Applied Optimization: Formulation and Algorithms for Engineering Systems Slides

Ross Baldick

Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, TX 78712

Part I

Linear simultaneous equations

4

Case studies of linear simultaneous equations

(i) Solution of Kirchhoff's laws in a simple electrical circuit (Section 4.1), and

(ii) Search for a set of inputs to a "discrete-time linear system" that will bring the system to a desired state (Section 4.2).

4.1 Analysis of a direct current linear circuit

4.1.1 Motivation

- We want to calculate the behavior of a circuit.
- Circuits are characterized by **Kirchhoff's laws**.

4.1.2 Formulation

- Consider a circuit consisting of interconnected **resistors** and **current sources** as shown in Figure 4.1.
- A circuit can be thought of as a special type of graph where the branches are components.
- We want to:
  - calculate all the electrical quantities associated with the circuit, and
  - characterize how these quantities change if the circuit changes.

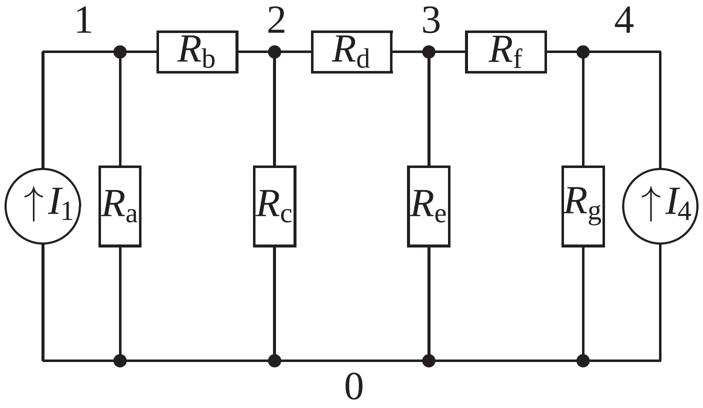


Fig. 4.1. A ladder circuit consisting of resistors and current sources.

4.1.2.1 Variables of interest

- Basic issue in problem formulation is to identify and distinguish:
  - the variables of interest from
  - the variables that are of less importance.
- This choice is one aspect of **Occam's razor**:
  - The model should be no more complicated than is necessary to represent the important issues, where "important" depends on our perspective.
- In typical circuits, we seek values of:
  - the voltages across the resistors and current sources, and
  - the currents through the resistors.
- We usually neglect most other quantities.
- For this problem, we could either:
  - use the **nodal voltages** as the independent variables and calculate the current flowing through each resistor in terms of the nodal voltages, or
  - use the **branch currents** as the independent variables and calculate the branch voltages in terms of the branch currents.
- A nodal based description will have less variables.

4.1.2.2 Kirchhoff's voltage law

- **Kirchhoff's voltage law** expresses the fact that the voltage around any loop is zero.
- This means that we can single out one of the nodes and call it the datum or ground node and measure all voltages with respect to the datum voltage.
- We write x_k for the voltage of node $k = 0, \dots, n$ with respect to the datum voltage, where $k = 0$ is the datum node.
- Kirchhoff's voltage law is an example of a **conservation law**.

4.1.2.3 Branch constitutive relations

- The **branch constitutive relations** express the relationship between branch current and voltage.
- For a resistor there is a linear relationship between resistor current and voltage.
- For a current source, the branch current is constant.

4.1.2.4 Kirchhoff's current law

- Kirchhoff's current law expresses conservation of charge when current is flowing in a circuit.
- The net current flowing from node 1 into the components incident to node 1 is:

$$\frac{x_1 - x_0}{R_a} + \frac{x_1 - x_2}{R_b} - I_1,$$

- Re-arranging:

$$\left(\frac{1}{R_a} + \frac{1}{R_b}\right) x_1 + \left(-\frac{1}{R_b}\right) x_2 = I_1. \tag{4.1}$$

$$\begin{aligned}
\left(-\frac{1}{R_b}\right) x_1 + \left(\frac{1}{R_b} + \frac{1}{R_c} + \frac{1}{R_d}\right) x_2 + \left(-\frac{1}{R_d}\right) x_3 &= 0, & (4.2)\\
\left(-\frac{1}{R_d}\right) x_2 + \left(\frac{1}{R_d} + \frac{1}{R_e} + \frac{1}{R_f}\right) x_3 + \left(-\frac{1}{R_f}\right) x_4 &= 0, & (4.3)\\
\left(-\frac{1}{R_f}\right) x_3 + \left(\frac{1}{R_f} + \frac{1}{R_g}\right) x_4 &= I_4. & (4.4)
\end{aligned}$$

4.1.2.5 Nodal admittance matrix and voltage and current vector

- Define:
$$
\begin{aligned}
A &= \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix}, \\
&= \begin{bmatrix} \frac{1}{R_a}+\frac{1}{R_b} & -\frac{1}{R_b} & 0 & 0 \\ -\frac{1}{R_b} & \frac{1}{R_b}+\frac{1}{R_c}+\frac{1}{R_d} & -\frac{1}{R_d} & 0 \\ 0 & -\frac{1}{R_d} & \frac{1}{R_d}+\frac{1}{R_e}+\frac{1}{R_f} & -\frac{1}{R_f} \\ 0 & 0 & -\frac{1}{R_f} & \frac{1}{R_f}+\frac{1}{R_g} \end{bmatrix}, \quad (4.5) \\
x &= \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}. \quad (4.6)
\end{aligned}
$$

- The matrix A is called the **nodal admittance matrix**.
- The variable x_0 is not included in our definition of the vector x.

Nodal admittance matrix and voltage and current vector, continued

$$b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} I_1 \\ 0 \\ 0 \\ I_4 \end{bmatrix}. \tag{4.7}$$

4.1.2.6 Linear equations

$$Ax = \begin{bmatrix} \sum_{k=1}^{4} A_{1k}x_k \\ \sum_{k=1}^{4} A_{2k}x_k \\ \sum_{k=1}^{4} A_{3k}x_k \\ \sum_{k=1}^{4} A_{4k}x_k \end{bmatrix}.$$

- If we write $Ax = b$, we reproduce the nodal equations (4.1)–(4.4) for our system.
- We call A the **coefficient matrix**, while b is called the **right-hand side.**

4.1.3 Changes

- In our ladder circuit, there are two types of circuit components that can change:
  - either the currents from the current sources vary, corresponding to a change in the right-hand side of the linear system, b, or
  - the resistances vary, corresponding to a change in the coefficient matrix of the system, A.
- For each type of component or parameter change, we can consider two related notions of change:

  (i) infinitesimal changes in component or parameter values, providing a **sensitivity analysis**, and

  (ii) **large changes** in component values or parameters.

4.1.3.1 Sensitivity

- **Sensitivity analysis**, sometimes called **small-signal sensitivity analysis** in the context of circuit theory, is the calculation of a partial derivative of the solution $x^\star$, or a function of the solution, with respect to some parameter.
- For example, we might want to calculate the partial derivative of the solution for a particular voltage, say x_2, with respect to the value of:
  - a current source, say I_4, to obtain $\frac{\partial x_2^\star}{\partial I_4}$,
  - a resistor, say R_b, to obtain $\frac{\partial x_2^\star}{\partial R_\text{b}}$, or
  - the temperature, T, to obtain $\frac{\partial x_2^\star}{\partial T}$.

Sensitivity, continued

- As another example of a sensitivity analysis, we may also want to consider the sensitivity of a **performance criterion** or **objective** function to variations in parameters.
- For example, consider the function $f : \mathbb{R}^4 \to \mathbb{R}$ defined by:

$$\forall x \in \mathbb{R}^4, f(x) = (x_1)^2 + 2(x_2)^2 + 3(x_3)^2 + 4(x_4)^2. \tag{4.8}$$

- We might want to calculate the derivative of $f^\star(\bullet) = f(x^\star(\bullet))$, with respect to the value of:
  - a current source, say I_4, to obtain $\frac{\partial f^\star}{\partial I_4}$,
  - a resistor, say R_b, to obtain $\frac{\partial f^\star}{\partial R_\text{b}}$, or
  - the temperature, T, to obtain $\frac{\partial f^\star}{\partial T}$.

4.1.3.2 Large changes

Change in current source

- Consider a variation in the current injection b_ℓ at node ℓ by an amount Δb_ℓ
- The new circuit must satisfy $Ax' = b + \Delta b$.

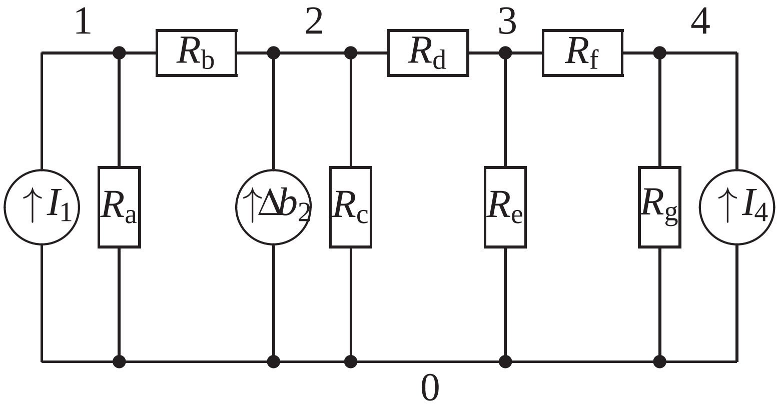


Fig. 4.2. The ladder circuit of Figure 4.1 with a change, Δb_ℓ, in the current injected at node $\ell = 2$.

Change in resistance

- Consider a variation in the resistance of the resistor joining nodes ℓ and k.
- The change in A is ΔA, where ΔA has zeros everywhere except in the $\ell\ell$-th, ℓk-th, $k\ell$-th, and kk-th entries.
- The new circuit must satisfy $(A+\Delta A)x' = b$.

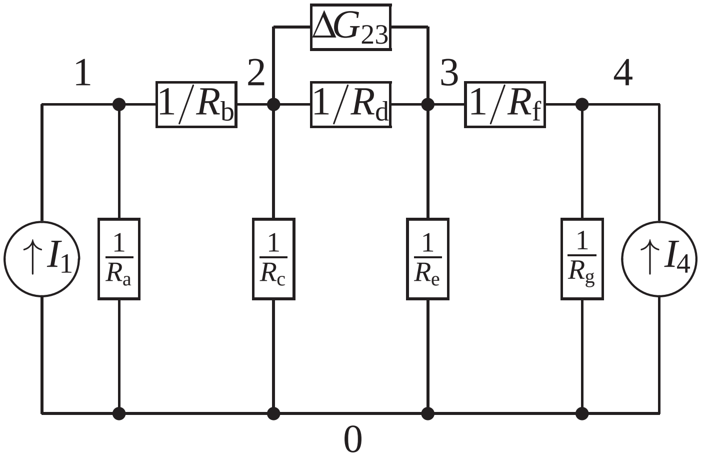


Fig. 4.3. The ladder circuit of Figure 4.1 with resistors re-labeled with their conductances and with a change in the conductance between nodes $\ell = 2$ and $k = 3$. Note that the convention for labeling the resistors has changed compared to the previous figures.

4.1.4 Problem characteristics

4.1.4.1 Numbers of variables and equations

- Since A is a square matrix, we call $Ax = b$ a **square system of linear equations.**

4.1.4.2 Solvability

- Since this circuit always has a unique solution, $Ax = b$ is solvable for any given b and there is only one solution.

4.1.4.3 Admittance matrix

$$\forall \ell = 1,\ldots,n, \forall k = 1,\ldots,n,$$

$$A_{\ell k} = \begin{cases} \text{sum of the conductances connected to node } \ell, \\ \hfill \text{if } \ell = k, \\ \\ \text{minus the conductance joining } \ell \text{ and } k, \\ \text{if } \ell \neq k \text{ and there is a resistor between } \ell \text{ and } k, \\ \\ \hfill 0, \\ \text{if } \ell \neq k \text{ and there is no resistor between } \ell \text{ and } k. \end{cases} \quad (4.9)$$

Symmetry

- A is **symmetric**.

Sparsity

- In a large ladder circuit, the A matrix would be mostly zeros.
- We call a matrix **sparse** if most of its entries are zero.
- By choosing as datum node the node with the most branches incident to it, we will minimize the number of non-zeros in the admittance matrix.

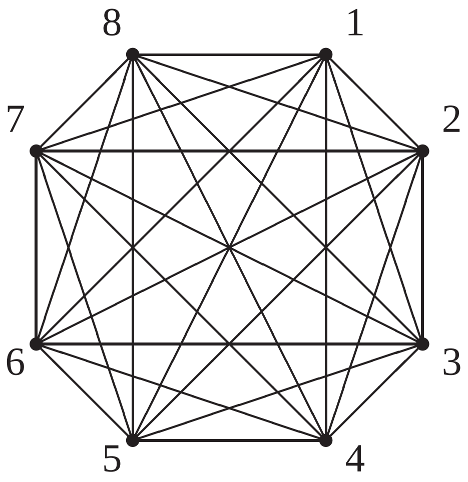


Fig. 4.4. A graph with $n = 8$ nodes and all $\frac{n(n-1)}{2} = 28$ possible branches. For clarity in this graph, each node is represented by a bullet ●, while each branch is represented by a line. This is a different convention to that used in figures 4.1–4.3.

Diagonal dominance

- Entry $A_{\ell\ell}$ in A is greater than the sum of the absolute values of the other entries in the ℓ-th column of A.
- Such a matrix is called **strictly diagonally dominant**.

Changes in the admittance matrix

- For a change in a resistor between nodes ℓ and k:

$$\Delta A = \Delta G_{\ell k} \left[\begin{array}{cccc} \overbrace{}^{\ell\text{-th column}} & \overbrace{}^{k\text{-th column}} & & \\ & & & \\ 1 & -1 & & \} \; \ell\text{-th row} \\ & & & \\ -1 & 1 & & \} \; k\text{-th row} \\ & & & \end{array} \right]. \tag{4.10}$$

- For a change in a resistor between node ℓ and the datum node:

$$\Delta A = \Delta G_{\ell 0} \left[\begin{array}{ccc} \overbrace{}^{\ell\text{-th column}} & & \\ & & \\ 1 & & \} \; \ell\text{-th row} \\ & & \end{array} \right]. \tag{4.11}$$

4.2 Control of a discrete-time linear system

4.2.1 Motivation

- Investigate the conditions under which we can shift the **state** of the system to a desired final value by adjusting the inputs over a sequence of time intervals.
- That is, we are going to consider the **open loop** control of the system.

Motivation, continued

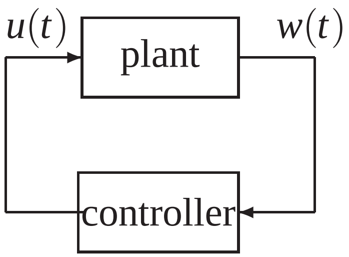


Fig. 4.5. A feedback control system applied to a plant.

- In a **feedback controller**, as illustrated in Figure 4.5, we use the output (or the state) of the system to decide on the controls.
- Furthermore, in **optimal control** we recognize the costs of certain control actions and states.
- These pose somewhat different problems to the one we investigate in this case study.
- However, several of the issues turn out to be similar.

4.2.2 Formulation

4.2.2.1 Variables

- The **state** of the system is the smallest set of variables, $w \in \mathbb{R}^m$, say, such that:
  - w includes all the variables of interest as a sub-vector, and
  - knowledge of the value of w for any particular time $t = t_0$, together with knowledge of the values of the input $u(t)$ to the plant for $t_1 \geq t \geq t_0$ completely specifies the value of w for any time $t_1 \geq t \geq t_0$.
- We write $w(kT)$ for the value of w at the k-th sampling instant.
- We assume that the input stays constant between the k-th and $(k+1)$-th sampling instant so that $u(t) = u(kT), kT \leq t < (k+1)T$.

4.2.2.2 Behavior of system

- By the definition of state, the value of the state at time kT and the value of the input for period k determines value of the state at time $(k+1)T$.
- That is, for each k, there exists a function $\phi^{(k)} : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^m$:

$$\forall k \in \mathbb{Z}, w((k+1)T) = \phi^{(k)}(w(kT), u(kT)),$$

Linear:
$$\forall k \in \mathbb{Z}, \forall w \in \mathbb{R}^m, \forall u \in \mathbb{R}, \phi^{(k)}(w,u) = G^{(k)}w + h^{(k)}u,$$

Time-invariant:
$$\forall k \in \mathbb{Z}, w((k+1)T) = \phi(w(kT), u(kT)),$$

Linear time-invariant:
$$\forall k \in \mathbb{Z}, \forall w \in \mathbb{R}^m, \forall u \in \mathbb{R}, \phi^{(k)}(w,u) = Gw + hu.$$

- Linear time-invariant systems behave according to the **difference equation:**

$$\forall k \in \mathbb{Z}, w([k+1]T) = Gw(kT) + hu(kT). \tag{4.12}$$

- G is called the **state transition matrix**.

4.2.2.3 Changing the state of the system

- At time $kT = 0$ the plant is in state $w(0) \in \mathbb{R}^m$ and we would like it instead to be in some other desired final state $w^{\text{desired}} \in \mathbb{R}^m$.
- That is, $w(nT) = w^{\text{desired}}$.

$$
\begin{aligned}
w(nT) &= Gw([n-1]T) + hu([n-1]T), \\
&\qquad \text{on substituting for } w(nT) \text{ from (4.12) for } k = n-1, \\
&= (G)^2 w([n-2]T) + Ghu([n-2]T) + hu([n-1]T), \\
&\qquad \text{on substituting for } w([n-1]T) \text{ from (4.12) for } k = n-2, \\
&= (G)^3 w([n-3]T) + (G)^2 hu([n-3]T) \\
&\qquad + Ghu([n-2]T) + hu([n-1]T), \text{ continuing}, \\
&\vdots \quad \vdots \\
&= (G)^n w(0) + \sum_{k=0}^{n-1} (G)^{n-1-k} hu(kT).
\end{aligned}
$$

Changing the state of the system, continued

- Define:

$$
\begin{aligned}
A &= \left[(G)^{n-1}h \quad (G)^{n-2}h \quad \cdots \quad Gh \quad h \right], \\
x &= \begin{bmatrix} u(0T) \\ u(1T) \\ \vdots \\ u([n-2]T) \\ u([n-1]T) \end{bmatrix}, \\
b &= w^{\text{desired}} - (G)^n w(0).
\end{aligned}
$$

Changing the state of the system, continued

- Then:

$$\begin{aligned} Ax &= \left[(G)^{n-1}h \;\; (G)^{n-2}h \;\; \cdots \;\; Gh \;\; h \right] \begin{bmatrix} u(0T) \\ u(1T) \\ \vdots \\ u([n-2]T) \\ u([n-1]T) \end{bmatrix}, \\ &= \sum_{k=0}^{n-1} (G)^{n-1-k} h u(kT). \end{aligned}$$

- $w(nT)$ will be equal to w^{desired} if:

$$w^{\text{desired}} = (G)^n w(0) + Ax.$$

- That is:

$$Ax = b. \tag{4.13}$$

4.2.2.4 Example

- Suppose that $n = 2$, $m = 2$, and:

$$\begin{aligned} h &= \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \\ G &= \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}, \\ w(0) &= \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \\ w^{\text{desired}} &= \begin{bmatrix} 3 \\ 7 \end{bmatrix}. \end{aligned}$$

Example, continued

- Then:

$$\begin{aligned} A &= \begin{bmatrix} Gh & h \end{bmatrix}, \\ &= \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \\ b &= w^{\text{desired}} - (G)^n w(0), \\ &= \begin{bmatrix} -1 \\ 0 \end{bmatrix}. \end{aligned}$$

- Solving for x, we obtain:

$$\begin{aligned} \begin{bmatrix} u(0T) \\ u(1T) \end{bmatrix} &= x, \\ &= \begin{bmatrix} -1 \\ 1 \end{bmatrix}. \end{aligned}$$

Example, continued

- For this particular example, it is also possible to find a control $u(0T)$ that achieved the desired final state in one time-step; that is, for $n = 1$.
- In particular, there is a solution $u(0T) \in \mathbb{R}$ to:

$$hu(0T) = w^{\text{desired}} - Gw(0),$$

  namely $u(0T) = 3$.
- However, it will typically require more than one time-step to achieve a desired state.

4.2.2.5 Labeling of vector and matrix entries

- The columns of A are labeled from 0 to $(n-1)$.
- This contrasts with the labeling of the variables in the case study in Section 4.1 where the entries were labeled from 1 to n.
- In general, we can label the entries of a vector in any way we choose.

4.2.3 Changes

4.2.3.1 Initial and desired state

- If w^{desired} or $w(0)$ change, then the right-hand side b in the linear equation (4.13) will also change correspondingly.

4.2.3.2 System

- If the behavior of the plant changes, then the state transition matrix G and therefore the coefficient matrix A in (4.13) will change.

4.2.4 Problem characteristics

4.2.4.1 Numbers of variables and equations

- The number of variables is n, which is equal to the number of entries in x, but the number of equations is equal to m, which is the number of entries in b.

4.2.4.2 Solvability

- It is not always the case that (4.13) is solvable.
- Solvability will depend on G, h, w^{desired}, $w(0)$, and on n.

4.2.4.3 Coefficient matrix

- The coefficient matrix is not symmetric and has different numbers of rows and columns.

5

Algorithms for linear simultaneous equations

- Consider generally how to solve large systems of the form:

$$Ax = b. \tag{5.1}$$

- A is called the **coefficient matrix**, while b is called the **right-hand side vector**.

- First consider special cases of coefficient matrices:

$$\textbf{Upper triangular}:\ U = \begin{bmatrix} 2 & 3 & 4 \\ 0 & -\frac{9}{2} & -9 \\ 0 & 0 & 1 \end{bmatrix}, \tag{5.2}$$

$$\textbf{Lower triangular}:\ L = \begin{bmatrix} 1 & 0 & 0 \\ \frac{7}{2} & 1 & 0 \\ 4 & \frac{2}{3} & 1 \end{bmatrix}. \tag{5.3}$$

Key issues

- Solution of **triangular systems** and **factorization of matrices,**
- **computational effort** and particular features of problems, such as **symmetry** and **sparsity** that can reduce the necessary computational effort,
- **sensitivity analysis** and **ill-conditioning**,
- solution of **non-square systems**.

5.1 Inversion of coefficient matrix

- Suppose that A is **invertible** with inverse A^{-1}.
- Let $x = A^{-1}b$.
- Then:

$$\begin{aligned} Ax &= AA^{-1}b, \\ &= \mathbf{I}b, \text{ by definition of inverse,} \\ &= b, \text{ by definition of } \mathbf{I}. \end{aligned}$$

- Cramér's rule says that $k\ell$-th entry of A^{-1} is given by:

  $(-1)^{\ell+k}$ times

  the determinant of the matrix obtained from A by deleting its ℓ-th row and k-th column, divided by

  the determinant of A.

- Computational effort is on the order of $n!$ arithmetic operations.
- If n is large then Cramér's rule is impractical because the calculation of determinants is too computationally intensive.

Inversion of coefficient matrix, continued

- Nevertheless, Cramér's rule can be extremely useful for:
  - proving properties of matrices,
  - inverting small matrices, since Cramér's rule allows the inverse to be written down explicitly. For example, for $A \in \mathbb{R}^{2\times 2}$, if $A_{11}A_{22} - A_{12}A_{21} \neq 0$ then A is invertible and:

$$A^{-1} = \frac{1}{A_{11}A_{22} - A_{12}A_{21}} \begin{bmatrix} A_{22} & -A_{12} \\ -A_{21} & A_{11} \end{bmatrix}, \qquad (5.4)$$

  - inverting specific types of matrices.

5.2 Solution of triangular systems

5.2.1 Forwards substitution

5.2.1.1 Analysis

- If L is lower triangular then $L_{\ell k} = 0, \forall \ell < k$.
- Suppose we want to find $y^\star \in \mathbb{R}^n$ satisfying $Ly = b$.
- (We will see in Section 5.3.1 the reason for choosing y as the decision variable instead of x.)

$$\begin{aligned}
b_1 &= L_{11}y_1, \\
b_2 &= L_{21}y_1 + L_{22}y_2, \\
b_3 &= L_{31}y_1 + L_{32}y_2 + L_{33}y_3, \\
\vdots &\quad \vdots \\
b_n &= L_{n1}y_1 + L_{n2}y_2 + L_{n3}y_3 + \cdots + L_{nn}y_n.
\end{aligned}$$

Analysis, continued

- Re-arranging:

$$
\begin{aligned}
y_1 &= \frac{b_1}{L_{11}}, \\
y_2 &= \frac{b_2 - L_{21}y_1}{L_{22}}, \\
y_\ell &= \frac{b_\ell - \sum_{k=1}^{\ell-1} L_{\ell k} y_k}{L_{\ell\ell}}. \qquad (5.5)
\end{aligned}
$$

- This process is called **forwards substitution**.

5.2.1.2 Example

$$
\begin{aligned}
L &= \begin{bmatrix} 1 & 0 & 0 \\ \frac{7}{2} & 1 & 0 \\ 4 & \frac{2}{3} & 1 \end{bmatrix}, \\
b &= \begin{bmatrix} 9 \\ 18 \\ 28 \end{bmatrix}, \\
y^\star &= \begin{bmatrix} 9 \\ -\frac{27}{2} \\ 1 \end{bmatrix}.
\end{aligned}
$$

5.2.2 Backwards substitution

5.2.2.1 Analysis

- If U is upper triangular then $U_{\ell k} = 0, \forall \ell > k$
- Suppose that $y \in \mathbb{R}^n$ is given and we want to solve $Ux = y$.

$$\begin{array}{ccccrcl}
U_{11}x_1 & + & \cdots & + & U_{1,n-2}x_{n-2} + U_{1,n-1}x_{n-1} + U_{1,n}x_n & = & y_1, \\
 & & & & \vdots & & \vdots \\
 & & & & U_{n-2,n-2}x_{n-2} + U_{n-2,n-1}x_{n-1} + U_{n-2,n}x_n & = & y_{n-2}, \\
 & & & & U_{n-1,n-1}x_{n-1} + U_{n-1,n}x_n & = & y_{n-1}, \\
 & & & & U_{n,n}x_n & = & y_n.
\end{array}$$

Analysis

- Re-arranging:

$$\begin{aligned} x_n &= \frac{y_n}{U_{n,n}}, \\ x_{n-1} &= \frac{y_{n-1} - U_{n-1,n}x_n}{U_{n-1,n-1}}, \\ x_\ell &= \frac{y_\ell - \sum_{k=\ell+1}^{n} U_{\ell k}x_k}{U_{\ell\ell}}. \end{aligned}$$

- This process is called **backwards substitution**.

5.2.2.2 Example

$$
\begin{aligned}
U &= \begin{bmatrix} 2 & 3 & 4 \\ 0 & -\frac{9}{2} & -9 \\ 0 & 0 & 1 \end{bmatrix}, \\
y &= \begin{bmatrix} 9 \\ -\frac{27}{2} \\ 1 \end{bmatrix}, \\
x^\star &= \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.
\end{aligned}
$$

5.2.3 Computational effort

5.2.3.1 Forwards substitution

- Forwards substitution calculates $y_\ell, \ell = 1, \dots, n$.
- Calculation of y_1 requires a division.
- Calculation of each y_ℓ for $\ell = 2, \dots, n$ requires: $(\ell - 1)$ multiplications, $(\ell - 2)$ additions, a subtraction, and a division.
- In total, this is:

$$\begin{aligned}
\sum_{\ell=2}^{n} (\ell - 1) &= \frac{1}{2}(n-1)n \text{ multiplications,} \\
\sum_{\ell=2}^{n} (\ell - 2) &= \frac{1}{2}(n-2)(n-1) \text{ additions,} \\
& \quad (n-1) \text{ subtractions,} \\
& \quad n \text{ divisions.}
\end{aligned}$$

5.2.3.2 *Backwards substitution*

- Backwards substitution calculates $x_\ell, \ell = n, \ldots, 1$.
- Calculation of x_n requires a division.
- Calculation of each x_ℓ for $\ell = (n-1), \ldots, 1$ requires: $(n-\ell)$ multiplications, $(n-\ell-1)$ additions, a subtraction, and a division.
- In total, this is:

$$\begin{aligned}\sum_{\ell=1}^{n-1}(n-\ell) &= \frac{1}{2}(n-1)n \text{ multiplications},\\ \sum_{\ell=1}^{n-1}(n-\ell-1) &= \frac{1}{2}(n-2)(n-1) \text{ additions},\\ & \quad (n-1) \text{ subtractions},\\ & \quad n \text{ divisions}.\end{aligned}$$

5.2.3.3 *Overall*

- Overall effort is on the order of the *square* of the number of variables.

5.3 Solution of square, non-singular systems

5.3.1 Combining forwards and backwards substitution

- Suppose that we can factorize $A \in \mathbb{R}^{n\times n}$ into LU, with L lower triangular and U upper triangular:

$$\begin{aligned} b &= Ax, \text{ the equation we want to solve,}\\ &= LUx, \text{ since } A = LU,\\ &= L(Ux),\\ &= Ly, \end{aligned}$$

- where $y = Ux$.
- We have transformed the problem of solving $Ax = b$ into the solution of three successive problems:
  - (i) factorization of A into LU,
  - (ii) forwards substitution to solve $Ly = b$, and
  - (iii) backwards substitution to solve $Ux = y$.
- If A is singular then we cannot factorize A into LU with L and U having non-zero diagonal entries.

5.3.2 LU factorization

- We will specify a series of “stages” to implement the algorithm.
- $M^{(j)}$ represents a matrix that is defined in the j-th stage of the algorithm.
- To factorize A, we will multiply it on the left by the non-singular matrices $M^{(1)}, M^{(2)}, \dots, M^{(n-1)}$ such that the matrix $U = M^{(n-1)}M^{(n-2)} \cdots M^{(1)}A$ is upper triangular.
- At each stage, the product:

$$A^{(j+1)} = M^{(j)}M^{(j-1)} \cdots M^{(1)}A,$$

  will become successively “closer” to being upper triangular.
- We will choose the $M^{(j)}$, $j = 1, \dots, n-1$ to have two additional properties:

  (i) each $M^{(j)}$ will be lower triangular and therefore have a lower triangular inverse, and

  (ii) $[M^{(j)}]^{-1}$, the inverse of $M^{(j)}$, will be easy to compute.

LU factorization, continued

- We let:

$$\begin{aligned} L &= [M^{(n-1)} \cdots M^{(1)}]^{-1}, \\ &= [M^{(1)}]^{-1} \cdots [M^{(n-1)}]^{-1}. \end{aligned}$$

- That is, L is the product of $(n-1)$ lower triangular matrices and, therefore, L is also lower triangular.

$$\begin{aligned} LU &= [M^{(n-1)} \cdots M^{(1)}]^{-1} M^{(n-1)} \cdots M^{(1)} A, \text{ by definition,} \\ &= A, \end{aligned}$$

- so that A has been factorized into LU.

5.3.2.1 *First stage*

Pivoting

- In the first stage of the algorithm, we let:

$$M^{(1)} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ -L_{21} & 1 & \ddots & & \vdots \\ -L_{31} & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ -L_{n1} & 0 & \cdots & 0 & 1 \end{bmatrix}, \tag{5.6}$$

- where $L_{\ell 1} = A_{\ell 1}/A_{11}, \ell = 2, \ldots, n$.
- Define $A^{(2)} = M^{(1)}A$.

Pivoting, continued

$$A^{(2)} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ 0 & A_{22}^{(2)} & \cdots & A_{2n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & A_{n2}^{(2)} & \cdots & A_{nn}^{(2)} \end{bmatrix}, \tag{5.7}$$

$$A_{\ell k}^{(2)} = A_{\ell k} - L_{\ell 1} A_{1k}, 1 < \ell, k \leq n.$$

$$\begin{aligned} A_{\ell 1}^{(2)} &= A_{\ell 1} - L_{\ell 1} A_{11}, \text{ for } 1 < \ell \leq n, \\ &= A_{\ell 1} - \frac{A_{\ell 1}}{A_{11}} A_{11}, \\ &= A_{\ell 1} - A_{\ell 1}, \\ &= 0. \end{aligned}$$

- We have zeroed the entries in the first column of A below its first entry.
- We say that we have **pivoted** on the entry A_{11}.

Pivoting, continued

- Note that:

$$\begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ -L_{21} & 1 & \ddots & & \vdots \\ -L_{31} & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ -L_{n1} & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ L_{21} & 1 & \ddots & & \vdots \\ L_{31} & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ L_{n1} & 0 & \cdots & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix},$$

- so that:

$$[M^{(1)}]^{-1} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ L_{21} & 1 & \ddots & & \vdots \\ L_{31} & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ L_{n1} & 0 & \cdots & 0 & 1 \end{bmatrix}.$$

Small or zero pivot

- The construction will fail if $A_{11} = 0$.
- If A_{11} is small in magnitude compared to $A_{\ell 1}$ then $L_{\ell 1} = A_{\ell 1}/A_{11}$ will be large in magnitude.
- For moderate to large values of A_{1k} this will mean that the product $L_{\ell 1}A_{1k}$ can be large compared to $A_{\ell k}$.
- If so, $A_{\ell k} - L_{\ell 1}A_{1k}$ will have an error, due to round-off error, that is large compared to $A_{\ell k}$.
- That is, the calculated value $A_{\ell k}^{(2,\mathrm{calc})}$ differs from the exact value by $A_{\ell k}^{(2,\mathrm{error})}$:

$$A_{\ell k}^{(2,\mathrm{calc})} = A_{\ell k} - L_{\ell 1}A_{1k} + A_{\ell k}^{(2,\mathrm{error})}. \tag{5.8}$$

Error analysis

- We can re-arrange (5.8) to:

$$A_{\ell k}^{(2,\mathrm{calc})} = (A_{\ell k} + A_{\ell k}^{(2,\mathrm{error})}) - L_{\ell 1}A_{1k},$$

- where we now imagine the error as being in the original entry of A.

Permuting rows and columns

- If $A_{11} = 0$ (or if A_{11} is small in magnitude), but $A_{\ell k} \neq 0$ for some ℓ and k then we can reorder the rows and columns and pivot on $A_{\ell k}$ instead.
- This simply corresponds to permuting:
  - equation ℓ is re-numbered to be equation 1, and
  - variable k is re-numbered to be variable 1.
- This approach is called **full pivoting**.

Partial pivoting

- Instead of permuting both rows and columns we can only permute, say, rows.
- The permutation of the rows can be represented by multiplying A on the left by a permutation matrix $P \in \mathbb{R}^{n \times n}$:

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

Diagonal pivoting

- Equation ℓ is re-numbered to be equation 1, and
- Variable ℓ is re-numbered to be variable 1.

Summary

- In the first stage of the algorithm, to calculate $A^{(2)}$ using A_{11} as pivot, we:
  - copy the first row of A into $A^{(2)}$;
  - zero the entries in the first column of $A^{(2)}$ below the diagonal; and
  - explicitly calculate the entries $A^{(2)}_{\ell k}$ for $1 < \ell \leq n, 1 < k \leq n$ using $A^{(2)}_{\ell k} = A_{\ell k} - L_{\ell 1} A_{1k}$.
- We call A_{11} the **standard pivot**.

5.3.2.2 Second stage

Pivoting

- In the second stage of the algorithm, we now choose $M^{(2)}$ to zero the second column of $A^{(2)}$ below the diagonal:

$$M^{(2)} = \begin{bmatrix} 1 & 0 & & \cdots & & 0 \\ 0 & 1 & \ddots & & & \\ & -L_{32} & 1 & & & \vdots \\ \vdots & -L_{42} & 0 & \ddots & \ddots & \\ & \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & -L_{n2} & 0 & \cdots & 0 & 1 \end{bmatrix}, \qquad (5.9)$$

- where $L_{\ell 2} = A^{(2)}_{\ell 2}/A^{(2)}_{22}, \ell = 3, \ldots, n.$

Pivoting, continued

- Let $A^{(3)} = M^{(2)}M^{(1)}A = M^{(2)}A^{(2)}$:

$$A^{(3)} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1n} \\ 0 & A_{22}^{(2)} & A_{23}^{(2)} & \cdots & A_{2n}^{(2)} \\ 0 & 0 & A_{33}^{(3)} & \cdots & A_{3n}^{(3)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & A_{n3}^{(3)} & \cdots & A_{nn}^{(3)} \end{bmatrix}, \tag{5.10}$$

$$A_{\ell k}^{(3)} = A_{\ell k}^{(2)} - L_{\ell 2}A_{2k}^{(2)}, 2 < \ell, k \le n.$$

$$[M^{(2)}]^{-1} = \begin{bmatrix} 1 & 0 & & \cdots & & 0 \\ 0 & 1 & \ddots & & & \\ & L_{32} & 1 & & & \vdots \\ \vdots & L_{42} & 0 & 1 & \ddots & \\ & \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & L_{n2} & 0 & \cdots & 0 & 1 \end{bmatrix}.$$

Error analysis

- As in Section 5.3.2.1, we can interpret round-off errors in the calculation of $A^{(3)}$ in terms of a perturbation introduced into $A^{(2)}$, which we can, in turn, interpret in terms of a perturbation in the original matrix A.

Permuting rows and columns

- The construction may again fail if $A_{22}^{(2)} = 0$.
- Again, full, partial, or diagonal pivoting can be used if there is a suitable non-zero pivot $A_{\ell k}$ for some $2 \leq \ell \leq n$ and $2 \leq k \leq n$.

Summary

- At the second stage of the algorithm, to calculate $A^{(3)}$ using $A_{22}^{(2)}$ as pivot, we:
  - copy the first two rows of $A^{(2)}$ into $A^{(3)}$;
  - zero the entries in the first two columns of $A^{(3)}$ below the diagonal; and
  - explicitly calculate the entries $A_{\ell k}^{(3)}$ for $2 < \ell \leq n, 2 < k \leq n$ using $A_{\ell k}^{(3)} = A_{\ell k}^{(2)} - L_{\ell 2} A_{2k}^{(2)}$.
- We call $A_{22}^{(2)}$ the standard pivot.

5.3.2.3 Subsequent stages

Pivot

$$\begin{aligned} \forall \ell > j, L_{\ell j} &= A_{\ell j}^{(j)} / A_{jj}^{(j)}, \\ \forall \ell > j, \forall k > j, A_{\ell k}^{(j+1)} &= A_{\ell k}^{(j)} - L_{\ell j} A_{jk}^{(j)}. \end{aligned} \quad (5.11)$$

Error analysis

- At each stage, errors in $A^{(j+1)}$ can be interpreted in terms of a perturbation in $A^{(j)}$, which can be interpreted in terms of a perturbation in the original matrix A.

Summary

- At stage j of the algorithm, to calculate $A^{(j+1)}$ using $A^{(j)}_{jj}$ as pivot, we:
  - copy the first j rows of $A^{(j)}$ into $A^{(j+1)}$;
  - zero the entries in the first j columns of $A^{(j+1)}$ below the diagonal; and
  - explicitly calculate the entries $A^{(j+1)}_{\ell k}$ for $j < \ell \leq n, j < k \leq n$ using $A^{(j+1)}_{\ell k} = A^{(j)}_{\ell k} - L_{\ell j} A^{(j)}_{jk}$.
- We call $A^{(j)}_{jj}$ the standard pivot.
- Again, if $A^{(j)}_{jj} = 0$ or if it is small in magnitude, then the rows and/or columns of $A^{(j)}$ can be reordered to place a suitable non-zero entry $A^{(jj)}_{\ell k}$, where $j \leq \ell \leq n$ and $j \leq k \leq n$, in the jj place.

5.3.2.4 *Last stage*

$$U = A^{(n)} = M^{(n-1)}M^{(n-2)}\cdots M^{(1)}A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & \cdots & A_{1n} \\ 0 & A_{22}^{(2)} & A_{23}^{(2)} & \cdots & A_{2n}^{(2)} \\ 0 & 0 & A_{33}^{(3)} & \cdots & A_{3n}^{(3)} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & A_{nn}^{(n)} \end{bmatrix}.$$

$$L = \left[M^{(n-1)}\cdots M^{(1)}\right]^{-1} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ L_{21} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ L_{n1} & \cdots & L_{n,n-1} & 1 \end{bmatrix}.$$

5.3.2.5 Example

- $A = \begin{bmatrix} 2 & 3 & 4 \\ 7 & 6 & 5 \\ 8 & 9 & 11 \end{bmatrix}$,
- $b = \begin{bmatrix} 9 \\ 18 \\ 28 \end{bmatrix}$.

$$M^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{7}{2} & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix}, A^{(2)} = M^{(1)}A = \begin{bmatrix} 2 & 3 & 4 \\ 0 & -\frac{9}{2} & -9 \\ 0 & -3 & -5 \end{bmatrix}.$$

$$M^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{2}{3} & 1 \end{bmatrix}, U = A^{(3)} = \begin{bmatrix} 2 & 3 & 4 \\ 0 & -\frac{9}{2} & -9 \\ 0 & 0 & 1 \end{bmatrix}, L = \begin{bmatrix} 1 & 0 & 0 \\ \frac{7}{2} & 1 & 0 \\ 4 & \frac{2}{3} & 1 \end{bmatrix}.$$

- $x^\star = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$.

5.3.2.6 *Singular matrices*

- Factorization can sometimes be performed on singular matrices.
- However, if factorization fails then (under the assumption of infinite precision calculations) the matrix is singular.

5.3.3 *Computational effort*

- At the j-th stage, we calculate:
  - the $(n-j)$ entries of L that are in its j-th column and lying below the diagonal,
  - the $(n-j)^2$ values of $A^{(j+1)}$ that are in the lower right of the matrix.
- The total effort therefore is:

$$\begin{aligned}
\sum_{j=1}^{n-1}(n-j) &= \frac{1}{2}n(n-1) \text{ divisions,} \\
\sum_{j=1}^{n-1}(n-j)^2 &= \frac{1}{6}(2n-1)n(n-1) \text{ multiplications,} \\
\sum_{j=1}^{n-1}(n-j)^2 &= \frac{1}{6}(2n-1)n(n-1) \text{ subtractions.}
\end{aligned}$$

- The overall effort for LU factorization is therefore on the order of the *cube* of the number of variables.

5.3.4 Variations

5.3.4.1 Factorization in place

- To implement the LU factorization algorithm, we can start with a copy of A and apply the pivot operations directly to update the entries in the copy of A, thereby transforming it into the LU factors.
- The entries of the lower triangle of L can be entered into the lower triangle of A as they are calculated, while the entries of the diagonal and upper triangle of U can be entered into the diagonal and upper triangle of A as they are calculated.

5.3.4.2 *Diagonal entries of L and U*

- L has ones on its diagonal, while the entries on the diagonal of U were the pivots.
- We can instead factorize A into two matrices L' and U' so that U' has ones on its diagonal, while the entries on the diagonal of L' are the pivots.

5.3.4.3 *LDU factorization*

- Suppose we factorize A into LU'.
- Let D have diagonal entries equal to the diagonal entries of U'
- Define $U = D^{-1}U'$.
- We now have a factorization of A into LDU, where D is a diagonal matrix and both L and U have ones on the diagonal.

5.4 Symmetric coefficient matrix

- If A is symmetric, we can save approximately half the work in factorization, so long as we only use diagonal pivots.
- Symmetric systems often arise in circuit applications, as we have seen, and also occur in optimization applications.
- Sometimes, a system that appears at first to be not symmetric can be made symmetric by **scaling** the rows or columns or re-arranging the rows or columns.

5.4.1 LU factorization

Lemma 5.1 *Suppose that A is symmetric and diagonal pivoting was used in the first stage of factorization to reorder rows and columns. Then:*

(i) *the first row of A is equal to A_{11} times the transpose of the first column of L, (that is, the entries in the first column of L arranged into a row), and*

(ii) *the submatrix of $A^{(2)}$ formed by deleting its first row and column is symmetric.*

Proof The proof involves calculation of the entries $A^{(2)}$. $\square$

Lemma 5.2 *Let $2 \leq j \leq (n-1)$ and consider the matrix $A^{(j)}$ formed at the $(j-1)$-th stage of the factorization. Suppose that the submatrix of $A^{(j)}$ obtained by deleting its first $(j-1)$ rows and $(j-1)$ columns is symmetric. Assume that diagonal pivoting is used at the j-th stage of factorization. Consider the matrix $A^{(j+1)}$ formed at the j-th stage of factorization. Then:*

(i) *the j-th row of $A^{(j+1)}$ is equal to $A^{(j)}_{jj}$ times the transpose of the j-th column of L, (that is, the entries in the j-th column of L arranged into a row), and*

(ii) *the submatrix of $A^{(j+1)}$ formed by deleting its first j rows and j columns is also symmetric.*

Proof The proof is analogous to that of Lemma 5.1. $\square$

Corollary 5.3 *Suppose that A is symmetric and that diagonal pivoting is used at each stage of the factorization. Then for each j, $2 \leq j \leq n$, the submatrix of $A^{(j)}$ formed by deleting its first $(j-1)$ rows and $(j-1)$ columns is symmetric. Moreover, at the end of the factorization, for each ℓ, the ℓ-th row of U is equal to $U_{\ell\ell}$ times the transpose of the ℓ-th column of L.*

Proof By induction. Lemma 5.1 proves the result for $j = 1$. Lemma 5.2 then proves the induction step. □

5.4.2 Example

- $A = \begin{bmatrix} 2 & 3 & 4 \\ 3 & 5 & 7 \\ 4 & 7 & 13 \end{bmatrix}$

5.4.2.1 First stage

- $M^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{3}{2} & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}$
- $A^{(2)} = M^{(1)}A = \begin{bmatrix} 2 & 3 & 4 \\ 0 & \frac{1}{2} & 1 \\ 0 & 1 & 5 \end{bmatrix}$
- $A^{(2)}_{32} = A^{(2)}_{23}$

5.4.2.2 Second stage

- $M^{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix}$
- $A^{(3)} = M^{(2)} M^{(1)} A = \begin{bmatrix} 2 & 3 & 4 \\ 0 & \frac{1}{2} & 1 \\ 0 & 0 & 3 \end{bmatrix}.$

5.4.2.3 Last stage

- $L = [M^{(1)}]^{-1} [M^{(2)}]^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{3}{2} & 1 & 0 \\ 2 & 2 & 1 \end{bmatrix}.$

5.4.3 Computational savings

- The savings in computational effort for the small example in Section 5.4.2 is modest.
- For a general matrix, calculating only the diagonal and upper triangle in the first stage of factorization saves a fraction of the work equal to:

$$\begin{aligned}
\frac{(n-1)^2-(n-1)n/2}{(n-1)^2} &= \frac{(n-1)^2/2-(n-1)/2}{(n-1)^2}, \\
&= \frac{n-2}{2(n-1)}, \\
&\approx \frac{1}{2}, \text{ for } n \text{ large.}
\end{aligned}$$

- Similarly, at each successive stage approximately half the work is saved, so that overall approximately half the computational effort is required compared to factorizing a non-symmetric matrix.

5.4.4 $LDL^\dagger$ and Cholesky factorization

- Suppose a symmetric matrix is factorized into LU using diagonal pivots at each stage of the factorization.
- Let D be a diagonal matrix with entries equal to the diagonal of U.
- By Corollary 5.3, $U = DL^\dagger$.
- That is, $A = LDL^\dagger$ with the diagonal entries of D being the pivots.
- If the entries of D are all positive, let $R = D^{\frac{1}{2}}L^\dagger$, where the matrix $D^{\frac{1}{2}}$ is diagonal with each diagonal entry equal to the positive square root of the corresponding entry of D.
- Then $A = R^\dagger R$ is called the **Cholesky factorization** of A.
- The matrix R is upper triangular.

5.4.5 Discussion of diagonal pivoting

- In the circuit case study from Section 4.1, the admittance matrix is strictly diagonally dominant and so the diagonal entries are relatively large compared to the off-diagonal.
- Diagonal pivoting is consequently adequate for the particular problem in our case study.
- In other circuit formulations and more generally in other applications, this may not be the case and off-diagonal pivoting becomes necessary.

5.4.6 Positive definite coefficient matrix

Lemma 5.4 *Suppose that $A \in \mathbb{R}^{n\times n}$ is symmetric and can be factorized as $A = LDL^\dagger$, with $D \in \mathbb{R}^{n\times n}$ diagonal and L lower triangular with ones on the diagonal. Then A is positive definite if and only if all the diagonal entries of D are strictly positive.*

Proof

$\Rightarrow$ We first prove that A being positive definite implies that the diagonal entries of D are strictly positive. To prove this, we prove the contra-positive. So, suppose that there is at least one diagonal entry, $D_{\ell\ell}$, say, of D that is not strictly positive. We will exhibit $x \neq \mathbf{0}$ such that $x^\dagger A x \leq 0$. To find such a x, solve the equation $L^\dagger x = \mathbf{I}_\ell$ for x. (This is possible since L is lower triangular and has ones on its diagonal. We just perform backwards substitution on $L^\dagger$.) Notice that $x \neq \mathbf{0}$, for else $\mathbf{I}_\ell = L^\dagger x = L^\dagger \mathbf{0} = \mathbf{0}$, which is a contradiction. Furthermore,

$$\begin{aligned}
x^\dagger A x &= x^\dagger L D L^\dagger x, \text{ by assumption on } A, \\
&= \mathbf{I}_\ell{}^\dagger D \mathbf{I}_\ell, \text{ by definition of } x, \\
&= D_{\ell\ell}, \text{ on direct calculation}, \\
&\leq 0, \text{ by supposition.}
\end{aligned}$$

Therefore, A is not positive definite.

$\Leftarrow$ We now prove that the diagonal entries of D being strictly positive implies that A is positive definite. So, suppose that all the diagonal entries of D are strictly positive. Define the matrix $D^{\frac{1}{2}}$ to be diagonal with each diagonal entry $\left[D^{\frac{1}{2}}\right]_{\ell\ell}$ equal to the positive square root of the corresponding diagonal entry of D. That is, $\left[D^{\frac{1}{2}}\right]_{\ell\ell} = \sqrt{D_{\ell\ell}}, \forall \ell$. Let $x \neq \mathbf{0}$ be given and define $y = D^{\frac{1}{2}} L^{\dagger} x$. We first claim that $y \neq \mathbf{0}$. For suppose the contrary. That is, suppose that $y = \mathbf{0}$. Then, $\left[D^{\frac{1}{2}}\right]^{-1} y = \mathbf{0}$. (Notice that the diagonal entries of $D^{\frac{1}{2}}$ are all strictly positive, so that $D^{\frac{1}{2}}$ is invertible.) But then $\mathbf{0} = \left[D^{\frac{1}{2}}\right]^{-1} y = L^{\dagger} x$. Solving $L^{\dagger} x = \mathbf{0}$ by backwards substitution we obtain $x = \mathbf{0}$, a contradiction. Therefore, $y \neq \mathbf{0}$.

⇐ **continued** Second, we observe that:

$$\begin{aligned} x^\dagger A x &= x^\dagger L D L^\dagger x, \text{ by assumption on } A, \\ &= x^\dagger L D^{\frac{1}{2}} D^{\frac{1}{2}} L^\dagger x, \text{ by definition of } D^{\frac{1}{2}}, \\ &= y^\dagger y, \text{ by definition of } y, \\ &= \|y\|_2^2, \text{ by definition of } \|\bullet\|_2, \\ &> 0, \text{ since the length of a non-zero vector is strictly positive.} \end{aligned}$$

That is, A is positive definite. □

Theorem 5.5 *If A is symmetric and positive definite, then:*

(i) *A is invertible,*

(ii) *A is factorizable as $LDL^\dagger$, with D diagonal having strictly positive diagonal entries and L lower triangular with ones on the diagonal, and*

(iii) *A^{-1} is also symmetric and positive definite.*

$\square$

5.4.7 *Indefinite coefficient matrix*

- Consider:

$$\mathcal{A} = \begin{bmatrix} A & B \\ B^\dagger & C \end{bmatrix}.$$

- Suppose that A is a square symmetric matrix that is positive semi-definite or positive definite and that C is a square symmetric matrix that is negative semi-definite or negative definite.
- The coefficient matrix $\mathcal{A}$ is **indefinite**; that is, it is neither positive semi-definite nor negative semi-definite.
- For example:

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

- For a non-singular indefinite matrix there are special purpose factorization techniques.

5.5 Sparsity techniques

- The non-zero entries in the admittance matrix occur only:
  - on the diagonal, and
  - at those off-diagonal entries corresponding to resistors,
- so that the admittance matrix is a **sparse matrix**.
- We may also have right-hand side vectors b that only have a few non-zero entries.

5.5.1 Sparse storage

5.5.1.1 Sparse matrices

- Store only *values* and *locations* of the non-zero entries in the matrix.

$$A = \begin{bmatrix} 1 & 2 & 0 & 5 \\ 2 & 1 & 3 & 0 \\ 0 & 3 & 1 & 4 \\ 5 & 0 & 4 & 1 \end{bmatrix} . \tag{5.12}$$

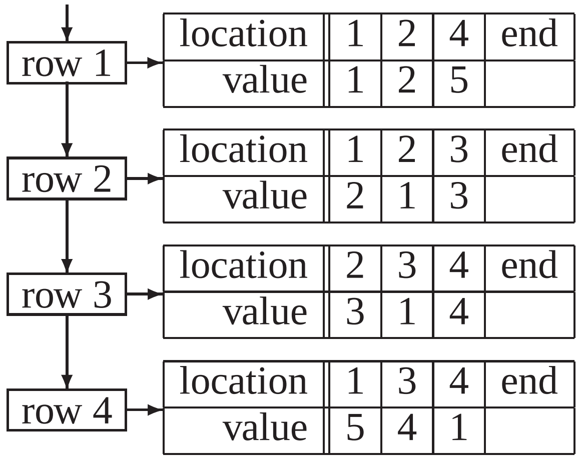


Fig. 5.1. Sparse matrix storage by rows of the matrix (5.12).

5.5.1.2 Sparse vectors

- **Sparse vectors** can be stored as a list of pairs of numbers representing the locations and values of the non-zero entries of the vector.
- For example, consider the change in the circuit case study of Section 4 illustrated in Figure 4.2, which is repeated for reference in Figure 5.2.
- In this circuit, the current injected at node 2 changes by Δb_2.

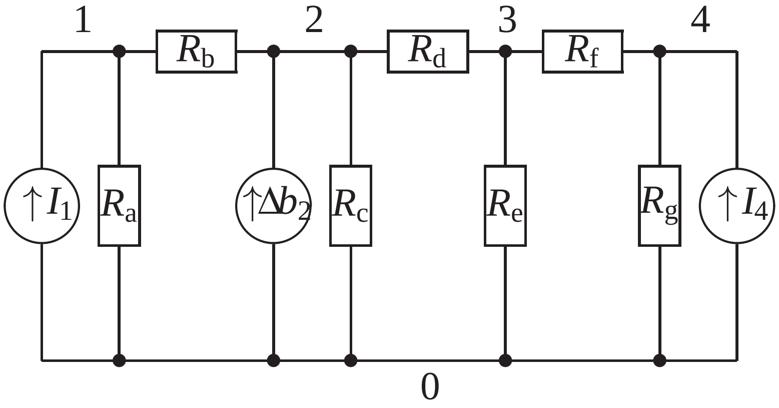


Fig. 5.2. The ladder circuit of Figure 4.2, showing a change, Δb_ℓ, in the current injected at node $\ell = 2$.

Sparse vectors, continued

- Suppose that the value of the change in the current source was $\Delta b_2 = 1$.
- Then, we could define a vector $\Delta b \in \mathbb{R}^4$ that represents the changes at all nodes as specified by:

$$\Delta b = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}. \tag{5.13}$$

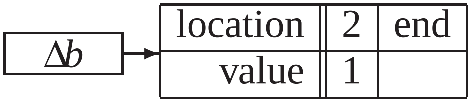


Fig. 5.3. Sparse storage of the vector (5.13).

5.5.1.3 Implementation

- A linked list of records can be easily modified by changing the pointers.

5.5.2 Forwards and backwards substitution

5.5.2.1 Sparse matrices

- Forward substitution to solve $Ly = b$:

$$\begin{aligned} y_\ell &= \frac{1}{L_{\ell\ell}}\left(b_\ell - \sum_{k=1}^{\ell-1} L_{\ell k} y_k\right), \\ &= b_\ell - \sum_{\substack{k<\ell \\ L_{\ell k}\neq 0 \\ y_k \neq 0}} L_{\ell k} y_k. \end{aligned} \tag{5.14}$$

- To calculate y_ℓ, we first initialize $y_\ell = b_\ell$.
- For each non-zero entry $L_{\ell k}, k < \ell$, in row ℓ, if $y_k \neq 0$, we calculate $L_{\ell k} y_k$ and subtract it from the current value of y_ℓ.

5.5.2.2 Sparse vectors

- If $b = \mathbf{I}_n$ then we only need perform (5.14) for $\ell = n$ since $y_\ell = 0$ for $\ell < n$.

5.5.3 Factorization

5.5.3.1 Fill-ins

$$A = \begin{bmatrix} 1 & 2 & 0 & 5 \\ 2 & 1 & 3 & 0 \\ 0 & 3 & 1 & 4 \\ 5 & 0 & 4 & 1 \end{bmatrix}.$$

- We can represent the zeros and non-zeros of A with the following diagram:

$$\begin{array}{|cccc|} \hline \circ & \circ & & \circ \\ \circ & \circ & \circ & \\ & \circ & \circ & \circ \\ \circ & & \circ & \circ \\ \hline \end{array}.$$

Fill-ins, continued

- Similarly, we can represent the zeros and non-zeros of L with:

$$\boxed{\begin{array}{cccc} \circ & & & \\ \circ & \circ & & \\ & \circ & \circ & \\ \circ & \bullet & \circ & \circ \end{array}},$$

- where $\circ$ and $\bullet$ both represent non-zeros:
  - $\circ$ corresponds to an entry that was non-zero in A, while
  - $\bullet$ corresponds to an entry that was zero in A.
- We refer to the latter entries, indicated by bullets $\bullet$, as **fill-ins** because they correspond to a non-zero entry in L that was created at a position of a zero in A.

5.5.3.2 Choosing pivots to minimize fill-ins

- Fill-ins necessitate later calculations.
- We seek an ordering of the rows and columns of the matrix that minimizes the number of fill-ins during factorization.

Heuristic criteria

- It is in general very difficult to find the optimal ordering to minimize the total number of fill-ins created during the complete factorization.
- Several heuristics available to approximately minimize the number of fill-ins created.
- Choose the pivot at stage j so as to minimize the number of fill-ins created at stage j, ignoring the effect of this decision on the number of fill-ins created at later stages.

Number of fill-ins with standard pivot

- We have the following upper bound on the number of fill-ins:

Lemma 5.6 *Suppose that* $A \in \mathbb{R}^{n\times n}$ *is symmetric. Let* $N(j)$ *be the number of fill-ins created at stage* j *of factorization using* $A_{jj}^{(j)}$ *as pivot. Then* $N(j) \leq \overline{N}(j)$*, where:*

$\overline{N}(j) = [$*(the number of non-zero entries in the* j*-th row of* $A^{(j)}$*) minus 1*$]^2$,

$\square$

- The upper bound $\overline{N}(j)$ is very easy to evaluate and represents the worst possible case of creation of fill-ins where every non-zero entry in the j-th row of $A^{(j)}$ creates a fill-in for every one of the non-zero elements in the j-th column of $A^{(j)}$ below the diagonal that must be explicitly annihilated.
- That is, it ignores the entries that are already non-zero in $A^{(j)}$.

Number of fill-ins with other pivots

- Define $N(\ell)$ to be the number of fill-ins created at stage j if we pivot on the entry $A_{\ell\ell}^{(j)}$
- Again, we can approximate $N(\ell)$ by ignoring the entries that are already non-zero in $A^{(j)}$ to obtain the upper bound $\overline{N}(\ell)$.
- $\overline{N}(\ell)$ is equal to the square of one less than the number of non-zero entries in the ℓ-th row of $A^{(j)}$.

Application of heuristic

- We choose to pivot on the entry $A_{\ell\ell}^{(j)}$ that minimizes $\overline{N}(\ell)$, which will also approximately minimize $N(\ell)$.
- That is, we pick the row ℓ of $A^{(j)}$, where $j \leq \ell \leq n$, that has the least number of non-zero entries.

5.5.3.3 Computational effort

- The computational effort for factorization is difficult to calculate exactly because it depends on the total number of fill-ins.
- However, the effort for factorization is typically much less than cubic in the number of variables.
- In practice, elapsed computation time sometimes grows only slightly faster than *linearly* in the number of variables.
- Solution time depends strongly on the number of non-zero entries in the A matrix.
- A very large, but sparse, system can be faster to solve than a small dense system having more non-zeros than the sparse system.

5.5.3.4 Other criteria for pivot selection

- We should try to avoid small pivots.
- This presents difficulties for our LU factorization algorithm for sparse matrices, because we would like to know the order of the pivots ahead of time so that we can create the appropriate fill-ins in the linked list representation of the matrix.

5.5.4 Special types of sparse matrices

5.5.4.1 Banded matrices and matrices with regular structure

- A **banded matrix** has zeros everywhere except on the diagonal and on entries that are close to the diagonal.
- A **tri-diagonal matrix** is a banded matrix that has non-zero entries only on the diagonal and adjacent to the diagonal.
- There are special factorization algorithms that have been developed for these types of matrices.

5.5.4.2 Block pivoting and sparsity

- Consider factorization of the block matrix:

$$
\begin{aligned}
\mathcal{A} &= \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \\
\mathcal{M}^{(1)} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -CA^{-1} & \mathbf{I} \end{bmatrix}, \\
\mathcal{A}^{(2)} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -CA^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \\
&= \begin{bmatrix} A & B \\ \mathbf{0} & D - CA^{-1}B \end{bmatrix}.
\end{aligned}
\tag{5.15}
$$

Block pivoting and sparsity, continued

- The pre-multiplication matrix $\mathcal{M}^{(1)}$ was obtained by "pretending" that the matrix:

$$\mathcal{A} = \begin{bmatrix} A & B \\ C & D \end{bmatrix},$$

- was a 2×2 matrix and pivoting on the block A.
- The first block column of $\mathcal{L}$ in the block $\mathcal{L}\mathcal{U}$ factorization of $\mathcal{A} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ is given by $\begin{bmatrix} \mathbf{I} \\ CA^{-1} \end{bmatrix}$ and the first block row of $\mathcal{U}$ is given by $\begin{bmatrix} A & B \end{bmatrix}$.
- We say that we have **pivoted on the block** A.
- We have that:

$$\begin{aligned} \mathcal{A} &= \mathcal{L}\mathcal{U}, \\ &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ CA^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} A & B \\ \mathbf{0} & D - CA^{-1}B \end{bmatrix}. \end{aligned}$$

Block pivoting and sparsity, continued

- If the sparsity pattern of the system is such that non-zero entries occur in blocks, then it can be more efficient to store the matrix as a sparse collection of blocks.
- The “entries” of the coefficient matrix will therefore consist of blocks and block pivoting can be used.
- For example, Figure 5.4 shows the storage of such a matrix.
- As in Figure 5.1, the matrix is stored by “rows;” however, in this case the rows are actually pairs of rows in the matrix and the entries are 2×2 blocks.

Block pivoting and sparsity, continued

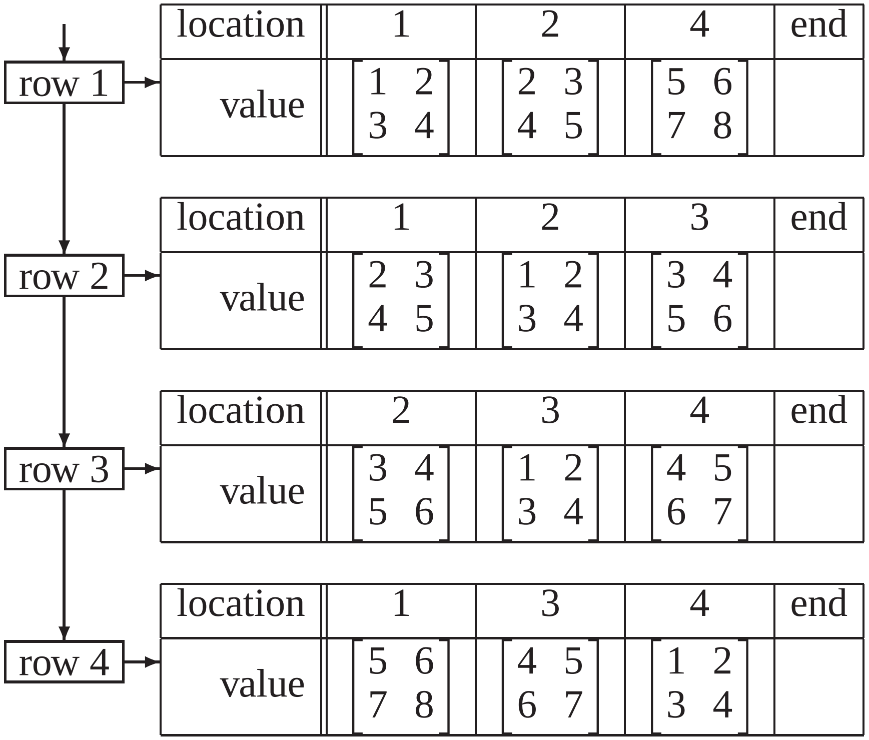


Fig. 5.4. Storage by block rows for a block matrix.

5.6 Changes

5.6.1 Sensitivity

5.6.1.1 Analysis

- We now generalize to the case where the coefficient matrix and right-hand sides are functions of a parameter $\chi \in \mathbb{R}^s$.
- That is, $A : \mathbb{R}^s \to \mathbb{R}^{n \times n}$ and $b : \mathbb{R}^s \to \mathbb{R}^n$ are matrix and vector valued functions of χ, respectively.
- We assume that we have already found $x = x^{\star\star} \in \mathbb{R}^n$ that satisfied $A(\mathbf{0})x = b(\mathbf{0})$.

Theorem 5.7 *Suppose that $A : \mathbb{R}^s \to \mathbb{R}^{n\times n}$ and $b : \mathbb{R}^s \to \mathbb{R}^n$ are partially differentiable with continuous partial derivatives and that $A(\mathbf{0})$ is non-singular. Then, there exists a function $x^\star : \mathbb{R}^s \to \mathbb{R}^n$ such that:*

- *for χ in a neighborhood of $\mathbf{0}$, the function $x^\star$ satisfies the linear simultaneous equations $A(\chi)x^\star(\chi) = b(\chi)$, and*
- *the function $x^\star$ is partially differentiable in the neighborhood with partial derivative with respect to χ_j at $\chi = \mathbf{0}$ given by:*

$$\frac{\partial x^\star}{\partial \chi_j}(\mathbf{0}) = [A(\mathbf{0})]^{-1}\left[\frac{\partial b}{\partial \chi_j}(\mathbf{0}) - \frac{\partial A}{\partial \chi_j}(\mathbf{0})x^{\star\star}\right], \tag{5.16}$$

where $x^{\star\star} \in \mathbb{R}^n$ satisfies the base-case linear simultaneous equations $A(\mathbf{0})x^{\star\star} = b(\mathbf{0})$.

Proof The matrix $A(\chi)$ is invertible for all χ in a neighborhood of $\mathbf{0}$. Consequently, there is a well-defined solution of $A(\chi)x = b(\chi)$ for all χ in this neighborhood and for each such χ we can define the value of $x^\star(\chi)$ to be this solution. That is, for all χ within a neighborhood of $\mathbf{0}$ we have that $A(\chi)x^\star(\chi) = b(\chi)$. (Since $A(\mathbf{0})$ is non-singular, the solution is unique and we have that $x^\star(\mathbf{0}) = x^{\star\star}$.)
That is, $x^\star(\chi) = [A(\chi)]^{-1}b(\chi)$ for all χ in this neighborhood. The inverse $[A(\chi)]^{-1}$ is partially differentiable with respect to χ_j in the neighborhood. Moreover, the partial derivative is continuous. Therefore, $x^\star(\chi)$, being the product of partially differentiable functions with continuous partial derivatives, is also partially differentiable with respect to χ_j in the neighborhood.
Totally differentiating $A(\chi)x^\star(\chi) = b(\chi)$ with respect to χ_j, evaluating at $\chi = \mathbf{0}$, and re-arranging yields (5.16). □

5.6.1.2 Discussion

- If we have already factorized the base-case coefficient matrix $A(\mathbf{0})$ then (5.16) shows that the sensitivity of $x^\star$ with respect to variation in χ_j can be calculated with one additional forwards and backwards substitution using the right-hand side $\frac{\partial b}{\partial \chi_j}(\mathbf{0}) - \frac{\partial A}{\partial \chi_j}(\mathbf{0})x^{\star\star}$.
- Finding the partial derivative of all entries of $x^\star$ with respect to all entries of $\chi \in \mathbb{R}^s$ requires s forwards and backwards substitutions.
- Each forwards and backwards substitution provides a sensitivity $\frac{\partial x^\star}{\partial \chi_j}(\mathbf{0})$.
- Since the base-case solution $x^{\star\star}$ in Theorem 5.7 is equal to $x^\star(\mathbf{0})$, we will from now on abuse notation somewhat and usually write $x^\star$ for the base-case solution and *also* for the function that represents the dependence of the solution on χ.

5.6.1.3 Direct sensitivity analysis

- A single forwards and backwards substitution is required to calculate the sensitivity of $x^\star$ to an entry χ_j of $\chi \in \mathbb{R}^s$.

Example

$$\begin{aligned}\forall\chi \in \mathbb{R}, A(\chi) &= \begin{bmatrix} 1 & 2+\chi \\ 3 & 4 \end{bmatrix}, \\ \forall\chi \in \mathbb{R}, b(\chi) &= \begin{bmatrix} 1 \\ 1+\chi \end{bmatrix}, \\ x^\star(0) &= \begin{bmatrix} -1 \\ 1 \end{bmatrix}.\end{aligned}$$

Example, continued

$$
\begin{aligned}
\frac{\partial A}{\partial \chi}(\chi) &= \frac{\partial A}{\partial \chi}(\mathbf{0}), \\
&= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \\
\frac{\partial b}{\partial \chi}(\chi) &= \frac{\partial b}{\partial \chi}(\mathbf{0}), \\
&= \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \\
\frac{\partial b}{\partial \chi}(\mathbf{0}) - \frac{\partial A}{\partial \chi}(\mathbf{0})x^\star(0) &= \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \\
\frac{\partial x^\star}{\partial \chi}(\mathbf{0}) &= [A(\mathbf{0})]^{-1}\left[\frac{\partial b}{\partial \chi}(\mathbf{0}) - \frac{\partial A}{\partial \chi}(\mathbf{0})x^\star\right], \\
&= \begin{bmatrix} 3 \\ -2 \end{bmatrix}.
\end{aligned}
$$

Circuit case study

- For the ladder circuit in Figure 4.3, which is repeated in Figure 5.5, there is an additional conductance of ΔG_{23} between nodes 2 and 3.

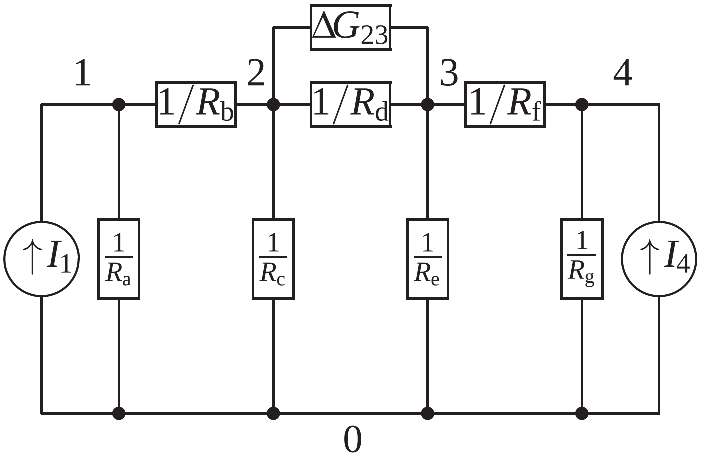


Fig. 5.5. The ladder circuit of Figure 4.3 that has a change in the conductance between nodes $\ell = 2$ and $k = 3$.

Circuit case study, continued

- The sensitivity of the solution of this circuit with respect to $\Delta G_{23} = \chi$, evaluated at $\Delta G_{23} = \chi = 0$, is given by the solution of a circuit with "current injections" (actually having units of voltage) equal to:

$$\frac{\partial b}{\partial \Delta G_{23}}(0) - \frac{\partial A}{\partial \Delta G_{23}}(0)x^\star = \mathbf{0} - \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x^\star,$$

- where:
  - $x^\star$ is the base-case solution,
  - the current injections do not depend on ΔG_{23} so that $\frac{\partial b}{\partial \Delta G_{23}}(0) = \mathbf{0}$, and
  - the dependence of the admittance matrix A on ΔG_{23} was discussed in Section 4.1.3.2.
- The solution of the circuit is a vector of "voltages" (actually having units of voltage divided by admittance) that represent the sensitivities with respect to ΔG_{23}.

5.6.1.4 Adjoint sensitivity

- In this section we will suppose that there is an **objective function** $f : \mathbb{R}^n \to \mathbb{R}$ that provides the value or payoff of the solution x.
- We define $f^\star : \mathbb{R}^s \to \mathbb{R}$ by:

$$\forall \chi \in \mathbb{R}^s, f^\star(\chi) = f(x^\star(\chi)).$$

- We are interested in calculating the partial derivative of $f^\star$, again assuming that we have a base-case solution $x^\star(\mathbf{0})$ corresponding to the parameter value $\chi = \mathbf{0}$ and also assuming that f is differentiable.

$$\begin{aligned}
\frac{\partial f^\star}{\partial \chi_j}(\mathbf{0}) &= \frac{d[f(x^\star(\chi))]}{d\chi_j}(\mathbf{0}), \\
&= \frac{\partial f}{\partial x}(x^\star(\mathbf{0}))\frac{\partial x^\star}{\partial \chi_j}(\mathbf{0}), \text{ by the chain rule,} \\
&= \frac{\partial f}{\partial x}(x^\star(\mathbf{0}))[A(\mathbf{0})]^{-1}\left[\frac{\partial b}{\partial \chi_j}(\mathbf{0}) - \frac{\partial A}{\partial \chi_j}(\mathbf{0})x^\star(\mathbf{0})\right]. \qquad (5.17)
\end{aligned}$$

Adjoint sensitivity, continued

- Let us define $\xi \in \mathbb{R}^n$ to be the solution of:

$$[A(\mathbf{0})]^\dagger \xi = \nabla f(x^\star(\mathbf{0})). \tag{5.18}$$

- Solving for ξ in (5.18) and taking the transpose of the result yields:

$$\begin{aligned} \xi^\dagger &= \frac{\partial f}{\partial x}(x^\star(\mathbf{0}))[A(\mathbf{0})]^{-1}, \\ \frac{\partial f^\star}{\partial \chi_j}(\mathbf{0}) &= \xi^\dagger \left[\frac{\partial b}{\partial \chi_j}(\mathbf{0}) - \frac{\partial A}{\partial \chi_j}(\mathbf{0}) x^\star(\mathbf{0}) \right]. \end{aligned}$$

- Calculation of the vector ξ in (5.18) requires the solution of a linear equation with coefficient matrix $[A(\mathbf{0})]^\dagger$.
- After ξ has been calculated with one forwards and backwards substitution, sensitivities of $f^\star$ with respect to all entries of χ can be evaluated.

Example

$$\begin{aligned}
\forall x \in \mathbb{R}^2, f(x) &= (x_1)^2 + (x_2)^2 + 2x_2 - 3, \\
\forall x \in \mathbb{R}^2, \nabla f(x) &= \begin{bmatrix} 2x_1 \\ 2x_2 + 2 \end{bmatrix}, \\
\nabla f(x^\star) &= \begin{bmatrix} -2 \\ 4 \end{bmatrix}.
\end{aligned}$$

Example, continued

- In this case, (5.18) becomes:

$$\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \xi = \begin{bmatrix} -2 \\ 4 \end{bmatrix},$$

- which has solution $\xi = \begin{bmatrix} 10 \\ -4 \end{bmatrix}$, so that:

$$\begin{aligned} \frac{\partial f^{\star}}{\partial \chi}(\mathbf{0}) &= \xi^{\dagger} \left[\frac{\partial b}{\partial \chi}(\mathbf{0}) - \frac{\partial A}{\partial \chi}(\mathbf{0}) x^{\star}(\mathbf{0}) \right], \\ &= \begin{bmatrix} 10 & -4 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \\ &= -14. \end{aligned}$$

Circuit case study

- The linear equation that is solved for adjoint sensitivity analysis corresponds to a circuit that has entries in its admittance matrix that are the *transpose* of those in the base-case and has entries in its “current vector” that are defined in terms of the sensitivity of the objective function.
- The circuit is called the **adjoint sensitivity circuit**.
- One solution of the adjoint sensitivity circuit suffices for sensitivities of an objective function with respect to all parameters of interest.
- In the case of resistive circuits with current sources, the admittance matrix is symmetric, so that the resistors in the adjoint circuit are the same as those in the base-case circuit.
- For some circuits, however, the admittance matrix is not symmetric and the adjoint circuit has components that are different from those in the base-case circuit.

5.6.2 Large changes

5.6.2.1 Right-hand side

- We can easily accommodate large changes in b, either by:
  - (i) re-solving the equations with the new value of b, or
  - (ii) solving for the change Δx in x to match the change Δb in b.
- In the second case, we assume that we have already have a solution $x^\star$ that satisfies $Ax^\star = b$ and now we want to find Δx that satisfies $A(x^\star + \Delta x) = b + \Delta b$, where Δb is the change in the right-hand side.
- We must solve $A\Delta x = \Delta b$.
- The computational effort using forwards and backwards substitution as described in Section 5.2 is on the order of $(n)^2$.
- We saw a case in Section 5.5.2.2 where the effort is much smaller than $(n)^2$ if Δb has only a few non-zero entries.
- The solution of the system $Ax = b$ is a linear function of the right-hand side vector b.
- If a sensitivity analysis is carried out with respect to parameters that are all entries of b then the sensitivity and large change analysis yield the same result.

5.6.2.2 Coefficient matrix

- Assuming that A has been factorized as $LDL^{\dagger}$, we have:

$$\begin{aligned}
A+\Delta A &= LDL^{\dagger}+\Delta A, \\
&= LDL^{\dagger}+LL^{-1}\Delta A[L^{-1}]^{\dagger}L^{\dagger}, \text{ since } LL^{-1}=\mathbf{I}, \\
&= L(DL^{\dagger}+L^{-1}\Delta A[L^{-1}]^{\dagger}L^{\dagger}), \\
&\qquad\qquad \text{collecting the common factor on the left,} \\
&= L(D+L^{-1}\Delta A[L^{-1}]^{\dagger})L^{\dagger}, \\
&\qquad\qquad \text{collecting the common factor on the right.}
\end{aligned}$$

Coefficient matrix, continued

- Suppose that we can factorize $D+L^{-1}\Delta A[L^{-1}]^{\dagger}$ into $\hat{L}\hat{D}\hat{L}^{\dagger}$ with $\hat{L}$ lower triangular with ones on its diagonal and $\hat{D}$ diagonal.
- Then, we would have:

$$\begin{aligned} A+\Delta A &= L(D+L^{-1}\Delta A[L^{-1}]^{\dagger})L^{\dagger}, \\ &= L\hat{L}\hat{D}\hat{L}^{\dagger}L^{\dagger}, \\ &= \tilde{L}\hat{D}\tilde{L}^{\dagger}. \end{aligned}$$

- The practicality of this approach depends on being able to factorize $D+L^{-1}\Delta A[L^{-1}]^{\dagger}$ using less effort than it takes to factorize $A+\Delta A$.
- This is not true if ΔA is an arbitrary change, but is true for some restricted forms of ΔA that are nevertheless extremely useful in applications.

Coefficient matrix, continued

- For example, suppose that:
  - $\gamma, \delta \in \mathbb{R}$ with γ and δ non-zero, and
  - $w, u \in \mathbb{R}^n$ with w and u linearly independent.
- Then, for the particular forms:

  $\Delta A = \gamma w w^\dagger \in \mathbb{R}^{n\times n}$, which is called a **symmetric rank one update**, and
  $\Delta A = \gamma w w^\dagger + \delta u u^\dagger \in \mathbb{R}^{n\times n}$, which is called a **symmetric rank two update**,
- the computational effort involved is on the order of $(n)^2$, which is considerably less than the effort involved in factorizing $A + \Delta A$ directly.

5.6.3 New variables and equations

- We can also consider augmenting a system of equations by adding a new variable or a new equation.

5.7 Ill-conditioning

5.7.1 Numerical conditioning and condition number

5.7.1.1 Discussion

- In Section 5.3 we showed that if A is non-singular then we can factorize it, while if it is singular then at some stage we will find that there are no non-zero pivots.
- We avoided discussion of the issue of when a coefficient matrix is “nearly” singular in the sense that a small perturbation of the matrix would make is singular.

5.7.1.2 Example

$$A = \begin{bmatrix} 1 & \delta \\ 1 & 0 \end{bmatrix}. \tag{5.19}$$

- If $\delta \neq 0$, then under the assumption of infinite precision arithmetic, we could reliably factorize A and solve $Ax = b$ exactly for the solution $x^\star$.
- However, if δ is small in magnitude, then A is “nearly” singular in that perturbing δ to make it equal to zero would make A singular.
- Small relative errors in the specification of A or b (or in the calculations to factorize A or to perform forwards or backwards substitution) lead to large relative errors in the value of the solution $x^\star$.
- That is, the problem of solving $Ax = b$ given the A defined in (5.19) is ill-conditioned according to Definition 2.21.

Example, continued

$$A^{-1} = \begin{bmatrix} 0 & 1 \\ 1/\delta & -1/\delta \end{bmatrix}.$$

$$x^\star = A^{-1}b = \begin{bmatrix} b_2 \\ (b_1/\delta) - (b_2/\delta) \end{bmatrix}. \tag{5.20}$$

- Let $b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ so that $\|b\|_2 = \sqrt{2}$.
- We have that $x^\star = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $\|x^\star\|_2 = 1$.
- We consider, in turn, changes to b and to A assuming infinite precision calculations.

Right-hand side

- $Ax = b + \Delta b$, with $\Delta b = \begin{bmatrix} \chi \\ 0 \end{bmatrix}$, so that $\|\Delta b\|_2 = |\chi|$.
- $\Delta x^\star = \begin{bmatrix} 0 \\ \chi/\delta \end{bmatrix}$ and $\|\Delta x^\star\|_2 = |\chi/\delta|$.

$$\frac{\|\Delta x^\star\|_2}{\|x^\star\|_2} = |\chi/\delta|\,,$$

$$\frac{\|\Delta b\|_2}{\|b\|_2} = \frac{|\chi|}{\sqrt{2}}.$$

- The relative change in the norm of the solution is on the order of $|1/\delta|$ times the relative change in the norm of the right-hand side.

Coefficient matrix

- $(A+\Delta A)x = b$, with $\Delta A = \begin{bmatrix} \chi & 0 \\ 0 & 0 \end{bmatrix}$, so that $\|\Delta A\|_2 = |\chi|$
- $x^\star + \Delta x^\star = \begin{bmatrix} 1 \\ \chi/\delta \end{bmatrix}$, $\Delta x^\star = \begin{bmatrix} 0 \\ \chi/\delta \end{bmatrix}$ and $\|\Delta x^\star\|_2 = |\chi/\delta|$.

$$\frac{\|\Delta x^\star\|_2}{\|x^\star + \Delta x^\star\|_2} \approx |\chi/\delta|\,,$$

$$\frac{\|\Delta A\|_2}{\|A\|_2} \approx \frac{|\chi|}{\sqrt{2}}.$$

- Again, the relative change in the solution is on the order of $|1/\delta|$ times the relative change in the coefficient matrix.

5.7.1.3 Analysis

- The degree of ill-conditioning is characterized by a measure known as the **condition number** of the matrix.

Definition 5.1 Let $\|\bullet\|$ stand for vector and matrix norms on $\mathbb{R}^n$ and $\mathbb{R}^{n\times n}$ that are compatible. For example, the matrix norm $\|\bullet\|$ could be the matrix norm induced by the vector norm. Suppose that $A \in \mathbb{R}^{n\times n}$ is non-singular. Then the **condition number** of A is defined by $\|A\| \, \|A^{-1}\|$. If $A \in \mathbb{R}^{n\times n}$ is singular then the condition number is defined to be ∞. $\square$

Theorem 5.8 *Let $\|\bullet\|$ stand for vector and matrix norms on $\mathbb{R}^n$ and $\mathbb{R}^{n\times n}$ that are compatible. Suppose that $A \in \mathbb{R}^{n\times n}$ is non-singular and $b \in \mathbb{R}^n$. We consider the relation between solutions of the system $Ax = b$ and solutions of the perturbed systems $Ax = b + \Delta b$ and $(A + \Delta A)x = b$. We have the following bounds:*

(i) *Consider the perturbed system $Ax = b + \Delta b$. The solution $x^\star + \Delta x^\star$ to this perturbed system satisfies:*

$$\frac{\|\Delta x^\star\|}{\|x^\star\|} \leq \|A\| \, \|A^{-1}\| \frac{\|\Delta b\|}{\|b\|},$$

where $x^\star$ is the solution to $Ax = b$. That is, the relative change in the solution is bounded by the product of the condition number and the relative change in the right-hand side.

(ii) *Consider the perturbed system $(A + \Delta A)x = b$. The solution $x^\star + \Delta x^\star$ to this system satisfies:*

$$\frac{\|\Delta x^\star\|}{\|x^\star + \Delta x^\star\|} \leq \|A\| \, \|A^{-1}\| \, \frac{\|\Delta A\|}{\|A\|},$$

where $x^\star$ is the solution to $Ax = b$. That is, the relative change in the solution is bounded by the product of the condition number and the relative change in the coefficient matrix.

□

Example

- Consider the matrix defined in (5.19) and suppose that δ is small.
- If the induced matrix norm $\|\bullet\|_2$ is chosen, then:

  $\|A\|_2 \approx \sqrt{2}$, and
  $\|A^{-1}\|_2 \approx |1/\delta|$,
- so that the condition number is proportional to $|1/\delta|$.
- According to Theorem 5.8, relatively small changes in either the right-hand side b or the coefficient matrix A of the system $Ax = b$ can potentially produce large relative changes in the solution with the amplification proportional to $|1/\delta|$.
- By Theorem 5.7, we obtain that the norm of the sensitivity to χ is $|1/\delta|$.
- These observations are consistent with the above calculations for the matrix defined in (5.19) since the changes in A and b were indeed amplified by $|1/\delta|$ in the solution.

5.7.2 Scaling and pre-conditioning

- Scaling can sometimes be used effectively to reduce the condition number of a matrix.
- For example, consider the matrix:

$$A = \begin{bmatrix} \delta & 0 \\ 1 & 1 \end{bmatrix}, \tag{5.21}$$

$$A^{-1} = \begin{bmatrix} 1/\delta & 0 \\ -1/\delta & 1 \end{bmatrix},$$

$$x^\star = A^{-1}b = \begin{bmatrix} b_1/\delta \\ -(b_1/\delta) + b_2 \end{bmatrix}. \tag{5.22}$$

- If the $\|\bullet\|_2$ induced matrix norm is again used, then for small δ we have that:
  - $\|A\|_2 \approx 1$,
  - $\|A^{-1}\|_2 \approx \left|\sqrt{2}/\delta\right|$,
- so that the condition number is again proportional to $|1/\delta|$.

Scaling and pre-conditioning, continued

- By scaling the first equation of $Ax = b$ by multiplying it by $1/\delta$ we obtain the new system:

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} x = \begin{bmatrix} b_1/\delta \\ b_2 \end{bmatrix}, \tag{5.23}$$

- and the coefficient matrix now has a condition number that is a small constant that is independent of δ.
- We still face the issue that the solution (5.22) is very dependent on the value of δ; however, the condition number of the coefficient matrix has improved.

- It is important to realize that pre-conditioning will not remove the sensitivity of the solution to changes in the originally specified coefficient matrix A and vector b.

5.7.3 Matrix factorization

5.7.3.1 LU factorizing ill-conditioned systems

- The condition number of the lower triangular matrix:

$$L = \left[M^{(n-1)} \ldots M^{(1)}\right]^{-1},$$

- is:

$$\|L\| \left\|L^{-1}\right\| = \left\|\left[M^{(n-1)} \ldots M^{(1)}\right]^{-1}\right\| \left\|M^{(n-1)} \ldots M^{(1)}\right\|,$$

- which is bounded by:

$$\left\|\left[M^{(n-1)}\right]^{-1}\right\| \ldots \left\|\left[M^{(1)}\right]^{-1}\right\| \left\|M^{(n-1)}\right\| \ldots \left\|M^{(1)}\right\| =$$
$$\left\|M^{(n-1)}\right\| \left\|\left[M^{(n-1)}\right]^{-1}\right\| \ldots \left\|M^{(1)}\right\| \left\|\left[M^{(1)}\right]^{-1}\right\|.$$

LU factorizing ill-conditioned systems, continued

- Similarly, the condition number of the upper triangular matrix:

$$U = M^{(1)} \dots M^{(n-1)} A,$$

- is bounded by:

$$\left\|M^{(n-1)}\right\| \left\|\left[M^{(n-1)}\right]^{-1}\right\| \dots \left\|M^{(1)}\right\| \left\|\left[M^{(1)}\right]^{-1}\right\| \|A\| \left\|A^{-1}\right\|.$$

- Both $M^{(j)}$ and $\left[M^{(j)}\right]^{-1}$ have entries that are proportional to the inverse of the pivot used at the j-th stage and their norms will both be correspondingly large.
- The effect of pre-conditioning by $M^{(1)}, \dots, M^{(n-1)}$ is to increase the condition number of the resulting system, which exacerbates the ill-conditioning of A.
- If A is ill-conditioned then the resulting systems $Ly = b$ and $Ux = y$ can be extremely ill-conditioned.

5.7.3.2 *$LDL^{\dagger}$ for positive definite A*

- In the case of a strictly diagonally dominant matrix such as in our circuit case study of Section 4.1, the largest pivots are on the diagonal and diagonal pivoting will keep the condition number of the system relatively low.
- This favorable circumstance also occurs for $LDL^{\dagger}$ factorization of any symmetric positive definite matrix.

5.7.3.3 QR

- An alternative factorization involves multiplying by a sequence of matrices $M^{(j)}$ for which $\left\| M^{(j)} \right\|_2 = \left\| [M^{(j)}]^{-1} \right\|_2 = 1$ so that the condition number of L is one and the condition number of U is the same as the condition number of A.
- The resulting factorization is called the QR factorization and can be applied to an $m \times n$ matrix A with $m \geq n$ and having linearly independent columns to produce a factorization $A = QR$ where $Q \in \mathbb{R}^{m \times m}$ is unitary and $R \in \mathbb{R}^{m \times n}$ is upper triangular.
- That is, $R_{\ell k} = 0$ for $\ell > k$.
- The main drawbacks of QR factorization are that:
  - it takes more computational effort than LU factorization, and
  - the matrix Q will not usually be sparse even if A is sparse.

5.8 Non-square systems

5.8.1 More variables than equations

- Consider the system $Ax = b$ where $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$, and $m < n$.

5.8.1.1 Inconsistent equations

- A system of equations is called **inconsistent** if there is no solution.
- This problem will turn out to be an optimization problem and we will treat it in Section 11.1.

5.8.1.2 Consistent equations and the null space

- If the m rows of A are **linearly independent** then there is an $m \times m$ sub-matrix of A with linearly independent columns.

First m columns linearly independent

- Let $n' = n - m$ and partition A into $\begin{bmatrix} A^{\|} & A^{\perp} \end{bmatrix}$ where $A^{\|} \in \mathbb{R}^{m \times m}$ and $A^{\perp} \in \mathbb{R}^{m \times n'}$.
- Similarly, partition x into $\begin{bmatrix} \omega \\ \xi \end{bmatrix}$ where $\omega \in \mathbb{R}^m$ and $\xi \in \mathbb{R}^{n'}$.
- Suppose that $A^{\|}$ has linearly independent columns, so that $A^{\|}$ is non-singular.

$$\begin{aligned} Ax = b \quad &\Leftrightarrow \quad \begin{bmatrix} A^{\|} & A^{\perp} \end{bmatrix} \begin{bmatrix} \omega \\ \xi \end{bmatrix} = b, \text{ by definition of } \begin{bmatrix} A^{\|} & A^{\perp} \end{bmatrix} \text{ and } \begin{bmatrix} \omega \\ \xi \end{bmatrix}, \\ &\Leftrightarrow \quad A^{\|}\omega + A^{\perp}\xi = b, \\ &\Leftrightarrow \quad A^{\|}\omega = b - A^{\perp}\xi, \\ &\Leftrightarrow \quad \omega = [A^{\|}]^{-1}(b - A^{\perp}\xi), \end{aligned}$$

- Let $\hat{\xi} = \mathbf{0}$ and $\hat{\omega} = [A^{\|}]^{-1} b$.

First m columns linearly independent, continued

- Then define:

$$\begin{aligned}\hat{x} &= \begin{bmatrix}\hat{\omega}\\ \hat{\xi}\end{bmatrix},\\ &= \begin{bmatrix}[A^{\|}]^{-1}b\\ \mathbf{0}\end{bmatrix}.\end{aligned}$$

- The vector $\hat{x}$ is one **particular solution** to $Ax = b$.
- The set of all solutions to $Ax = b$ is given by $\{\hat{x} + \Delta x \in \mathbb{R}^n | A\Delta x = \mathbf{0}\}$.
- The set:

$$\mathcal{N}(A) = \{\Delta x \in \mathbb{R}^n | A\Delta x = \mathbf{0}\},$$

- is called the **null space** of A.

First m columns linearly independent, continued

- Partition Δx into $\begin{bmatrix} \Delta\omega \\ \Delta\xi \end{bmatrix}$, where $\Delta\omega \in \mathbb{R}^m$ and $\Delta\xi \in \mathbb{R}^{n'}$.

$$\begin{aligned} A\Delta x = \mathbf{0} \;&\Leftrightarrow\; \begin{bmatrix} A^{\|} & A^{\perp} \end{bmatrix} \begin{bmatrix} \Delta\omega \\ \Delta\xi \end{bmatrix} = \mathbf{0}, \\ &\Leftrightarrow\; A^{\|}\Delta\omega + A^{\perp}\Delta\xi = \mathbf{0}, \\ &\Leftrightarrow\; A^{\|}\Delta\omega = -A^{\perp}\Delta\xi, \\ &\Leftrightarrow\; \Delta\omega = -[A^{\|}]^{-1}A^{\perp}\Delta\xi. \end{aligned}$$

First m columns linearly independent, continued

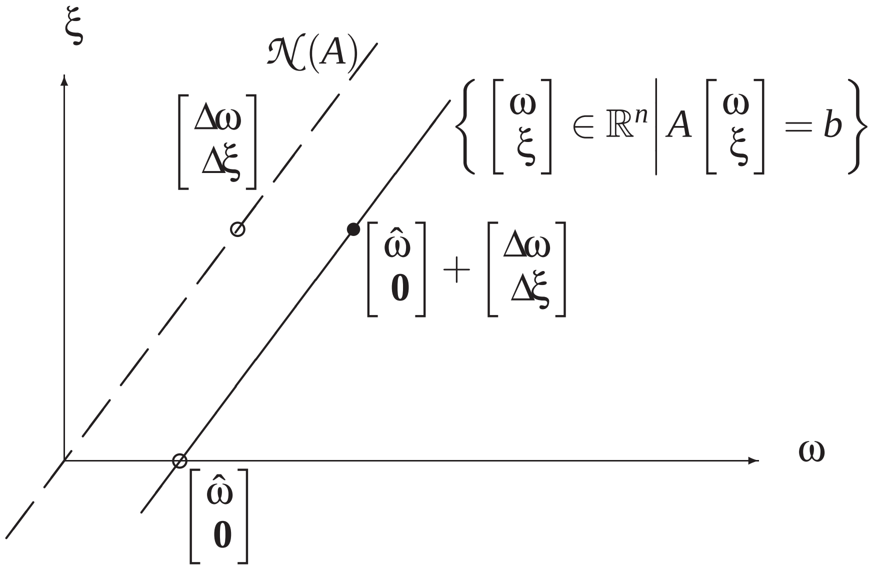
$$\begin{aligned}
\mathcal{N}(A) &= \{\Delta x \in \mathbb{R}^n | A\Delta x = \mathbf{0}\}, \\
&= \left\{ \begin{bmatrix} -[A^{\|}]^{-1} A^{\perp}\Delta\xi \\ \Delta\xi \end{bmatrix} \middle| \Delta\xi \in \mathbb{R}^{n'} \right\}, \\
&= \{Z\Delta\xi | \Delta\xi \in \mathbb{R}^{n'}\}, \\
\text{where: } Z &= \begin{bmatrix} -[A^{\|}]^{-1} A^{\perp} \\ \mathbf{I} \end{bmatrix}.
\end{aligned}$$

- The columns of Z form a **basis** for the null space of A.
- Every solution of $Ax = b$ is of the form $x = \begin{bmatrix} \hat{\omega} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \Delta\omega \\ \Delta\xi \end{bmatrix}$,
- where:

  $\hat{x} = \begin{bmatrix} \hat{\omega} \\ \mathbf{0} \end{bmatrix}$ is a particular solution of $Ax = b$, and

  $\begin{bmatrix} \Delta\omega \\ \Delta\xi \end{bmatrix} \in \mathcal{N}(A)$.

First m columns linearly independent, continued



ξ
$\mathcal{N}(A)$
$\begin{bmatrix} \Delta\omega \\ \Delta\xi \end{bmatrix}$
$\left\{ \begin{bmatrix} \omega \\ \xi \end{bmatrix} \in \mathbb{R}^n \middle| A \begin{bmatrix} \omega \\ \xi \end{bmatrix} = b \right\}$
$\begin{bmatrix} \hat{\omega} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \Delta\omega \\ \Delta\xi \end{bmatrix}$
ω
$\begin{bmatrix} \hat{\omega} \\ \mathbf{0} \end{bmatrix}$

Fig. 5.6. Solution of linear equations. The solid line represents the set of points satisfying the linear equations. The null space of the coefficient matrix is shown as the dashed line.

Linearly independent columns unknown

- An analogous factorization to the QR factorization can be used to write $PA = LQ$, where now:

  $P \in \mathbb{R}^{m \times m}$ is a permutation matrix,
  $L \in \mathbb{R}^{m \times n}$ is lower triangular, with its first m' columns linearly independent and its last $n' = n - m'$ columns zero, and
  $Q \in \mathbb{R}^{n \times n}$ is unitary.
- Partition L into $\begin{bmatrix} L^{\|} & \mathbf{0} \end{bmatrix}$ where $L^{\|} \in \mathbb{R}^{m \times m'}$ is lower triangular with its m' columns linearly independent.
- If A has m linearly independent columns then $m' = m$.

Linearly independent columns unknown, continued

- Let $n' = n - m'$ and $y = Qx$ and partition $y \in \mathbb{R}^n$ into $y = \begin{bmatrix} \omega \\ \xi \end{bmatrix}$ where $\omega \in \mathbb{R}^{m'}$ and $\xi \in \mathbb{R}^{n'}$.

$$\begin{aligned}
Ax = b \quad &\Leftrightarrow \quad PAx = Pb, \text{ since } P \text{ is non-singular,} \\
&\Leftrightarrow \quad LQx = Pb, \text{ by definition of } LQ, \\
&\Leftrightarrow \quad Ly = Pb \text{ and } y = Qx, \\
&\Leftrightarrow \quad \begin{bmatrix} L^{\|} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \omega \\ \xi \end{bmatrix} = Pb \text{ and } y = Qx, \\
&\Leftarrow \quad L^{\|}\omega = Pb \text{ and } y = \begin{bmatrix} \omega \\ \mathbf{0} \end{bmatrix} = Qx.
\end{aligned}$$

- If $m' = m$ then similar arguments to before show that $\hat{y} = \begin{bmatrix} [L^{\|}]^{-1} Pb \\ \mathbf{0} \end{bmatrix}$ satisfies $Ly = Pb$ and that $\hat{x} = Q^{-1}\hat{y} = Q^{\dagger}\hat{y}$ satisfies $Ax = b$.

Linearly independent columns unknown, continued

$$\begin{aligned}\mathcal{N}(A) &= \{\Delta x \in \mathbb{R}^n | A\Delta x = \mathbf{0}\}, \\ &= \left\{ Q^\dagger \begin{bmatrix} \mathbf{0} \\ \Delta\xi \end{bmatrix} \middle| \Delta\xi \in \mathbb{R}^{n'} \right\}, \\ &= \{Z\Delta\xi | \Delta\xi \in \mathbb{R}^{n'}\},\end{aligned}$$

- where Z is the last n' columns of $Q^\dagger$.

5.8.2 More equations than variables

- Consider the system $Ax = b$ where $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$, and $m > n$.

5.8.2.1 Inconsistent equations

- **Inconsistent equations** typically occur if A is non-square with more equations than variables.
- We will investigate this type of problem in Section 11.1.

5.8.2.2 Consistent equations

- For $b \in \mathcal{R}(A)$, the system $Ax = b$ will have one or more solutions, even if there are more equations than variables.
- For such b, we say that the equations are **consistent**.
- There are redundant equations.

5.8.3 The pseudo-inverse

- The preceding discussion can be unified by defining the notion of the **pseudo-inverse**, which is defined to be the (unique) matrix $A^+ \in \mathbb{R}^{n \times m}$ such that the vector $x = A^+ b$ is the vector having the minimum value of norm $\|x\|_2$ over all vectors that minimize $\|Ax - b\|_2$.

5.9 Iterative methods

- Very large, but sparse, systems can be solved effectively by factorization.
- However, if the coefficient matrix is extremely large or is dense, then the factorization approaches become too time consuming.
- An alternative approach involves an **iterative algorithm** where a sequence $\{x^{(\nu)}\}_{\nu=0}^{\infty}$ of approximations to the solution of $Ax = b$ are calculated.

5.10 Summary

- In this chapter we have described LU factorization (and its variants) and forwards and backward substitution as an efficient approach to solving systems of linear equations, paying particular attention to symmetric systems.
- We considered the selection of pivots and discussed the solution of perturbed systems, sparse methods, and the issue of ill-conditioning.
- We briefly discussed the solution of non-square systems and iterative techniques.
- In later chapters we will need to solve large linear systems repeatedly so that the algorithms developed in this chapter will be incorporated into all subsequent algorithms.