Applied Numerical Linear Algebra. Lecture 6

Special Linear Systems

It is important to exploit any special structure of the matrix to increase speed of solution and decrease storage. We will consider only real matrices:

- s.p.d. matrices,
- symmetric indefinite matrices,
- band matrices,
- general sparse matrices,
- dense matrices depending on fewer than n^2 independent parameters.

Real Symmetric Positive Definite Matrices

Recall that a real matrix A is s.p.d. if and only if $A = A^T$ and $x^T Ax > 0$ for all $x \neq 0$. In this section we will show how to solve $Ax = b$ in half the time and half the space of Gaussian elimination when A is s.p.d.

PROPOSITION

1. *If X is nonsingular, then A is s.p.d. if and only if $X^T AX$ is s.p.d.*
2. *If A is s.p.d. and H is any principal submatrix of A($H = A(j:k, j:k)$ for some $j \leq k$), then H is s.p.d.*
3. *A is s.p.d. if and only if $A = A^T$ and all its eigenvalues are positive.*
4. *If A is s.p.d., then all $a_{ii} > 0$, and $\max_{ij} |a_{ij}| = \max_i a_{ii} > 0$.*
5. *A is s.p.d. if and only if there is a unique lower triangular nonsingular matrix L, with positive diagonal entries, such that $A = LL^T$. $A = LL^T$ is called the Cholesky factorization of A, and L is called the Cholesky factor of A.*

We may rewrite this induction as the following algorithm.

ALGORITHM *Cholesky algorithm:*

for $j = 1$ *to* n
  $l_{jj} = (a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2)^{1/2}$
  for $i = j + 1$ *to* n
    $l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk})/l_{jj}$
  end for
end for

If A is not positive definite, then (in exact arithmetic) this algorithm will fail by attempting to compute the square root of a negative number or by dividing by zero; this is the cheapest way to test if a symmetric matrix is positive definite.

The number of flops in Cholesky algorithm

In Cholesky algorithm L can overwrite the lower half of A. Only the lower half of A is referred to by the algorithm, so in fact only $n(n+1)/2$ storage is needed instead of n^2. The number of flops is

Number of operations in Cholesky algorithm

$$= \sum_{j=1}^{n} \left(2j + \sum_{i=j+1}^{n} 2j \right) = \frac{1}{3} n^3 + O(n^2). \tag{1}$$

The number of operations for LU decomposition is $\frac{2}{3} n^3 + O(n^2)$. Pivoting is not necessary for Cholesky to be numerically stable. Why? To explain this we perform the same analysis as for Gaussian elimination. We can show that we will have similar formula for error E in Cholesky decomposition as in the LU decomposition:

$$A = LL^T + E, \tag{2}$$

where error in Cholesky decomposition will be bounded as

$$|E| \leq n\epsilon |L| \cdot |L^T|.$$

Taking norms we get

$$\|E\| \leq n\epsilon \| \, |L| \, \| \cdot \| \, |L^T| \, \|.$$

We can rewrite expression above as

$$\|E\| \leq n\epsilon \|L\| \cdot \|L^T\|. \tag{3}$$

Thus, in formula (3) we have obtained error estimate in decomposition $A = L \cdot L^T$. To show how the error (2) can be obtained we again solve $L \underbrace{L^T x}_{y} = b$ via $Ly = b$ and $L^T x = y$. Solving $Ly = b$ gives as a computed solution $\hat{y}$ such that $(L + \delta L)\hat{y} = b$ where $|\delta L| \leq n\varepsilon |L|$. The same is true for $(L^T + \delta L^T)\hat{x} = \hat{y}$ with $|\delta L^T| \leq n\varepsilon |L^T|$. Combining both estimates into one we get

$$\begin{aligned} b = (L + \delta L)\hat{y} &= (L + \delta L)(L^T + \delta L^T)\hat{x} \\ &= (LL^T + L\delta L^T + \delta L L^T + \delta L \delta L^T)\hat{x} \\ &= (A \underbrace{-E + L\delta L^T + \delta L L^T + \delta L \delta L^T}_{\delta A})\hat{x} \\ &= (A + \delta A)\hat{x}. \end{aligned}$$

Recall

$$\delta A = -E + L\delta L^T + \delta L L^T + \delta L \delta L^T.$$

Now we combine all bounds for $E, \delta L^T, \delta L$ and use triangle inequality to get

$$\begin{aligned}
|\delta A| &\leq |-E + L\delta L^T + \delta L L^T + \delta L \delta L^T| \\
&\leq |E| + |L| \cdot |\delta L^T| + |\delta L| \cdot |L^T| + |\delta L| \cdot |\delta L^T| \\
&\leq n\varepsilon |L| \cdot |L^T| + n\varepsilon |L| \cdot |L^T| + n\varepsilon |L| \cdot |L^T| + n^2\varepsilon^2 |L| \cdot |L^T| \\
&\approx 3n\varepsilon |L| \cdot |L^T|.
\end{aligned}$$

Assuming that $\| \, |X| \, \| = \|X\|$ is true (as before for Frobenius, infinity, one-norms but not for two-norms) we obtain

$$||\delta A|| \leq 3n\varepsilon ||L|| \cdot ||L^T||. \tag{4}$$

Thus, from (4) follows that the computed solution $\tilde{x}$ satisfies $(A + \delta A)\tilde{x} = b$ with $|\delta A| \leq 3n\varepsilon |L| \cdot |L^T|$. But by the Cauchy-Schwartz inequality and proposition (part 4) we have that for every entry (i, j) of $|L| \cdot |L^T|$ we can write estimate

$$\begin{aligned} (|L| \cdot |L^T|)_{ij} &= \sum_k |l_{ik}| \cdot |l_{jk}| \\ &\leq \sqrt{\textstyle\sum_k l_{ik}^2} \sqrt{\textstyle\sum_k l_{jk}^2} \\ \leq \sqrt{a_{ii}} \cdot \sqrt{a_{jj}} &\leq \max_{ij} |a_{ij}| = \max_i a_{ii} > 0. \end{aligned}$$

Cholesky algorithm

```
for j = 1 to n
l_jj = (a_jj − Σ_{k=1}^{j−1} l_jk^2)^{1/2}
for i = j + 1 to n
l_ij = (a_ij − Σ_{k=1}^{j−1} l_ik l_jk)/l_jj
end for
end for
```

Then applying this estimate to all n entries of $|L| \cdot |L^T|$ we have

$$\| \, |L| \cdot |L^T| \, \|_\infty \leq n \|A\|_\infty. \tag{5}$$

Substituting (5) into (4) we get the following estimate

$$\|\delta A\|_\infty \leq 3n^2 \varepsilon \|A\|_\infty. \tag{6}$$

From it follows that Cholesky is stable when

$$\frac{\|\delta A\|_\infty}{\|A\|_\infty} \leq 3n^2 \varepsilon. \tag{7}$$

Symmetric Indefinite Matrices

- The question of whether we can still save half the time and half the space when solving a symmetric but indefinite (neither positive definite nor negative definite) linear system naturally arises. It turns out to be possible, but a more complicated pivoting scheme and factorization is required.
- If A is nonsingular, one can show that there exists a permutation P, a unit lower triangular matrix L, and a block diagonal matrix D with 1-by-1 and 2-by-2 blocks such that $PAP^T = LDL^T$.
- To see why 2-by-2 blocks are needed in D, consider the matrix $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. This factorization can be computed stably, saving about half the work and space compared to standard Gaussian elimination.

Band Matrices

A matrix A is called a *band matrix* with *lower bandwidth* b_L, and *upper bandwidth* b_U if $a_{ij} = 0$ whenever $i > j + b_L$ or $i < j - b_U$:

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1,b_U+1} & & 0 \\ \vdots & & & a_{2,b_U+2} & \\ a_{b_L+1,1} & & & & \ddots \\ & a_{b_L+2,2} & & & a_{n-b_U,n} \\ & & \ddots & & \vdots \\ 0 & & & a_{n,n-b_L} & \cdots \quad a_{n,n} \end{bmatrix}.$$

Band matrices arise often in practice and are useful to recognize because their L and U factors are also "essentially banded", making them cheaper to compute and store. We consider LU factorization without pivoting and show that L and U are banded in the usual sense, with the same band widths as A.

Example of a bandmatrix

The matrix A which appears after discretization of Laplacian in the Poisson's equation $-\triangle u = f$ is a bandmatrix. After discretization of laplacian we should solve system in the form $Au = b$. The vector b has the components $b_{i,j} = h^2 f_{i,j}$. The explicit elements of the matrix A are given by the following block matrix

$$A = \left(\begin{array}{c|c|c} A_N & -I_N & \\ \hline -I_N & \ddots & \ddots \\ \hline & \ddots & \ddots & -I_N \\ \hline & & -I_N & A_N \end{array}\right)$$

with blocks A_N of order N given by

$$A_N = \begin{pmatrix} 4 & -1 & 0 & 0 & \cdots & 0 \\ -1 & 4 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 4 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & \cdots & 0 & -1 & 4 \end{pmatrix},$$

Example: ODE

Example

Consider the ordinary differential equation (ODE) $y''(x) - p(x)y'(x) - q(x)y(x) = r(x)$ on the interval $[a, b]$ with boundary conditions $y(a) = \alpha$, $y(b) = \beta$. We also assume $q(x) \geq \underline{q} > 0$. This equation may be used to model the heat flow in a long, thin rod, for example. To solve the differential equation numerically, we *discretize* it by seeking its solution only at the evenly spaced mesh points $x_i = a + ih$, $i = 0, \ldots, N + 1$, where $h = (b - a)/(N + 1)$ is the mesh spacing. Define $p_i = p(x_i)$, $r_i = r(x_i)$, and $q_i = q(x_i)$.

Example

We need to derive equations to solve for our desired approximations $y_i \approx y(x_i)$, where $y_0 = \alpha$ and $y_{N+1} = \beta$. To derive these equations, we approximate the derivative $y\prime(x_i)$ by the following *finite difference approximation*:

$$y\prime(x_i) \approx \frac{y_{i+1} - y_{i-1}}{2h}.$$

(Note that as h gets smaller, the right-hand side approximates $y\prime(x_i)$ more and more accurately.) We can similarly approximate the second derivative by

$$y\prime\prime(x_i) \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}.$$

Inserting these approximations into the differential equation yields

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} - p_i \frac{y_{i+1} - y_{i-1}}{2h} - q_i y_i = r_i, \quad 1 \leq i \leq N.$$

Example

Multiplying by $-h^2/2$ we get:

$$-\frac{y_{i+1}}{2} + y_i - \frac{y_{i-1}}{2} + p_i\frac{y_{i+1}h}{4} - p_i\frac{y_{i-1}h}{4} + q_i y_i\frac{h^2}{2} = -\frac{-r_i h^2}{2}$$

Rewriting this as a linear system we get $Ay = b$, where

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}, \quad b = \frac{-h^2}{2}\begin{bmatrix} r_1 \\ \vdots \\ r_N \end{bmatrix} + \begin{bmatrix} (\frac{1}{2} + \frac{h}{4}p_1)\alpha \\ 0 \\ \vdots \\ 0 \\ (\frac{1}{2} - \frac{h}{4}p_N)\beta \end{bmatrix},$$

Example

and recalling

$$-\frac{y_{i+1}}{2} + y_i - \frac{y_{i-1}}{2} + p_i\frac{y_{i+1}h}{4} - p_i\frac{y_{i-1}h}{4} + q_i y_i\frac{h^2}{2} = -\frac{-r_i h^2}{2}$$

$$-\frac{y_{i+1}}{2} + y_i - \frac{y_{i-1}}{2} + p_i\frac{y_{i+1}h}{4} - p_i\frac{y_{i-1}h}{4} + q_i y_i\frac{h^2}{2} = -\frac{-r_i h^2}{2}$$

we have

$$A = \begin{bmatrix} a_1 & -c_1 & & \\ -b_2 & \ddots & \ddots & \\ & \ddots & \ddots & -c_{N-1} \\ & & -b_N & a_N \end{bmatrix}, \qquad \begin{aligned} a_i &= 1 + \tfrac{h^2}{2} q_i, \\ b_i &= \tfrac{1}{2}[1 + \tfrac{h}{2} p_i], \\ c_i &= \tfrac{1}{2}[1 - \tfrac{h}{2} p_i]. \end{aligned}$$

Note that $a_i > 0$ and also $b_i > 0$ and $c_i > 0$ if h is small enough. This is a nonsymmetric *tridiagonal* system to solve for y. We will show how to change it to a symmetric positive definite tridiagonal system, so that we may use *band Cholesky* to solve it.

Example

Choose $D = diag(1, \sqrt{\frac{c_1}{b_2}}, \sqrt{\frac{c_1 c_2}{b_2 b_3}}, \ldots, \sqrt{\frac{c_1 c_2 \cdots c_{N-1}}{b_2 b_3 \cdots b_N}})$. Then we may change $Ay = b$ to $(\underbrace{DAD^{-1}}_{\tilde{A}})\underbrace{(Dy)}_{\tilde{y}} = \underbrace{Db}_{\tilde{b}}$ or $\tilde{A}\tilde{y} = \tilde{b}$, where

$$\tilde{A} = \begin{bmatrix} a_1 & -\sqrt{c_1 b_2} & & \\ -\sqrt{c_1 b_2} & a_2 & -\sqrt{c_2 b_3} & \\ & -\sqrt{c_2 b_3} & \ddots & \\ & \ddots & \ddots & -\sqrt{c_{N-1} b_N} \\ & & -\sqrt{c_{N-1} b_N} & a_N \end{bmatrix}.$$

It is easy to see that $\tilde{A}$ is symmetric, and it has the same eigenvalues as A because A and $\tilde{A} = DAD^{-1}$ are *similar*. We will use the next theorem to show it is also positive definite.

Gershgorin's Theorem

THEOREM (Gershgorin's) *Let B be an arbitrary matrix. Then the eigenvalues λ of B are located in the union of the n disks*

$$|\lambda - b_{kk}| \leq \sum_{j \neq k} |b_{kj}|.$$

Proof. Given λ and $x \neq 0$ such that $Bx = \lambda x$, let $1 = ||x||_\infty = x_k$ by scaling x if necessary. Then $\sum_{j=1}^{N} b_{kj} x_j = \lambda x_k = \lambda$, so $\lambda - b_{kk} = \sum_{\substack{j=1 \\ j \neq k}}^{N} b_{kj} x_j$, implying

$$|\lambda - b_{kk}| \leq \sum_{j \neq k} |b_{kj} x_j| \leq \sum_{j \neq k} |b_{kj}||x_j| \leq \sum_{j \neq k} |b_{kj}| = R_k. \quad \square$$

Examples of using Gershgorin's circle theorem

Example

Example 1.
Use the Gershgorin circle theorem to estimate the eigenvalues of

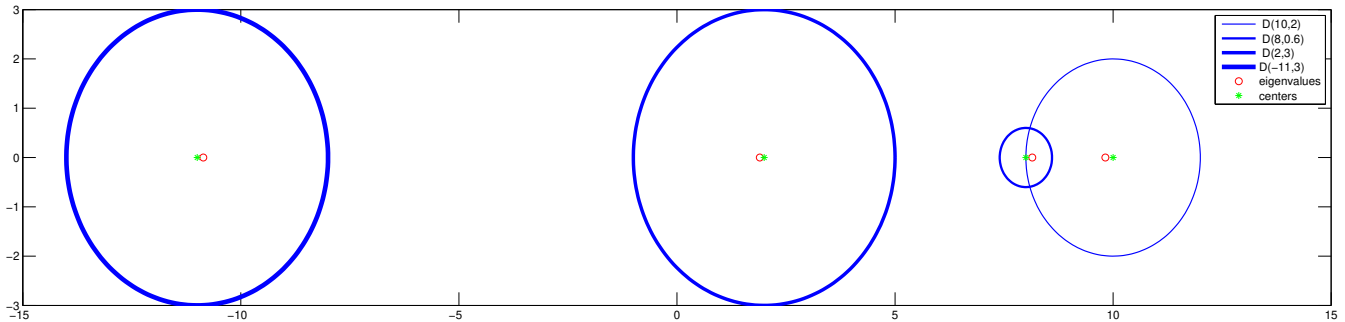
$$\mathrm{A} = \begin{bmatrix} 10 & -1 & 0 & 1 \\ 0.2 & 8 & 0.2 & 0.2 \\ 1 & 1 & 2 & 1 \\ -1 & -1 & -1 & -11 \end{bmatrix}.$$

Starting with row one, we take the element on the diagonal, a_{ii} as the center for the disc. We then take the remaining elements in the row and apply the formula:

$$\sum_{j \neq i} |a_{ij}| = R_i$$

to obtain the following four discs:
$D(10, 2), D(8, 0.6), D(2, 3),$ and $D(-11, 3)$
The eigenvalues are: 9.8218, 8.1478, 1.8995, -10.86

Example



Example 1

In example 1 the eigenvalues are: 9.8218, 8.1478, 1.8995, -10.86

Examples of using Gershgorin's circle theorem

Example

Example 2.
Use the Gershgorin circle theorem to estimate the eigenvalues of

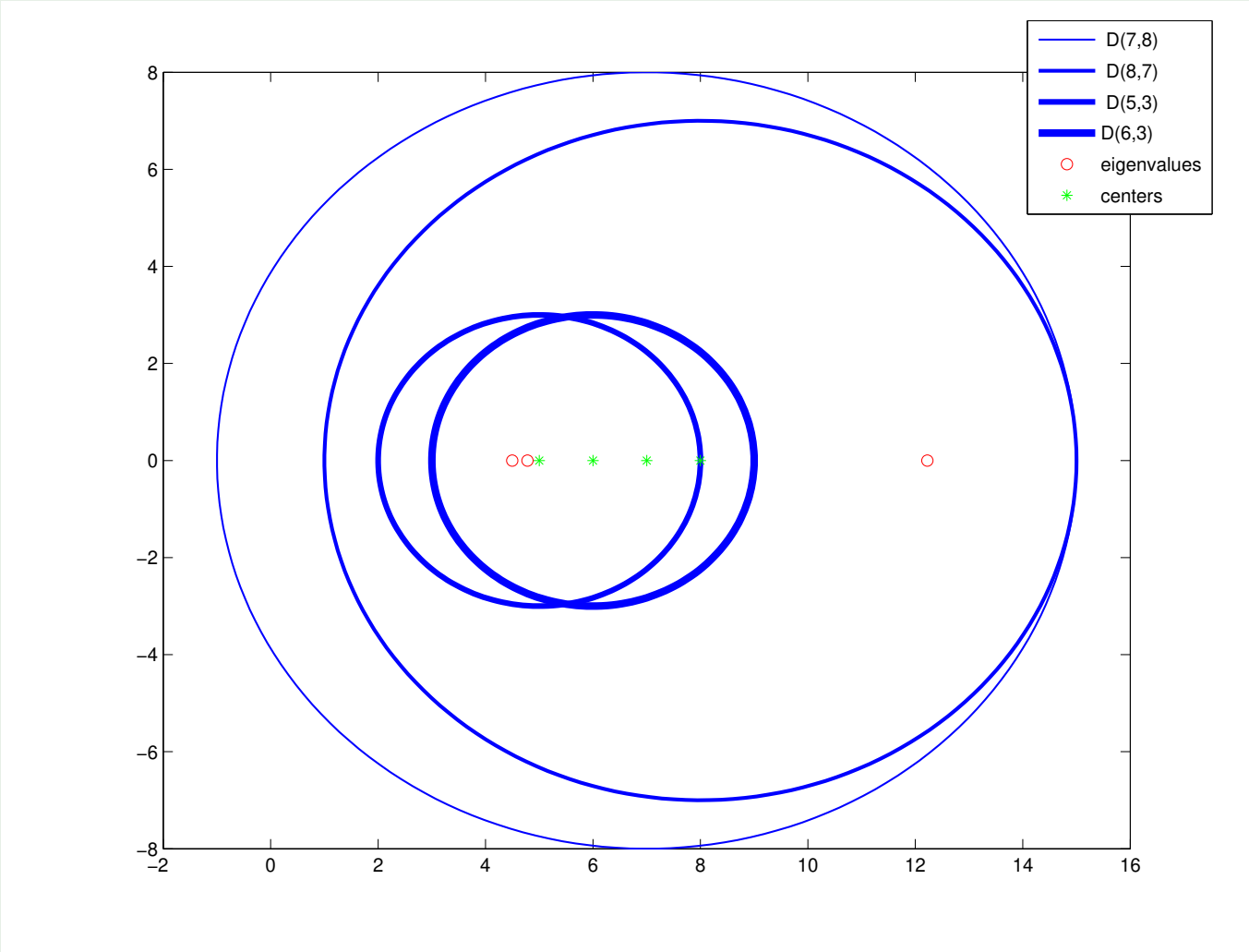
$$A = \begin{bmatrix} 7 & 5 & 2 & 1 \\ 2 & 8 & 3 & 2 \\ 1 & 1 & 5 & 1 \\ 1 & 1 & 1 & 6 \end{bmatrix}.$$

Starting with row one, we take the element on the diagonal, a_{ii} as the center for the disc. We then take the remaining elements in the row and apply the formula:

$$\sum_{j \neq i} |a_{ij}| = R_i$$

to obtain the following four discs:
$D(7,8), D(8,7), D(5,3), D(6,3)$
The eigenvalues are: 12.2249 + 0.0000i; 4.4977 + 0.6132i; 4.4977 - 0.6132i; 4.7797 + 0.0000i;

Example



Example 2

In example 2 the eigenvalues are: 12.2249 + 0.0000i; 4.4977 + 0.6132i; 4.4977 - 0.6132i; 4.7797 + 0.0000i.

Example: ODE (continuation)

Example

- If h is so small that for all i, $|\frac{h}{2}p_i| < 1$, then

$$|b_i|+|c_i| = \frac{1}{2}\left(1+\frac{h}{2}p_i\right)+\frac{1}{2}\left(1-\frac{h}{2}p_i\right) = 1 < 1+\frac{h^2}{2}\underline{q} \leq 1+\frac{h^2}{2}q_i = a_i.$$

  Therefore all eigenvalues of A lie inside the disks centered at $1 + h^2 q_i/2 \geq 1 + h^2\underline{q}/2$ with radius 1; in particular, they must all have positive real parts.

- Since $\tilde{A}$ is symmetric, its eigenvalues are real and hence positive, so $\tilde{A}$ is positive definite. Its smallest eigenvalue is bounded below by $\underline{q}h^2/2$.
- Thus, it can be solved by Cholesky.

Linear Least Squares Problems

- Suppose that we have a matrix A of the size $m \times n$ and the vector b of the size $m \times 1$. The linear least square problem is to find a vector x of the size $n \times 1$ which will minimize $||Ax - b||_2$.
- In the case when $m = n$ and the matrix A is nonsingular we can get solution to this problem as $x = A^{-1}b$.
- When $m > n$ (more equations than unknows) the problem is overdetermined
- When $m < n$ (more unknows than equations) the problem is underdetermined
- Applications: curve fitting, statistical modelling.

Matrix Factorizations that Solve the Linear Least Squares Problem

The linear least squares problem has several explicit solutions that we will discuss:

1. normal equations: the fastest but least accurate; it is adequate when the condition number is small.
2. QR decomposition,

   is the standard one and costs up to twice as much as the first method.
3. SVD,

   is of most use on an ill-conditioned problem, i.e., when A is not of full rank; it is several times more expensive again.
4. Iterative refinement to improve the solution when the problem is ill-conditioned. Can be adapted to deal efficiently with sparse matrices [Å. Björck. Numerical Methods for Least Squares Problems].

We assume initially for methods 1 and 2 that A has full column rank n.

Linear Least Squares Problems

Further we assume that we will deal with overdetermined problems when we have more equations than unknowns. This means that we will be interested in the solution of linear system of equations

$$Ax = b, \tag{8}$$

where A is of the size $m \times n$ with $m > n$, b is vector of the size m, and x is vector of the size n.
In a general case we are not able to get vector b of the size m as a linear combination of the n columns of the matrix A and n components of the vector x, or there is no solution to (8) in the usual case. In this chapter we will consider methods which can minimize the residual $r = b - Ax$ as a function on x in principle in any norm, but we will use 2-norm because of the convenience from theoretical (relationships of 2-norm with the inner product and orthogonality, smoothness and strict convexity properties) and computational points of view. Also, because of using 2-norm method is called least squares.

We can write the least squares problem as problem of the minimizing of the squared residuals

$$\|r\|_2^2 = \sum_{i=1}^{m} r_i^2 = \sum_{i=1}^{m} (Ax_i - b)^2. \tag{9}$$

In other words, our goal is to find minimum of this residual using least squares:

$$\min_x \|r\|_2^2 = \min_x \sum_{i=1}^{m} r_i^2 = \min_x \sum_{i=1}^{m} (Ax_i - b)^2. \tag{10}$$

Normal Equations

Our goal is to minimize $\|r(x)\|_2^2 = ||Ax - b||_2^2$. To find minimum we derive the *normal equations*: look for the x where the gradient of $||Ax - b||_2^2 = (Ax - b)^T(Ax - b)$ vanishes, or where $\|r'(x)\|_2^2 = 0$. So we want

$$\begin{aligned} 0 &= \lim_{e \to 0} \frac{(A(x+e) - b)^T(A(x+e) - b) - (Ax - b)^T(Ax - b)}{||e||_2} \\ &= \lim_{e \to 0} \frac{2e^T(A^TAx - A^Tb) + e^TA^TAe}{||e||_2} \end{aligned}$$

The second term $\frac{|e^TA^TAe|}{||e||_2} \leq \frac{||A||_2^2||e||_2^2}{||e||_2} = ||A||_2^2||e||_2$ approaches 0 as e goes to 0, so the factor $A^TAx - A^Tb$ in the first term must also be zero, or $A^TAx = A^Tb$. This is a system of n linear equations in n unknowns, the normal equations.

Data fitting

In this example we present the typical application of least squares called data or curve fitting problem. This problem appears in statistical modelling and experimental engineering when data are generated by laboratory or other measurements.
Suppose that we have data points $(x_i, y_i), i = 1, ..., m$, and our goal is to find the vector of parameters c of the size n which will fit best to the data y_i of the model function $f(x_i, c)$, where $f : R^{n+1} \to R$, in the least squares sense:

$$\min_c \sum_{i=1}^{m} (y_i - f(x_i, c))^2. \tag{11}$$

If the function $f(x, c)$ is linear then we can solve the problem (11) using least squares method.

The function $f(x, c)$ is linear if we can write it as a linear combination of the functions $\phi_j(x), j = 1, ..., n$ as:

$$f(x, c) = c_1\phi_1(x) + c_2\phi_2(x) + ... + c_n\phi_n(x). \tag{12}$$

Functions $\phi_j(x), j = 1, ..., n$ are called basis functions.
Let now the matrix A will have entries
$a_{ij} = \phi_j(x_i), i = 1, ..., m; j = 1, ..., n$, and vector b will be such that
$b_i = y_i, i = 1, ..., m$. Then a linear data fitting problem takes the form of (8) with $x = c$:

$$Ac \approx b \tag{13}$$

Elements of the matrix A are created by basis functions
$\phi_j(x), j = 1, ..., n$. We will consider now different examples of choosing basis functions $\phi_j(x), j = 1, ..., n$.