最优化大作业

张晋

北京航空航天大学，数学与系统科学学院

2017 年 12 月 25 日

目录

1 Problem 5.19

1.1 重要数据展示

求解对称正定的方程组 $\boldsymbol{Gx}=\boldsymbol{b}$ 可以看作极小化二次函数

$$q(\boldsymbol{x})=\frac{1}{2}\boldsymbol{x^TGx-b^Tx}$$

$$\boldsymbol{g}(\boldsymbol{x})=\nabla q(\boldsymbol{x})=\boldsymbol{Gx}-\boldsymbol{b}$$

其中，$\nabla^2 q(\boldsymbol{x})=\boldsymbol{G}$ 为 Hilbert 矩阵。

希尔伯特矩阵是一种系数都是单位分数的方块矩阵，希尔伯特矩阵 $\boldsymbol{H}$ 的第 i 横行第 j 纵列的系数是 $H_{ij}=\dfrac{1}{i+j-1}$

以 5 阶的 Hilbert 矩阵为例，其矩阵如下:

$$H_5=\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{pmatrix}$$

希尔伯特矩阵是一种数学变换矩阵，正定，且高度病态（即，任何一个元素发生一点变动，整个矩阵的行列式的值和逆矩阵都会发生巨大变化），病态程度和阶数相关。

希尔伯特矩阵的一个特点就是条件数特别大，以上面的 5 阶 Hilbert 矩阵为例，当范数为 l_2 矩阵范数时，其条件数大约是 4.8×10^5

因此，用经证明: 当 $n\to\infty$ 的时候，$n\times n$ 的希尔伯特矩阵的条件数近似为 $O((1+\sqrt{2})^{4n}/\sqrt{n})$

MATLAB 中有直接生成 N 阶希尔伯特矩阵的函数:$\boxed{hilb(N)}$

1.2 算法伪代码

Algorithm 1 Conjugate gradient method method for problem(5.19)

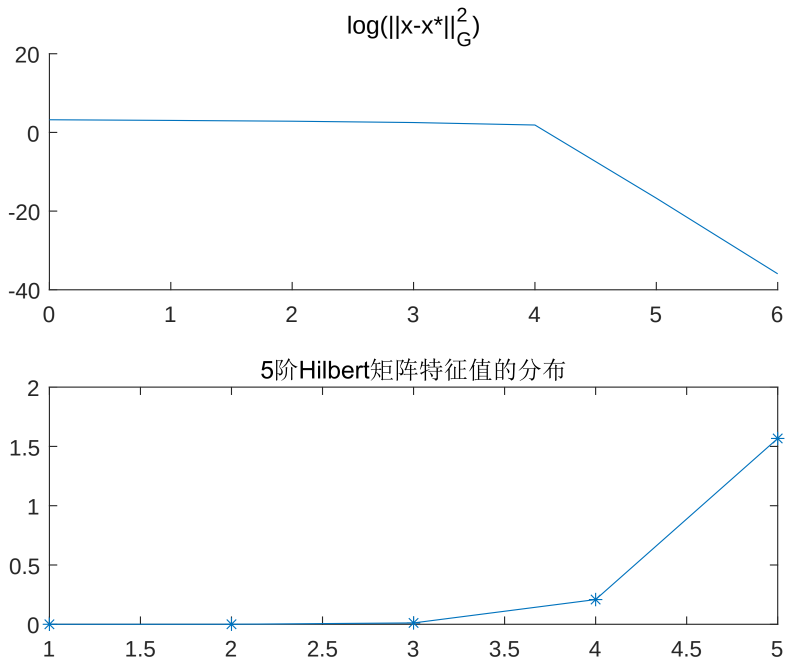
Given $\boldsymbol{x}^{(0)}$ and $\boldsymbol{G}$
Set $\boldsymbol{g}^{(0)} = \boldsymbol{G}\boldsymbol{x}^{(0)} - \boldsymbol{b}$
Set $\boldsymbol{p}^{(0)} = -\boldsymbol{g}^{(0)}, k = 0$
while $\|\boldsymbol{g}^{(k)}\| > \epsilon$ **do**
  Set $\boldsymbol{d} = \boldsymbol{G}\boldsymbol{p}^{(k)}$
  Set $\alpha_k = \dfrac{{\boldsymbol{g}^{(k)}}^T \boldsymbol{g}^{(k)}}{{\boldsymbol{p}^{(k)}}^T \boldsymbol{d}}$
  Set $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{p}^{(k)}$
  Set $\boldsymbol{g}^{(k+1)} = \boldsymbol{g}^{(k)} + \alpha_k \boldsymbol{d}$
  Set $\beta_{k+1} = \dfrac{{\boldsymbol{g}^{(k+1)}}^T \boldsymbol{g}^{(k+1)}}{{\boldsymbol{g}^{(k)}}^T \boldsymbol{g}^{(k)}}$
  Set $\boldsymbol{p}^{(k+1)} = -\boldsymbol{g}^{(k+1)} + \beta_{k+1} \boldsymbol{p}^{(k)}$
  Set $k = k + 1$
end while
return $\boldsymbol{x}^{(k)}$ as $\boldsymbol{x}^{\star}$

1.3 迭代结果展示

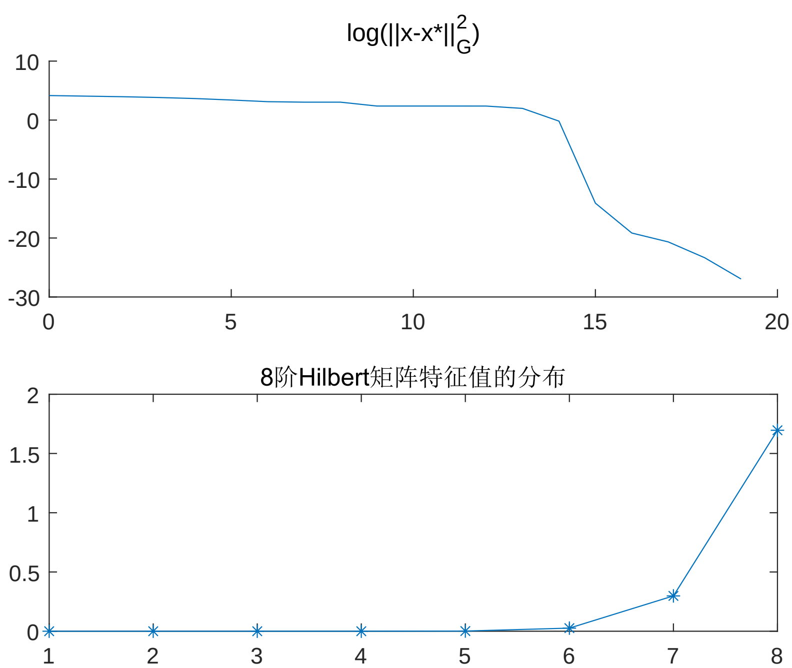
求解不同阶数 Hilbert 矩阵的迭代次数列成表格如下:

表 1: 迭代次数

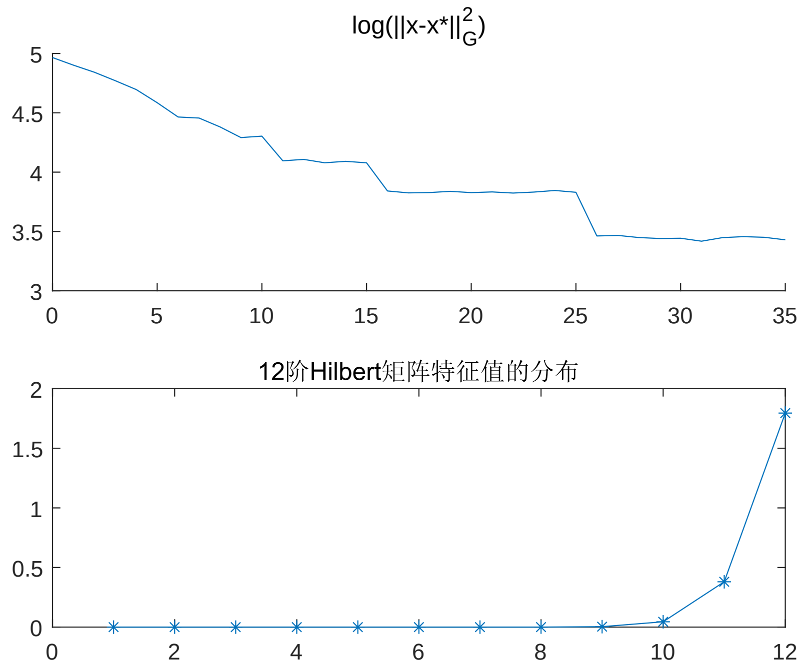
n	5	8	12	20
step	6	19	35	66

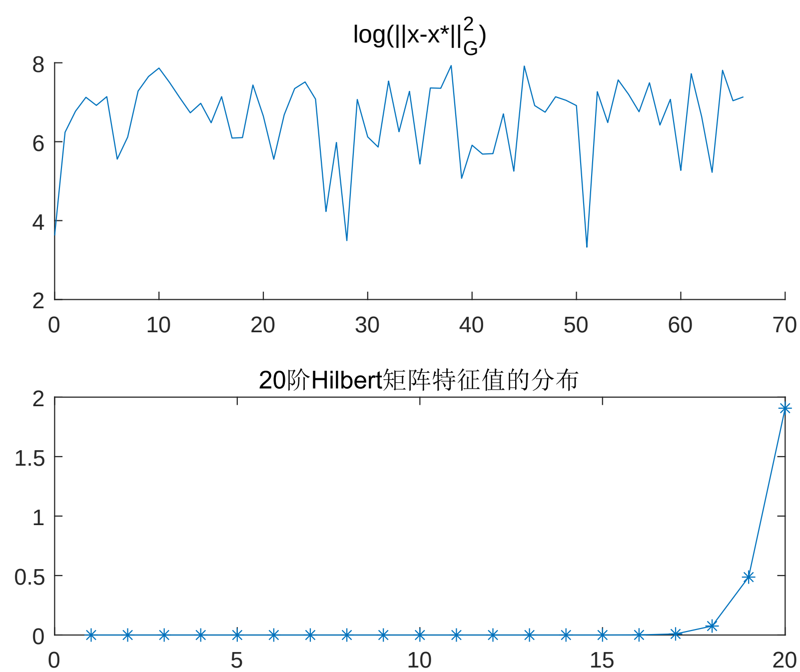


log(||x-x*||$_G^2$)
20
0
-20
-40
0 1 2 3 4 5 6
5阶Hilbert矩阵特征值的分布
2
1.5
1
0.5
0
1 1.5 2 2.5 3 3.5 4 4.5 5



log(||x-x*||$_G^2$)
10
0
-10
-20
-30
0 5 10 15 20
8阶Hilbert矩阵特征值的分布
2
1.5
1
0.5
0
1 2 3 4 5 6 7 8


log(||x-x*||$_G^2$)
12阶Hilbert矩阵特征值的分布


log(||x-x*||$_G^2$)
20阶Hilbert矩阵特征值的分布

1.4 总结分析

首先我们通过一个小程序考察一下 Hilbert 矩阵的病态性质:

```
N=20;
G=random('unif',0,1,[N N]);
G1=inv(G);
I=eye(N);
norm(G*G1-I)

%ans =
%
%  2.9348e-14
```

先生成一个 20×20 的随机矩阵，然后用 MATLAB 自带的求逆函数 inv() 求逆，然后将原矩阵与其逆相乘再减去单位阵，之后求矩阵范数，求得结果为 2.9348e-14，可见 MATLAB 自带的求逆函数在面对一般问题时还是比较精确的。

```
N=20;
H=hilb(N);
H_inv=inv(H);
I=eye(N);
norm(H*H_inv-I)

%ans =
%
%  33.8220
```

然后我们对 Hilbert 矩阵进行相同操作，结果为 33.8220，远远大于上面那个结果，足以见 Hilbert 矩阵的高度病态性质。

另外，我们由书本知识可以知道: 共轭梯度法与特征值的分布有关，当特征值分布较密集时，相同步数迭代精度更高。所以我们可以对 Hilbert 矩阵进行预处理，降低其条件数。

嗯，那如何对 Hilbert 矩阵进行预处理呢？我在网上经过不懈的搜索，终于在 Alfi Quarteroni Fausto Saleri 的 Scientifi Computing with MATLAB and Octave 中找到这么一个题目 (见图1)，题目中点出可以用 Hilbert 矩阵对角元素构建一个对角矩阵，作为预处理矩阵。

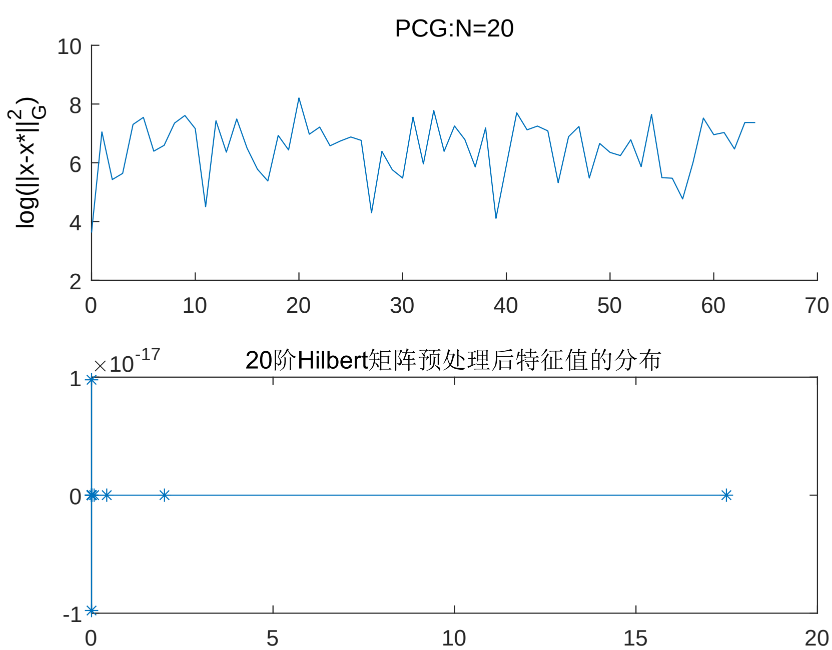
Example 5.16 Let us go back to Example 5.9 on the Hilbert matrix and solve the related system (for different values of n) by the preconditioned gradient (PG) and the preconditioned conjugate gradient (PCG) methods, using as preconditioner the diagonal matrix D made of the diagonal entries of the Hilbert matrix. We fix $\mathbf{x}^{(0)}$ to be the null vector and iterate until the relative residual (5.64) is less than 10^{-6}. In Table 5.4 we report the absolute errors (with respect to the exact solution) obtained with PG and PCG methods, as well as the errors obtained using the MATLAB command \. For the latter, the error degenerates when n gets large. On the other hand, we can appreciate the beneficial effect that a suitable iterative method such as the PCG scheme can have on the number of iterations. ■

图 1: 从书上摘的题

也就是说，设 C 是由 G 的对角线元素开方构成的对角矩阵，令 $\hat{G} = C^{-T}GC^{-1}$，$\hat{x} = Cx$,经过变换可调整为 $M^{-1}Gx = M^{-1}b$,其中 $M = C^TC$，即由 G 对角线元素构成的对角阵。

经过这么一番预处理后，不难看出不难看出该矩阵仍然是对称正定矩阵，而且其对角元素全为 1.

那么我就使用书上的预处理共轭梯度法首先对 $N = 20$ 的 Hilbert 矩阵进行检验：



可见这个预处理确实把特征值都聚集到一起了，但是迭代次数却为 65 次，仅仅比之前减少了一步，优化不太明显，这令我非常困惑，于是我改变终止精度进行测试，结果如下:

表 2: 不同精度下迭代次数的比较 ($N = 20$)

Error	10^{-6}	10^{-7}	10^{-8}
Steps of CG	66	121	520
Steps of PCG	65	115	314

可见在精度要求较低的情况下，两者的相差并不明显。

然后我试着将 N 调到 100，观察其迭代情况，发现在精度较低时，PCG 法所迭代的次数有时候甚至比 CG 法还要多，但在精度要求达 10^{-8} 时，CG 法无论如何迭代都无法满足精度要求，而 PCG 略胜一筹，精度高出了一个数量级.

表 3: 不同精度下迭代次数的比较 ($N = 100$)

Error	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
Steps of CG	12	25	47	175	748	2132(error=1.132e-08)
Steps of PCG	15	27	65	111	756	1686(error=9.683e-09)

此外，该书上还列举了梯度下降法 (PG) 和预处理共轭梯度法 (PCG) 在求解该问题上的比较，我认为比较有借鉴意义，贴在图下：

Table 5.4. Errors obtained using the preconditioned gradient method (PG), the preconditioned conjugate gradient method (PCG), and the direct method implemented in the MATLAB command \ for the solution of the Hilbert system. For the iterative methods also the number of iterations is reported

		\	PG		PCG	
n	$K(\mathrm{A}_n)$	Error	Error	Iter	Error	Iter
4	1.55e+04	7.72e-13	8.72e-03	995	1.12e-02	3
6	1.50e+07	7.61e-10	3.60e-03	1813	3.88e-03	4
8	1.53e+10	6.38e-07	6.30e-03	1089	7.53e-03	4
10	1.60e+13	5.24e-04	7.98e-03	875	2.21e-03	5
12	1.70e+16	6.27e-01	5.09e-03	1355	3.26e-03	5
14	6.06e+17	4.12e+01	3.91e-03	1379	4.32e-03	5

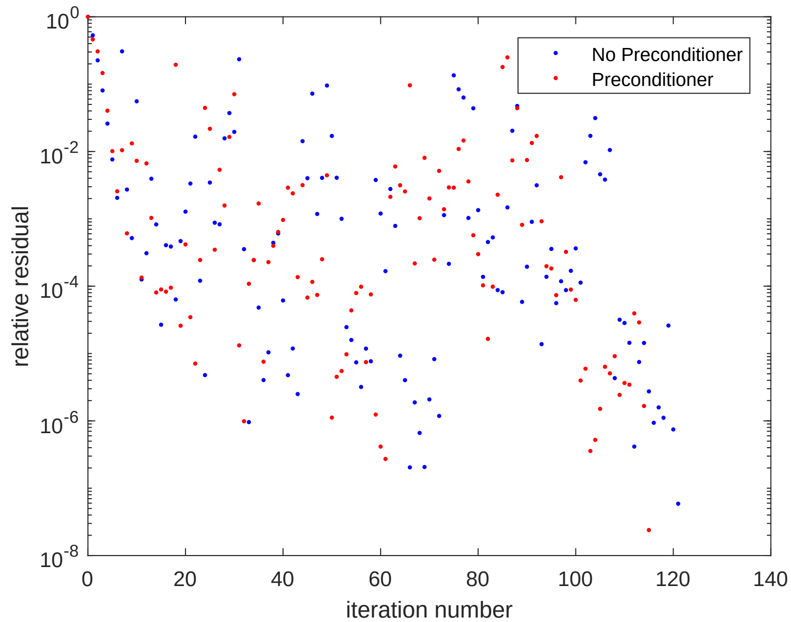


图 2: $(N = 20)$,error=e-7

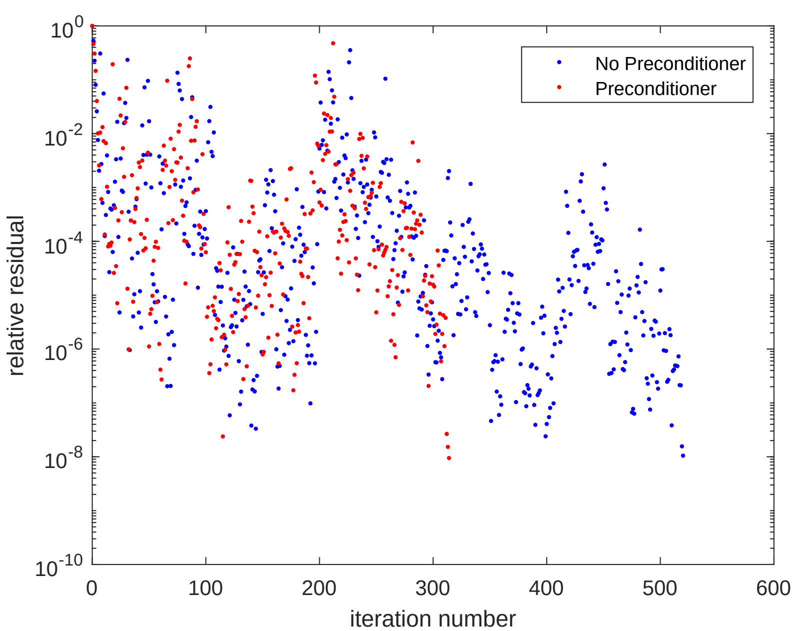


图 3: $(N = 20)$,error=e-8