 n 步, 下一页给出证明。

Tip

本页要点: 归一化 Lanczos 的 β_j 是残量范数; 先判断它是否为零, 再决定是否构造下一个向量。

3.75 Lucky Breakdown and Finite Termination

以下结论均在精确算术下讨论。若第 m 步出现 $\beta_m = 0$, 由第 3.74 页有 $AV_m = V_mT_m$ 。设 $T_m\mathbf{y} = \theta\mathbf{y}$, $\mathbf{y} \neq 0$, 则

$$A(V_m\mathbf{y}) = V_mT_m\mathbf{y} = \theta(V_m\mathbf{y}), \quad \|V_m\mathbf{y}\|_2 = \|\mathbf{y}\|_2 > 0. \quad (3.224)$$

因此 T_m 的每个特征对都给出 A 的一个特征对。这种提前终止称为 **lucky breakdown**: 已经找到了一个不变子空间, 无需继续构造下一个向量。

若还需要其他特征对, 可从 $\text{range}(V_m)^\perp$ 中另取单位向量开始。这样做不会在精确算术下重新进入原空间, 因为对任意 $\mathbf{x} \perp \text{range}(V_m)$,

$$V_m^T A\mathbf{x} = (AV_m)^T \mathbf{x} = (V_mT_m)^T \mathbf{x} = T_m V_m^T \mathbf{x} = \mathbf{0}. \quad (3.225)$$

这里用了 $A = A^T$ 和 $T_m = T_m^T$, 说明不变子空间的正交补也对 A 不变。

为什么没有提前终止时必有 $\beta_n = 0$? 此时 $\mathbf{v}_1, \dots, \mathbf{v}_n$ 已构成 $\mathbb{R}^n$ 的标准正交基。第 3.71 - 3.72 页的正交性与短递推证明给出: 残量 $\mathbf{w}_n = A\mathbf{v}_n - \alpha_n\mathbf{v}_n - \beta_{n-1}\mathbf{v}_{n-1}$ 与所有旧基向量正交。因此

$$V_n^T \mathbf{w}_n = \mathbf{0}, \quad \mathbf{w}_n = V_n(V_n^T \mathbf{w}_n) = \mathbf{0}, \quad \beta_n = \|\mathbf{w}_n\|_2 = 0. \quad (3.226)$$

于是递推在第 n 步终止, 并得到

$$AV_n = V_nT_n, \quad V_n^T AV_n = T_n, \quad A = V_nT_nV_n^T. \quad (3.227)$$

若提前终止, 就不再定义后续系数。浮点计算中, 正交性可能丢失, 算出的 β_n 也不保证恰好为零, 不能把上述有限步终止结论直接当作程序的停止准则。

Tip

本页要点: 零残量表示已找到不变子空间; 精确算术下, Lanczos 至多 n 步终止。

3.76 Approximating Eigenpairs

不必等到递推终止, T_m 的特征对就可用来近似 A 的特征对。设 $\beta_m > 0$, 取

$$T_m \mathbf{y}_i = \theta_i \mathbf{y}_i, \quad \|\mathbf{y}_i\|_2 = 1, \quad \mathbf{u}_i = V_m \mathbf{y}_i. \quad (3.228)$$

θ_i 称为 Ritz 值, $\mathbf{u}_i$ 称为 Ritz 向量。由 $V_m^T V_m = I$, 有 $\|\mathbf{u}_i\|_2^2 = \mathbf{y}_i^T V_m^T V_m \mathbf{y}_i = 1$ 。

把第 3.74 页的 Lanczos 矩阵关系右乘 $\mathbf{y}_i$, 可得残量

$$\begin{aligned} \mathbf{r}_i &= A\mathbf{u}_i - \theta_i \mathbf{u}_i \\ &= (AV_m - V_m T_m) \mathbf{y}_i \\ &= \beta_m \mathbf{v}_{m+1} (\mathbf{e}_m^T \mathbf{y}_i). \end{aligned} \quad (3.229)$$

记 $y_{i,m} = \mathbf{e}_m^T \mathbf{y}_i$ 为小特征向量 $\mathbf{y}_i$ 的最后一个分量。由于 $\|\mathbf{v}_{m+1}\|_2 = 1$,

$$\|\mathbf{r}_i\|_2 = \beta_m |y_{i,m}|. \quad (3.230)$$

因此, 在精确算术下, 无须再作一次大矩阵-向量乘, 就能计算该近似特征对的残量范数。

小残量能说明特征值准确吗? 由第 2.41 页从 Schur 分解得到的结论, 实对称矩阵 A 有标准正交特征向量基 $\mathbf{q}_1, \dots, \mathbf{q}_n$ 。写成

$$A\mathbf{q}_j = \lambda_j \mathbf{q}_j, \quad \mathbf{u}_i = \sum_{j=1}^n c_j \mathbf{q}_j, \quad \sum_{j=1}^n |c_j|^2 = 1. \quad (3.231)$$

则 $\mathbf{r}_i = \sum_j c_j (\lambda_j - \theta_i) \mathbf{q}_j$ 。利用正交性,

$$\begin{aligned} \|\mathbf{r}_i\|_2^2 &= \sum_{j=1}^n |c_j|^2 |\lambda_j - \theta_i|^2 \\ &\geq \left(\min_j |\lambda_j - \theta_i| \right)^2 \sum_{j=1}^n |c_j|^2, \\ \min_{\lambda \in \lambda(A)} |\theta_i - \lambda| &\leq \|\mathbf{r}_i\|_2. \end{aligned} \quad (3.232)$$

这保证至少有一个真实特征值距离 θ_i 不超过残量范数。但它不保证 $\mathbf{u}_i$ 接近某个指定特征向量: 相邻特征值很接近时, 不同特征方向的组合也可能产生小残量。

Tip

本页要点: Lanczos 近似特征对的残量由 $\beta_m |y_{i,m}|$ 给出, 并为最近特征值的误差提供上界。

3.77 Lanczos and Characteristic Polynomial

若精确算术下没有提前终止, 由第 3.75 页有 $A = V_n T_n V_n^T$, 所以

$$\det(\lambda I - A) = \det(\lambda I - T_n). \quad (3.233)$$

令 T_j 为 T_n 的前 j 阶主子矩阵, 定义 $p_j(\lambda) = \det(\lambda I_j - T_j)$, 并约定

$$p_0(\lambda) = 1, \quad p_1(\lambda) = \lambda - \alpha_1. \quad (3.234)$$

对三对角行列式的最后一行展开。对角元贡献 $(\lambda - \alpha_j)p_{j-1}(\lambda)$ ；次对角元为 $-\beta_{j-1}$ ，其代数余子式的符号为负，故另一项是 $+\beta_{j-1}\det(M_{j-1})$ 。当 $j \geq 3$ 时，删去最后一行和倒数第二列得到

$$M_{j-1} = \begin{bmatrix} \lambda I_{j-2} - T_{j-2} & \mathbf{0} \\ -\beta_{j-2}\mathbf{e}_{j-2}^T & -\beta_{j-1} \end{bmatrix}, \quad \det(M_{j-1}) = -\beta_{j-1}p_{j-2}(\lambda). \quad (3.235)$$

当 $j = 2$ 时，直接展开二阶行列式即可。两种情况都给出

$$p_j(\lambda) = (\lambda - \alpha_j)p_{j-1}(\lambda) - \beta_{j-1}^2 p_{j-2}(\lambda), \quad j \geq 2. \quad (3.236)$$

平方 β_{j-1}^2 来自上、下两个相邻对角元的乘积。按第 3.74 页的系数对应关系，新 α_j 是旧 α_{j-1} ，新 β_{j-1}^2 是旧 β_{j-2} ，因此它与第 3.73 页的首一多项式递推完全一致。

当 $j < n$ 时， p_j 是投影矩阵 T_j 的特征多项式，根为 Ritz 值；当 $j = n$ 且没有提前终止时， p_n 就是 A 的特征多项式。若在 $m < n$ 时终止，得到的是 A 在已找到的不变子空间上的特征多项式，不一定包含其他子空间的特征值。

Tip

本页要点： 三项多项式递推等价于三对角行列式展开；完整 n 步给出原矩阵的特征多项式。

3.78 Extension to Nonsymmetric Matrices

非对称矩阵的右、左特征向量分别满足 $A\mathbf{x}_k = \lambda_k\mathbf{x}_k$ 、 $\mathbf{y}_k^T A = \lambda_k\mathbf{y}_k^T$ ，两者一般不同。这提示我们同时构造右 Krylov 基 $V_j = [\mathbf{v}_1, \dots, \mathbf{v}_j]$ 和左 Krylov 基 $W_j = [\mathbf{w}_1, \dots, \mathbf{w}_j]$ ，要求它们双正交：

$$W_j^T V_j = I_j. \quad (3.237)$$

Note

左右特征向量为什么与双正交性有关？ 在实数情形下，由左右特征方程， $\lambda_j \mathbf{y}_i^T \mathbf{x}_j = \mathbf{y}_i^T A \mathbf{x}_j = \lambda_i \mathbf{y}_i^T \mathbf{x}_j$ ，故当 $\lambda_i \neq \lambda_j$ 时， $\mathbf{y}_i^T \mathbf{x}_j = 0$ 。若 A 可对角化，写成 $A = X\Lambda X^{-1}$ ，取 $Y^T = X^{-1}$ ，则 $Y^T A = \Lambda Y^T$ 、 $Y^T X = I$ ，所以可选出完整的左右双正交特征向量基。重特征值对应的左右基需要配对选择，任意选取并不自动双正交；不可对角化时，则不存在完整的特征向量基。复数情形将转置换为共轭转置，结论相同。

这里的迭代向量不要求已经是特征向量。对实矩阵 A ，初始化时任选 $\tilde{\mathbf{v}}_1, \tilde{\mathbf{w}}_1$ ，使 $\tilde{\mathbf{w}}_1^T \tilde{\mathbf{v}}_1 \neq 0$ ，先令

$$\mathbf{v}_1 = \frac{\tilde{\mathbf{v}}_1}{\|\tilde{\mathbf{v}}_1\|_2}, \quad \mathbf{w}_1 = \frac{\tilde{\mathbf{w}}_1}{\tilde{\mathbf{w}}_1^T \mathbf{v}_1}, \quad (3.238)$$

便有 $\mathbf{w}_1^T \mathbf{v}_1 = 1$ ；第二个分母使用的是归一化后的 $\mathbf{v}_1$ 。再置 $\mathbf{v}_0 = \mathbf{w}_0 = 0$ 、 $\beta_0 = \gamma_0 = 0$ ，后续同时对 A 和 A^T 做三项递推。双正交要求 $\mathbf{w}_i^T \mathbf{v}_j = \delta_{ij}$ ，但不要求两组向量各自正交，也不要求每个向量的二范数为 1。

Tip

本页要点： 双边 Lanczos 用左右基保持双正交，以处理非对称矩阵的左右谱信息。

3.79 Lanczos Method for Nonsymmetric Matrices

使用上一页的初始化，并保持 $\mathbf{w}_j^T \mathbf{v}_j = 1$ 。第 j 步计算

$$\begin{aligned}\alpha_j &= \mathbf{w}_j^T A \mathbf{v}_j, \\ \mathbf{p}_j &= A \mathbf{v}_j - \alpha_j \mathbf{v}_j - \beta_{j-1} \mathbf{v}_{j-1}, \\ \mathbf{q}_j &= A^T \mathbf{w}_j - \alpha_j \mathbf{w}_j - \gamma_{j-1} \mathbf{w}_{j-1}, \\ \delta_j &= \mathbf{q}_j^T \mathbf{p}_j.\end{aligned}\tag{3.239}$$

若 $\delta_j \neq 0$ ，取

$$\gamma_j = \sqrt{|\delta_j|}, \quad \beta_j = \frac{\delta_j}{\gamma_j},\tag{3.240}$$

并令

$$\mathbf{v}_{j+1} = \frac{\mathbf{p}_j}{\gamma_j}, \quad \mathbf{w}_{j+1} = \frac{\mathbf{q}_j}{\beta_j}.\tag{3.241}$$

归一化满足

$$\mathbf{w}_{j+1}^T \mathbf{v}_{j+1} = \frac{\mathbf{q}_j^T \mathbf{p}_j}{\beta_j \gamma_j} = \frac{\delta_j}{\delta_j} = 1.\tag{3.242}$$

还需确认新向量与旧向量双正交。以下在精确算术下讨论。假设前 j 对已双正交，先前的左侧递推给出 $A^T \mathbf{w}_i = \gamma_{i-1} \mathbf{w}_{i-1} + \alpha_i \mathbf{w}_i + \beta_i \mathbf{w}_{i+1}$ ，其中 $1 \leq i < j$ 。因此， $i \leq j-2$ 时有 $\mathbf{w}_i^T A \mathbf{v}_j = (A^T \mathbf{w}_i)^T \mathbf{v}_j = 0$ ； $i = j-1$ 时该内积为 β_{j-1} ； $i = j$ 时由定义等于 α_j 。代回 $\mathbf{p}_j$ 的定义即得 $\mathbf{w}_j^T \mathbf{p}_j = 0$ 。同理，利用先前的右侧递推可得 $\mathbf{q}_j^T \mathbf{v}_j = 0$ 。结合归一化等式，归纳得到全部双正交关系。

若 $\delta_j \neq 0$ ，右侧三项递推给出

$$A \mathbf{v}_j = \mathbf{v}_j T_j + \gamma_j \mathbf{v}_{j+1} \mathbf{e}_j^T.\tag{3.243}$$

左乘 $\mathbf{w}_j^T$ ，得到

$$\mathbf{w}_j^T A \mathbf{v}_j = T_j,\tag{3.244}$$

其中 T_j 为非对称三对角矩阵，对角元为 $\alpha_1, \dots, \alpha_j$ ，上、下对角元分别为 $\beta_1, \dots, \beta_{j-1}$ 和 $\gamma_1, \dots, \gamma_{j-1}$ 。

若第 n 步之前没有发生 breakdown，则 $\mathbf{V}_n, \mathbf{W}_n$ 都是可逆方阵。由 $\mathbf{W}_n^T \mathbf{p}_n = 0$ 和 $\mathbf{q}_n^T \mathbf{V}_n = 0$ ，可知 $\mathbf{p}_n = \mathbf{q}_n = 0$ 。因此 $A \mathbf{V}_n = \mathbf{V}_n T_n$ 、 $\mathbf{W}_n^T A \mathbf{V}_n = T_n$ ，算法在第 n 步终止，不再构造下一对向量。

若 $\mathbf{p}_j = \mathbf{0}$ ，右侧 Krylov 子空间已对 A 不变；若 $\mathbf{q}_j = \mathbf{0}$ ，左侧 Krylov 子空间已对 A^T 不变，应先判断相应的终止情形。只有当 $\mathbf{p}_j, \mathbf{q}_j$ 均非零而 $\delta_j = \mathbf{q}_j^T \mathbf{p}_j = 0$ 时，才出现 serious breakdown：无法用标量缩放保持下一对向量双正交，需要 look-ahead Lanczos 等策略。浮点计算中， δ_j 过小也可能导致归一化时误差被放大。

💡 Tip

本页要点： 非对称 Lanczos 的三项递推依赖双正交条件，有限精度下须留意双正交性损失和近崩溃。

3.80 Historic Comments on Lanczos Methods

Lanczos 在 1950 年提出迭代三对角化，原目标是大规模特征值问题。Hestenes 与 Stiefel 在 1952 年发表共轭梯度法 (CG)：对 n 阶对称正定系统，精确算术下至多 n 步得到精确解。两者都以 Krylov 子空间和短递推为核心，数学上紧密相连。

早期实践面临的问题并不完全相同。Lanczos 向量在有限精度下会失去正交性，产生重复 Ritz 值等现象。CG 早期受到的关注有限，部分原因是当时着重考察“至多 n 步终止”：舍入误差破坏严格共轭后，这一保证不再成立。

把 CG 用作大规模稀疏对称正定系统的迭代法时，更应关心达到目标精度需要多少步，以及每步的存储和计算成本。Barrett 等人的 *Templates for the Solution of Linear Systems* (第 2 版，§ 2.5, pp. 30–31) 指出，Reid 在 1971 年强调了 CG 在稀疏对称正定系统中的应用，重新激发了研究兴趣。

Note

备注：Lanczos 失去正交性后怎么办？

正交性损失不等于有限精度 Lanczos 完全失效。它与收敛 Ritz 向量有关，可以通过完全、选择性或局部再正交化控制；如何处理取决于求解目标和内存成本。

Tip

本页要点： 实际评价 Krylov 方法，应看达到目标精度的成本，不能只看精确算术下的有限步终止。

本章小结

Important

第 3 章回顾

- 有限精度改变了运算规则。** 浮点数的精度与范围是两回事：舍入误差、相消、溢出和下溢需要分别判断。在无溢出且结果处于适当范围时，单位舍入误差给出基本运算的相对误差尺度；次正规数附近不能照搬这一相对误差模型。
- 先分清问题敏感性与算法稳定性。** 条件数衡量输入扰动的放大，后向误差衡量算出了哪个邻近问题，前向误差衡量答案偏离多少。在小扰动条件下，前向误差受条件数与后向误差共同控制；小残量不自动意味着准确解。
- 迭代精校各部分可采用不同精度。** 低精度分解降低校正成本，高精度残量保留误差信息，工作精度完成更新。LU-IR 的校正必须足够准确才能收缩；GMRES-IR 可扩大适用范围，但增加矩阵乘与正交化成本，可达到的精度仍有限。
- 正交化既是代数工具，也是稳定性工具。** Householder 适合成批消元，Givens 适合局部消元，Gram-Schmidt 便于逐列构造基；三者都可用于 QR 分解。完整 QR 与薄 QR 的因子维数不同，使用时要对应。MGS 通常比 CGS 更能保持正交性，但有限精度下仍可能需要再正交化；实现时不必显式形成每个变换矩阵。
- 矩阵约化减少后续计算量。** 正交相似变换得到 Hessenberg 或对称三对角形式，左右正交变换得到双对角形式。Lanczos 通过矩阵-向量乘构造 Krylov 基；对称性支持短递推，非对称 Lanczos 采用双正交，两者都要留意终止条件与正交性损失。
- 计算顺序同时影响误差与性能。** 内积误差分析说明舍入如何累积，成对求和缩短累积链，分块改善数

据复用。编译器重排、FMA 和并行归约可能改变结果；评价混合精度收益，要看带宽、通信与依赖关系，而不能只比较峰值算力。

读完本章，应能回答： 一个结果不够准确时，如何区分问题病态、算法不稳定和精度不足，并选择有针对性的改进？