Teo Bucci

Appunti di
Matematica Numerica

fubini ⊗ tonelli
edizioni e convoluzioni

Appunti di Matematica Numerica

Il codice sorgente LaTeX è disponibile su
`https://github.com/fubinitonelli/matematica-numerica`

Documento creato il 1 marzo 2024
Revisione `e955ae3a3965160ff8b0fbdc8f44224495d0b6cc`

Developed by Teo Bucci - `teo@fubinitonelli.it`
Powered by Fubini ⊗ Tonelli

Per segnalare eventuali errori o suggerimenti potete contattare l'autore.

Prefazione

Questo testo è una rielaborazione di appunti presi durante un corso di Matematica Numerica. Si rivolge a tutti quegli studenti che si stanno affacciando per la prima volta ai metodi numerici e vogliono avere delle basi solide e dei concetti chiari, col fine di affrontare al meglio le applicazioni future.

Essendo una raccolta di appunti, non si pone come un sostituto di libri di testo maggiori e più approfonditi, né delle lezioni frontali con il docente, il quale rimane una risorsa fondamentale per scavare a fondo della materia. Questo libro si pone invece come un supporto ulteriore che affianca gli strumenti sopracitati e accompagna lo studente nei primi passi all'interno della Matematica Numerica.

Ringraziamo l'autore del libro, new entry nel nostro team, per aver realizzato un lavoro completo e approfondito, e per la fiducia che ci ha accordato nella pianificazione della sua revisione e pubblicazione. Essenziali sono stati anche i contributi di Bruno Guindani, nella forma di revisione contenutistica e stilistica; di Aron Wussler, che si è occupato della revisione grafica; Fabrizio Bernardi per un'altra revisione contenutistica più approfondita; e Gabriele Gabrielli per la copertina dell'opera.

Siamo orgogliosi che il progetto e l'idea del libro "Appunti di Probabilità", da noi realizzato ormai quattro anni fa, possa continuare in altre materie, a beneficio di tanti nuovi studenti, e che possa espandere la nostra piccola comunità di nerd di LaTeX.

L'associazione Fubini-Tonelli

Indice

Capitolo 1

Introduzione all'analisi numerica

In questo testo si affronteranno i concetti fondamentali dell'analisi numerica. Proponiamo questo capitolo introduttivo in modo da dare qualche motivazione sull'utilità della materia. In generale, quando studiamo un problema fisico, dopo aver determinato una *formula* della soluzione è anche fondamentale saper calcolare numericamente, o approssimare, questa soluzione per la specifica istanza del problema in questione. Nei casi reali, molto spesso la soluzione esatta ci è sconosciuta e non possiamo ottenerla in forma chiusa: il compito dell'analisi numerica sarà allora quello di trovare la migliore approssimazione possibile. Un altro caso di interesse è la *simulazione* di un certo fenomeno, per sapere come una struttura reagisce ai carichi, o se un vaso sanguigno è a rischio rottura per la sua particolare conformazione. In tutti questi casi abbiamo una procedura sequenziale composta da diverse fasi:

1. Un **problema fisico (PF)**: è il punto di partenza, in cui analizziamo una data situazione nel mondo reale.
2. Un **problema matematico (PM)**: scegliamo un modello matematico che rappresenti il (PF), traducendo le informazioni su di esso in formule ed eventualmente operando le dovute ipotesi semplificative. Per esempio, se si è in una situazione in cui l'attrito ha valori trascurabili e non è quindi determinante per il problema in questione, esso può essere ignorato in modo da semplificare le formule.
3. Un **problema numerico (PN)**: significa compiere un'ulteriore traduzione, dalle equazioni del (PM) a una serie di algoritmi e schemi numerici che permettano a un computer di risolvere il (PM). Ricordiamo infatti che un computer è uno strumento "stupido": esegue tutte e sole le istruzioni che

gli vengono fornite, molto velocemente, ma non è in grado di comprendere il significato di concetti come "integrale" o "serie infinita". Un computer è in grado di eseguire solo le operazioni di base, e deve necessariamente approssimare i numeri secondo il suo metodo di rappresentazione, rendendo quindi impossibile utilizzare risultati numerici completamente esatti. Per esempio, è possibile che eseguendo la divisione $2/2$ e visualizzando il risultato si trovi 1.000000002. Questi errori sono qualcosa di cui bisogna tenere conto nel progettare un algoritmo.

Non scontato è chiedersi perché *complicarsi* la vita così. La ragione è che spesso quando si studiano situazioni reali e si vuole un livello di precisione molto alto, si presentano dati e sistemi di n equazioni dove n è un numero che può valere anche 10000 o 100000. Procedere a mano non è quindi fattibile.

Inoltre, prendendo per esempio lo studio della resistenza delle pareti di un vaso sanguigno o del cuore, la modellistica numerica fornisce un'accuratezza *ineguagliabile* da qualunque test clinico, senza essere invasiva per il paziente.

Procediamo quindi ad analizzare un primo esempio, per illustrare meglio queste affermazioni.

1.1 Problema esempio

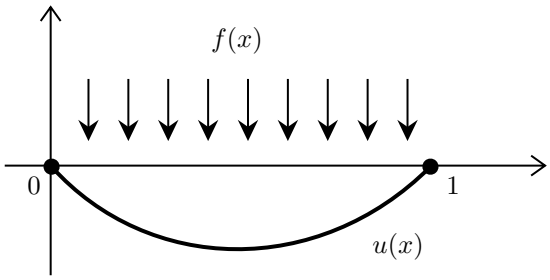
Vogliamo determinare la configurazione di equilibrio di un filo elastico fissato agli estremi e soggetto ad una generica forza verticale di intensità variabile. Questo è il nostro (PF).

1.1.1 Formulazione forte

Descriveremo ora una prima formulazione del problema. In particolare, fissato un asse cartesiano orizzontale avente $x \in [0,1]$ come coordinata, si indichi con $f(x)$ l'intensità di tale forza per unità di massa nel punto x. Detta $u(x)$ la funzione che descrive lo spostamento verticale del filo in x rispetto alla posizione di riposo $u = 0$, dobbiamo trovare $u(x)$ tale che:

$$\begin{cases} -u''(x) = f(x) & \\ u(0) = 0 & x \in (0,1). \\ u(1) = 0 & \end{cases} \tag{PM}$$

Questo è il nostro (PM). Non entriamo nel dettaglio delle ragioni della scelta di questo modello per questa situazione fisica, in quanto non sono oggetto di questo corso.



Per ogni $f \in C^0([0,1])$ esiste un'unica soluzione $u \in C^2([0,1])$ del problema (PM) che ammette la seguente rappresentazione:

$$u(x) = x\int_0^1 (1-s)f(s)ds - \int_0^x (x-s)f(s)\mathrm{d}s. \tag{1.1}$$

Dimostrazione.

Separiamo le variabili e integriamo

$$-\frac{du'(x)}{\mathrm{d}x} = f(x) \quad \Rightarrow \quad -du'(x) = f(x)\mathrm{d}x \quad \Rightarrow \quad -\int_0^x du'(s) = \int_0^x f(s)\mathrm{d}s.$$

Il risultato dell'integrazione è

$$u'(x) - u'(0) = -\int_0^x f(s)\mathrm{d}s \quad \Rightarrow \quad u'(x) = -\int_0^x f(s)\mathrm{d}s + c_1,$$

integrando quindi ulteriormente con la stessa procedura

$$u(x) = -\int_0^x F(s)\mathrm{d}s + c_1 x + c_2, \quad \text{indicando } F(s) = \int_0^s f(t)\mathrm{d}t.$$

Integrando per parti il termine $\int_0^x F(s)\mathrm{d}s$:

$$\begin{aligned}\int_0^x F(s)\mathrm{d}s &\overset{\text{ipp}}{=} [sF(s)]_0^x - \int_0^x sF'(s)\mathrm{d}s \\ &= [xF(x) - \cancel{0F(0)}] - \int_0^x sf(s)\mathrm{d}s \\ &= x\int_0^x f(s)\mathrm{d}s - \int_0^x sf(s)\mathrm{d}s \\ &= \int_0^x f(s)(x-s)\mathrm{d}s,\end{aligned}$$

da cui

$$u(x) = -\int_0^x f(s)(x-s)\mathrm{d}s + c_1 x + c_2.$$

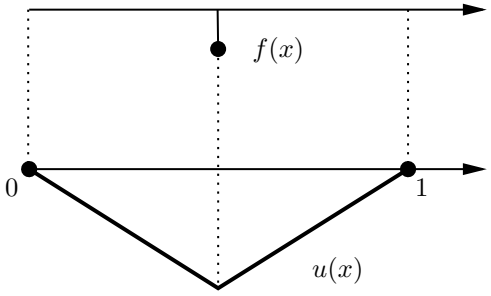
Le costanti c_1, c_2 le otteniamo imponendo le condizioni al bordo $u(0) = u(1) = 0$:

$$\begin{aligned} 0 = u(0) &= -\cancel{\int_0^0 f(s)(0-s)\mathrm{d}s} + \cancel{c_1 \cdot 0} + c_2 = c_2 \\ &\Rightarrow \quad c_2 = 0 \\ 0 = u(1) &= -\int_0^1 f(s)(1-s)\mathrm{d}s + c_1 \cdot 1 + \cancel{c_2} \\ &\Rightarrow \quad c_1 = \int_0^1 f(s)(1-s)\mathrm{d}s. \end{aligned}$$

Sostituendo, otteniamo la (1.1). ■

Si può dimostrare inoltre che se $f \in C^m([0,1])$, $m \geq 0$, allora $u \in C^{m+2}([0,1])$.

La $f(x)$ è detta *forzante* e rappresenta l'azione esogena, proveniente dall'esterno del sistema di cui studiamo l'evoluzione. Cosa succede se $f \notin C^0([0,1])$, cioè se vogliamo studiare stimoli discontinui a tratti? È chiaro che se alla corda agganciamo una massa nel suo punto medio allora assumerà una configurazione a triangolo:



Anche considerare forzanti continue a tratti è di grande interesse fisico. Tuttavia, queste non possono essere trattate con la formulazione (PM), che è detta **formulazione forte**, in quanto essa è troppo limitata a scelte di f regolare. Ci proponiamo quindi di espandere il concetto di soluzione con la seguente nuova formulazione.

1.1.2 Formulazione debole

Vogliamo passare da un problema differenziale del secondo ordine ad uno in forma integrale del primo ordine (**formulazione debole**)[1].

[1] I seguenti passaggi potrebbero risultare un po' nebulosi. Ciò non deve scoraggiare perché è solo per dare qualche motivazione ai vari argomenti che si affronteranno nel corso. Inoltre, gran parte di questi contenuti più teorici (funzioni test, spazi L^p, spazi a dimensione infinita) verranno approfonditi in corsi successivi.

Formalmente moltiplichiamo l'equazione (PM) per una funzione $v \in V$, detta *funzione test*, ed integriamo tra 0 e 1:

$$-u''(x) = f(x) \quad \Rightarrow \quad \int_0^1 -u''(x)v(x)\mathrm{d}x = \int_0^1 f(x)v(x)\mathrm{d}x.$$

Per il momento supponiamo che le operazioni siano tutte lecite; ci occuperemo dei dettagli formali, come la definizione dello spazio V, in un secondo momento. Integriamo il primo integrale per parti, con lo scopo di far scomparire la derivata seconda di u:

$$\int_0^1 u'(x)v'(x)\mathrm{d}x - [u'(x)v(x)]_0^1 = \int_0^1 f(x)v(x)\mathrm{d}x.$$

Si può dimostrare[2] che poiché la soluzione è fissa agli estremi $u(0) = u(1) = 0$, allora la scelta ottimale delle funzioni test in V è tale per cui esse siano nulle agli estremi dell'intervallo, cioè:

$$v \in V \quad \Rightarrow \quad v(0) = v(1) = 0,$$

$$\int_0^1 u'(x)v'(x)\mathrm{d}x - \cancel{u'(1)v(1)} + \cancel{u'(0)v(0)} = \int_0^1 f(x)v(x)\mathrm{d}x.$$

Definiamo lo spazio

$$L^2(0,1) \coloneqq \left\{ v : (0,1) \to \mathbb{R} \text{ tali che } \left(\int_0^1 |v|^2 \mathrm{d}x \right)^{1/2} < +\infty \right\},$$

e definiamo di conseguenza lo spazio V come

$$V = \left\{ v : (0,1) \to \mathbb{R} \mid v \in L^2(0,1),\ v' \in L^2(0,1),\ v(0) = v(1) = 0 \right\}.$$

Arriviamo così alla formulazione debole del problema. Data $f \in L^2(0,1)$ trovare $u \in V$ tale che:

$$\int_0^1 u'(x)v'(x)\mathrm{d}x = \int_0^1 f(x)v(x)\mathrm{d}x, \quad \forall v(x) \in V. \tag{PM'}$$

La formulazione debole è quindi piu generale di quella forte, dal momento che non richiede che f sia derivabile, ma formula la soluzione sotto forma di integrale. Di conseguenza la richiesta su f sarà di integrabilità: lo spazio L^2 è infatti lo spazio di funzioni il cui quadrato è integrabile[3].

Osservazioni.

- In (PM') compare solo la derivata prima di u.
- Lo spazio V è più grande dello spazio $C^2([0,1])$.

[2] ma non è argomento di questo testo.

[3] Tra tutti gli spazi L^p c'è una preferenza a scegliere L^2 dato che è l'unico ad essere uno spazio di Hilbert, e questo porta con sé una serie di interessanti e utili proprietà, non oggetto di questo corso.

1.1.3 Problema numerico

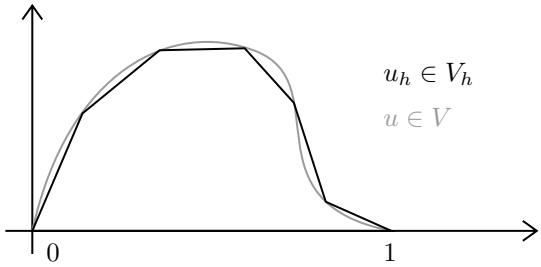
Il prossimo obiettivo è costruire una successione approssimante u_h della soluzione originale u di (PM), dove u_h è soluzione di un problema *simile* a (PM), ma che sia finito-dimensionale, al contrario del problema "originale" che è infinito-dimensionale. Parliamo di dimensione nel senso usato nell'algebra lineare: ogni elemento di uno spazio può essere generato da un'opportuna combinazione lineare di elementi della base, la cui cardinalità è la dimensione dello spazio. Bisogna ora iniziare a pensare agli *elementi* della base non più come vettori come nell'algebra lineare classica, ma come vere e proprie funzioni. Per generare un elemento (una funzione) dello spazio finito dimensionale, serve un numero finito di funzioni; mentre per generare un elemento dello spazio infinito-dimensionale è necessaria una base composta da infiniti elementi.

In altre parole, definiamo uno spazio $V_h \subset V$ tale che $\dim(V_h) = N_h < +\infty$, e chiamiamo $u_h \in V_h$ la soluzione del (PM) *ristretto* allo spazio finito-dimensionale V_h[4]. Data $f \in L^2(0,1)$ trovare $u_h \in V_h$ tale che:

$$\int_0^1 u_h'(x)v_h'(x)\mathrm{d}x = \int_0^1 f(x)v_h(x)\mathrm{d}x, \quad \forall v_h \in V_h. \tag{PN}$$

Osservazioni.

- In base a come costruiamo lo spazio discreto $V_h \subset V$, e quindi l'approsimazione u_h di u, possiamo ottenere metodi numerici diversi.
- Nel **metodo degli elementi finiti** (FEM, Finite Elements Method), l'approssimazione $u_h \in V_h$ di $u \in V$ è costruita come un polinomio (spesso lineare) a tratti continuo e che si annulla agli estremi.



Dobbiamo saper stimare l'errore della soluzione approssimata u_h rispetto a quella reale u, senza conoscere quest'ultima. Dobbiamo poi trovare un modo per controllare questo errore, cioè renderlo arbitrariamente piccolo.

I passi per costruire u_h sono:

[4] Si sottolinea che la finita-dimensionalità dello spazio è essenziale per poter dare in pasto al calcolatore i nostri problemi. Non è scontato che l'approssimazione discreta *converga* in qualche senso a quella originale, tuttavia vi sono risultati teorici (che si vedranno in corsi successivi) che garantiscono tutto ciò che facciamo in questo corso sia lecito.

1. Costruiamo lo spazio V_h. Ricordiamo che $\dim(V_h) = N_h < +\infty$, e che qualunque spazio di dimensione N_h può essere generato da N_h vettori indipendenti. Per indicarlo scriviamo:

$$V_h = \text{span}\{\varphi_j(x), j = 1, \dots, N_h\}.$$

2. Scriviamo u_h come combinazione lineare delle funzioni di base $\varphi_j(x)$ secondo i coefficienti $u_j, j = 1, \dots, N_h$, ossia

$$u_h(x) = \sum_{j=1}^{N_h} u_j \varphi_j(x), \quad u_1, u_2, \dots, u_{N_h} \in \mathbb{R}.$$

3. Utilizzando questa espansione riscriviamo (PM). Trovare $u_1, u_2, \dots, u_{N_h} \in \mathbb{R}$ tali che:

$$\int_0^1 \sum_{j=1}^{N_h} u_j \varphi_j'(x) v_h'(x) \mathrm{d}x = \int_0^1 f(x) v_h(x) \mathrm{d}x, \quad \forall v_h(x) \in V_h.$$

4. Osserviamo che la condizione $\forall v_h(x) \in V_h$ è equivalente a $\forall \varphi_i(x), i = 1, \dots, N_h$. Se l'identità vale per le funzioni di base φ_i allora vale anche per le funzioni v_h dato che si devono poter scrivere come combinazione delle funzioni di base:

$$v_h(x) \ = \ \sum_{i=1}^{N_h} v_i \varphi_i(x).$$

   Si ha dunque il seguente problema. Trovare $u_1, u_2, \dots, u_{N_h} \in \mathbb{R}$ tali che:

$$\int_0^1 \sum_{j=1}^{N_h} u_j \varphi_j'(x) \varphi_i'(x) \mathrm{d}x = \int_0^1 f(x) \varphi_i(x) \mathrm{d}x, \quad \forall i = 1, \dots, N_h.$$

5. Utilizziamo la linearità dell'integrale per affermare che questa espressione è equivalente a un sistema lineare di N_h equazioni in N_h incognite. Trovare $u_1, u_2, \dots, u_{N_h} \in \mathbb{R}$ tali che:

$$\sum_{j=1}^{N_h} u_j \int_0^1 \varphi_j'(x) \varphi_i'(x) \mathrm{d}x = \int_0^1 f(x) \varphi_i(x) \mathrm{d}x, \quad \forall i = 1, \dots, N_h.$$

   Definiamo:

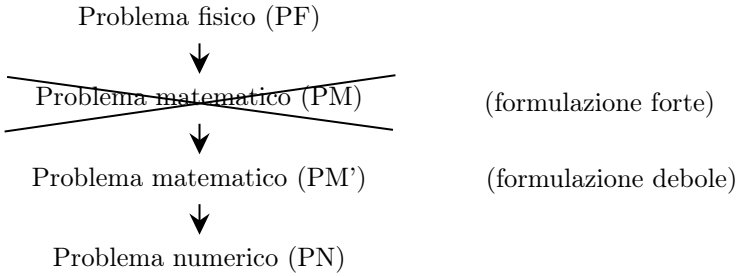
$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N_h} \end{bmatrix} \in \mathbb{R}^{N_h}, \quad \mathbf{F} = \begin{bmatrix} \int_0^1 f(x)\varphi_1(x)\mathrm{d}x \\ \int_0^1 f(x)\varphi_2(x)\mathrm{d}x \\ \vdots \\ \int_0^1 f(x)\varphi_{N_h}(x)\mathrm{d}x \end{bmatrix} \in \mathbb{R}^{N_h},$$

$$A = \begin{bmatrix} \int_0^1 \varphi_1'(x)\varphi_1'(x)\mathrm{d}x & \int_0^1 \varphi_1'(x)\varphi_2'(x)\mathrm{d}x & \dots & \int_0^1 \varphi_1'(x)\varphi_{N_h}'(x)\mathrm{d}x \\ \int_0^1 \varphi_2'(x)\varphi_1'(x)\mathrm{d}x & \int_0^1 \varphi_2'(x)\varphi_2'(x)\mathrm{d}x & \dots & \int_0^1 \varphi_2'(x)\varphi_{N_h}'(x)\mathrm{d}x \\ \vdots & \vdots & \ddots & \vdots \\ \int_0^1 \varphi_{N_h}'(x)\varphi_1'(x)\mathrm{d}x & \int_0^1 \varphi_{N_h}'(x)\varphi_2'(x)\mathrm{d}x & \dots & \int_0^1 \varphi_{N_h}'(x)\varphi_{N_h}'(x)\mathrm{d}x \end{bmatrix}.$$

Possiamo quindi riassumere il tutto scrivendo

$$A\mathbf{u} = \mathbf{F}. \tag{1.2}$$

Di seguito un riassunto del procedimento enunciato, tipico dei problemi di matematica numerica:



Per svolgere questi passaggi è necessario saper:

- risolvere sistemi lineari,
- calcolare integrali definiti,
- approssimare funzioni e dati, e le loro derivate.

Se il problema da risolvere dipendesse anche dal tempo, avremmo bisogno di ulteriori strumenti numerici (risoluzione di sistemi di EDO non lineari) che vedremo negli ultimi capitoli.

Capitolo 2

Risoluzione di sistemi lineari con metodi diretti

Ci soffermeremo dapprima su metodi numerici per la risoluzione di sistemi lineari. Supponiamo siano assegnate la matrice $A \in \mathbb{R}^{n\times n}$ e il vettore $\mathbf{b} \in \mathbb{R}^n$. Si vuole determinare il vettore $\mathbf{x} \in \mathbb{R}^n$ tale che

$$A\mathbf{x} = \mathbf{b}.$$

Possiamo scrivere il sistema in forma estesa come

$$\sum_{j=1}^{n} a_{ij}x_j = b_i, \quad \forall i = 1, \dots, n, \quad a_{ij} = (A)_{ij}.$$

Siamo interessati a problemi *ben posti*, ossia in cui la soluzione esiste ed è unica. Le condizioni affinché ciò sia verificato possono essere diverse, tutte equivalenti tra loro.

Teorema 2.1. La soluzione $\mathbf{x} \in \mathbb{R}^n$ del sistema lineare $A\mathbf{x} = \mathbf{b}$ esiste ed è unica se e solo se una delle seguenti equivalenti condizioni è soddisfatta:

- $\det(A) \neq 0$, ovvero la matrice è invertibile;
- $\text{rank}(A) = n$, ovvero il suo rango è massimo;
- $A\mathbf{x} = \mathbf{0} \Leftrightarrow \mathbf{x} = \mathbf{0}$, ovvero l'unica soluzione del cosiddetto sistema omogeneo è $\mathbf{0}$.

Nel corso dei vari capitoli, molti risultati coinvolgeranno le matrici simmetriche e definite positive (SDP). Ricordiamo il concetto di definita positività, di cui il lettore può trovare un approfondimento nell'appendice A.27.

Definizione 2.2 — Matrice definita positiva/negativa. Una matrice quadrata $A \in \mathbb{R}^{n\times n}$ si dice definita positiva (risp. negativa) se $\mathbf{x}^T A\mathbf{x} > 0$ (risp. < 0) per ogni $\mathbf{x} \neq \mathbf{0}$, $\mathbf{x} \in \mathbb{R}^n$.

Per questo tipo di matrici, riportiamo un primo importante risultato.

Teorema 2.3. Se $A \in \mathbb{R}^{n\times n}$ è una matrice quadrata definita positiva, allora è invertibile.

Ciò si può giustificare pensando che il prodotto degli autovalori della matrice coincide col suo determinante, e se essa è definita positiva, tutti gli autovalori sono strettamente positivi, quindi il determinante è non nullo.

Per il resto della trattazione considereremo solo matrici per cui $A\mathbf{x} = \mathbf{b}$ ammette una e una sola soluzione, per avere a che fare solo con problemi ben posti.

Definizione 2.4 — Matrice triangolare. Una matrice quadrata $A \in \mathbb{R}^{n\times n}$ si dice triangolare superiore (risp. inferiore) se $a_{ij} = 0$ per ogni $i > j$ (risp. $i < j$), cioè se gli elementi sotto (risp. sopra) la diagonale si annullano.

2.1 Regola di Cramer

È una regola che permette di calcolare ogni componente della soluzione di un certo sistema lineare $A\mathbf{x} = \mathbf{b}$

$$x_i = \frac{\det(A_i)}{\det(A)} \quad \forall i = 1, \dots, n$$

dove

$$A_i = [a_1, a_2, \dots, a_{i-1}, b, a_{i+1}, \dots, a_n],$$

ovvero la matrice ottenuta sostituendo la i-esima colonna della matrice A con il termine noto.

Questo algoritmo è inutilizzabile nella pratica per il suo eccessivo carico computazionale: il costo di calcolare il determinante di una matrice $n \times n$ è $\approx n!$ operazioni, e questa regola richiede il calcolo di $n + 1$ determinanti, ossia $n + 1$ operazioni di complessità fattoriale.

Cogliamo l'occasione per far notare un aspetto particolarmente importante nella risoluzione di sistemi lineari. Il lettore forse si sarà chiesto perché, nel voler risolvere il sistema $A\mathbf{x} = \mathbf{b}$, non è stato menzionato il banale passaggio di inversione della matrice che permette di scrivere la soluzione come $\mathbf{x} = A^{-1}\mathbf{b}$. Il calcolo della matrice inversa è un'operazione che richiede la computazione di numerosi determinanti. Tale operazione operazione è estremamente costosa, come appena mostrato. Pertanto, diventa chiara la ragione per cui, salvo specifiche

esigenze, si dovrebbe evitare il calcolo di una matrice inversa. Per compiti come la risoluzione dei sistemi lineari, sono preferibili i metodi presentati in seguito, i quali consentono di determinare la soluzione, o una sua ragionevole approssimazione, senza passare per la matrice inversa.

2.2 Metodi diretti: Fattorizzazione LU

Supponiamo di saper decomporre la matrice A in questo modo:

$$A = LU \quad L, U \in \mathbb{R}^{n \times n}. \tag{2.1}$$

dove L e U sono matrici triangolari, rispettivamente triangolare inferiore (Lower) e superiore (Upper). Risolvere $A\mathbf{x} = \mathbf{b}$ è equivalente a risolvere

$$L \underbrace{U\mathbf{x}}_{\mathbf{y}} = \mathbf{b} \quad \Leftrightarrow \quad \begin{cases} L\mathbf{y} = \mathbf{b} \\ U\mathbf{x} = \mathbf{y}. \end{cases}$$

Se avessimo accesso a un algoritmo poco costoso per risolvere questi due sistemi triangolari, abbatteremmo il costo computazionale totale rispetto alla regola di Cramer.

2.2.1 Algoritmo di sostituzione in avanti

Supponiamo che L sia triangolare inferiore:

$$\underbrace{\begin{bmatrix} l_{11} & 0 & \dots & 0 \\ l_{21} & l_{22} & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ l_{n1} & l_{n2} & \dots & l_{nn} \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}}_{\mathbf{b}}. \tag{2.2}$$

Osservazione. $l_{11} \neq 0, l_{22} \neq 0, \dots, l_{nn} \neq 0$ altrimenti L sarebbe singolare, quindi anche A sarebbe singolare[1] e non rispetterebbe la condizione per l'esistenza e unicità della soluzione. Vediamo ora cosa succede alle varie equazioni risolutive del sistema (2.2)

- alla 1^a equazione:

$$l_{11}y_1 = b_1 \;\Rightarrow\; y_1 = \frac{b_1}{l_{11}};$$

[1] In una matrice triangolare, L in questo caso, gli autovalori sono gli elementi sulla diagonale, e in generale il determinante è il prodotto degli autovalori. Se un elemento della diagonale di L fosse nullo, allora anche il determinante lo sarebbe. Essendo $A = LU$, per il teorema di Binet $\det A = \det L \cdot \det U$, quindi sarebbe singolare anche A.

- alla 2^a equazione, sfruttando il calcolo appena svolto per y_1:

$$l_{21}y_1 + l_{22}y_2 = b_2 \;\Rightarrow\; y_2 = \frac{1}{l_{22}}(b_2 - l_{21}y_1);$$

- alla n-esima equazione, iterando il procedimento:

$$l_{n1}y_1 + l_{n2}y_2 + \cdots + l_{nn}y_n = b_n \;\Rightarrow\; y_n = \frac{1}{l_{nn}}\left[b_n - \sum_{j=1}^{n-1} l_{nj}y_j\right].$$

Riscriviamo ora questo algoritmo sotto forma di pseudocodice:

Algoritmo 1: Algoritmo di sostituzione in avanti

```
1 y_1 = b_1 / l_11
2 for i = 2 to n do
3 |   y_i = (1/l_ii) ( b_i - sum_{j=1}^{i-1} l_ij y_j )
4 |   if criterio di arresto then
5 |   |   termina algoritmo
6 |   end
7 end
```

Osservazione. Determiniamo il costo computazionale dell'algoritmo:

- 1 divisione fuori dal ciclo;
- per ogni passo del ciclo $(i-1)$ prodotti, $(i-1)$ somme e 1 divisione per il calcolo di y_i.

Quindi il costo totale è:

$$\begin{aligned}
1 + \sum_{i=2}^{n}[(i-1) + (i-1) + 1] &= 1 + \sum_{i=2}^{n}[2i-1] \\
&= 1 + \sum_{i=1}^{n}[2i-1] - 1 \\
&= \sum_{i=1}^{n}[2i-1] \\
&= 2\sum_{i=1}^{n} i - n \\
&= 2\frac{n(n+1)}{2} - n \approx n^2 \text{ operazioni,}
\end{aligned}$$

che risulta molto inferiore al $n!$ della regola di Cramer. In generale, n^2 è la soglia minima possibile per il numero di operazioni di un algoritmo di algebra matriciale, visto che è lo stesso ordine di grandezza del numero di elementi della matrice, ed è pertanto ottimale per un metodo numerico.

2.2.2 Algoritmo di sostituzione all'indietro

Supponiamo invece che la matrice U sia triangolare superiore:

$$\underbrace{\begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & u_{nn} \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{\mathbf{y}}. \tag{2.3}$$

Osservazione. $u_{11} \neq 0, u_{22} \neq 0, \dots, u_{nn} \neq 0$, per le stesse ragioni enunciate per L.

L'idea è fare la stessa sostituzione della sezione precedente, ma andando a ritroso partendo dall'ultima riga:

Algoritmo 2: Algoritmo di sostituzione all'indietro

```
1 x_n = y_n / u_nn
2 for i = n − 1 to 1 do
3 |   x_i = (1/u_ii) ( y_i − Σ_{j=i+1}^{n} u_ij x_j )
4 |   if criterio di arresto then
5 |   |   termina algoritmo
6 |   end
7 end
```

Osservazione. Determiniamo il costo computazionale dell'algoritmo. Ci aspettiamo che sia lo stesso della sostituzione in avanti:

- 1 divisione fuori dal ciclo;
- per ogni passo del ciclo $(n-i)$ prodotti, $(n-i)$ somme e 1 divisione per il calcolo di x_i.

Il costo è dunque:

$$\begin{aligned} 1 + \sum_{i=n-1}^{1} [(n-i) + (n-i) + 1] &= 1 + \sum_{i=1}^{n-1} [2n - 2i + 1] \\ &= 1 + 2n(n-1) - 2\sum_{i=1}^{n-1} i + (n-1) \end{aligned}$$

$$= 1 + 2n(n-1) - 2\frac{(n-1)n}{2} + (n-1)$$
$$= 1 + 2n^2 - 2n - n^2 + n + n - 1 = n^2 \text{ operazioni}$$

Ricapitolando:

$$A\mathbf{x} = \mathbf{b} \Leftrightarrow \begin{cases} L\mathbf{y} = \mathbf{b} \\ U\mathbf{x} = \mathbf{y}. \end{cases}$$

Se riuscissimo a trovare la scomposizione $A = LU$ dove L è una matrice triangolare inferiore (che si risolve con sostituzione in avanti) e U triangolare superiore (che si risolve con sostituzione all'indietro), riusciremmo a risolvere il sistema con solo approssimativamente $\approx 2n^2$ operazioni, aggirando l'enorme costo della regola di Cramer.

2.2.3 Come trovare la fattorizzazione LU

Esempio.

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix}$$

in questo sistema abbiamo 3+3 incognite nel menbro destro e i 4 vincoli del membro sinistro. Questo problema è irrisolvibile in quanto sottodimensionato: abbiamo troppe incognite rispetto ai vincoli, il che produrrebbe infinite fattorizzazioni possibili. Dobbiamo dunque eliminare almeno 2 incognite. Per esempio, possiamo fissare gli elementi $l_{11} = 1, l_{22} = 1$. Avendo ora incognite e vincoli in egual numero, siamo nelle condizioni di poter trovare una soluzione, e non ci resta che svolgere i calcoli per trovarla.

Definizione 2.5 — Fattorizzazione LU. Una matrice quadrata $A \in \mathbb{R}^{n\times n}$ ammette una fattorizzazione LU se esistono $L \in \mathbb{R}^{n\times n}$ matrice triangolare inferiore tale che $l_{ii} = 1\ \forall i = 1, \ldots, n$ e $U \in \mathbb{R}^{n\times n}$ matrice triangolare superiore tali che $A = LU$.

I valori degli elementi sulla diagonale principale di L sono irrilevanti a patto che non siano nulli. Per ragioni di stabilità numerica degli algoritmi, li scegliamo uguali a 1. Ricordiamo infatti che gli autovalori di una matrice triangolare sono i valori sulla diagonale principale.

Definizione 2.6 — Minori principali. Data una matrice quadrata $A \in \mathbb{R}^{n\times n}$, si definiscono **minori principali** di ordine k, con $k = 0, \ldots, n$, le sottomatrici quadrate di dimensione k ottenute eliminando righe e colonne *dello stesso indice*. Più in particolare, si definisce **minore principale di testa** (o **minore Nord-Ovest**) di ordine k quel minore principale ottenuto eliminando le ultime $(n-k)$ righe e colonne, con $0 \leq k \leq n-1$.

Naturalmente quindi una matrice quadrata di dimensione n ha esattamente n minori principali di testa.

Teorema 2.7 — Esistenza e unicità della fattorizzazione LU. Sia $A \in \mathbb{R}^{n\times n}$ una matrice non singolare. La matrice A ammette una e una sola fattorizzazione LU se e solo se tutti i minori principali di testa A_i di ordine $i = 1, \dots, n$ sono non singolari.

Esempio.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 0 & 7 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

I suoi minori di testa principali sono:

$$A_1 = [1] \qquad A_2 = \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix} \qquad A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 0 & 7 & 1 \end{bmatrix}.$$

Essendo tutti non singolari, la matrice soddisfa le ipotesi del teorema 2.7. Per determinare i coefficienti delle matrici L e U si scrivono tutti i prodotti riga per colonna e si risolvono i sistemi:

$$\begin{aligned} u_{11} &= 1 \\ u_{12} &= 2 \\ u_{13} &= 3 \\ l_{21}u_{11} = 4 &\Rightarrow l_{21} = 4 \\ l_{21}u_{12} + u_{22} = 5 &\Rightarrow u_{22} = 5 - 4 \cdot 2 = -3 \\ l_{21}u_{13} + u_{23} = 6 &\Rightarrow u_{23} = 6 - 4 \cdot 3 = -6 \\ l_{31}u_{11} = 0 &\Rightarrow l_{31} = 0 \\ l_{31}u_{12} + l_{32}u_{22} = 7 &\Rightarrow l_{32} = -7/3 \\ l_{31}u_{13} + l_{32}u_{23} + u_{33} = 1 &\Rightarrow u_{33} = 1 - (-7/3)(-6) = -13. \end{aligned}$$

Pertanto:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 0 & 7 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 0 & -\frac{7}{3} & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & 0 & -13 \end{bmatrix}.$$

Richiamiamo ora una definizione dall'algebra lineare che ci sarà utile nelle prossime pagine.

Definizione 2.8 — Predominanza diagonale stretta. Una matrice $A \in \mathbb{R}^{n\times n}$ si dice:

- a **predominanza diagonale stretta per righe** se

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|, \quad \forall i = 1, \ldots, n$$

  cioè se ogni elemento della diagonale è in modulo maggiore strettamente della somma degli altri elementi sulla sua stessa riga.

- a **predominanza diagonale stretta per colonne** se

$$|a_{jj}| > \sum_{\substack{i=1 \\ i\neq j}}^{n} |a_{ij}|, \quad \forall j = 1, \ldots, n.$$

Esempio. La matrice

$$\begin{bmatrix} -3 & 1 & 1 \\ 0 & 2 & 1 \\ 1 & 0 & -4 \end{bmatrix}$$

è a predominanza diagonale stretta per righe. Infatti:

$$\begin{aligned} |-3| &> |1| + |1| \\ |+2| &> |0| + |1| \\ |-4| &> |1| + |0|, \end{aligned}$$

mentre la matrice

$$\begin{bmatrix} 5 & -1 & -2 \\ -1 & 2 & -1 \\ 2 & -1 & 4 \end{bmatrix}$$

non lo è. Infatti:

$$\begin{aligned} |5| &> |-1| + |-2| \\ |2| &\not> |-1| + |-1| \\ |4| &> |+2| + |-1|. \end{aligned}$$

Esempio. La matrice

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

è a predominanza diagonale stretta per colonne. Infatti:

$$\begin{aligned} |3| &> |1| + |0| \\ |2| &> |1| + |0| \\ |3| &> |1| + |1|. \end{aligned}$$

Esistono delle particolari classi di matrici in cui la fattorizzazione LU esiste ed è unica:

Teorema 2.9 — Condizioni sufficienti per fattorizzazione LU. Sia $A \in \mathbb{R}^{n\times n}$ non singolare:

- Se A è a predominanza diagonale stretta per righe o per colonne allora $\exists!$ fattorizzazione LU.
- Se A è SDP (simmetrica e definita positiva) allora $\exists!$ fattorizzazione LU.

2.3 Metodo di eliminazione gaussiana (MEG)

Presentiamo ora il metodo ideato da Gauss per la risoluzione di sistemi lineari. L'idea generale è manipolare in maniera incrementale la matrice di partenza in modo da renderla, dopo n passaggi, una matrice triangolare superiore, da risolvere successivamente con il metodo di sostituzione all'indietro. In un sistema lineare possiamo sommare e sottrarre tra loro due righe e moltiplicare due righe per uno scalare non nullo, lasciando inalterato il sistema. Vogliamo rendere nulli tutti gli elementi sulla prima colonna, dalla seconda riga in poi; poi sulla seconda colonna, dalla terza riga in poi e così via.

$$A = A^{(1)} \to A^{(2)} \to A^{(3)} \to \cdots \to A^{(k)} \to \cdots \to A^{(n)}$$

$$\underbrace{\begin{bmatrix} a_{11}^{(1)} & \dots & \dots & \dots & \dots & a_{1n}^{(1)} \\ \vdots & & & & & \vdots \\ \vdots & & & & & \vdots \\ \vdots & & & & & \vdots \\ \vdots & & & & & \vdots \\ a_{n1}^{(1)} & \dots & \dots & \dots & \dots & a_{nn}^{(1)} \end{bmatrix}}_{A=A^{(1)}} \to \underbrace{\begin{bmatrix} a_{11}^{(1)} & \dots & \dots & \dots & \dots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \dots & \dots & \dots & a_{2n}^{(2)} \\ \vdots & \vdots & & & & \vdots \\ \vdots & \vdots & & & & \vdots \\ \vdots & \vdots & & & & \vdots \\ 0 & a_{n2}^{(2)} & \dots & \dots & \dots & a_{nn}^{(2)} \end{bmatrix}}_{A^{(2)}} \to$$

$$\to \dots \to A^{(k)} = \begin{bmatrix} a_{11}^{(1)} & \dots & \dots & \dots & \dots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & & & & \vdots \\ \vdots & \ddots & \ddots & & & \vdots \\ \vdots & & 0 & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{bmatrix}.$$

Al passo n-esimo (l'ultimo) la matrice avrà la forma richiesta al fattore U:

$$A^{(n)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \dots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \dots & a_{2n}^{(2)} \\ \vdots & \ddots & a_{33}^{(3)} & \dots & a_{3n}^{(3)} \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \dots & \dots & 0 & a_{nn}^{(n)} \end{bmatrix}. \tag{2.4}$$

Per fare ciò, ricaveremo una serie di *moltiplicatori* m_{ik} dividendo ogni elemento della riga per l'elemento pivotale, cioè quello sulla diagonale. In particolare, vediamo lo pseudo-codice dell'algoritmo, che processerà la k-esima riga al k-esimo passo:

Algoritmo 3: Metodo di Eliminazione Gaussiana

```
for k = 1 to n − 1 do
    for i = k + 1 to n do
        if a_kk = 0 then
            termina algoritmo senza successo
        else
            m_ik = a_ik / a_kk
            for j = k to n do
                a_ij = a_ij − m_ik · a_kj
            end
        end
    end
end
```

Ovviamente, l'algoritmo prosegue solo fintanto che non troviamo un **elemento pivotale** nullo, ovvero $a_{kk} = 0$, visto che è il numero per cui dividiamo per trovare i moltiplicatori m_{ik}.

Esempio. Indichiamo con $(I), (II)$, etc la prima, seconda, etc riga della matrice.

$$\underbrace{\begin{bmatrix} 1 & 2 & 3 \\ 2 & 2 & 2 \\ 3 & 7 & 4 \end{bmatrix}}_{A^{(1)}} \xrightarrow[(III)=(III)-3I]{(II)=(II)-2I} \underbrace{\begin{bmatrix} 1 & 2 & 3 \\ 0 & -2 & -4 \\ 0 & 1 & -5 \end{bmatrix}}_{A^{(2)}} \xrightarrow{(III)=(III)+\frac{(II)}{2}} \underbrace{\begin{bmatrix} 1 & 2 & 3 \\ 0 & -2 & -4 \\ 0 & 0 & -7 \end{bmatrix}}_{A^{(3)}}$$

Osservazione.

Al passo k-esimo del primo ciclo sono necessarie:

- $(n - k)$ divisioni per il calcolo dei moltiplicatori,

- $(n-k)(n-k+1)$ moltiplicazioni e $(n-k)(n-k+1)$ somme per il calcolo di a_{ij},
- $(n-k)$ moltiplicazioni per il calcolo di b_j.

Si ha quindi un costo computazionale totale di:

$$\sum_{k=1}^{n-1}[(n-k)+2(n-k)(n-k+1)+(n-k)] \approx \frac{2}{3}n^3 \text{ operazioni,}$$

questo usando le identità

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2} \quad \sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}.$$

A questo costo va aggiunto il costo della sostituzione all'indietro del sistema triangolare:

$$\text{costo totale} = \text{eliminazione} + \text{sostituzione all'indietro} = \frac{2}{3}n^3 + \frac{n^2}{2} \approx \frac{2}{3}n^3.$$

Osserviamo ora che il MEG può essere usato non solo per determinare la soluzione sistema lineare, ma anche per determinare la fattorizzazione LU. Il passo di eliminazione porta come risultato una matrice triangolare superiore, che possiamo pensare come la candidata a diventare la matrice U. Per ricavare L, non resta che trovare l'unica matrice che moltiplicata per U produce A. Sotto le ipotesi di uno dei teoremi di unicità sopra citati, siamo in grado affermare che L è effettivamente unica. C'è modo di modificare l'algoritmo in modo da tenere traccia anche della matrice L per non doverla ricostruire a posteriori. Si può dimostrare che infatti L è la matrice dei moltiplicatori:

$$L = \begin{bmatrix} 1 & 0 & \dots & 0 \\ m_{21} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ m_{n1} & \dots & m_{n,n-1} & 1 \end{bmatrix}.$$

In termini di costo computazionale, le operazioni svolte finora si riassumono come segue:

costo	azione da eseguire
$\approx 2n^2$	risoluzione di 2 sistemi triangolari
$\approx 2/3\ n^3$	trovare la fattorizzazione con MEG
$\approx 2/3\ n^3$	risoluzione complessiva del sistema

La soluzione presentata è quindi molto più vantaggiosa del costo fattoriale della regola di Cramer.

2.4 Tecniche di Pivoting

Esempio. Consideriamo ora una matrice

$$A = \begin{bmatrix} 1 & 1 & 3 \\ 2 & 2 & 2 \\ 3 & 6 & 4 \end{bmatrix},$$

facciamo un passo del MEG:

$$A^{(2)} = \begin{bmatrix} 1 & 1 & 3 \\ 0 & 0 & -1 \\ 0 & 3 & -5 \end{bmatrix}.$$

L'elemento pivotale $a_{22}^{(2)} = 0$ quindi il MEG si interrompe. Per risolvere questo intoppo, introduciamo le tecniche di **pivoting**.

L'idea in questo caso è scambiare la II e III riga. Si noti inoltre che il minore nord-ovest di ordine 2 della matrice A è singolare:

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

È quindi violata la condizione necessaria e sufficiente per la fattorizzazione LU. Dobbiamo trovare un'altra disposizione di righe o colonne, creando una nuova matrice PA che verifichi la condizione e ci permetta di fare MEG per trovare la fattorizzazione LU, pre-moltiplicando A per una matrice P appropriata. In generale, nell'implementazione di un algoritmo dovremo includere precise istruzioni per il calcolatore su come effettuare il pivoting.

Ipotizziamo che al passo k del MEG troviamo l'elemento pivotale $a_{kk}^{(k)} = 0$:

$$\begin{bmatrix} a_{11}^{(1)} & \dots & \dots & \dots & a_{1n}^{(1)} \\ & \ddots & & & \vdots \\ & & a_{kk}^{(k)} = 0 & \dots & a_{kn}^{(k)} \\ & & \vdots & & \\ & & \vdots & & \end{bmatrix}.$$

Scorriamo la k-esima colonna finché troviamo un elemento non nullo. Ancora meglio, troviamo l'elemento *più grande in modulo* della colonna, e quello corrisponderà alla riga che vogliamo scambiare con la k-esima. Questa scelta dell'elemento è giustificata dal fatto che il calcolatore fa degli errori di approssimazione dato che deve usare un sistema floating-point. Si può vedere che l'operazione di divisione è ad alto rischio se il denominatore è molto piccolo, per

cui dato che l'elemento pivotale, come visto, andrà a denominatore, scegliamo l'elemento in modulo maggiore, così da minimizzare gli errori di rappresentazione.

Questa tecnica prende il nome di MEG con **pivoting per righe**. Scambiare righe di A è equivalente a pre-moltiplicare per una appropriata matrice P. Troviamo quindi una nuova matrice PA tale che

$$PA = LU.$$

P è detta **matrice di permutazione** ed è una matrice binaria, ossia composta da soli 0 e 1, che tiene traccia degli scambi di riga. Nell'esempio di prima, per tenere traccia dello scambio tra la II e III riga, si ha la seguente matrice P:

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

Il problema iniziale diventa allora

$$PA\mathbf{x} = P\mathbf{b} \quad \Leftrightarrow \quad L\underbrace{U\mathbf{x}}_{\mathbf{y}} = P\mathbf{b} \quad \Leftrightarrow \quad \begin{cases} L\mathbf{y} = P\mathbf{b} \\ U\mathbf{x} = \mathbf{y}. \end{cases}$$

Similmente, esiste anche il **pivoting per colonne**, ma Matlab raramente lo utilizza.

$$A^{(k)} = \begin{bmatrix} a_{11}^{(1)} & \cdots & \cdots & & \\ & \ddots & & & \\ & & a_{kk}^{(k)} = 0 & \cdots & \cdots \\ & & \vdots & & \\ & & a_{nk}^{(k)} & & \end{bmatrix}$$

Un punto a cui bisogna dedicare particolare attenzione è che il pivoting per colonne *cambia l'ordine delle incognite*. Il pivoting per colonne induce una scomposizione del tipo

$$AQ = LU$$

con Q matrice di permutazione. Il problema iniziale diventa allora

$$A\mathbf{x} = \mathbf{b} \quad \Leftrightarrow \quad \underbrace{AQ}_{LU}\underbrace{Q^T\mathbf{x}}_{\mathbf{z}} = \mathbf{b} \quad \Leftrightarrow \quad L\underbrace{U\mathbf{z}}_{\mathbf{y}} = \mathbf{b} \quad \Leftrightarrow \quad \begin{cases} L\mathbf{y} = \mathbf{b} \\ U\mathbf{z} = \mathbf{y} \\ Q^T\mathbf{x} = \mathbf{z}. \end{cases}$$

Infine, è possibile anche applicare contemporaneamente il pivoting per righe e per colonne.

2.5 Casi particolari di Fattorizzazioni LU

In questa sezione presentiamo alcuni casi particolari di matrici in cui è particolarmente comodo trovare, e gestire a livello di memoria, la fattorizzazione LU. Vedremo prima le matrici SDP e poi le matrici tridiagonali.

2.5.1 Matrici simmetriche e definite positive

Supponiamo che A sia simmetrica e definita positiva. In tal caso sappiamo dal teorema già visto 2.9 che esiste ed è unica la fattorizzazione LU. In questo caso, essa prende il nome di fattorizzazione di Cholesky.

Teorema 2.10 — Fattorizzazione di Cholesky. Sia A SDP. Allora $\exists!$ matrice H triangolare inferiore con elementi positivi sulla diagonale tale che

$$A = HH^T.$$

Questo equivale a pensare la risoluzione del sistema come:

$$A\mathbf{x} = \mathbf{b} \quad \Leftrightarrow \quad \begin{cases} H\mathbf{y} = \mathbf{b} \\ H^T\mathbf{x} = \mathbf{y}. \end{cases}$$

Osservazione. Possiamo sfruttare lo spazio di memoria che avevamo già allocato per tenere sia la matrice A sia il suo fattore di Cholesky H:

$$\begin{bmatrix} \ddots & & A \\ & \ddots & \\ H & & \ddots \end{bmatrix},$$

in particolare, è *sempre* conveniente quando la matrice è SDP.

Algoritmo 4: Fattorizzazione di Cholesky

$h_{11} = \sqrt{a_{11}}$
for $i = 2$ **to** n **do**
  for $j = 1$ **to** $i - 1$ **do**
    $h_{ij} = \frac{1}{h_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} h_{ik} h_{jk} \right)$
  end
  $h_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} h_{ik}^2}$
end

Questo algoritmo costa $\approx \frac{n^3}{3}$, circa la metà del MEG.

2.5.2 Matrici tridiagonali

Supponiamo che A sia tridiagonale, ovvero che abbia questa forma:

$$A = \begin{bmatrix} a_1 & c_1 & & & 0 \\ b_2 & a_2 & c_2 & & \\ & b_3 & a_3 & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ 0 & & & b_n & a_n \end{bmatrix}.$$

Nel caso di matrici tridiagonali si usa l'**algoritmo di Thomas** che scrive la fattorizzazione e istruisce su come risolvere i due sistemi a valle della fattorizzazione. Necessitiamo solo di 3 vettori per memorizzare completamente la matrice:

$$\begin{aligned} \mathbf{a} &= [a_1, \dots, a_n]^T \in \mathbb{R}^n \\ \mathbf{b} &= [b_2, \dots, b_n]^T \in \mathbb{R}^{n-1} \\ \mathbf{c} &= [c_1, \dots, c_{n-1}]^T \in \mathbb{R}^{n-1}. \end{aligned}$$

Si dimostra che i fattori L ed U di A hanno la seguente struttura:

$$L = \begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ \beta_2 & 1 & & & \vdots \\ 0 & \beta_3 & 1 & & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \beta_n & 1 \end{bmatrix} \qquad U = \begin{bmatrix} \alpha_1 & c_1 & 0 & \dots & 0 \\ 0 & \alpha_2 & c_2 & \ddots & \vdots \\ \vdots & & \alpha_3 & \ddots & 0 \\ \vdots & & & \ddots & c_{n-1} \\ 0 & \dots & \dots & 0 & \alpha_n \end{bmatrix}.$$

Si noti che i termini c_k sono rimasti invariati. Per determinare i fattori L ed U è quindi sufficiente determinare:

$$\alpha_1, \alpha_2, \dots, \alpha_n \quad \text{e} \quad \beta_1, \beta_2, \dots, \beta_n.$$

Questi si possono determinare risolvendo col seguente algoritmo.

Algoritmo 5: Algoritmo di Thomas

```
1 α_1 = a_1
2 for i = 2 to n do
3 |   β_i = b_i / α_{i-1}
4 |   α_i = a_i − β_i c_{i-1}
5 end
```

Possiamo usare queste relazioni per risolvere direttamente i sistemi triangolari a valle della fattorizzazione trovata:

- per risolvere $L\mathbf{y} = \mathbf{b}$

$$y_1 = b_1, \quad y_i = b_i - \beta_i y_{i-1}, \quad i = 2, \dots, n,$$

- per risolvere $U\mathbf{x} = \mathbf{y}$

$$x_n = \frac{y_n}{\alpha_n}, \quad x_i = \frac{y_i - c_i x_{i+1}}{\alpha_i}, \quad i = n-1, \dots, 1.$$

Usare matrici tridiagonali è particolarmente vantaggioso perché abbatte il costo computazionale a un numero di operazioni proporzionale a n, invece che a n^2 o più. In particolare, si hanno $8n - 7$ operazioni.

Esempio ($n = 3$). Assegnati $a_1, a_2, a_3, c_1, c_2, l_2, l_3$ trovare $\alpha_1, \alpha_2, \alpha_3, \beta_2, \beta_3$.

$$\underbrace{\begin{bmatrix} a_1 & c_1 & 0 \\ l_2 & a_2 & c_2 \\ 0 & l_3 & a_3 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ \beta_2 & 1 & 0 \\ 0 & \beta_3 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} \alpha_1 & c_1 & 0 \\ 0 & \alpha_2 & c_2 \\ 0 & 0 & \alpha_3 \end{bmatrix}}_{U}$$

Allora

$$\begin{aligned} a_1 = 1 \cdot \alpha_1 &\Rightarrow \alpha_1 = a_1 \\ l_2 = \beta_2 \alpha_1 &\Rightarrow \beta_2 = l_2/\alpha_1 \\ a_2 = \beta_2 c_1 + \alpha_2 &\Rightarrow \alpha_2 = a_2 - \beta_2 c_1 \\ l_3 = \beta_3 \alpha_2 &\Rightarrow \beta_3 = l_3/\alpha_2 \\ a_3 = \beta_3 c_2 + \alpha_3 &\Rightarrow \alpha_3 = a_3 - \beta_3 c_2. \end{aligned}$$

2.6 Condizionamento di una matrice

Gli algoritmi che abbiamo visto finora manipolano la matrice interessata. Questo comporta che il calcolatore debba fare delle operazioni floating-point, a cominciare dal memorizzare certi numeri. Queste operazioni saranno inevitabilmente approssimazioni, in quanto è spesso impossibile avere calcoli esattamente corretti. Per esempio, un numero irrazionale dovrà essere troncato visto che ha infinite

cifre decimali, e in particolare dopo circa 16 cifre dopo la virgola in una normale architettura floating-point.

È importante, quindi, capire quanto queste approssimazioni influiscano sul risultato finale. Questi metodi, tra cui la fattorizzazione LU, cadono nella categoria dei *metodi diretti*. Assumendo di non manipolare la matrice, possiamo invece usare dei **metodi indiretti** che ci permettono di approssimare la *soluzione esatta* $\mathbf{x}$ con una **soluzione approssimata** $\tilde{\mathbf{x}}$. In tal caso è però anche necessario saper stimare dall'alto l'errore: $\|\mathbf{x} - \tilde{\mathbf{x}}\|$ deve essere minore di una certa quantità nota, ovvero che non dipenda dalla soluzione esatta che è incognita. In generale, l'errore deve anche poter essere ridotto a piacere, sotto una soglia di tolleranza arbitrariamente fissata.

La soluzione approssimata $\tilde{x}$ può essere vista come la soluzione esatta di un certo **sistema perturbato**.

Il **numero di condizionamento** di una matrice rivela quanto essa sia "buona" o "cattiva", ossia in quale misura eseguire operazioni su di essa provochi errori di rappresentazione e computazione.

Il **teorema di stabilità** riassumerà quando una matrice è "buona" o "cattiva" e come ciò influenzi il risultato del calcolatore rispetto alla soluzione esatta.

Definizione 2.11 — Norma L^p di un vettore. Sia $\mathbf{z} \in \mathbb{R}^n$. Definiamo la sua norma L^p come:

$$\|\mathbf{z}\|_p = \left(\sum_{i=1}^{n} |z_i|^p \right)^{1/p} \quad \forall p \in [1, +\infty).$$

Esempi di norme di vettore.

1. Se $p = \infty$, si ha $\|\mathbf{z}\|_\infty = \max_{i=1,\dots,n} |z_i|$.
2. Se $p = 2$, si ha la **norma euclidea** (vedi A.7).

Possiamo usare la definizione di norma di vettore ed estenderla a norma di matrice.

Definizione 2.12 — Norma L^p di una matrice. Sia $A \in \mathbb{R}^{n \times n}$. Definiamo la sua norma L^p come:

$$\|A\|_p = \sup_{\substack{\mathbf{z} \in \mathbb{R}^n \\ \mathbf{z} \neq \mathbf{0}}} \frac{\|A\mathbf{z}\|_p}{\|\mathbf{z}\|_p} \quad \forall p \in [1, +\infty).$$

È detta **norma indotta**, cioè è indotta da quella vettoriale.

Esempi di norme matriciali.

- Se $p = 1$, si ha:
$$\|A\|_1 = \max_{j=1,\dots,n} \sum_{i=1}^{n} |a_{ij}|,$$
cioè il massimo della somma delle colonne in modulo.
- Se $p = \infty$, si ha:
$$\|A\|_\infty = \max_{i=1,\dots,n} \sum_{j=1}^{n} |a_{ij}|,$$
cioè il massimo della somma delle righe in modulo.
- Se $p = 2$, si ha $\|A\|_2 = \sqrt{\rho\left(A^T A\right)} \equiv \sqrt{\rho\left(AA^T\right)}$, definendo ρ come a seguire.

Definizione 2.13. Siano λ_i gli autovalori di M. Definiamo il **raggio spettrale** di M
$$\rho(M) = \max_{i=1,\dots,n} |\lambda_i|.$$

Se $p = 2$, $\|A\|_2$ si chiama **norma spettrale** e ha le seguenti proprietà:

- se A è SDP, allora $\|A\|_2 = \sqrt{\rho\left(A^2\right)} = \sqrt{\rho(A)^2} = \rho(A) = \lambda_{\max}(A)$,
- $\|A\|_2 \geq 0$,
- $\|\alpha A\|_2 = \alpha \|A\|_2$,
- $\|A + B\|_2 \leq \|A\|_2 + \|B\|_2$,
- $\|A\mathbf{v}\|_2 \leq \|A\|_2 \|\mathbf{v}\|_2$ con $\mathbf{v}$ vettore.

2.6.1 Il numero di condizionamento

Definizione 2.14 — Numero di condizionamento. Sia $A \in \mathbb{R}^{n\times n}$ invertibile. Definiamo il numero di condizionamento in norma p come:
$$K_p(A) = \|A\|_p \left\|A^{-1}\right\|_p \quad p = 1, 2, \dots, +\infty.$$
Se A non è invertibile, lo si definisce come $K_p(A) = \infty$.

Il numero di condizionamento riassume quanto una matrice è "buona", o "ben condizionata", nel senso spiegato prima. Tanto più il numero condizionamento della matrice è grande, tanto più è "cattiva", perché si avvicina ad una matrice singolare.

Possiamo misurare K_p rispetto a un qualsiasi p, ma solitamente useremo $p = 2$. In tal caso esso prende il nome di **numero di condizionamento spettrale**.

Proprietà.

1. $K_p(A) \geq K_p(I) = 1$: più è basso e più la matrice si comporta come l'identità[2].
2. $K_p(A) = K_p\left(A^{-1}\right)$: una matrice è "brutta" tanto quanto la sua inversa.
3. $K_p(\alpha A) = K_p(A), \quad \forall \alpha \in \mathbb{R}, \alpha \neq 0$: se abbiamo una matrice "buona" e la moltiplichiamo per uno scalare, essa continua a rimanere "buona".

Osservazione (condizionamento spettrale).

Se A è SDP, allora:

$$K_2(A) = \underbrace{\|A\|_2}_{\lambda_{\max}(A)} \quad \underbrace{\left\|A^{-1}\right\|_2}_{\frac{1}{\lambda_{\min}(A)}} = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}.$$

Più gli autovalori estremali, cioè il minimo e il massimo, di A sono lontani tra di loro, più A è una matrice mal condizionata. Il caso ottimale è che tutti gli autovalori siano uguali, in cui il loro rapporto è unitario.

Definizione 2.15 — Distanza L^p. Sia $A \in \mathbb{R}^{n\times n}$ non singolare. Definiamo la distanza L^p di A come:

$$\operatorname{dist}_p(A) = \min\left\{ \frac{\|\delta A\|_p}{\|A\|_p} : \ A + \delta A \text{ è singolare} \right\}.$$

Si può dimostrare che la distanza è inversamente proporzionale al numero di condizionamento:

$$\operatorname{dist}_p(A) = \frac{1}{K_p(A)}.$$

2.7 Analisi di stabilità

Siano:

- $A\mathbf{x} = \mathbf{b}$ il **problema originale (PO).**
- $(A + \delta A)(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b} + \delta\mathbf{b}$ il **problema perturbato (PP)**, quello che risolviamo con un calcolatore, introducendo perturbazioni per ciascuno dei tre elementi che lo compongono.

Teorema 2.16 — stabilità. Sia $A \in \mathbb{R}^{n\times n}$ non singolare. Sia inoltre $\delta A \in \mathbb{R}^{n\times n}$ una perturbazione tale che $\left\|A^{-1}\right\|_p \|\delta A\|_p < 1$ per una generica norma indotta p. Allora se $\mathbf{x} \in \mathbb{R}^n$ è soluzione di (PO) con $\mathbf{b} \in \mathbb{R}^n$, $\mathbf{b} \neq \mathbf{0}$ e $\mathbf{x}+\delta\mathbf{x} \in \mathbb{R}^n$

[2] "La matrice più *bella* di tutte".

è soluzione di (PP) per $\delta\mathbf{b} \in \mathbb{R}^n$, si ha

$$\frac{\|\delta\mathbf{x}\|_p}{\|\mathbf{x}\|_p} \leq \left[\frac{K_p(A)}{1 - K_p(A)\frac{\|\delta A\|_p}{\|A\|_p}}\right]\left(\frac{\|\delta\mathbf{b}\|_p}{\|\mathbf{b}\|_p} + \frac{\|\delta A\|_p}{\|A\|_p}\right).$$

Dimostrazione.

Si ha innanzitutto che:

$$\left\|A^{-1}\delta A\right\|_p \leq \underbrace{\left\|A^{-1}\right\|_p \ \|\delta A\|_p < 1}_{\text{per ipotesi}}.$$

Si può inoltre dimostrare che $I + A^{-1}\delta A$ è invertibile se vale l'ipotesi considerata. Inoltre, vale anche la seguente disuguaglianza:

$$\left\|\left(I + A^{-1}\delta A\right)^{-1}\right\|_p \leq \frac{1}{1 - \|A^{-1}\|_p \ \|\delta A\|_p}. \tag{2.5}$$

A questo punto prendiamo il problema perturbato (PP):

$$\begin{aligned}
(A + \delta A)(\mathbf{x} + \delta\mathbf{x}) &= \mathbf{b} + \delta\mathbf{b} \\
A\mathbf{x} + \delta A\mathbf{x} + A\delta\mathbf{x} + \delta A\delta\mathbf{x} &= \mathbf{b} + \delta\mathbf{b} \\
\delta A\mathbf{x} + A\delta\mathbf{x} + \delta A\delta\mathbf{x} &= \delta\mathbf{b} && \text{(ricordando che } A\mathbf{x} = \mathbf{b}) \\
(A + \delta A)\delta\mathbf{x} &= \delta\mathbf{b} - \delta A\mathbf{x} \\
A^{-1}[(A + \delta A)\delta\mathbf{x}] &= A^{-1}[\delta\mathbf{b} - \delta A\mathbf{x}] && \text{(moltiplicando per } A^{-1}) \\
\left(I + A^{-1}\delta A\right)\delta\mathbf{x} &= A^{-1}\delta\mathbf{b} - A^{-1}\delta A\mathbf{x}.
\end{aligned}$$

Ricordiamoci che possiamo, per ipotesi, moltiplicare per $\left(I + A^{-1}\delta A\right)^{-1}$, ottenendo:

$$\delta\mathbf{x} = \left(I + A^{-1}\delta A\right)^{-1}\left(A^{-1}\delta\mathbf{b} - A^{-1}\delta A\mathbf{x}\right).$$

Passando alle norme:

$$\|\delta\mathbf{x}\|_p \leq \left\|\left(I + A^{-1}\delta A\right)^{-1}\right\|_p \ \left\|A^{-1}\delta\mathbf{b} - A^{-1}\delta A\mathbf{x}\right\|_p.$$

Usando la disuguaglianza triangolare delle norme:

$$\|\delta\mathbf{x}\|_p \leq \left\|\left(I + A^{-1}\delta A\right)^{-1}\right\|_p \left(\left\|A^{-1}\delta\mathbf{b}\right\|_p + \left\|A^{-1}\delta A\mathbf{x}\right\|_p\right).$$

Usiamo ora la disuguaglianza (2.5):

$$\|\delta\mathbf{x}\|_p \leq \frac{1}{1 - \|A^{-1}\|_p \ \|\delta A\|_p}\left(\left\|A^{-1}\right\|_p \ \|\delta\mathbf{b}\|_p + \ \left\|A^{-1}\right\|_p \ \|\delta A\|_p \ \|\mathbf{x}\|_p\right).$$

Dividiamo per $\|\mathbf{x}\|_p$:

$$\frac{\|\delta\mathbf{x}\|_p}{\|\mathbf{x}\|_p} \leq \frac{1}{1 - \|A^{-1}\|_p \ \|\delta A\|_p} \left(\frac{\left\|A^{-1}\right\|_p \ \|\delta\mathbf{b}\|_p}{\|\mathbf{x}\|_p} + \ \left\|A^{-1}\right\|_p \ \|\delta A\|_p \right).$$

Raccogliendo $\left\|A^{-1}\right\|_p$ e dividendo e moltiplicando per $\|A\|_p$ si ottiene:

$$\frac{\|\delta\mathbf{x}\|_p}{\|\mathbf{x}\|_p} \leq \frac{\overbrace{\left\|A^{-1}\right\|_p \ \|A\|_p}^{K_p(A)}}{1 - \|A^{-1}\|_p \ \|\delta A\|_p} \left(\frac{\|\delta\mathbf{b}\|_p}{\|\mathbf{x}\|_p \|A\|_p} + \frac{\|\delta A\|_p}{\|A\|_p} \right).$$

Ricordiamo ora che:

$$A\mathbf{x} = \mathbf{b} \quad \Rightarrow \quad \|\mathbf{b}\|_p = \|A\mathbf{x}\|_p \leq \|A\|_p \|\mathbf{x}\|_p \quad \Rightarrow \quad \frac{1}{\|A\|_p \|\mathbf{x}\|_p} \leq \frac{1}{\|\mathbf{b}\|_p}.$$

Abbiamo così stimato il primo termine dentro la parentesi. Ora moltiplichiamo e dividiamo a denominatore per $\|A\|_p$ ottenendo il numero di condizionamento $K_p(A)$ anche a denominatore:

$$\frac{\|\delta\mathbf{x}\|_p}{\|\mathbf{x}\|_p} \leq \frac{K_p(A)}{1 - \|A^{-1}\|_p \|\delta A\|_p \cdot \frac{\|A\|_p}{\|A\|_p}} \left(\frac{\|\delta\mathbf{b}\|_p}{\|\mathbf{b}\|_p} + \frac{\|\delta A\|_p}{\|A\|_p} \right)$$

ovvero:

$$\frac{\|\delta\mathbf{x}\|_p}{\|\mathbf{x}\|_p} \leq \frac{K_p(A)}{1 - K_p(A) \cdot \frac{\|\delta A\|_p}{\|A\|_p}} \left(\frac{\|\delta\mathbf{b}\|_p}{\|\mathbf{b}\|_p} + \frac{\|\delta A\|_p}{\|A\|_p} \right). \qquad \blacksquare$$

Osservazione. Se $\delta A = 0$ si ha che:

$$\frac{\|\delta\mathbf{x}\|_p}{\|\mathbf{x}\|_p} \leq K_p(A) \left(\frac{\|\delta\mathbf{b}\|_p}{\|\mathbf{b}\|_p} \right).$$

Ovvero, anche se la perturbazione è molto piccola ma il condizionamento della matrice è grande, ossia la matrice è molto "brutta", l'errore rispetto alla soluzione esatta rimane comunque non trascurabile. Da ciò segue che:

Corollario 2.17. Se $\delta A = 0$, si ha che:

$$\frac{1}{K_p(A)} \frac{\|\delta\mathbf{b}\|_p}{\|\mathbf{b}\|_p} \leq \frac{\|\delta\mathbf{x}\|_p}{\|\mathbf{x}\|_p} \leq K_p(A) \ \frac{\|\delta\mathbf{b}\|_p}{\|\mathbf{b}\|_p}.$$

Cioè possiamo stimare l'errore anche dal basso. Questo è notevole perché è spesso facile trovare stime dall'alto, mentre fornire stime dal basso è più difficile, o non sempre possibile.

Dimostrazione del corollario. Scriviamo il problema perturbato con $\delta A = 0$ e passiamo alle norme:

$$\begin{aligned} A\delta\mathbf{x} &= \delta\mathbf{b} && ((\text{PP}) \text{ con } \delta A = 0) \\ \|\delta\mathbf{b}\|_p = \|A\delta\mathbf{x}\|_p &\leq \|A\|_p \|\delta\mathbf{x}\|_p. && (\text{passando alle norme}) \end{aligned}$$

Ricordiamo ora che:

$$\|\mathbf{x}\|_p = \left\|A^{-1}\mathbf{b}\right\|_p \leq \left\|A^{-1}\right\|_p \ \|\mathbf{b}\|_p.$$

e moltiplichiamo quindi per $\|\mathbf{x}\|_p$:

$$\begin{aligned} \|\mathbf{x}\|_p \ \|\delta\mathbf{b}\|_p &\leq \|A\|_p \ \|\delta\mathbf{x}\|_p \ \underbrace{\left\|A^{-1}\right\|_p \ \|\mathbf{b}\|_p}_{\|\mathbf{x}\|_p} \\ \frac{\|\delta\mathbf{b}\|_p}{\|\mathbf{b}\|_p} &\leq \frac{\|\delta\mathbf{x}\|_p}{\|\mathbf{x}\|_p} \ \underbrace{\|A\|_p \ \left\|A^{-1}\right\|_p}_{K_p(A)} \\ \frac{1}{K_p(A)} \frac{\|\delta\mathbf{b}\|_p}{\|\mathbf{b}\|_p} &\leq \frac{\|\delta\mathbf{x}\|_p}{\|\mathbf{x}\|_p}. \end{aligned}$$

■

2.8 Problema del fill-in

Supponiamo di avere una matrice tridiagonale, che sappiamo poter fattorizzare con l'**algoritmo di Thomas**. La fattorizzazione LU preserva questa struttura *a banda*:

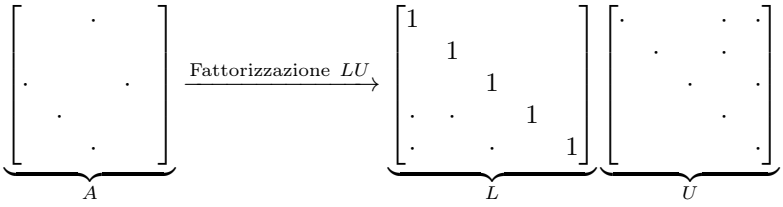
$$\underbrace{\begin{bmatrix} \cdot & \cdot & & & \\ \cdot & \cdot & \cdot & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot \end{bmatrix}}_{A} \xrightarrow{\text{Thomas}} \underbrace{\begin{bmatrix} 1 & & & & \\ \cdot & 1 & & & \\ & \cdot & 1 & & \\ & & \cdot & 1 & \\ & & & \cdot & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} \cdot & \cdot & & & \\ & \cdot & \cdot & & \\ & & \cdot & \cdot & \\ & & & \cdot & \cdot \\ & & & & \cdot \end{bmatrix}}_{U}$$

abbiamo quindi a che fare con matrici quasi vuote.

Definizione 2.18 — Matrice sparsa. Diciamo che A è **sparsa** se il numero di elementi non nulli (NNZ: Number of Non Zero) è circa $O(n)$.

Se applichiamo la fattorizzazione LU a una matrice sparsa generica (non a banda) i fattori L e U in generale *non preservano la struttura di sparsità*: anzi, talvolta sono addirittura piene. Per questa ragione, non possiamo fare previsioni su quanta memoria servirà per memorizzarle. Questo è noto come **problema del fill-in**. Per quantificare questo problema, si guarda il numero di elementi non zero di A e di LU, e si fa la loro differenza.

Esempio.



Illustriamo qualche possibile soluzione a questo problema:

1. Riordinare le righe e le colonne in modo da ottenere le righe e le colonne di A in una configurazione che minimizza il fill-in (approccio utilizzato da Matlab).
2. Cambiare completamente prospettiva, cercando di risolvere il problema *senza manipolare affatto la matrice A*. L'idea sarà sviluppata nel prossimo capitolo nei cosiddetti **metodi iterativi**.

Capitolo 3

Metodi iterativi per sistemi lineari

L'idea di un metodo iterativo per la risoluzione di un sistema $A\mathbf{x} = \mathbf{b}$ è quella di costruire una successione di vettori

$$\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(k)} \quad \text{tali che} \quad \mathbf{x}^{(k)} \xrightarrow{k\to\infty} \mathbf{x}. \tag{3.1}$$

Per ottenere tale risultato, dobbiamo saper rispondere a due questioni principali:

1. Come scegliere la successione di vettori in modo che (3.1) sia verificata?
2. Come scegliere k (dato che non possiamo svolgere infinite iterazioni) tale che $\left\|\mathbf{x} - \mathbf{x}^{(k)}\right\| < \text{TOL}$, dove TOL è una tolleranza scelta dall'utente?

La forma generale di un metodo iterativo è:

$$\text{Dato } \mathbf{x}^{(0)}, \quad \mathbf{x}^{(k+1)} = B\mathbf{x}^{(k)} + \mathbf{g}, \quad k \geq 0 \tag{3.2}$$

B è detta **matrice di iterazione** e dipende solo dalla matrice A, invece $\mathbf{g}$ può dipendere sia da A che da $\mathbf{b}$.

Presentiamo ora due concetti molto importanti per la trattazione:

- **Convergenza:**

  $$\lim_{k\to\infty} \left\|\mathbf{e}^{(k)}\right\| = 0 \quad \text{con} \quad \mathbf{e}^{(k)} = \mathbf{x} - \mathbf{x}^{(k)} \quad k = 0, 1, 2 \ldots$$

  Per avere convergenza la successione degli errori deve tendere a zero, ricordando che vogliamo poter garantire tale risultato nonostante non conosciamo la soluzione esatta $\mathbf{x}$.

- **Consistenza:**

  $$\mathbf{x} = B\mathbf{x} + \mathbf{g} \quad \left(\Rightarrow \mathbf{g} = (I - B)A^{-1}\mathbf{b}\right)$$

Se a un certo passo k troviamo la soluzione esatta, il metodo deve restituire proprio quella. In altre parole, la soluzione esatta deve soddisfare l'algoritmo, ovvero essa deve essere un punto fisso dell'algoritmo.

Controesempio. La consistenza non implica la convergenza, come si vede scegliendo $B = I, \mathbf{g} = \mathbf{0}$:

$$\mathbf{x} = I\mathbf{x} + \mathbf{0} = \mathbf{x} \quad \text{(consistenza)},$$

ma

$$\forall k,\ \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} \quad \Rightarrow \quad \lim_{n\to\infty} \mathbf{x}^{(k)} = \mathbf{x}^{(0)} \neq \mathbf{x}.$$

A meno di non essere estremamente fortunati nella scelta di $\mathbf{x}^{(0)}$.

Diamo ora un'importante definizione.

Definizione 3.1 — Residuo. Sia dato un sistema lineare $A\mathbf{x} = \mathbf{b}$. Supponiamo di usare un metodo iterativo per la sua risoluzione e chiamiamo $\mathbf{x}^{(k)}$ il vettore della soluzione all'iterata k-esima. Definiamo residuo all'iterata k-esima la quantità

$$\mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}. \tag{3.3}$$

Il residuo gioca un ruolo importante nel determinare quando è opportuno arrestare un metodo iterativo. Idealmente vorremmo fissare un errore arbitrariamente piccolo, tuttavia dobbiamo farlo senza conoscere la soluzione esatta. Ciò può apparentemente sembrare impossibile, ma vedremo degli opportuni criteri che garantiscono, ad esempio, che il residuo è una buona stima dall'alto dell'errore, in termini di norma. Chiederemo quindi che il residuo, che è una quantità calcolabile e nota ad ogni iterata, sia sufficientemente piccolo. Essendo inoltre tutte le norme non negative, potremo sostanzialmente applicare il teorema dei due carabinieri[1]:

$$0 \leq \underbrace{\left\|\mathbf{e}^{(k)}\right\|}_{\text{errore}} \leq \underbrace{\left\|\mathbf{r}^{(k)}\right\| = \left\|\mathbf{b} - A\mathbf{x}^{(k)}\right\|}_{\text{residuo}} \xrightarrow{k\to\infty} 0.$$

Teorema 3.2. Dato un metodo iterativo della forma (3.2) che sia consistente, si ha che:

$$\text{convergenza} \quad \Leftrightarrow \quad \rho(B) < 1.$$

Possiamo sapere se il metodo converge semplicemente studiando il raggio spettrale di B. Chiedere che il raggio spettrale sia minore di 1, significa chiedere che tutti gli autovalori stiano nel cerchio unitario. Si ha inoltre che se $\rho(B) \ll 1$ allora la convergenza è più veloce.

[1] Se $f(x) \leq g(x) \leq h(x),\ \forall x \in I$ e inoltre $f(x), h(x) \to l$ per $x \to x_0$, allora anche $g(x) \to l$ per $x \to x_0$.

Dimostrazione. Al fine di dimostrare questo teorema, enunciamo la seguente proprietà: se $A \in \mathbb{R}^{n\times n}$,

$$A^k \xrightarrow{k\to\infty} 0 \quad \Leftrightarrow \quad \rho(A) < 1. \tag{3.4}$$

Procediamo:

$$\begin{aligned} \mathbf{x}^{(k+1)} &= B\mathbf{x}^{(k)} + \mathbf{g} && \text{(metodo)} \\ \mathbf{x} &= B\mathbf{x} + \mathbf{g} && \text{(consistenza)} \\ \mathbf{x} - \mathbf{x}^{(k+1)} &= B(\mathbf{x} - \mathbf{x}^{(k)}) && \text{(sottraendo membro a membro)} \\ \mathbf{e}^{(k+1)} &= B\mathbf{e}^{(k)}. \end{aligned}$$

Possiamo iterare l'equazione ottenuta per scriverla in funzione dell'errore al passo iniziale:

$$\mathbf{e}^{(k)} = B\ \mathbf{e}^{(k-1)} = B\ \underbrace{B\ \mathbf{e}^{(k-2)}}_{\mathbf{e}^{(k-1)}} = \ldots = B^k\mathbf{e}^{(0)}$$

Otteniamo dunque l'**equazione dell'errore**:

$$\mathbf{e}^{(k)} = B^k\ \mathbf{e}^{(0)} \tag{3.5}$$

Dimostriamo ora i due versi della coimplicazione del teorema:

- ($\Rightarrow$) usando nell'ultimo passaggio la proprietà (3.4):

$$\underbrace{\mathbf{e}^{(k)} \xrightarrow{k\to\infty} \mathbf{0}}_{\text{convergenza}} \quad \Rightarrow \quad B^k \xrightarrow{k\to\infty} 0 \quad \Rightarrow \quad \rho(B) < 1$$

- ($\Leftarrow$) usando nel primo passaggio la proprietà (3.4):

$$\rho(B) < 1 \quad \Rightarrow \quad B^k \xrightarrow{k\to\infty} 0 \quad \Rightarrow \quad \underbrace{\mathbf{e}^{(k)} \xrightarrow{k\to\infty} \mathbf{0}}_{\text{convergenza}}. \qquad \blacksquare$$

3.1 Costruzione di metodi iterativi

Vogliamo costruire un generico metodo iterativo della forma $\mathbf{x}^{(k+1)} = B\mathbf{x}^{(k)} + \mathbf{g}$, per risolvere il sistema $A\mathbf{x} = \mathbf{b}$.

1. Sia P una matrice invertibile (ovvero con $\det(P) \neq 0$) detta **matrice di precondizionamento**. Dividiamo la matrice A in due addendi:

$$A = P - N.$$

2. Sostituiamo questa espressione nel sistema lineare:

$$A\mathbf{x} = \mathbf{b}$$

$$
\begin{aligned}
(P-N)\mathbf{x} &= \mathbf{b} \\
P\mathbf{x} &= N\mathbf{x} + \mathbf{b} \\
\mathbf{x} &= \underbrace{P^{-1}N}_{B}\mathbf{x} + \underbrace{P^{-1}\mathbf{b}}_{\mathbf{g}}.
\end{aligned}
$$

3. Il metodo iterativo diventa, per $k \geq 0$:

$$
\mathbf{x}^{(k+1)} = \underbrace{P^{-1}N}_{B}\mathbf{x}^{(k)} + \underbrace{P^{-1}\mathbf{b}}_{\mathbf{g}}. \tag{3.6}
$$

o equivalentemente, dopo un po' di passaggi:

$$
\begin{aligned}
\mathbf{x}^{(k+1)} &= P^{-1}N\mathbf{x}^{(k)} + P^{-1}\mathbf{b} \\
\mathbf{x}^{(k+1)} &= P^{-1}(P-A)\mathbf{x}^{(k)} + P^{-1}\mathbf{b} \\
\mathbf{x}^{(k+1)} &= \left(I - P^{-1}A\right)\mathbf{x}^{(k)} + P^{-1}\mathbf{b} \\
\mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} - P^{-1}A\mathbf{x}^{(k)} + P^{-1}\mathbf{b} \\
\mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + P^{-1}\left(-A\mathbf{x}^{(k)} + \mathbf{b}\right).
\end{aligned}
$$

In quest'ultima espressione riconosciamo i termini

$$
\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \overbrace{P^{-1}\underbrace{\left(\mathbf{b} - A\mathbf{x}^{(k)}\right)}_{\text{residuo } \mathbf{r}^{(k)}}}^{\mathbf{z}^{(k)}}. \tag{3.7}
$$

Osservazioni.

- P caratterizza un metodo con questa forma; quindi è sufficiente scegliere P per ottenere il metodo.
- Un metodo così costruito è automaticamente consistente per costruzione, poiché lo abbiamo ottenuto a partire da $A\mathbf{x} = \mathbf{b}$.
- Per calcolare $P^{-1}\mathbf{r}^{(k)}$, non serve calcolare l'inversa, bensì basta risolvere il sistema $\mathbf{z}^{(k)} = P^{-1}\mathbf{r}^{(k)}$[2]. Ovviamente risolvere questo sistema dovrebbe essere notevolmente più facile rispetto a quello iniziale $A\mathbf{x} = \mathbf{b}$, a patto di aver scelto una matrice di precondizionamento P semplice, per esempio diagonale o triangolare.
- La **velocità di convergenza** di un metodo è data da: $R(B) = -\log(\rho_B)$.

In pseudocodice, gli algoritmi descritti hanno la seguente forma:

[2]In generale, l'inversione di matrice è un'operazione dispendiosa e potenzialmente instabile dal punto di vista numerico, quindi si cerca di utlizzarla il meno possibile, preferendo risoluzioni di sistemi lineari come nell'esempio appena mostrato.

Algoritmo 6: Algoritmo iterativo per la forma (3.6)

```
1 inizializza x^(0)
2 for k = 0, 1, ... do
3 |   risolvi x^(k+1) = P^{-1} N x^(k) + P^{-1} b
4 |   if criterio di arresto then
5 |   |   termina algoritmo
6 |   end
7 end
```

Vediamo ora l'altra forma, che mette in evidenza il ruolo di P:

Algoritmo 7: Algoritmo iterativo per la forma (3.7)

```
1  inizializza x^(0)
2  calcola r^(0) = b − A x^(0)
3  for k = 0, 1, ... do
4  |   risolvi z^(k) = P^{-1} r^(k) con un metodo diretto
5  |   aggiorna x^(k+1) = x^(k) + z^(k)
6  |   aggiorna r^(k+1) = b − A x^(k+1)
7  |   if criterio di arresto then
8  |   |   termina algoritmo
9  |   end
10 end
```

NB. Per il teorema visto prima, la matrice di iterazione associata

$$\begin{aligned} B &= P^{-1}N \\ &= P^{-1}(P - A) \\ &= I - P^{-1}A \end{aligned}$$

converge se e solo se $\rho\left(I - P^{-1}A\right) < 1$.

Il motivo per cui servono sia P che la sua inversa è che P^{-1} compare nel passo per trovare $\mathbf{z}^{(k)}$, mentre P compare nel passo per aggiornare il residuo, precisamente in $A = P - N$.

Il prossimo passo è trovare alcune scelte intelligenti per P.

3.2 Metodo di Jacobi

Nel metodo di Jacobi, scomporremo A in tre addendi: la sua diagonale e le parti triangolari inferiori e superiori:

$$A = \begin{bmatrix} \ddots & & -F \\ & D & \\ -E & & \ddots \end{bmatrix} = D - E - F.$$

In particolare, prendiamo come precondizionatore D, la diagonale di A:

$$P = D.$$

Di conseguenza, la scomposizione di A diventa:

$$A = P - N,\ P = D \quad \Rightarrow \quad A = D - N.$$

Inoltre, abbiamo:

$$A = D - E - F \quad \Rightarrow \quad N = E + F.$$

La matrice di iterazione per il metodo di Jacobi è quindi

$$B_J = P^{-1}N = D^{-1}(E + F).$$

Riassumendo:

Algoritmo 8: Algoritmo di Jacobi

```
1 inizializza x^(0)
2 for k = 0, 1, 2, ... do
3 |   x^(k+1) = D^-1 (E + F) x^(k) + D^-1 b
4 |   if criterio di arresto then
5 |   |   termina algoritmo
6 |   end
7 end
```

Per esplicitare le componenti, scriviamo la formula per $\mathbf{x}^{(k+1)}$ dapprima come $D\mathbf{x}^{(k+1)} = (E + F)\mathbf{x}^{(k)} + \mathbf{b}$ (moltiplicando per D) e poi in forma matriciale:

$$\underbrace{\begin{bmatrix} a_{11} & & & 0 \\ & a_{22} & & \\ & & \ddots & \\ 0 & & & a_{nn} \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ \vdots \\ x_n^{(k+1)} \end{bmatrix}}_{\mathbf{x}^{(k+1)}}$$

$$= \underbrace{\begin{bmatrix} 0 & -a_{1,2} & \dots & -a_{1,n} \\ -a_{2,1} & 0 & & \vdots \\ \vdots & & \ddots & -a_{n-1,n} \\ -a_{n,1} & \dots & -a_{n,n-1} & 0 \end{bmatrix}}_{E+F} \underbrace{\begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_n^{(k)} \end{bmatrix}}_{\mathbf{x}^{(k)}} + \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}}_{\mathbf{b}}$$

Otteniamo dunque:

Algoritmo 9: Algoritmo di Jacobi per componenti

```
1 inizializza x^(0)
2 for k = 1, 2, ... do
3 |   for i = 1 to n do
4 |   |   x_i^(k+1) = (1/a_ii) [ b_i - sum_{j=1, j≠i}^{n} a_ij x_j^(k) ]
5 |   end
6 |   if criterio di arresto then
7 |   |   termina algoritmo
8 |   end
9 end
```

Osservazioni.

- Il metodo di Jacobi può essere applicato solo se D è invertibile, cioè se $a_{ii} \neq 0 \quad \forall i = 1, \dots, n$.
- $P = D$ è una scelta computazionalmente molto economica per il precondizionatore. Infatti, ad ogni passo dobbiamo risolvere il sistema lineare che lo coinvolge. La matrice più semplice possibile per questo è una matrice diagonale perché tutte le equazioni sono *disaccoppiate* e D^{-1} è facile da calcolare, visto che basta invertire i termini sulla diagonale.
- Possiamo quindi immaginare che se il metodo converge, produrrà un'approssimazione poco precisa, perché stiamo approssimando A solo con la sua diagonale.

3.3 Metodo di Gauss-Seidel

Usando la notazione introdotta precedentemente, il metodo di Gauss-Seidel (GS) prende come precondizionatore

$$P = D - E.$$

Quindi le matrici per caratterizzare il metodo sono:

$$A = \underbrace{D - E}_{P} - \underbrace{F}_{N} = P - N, \quad N = F.$$

La matrice di iterazione per il metodo di Gauss-Seidel è quindi

$$B_{GS} = P^{-1}N = (D - E)^{-1}F.$$

Segue il corrispondente pseudocodice:

Algoritmo 10: Algoritmo di Gauss-Seidel

```
1  inizializza x^(0)
2  calcola r^(0) = b − Ax^(0)
3  for k = 1, 2, ... do
4  |  risolvi z^(k) = (D − E)^(−1) r^(k) con sostituzione in avanti
5  |  calcola x^(k+1) = x^(k) + z^(k)
6  |  aggiorna il residuo r^(k+1) = b − Ax^(k+1)
7  |  if criterio di arresto then
8  |  |  termina algoritmo
9  |  end
10 end
```

Per esplicitare le componenti, similmente a prima, scriviamo la relazione come $(D - E)\mathbf{x}^{(k+1)} = F\mathbf{x}^{(k)} + \mathbf{b}$ e poi in forma matriciale:

$$\underbrace{\begin{bmatrix} a_{11} & & & 0 \\ a_{21} & a_{22} & & \\ \vdots & & \ddots & \\ a_{n1} & & & a_{nn} \end{bmatrix}}_{D-E} \underbrace{\begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ \vdots \\ x_n^{(k+1)} \end{bmatrix}}_{\mathbf{x}^{(k+1)}} = \underbrace{\begin{bmatrix} 0 & -a_{12} & \dots & -a_{1n} \\ 0 & 0 & & \vdots \\ \vdots & & \ddots & -a_{n-1\,n} \\ 0 & \dots & 0 & 0 \end{bmatrix}}_{F} \underbrace{\begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ \vdots \\ x_n^{(k)} \end{bmatrix}}_{\mathbf{x}^{(k)}} + \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}}_{\mathbf{b}}.$$

Otteniamo così il seguente algoritmo:

Algoritmo 11: Algoritmo di Gauss-Seidel per componenti

```
1 inizializza x^(0)
2 for k = 1, 2, ... do
3 |   for i = 1 to n do
4 |   |   a_{i1} x_1^(k+1) + ... + a_ii x_i^(k+1) = b_i - [a_{i,i+1} x_{i+1}^(k) + ... + a_{i,n} x_n^(k)]
5 |   end
6 |   if criterio di arresto then
7 |   |   termina algoritmo
8 |   end
9 end
```

Osservazioni.

- La prima riga della matrice nell'algoritmo, ovvero per $i = 1$, presenta la seguente soluzione:

$$a_{11}x_1^{(k+1)} = b_1 - \sum_{j=2}^{n} a_{1j}x_j^{(k)} \quad \Rightarrow \quad x_1^{(k+1)} = \frac{1}{a_{11}}\left[b_1 - \sum_{j=2}^{n} a_{1j}x_j^{(k)}\right].$$

- Per $i = 2$ si ha:

$$a_{21}x_1^{(k+1)} + a_{22}x_2^{(k+1)} = b_2 - \sum_{j=3}^{n} a_{2j}x_j^{(k)}$$

$$\Rightarrow \quad x_2^{(k+1)} = \frac{1}{a_{22}}\left[b_2 - \sum_{j=3}^{n} a_{2j}x_j^{(k)} - a_{21}x_1^{(k+1)}\right].$$

- Per la generica riga i-esima:

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left[b_i - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)} - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)}\right], \quad i = 1, \ldots, n.$$

Osservazione. Anche questo metodo è applicabile solo se tutti gli elementi diagonali $a_{ii}, i = 1, \ldots, n$ sono diversi da 0, altrimenti $D - E$ non è invertibile.

Teorema 3.3 — Convergenza Jacobi e Gauss-Seidel 1. Sia $A \in \mathbb{R}^{n\times n}$ invertibile. Allora:

$$A \text{ è a dominanza diagonale stretta per righe} \quad \Rightarrow \quad \text{J e GS convergono,}$$
$$A \text{ è simmetrica e defefinita positiva} \quad \Rightarrow \quad \text{GS converge.}$$

Teorema 3.4 — Convergenza Jacobi e Gauss-Seidel 2. Sia $A \in \mathbb{R}^{n\times n}$ invertibile e tridiagonale. Allora:

$$\text{J converge} \quad \Leftrightarrow \quad \text{GS converge.}$$

In particolare, se convergono, allora

$$\rho(B_{GS}) = [\rho(B_J)]^2,$$

ovvero il metodo di Gauss-Seidel converge il doppio più velocemente di quello di Jacobi.

3.4 Metodi di rilassamento

Ricordiamo che l'espressione generica di un metodo iterativo è data dalla (3.7)

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + P^{-1}\mathbf{r}^{(k)} \quad k = 0, 1, \ldots$$

dove il residuo $\mathbf{r}^{(k)}$ è dato da $\mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}$.

È possibile generalizzare i metodi di Jacobi e Gauss-Seidel ottenendo i cosiddetti **metodi di rilassamento**.

3.4.1 Metodo Jacobi rilassato (JOR)

Il metodo Jacobi rilassato, o di rilassamento simultaneo (Jacobi Over Relaxation, JOR), è una variante minimale del metodo di Jacobi, il quale è poco costoso, ma anche eccessivamente lento a convergere. Introduciamo un parametro ω che avrà l'obiettivo di accelerarne la convergenza. Definiamo la matrice precondizionante P come:

$$P = \frac{1}{\omega}D, \qquad \text{con } 0 < \omega < 1.$$

La matrice di iterazione associata al metodo JOR è data da:

$$B_{JOR} = \omega B_J + (1-\omega)I = I - \omega D^{-1}A,$$

dove B_J è la matrice di iterazione associata al metodo di Jacobi $B_J = D^{-1}(E+F)$. Questo metodo è una generalizzazione del metodo di Jacobi: se $\omega = 1$, ritroviamo l'algoritmo originale.

Il metodo JOR, scritto per componenti, risulta quindi essere:

Algoritmo 12: Metodo JOR per componenti

```
1 inizializza x^(0)
2 for k = 0, 1, 2, ... do
3 |   for i = 1 to n do
4 |   |   x_i^(k+1) = (ω/a_ii) [ b_i − Σ_{j=1, j≠i}^{n} a_ij x_j^(k) ] + (1 − ω) x_i^(k)
5 |   end
6 |   if criterio di arresto then
7 |   |   termina algoritmo
8 |   end
9 end
```

Nella forma (3.7) il metodo JOR risulta essere

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \omega D^{-1}\mathbf{r}^{(k)} \quad k = 0, 1, 2, \ldots$$

3.4.2 Metodo Gauss-Seidel rilassato (SOR)

Il metodo Gauss-Seidel rilassato, o di rilassamento successivo (Successive Over Relaxation, SOR), è ottenuto con un procedimento simile allo JOR, stavolta a partire dal metodo di Gauss-Seidel. Possiamo modificare la scelta di P introducendo un parametro ω di rilassamento:

$$P = \frac{1}{\omega}D - E, \qquad \text{con } 0 < \omega < 1.$$

La matrice di iterazione associata al metodo SOR è data da:

$$B_{SOR} = \left(I - \omega D^{-1}E\right)^{-1}\left[(1-\omega)I + \omega D^{-1}F\right].$$

Ancora, se $\omega = 1$ si ritrova il metodo di Gauss-Seidel.

Nella forma (3.7) il metodo SOR risulta essere:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \left(\frac{1}{\omega}D - E\right)^{-1}\mathbf{r}^{(k)}, \quad k = 0, 1, 2, \ldots$$

Moltiplicando entrambi i membri per $\omega D^{-1}\left(\frac{1}{\omega}D - E\right)$, ricordando che $A = D - (E + F)$, che $\mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}$ ed eseguendo i calcoli:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \left(\frac{1}{\omega}D - E\right)^{-1}\mathbf{r}^{(k)}$$

$$
\begin{aligned}
\omega D^{-1}\left(\frac{1}{\omega}D - E\right)\mathbf{x}^{(k+1)} &= \omega D^{-1}\left(\frac{1}{\omega}D - E\right)\mathbf{x}^{(k)} + \omega D^{-1}\mathbf{r}^{(k)} \\
\left(I - \omega D^{-1}E\right)\mathbf{x}^{(k+1)} &= \left(I - \omega D^{-1}E\right)\mathbf{x}^{(k)} + \omega D^{-1}\mathbf{r}^{(k)} \\
&= \left(I - \omega D^{-1}(D - A - F)\right)\mathbf{x}^{(k)} + \omega D^{-1}\mathbf{r}^{(k)} \\
&= \left(I - \left(\omega D^{-1}D - \omega D^{-1}A - \omega D^{-1}F\right)\right)\mathbf{x}^{(k)} + \omega D^{-1}\mathbf{r}^{(k)} \\
&= \left(I - \omega I + \omega D^{-1}A + \omega D^{-1}F\right)\mathbf{x}^{(k)} + \omega D^{-1}\mathbf{r}^{(k)} \\
&= \left((1-\omega)I + \omega D^{-1}F\right)\mathbf{x}^{(k)} + \omega D^{-1}A\mathbf{x}^{(k)} \\
&\quad + \omega D^{-1}\left(\mathbf{b} - A\mathbf{x}^{(k)}\right) \\
&= \left((1-\omega)I + \omega D^{-1}F\right)\mathbf{x}^{(k)} + \cancel{\omega D^{-1}A\mathbf{x}^{(k)}} \\
&\quad + \omega D^{-1}\mathbf{b} \cancel{- \omega D^{-1}A\mathbf{x}^{(k)}} \\
&= \left((1-\omega)I + \omega D^{-1}F\right)\mathbf{x}^{(k)} + \omega D^{-1}\mathbf{b}
\end{aligned}
$$

cioè:

$$
\left(I - \omega D^{-1}E\right)\mathbf{x}^{(k+1)} = \left[(1-\omega)I + \omega D^{-1}F\right]\mathbf{x}^{(k)} + \omega D^{-1}\mathbf{b}, \quad k = 0, 1, 2\ldots
$$

In pseudocodice, abbiamo:

Algoritmo 13: Metodo SOR per componenti

```
inizializza x^(0)
for k = 0, 1, 2, ... do
    for i = 1 to n do
        x_i^(k+1) = (ω / a_ii) [ b_i − Σ_{j=1}^{i−1} a_ij x_j^(k+1) − Σ_{j=i+1}^{n} a_ij x_j^(k) ] + (1 − ω) x_i^(k)
    end
    if criterio di arresto then
        termina algoritmo
    end
end
```

3.4.3 Convergenza dei metodi di rilassamento

Teorema 3.5 — Convergenza metodi JOR. Sia A una matrice SDP. Allora

$$
\text{JOR converge} \quad \Leftrightarrow \quad 0 < \omega < \frac{2}{\rho\left(D^{-1}A\right)}.
$$

Teorema 3.6 — Convergenza metodo JOR. Se il metodo di Jacobi converge, allora anche il metodo JOR converge purché $0 < \omega \leq 1$.

Teorema 3.7 — Convergenza metodo SOR. Sia A una matrice SDP. Allora:

$$\text{SOR converge} \Leftrightarrow 0 < \omega < 2.$$

Se A è anche tridiagonale il parametro ottimale[a] ω_{opt} vale:

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho^2(B_J)}}.$$

[a] Il parametro è "ottimale" nel senso che porta nel minor numero di iterazioni possibile a convergenza, ovvero a una soluzione con un errore inferiore a una tolleranza arbitraria.

Teorema 3.8 — Convergenza metodo SOR. Sia A una matrice a predominanza diagonale stretta per righe. Allora:

$$0 < \omega \leq 1 \quad \Rightarrow \quad \text{SOR converge.}$$

3.5 Riassunto matrici di iterazioni

Jacobi	$B_J = D^{-1}(E + F)$
Gauss-Seidel	$B_{GS} = (D - E)^{-1}F$
Jacobi rilassato (JOR)	$B_{JOR} = \omega B_J + (1 - \omega)I = I - \omega D^{-1}A$
Gauss-Seidel rilassato (SOR)	$B_{SOR} = \left(I - \omega D^{-1}E\right)^{-1}\left[(1 - \omega)I + \omega D^{-1}F\right]$

3.6 Metodo di Richardson

Riprendiamo la forma generale di un metodo iterativo:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + P^{-1}\mathbf{r}^{(k)}.$$

Nel metodo di Richardson introduciamo il parametro di accelerazione α, la cui aggiunta non altera la consistenza del metodo. A seconda che il parametro per cui moltiplichiamo sia costante o dipenda dall'iterata k del metodo, otteniamo due varianti del metodo:

- *Metodo di Richardson stazionario*:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha P^{-1}\mathbf{r}^{(k)} \quad \text{ovvero } \alpha_k = \alpha \ \ \forall k.$$

- *Metodo di Richardson dinamico*:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k P^{-1}\mathbf{r}^{(k)}.$$

Osservazioni.

- Jacobi, Gauss-Seidel, JOR, SOR sono tutti casi particolari di metodi di Richardson con $\alpha = 1\,\forall k$.
- Se $P = I$ otteniamo il metodo di Richardson *non precondizionato,* altrimenti abbiamo il metodo di Richardson *precondizionato.*
- La matrice di iterazione per il metodo di Richardson dinamico precondizionato è:

$$B_{\alpha_k} = I - \alpha_k P^{-1} A.$$

Segue lo pseudocodice per il metodo di Richardson nella sua forma più generica:

Algoritmo 14: Algortimo di Richardson dinamico precodizionato

```
1  inizializza x^(0)
2  calcola r^(0) = b − Ax^(0)
3  for k = 0, 1, 2, ... do
4  |  risolvi Pz^(k) = r^(k)
5  |  calcola il parametro di accelerazione α_k
6  |  aggiorna x^(k+1) = x^(k) + α_k z^(k)
7  |  aggiorna r^(k+1) = b − Ax^(k+1)
8  |  if criterio di arresto then
9  |  |  termina algoritmo
10 |  end
11 end
```

Osservazione. Il prodotto matrice-vettore dell'aggiornamento del residuo rappresenta il grosso del costo computazionale, ovvero n^2. Il residuo può essere calcolato in modo del tutto equivalente nel seguente modo, che dal punto di vista computazionale è più efficiente

$$\begin{aligned}
\mathbf{r}^{(k+1)} &= \mathbf{b} - A\left[\mathbf{x}^{(k)} + \alpha_k \mathbf{z}^{(k)}\right] \\
\mathbf{r}^{(k+1)} &= \underbrace{\mathbf{b} - A\mathbf{x}^{(k)}}_{\mathbf{r}^{(k)}} - A\alpha_k \mathbf{z}^{(k)} \\
\mathbf{r}^{(k+1)} &= \mathbf{r}^{(k)} - \alpha_k A\mathbf{z}^{(k)}.
\end{aligned}$$

Esso sfrutta il fatto che il calcolo di α_k contiene già internamente il prodotto matrice-vettore $A\mathbf{z}^{(k)}$.

Teorema 3.9 — Convergenza del metodo di Richardson stazionario 1. Sia P una matrice invertibile. Allora:

$$\text{Richardson stazionario converge} \quad \Leftrightarrow \quad \frac{2\Re(\lambda_i)}{\alpha|\lambda_i|^2} > 1, \quad \forall i = 1, \ldots, n,$$

dove $\Re(\lambda_i)$ è la parte reale dell'i-esimo autovalore λ_i di $P^{-1}A$.

Teorema 3.10 — Convergenza del metodo di Richardson stazionario 2. Sia P una matrice invertibile e $P^{-1}A$ con autovalori reali e positivi, ordinati come $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n > 0$. Allora

$$\text{Richardson stazionario converge} \quad \Leftrightarrow \quad 0 < \alpha < \frac{2}{\lambda_1}.$$

Inoltre la scelta ottimale di α_{opt} è data da

$$\alpha_{opt} = \frac{2}{\lambda_1 + \lambda_n}.$$

Questo metodo è difficilmente utilizzabile nella pratica perché richiedere il calcolo degli autovalori, operazione molto costosa.

3.7 Metodo del gradiente (Richardson dinamico)

Consideriamo dapprima il caso non precondizionato, ovvero con $P = I$, del metodo di Richardson:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{r}^{(k)}.$$

Se A è SDP, sappiamo scegliere α_k nel metodo di Richardson non precondizionato dinamico in maniera ottimale. Osserviamo inoltre che in tal caso, risolvere il sistema è equivalente a minimizzare la seguente forma quadratica, detta **energia del sistema**:

$$\Phi : \mathbb{R}^n \to \mathbb{R}, \quad \Phi(\mathbf{y}) = \frac{1}{2}\, \mathbf{y}^T A\mathbf{y} - \mathbf{y}^T\mathbf{b}.$$

Ovvero:

$$\mathbf{x} \text{ soluzione di } A\mathbf{x} = \mathbf{b} \quad \Leftrightarrow \quad \Phi(\mathbf{x}) = \min_{\mathbf{y}\in\mathbb{R}^n} \Phi(\mathbf{y}).$$

Dimostrazione.

($\Leftarrow$) Per minimizzare un funzionale, ne annulliamo il gradiente:

$$\nabla\Phi(\mathbf{y}) = \nabla\left[\frac{1}{2}\, \mathbf{y}^T A\mathbf{y} - \mathbf{y}^T\mathbf{b}\right]$$

$$
\begin{aligned}
&= \frac{1}{2}A\mathbf{y} + \frac{1}{2}\mathbf{y}^T A - \mathbf{b} && \text{ma } \mathbf{y}^T A = A^T\mathbf{y} = A\mathbf{y} \\
&= \frac{1}{2}A\mathbf{y} + \frac{1}{2}A\mathbf{y} - \mathbf{b} \\
&= A\mathbf{y} - \mathbf{b}.
\end{aligned}
$$

Quindi se troviamo $\mathbf{x}$ tale che $\nabla\Phi(\mathbf{x}) = 0$, allora $A\mathbf{x} - \mathbf{b} = \mathbf{0}$, ovvero $\mathbf{x} \in \mathbb{R}^n$ risolve $A\mathbf{x} = \mathbf{b}$.

($\Rightarrow$) Supponiamo che $\mathbf{x} \in \mathbb{R}^n$ sia soluzione di $A\mathbf{x} = \mathbf{b}$. Allora:

$$
\begin{aligned}
\Phi(\mathbf{y}) &= \Phi(\mathbf{x} + (\mathbf{y} - \mathbf{x})) \\
&= \frac{1}{2}[\mathbf{x} + (\mathbf{y} - \mathbf{x})]^T A[\mathbf{x} + (\mathbf{y} - \mathbf{x})] - [\mathbf{x} + (\mathbf{y} - \mathbf{x})]^T\mathbf{b} && \text{(per definizione)} \\
&= \frac{1}{2}\mathbf{x}^T A\mathbf{x} + \frac{1}{2}(\mathbf{y} - \mathbf{x})^T A\mathbf{x} + \frac{1}{2}\mathbf{x}^T A(\mathbf{y} - \mathbf{x}) + \\
&\quad + \frac{1}{2}(\mathbf{y} - \mathbf{x})^T A(\mathbf{y} - \mathbf{x}) - \mathbf{x}^T\mathbf{b} - (\mathbf{y} - \mathbf{x})^T\mathbf{b} && \text{(riordinando)} \\
&= \underbrace{\frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{x}^T\mathbf{b}}_{\Phi(\mathbf{x})} + \underbrace{\frac{1}{2}(\mathbf{y} - \mathbf{x})^T A\mathbf{x} + \frac{1}{2}\mathbf{x}^T A(\mathbf{y} - \mathbf{x})}_{\text{sono uguali per simmetria di } A} + \\
&\quad + \frac{1}{2}(\mathbf{y} - \mathbf{x})^T A(\mathbf{y} - \mathbf{x}) - (\mathbf{y} - \mathbf{x})^T\mathbf{b} \\
&= \Phi(\mathbf{x}) + (\mathbf{y} - \mathbf{x})^T A\mathbf{x} + \frac{1}{2}(\mathbf{y} - \mathbf{x})^T A(\mathbf{y} - \mathbf{x}) - (\mathbf{y} - \mathbf{x})^T\mathbf{b} \\
&= \Phi(\mathbf{x}) + (\mathbf{y} - \mathbf{x})^T \underbrace{[A\mathbf{x} - \mathbf{b}]}_{=0} + \frac{1}{2}(\mathbf{y} - \mathbf{x})^T A(\mathbf{y} - \mathbf{x}) && \text{(per ipotesi)} \\
&= \Phi(\mathbf{x}) + \underbrace{\frac{1}{2}(\mathbf{y} - \mathbf{x})^T A(\mathbf{y} - \mathbf{x})}_{>0 \text{ perché } A \text{ SDP}}.
\end{aligned}
$$

Quindi $\Phi(\mathbf{y}) > \Phi(\mathbf{x})\ \forall \mathbf{x} \neq \mathbf{y}, \mathbf{y} \in \mathbb{R}^n$ e dunque $\mathbf{x}$ è il punto di minimo di $\Phi(\cdot)$. ■

Una volta stabilito che la risoluzione del sistema è equivalente alla minimizzazione dell'energia Φ, valutiamola nell'iterata $k+1$-esima:

$$
\begin{aligned}
\Phi\left(\mathbf{x}^{(k+1)}\right) &= \Phi\left(\mathbf{x}^{(k)} + \alpha_k\mathbf{r}^{(k)}\right) \\
&= \frac{1}{2}\left(\mathbf{x}^{(k)} + \alpha_k\mathbf{r}^{(k)}\right)^T A\left(\mathbf{x}^{(k)} + \alpha_k\mathbf{r}^{(k)}\right) - \left(\mathbf{x}^{(k)} + \alpha_k\mathbf{r}^{(k)}\right)^T\mathbf{b}.
\end{aligned}
$$

Il valore ottimale di α_k è quello che all'iterata k minimizza $\Phi\left(\mathbf{x}^{(k+1)}\right)$. Quindi

scegliamo α_k tale che[3]:

$$\begin{aligned}
0 &= \frac{\partial \Phi\left(\mathbf{x}^{(k+1)}\right)}{\partial \alpha_k} \\
&= \frac{\partial}{\partial \alpha}\left[\frac{1}{2}(\mathbf{x}+\alpha\mathbf{r})^T A(\mathbf{x}+\alpha\mathbf{r}) - (\mathbf{x}+\alpha\mathbf{r})^T\mathbf{b}\right] \\
&= \frac{\partial}{\partial \alpha}\left[\frac{1}{2}\left(\mathbf{x}^T A\mathbf{x} + \mathbf{x}^T A\alpha\mathbf{r} + \alpha\mathbf{r}^T A\mathbf{x} + \alpha\mathbf{r}^T A\alpha\mathbf{r}\right) - (\mathbf{x}+\alpha\mathbf{r})^T\mathbf{b}\right] \\
&= \frac{\partial}{\partial \alpha}\left[\frac{1}{2}\left(\mathbf{x}^T A\mathbf{x} + \mathbf{x}^T A\alpha\mathbf{r} + \alpha\mathbf{r}^T A\mathbf{x} + \alpha\mathbf{r}^T A\alpha\mathbf{r}\right) - \mathbf{x}^T\mathbf{b} - \alpha\mathbf{r}^T\mathbf{b}\right] \\
&= \frac{1}{2}\left(\mathbf{x}^T A\mathbf{r} + \mathbf{r}^T A\mathbf{x} + 2\alpha\mathbf{r}^T A\mathbf{r}\right) - \mathbf{r}^T\mathbf{b} \\
&= \frac{1}{2}\mathbf{x}^T A\mathbf{r} + \frac{1}{2}\mathbf{r}^T A\mathbf{x} + \alpha\mathbf{r}^T A\mathbf{r} - \mathbf{r}^T\mathbf{b}
\end{aligned}$$

$$\begin{aligned}
\alpha &= \frac{-\frac{1}{2}\mathbf{x}^T A\mathbf{r} - \frac{1}{2}\mathbf{r}^T A\mathbf{x} + \mathbf{r}^T\mathbf{b}}{\mathbf{r}^T A\mathbf{r}} = \frac{-\frac{1}{2}\mathbf{x}^T A\mathbf{r} - \frac{1}{2}\mathbf{r}^T A\mathbf{x} + \mathbf{r}^T\mathbf{r} + \mathbf{r}^T A\mathbf{x}}{\mathbf{r}^T A\mathbf{r}} \\
&= \frac{\cancel{-\frac{1}{2}\mathbf{x}^T A\mathbf{r}} + \cancel{\frac{1}{2}\mathbf{r}^T A\mathbf{x}} + \mathbf{r}^T\mathbf{r}}{\mathbf{r}^T A\mathbf{r}} = \frac{\mathbf{r}^T\mathbf{r}}{\mathbf{r}^T A\mathbf{r}}.
\end{aligned}$$

L'ultimo passaggio vale per simmetria di A, quindi:

$$\alpha_k = \frac{\left[\mathbf{r}^{(k)}\right]^T \mathbf{r}^{(k)}}{\left[\mathbf{r}^{(k)}\right]^T A\ \mathbf{r}^{(k)}} \tag{3.8}$$

Questo conferma che il calcolo di α_k contiene dentro di sé la risoluzione del sistema $A\mathbf{r}^{(k)}$, come accennato in precedenza.

In accordo con quanto visto in precedenza è possibile aggiungere un precondizionatore per migliorare le proprietà di convergenza. A titolo di esempio consideriamo il sistema lineare $A\mathbf{x} = \mathbf{b}$:

$$A = \begin{bmatrix} 4.24 & -4.32 \\ -4.32 & 6.76 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 4 \\ 8 \end{bmatrix}.$$

In questo caso, essendo la dimensione $n = 2$ l'energia del sistema (3.7) si può rappresentare come una superficie. Aggiungendo il precondizionatore:

$$P = \begin{bmatrix} 1.0912 & -0.8587 \\ -0.8587 & 1.5921 \end{bmatrix},$$

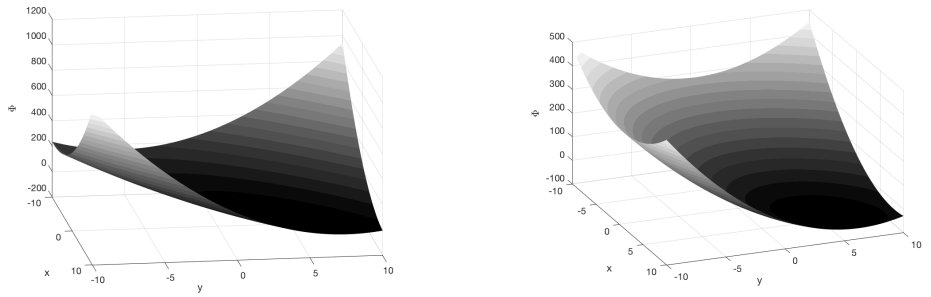
si modifica il condizionamento spettrale, ovvero si *avvicinano* gli autovalori cosicché il loro rapporto sia più vicino a 1 e si migliora il condizionamento della

[3] Trascureremo l'indice (k) per facilità di notazione.

matrice. In figura 3.1 si possono notare le due superfici rappresentanti l'energia del sistema prima e dopo il precondizionamento.

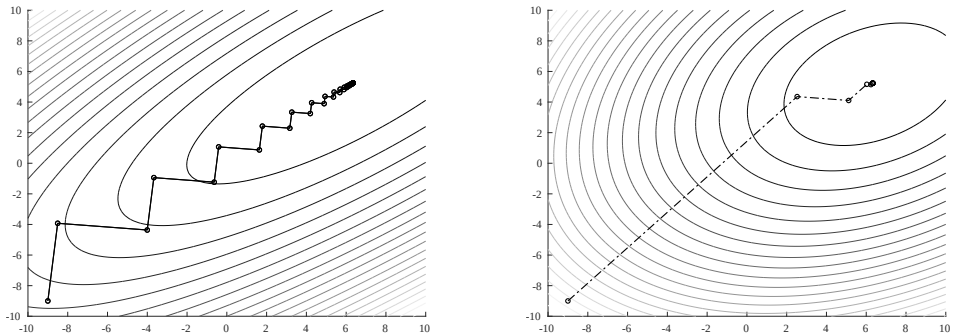
Questo fa si che il numero di iterazioni per convergere sia nettamente minore, nell'esempio in questione si hanno 14 passi contro 88 per raggiungere una soluzione con tollerenza all'errore di 10^{-7}. Ciò si può vedere in figura 3.2.

I passaggi dell'algoritmo sono analoghi, per cui ci limitiamo a riportare direttamente gli pseudocodici delle due varianti.



(a) Prima. (b) Dopo.

Figura 3.1: Effetto del precondizionamento sull'energia del sistema.



(a) Prima. (b) Dopo.

Figura 3.2: Confronto sul numero di iterazioni necessarie con e senza precondizionamento.

Algoritmo 15: Algoritmo del gradiente non precondizionato

inizializza $\mathbf{x}^{(0)}$
calcola $\mathbf{r}^{(0)}$
for $k = 0, 1, 2, \ldots$ **do**
  calcola il parametro ottimale di accelerazione $\alpha_k = \frac{\left[\mathbf{r}^{(k)}\right]^T \mathbf{r}^{(k)}}{\left[\mathbf{r}^{(k)}\right]^T A\ \mathbf{r}^{(k)}}$
  aggiorna $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{r}^{(k)}$
  aggiorna $\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha_k A\ \mathbf{r}^{(k)}$
  if *criterio di arresto* **then**
    termina algoritmo
  end
end

Algoritmo 16: Algoritmo del gradiente precondizionato, P SDP

inizializza $\mathbf{x}^{(0)}$
calcola $\mathbf{r}^{(0)}$
for $k = 0, 1, 2, \ldots$ **do**
  calcola del residuo precondizionato $P\mathbf{z}^{(k)} = \mathbf{r}^{(k)}$
  calcola il parametro ottimale di accelerazione $\alpha_k = \frac{\left[\mathbf{z}^{(k)}\right]^T \mathbf{r}^{(k)}}{\left[\mathbf{z}^{(k)}\right]^T A\ \mathbf{z}^{(k)}}$
  aggiorna $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{z}^{(k)}$
  aggiorna $\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha_k A\ \mathbf{z}^{(k)}$
  if *criterio di arresto* **then**
    termina algoritmo
  end
end

Teorema 3.11 — Convergenza metodo del gradiente. Siano A e P due matrici SDP. Allora il metodo del gradiente (con o senza precondizionatore) converge per ogni scelta di $\mathbf{x}^{(0)}$, e inoltre la successione delle iterate convergenti è monotona:

$$\left\|\mathbf{e}^{(k+1)}\right\|_A \leq \left[\frac{K_2\left(P^{-1}A\right) - 1}{K_2\left(P^{-1}A\right) + 1}\right] \left\|\mathbf{e}^{(k)}\right\|_A, \quad \forall k = 0, 1, 2, \ldots$$

dove $\|\cdot\|_A$ è la **norma dell'energia** definita come

$$\|\mathbf{w}\|_A = \sqrt{\mathbf{w}^T A \mathbf{w}} \qquad \forall \mathbf{w} \in \mathbb{R}^n.$$

Osservazioni.

- La velocità di convergenza è legata al rapporto

$$\frac{K_2\left(P^{-1}A\right)-1}{K_2\left(P^{-1}A\right)+1}. \tag{3.9}$$

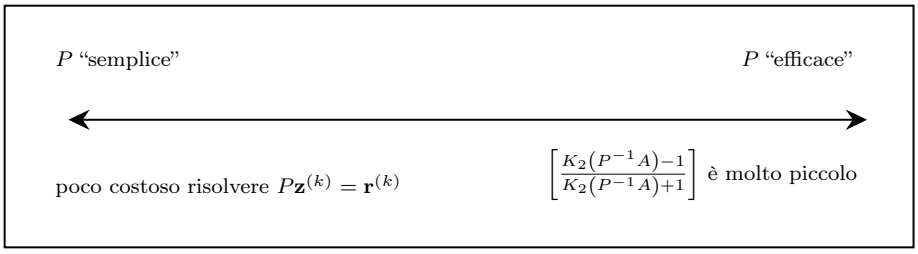
- Esso è sempre minore di 1, e questo garantisce la convergenza del metodo:

$$\left[\frac{K_2\left(P^{-1}A\right)-1}{K_2\left(P^{-1}A\right)+1}\right]<1 \quad \Rightarrow \quad \left\|\mathbf{e}^{(k)}\right\|_A \xrightarrow{k\to\infty} 0.$$

- Se A è mal condizionata si ha

$$K_2 \gg 1 \quad \Rightarrow \quad \frac{K_2-1}{K_2+1}\approx 1 \quad \Rightarrow \quad \text{convergenza lenta.}$$

- Il precondizionamento migliora la proprietà di convergenza. L'idea è di scegliere P tale che (3.9) sia il più piccolo possibile (o analogamente che $K_2\left(P^{-1}A\right)$ sia il più vicino a 1). Ovviamente bisogna tenere conto del fatto che per ogni iterazione dobbiamo risolvere $P\mathbf{z}^{(k)}=\mathbf{r}^{(k)}$. Si ha quindi un *trade-off*[4] in cui bisogna bilanciare due aspetti positivi opposti:



NB. Osserviamo che due vettori del residuo consecutivi nel metodo del gradiente non precondizionato (ma vale anche in quello precondizionato) soddisfano la seguente proprietà:

$$\left(\mathbf{r}^{(k+1)}\right)^T \mathbf{r}^{(k)}=0, \tag{3.10}$$

ovvero i valori del residuo sono *a due a due ortogonali*. Quindi, ad ogni passo k la nuova soluzione $\mathbf{x}^{(k+1)}$ è ottimale rispetto alla direzione di discesa $\mathbf{r}^{(k)}$.

Dimostrazione di (3.10). Ci ricordiamo la relazione di $\mathbf{r}^{(k+1)}$ che compare nell'algoritmo del gradiente e il valore di α_k (3.8):

$$\mathbf{r}^{(k+1)}=\mathbf{r}^{(k)}-\alpha_k A\mathbf{r}^{(k)}, \qquad \alpha_k=\frac{\left[\mathbf{r}^{(k)}\right]^T\mathbf{r}^{(k)}}{\left[\mathbf{r}^{(k)}\right]^T A\mathbf{r}^{(k)}}.$$

[4] "There are no solutions. There are only trade-offs." – Thomas Sowell

Calcoliamo il prodotto direttamente:

$$\begin{aligned}
\left(\mathbf{r}^{(k+1)}\right)^T \mathbf{r}^{(k)} &= \left(\mathbf{r}^{(k)} - \alpha_k A\mathbf{r}^{(k)}\right)^T \mathbf{r}^{(k)} \\
&= \left(\left[\mathbf{r}^{(k)}\right]^T - \alpha_k \left[\mathbf{r}^{(k)}\right]^T A^T\right) \mathbf{r}^{(k)} && \text{(trasposizione)} \\
&= \left[\mathbf{r}^{(k)}\right]^T \mathbf{r}^{(k)} - \alpha_k \left[\mathbf{r}^{(k)}\right]^T A\mathbf{r}^{(k)} && \text{(per simmetria)} \\
&= \left[\mathbf{r}^{(k)}\right]^T \mathbf{r}^{(k)} - \frac{\left[\mathbf{r}^{(k)}\right]^T \mathbf{r}^{(k)}}{\left[\mathbf{r}^{(k)}\right]^T A\mathbf{r}^{(k)}} \left[\mathbf{r}^{(k)}\right]^T A\mathbf{r}^{(k)} && \text{(per ipotesi)} \\
&= \left[\mathbf{r}^{(k)}\right]^T \mathbf{r}^{(k)} - \left[\mathbf{r}^{(k)}\right]^T \mathbf{r}^{(k)} \\
&= 0 \qquad \blacksquare
\end{aligned}$$

Notiamo tuttavia che *non è garantito* che $\mathbf{x}^{(k+1)}$ sia ottimale rispetto a tutti i residui calcolati ai passi precedenti a k, ad esempio tutti i passi $j = 0, \ldots, k-1$:

$$\left[\mathbf{r}^{(k+1)}\right]^T \mathbf{r}^{(k)} = 0, \quad \text{ma} \quad \left[\mathbf{r}^{(k+1)}\right]^T \mathbf{r}^{(j)} \neq 0, \quad j = 0, \ldots, k-1.$$

Infatti, il metodo del gradiente non tiene traccia di tutte le direzioni precedenti, ma solo di quella immadiatamente precedente.

3.8 Metodo del gradiente coniugato

È possibile modificare la direzione di discesa in modo da garantire un qualche criterio di *ottimalità*? Questa è l'idea che sta alla base del metodo del gradiente coniugato (GC o CG)[5]. In particolare, dato un metodo iterativo

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}$$

vogliamo scegliere direzioni di discesa che costituiscano una successione ottimale di vettori $\mathbf{p}^{(k)}$.

Definizione 3.12 — Vettori A-coniugati. Sia A SDP fissata. Due vettori $\mathbf{w}, \mathbf{v} \in \mathbb{R}^n$ si dicono A-coniugati (o A-ortogonali) se

$$\mathbf{w}^T A\ \mathbf{v} = 0 \quad \text{cioè} \quad (A\mathbf{w})^T \mathbf{v} = 0.$$

Osservazioni.

- Simmetria: $0 = \left(\mathbf{w}^T A\ \mathbf{v}\right)^T = \mathbf{v}^T A^T \mathbf{w} = \mathbf{v}^T A\ \mathbf{w}$.
- Annullamento: $\mathbf{v}^T A\mathbf{v} = 0 \ \Rightarrow\ \mathbf{v} = \mathbf{0}$.

[5] che "è la Ferrari dei metodi iterativi".

3.8.1 Scelta della direzione di discesa

Scegliamo la seguente successione di vettori:

$$\begin{cases} \mathbf{p}^{(0)} = \mathbf{r}^{(0)} \\ \mathbf{p}^{(k+1)} = \mathbf{r}^{(k+1)} - \beta_k \mathbf{p}^{(k)} \end{cases} \qquad \beta_k = \frac{\left(A\mathbf{p}^{(k)}\right)^T \mathbf{r}^{(k+1)}}{\left(A\mathbf{p}^{(k)}\right)^T \mathbf{p}^{(k)}} \quad k = 0, 1, 2, \dots \tag{3.11}$$

Proprietà.

Si può dimostrare per induzione che con la scelta (3.11) si ha:

1. $\left[\mathbf{p}^{(j)}\right]^T \mathbf{r}^{(k+1)} = 0 \quad \forall j = 0, 1, \dots, k$

   La soluzione $\mathbf{x}^{(k+1)}$ è ottimale rispetto a *tutte* le direzioni di discesa precedenti $\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \dots, \mathbf{p}^{(k)}$.

2. $\left[A\mathbf{p}^{(j)}\right]^T \mathbf{p}^{(k+1)} = 0 \quad \forall j = 0, 1, \dots, k$

   La direzione di discesa $\mathbf{p}^{(k+1)}$ è A-ortogonale rispetto a *tutte* le direzioni $\mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \mathbf{p}^{(2)}, \dots, \mathbf{p}^{(k)}$. Ad ogni passo, l'algoritmo esplora e minimizza una direzione nuova, mai usata prima, tralasciando lo spazio già precedentemente esplorato.

3. Combinando (1) e (2) al passo n-esimo si ha $\mathbf{r}^{(n)} = 0$, ossia $\mathbf{x}^{(n)}$ *è la soluzione esatta*[6].

Dimostriamo la proprietà (3), lasciando (1) e (2) al lettore come esercizio. Per la proprietà (1), al passo $k = n - 1$ si ha:

$$\underbrace{\mathbf{r}^{(n)}}_{\in \mathbb{R}^n} \perp \mathbf{p}^{(0)}, \mathbf{p}^{(1)}, \dots, \mathbf{p}^{(n-1)}.$$

Non è ancora possibile concludere che $\mathbf{r}^{(n)}$ è nullo in quanto ortogonale ad altri n vettori, perché non sappiamo se gli altri $\mathbf{p}^{(k)}$ sono indipendenti. Ma vedremo che questa proprietà è assicurata dalla (2). Prendiamo la combinazione lineare dei vettori precedenti e poniamola uguale a 0: vogliamo dimostrare che ciò implica necessariamente che tutti i coefficienti sono nulli:

$$a_0\mathbf{p}^{(0)} + a_1\mathbf{p}^{(1)} + \ldots + a_{n-1}\mathbf{p}^{(n-1)} = 0 \xrightarrow{?} a_i = 0 \quad \forall i = 0, \dots, n-1.$$

Moltiplichiamo a destra e sinistra per $\left(A\mathbf{p}^{(0)}\right)^T$:

$$a_0 \underbrace{\left(A\mathbf{p}^{(0)}\right)^T \mathbf{p}^{(0)}}_{>0 \text{ se } \mathbf{p}^{(0)} \neq 0} + a_1 \underbrace{\left(A\mathbf{p}^{(0)}\right)^T \mathbf{p}^{(1)}}_{=0 \text{ per (2)}} + \ldots + a_{n-1} \left(A\mathbf{p}^{(0)}\right)^T \mathbf{p}^{(n-1)} = 0 \;\Rightarrow\; a_0 = 0.$$

Analogamente moltiplicando per $\left(A\mathbf{p}^{(1)}\right)^T$ si ricava $a_1 = 0$ e così via. Infine si ottiene $a_i = 0 \;\; \forall i$, ovvero che tutti i $\mathbf{p}^{(i)}$ sono linearmente indipendenti. ■

[6] Mai notizia più buona fu data.

3.8.2 Scelta del parametro di accelerazione

Ripercorrendo gli stessi passi della sezione (3.7) sulla minimizzazione del funzionale $\Phi(\cdot)$, valutato questa volta in $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}$, troviamo:

$$\alpha_k = \frac{\left[\mathbf{p}^{(k)}\right]^T \mathbf{r}^{(k)}}{\left[\mathbf{p}^{(k)}\right]^T A\ \mathbf{p}^{(k)}}.$$

Riassumiamo il gradiente coniugato in pseudocodice:

Algoritmo 17: Algoritmo del metodo del gradiente coniugato, non precondizionato

```
inizializza x^(0)
calcola r^(0) = b − Ax^(0)
inizializza p^(0) = r^(0)
for k = 0, 1, 2, ... do
    calcola il parametro α_k = ([p^(k)]^T r^(k)) / ([p^(k)]^T A p^(k))
    aggiorna x^(k+1) = x^(k) + α_k p^(k)
    aggiorna r^(k+1) = r^(k) − α_k A p^(k)
    calcola il parametro β_k = ((Ap^(k))^T r^(k+1)) / ((Ap^(k))^T p^(k))
    aggiorna p^(k+1) = r^(k+1) − β_k p^(k)
    if criterio di arresto then
        termina algoritmo
    end
end
```

Teorema 3.13 — Convergenza del metodo GC. Sia A una matrice SDP in aritmetica esatta. Allora il metodo GC converge alla soluzione esatta di $A\mathbf{x} = \mathbf{b}$ in al più n passi. Inoltre, per ogni iterazione $k = 0, \ldots, n$, l'errore $\mathbf{e}^{(k)}$ è ortogonale alla direzione $\mathbf{p}^{(j)} \quad j = 0, \ldots, k-1$ e

$$\left\|\mathbf{e}^{(k)}\right\|_A \leq \left[\frac{2c^k}{1+c^{2k}}\right] \left\|\mathbf{e}^{(0)}\right\|_A \quad \text{con} \quad c = \frac{\sqrt{K_2(A)}-1}{\sqrt{K_2(A)}+1}.$$

Osservazioni.

- Se A è SDP, allora $K_2(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$.

- Come per il metodo del gradiente la convergenza è monotona, e in particolare la sua velocità dipende da $\sqrt{K_2(A)}$.

Scriviamo ora la versione precondizionata del gradiente coniugato:

Algoritmo 18: Algoritmo del metodo del gradiente coniugato, precondizionato

1 inizializza $\mathbf{x}^{(0)}$
2 calcola $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}$
3 risolvi $P\mathbf{z}^{(0)} = \mathbf{r}^{(0)}$
4 inizializza $\mathbf{p}^{(0)} = \mathbf{z}^{(0)}$
5 **for** $k = 0, 1, 2, \ldots$ **do**
6 calcola il parametro $\alpha_k = \frac{\left[\mathbf{p}^{(k)}\right]^T \mathbf{r}^{(k)}}{\left[\mathbf{p}^{(k)}\right]^T A\ \mathbf{p}^{(k)}}$
7 aggiorna $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}^{(k)}$
8 aggiorna $\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha_k A\mathbf{p}^{(k)}$
9 risolvi $P\mathbf{z}^{(k+1)} = \mathbf{r}^{(k+1)}$
10 calcola il parametro $\beta_k = \frac{\left(A\mathbf{p}^{(k)}\right)^T \mathbf{z}^{(k+1)}}{\left(A\mathbf{p}^{(k)}\right)^T \mathbf{p}^{(k)}}$
11 aggiorna $\mathbf{p}^{(k+1)} = \mathbf{z}^{(k+1)} - \beta_k \mathbf{p}^{(k)}$
12 **if** *criterio di arresto* **then**
13 termina algoritmo
14 **end**
15 **end**

Per esso vale lo stesso teorema di convergenza 3.13, con

$$c = \frac{\sqrt{K_2\left(P^{-1}A\right)} - 1}{\sqrt{K_2\left(P^{-1}A\right)} + 1}.$$

3.9 Criteri di arresto

I criteri di arresto non dipendono nello specifico dai metodi che stiamo utilizzando. Fissiamo una tolleranza TOL, ovvero una quantità decisa dall'utente, che la sceglie in base a quanta precisione desidera nel calcolo della soluzione. Più piccola è TOL, più iterazioni l'algoritmo necessita per convergere: dobbiamo trovare un equilibrio, un trade-off, tra quanto vogliamo essere accurati e quante iterazioni siamo disposti ad aspettare.

1. **Criterio sul residuo.** Arrestiamo il ciclo quando

$$\frac{\left\|\mathbf{r}^{(k+1)}\right\|}{\|\mathbf{b}\|} \leq \mathrm{TOL}\,.$$

2. **Criterio sull'incremento.** Arrestiamo il ciclo quando

$$\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\| \leq \text{TOL}\,.$$

3. **Criterio di controllo.** Arrestiamo il ciclo dopo $n_{\max}$ iterazioni.

Osservazioni.

- I criteri (1) e (2) agiscono in funzione della *qualità della soluzione*, mentre (3) è un criterio di emergenza.
- Di norma si sceglie di usare uno tra i primi due criteri, abbinati sempre al terzo. Infatti ci sono matrici che convergono lentissimamente, che hanno bisogno di molte iterazioni per raggiungere un risultato soddisfacente.
- Ogni iterazione di Gauss-Seidel costa n^2 operazioni. Dopo n iterazioni, il costo complessivo diventa n^3, esattamente il costo di una fattorizzazione con metodo diretto. Per tutti i problemi in cui non c'è problema di memoria limitata e quindi possiamo scegliere fra metodi diretti e iterativi, il metodo iterativo è molto più conveniente nella misura in cui andiamo a usare un criterio di arresto in meno di n operazioni. Quindi $n_{\max} \approx n$. Se abbiamo troppo poco spazio, i metodi iterativi sono d'obbligo.
- Il criterio (2) lascia proseguire l'algoritmo fintanto che fra l'iterazione (k) e l'iterazione $(k+1)$ c'è sufficiente progresso. Se $\mathbf{x}^{(k)}$ e $\mathbf{x}^{(k+1)}$ sono molto vicine, significa che l'algoritmo sta stagnando.
- Un criterio si dice **affidabile** se nel momento in cui si verifica la condizione, abbiamo la garanzia che anche l'errore (normalizzato) sia minore della tolleranza moltiplicata per una costante.

3.9.1 Criterio sul residuo

Controlliamo l'affidabilità del criterio sul residuo. L'algoritmo si arresta quando si verifica:

$$\frac{\left\|\mathbf{r}^{(k)}\right\|}{\|\mathbf{b}\|} \leq \text{TOL}\,. \tag{3.12}$$

Dobbiamo verificare se valga la seguente implicazione, come notato nelle osservazioni:

$$\frac{\left\|\mathbf{r}^{(k)}\right\|}{\|\mathbf{b}\|} \leq \text{TOL} \quad \stackrel{?}{\Rightarrow} \quad \frac{\left\|\mathbf{e}^{(k)}\right\|}{\|\mathbf{x}\|} = \frac{\left\|\mathbf{x} - \mathbf{x}^{(k)}\right\|}{\|\mathbf{x}\|} \leq C\ \text{TOL}\,.$$

A tal proposito, ricordiamo il teorema di stabilità dei sistemi lineari 2.16:

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq K_2(A)\frac{\|\mathbf{b} - A\tilde{\mathbf{x}}\|}{\|\mathbf{b}\|}$$

Prendendo $\tilde{\mathbf{x}} = \mathbf{x}^{(k)}$ e ricordando che $\mathbf{b} - A\mathbf{x}^{(k)} = \mathbf{r}^{(k)}$, l'enunciato del teorema diventa:

$$\frac{\left\|\mathbf{x} - \mathbf{x}^{(k)}\right\|}{\|\mathbf{x}\|} \leq K_2(A)\frac{\left\|\mathbf{r}^{(k)}\right\|}{\|\mathbf{b}\|} \leq K_2(A)\ \text{TOL}\,. \tag{3.13}$$

Quindi il criterio d'arresto sul residuo (3.12) è affidabile solo se la matrice è ben condizionata. Nel caso usassimo un precondizionatore, bisogna sostituire il criterio con il seguente:

$$\frac{\left\|P^{-1}\mathbf{r}^{(k)}\right\|}{\left\|P^{-1}\mathbf{r}^{(0)}\right\|} \leq \text{TOL} \quad \text{oppure analogamente} \quad \frac{\left\|\mathbf{z}^{(k)}\right\|}{\left\|\mathbf{z}^{(0)}\right\|} \leq \text{TOL},$$

in tal caso la (3.13) diventa:

$$\frac{\left\|\mathbf{x} - \mathbf{x}^{(k)}\right\|}{\|\mathbf{x}\|} \leq K_2\left(P^{-1}A\right)\ \text{TOL}\,.$$

3.9.2 Criterio sull'incremento

Studiamo ora l'affidabilità del criterio sull'incremento:

$$\frac{\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\|}{\left\|\mathbf{x}^{(k+1)}\right\|} \leq \text{TOL}\,. \tag{3.14}$$

Bisogna verificare la seguente implicazione:

$$\frac{\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\|}{\left\|\mathbf{x}^{(k+1)}\right\|} \leq \text{TOL} \quad \overset{?}{\Rightarrow} \quad \frac{\left\|\mathbf{e}^{(k)}\right\|}{\|\mathbf{x}\|} = \frac{\left\|\mathbf{x} - \mathbf{x}^{(k)}\right\|}{\|\mathbf{x}\|} \leq C\ \text{TOL}\,.$$

A tal proposito, ricordiamo:

$$\begin{aligned}
\mathbf{x}^{(k+1)} &= B\mathbf{x}^{(k)} + \mathbf{g} && \text{(forma generale)}\\
\mathbf{x} &= B\mathbf{x} + \mathbf{g} && \text{(consistenza)}\\
\mathbf{x}^{(k+1)} - \mathbf{x} &= B(\mathbf{x}^{(k)} - \mathbf{x}). && \text{(differenza)}
\end{aligned}$$

L'errore k-esimo si può scrivere come:

$$\begin{aligned}
\left\|\mathbf{e}^{(k)}\right\| = \left\|\mathbf{x}^{(k)} - \mathbf{x}\right\| &= \left\|\mathbf{x}^{(k)} - \mathbf{x}^{(k+1)} + \mathbf{x}^{(k+1)} - \mathbf{x}\right\|\\
&\leq \left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\| + \left\|\mathbf{x}^{(k+1)} - \mathbf{x}\right\| && \text{(dis. triangolare)}\\
&\leq \left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\| + \|B\|\ \left\|\mathbf{x}^{(k)} - \mathbf{x}\right\|. && \text{(per quanto detto)}
\end{aligned}$$

Portiamo a sinistra e dividiamo per $1 - \|B\|$:

$$(1 - \|B\|)\left\|\mathbf{x}^{(k)} - \mathbf{x}\right\| \leq \left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\|$$

$$\underbrace{\left\|\mathbf{x}^{(k)} - \mathbf{x}\right\|}_{\|\mathbf{e}^{(k)}\|} \leq \frac{1}{1 - \|B\|} \left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\|.$$

Infine, ricordando la formula di rappresentazione dell'errore $\mathbf{e}^{(k+1)} = B\mathbf{e}^{(k)}$:

$$\begin{aligned}
\left\|\mathbf{x}^{(k+1)} - \mathbf{x}\right\| &= \left\|\mathbf{e}^{(k+1)}\right\| \\
&= \left\|B\mathbf{e}^{(k)}\right\| && \text{(formula di rappr. dell'errore)} \\
&\leq \|B\| \ \left\|\mathbf{e}^{(k)}\right\| && \text{(proprietà della norma)} \\
&\leq \frac{\|B\|}{1 - \|B\|} \left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\|. && \text{(per quanto appena trovato)}
\end{aligned}$$

Otteniamo infine:

$$\left\|\mathbf{x} - \mathbf{x}^{(k+1)}\right\| \leq \frac{\|B\|}{1 - \|B\|} \ \left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\|. \tag{3.15}$$

Poiché il metodo è convergente si ha $\rho(B) < 1$, quindi $\|B\| < 1$. Il criterio è quindi affidabile. La (3.15) si può anche riscrivere come

$$\left\|\mathbf{x} - \mathbf{x}^{(k+1)}\right\| \leq \left[\frac{1}{1 - \rho(B)}\right] \left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\|.$$

Quindi se $\|B\|$ è molto vicina a 1, cioè se A è mal condizionata, la differenza tra l'errore vero e quello che sappiamo misurare diventa molto grande e quindi dà poche informazioni.

Capitolo 4

Approssimazione di funzioni e dati

L'obiettivo di un'approssimazione funzionale è sostituire una funzione *complicata* con una funzione *semplice* ricercata in una classe prefissata di funzioni, in genere dei polinomi.

Esempio. L'espressione analitica di una funzione f non è nota, ma essa è solo campionata, ovvero il suo valore è noto solamente su un insieme di punti $x_0, \ldots, x_n$, e vogliamo determinare una possibile espressione di f.

x	$f(x)$
x_0	$y_0 = f(x_0)$
x_1	$y_1 = f(x_1)$
$\vdots$	$\vdots$
x_n	$y_n = f(x_n)$

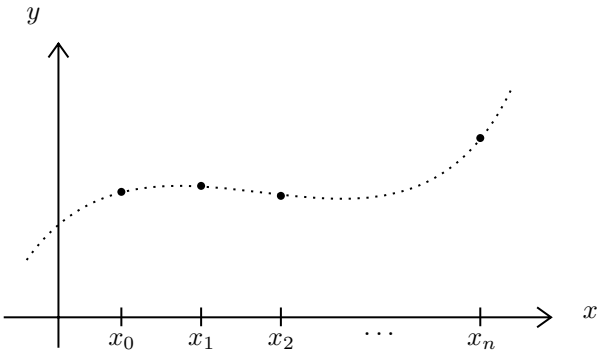


Figura 4.1: Esempio monodimensionale in cui conosciamo il valore della funzione, solo in alcuni punti, ma vogliamo in qualche modo determinarne un'approssimazione. La linea tratteggiata indica una delle tante possibilità di f, infatti passa per tutti i campionamenti.

Esempio. Conosciamo l'espressione analitica di f, ma vogliamo sostituirla con

una funzione più *semplice* per il calcolo numerico oppure, ad esempio, per rendere possibili o più semplici delle operazioni funzionali quali l'integrazione o la derivazione.

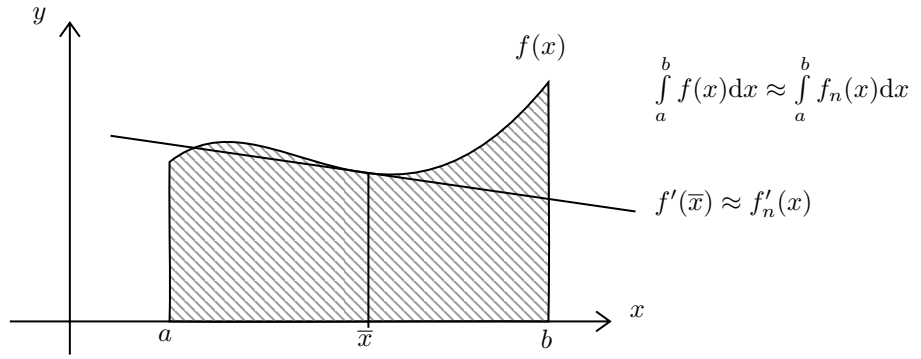


Figura 4.2: Cercheremo di approssimare anche le derivate di una funzione e gli integrali definiti in base ai campionamenti della funzione stessa (ovviamente non si può *campionare la derivata o l'integrale*).

Nel contesto delle equazioni differenziali è fondamentale saper approssimare funzioni e dati.

4.1 Polinomio di interpolazione di Lagrange

Siano assegnate $(n+1)$ coppie, $n \geq 0$, di punti *distinti* nel piano, come in figura 4.1.

Vogliamo costruire un polinomio $\Pi_n(x) \in \mathbb{P}^n$ di grado n tale che

$$\Pi_n(x_i) = y_i \quad \forall i = 0, 1, \ldots, n.$$

Questa procedura è definita **interpolazione**.

Osservazione. Se al posto di avere i soli punti di campionamento $(x_i, y_i), i = 0, \ldots, n$ avessimo una funzione complicata che vogliamo approssimare, allora l'insieme di condizioni da imporre è dato da

$$(x_i, f(x_i)) \quad i = 0, \ldots, n$$

e le condizioni diventano

$$\Pi_n f(x_i) = f(x_i) \quad \forall i = 0, \ldots, n.$$

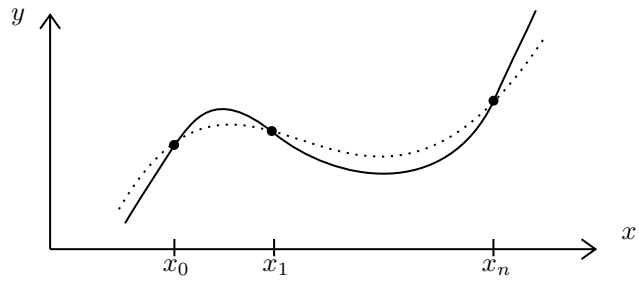


Figura 4.3: Esistono diverse possibilità che rispettano le condizioni di passaggio nei nodi, è importante formalizzare delle strategie e dei criteri per determinare la migliore interpolazione.

Teorema 4.1. Dati $n+1$ punti distinti $x_0, x_1, \ldots, x_n$, detti nodi di interpolazione, e $n+1$ corrispondenti valori $y_0, y_1, \ldots, y_n$, allora esiste un unico polinomio di interpolazione $\Pi_n(x)$ di grado n tale che

$$\Pi_n(x_i) = y_i \quad \forall i = 0, \ldots, n. \tag{4.1}$$

NB. Per giustificare intuitivamente il fatto che servano $n+1$ punti, ricordiamo che esiste una e una sola retta (polinomio di grado 1) passante per due punti, mentre esistono infinite parabole (polinomio di grado 2) passanti per due punti. Pertanto per fissare il nostro oggetto di grado n ci servono $n+1$ punti.

Dimostrazione. La dimostrazione è costruttiva, ovvero contiene il modo operativo per calcolare il polinomio di interpolazione. Dobbiamo mostrare che questo esista e che sia unico.

Esistenza. Forniamo una rappresentazione esplicita del polinomio Π_n. Abbiamo bisogno di costruire una base per lo spazio dei polinomi di grado $n \to \mathbb{P}^n$ definiti su $\mathbb{R}$. Una scelta molto semplice è utilizzare la base dei monomi

$$\mathbb{P}^n = \text{span}\left\{1, x, x^2, \ldots, x^n\right\},$$

cioè scrivere $\Pi_n(x)$ come

$$\Pi_n(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n,$$

con $a_0, a_1, \ldots, a_n \in \mathbb{R}$ da determinare imponendo i vincoli di passaggio (4.1). Costruiamo quindi una nuova base, detta **base di Lagrange**:

$$\mathbb{P}^n = \text{span}\{\mathcal{L}_0(x), \ldots, \mathcal{L}_n(x)\}$$

dove per ogni $i = 0, \ldots, n$ l'i-esimo **polinomio di Lagrange** $\mathcal{L}_i(x)$ è definito in questo modo:

$$\mathcal{L}_i(x) \text{ è un polinomio di grado } n$$
$$\mathcal{L}_i(x) = \begin{cases} 1 & \text{se } x = x_i \\ 0 & \text{se } x = x_j,\ j \neq i. \end{cases}$$

L'espressione analitica di ogni $\mathcal{L}_i(x), i = 0, \ldots, n$ è:

$$\mathcal{L}_i(x) = \prod_{j=0, j\neq i}^{n} \frac{(x - x_j)}{(x_i - x_j)} \quad i = 0, \ldots, n.$$

In figura 4.4 si può vedere un esempio di un set di polinomi di Lagrange costruiti su un specifico insieme di nodi.

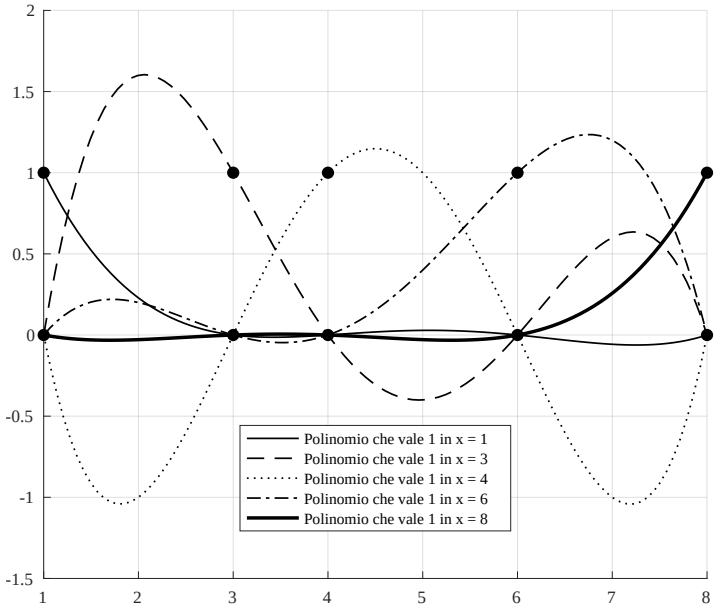


Figura 4.4: Polinomi di Lagrange nei nodi $x_i = 1, 3, 4, 6, 8$.

Si può dimostrare che questi polinomi sono indipendenti e che quindi costituiscono una base per i polinomi $\mathbb{P}^n$. Con questa base è immediato costruire il polinomio di interpolazione $\Pi_n(x)$, infatti:

$$\Pi_n(x) = y_0\mathcal{L}_0(x) + y_1\mathcal{L}_1(x) + \ldots + y_n\mathcal{L}_n(x).$$

Ovvero in forma compatta otteniamo:

$$\Pi_n(x) = \sum_{i=0}^{n} y_i\mathcal{L}_i(x).$$

Questo ne dimostra l'esistenza.

Unicità. Per assurdo supponiamo che esistano due polinomi distinti $\Pi_n(x)$ e $\tilde{\Pi}_n(x)$ tali che rispettano le proprietà richieste, ovvero:

$$\Pi_n(x_i) = y_i \quad \forall i = 0, \dots, n \quad \text{e} \quad \tilde{\Pi}_n(x_i) = y_i \quad \forall i = 0, \dots, n.$$

Definiamo ora $\mathcal{Y}_n(x) \coloneqq \Pi_n(x) - \tilde{\Pi}_n(x)$, che chiaramente è un polinomio di grado n ed è nullo su tutti i nodi interpolazione. Abbiamo $\forall i = 0, \dots, n$:

$$\mathcal{Y}_n(x_i) = \Pi_n(x_i) - \tilde{\Pi}_n(x_i) = y_i - y_i = 0.$$

Quindi $\mathcal{Y}_n(x)$ è un polinomio di grado n che ha $n+1$ zeri. Dunque per il teorema fondamentale dell'algebra:

$$\mathcal{Y}_n(x) \equiv 0 \quad \Rightarrow \quad \Pi_n(x) = \tilde{\Pi}_n(x)$$

ma questo è assurdo in quanto si è supposto fossero distinti, pertanto abbiamo dimostrato l'unicità. ■

Dal teorema abbiamo anche imparato a costruire il **polinomio di interpolazione di Lagrange**:

$$\Pi_n(x) = \sum_{i=0}^{n} y_i \mathcal{L}_i(x).$$

Se al posto di avere le coppie di punti $\{(x_i, y_i)\}_{i=0}^n$ avessimo una funzione f da interpolare, potremmo scrivere analogamente

$$\Pi_n f(x) = \sum_{i=0}^{n} f(x_i) \mathcal{L}_i(x).$$

Osservazioni.

- Le espressioni dei polinomi della base di Lagrange $\mathcal{L}_0(x), \dots, \mathcal{L}_n(x)$ dipendono *solo* dai nodi di interpolazione $x_i, i = 0, \dots, n$.
- Si può anche dimostrare che i polinomi di Lagrange soddisfano

  $$\mathcal{L}_i(x) = \frac{w_{n+1}(x)}{(x - x_i) w'_{n+1}(x_i)} \qquad \forall i = 0, \dots, n$$

  dove $w_{n+1}(x)$ è detto **polinomio nodale** ed è definito come

  $$w_{n+1}(x) = \prod_{i=0}^{n} (x - x_i).$$

  Ne consegue che

  $$\Pi_n(x) = \sum_{i=0}^{n} y_i \frac{w_{n+1}(x)}{(x - x_i) w'_{n+1}(x_i)}.$$

Teorema 4.2 — Errore di interpolazione. Siano $x_0, x_1, \ldots, x_n$ un set di $n+1$ punti distinti in $I \subseteq \mathbb{R}$ e sia $f \in C^{n+1}(I)$. Per ogni $x \in I$ l'errore $E(x) = f(x) - \Pi_n f(x)$ è dato da

$$E(x) = \frac{\omega_{n+1}(x)}{(n+1)!} f^{(n+1)}(\xi)$$

dove $\xi(x) \in I$ e $\omega_{n+1}(x)$ è il polinomio nodale associato ai nodi $x_0, x_1, \ldots, x_n$, ossia $\omega_{n+1}(x) = \prod_{i=0}^{n}(x - x_i)$.

Per capire l'utilizzo del teorema, diamo la seguente definizione.

Definizione 4.3 — Norma infinito. Data f una funzione definita su un insieme A, denotiamo la sua norma infinito con[a]

$$\|f\|_\infty = \max_{x \in A} |f(x)| .$$

[a] La definizione esatta è in realtà più complessa, ma coinvolge concetti come misurabilità e insiemi di misura nulla, che non sono oggetto di questo libro e non necessari per la comprensione della materia.

L'utilizzo della norma infinito è fondamentale in quanto la formula dell'errore è valida quando la funzione viene valutata in un punto ξ, ma questo punto non è noto, e si sa solo che esso appartiene all'intervallo I. Possiamo naturalmente maggiorare il valore che la $f^{(n+1)}$ assume in ξ con il massimo valore assunto nell'intervallo. Questo avviene proprio grazie alla norma infinito, che essendo un massimo su x è una costante rispetto a x:

$$|E(x)| \leq \frac{1}{(n+1)!} \, |\omega_{n+1}(x)| \, \underbrace{\left\| f^{(n+1)}(x) \right\|_\infty}_{\leq C} \leq \frac{C}{(n+1)!} \, |\omega_{n+1}(x)|.$$

Dimostrazione. Fissando $x \in I$, vogliamo stimare $E(x) = f(x) - \Pi_n f(x)$. Abbiamo due casi:

- Se x coincide con uno dei nodi di interpolazione, ossia $x = x_i$ per qualche i, allora il risultato è banale perché $E(x) = 0$ dal momento che

$$E(x) = \frac{\omega_{n+1}(x)}{(n+1)!} f^{(n+1)}(\xi) = \frac{\prod_{i=0}^{n} \overbrace{(x - x_i)}^{=0}}{(n+1)!} f^{(n+1)}(\xi).$$

- Se x *non* coincide con uno dei nodi di interpolazione, definiamo la seguente funzione:

$$G : I \to \mathbb{R} \quad G(t) = E_n(t) - \omega_{n+1}(t) \frac{E_n(x)}{\omega_{n+1}(x)}.$$

Osserviamo che:

- $G(t) \in C^{n+1}(I)$ perché $f \in C^{n+1}(I)$ e $\omega_{n+1}(t)$ è un polinomio, quindi anch'esso è infinitamente derivabile.
- $G(t)$ ha almeno $n+2$ zeri nell'intervallo I; infatti, $\forall i = 0, \dots, n$:

$$G(x_i) = \underbrace{E_n(x_i)}_{=0} - \underbrace{\omega_{n+1}(x_i)}_{=0} \frac{E_n(x)}{\omega_{n+1}(x)} \quad \rightarrow \quad n+1 \text{ zeri}$$

$$G(x) = E_n(x) - \omega_{n+1}(x)\frac{E_n(x)}{\omega_{n+1}(x)} = 0 \quad \rightarrow \quad 1 \text{ zero.}$$

Quindi $G(t)$ è una funzione di classe C^{n+1} in I con almeno $n+2$ zeri distinti. Quindi per il teorema del valor medio possiamo concludere che $G'(t)$ ha almeno $n+1$ zeri distinti e, con lo stesso ragionamento, che $G^{(n+1)}(t)$ ammette uno zero che chiamiamo $\xi = \xi(x)$, cioè $G^{(n+1)}(\xi) = 0$.

Adesso calcoliamo l'espressione di $G^{(n+1)}(t)$. Per linearità si ha:

$$\begin{aligned} G^{(n+1)}(t) &= E_n^{(n+1)}(t) - \omega_{n+1}^{(n+1)}(t)\frac{E_n(x)}{\omega_{n+1}(x)} \\ &= f^{(n+1)}(t) - \underbrace{(\Pi_n f)^{(n+1)}(t)}_{=0} - \underbrace{\omega_{n+1}^{(n+1)}(t)}_{=(n+1)!} \frac{E_n(x)}{\omega_{n+1}(x)} \\ &= f^{(n+1)}(t) - (n+1)!\frac{E_n(x)}{\omega_{n+1}(x)}. \end{aligned}$$

Valutiamo infine il polinomio in ξ:

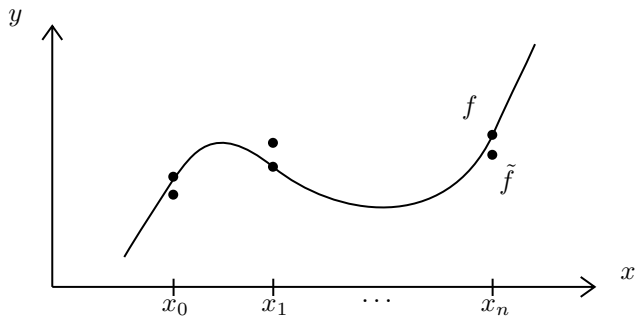
$$\begin{aligned} \underbrace{G^{(n+1)}(\xi)}_{=0} &= f^{(n+1)}(\xi) - (n+1)!\frac{E_n(x)}{\omega_{n+1}(x)} \\ 0 &= f^{(n+1)}(\xi) - (n+1)!\frac{E_n(x)}{\omega_{n+1}(x)} \\ E_n(x) &= \frac{\omega_{n+1}(x)}{(n+1)!} f^{(n+1)}(\xi). \end{aligned}$$

■

4.2 Stabilità del polinomio di interpolazione

Una proprietà desiderabile per un polinomio di interpolazione $\Pi_n f(x)$ è che l'errore tenda a zero se $n \to \infty$. In questa sezione sarà presentato un significativo controesempio (detto *di Runge*), che mostra come basti aumentare n di poco per far emergere problemi di stabilità agli estremi dell'intervallo. Nelle sezioni successive saranno presentati metodi per risolvere questo problema, ad esempio usando nodi non equispaziati.

Sia $x_0, x_1, \dots, x_n$ un insieme di $n+1$ punti distinti:



Costruiamo il polinomio di interpolazione di Lagrange di $f(x)$ dato da:

$$\Pi_n f(x) = \sum_{i=0}^{n} f(x_i)\mathcal{L}_i(x)$$

dove

$$\mathcal{L}_i(x) = \prod_{j=0, j\neq i}^{n} \frac{(x - x_j)}{(x_i - x_j)} \quad \forall i = 0, \ldots, n.$$

I valori $f(x_i)$ possono essere affetti da errori di rappresentazione al calcolatore. Consideriamo quindi il polinomio di interpolazione ottenuto perturbando le valutazioni di $f(\cdot)$ nei nodi x_i:

$$\tilde{\Pi}_n f(x) = \sum_{i=0}^{n} \tilde{f}(x_i)\mathcal{L}_i(x).$$

Come si comporta l'errore di perturbazione $\Pi_n f(x) - \tilde{\Pi}_n f(x)$ per $n \to \infty$?

$$\begin{aligned}\Pi_n f(x) - \tilde{\Pi}_n f(x) &= \sum_{i=0}^{n} f(x_i)\mathcal{L}_i(x) - \sum_{i=0}^{n} \tilde{f}(x_i)\mathcal{L}_i(x) \\ &= \sum_{i=0}^{n} \mathcal{L}_i(x) \underbrace{\left[f(x_i) - \tilde{f}(x_i)\right]}_{\text{errore di perturbazione}}.\end{aligned}$$

Passando ora alla norma infinito:

$$\left\|\Pi_n f(x) - \tilde{\Pi}_n f(x)\right\|_\infty \leq \underbrace{\left\|\sum_{i=0}^{n} \mathcal{L}_i(x)\right\|_\infty}_{\Lambda_n(x)} \cdot \underbrace{\max_{i=1,\ldots,n} \left|f(x_i) - \tilde{f}(x_i)\right|}_{\text{dipende solo dalla perturbazione}}$$

$$\left\|\Pi_n f(x) - \tilde{\Pi}_n f(x)\right\|_\infty \leq \Lambda_n(x) \cdot \max_{i=1,\ldots,n} \left|f(x_i) - \tilde{f}(x_i)\right|.$$

Notiamo che $\Lambda_n(x)$ dipende solo dalla distribuzione dei nodi di approssimazione. Essa prende il nome di **costante di Lebesgue** in quanto valore noto se i nodi sono fissati, e indipendente dalla funzione approssimata. In un certo senso è il *condizionamento* del problema.

Concludiamo che l'errore di perturbazione è piccolo solo se $\Lambda_n(x)$ è piccola. In generale, non possiamo garantirlo, in quanto cresce per $n \to +\infty$. Si può però dimostrare che prendendo i nodi di interpolazione *equispaziati*, allora:

$$\Lambda_n(x) = \left\| \sum_{i=0}^{n} \mathcal{L}_i(x) \right\|_\infty \approx \frac{2^{n+1}}{e\, n\, (\log n + \gamma)}, \quad \gamma \approx \frac{1}{2}.$$

Controesempio (Fenomeno di Runge). Consideriamo la cosiddetta funzione di Runge, definita come:

$$f : [-5, 5] \to \mathbb{R} \quad f(x) = \frac{1}{1 + x^2}.$$

Se si prova a interpolare la **funzione di Runge** su un set di nodi equispaziati, per n che cresce si generano delle oscillazioni via via sempre più marcate, come si vede in figura 4.5.

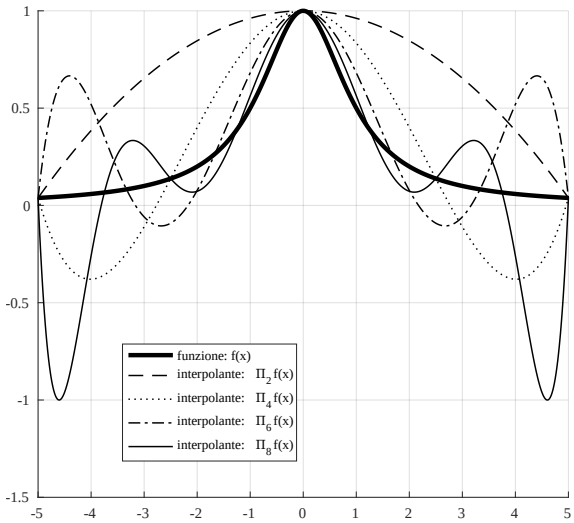


Figura 4.5: Fenomeno di Runge con polinomi interpolanti di grado n crescente.

Le oscillazioni, all'aumentare del grado, peggiorano solo *in prossimità degli estremi dell'intervallo*. Le possibili soluzioni al fenomeno di Runge sono:

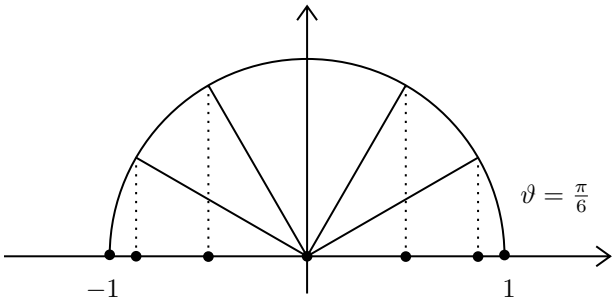
1. Utilizzo di **nodi non equispaziati**, concentrati dove ci sono oscillazioni agli estremi

2. Utilizzo di **nodi equispaziati** ma con metodi diversi:
   (a) *Interpolazione composita*: si divide l'intervallo di approssimazione in più parti calcolando su ciascun sottointervallo un polinomio interpolante di grado n non elevato $\Pi_h^n f$.
   (b) *Approssimazione nel senso dei minimi quadrati*, un approccio diverso dall'interpolazione vista finora.

4.3 Utilizzo dei nodi non equispaziati

4.3.1 Nodi di Chebyshev-Gauss-Lobatto (CGL)

Il seguente ragionamento sarà applicato all'intervallo $I = [-1, 1]$, ma esso può essere esteso a tutto $\mathbb{R}$. Consideriamo la semicirconferenza unitaria, fissando n e dividendola in n spicchi di ampiezza $\frac{\pi}{n}$. Per esempio, se $n = 6$ si ha:



I nodi x_i sono definiti come

$$x_i = -\cos\left(\frac{\pi i}{n}\right), \quad i = 0, \dots, n. \tag{4.2}$$

Si può dimostrare che:

- $E_n = |\Pi_n f - f| \xrightarrow{n\to\infty} 0$ con $\Pi_n f$ l'interpolante di Lagrange associata ai nodi di CGL.
- Vale che:
$$\Lambda_n(x) \leq \frac{2}{\pi}\left[\log n + \gamma + \log\frac{8}{\pi^2}\right] + \frac{\pi}{72n^2}.$$

4.3.2 Nodi di Chebyshev-Gauss (CG)

Scegliamo solo i nodi che sono interni ad $I = [-1, 1]$, quindi tralasciando gli estremi. In questo modo:

$$x_i = -\cos\left(\frac{\pi(2i+1)}{2(n+1)}\right), \quad i = 0, \dots, n. \tag{4.3}$$

Si può dimostrare che:

- $E_n = |\Pi_n f - f| \xrightarrow{n\to\infty} 0$ con $\Pi_n f$ l'interpolante di Lagrange associata ai nodi di CG (f sufficientemente regolare).
- Vale che:

$$\Lambda_n(x) \leq \frac{2}{\pi}\left[\log(n+1) + \gamma + \log\frac{8}{\pi}\right] + \frac{\pi}{72(n+1)^2}.$$

NB. Sull'intervallo generico $[a, b]$ i nodi di CGL e CG si ottengono da (4.2) e da (4.3) con una trasformazione lineare:

$$\hat{x}_i = \frac{a+b}{2} + \frac{b-a}{2}x_i, \quad i = 0, \ldots, n.$$

4.4 Interpolazione composita

Assegnata $f : [a, b] \to \mathbb{R}$, vogliamo approssimare f usando nodi equispaziati con n piccolo e dividendo $[a, b]$ in sottointervalli. In particolare:

1. Dividiamo $[a, b]$ in M intervalli di ampiezza h, con

$$I_i = [x_i, x_{i+1}] \quad i = 0, \ldots, M-1 \quad \text{e} \quad x_i = a + ih.$$

2. Su ciascun sottointervallo I_i approssimiamo la funzione con un polinomio $\Pi_h^k f(x)$ di interpolazione di Lagrange di grado $k \geq 1$, con k *non troppo grande* in modo da ovviare ai problemi di stabilità. Quindi il polinomio di interpolazione composita $\Pi_h^k f(x)$ appartiene allo spazio

$$\Pi_h^k f(x) \in X_h^k = \left\{ v \in C^0([a,b]) \ t.c. \ v|_{I_i} \in \mathbb{P}_k(I_i), \quad \forall i = 0, \ldots, n-1 \right\}.$$

Teorema 4.4. Sia $f \in C^{k+1}(I), I = [a, b]$ e sia $\Pi_h^k f(x)$ il suo polinomio di interpolazione composita. Allora:

$$\left\| f(x) - \Pi_h^k f(x) \right\|_\infty \leq C \ h^{k+1} \ \left\| f^{(k+1)} \right\|_\infty.$$

Osservazione. Per $k = 1$ il teorema diventa

$$\left\| f(x) - \Pi_h^1 f(x) \right\|_\infty \leq C \ h^2 \ \|f''\|_\infty \xrightarrow{h\to 0} 0.$$

4.5 Approssimazione nel senso dei minimi quadrati

Se i dati di cui siamo in possesso sono molto numerosi, potrebbe non aver senso cercare un'interpolazione su di essi, sia per ragioni di complessità computazionale,

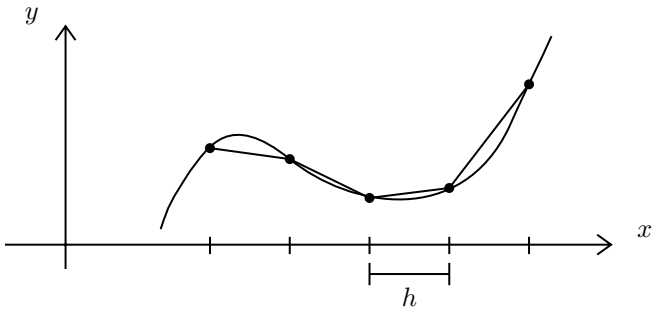
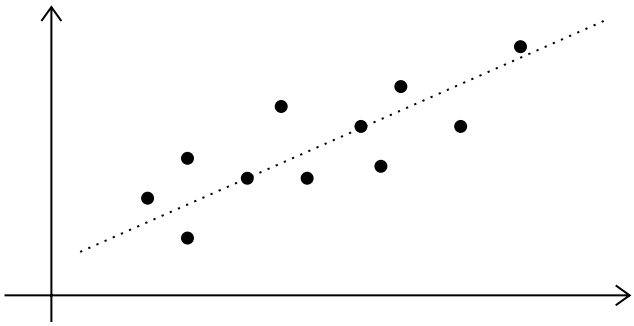


Figura 4.6: Già nel caso di interpolazione composita con $k = 1$ si può notare visivamente una maggiore *precisione* nell'interpolazione, data dal fatto che stiamo lavorando su tanti intervalli di ampiezza h, pur usando un banale polinomio lineare.

sia per il rischio di *overfitting*, cioè la creazione di un'approssimazione eccessivamente accurata dei dati quando probabilmente essi sono affetti da errore. In un esempio bidimensionale, potrebbe essere sufficiente una retta che descrive il comportamento qualitativo di una "nuvola" di punti.

Sia (x_i, y_i), $i = 0, \ldots, n$ un insieme di $n + 1$ coppie di punti.



Vogliamo costruire il miglior polinomio possibile $p_m(x)$ di grado $m < n$, che approssimi la nuvola di dati nel seguente senso:

$$\sum_{i=0}^{n}(y_i - p_m(x_i))^2 \leq \sum_{i=0}^{n}(y_i - q_m(x_i))^2, \quad \forall q_m \in \mathbb{P}^m. \tag{4.4}$$

Se esiste il polinomio $p_m(x)$ che realizza il minimo, esso è detto polinomio di approssimazione di grado m nel senso dei **minimi quadrati**.

Osservazione. La nozione di approssimazione nel senso dei minimi quadrati è

consistente con quanto discusso fino ad ora: con $m = n$ la (4.4) diventa

$$\underbrace{\sum_{i=0}^{n}(y_i - \overbrace{p_n(x_i)}^{\Pi_n(x_i)})^2}_{=0} \leq \sum_{i=0}^{n}(y_i - q_n(x_i))^2,$$

ovvero otteniamo il polinomio *migliore* imponendo il vincolo di passaggio su ogni singolo nodo. Tuttavia questa scelta porta ovviamente a un polinomio di grado altissimo, e quindi fortemente instabile, come mostrato nel controesempio di Runge.

4.5.1 Caso lineare

Sia $m = 1$. In tal caso si ha la **retta dei minimi quadrati** o **retta di regressione**. Abbiamo $n + 1$ coppie di dati $(x_i, y_i), i = 0, \ldots, n$. Vogliamo costruire la miglior retta possibile nell'insieme delle rette, cioè dei polinomi di primo grado. Ovvero, cerchiamo $p_1(x)$ tale che:

$$\sum_{i=0}^{n}(y_i - p_1(x_i))^2 \leq \sum_{i=0}^{n}(y_i - q_1(x_i))^2, \quad \forall q_1 \in \mathbb{P}^1. \tag{4.5}$$

Poiché $p_1(x)$ è un polinomio lineare, avrà la forma:

$$p_1(x) = \tilde{\alpha} + \tilde{\beta}x \quad \text{con } \tilde{\alpha}, \tilde{\beta} \in \mathbb{R}.$$

Inoltre, anche ogni altro polinomio lineare $q_1(x) \in \mathbb{P}^1$ si può scrivere come:

$$q_1(x) = \alpha + \beta x \quad \text{con } \alpha, \beta \in \mathbb{R}.$$

Quindi possiamo riscrivere (4.5): trovare $\tilde{\alpha}, \tilde{\beta} \in \mathbb{R}$ tali che

$$\sum_{i=0}^{n}\Big(y_i - \Big(\tilde{\alpha} + \tilde{\beta}x_i\Big)\Big)^2 \leq \sum_{i=0}^{n}(y_i - (\alpha + \beta x_i))^2.$$

Analogamente, definendo $\Phi(\alpha, \beta) = \sum_{i=0}^{n}(y_i - (\alpha + \beta x_i))^2$, il problema diventa trovare:

$$\Phi(\tilde{\alpha}, \tilde{\beta}) = \min_{\alpha,\beta\in\mathbb{R}} \Phi(\alpha, \beta).$$

Osserviamo che $\Phi(\cdot, \cdot)$ è un paraboloide convesso, quindi il suo unico punto di minimo si trova imponendo la condizione:

$$\frac{\partial\Phi(\alpha, \beta)}{\partial\alpha} = 0 \qquad \text{e} \qquad \frac{\partial\Phi(\alpha, \beta)}{\partial\beta} = 0. \tag{4.6}$$

Calcoliamo $\tilde{\alpha}, \tilde{\beta}$ tale che (4.6) sia soddisfatta:

$$\begin{cases} \dfrac{\partial \Phi(\alpha,\beta)}{\partial \alpha} = \displaystyle\sum_{i=0}^{n} -2(y_i - \alpha - \beta x_i) = 0 \\ \dfrac{\partial \Phi(\alpha,\beta)}{\partial \beta} = \displaystyle\sum_{i=0}^{n} -2x_i(y_i - \alpha - \beta x_i) = 0 \end{cases}$$

$$\Rightarrow \quad \begin{cases} \displaystyle\sum_{i=0}^{n} -2y_i + 2\alpha + 2\beta x_i = 0 \\ \displaystyle\sum_{i=0}^{n} -2x_i y_i + 2x_i\alpha + 2\beta x_i^2 = 0. \end{cases}$$

Abbiamo quindi ottenuto un sistema lineare di condizioni. Dividendo per 2 e riscrivendolo in forma matriciale:

$$\begin{bmatrix} (n+1) & \displaystyle\sum_{i=0}^{n} x_i \\ \displaystyle\sum_{i=0}^{n} x_i & \displaystyle\sum_{i=0}^{n} x_i^2 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \displaystyle\sum_{i=0}^{n} y_i \\ \displaystyle\sum_{i=0}^{n} x_i y_i \end{bmatrix}.$$

La soluzione $\begin{bmatrix} \tilde{\alpha} \\ \tilde{\beta} \end{bmatrix}$ di questo sistema lineare è il punto di minimo di $\Phi(\alpha, \beta)$, e fornisce quindi i coefficienti di $p_1(x)$. Pertanto, la retta di regressione lineare $p_1(x) = \tilde{\alpha} + \tilde{\beta}x$ si può calcolare attraverso la soluzione del seguente sistema lineare (2×2):

$$\begin{bmatrix} (n+1) & \displaystyle\sum_{i=0}^{n} x_i \\ \displaystyle\sum_{i=0}^{n} x_i & \displaystyle\sum_{i=0}^{n} x_i^2 \end{bmatrix} \begin{bmatrix} \tilde{\alpha} \\ \tilde{\beta} \end{bmatrix} = \begin{bmatrix} \displaystyle\sum_{i=0}^{n} y_i \\ \displaystyle\sum_{i=0}^{n} x_i y_i \end{bmatrix}.$$

Osserviamo anche che la matrice è SDP, quindi il sistema ammette una e una sola soluzione come visto nei teoremi 2.1 e 2.3.

4.5.2 Caso generale

Sia $m \geq 1$. Si può procedere come prima e riscrivere $p_m(x)$ come:

$$p_m(x) = \tilde{\alpha}_0 + \tilde{\alpha}_1 x \ + \tilde{\alpha}_2 x^2 + \ldots + \tilde{\alpha}_m x^m = \sum_{j=0}^{m} \tilde{\alpha}_j x^j \quad \text{con } \tilde{\alpha}_j \in \mathbb{R}.$$

Analogamente possiamo definire un apposito funzionale e minimizzarlo:

$$\min_{[\alpha_i, i=0,\ldots,m]} \Phi(\alpha_0, \alpha_1, \ldots, \alpha_m),$$

ossia differenziando rispetto a tutte le variabili e ponendo le derivate uguali a zero. Si ha il sistema di $m+1$ incognite:

$$\begin{cases} \dfrac{\partial \Phi}{\partial \alpha_0} = 0 \\ \vdots \\ \dfrac{\partial \Phi}{\partial \alpha_m} = 0 \end{cases},$$

allora:

$$\begin{bmatrix} (n+1) & \sum_{i=0}^{n} x_i & \dots & \sum_{i=0}^{n} x_i^m \\ \sum_{i=0}^{n} x_i & \sum_{i=0}^{n} x_i^2 & \dots & \sum_{i=0}^{n} x_i^{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=0}^{n} x_i^m & \sum_{i=0}^{n} x_i^{m+1} & \dots & \sum_{i=0}^{n} x_i^{2m} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_m \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} y_i \\ \sum_{i=0}^{n} x_i y_i \\ \vdots \\ \sum_{i=0}^{n} x_i^m y_i \end{bmatrix}.$$

Esso è un sistema di $(m+1)$ equazioni in $(m+1)$ incognite. La matrice è SDP, quindi il sistema ammette una e una sola soluzione:

$$\begin{bmatrix} \tilde{\alpha}_0 \\ \tilde{\alpha}_1 \\ \vdots \\ \tilde{\alpha}_m \end{bmatrix}$$

che è l'insieme di coefficienti che identificano il polinomio di approssimazione nel senso dei minimi quadrati.

4.6 Sistemi lineari sovradeterminati

Prendiamo un sistema lineare della forma:

$$A\mathbf{x} = \mathbf{b}$$

dove $A \in \mathbb{R}^{m \times n}, \mathbf{b} \in \mathbb{R}^m, \mathbf{x} \in \mathbb{R}^n$. Il sistema si dice:

- **sovradeterminato** se $m > n$,
- **sottodeterminato** se $m < n$.

Approfondiremo ora i sistemi sovradeterminati, caso di maggiore interesse. Sia dunque $A \in \mathbb{R}^{m \times n}, m \geq n$. Consideriamo il sistema lineare

$$A\mathbf{x} = \mathbf{b}, \qquad \mathbf{b} \in \mathbb{R}^m, \tag{4.7}$$

osserviamo che (4.7) ha soluzione nel senso classico del termine solo se il termine noto è un elemento del seguente spazio:

$$\text{range}(A) = \{\mathbf{y} \in \mathbb{R}^m \text{ tale che } A\mathbf{x} = \mathbf{y} \text{ per qualche } \mathbf{x} \in \mathbb{R}^n\}.$$

Dobbiamo quindi allargare il concetto di *soluzione* per studiare un sistema sovradeterminato.

Definizione 4.5 — Soluzione di un sistema lineare nel senso dei minimi quadrati. Dati $A \in \mathbb{R}^{m\times n}$, $m \geq n$, e $\mathbf{b} \in \mathbb{R}^m$, diciamo che $\mathbf{x}^\star \in \mathbb{R}^n$ è soluzione di (4.7) nel senso dei minimi quadrati se

$$\Phi(\mathbf{x}^\star) = \min_{\mathbf{x}\in\mathbb{R}^n} \Phi(\mathbf{x}),$$

dove il funzionale da minimizzare è $\Phi(\mathbf{w}) = \|A\mathbf{w} - \mathbf{b}\|_2^2$, il residuo nella norma euclidea.

Osservazione. Se $m = n$, la soluzione $\mathbf{x}^\star$ nel senso dei minimi quadrati coincide con la soluzione classica perché $\Phi(\mathbf{x}^\star) = 0$.

Dobbiamo chiarire se la soluzione ai minimi quadrati esista e se sia unica.

Lemma 4.6. Se la soluzione $\mathbf{x}^\star$ nel senso dei minimi quadrati di un sistema lineare esiste, essa coincide con la soluzione $\mathbf{x}^\star$ nel senso classico del *sistema di equazioni normali*:

$$A^T A\mathbf{x}^\star = A^T\mathbf{b}.$$

Dimostrazione. Scriviamo il funzionale da minimizzare:

$$\begin{aligned}
\Phi(\mathbf{x}) &= \|A\mathbf{x} - \mathbf{b}\|_2^2 \\
&= (A\mathbf{x} - \mathbf{b})^T(A\mathbf{x} - \mathbf{b}) \\
&= \left((A\mathbf{x})^T - \mathbf{b}^T\right)(A\mathbf{x} - \mathbf{b}) \\
&= (A\mathbf{x})^T A\mathbf{x} \underbrace{-\mathbf{b}^T A\mathbf{x} - (A\mathbf{x})^T\mathbf{b}}_{\langle A\mathbf{x},\mathbf{b}\rangle} + \mathbf{b}^T\mathbf{b} \\
&= \mathbf{x}^T A^T A\mathbf{x} - 2(A\mathbf{x})^T\mathbf{b} + \mathbf{b}^T\mathbf{b} \\
&= \mathbf{x}^T A^T A\mathbf{x} - 2\mathbf{x}^T A^T\mathbf{b} + \mathbf{b}^T\mathbf{b}.
\end{aligned}$$

Calcoliamone poi il gradiente

$$\nabla\Phi(\mathbf{x}) = 2A^T A\mathbf{x} - 2A^T\mathbf{b}$$

e poniamolo uguale a zero per trovare il minimo:

$$\begin{gathered}
\nabla\Phi(\mathbf{x}^\star) = 0 \\
\Updownarrow \\
A^T A\mathbf{x}^\star - A^T\mathbf{b} = 0 \\
\Updownarrow \\
A^T A\mathbf{x}^\star = A^T\mathbf{b}.
\end{gathered}$$

Teorema 4.7. Sia $A \in \mathbb{R}^{m\times n}$, $m \geq n$. Se A ha rango pieno, cioè $\text{rank}(A) = n$, allora $A^TA \in \mathbb{R}^{n\times n}$ è una matrice SDP e quindi il sistema di equazioni normali

$$A^TA\mathbf{x}^\star = A^T\mathbf{b}$$

ammette una e una sola soluzione.

Osservazione. Se $A^TA\mathbf{x}^\star = A^T\mathbf{b}$ ammette una e una sola soluzione in senso classico, allora (4.7) ammette una e una sola soluzione nel senso dei minimi quadrati.

Osservazione. In genere, la matrice A^TA è molto mal condizionata, quindi nella pratica è spesso difficile risolvere il sistema di equazioni normali per calcolare $\mathbf{x}^\star$. Dobbiamo quindi trovare un altro modo di procedere per la risoluzione.

4.6.1 Fattorizzazione QR

Proviamo a generalizzare il concetto di fattorizzazione LU, visto nella sezione 2.2 e proprio delle matrici quadrate, anche a matrici rettangolari come A.

Definizione 4.8 — Fattorizzazione QR. Sia $A \in \mathbb{R}^{m\times n}$, $m \geq n$. Si dice che A ammette una fattorizzazione QR se esistono:

- $Q \in \mathbb{R}^{m\times m}$ ortogonale ($Q^{-1} = Q^T$);
- $R \in \mathbb{R}^{m\times n}$ trapezoidale superiore (con le righe dalla $n+1$ in poi tutte nulle)

tali che

$$A = QR$$

$$\underbrace{\begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}}_{A\in\mathbb{R}^{m\times n}} = \underbrace{\begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}}_{Q\in\mathbb{R}^{m\times m}} \underbrace{\begin{bmatrix} \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot \\ 0 & 0 & \cdot \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{R\in\mathbb{R}^{m\times n}}$$

Proprietà (Fattorizzazione QR ridotta). Sia $A \in \mathbb{R}^{m\times n}$, $m \geq n$, di rango massimo di cui esista la fattorizzazione QR. Allora esiste un'unica fattorizzazione di A tale che:

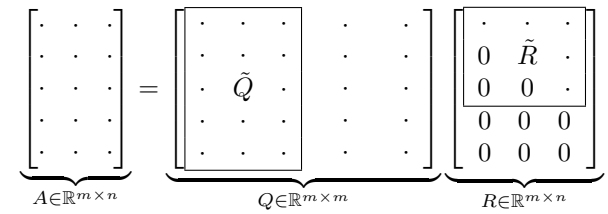
$$A = \tilde{Q}\ \tilde{R}$$

dove $\tilde{Q}$ e $\tilde{R}$ sono le sottomatrici ottenute da Q e R nel seguente modo:

- $\tilde{Q} = Q(1:m, 1:n) \in \mathbb{R}^{m\times n}$;

- $\tilde{R} = R(1:n, 1:n) \in \mathbb{R}^{n\times n}$.

Inoltre le colonne di $\tilde{Q}$ sono vettori ortonormali e $\tilde{R}$ coincide con il fattore di Cholesky della matrice A^TA, ossia $A^TA = \tilde{R}^T\tilde{R}$.

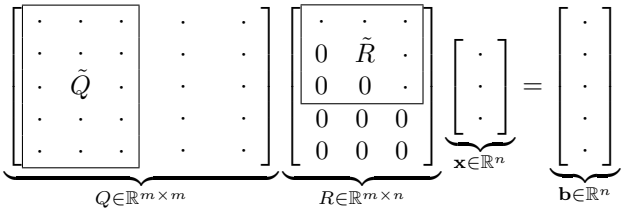


Come usiamo $A = \tilde{Q}\ \tilde{R}$ per risolvere $A\mathbf{x} = \mathbf{b}$?

Teorema 4.9. Sia $A \in \mathbb{R}^{m\times n}$, $m \geq n$, di rango massimo, e sia $\mathbf{b} \in \mathbb{R}^m$. Allora esiste un'unica soluzione $\mathbf{x}^\star \in \mathbb{R}^n$ nel senso dei minimi quadrati del sistema sovradimensionato $A\mathbf{x} = \mathbf{b}$ ed essa è data da:

$$\mathbf{x}^\star = \tilde{R}^{-1}\tilde{Q}^T\mathbf{b} \qquad \text{ovvero} \qquad \tilde{R}\mathbf{x}^\star = \tilde{Q}^T\mathbf{b}$$

dove $\tilde{Q}$ ed $\tilde{R}$ sono i fattori della decomposizione ridotta di A.



Idea di dimostrazione. Ci concentreremo sulla dimostrazione dell'esistenza, tralasciando quella dell'unicità. Supponendo che A ammenta una fattorizzazione QR, allora:

$$\begin{aligned}
\|A\mathbf{x} - \mathbf{b}\|_2^2 &= \|QR\mathbf{x} - \mathbf{b}\|_2^2 \\
&= \left\|Q^TQR\mathbf{x} - Q^T\mathbf{b}\right\|_2^2 && \|\mathbf{z}\|_2 = \left\|Q^T\mathbf{z}\right\|_2\ \forall\mathbf{z} \\
&= \left\|R\mathbf{x} - Q^T\mathbf{b}\right\|_2^2 && (Q \text{ è ortogonale}) \\
&= \left\|\tilde{R}\mathbf{x} - \tilde{Q}^T\mathbf{b}\right\|_2^2 + \sum_{i=n+1}^{m}\left[\left(Q^T\mathbf{b}\right)_i\right]^2 && \forall\mathbf{x}\in\mathbb{R}^n.
\end{aligned}$$

Osserviamo in particolare l'ultimo passaggio: abbiamo la norma (al quadrato, ma ciò è irrilevante) di un vettore, ovvero $R\mathbf{x} - Q^T\mathbf{b}$. Dato che la norma 2 al quadrato è per definizione (vedi 2.11) la somma delle componenti al quadrato, possiamo liberamente separare in due somme. Il termine $R\mathbf{x}$ si separa in $\tilde{R}\mathbf{x}$ e

una somma di zeri. Il termine $Q^T\mathbf{b}$ si separa in $\tilde{Q}^T\mathbf{b}$ e il termine in sommatoria (si noti l'indice che va da $n+1$ a m). Il minimo di $\Phi(\mathbf{x})$ è raggiunto per $\mathbf{x}^\star$ che annulla tutto il termine in cui compare:

$$\tilde{R}\mathbf{x} - \tilde{Q}^T\mathbf{b} = 0 \quad \Rightarrow \quad \left\| \tilde{R}\mathbf{x} - \tilde{Q}^T\mathbf{b} \right\| = 0$$

e dunque $\mathbf{x}^\star$ soddisfa

$$\tilde{R}\mathbf{x}^\star - \tilde{Q}^T\mathbf{b} = 0 \quad \Rightarrow \quad \mathbf{x}^\star = \tilde{R}^{-1}\tilde{Q}^T\mathbf{b}. \qquad \blacksquare$$

Osservazione. Il costo di calcolare la fattorizzazione QR è $\approx mn^2$. L'algoritmo per il calcolo di Q è basato sull'algoritmo di Gram-Schmidt, la tecnica dall'algebra lineare per la costruzione di una base ortogonale.

Capitolo 5

Integrazione numerica

Sia assegnata la funzione continua $f : [a, b] \to \mathbb{R}$, supponiamo di volerne calcolare l'integrale definito

$$I(f) = \int_a^b f(x)\mathrm{d}x. \tag{5.1}$$

Calcolare analiticamente la primitiva di f può essere complicato o addirittura impossibile. L'idea è quindi quella di approssimare f con un polinomio, con i metodi mostrati nel capitolo precedente, e calcolare l'integrale del polinomio approssimante:

$$I(f) = \int_a^b f(x)\mathrm{d}x \approx I_n(f) = \int_a^b \Pi_n f(x)\mathrm{d}x.$$

Se

$$\Pi_n f(x) = \sum_{i=0}^{n} f(x_i)\mathcal{L}_i(x)$$

possiamo scrivere:

$$I_n(f) = \underbrace{\int_a^b \sum_{i=0}^{n} f(x_i)\mathcal{L}_i(x)\mathrm{d}x = \sum_{i=0}^{n} f(x_i) \underbrace{\int_a^b \mathcal{L}_i(x)\mathrm{d}x}_{\alpha_i}}_{\text{per linearità}}.$$

Si noti che α_i non dipende dalla funzione che stiamo integrando, ma solo dall'intervallo di integrazione e dal i-esimo polinomio di Lagrange (cioè dal nodo i-esimo). Quindi

$$I_n(f) = \sum_{i=0}^{n} \alpha_i\, f(x_i) \tag{5.2}$$

è una **formula di quadratura di tipo interpolatorio**.

- $x_i, i = 0, \dots, n$ si chiamano **nodi di quadratura**;
- $\alpha_i, i = 0, \dots, n$ si chiamano **pesi di quadratura**.

Cerchiamo di quantificare l'errore compiuto approssimando l'integrale con l'interpolante: possiamo intuire che sia legato all'errore commesso nell'approssimazione del polinomio.

5.1 Formule di quadratura semplici

5.1.1 Formula del punto medio

Vediamo il caso $n = 0$. Approssimiamo $f(x)$ come $\Pi_0 f(x)$, cioè con il valore assunto nel punto medio dell'intervallo $[a, b]$ detto $x_0 = (a + b)/2$.

$$\begin{aligned} I(f) \approx I_0(f) &= \int_a^b \Pi_0 f(x)\mathrm{d}x = \int_a^b f(x_0)\mathcal{L}_0(x)\mathrm{d}x \\ &= f(x_0) \underbrace{\int_a^b \mathcal{L}_0(x)\mathrm{d}x}_{\alpha_0} \\ &= f(x_0)\alpha_0. \end{aligned}$$

Il peso è quindi[1]

$$\alpha_0 = \int_a^b \mathcal{L}_0(x)\mathrm{d}x = \int_a^b \mathrm{d}x = [x]_a^b = b - a,$$

ovvero:

$$I_0(f) = (b - a) f\left(\frac{a + b}{2}\right).$$

5.1.2 Formula del trapezio

Vediamo il caso $n = 1$. Approssimiamo $f(x)$ come $\Pi_1 f(x)$, cioè il polinomio di grado 1 (una retta) che congiunge i due nodi (ovvero gli estremi), si ha

$$I(f) \approx I_1(f) = \int_a^b \Pi_1 f(x)\mathrm{d}x = \int_a^b \sum_{i=0}^{1} f(x_i)\mathcal{L}_i(x)\mathrm{d}x$$

[1] Si noti che i polinomi di Lagrange saranno diversi tra i vari casi $n = 0, 1, 2, \dots$ Per esempio nel caso $n = 0$ è presente un solo nodo, quindi è sufficiente imporre che in x_0 valga 1, ovvero è il polinomio costante 1. Nel caso $n = 1$ vi sono due nodi, quindi i polinomi dovranno rispettivamente valere 1 in un nodo e 0 nell'altro, avranno quindi espressioni diverse dal caso $n = 0$.

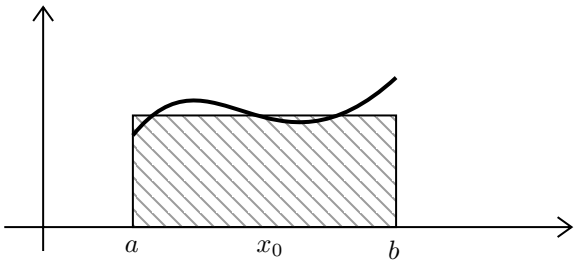


Figura 5.1: Formula del punto medio.

$$= f(a)\underbrace{\int_a^b \mathcal{L}_0(x)\mathrm{d}x}_{\alpha_0} + f(b)\underbrace{\int_a^b \mathcal{L}_1(x)\mathrm{d}x}_{\alpha_1}$$
$$= f(a)\alpha_0 + f(b)\alpha_1.$$

I pesi sono quindi

$$\alpha_0 = \int_a^b \mathcal{L}_0(x)\mathrm{d}x = \ldots = \frac{b-a}{2}$$
$$\alpha_1 = \int_a^b \mathcal{L}_1(x)\mathrm{d}x = \ldots = \frac{b-a}{2},$$

ovvero:

$$I_1(f) = (b-a)\left[\frac{f(a)+f(b)}{2}\right].$$

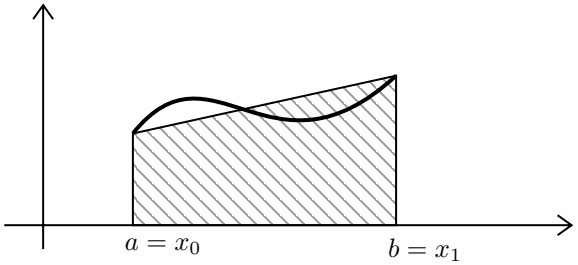


Figura 5.2: Formula del trapezio.

5.1.3 Formula di Cavalieri-Simpson

Vediamo infine il caso $n = 2$. Approssimiamo $f(x)$ come $\Pi_2 f(x)$, cioè il polinomio di grado 2 (una parabola) che congiunge i tre nodi, ovvero gli estremi e il punto

medio dell'intervallo $[a, b]$:

$$\begin{aligned} I(f) \approx I_2(f) &= \int_a^b \Pi_2 f(x)\mathrm{d}x = \int_a^b \sum_{i=0}^{2} f(x_i)\mathcal{L}_i(x)\mathrm{d}x \\ &= f(a)\underbrace{\int_a^b \mathcal{L}_0(x)\mathrm{d}x}_{\alpha_0} + f\left(\frac{a+b}{2}\right)\underbrace{\int_a^b \mathcal{L}_1(x)\mathrm{d}x}_{\alpha_1} + f(b)\underbrace{\int_a^b \mathcal{L}_2(x)\mathrm{d}x}_{\alpha_2} \\ &= f(a)\alpha_0 + f\left(\frac{a+b}{2}\right)\alpha_1 + f(b)\alpha_2. \end{aligned}$$

I pesi sono quindi

$$\begin{aligned} \alpha_0 &= \int_a^b \mathcal{L}_0(x)\mathrm{d}x = \int_a^b \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}\mathrm{d}x = \ldots = \frac{b-a}{6} \\ \alpha_1 &= \int_a^b \mathcal{L}_1(x)\mathrm{d}x = \int_a^b \frac{(x-x_2)(x-x_0)}{(x_1-x_0)(x_1-x_2)}\mathrm{d}x = \ldots = \frac{4}{6}(b-a) \\ \alpha_2 &= \int_a^b \mathcal{L}_2(x)\mathrm{d}x = \int_a^b \frac{(x-x_1)(x-x_0)}{(x_2-x_1)(x_2-x_0)}\mathrm{d}x = \ldots = \frac{b-a}{6}, \end{aligned}$$

ovvero:

$$I_2(f) = \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right].$$

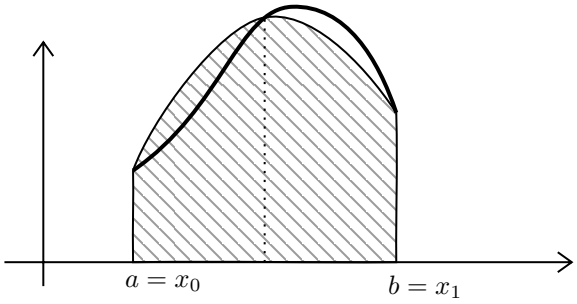


Figura 5.3: Formula di Cavalieri-Simpson.

5.2 Errore delle formule di quadratura semplici

Studiamo l'accuratezza delle approssimazioni appena enunciate.

Definizione 5.1 — Errore di quadratura. Siano

$$I(f) = \int_a^b f(x)\mathrm{d}x \quad \text{e} \quad I_n(f) = \sum_{i=0}^{n} f(x_i)\alpha_i.$$

Definiamo l'**errore di quadratura** come

$$E_n(f) \coloneqq I(f) - I_n(f).$$

Definizione 5.2 — Grado di esattezza. Diciamo che una formula di quadratura ha **grado di esattezza** $r \geq 0$ se

$$E_n(p_r) = 0 \quad \forall p_r \in \mathbb{P}^r.$$

5.2.1 Calcolo dell'errore

Ricordiamo anzitutto il teorema della media integrale pesata.

Teorema 5.3 — Media integrale pesata. Siano $f, g \in C^0([a,b])$ tali che g abbia segno costante in (a,b). Allora $\exists \xi \in [a,b]$ tale che

$$\int_a^b f(x)g(x)\mathrm{d}x = f(\xi)\int_a^b g(x)\mathrm{d}x.$$

Riportiamo ora i teoremi relativi agli errori delle tre formule presentate, dimostrando solo il primo[2].

Teorema 5.4 — Errore della formula del punto medio. Data $f \in C^2([a,b])$, esiste $\xi \in (a,b)$ tale che

$$E_0(x) = \frac{1}{24} f''(\xi)(b-a)^3.$$

Dimostrazione. Per ipotesi abbiamo:

$$I(f) = \int_a^b f(x)\mathrm{d}x \quad \text{e} \quad I_0(f) = (b-a) f\underbrace{\left(\frac{a+b}{2}\right)}_{x_m}.$$

Supponiamo che f sia sufficientemente regolare in modo da calcolare l'espansione di Taylor: $\exists \eta(x) \in (x, x_m)$ tale che

$$f(x) = f(x_m) + (x - x_m)f'(x_m) + \frac{(x-x_m)^2}{2} f''(\eta(x)).$$

[2]Le altre due dimostrazioni sono lasciate al lettore come esercizio.

Calcoliamo l'errore:

$$\begin{aligned} E_0(f) &= I(f) - I_0(f) \\ &= \int_a^b f(x)\mathrm{d}x - (b-a)f(x_m) \\ &= \int_a^b [f(x) - f(x_m)]\mathrm{d}x \\ &= \underbrace{\int_a^b (x-x_m)f'(x_m)\mathrm{d}x}_{A} + \underbrace{\int_a^b \frac{(x-x_m)^2}{2} f''(\eta(x))\mathrm{d}x}_{B}. \end{aligned}$$

- A rappresenta l'integrale di una retta che passa per il punto medio dell'intervallo, quindi $A = f'(x_m)\int_a^b (x-x_m)\mathrm{d}x = 0$.
- Per B utilizziamo il teorema della media integrale pesata:

$$\begin{aligned} B &= \frac{1}{2}\int_a^b (x-x_m)^2 f''(\eta(x))\mathrm{d}x \\ &= \frac{1}{2} f''(\xi)\int_a^b (x-x_m)^2\mathrm{d}x \\ &= \frac{1}{6} f''(\xi)\left[(x-x_m)^3\right]_a^b \\ &= \frac{1}{24} f''(\xi)(b-a)^3. \end{aligned}$$

Allora otteniamo che:

$$E_0(x) = A + B = \frac{1}{24} f''(\xi)(b-a)^3. \qquad \blacksquare$$

Teorema 5.5 — Errore della formula del trapezio. Data $f \in C^2([a,b])$, esiste $\xi_2 \in (a,b)$ tale che

$$E_1(x) = -\frac{1}{12} f''(\xi_2)(b-a)^3.$$

Teorema 5.6 — Errore della formula di Cavalieri-Simpson. Data $f \in C^4([a,b])$, esiste $\xi_3 \in (a,b)$ tale che

$$E_2(x) = -\frac{1}{90 \cdot 32} f^{(4)}(\xi_3)(b-a)^5.$$

Osservazioni.

1. Dall'espressione dell'errore di una formula di quadratura possiamo leggere il grado di esattezza:
   (a) nel caso $n = 0$, ovvero l'errore della formula del punto medio, è presente la derivata seconda. Chiaramente in qualunque punto dell'intervallo, la derivata seconda di un polinomio di grado 1 è sempre nulla, pertanto tutti i polinomi di grado fino a 1 sono integrati esattamente da quella formula (cioè l'errore è nullo). Il grado di esattezza è quindi 1.
   (b) nel caso $n = 1$, ovvero l'errore della formula del trapezio, si ragiona analogamente e si deduce che il grado di esattezza è 1, come nella formula del punto medio.
   (c) nel caso $n = 2$, ovvero la formula di Cavalieri-Simpson, compare la derivata quarta. Il grado di esattezza è dunque 3.
2. Se n è pari, il grado di esattezza è $n + 1$, altrimenti è n.
3. Una formula di quadratura a $n + 1$ nodi ha sempre grado di esattezza $\geq n$ perché $p_n(x) \equiv \Pi_n p_n(x)$, ovvero un polinomio coincide con il suo polinomio approssimante.
4. Le stime indicate hanno un'utilità limitata: includono il punto ξ di cui è sconosciuta la posizione esatta. Per poterle utilizzare nella pratica, dobbiamo quindi passare alla norma infinito per limitare dall'alto l'errore:

$$
\begin{aligned}
|E_0(x)| &\leq \tfrac{1}{24}(b-a)^3 \underbrace{\|f''(x)\|_\infty}_{\leq C} \\
|E_1(x)| &\leq \tfrac{1}{12}(b-a)^3 \underbrace{\|f''(x)\|_\infty}_{\leq C} \\
|E_2(x)| &\leq \tfrac{1}{90}\tfrac{1}{32}(b-a)^5 \underbrace{\left\|f^{(4)}(x)\right\|_\infty}_{\leq C}.
\end{aligned}
$$

Torniamo al caso generale di una formula di tipo interpolatorio.

$$
\left.
\begin{aligned}
I(f) &= \int_a^b f(x)\mathrm{d}x \\
I_n(f) &= \int_a^b \Pi_n f(x)\mathrm{d}x
\end{aligned}
\right\} \Rightarrow E_n(f) = \int_a^b \underbrace{[f(x) - \Pi_n f(x)]}_{\text{errore del polinomio}} \mathrm{d}x.
$$

Osservazione. $E_n(f)$ erediterà tutti i problemi di stabilità dell'errore di interpolazione del polinomio su nodi equispaziati visti nella sezione 4.2. Aumentando n per migliorare la qualità dell'approssimazione, ci scontriamo con il fenomeno di Runge nell'approssimazione del polinomio. Possiamo risolvere il problema usando:

- interpolazione composita, che darà origine alle formule di quadratura composite;

- nodi non equispaziati, che daranno origine alle formule di quadratura su nodi non equispaziati.

5.3 Formule di quadratura composite

L'idea generale è molto semplice: dividiamo l'intervallo $[a, b]$ in N sottointervalli di ampiezza H ed applichiamo quanto visto a ciascun sottointervallo. Già dal primo caso sarà chiaro come l'uso di formule composite permette di raggiungere risultati migliori senza che sia necessario aumentare n, ottenendo formule parecchio complicate, ma semplicemente raffinando la griglia di sottointervalli (diminuendo H).

5.3.1 Punto medio composito

Fissiamo $N \geq 1$ e $H = (b-a)/N$. Ogni nodo è $x_k = a + kH$ con $k = 0, 1, \ldots, N$.

$$I(f) = \int_a^b f(x)\mathrm{d}x = \sum_{k=0}^{N-1} \underbrace{\int_{x_k}^{x_{k+1}} f(x)\mathrm{d}x}_{\text{formula del punto medio}}$$
$$\approx \sum_{k=0}^{N-1} Hf\left(\frac{x_k + x_{k+1}}{2}\right) = I_0^c(f).$$

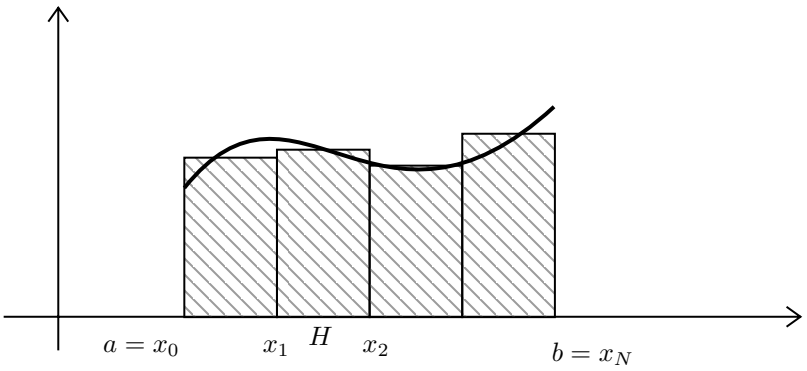


Figura 5.4: Punto medio composito su N intervalli di ampiezza H.

Teorema 5.7. Data $f \in C^2([a, b])$, esiste $\xi \in (a, b)$ tale che:

$$E_0^c(f) = I(f) - I_0^c(f) = \frac{b-a}{24} H^2 f''(\xi).$$

Questo ha ordine di accuratezza 2 e grado di esattezza 1.

Definizione 5.8. L'esponente di H è detto **ordine di accuratezza** della formula di quadratura. Dice quanto velocemente l'errore va a 0 al tendere di H a 0, dove H è l'ampiezza dell'intervallo.

5.3.2 Trapezio composito

Fissiamo $N \geq 1$ e $H = (b-a)/N$. Ogni nodo è $x_k = a + kH$ con $k = 0, 1, \ldots, N$.

$$I(f) = \int_a^b f(x)\mathrm{d}x = \sum_{k=0}^{N-1} \underbrace{\int_{x_k}^{x_{k+1}} f(x)\mathrm{d}x}_{\text{formula del trapezio}}$$
$$\approx \sum_{k=0}^{N-1} \frac{H}{2}[f(x_k) + f(x_{k+1})] = I_1^c(f).$$

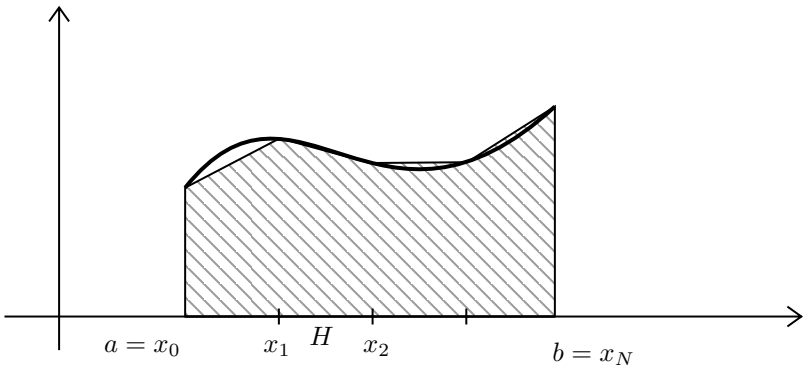


Figura 5.5: Trapezio composito su N intervalli di ampiezza H.

Teorema 5.9. Data $f \in C^2([a,b])$, esiste $\eta \in (a,b)$ tale che:

$$E_1^c(f) = I(f) - I_1^c(f) = -\frac{b-a}{12} H^2 f''(\eta).$$

Questo ha ordine di accuratezza 2 e grado di esattezza 1.

5.3.3 Cavalieri-Simpson composito

Fissiamo $N \geq 1$ e $H = \frac{b-a}{N}$. Ogni nodo è $x_k = a + kH$ con $k = 0, 1, \ldots, N$.

$$I(f) = \int_a^b f(x)\mathrm{d}x = \sum_{k=0}^{N-1} \underbrace{\int_{x_k}^{x_{k+1}} f(x)\mathrm{d}x}_{\text{formula di CS}}$$
$$\approx \sum_{k=0}^{N-1} \frac{H}{6}\left[f(x_k) + 4f\left(\frac{x_k + x_{k+1}}{2}\right) + f(x_{k+1})\right] = I_2^c(f).$$

Teorema 5.10. Data $f \in C^4([a,b])$, esiste $\zeta \in (a,b)$ tale che:

$$E_2^c(f) = I(f) - I_2^c(f) = -\frac{b-a}{180}\frac{1}{16}H^4 f^{(4)}(\zeta).$$

Questo ha ordine di accuratezza 4 e grado di esattezza 3.

Esempio. Calcoliamo l'errore commesso usando la formula del punto medio su

$$I(f) = \int_0^\pi f(x)\mathrm{d}x \quad \text{con} \quad f(x) = \sin(x).$$

Abbiamo dunque che:

$$E_0(f) = \frac{1}{24}(b-a)^3 f''(\xi) \quad \xi \in [a,b]$$
$$= \frac{1}{24}(\pi - 0)^3 f''(\xi)$$
$$|E_0(f)| \leq \frac{\pi^3}{24}\|f''(x)\|_\infty = \frac{\pi^3}{24}\max_{x\in[0,\pi]} |-\sin x| \leq \frac{\pi^3}{24}.$$

5.4 Formule di Newton-Cotes (NC)

Sono basate sul metodo di interpolazione di Lagrange su nodi *equispaziati*, nonché una generalizzazione dei metodi visti finora. Fissiamo $n > 0$ e indichiamo i nodi di quadratura con:

$$x_k = x_0 + kh \quad \text{con} \quad k = 0, \ldots, n \quad \text{dove} \quad h = \frac{b-a}{n}.$$

La formula generale è data da:

$$I_n(f) = \sum_{i=0}^{n} \alpha_i f(x_i).$$

Le formule del punto medio, del trapezio e la formula di Simpson sono esempi di formule di Newton-Cotes, dove, rispettivamente, $n = 0$, $n = 1$ e $n = 2$. Nel caso generale si definiscono:

- **formule aperte**, quelle in cui $x_0 = a + h, x_n = b - h, h = (b - a)/(n + 2)$ con $n \geq 0$, per esempio le formule di punto medio;
- **formule chiuse**, quelle in cui $x_0 = a, x_n = b, h = (b - a)(n)$ con $n \geq 1$, per esempio le formule del trapezio e di Cavalieri-Simpson.

Proprietà. Le formule di Newton-Cotes hanno pesi di quadratura α_i che dipendono solo da n e da h, ma non dall'intervallo di integrazione $[a, b]$.

Dimostrazione. Per semplicità consideriamo il caso delle formule di NC chiuse. Effettuiamo il seguente cambio di variabile:

$$x = \Psi(t) = x_0 + th \quad (= a + th)$$

e notiamo che:

$$\Psi(0) = a \qquad \Psi(n) = b \qquad \Psi(k) = x_k \quad \text{con} \quad k = 0, \ldots, n.$$

Si vogliono ora calcolare le frazioni dei polinomi di Lagrange:

$$\frac{x - x_k}{x_i - x_k} = \frac{a + th - (a + kh)}{a + ih - (a + kh)} = \frac{h(t - k)}{h(i - k)} = \frac{t - k}{i - k},$$

ma quindi

$$\mathcal{L}_i(x) = \prod_{k=0, k\neq i}^{n} \frac{x - x_k}{x_i - x_k} = \prod_{k=0, k\neq i}^{n} \frac{t - k}{i - k} := \varphi_i(t) \qquad \text{con } 0 \leq i \leq n.$$

Usando l'uguaglianza $\alpha_i = \int_a^b \mathcal{L}_i(x)\mathrm{d}x$ otteniamo che

$$\alpha_i = \int_a^b \mathcal{L}_i(x)\mathrm{d}x = \int_0^n \varphi_i(t)h dt = h \int_0^n \varphi_i(t)\mathrm{d}t$$

da cui si ottiene la formula di quadratura

$$I_n(f) = h \sum_{i=0}^{n} w_i f(x_i), \quad w_i = \int_0^n \varphi_i(t)\mathrm{d}t. \qquad \blacksquare$$

Osservazione. Si può svolgere un ragionamento analogo per le formule di NC aperte, ottenendo:

$$I_n(f) = h \sum_{i=0}^{n} w_i f(x_i), \quad w_i = \int_{-1}^{n+1} \varphi_i(t)\mathrm{d}t.$$

Nelle seguenti tabelle sono visualizzati i pesi delle formule di quadratura di Newton-Cotes chiuse e aperte:

Tabella 5.1: Pesi delle formule di Newton-Cotes

(a) Chiuse

n	1	2	3	4	5	6
w_0	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{3}{8}$	$\frac{14}{45}$	$\frac{95}{288}$	$\frac{41}{140}$
w_1	0	$\frac{4}{3}$	$\frac{9}{8}$	$\frac{64}{45}$	$\frac{375}{288}$	$\frac{216}{140}$
w_2	0	0	0	$\frac{24}{45}$	$\frac{250}{288}$	$\frac{27}{140}$
w_3	0	0	0	0	0	$\frac{272}{140}$

(b) Aperte

n	0	1	2	3	4	5
w_0	2	$\frac{3}{2}$	$\frac{8}{3}$	$\frac{5}{24}$	$\frac{66}{20}$	$\frac{4277}{1440}$
w_1	0	0	$-\frac{4}{3}$	$\frac{5}{24}$	$-\frac{84}{20}$	$-\frac{3171}{1440}$
w_2	0	0	0	0	$\frac{156}{20}$	$\frac{3934}{1440}$

Teorema 5.11 — Errore delle formule di Newton-Cotes. Data una formula di Newton-Cotes con n pari, aperta o chiusa, e data $f \in C^{n+2}([a,b])$, l'errore è dato da:

$$E_n(f) = \frac{M_n}{(n+2)!} h^{n+3} f^{(n+2)}(\xi) \tag{5.3}$$

dove $\xi \in (a,b)$ e

$$M_n = \begin{cases} \int_0^n t\ \pi_{n+1}(t)\mathrm{d}t < 0 & \text{per formule chiuse} \\ \int_{-1}^{n+1} t\ \pi_{n+1}(t)\mathrm{d}t > 0 & \text{per formule aperte} \end{cases}$$

e dove

$$\pi_{n+1}(t) = \prod_{i=0}^{n} (t-i).$$

Dalla (5.3) si deduce che il grado di esattezza è pari a $n+1$ e che l'ordine di accuratezza è h^{n+3}.

Similmente, per n dispari e $f \in C^{n+1}([a,b])$, si ha

$$E_n(f) = \frac{K_n}{(n+1)!} h^{n+2} f^{(n+1)}(\eta), \tag{5.4}$$

con

$$K_n = \begin{cases} \int_0^n \pi_{n+1}(t)\mathrm{d}t < 0 & \text{per formule chiuse} \\ \int_{-1}^{n+1} \pi_{n+1}(t)\mathrm{d}t > 0 & \text{per formule aperte} \end{cases}$$

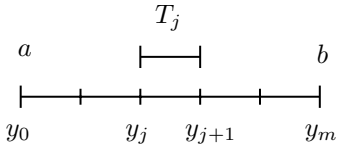
dunque il grado di esattezza è n e l'ordine di accuratezza è h^{n+2}.

Si noti che i pesi delle formule di NC non hanno segno costante, e ciò può essere fonte di errore di cancellazione[3].

Inoltre da $n = 9$ in poi le costanti che entrano nella formula dell'errore tendono a diventare troppo grandi, motivo per cui cercheremo di guadagnare qualcosa in termini grado di esattezza e ordine di accuratezza con le formule composite e con l'integrazione su nodi non equispazioni (Integrazione Gaussiana) nelle sezioni 5.5 e 5.6.

5.5 Formule di Newton-Cotes composite

Dividiamo $[a,b]$ in m sottointervalli $T_j = [y_j, y_{j+1}]$, tali che $y_i = a + jH$, essendo $H = (b-a)/m$, per $j = 0, \ldots, m$.



Si utilizza quindi in ogni sottointervallo una formula interpolatoria avente per nodi i punti $\left\{x_k^{(j)},\ 0 \le k \le n\right\}$ e per pesi i coefficienti $\left\{\alpha_k^{(j)},\ 0 \le k \le n\right\}$. Poiché

$$I(f) = \int_a^b f(x)\mathrm{d}x = \sum_{j=0}^{m-1} \int_{T_j} f(x)\mathrm{d}x = \sum_{j=0}^{m-1} \int_{y_j}^{y_{j+1}} f(x)\mathrm{d}x,$$

[3]L'errore di cancellazione avviene quando il calcolatore somma due numeri molto piccoli e di segno opposto, a causa del fatto che esso può rappresentare solo un numero *finito* di cifre decimali. Per esempio valutando le due forme del polinomio $p(x) = (x-1)^6 = x^6 - 6x^5 + 15x^4 - 20x^3 + 15x^2 - 6x + 1$ su un intervallo molto fine si noteranno delle oscillazioni, anche se il polinomio è lo stesso.

si può ottenere una formula di quadratura interpolatoria composita sostituendo $I(f)$ con

$$I_{n,m}(f) = \sum_{j=0}^{m-1} \sum_{k=0}^{n} \alpha_k^{(j)} f\left(x_k^{(j)}\right).$$

Teorema 5.12 — Errore delle formule di Newton-Cotes composite. Data una formula di Newton-Cotes composita, aperta o chiusa su ogni sottointervallo e con n pari, se $f \in C^{n+2}([a,b])$, si ha, per $\xi \in (a,b)$:

$$E_{n,m}(f) = \frac{b-a}{(n+2)!} \frac{M_n}{\gamma_n^{n+3}} H^{n+2} f^{(n+2)}(\xi).$$

Nel caso in cui n sia dispari, se $f \in C^{n+1}([a,b])$, si ha invece, per $\eta \in (a,b)$:

$$E_{n,m}(f) = \frac{b-a}{(n+1)!} \frac{K_n}{\gamma_n^{n+2}} H^{n+1} f^{(n+1)}(\eta).$$

Nelle due formule è stato definito

$$\gamma_n = \begin{cases} n+2 & \text{per formule aperte} \\ n & \text{per formule chiuse.} \end{cases}$$

5.6 Quadratura su nodi non equispaziati (Integrazione Gaussiana)

Possiamo guadagnare gradi di esattezza e ordini di accuratezza usando nodi non equispaziati con le **formule di quadratura di Gauss-Legendre** (GL) o di **Gauss-Legendre-Lobatto** (GLL). In particolare:

$$\{(x_i, \alpha_i)\}_{i=0}^{n} \quad \rightarrow \quad I_n(f) = \sum_{i=0}^{n} \alpha_i f(x_i) \quad \begin{cases} \text{g.d.e.} = (2n+1) & \text{con GL} \\ \text{g.d.e.} = (2n-1) & \text{con GLL.} \end{cases}$$

5.6.1 Polinomi di Legendre

Su $I = [-1,1]$ consideriamo un insieme di polinomi, detti **polinomi di Legendre**:

$$\{L_0(x), L_1(x), \ldots, L_n(x)\} \quad \text{con} \quad n \geq 0.$$

Ciascun $L_i(x)$ è un polinomio di grado i ed è tale che

$$\int_{-1}^{1} L_i(x) L_j(x) \mathrm{d}x = 0, \quad \forall j = 0, \ldots, i-1,$$

ossia essi sono ortogonali a due a due tra loro. Allora si può dimostrare che questa è una base per lo spazio dei polinomi di grado n su I:

$$\mathbb{P}^n(I) = \text{span}\{L_0(x), L_1(x), \dots, L_n(x)\}.$$

Presentiamo una formula ricorsiva per il calcolo dei polinomi di Legendre:

$$\begin{cases} L_{k+1} = \frac{2k+1}{k+1} x L_k(x) - \frac{k}{k+1} L_{k-1}(x) \quad \text{con} \quad k = 1, 2, \dots \\ L_0(x) = 1 \\ L_1(x) = x. \end{cases}$$

5.6.2 Nodi e pesi di Gauss-Legendre (GL)

Per $n \geq 0$, i **nodi di Gauss-Legendre** ed i relativi pesi sono dati da:

$$x_i \text{ zeri di } L_{n+1}(x), \quad \alpha_i = \frac{2}{(1-x_i)^2 [L'_{n+1}(x_i)]^2}, \quad i = 0, \dots, n.$$

Osservazioni.

- Nei nodi x_i sono esclusi gli estremi dell'intervallo $(-1, 1)$.
- Il grado di esattezza della formula è $2n + 1$, che si può dimostrare essere il massimo raggiungibile.

Esempio. Integriamo su $(-1, 1)$, con $n = 1$. Abbiamo 2 nodi, i cui pesi, si può verificare, sono $\alpha_i = \{1, 1\}$ mentre i nodi sono $x_i = \{\pm 1/\sqrt{3}\}$. Consideriamo una $f(x), x \in [-1, 1]$.

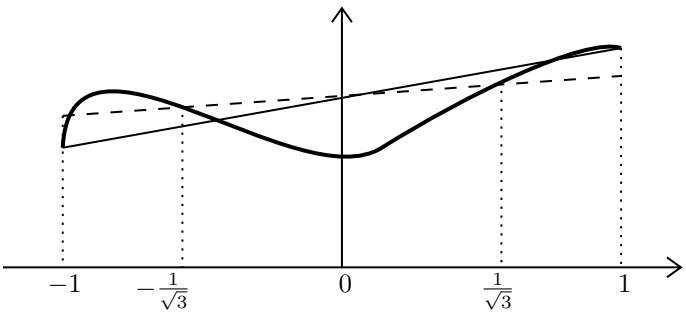


Figura 5.6: La linea spessa denota la $f(x)$, la linea sottile continua denota l'approssimazione che si ottiene con la formula del trapezio, mentre la linea sottile tratteggiata denota l'approssimazione con Gauss-Lobatto.

La formula del trapezio integra in modo esatto fino a polinomi di grado 1, mentre la **formula di Gauss-Lobatto** integra in modo esatto fino a grado 3.

$$I_1^{\text{GL}}(f) = \sum_{i=0}^{2} \alpha_i f(x_i) = 1 \cdot f\left(-\frac{1}{\sqrt{3}}\right) + 1 \cdot f\left(\frac{1}{\sqrt{3}}\right)$$

$$I_1^{\text{trap}}(f) = (b-a)\left[\frac{f(a)+f(b)}{2}\right] = f(-1)+f(1)$$

5.6.3 Nodi e pesi di Gauss-Legendre-Lobatto (GLL)

Per $n \geq 0$, i **nodi di Gauss-Legendre-Lobatto** ed i relativi pesi sono dati da:

$$x_0 = -1, \quad \{x_i \text{ zeri di } L_n'(x), \quad i = 1, \ldots, n-1\}, \quad x_n = 1,$$
$$\alpha_i = \frac{2}{n(n+1)}\frac{1}{[L_n(x_i)]^2}, \quad i = 0, \ldots, n.$$

Osservazioni.

- I nodi x_i comprendono gli estremi.
- Il grado di esattezza della formula è $2n-1$.

5.6.4 Errore delle formule di GL e GLL

Data f sufficientemente regolare, vale la seguente rappresentazione dell'errore:

$$|E_n(f)| \leq \frac{C}{n^5}\|f\|_s$$

dove

$$\|f\|_s = \left(\sum_{k=0}^{s}\left\|f^{(k)}\right\|^2_{L^2(-1,1)}\right)^{1/2}$$

nel quale

$$\|f\|_{L^2(-1,1)} = \left[\int_{-1}^{1}[f(x)]^2 \mathrm{d}x\right]^{1/2}.$$

Per concludere il capitolo, presentiamo nella tabella 5.2 un riassunto dei principali risultati di accuratezza ed esattezza delle formule viste.

	grado di esattezza	ordine di accuratezza
Punto medio semplice	1	
Trapezio semplice	1	
CS semplice	3	
n pari	$n+1$	
n dispari	n	
Punto medio composito	1	2
Trapezio composito	1	2
CS composito	3	4
NC, n pari	$n+1$	$n+3$
NC, n dispari	n	$n+2$
NC composite, n pari	$n+1$	$n+2$
NC composite, n dispari	n	$n+1$
GL	$2n+1$	
GLL	$2n-1$	

Tabella 5.2: Tabella riassuntiva dei gradi di esattezza e ordini di accuratezza dei metodi numerici analizzati.

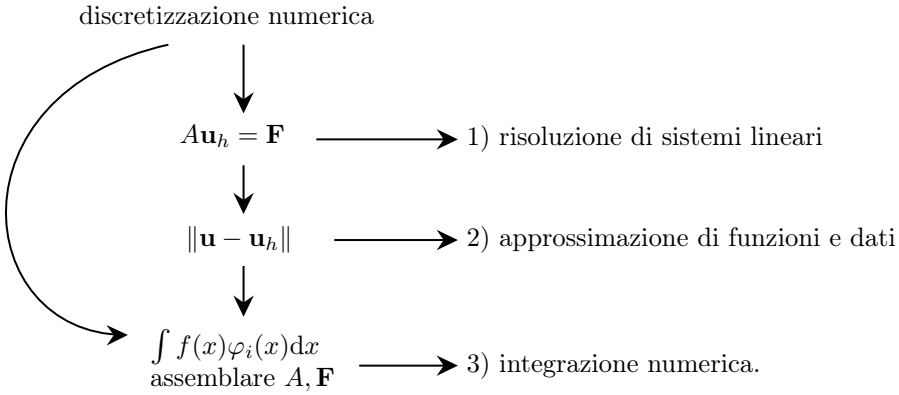
Capitolo 6

Approssimazione di derivate

Negli scorsi capitoli siamo partiti da un problema stazionario, ovvero non dipendente dal tempo o da altre variabili indipendenti:

$$\begin{cases} -u''(x) = f(x) \\ u(0) = 0 \\ u(1) = 0 \end{cases} \qquad x \in (0,1)$$

seguendo una serie di passaggi:



Aggiungiamo un ulteriore grado di complessità rispetto al problema proposto all'inizio del capitolo 1, cioè la variazione nel tempo oltre che nello spazio. Invece di avere una corda fissata agli estremi, osserviamo una barra di metallo che riscaldiamo. Vogliamo modellizzare come varia la tempertura $u(x,t)$ in funzione dello spazio x e del tempo t, assegnate la temperatura agli estremi, la temperatura all'istante iniziale e la sorgente termica.

Si può mostrare (ma non rientra nella nostra trattazione) che questo problema è modellato dall'*equazione del calore*, abbinata ad opportune condizioni:

$$\begin{cases} u_t(x,t) - u_{xx}(x,t) = f(x,t) & x \in (0,1), \forall t \quad \text{(equazione del calore)} \\ u(0,t) = 0, \quad u(1,t) = 0 & \forall t > 0 \quad \text{(condizioni ai limiti)} \\ u(x,0) = u_0(x) & \forall x \in (a,b) \quad \text{(condizione iniziale).} \end{cases} \tag{PM}$$

La funzione cercata $u(x,t)$ è la soluzione dell'equazione del calore. Questa è un esempio di *equazione differenziale*, che approfondiremo nel capitolo 7. La risoluzione di questo tipo di equazione è uno dei principali motivi di interesse per l'approssimazione numerica di derivate.

Riscriviamo (PM) nel seguente modo, moltiplicando per una certa funzione $v = v(x)$ e integrando in $(0,1)$:

$$\int_0^1 (u_t v - u_{xx} v)\mathrm{d}x = \int_0^1 f v \mathrm{d}x.$$

Integriamo poi per parti il primo membro:

$$\int_0^1 u_t v \mathrm{d}x + \int_0^1 u_x v_x \mathrm{d}x - [u_x v]_0^1 = \int_0^1 f v \mathrm{d}x.$$

Come nel caso già affrontato in sezione 1.1.2, v è una funzione, in uno spazio ancora non definito, tale che $v(0) = 0$, $v(1) = 0$. Si ha dunque:

$$[u_x v]_0^1 = u_x(1,t)v(1) - u_x(0,t)v(0) = 0.$$

Riscriviamo quindi il problema (PM). $\forall t > 0$, trovare $u(x,t) \in V$ tale che:

$$\begin{cases} \int_0^1 u_t(x,t)v(x)\mathrm{d}x + \int_0^1 u_x(x,t)v_x(x)\mathrm{d}x = \int_0^1 f(x,t)v(x)\mathrm{d}x, \quad \forall v \in V, \\ u(x,0) = u_0(x), \end{cases} \tag{PM'}$$

dove

$$V = \{v : (0,1) \to \mathbb{R} \text{ tale che } v, v_x \in L^2(0,1),\ v(0) = 0,\ v(1) = 0\}$$

è uno spazio infinito-dimensionale. Vogliamo utilizzare invece uno spazio finito-dimensionale $V_h \subseteq V$, cioè tale che $\dim(V_h) = N_h < +\infty$. Riscriviamo (PM) in V_h. $\forall t > 0$ trovare $u_h(x,t) \in V_h$ tale che:

$$\begin{cases} \int_0^1 \frac{\partial u_h(x,t)}{\partial t} v_h(x)\mathrm{d}x + \int_0^1 \frac{\partial u_h(x,t)}{\partial x} \frac{\partial v_h(x)}{\partial x}\mathrm{d}x = \int_0^1 f(x,t)v_h(x)\mathrm{d}x, \quad \forall v_h \in V_h, \\ u_h(x,0) = u_0^h(x), \end{cases} \tag{PN}$$

dove $u_0^h(x) \in V_h$ è un'approssimazione di $u_0(x)$.

Proprietà. (PM) è un sistema di equazioni differenziali del primo ordine della forma:

$$\begin{cases} M\frac{\partial \mathbf{u}_h(t)}{\partial t} + A\mathbf{u}_h(t) = \mathbf{F}(t), \quad \forall t > 0, \\ \mathbf{u}_h(0) = \mathbf{u}_0^h. \end{cases} \tag{6.1}$$

Dimostrazione. Costruiamo una base per V_h

$$V_h = \text{span}\{\varphi_1(x), \varphi_2(x), \dots, \varphi_{N_h}(x)\},$$

allora la soluzione $u_h(x,t)$ può essere scritta come combinazione lineare delle funzioni di base:

$$u_h(x,t) = \sum_{j=1}^{N_h} u_j(t)\varphi_j(x). \tag{6.2}$$

La dipendenza dal tempo è inclusa nei coefficienti $u_j(t)$. Abbiamo pertanto costruito il vettore

$$\mathbf{u}_h(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_{N_h}(t) \end{bmatrix} \in \mathbb{R}^{N_h}.$$

Riscriviamo (PN) utilizzando (6.2) e otteniamo

$$\int_0^1 \frac{\partial}{\partial t}\left(\sum_{j=1}^{N_h} u_j(t)\varphi_j(x)\right)\varphi_i(x)\mathrm{d}x + \int_0^1 \frac{\partial}{\partial x}\left[\sum_{j=1}^{N_h} u_j(t)\varphi_j(x)\right]\frac{\partial}{\partial x}\varphi_i(x)\mathrm{d}x = \\ = \int_0^1 f(x,t)\varphi_i(x)\mathrm{d}x, \quad \forall i = 1, \dots, N_h.$$

Usando la linearità e uno scambio tra derivata e integrale (possibile perché le due variabili coinvolte sono distinte), possiamo riorganizzare per ottenere:

$$\sum_{j=1}^{N_h} \frac{\partial}{\partial t}u_j(t)\int_0^1 \varphi_j(x)\varphi_i(x)\mathrm{d}x + \sum_{j=1}^{N_h} u_j(t)\int_0^1 \frac{\partial}{\partial x}\varphi_j(x)\frac{\partial}{\partial x}\varphi_i(x)\mathrm{d}x = \\ = \int_0^1 f(x,t)\varphi_i(x)\mathrm{d}x, \quad \forall i = 1, \dots, N_h.$$

Definiamo ora la **matrice di massa**

$$M \in \mathbb{R}^{N_h \times N_h}, \quad M_{ij} \coloneqq \int_0^1 \varphi_j(x)\varphi_i(x)\mathrm{d}x, \quad \forall i,j = 1, \dots, N_h,$$

e la matrice **matrice di rigidezza** (o *stiffness matrix*)

$$A \in \mathbb{R}^{N_h \times N_h}, \quad A_{ij} \coloneqq \int_0^1 \frac{\partial}{\partial x}\varphi_j(x)\frac{\partial}{\partial x}\varphi_i(x)\mathrm{d}x, \quad \forall i,j = 1, \dots, N_h,$$

quest'ultima è la stessa dell'equazione (1.2) presentata a inizio corso. Infine definiamo il vettore

$$\mathbf{F} \in \mathbb{R}^{N_h}, \quad F_i = \int_0^1 f(x,t)\varphi_i(x)\mathrm{d}x, \quad \forall i = 1, \dots, N_h.$$

Con queste definizioni possiamo riscrivere

$$\begin{cases} M\frac{\partial \mathbf{u}_h}{\partial t} + A\mathbf{u}_h(t) = \mathbf{F}(t), & \forall t > 0, \\ \mathbf{u}_h(0) = \mathbf{u}_0^h. \end{cases} \tag{EDO}$$

Per procedere alla risoluzione di questo *problema non stazionario* avremo bisogno di far risolvere al calcolatore:

1. sistemi di EDO (Equazioni Differenziali Ordinarie),
2. sistemi di Equazioni Non Lineari.

6.1 Approssimazione di derivate

Sia $f : [a,b] \to \mathbb{R}$ derivabile con continuità e sia $\overline{x} \in (a,b)$. Vogliamo approssimare $f'(\overline{x})$. Per definizione la derivata di f in $\overline{x}$ è

$$f'(\overline{x}) \coloneqq \lim_{h\to 0} \frac{f(\overline{x}+h) - f(\overline{x})}{h},$$

ma a livello numerico l'operazione di limite è ovviamente impossibile, e dovrà essere approssimata con h piccoli. Definiamo quindi la **differenza finita in avanti**:

$$f'(\overline{x}) \approx (\delta_+ f)(\overline{x}) := \frac{f(\overline{x}+h) - f(\overline{x})}{h}. \tag{6.3}$$

La **differenza finita all'indietro**:

$$f'(\overline{x}) \approx (\delta_- f)(\overline{x}) := \frac{f(\overline{x}) - f(\overline{x}-h)}{h}. \tag{6.4}$$

Infine, la media tra le due è detta **differenza finita centrata**:

$$f'(\overline{x}) \approx (\delta f)(\overline{x}) := \frac{f(\overline{x}+h) - f(\overline{x}-h)}{2h}. \tag{6.5}$$

6.1.1 Errore di approssimazione delle derivate

Supponiamo $f \in C^2((a,b))$. Possiamo sviluppare f in serie di Taylor:

$$f(\overline{x}+h) = f(\overline{x}) + hf'(\overline{x}) + \frac{h^2}{2}f''(\xi), \quad \xi \in (\overline{x}, \overline{x}+h). \tag{6.6}$$

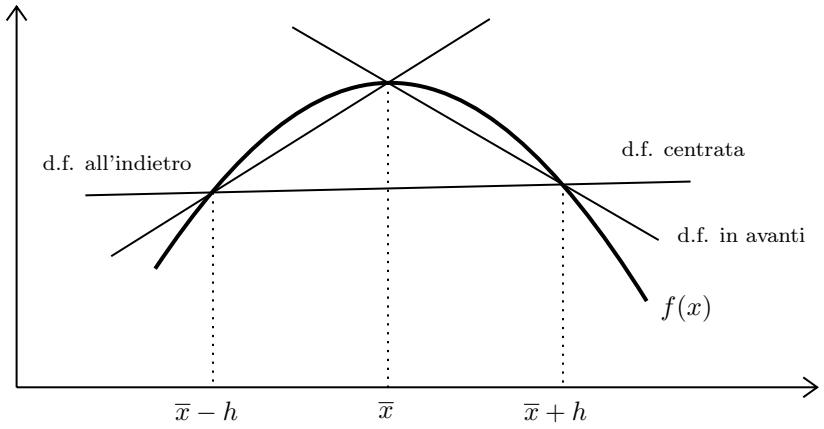


Figura 6.1: Intuizione grafica delle differenze all'indietro, centrate e in avanti con passo h.

Analogamente, scrivendo l'espressione in $\overline{x}-h$:

$$f(\overline{x}-h) = f(\overline{x}) - hf'(\overline{x}) + \frac{h^2}{2}f''(\eta), \quad \eta \in (\overline{x}-h, \overline{x}). \tag{6.7}$$

Possiamo riscrivere (6.6) ottenendo l'errore di approssimazione:

$$\begin{aligned} \underbrace{\frac{f(\overline{x}+h)-f(\overline{x})}{h}}_{(\delta_+ f)(\overline{x})} - f'(\overline{x}) &= \frac{h}{2}f''(\xi) \\ (\delta_+ f)(\overline{x}) - f'(\overline{x}) &= \frac{h}{2}f''(\xi) \\ E^+(\overline{x}) &= \frac{h}{2}f''(\xi) \end{aligned}$$

e limitare superiormente questo errore:

$$\left|E^+(\overline{x})\right| \leq \frac{h}{2}|f''(\xi)| \leq \frac{h}{2}\underbrace{\|f''(x)\|_\infty}_{\leq C}.$$

Analogamente per (6.7):

$$\begin{aligned} \underbrace{\frac{f(\overline{x})-f(\overline{x}-h)}{h}}_{(\delta_- f)(\overline{x})} - f'(\overline{x}) &= -\frac{h}{2}f''(\xi) \\ (\delta_- f)(\overline{x}) - f'(\overline{x}) &= -\frac{h}{2}f''(\xi) \end{aligned}$$

$$E^-(\overline{x}) = -\frac{h}{2}f''(\xi)$$
$$\Rightarrow \quad \left|E^-(\overline{x})\right| \leq \frac{h}{2}|f''(\xi)| \leq \frac{h}{2}\underbrace{\|f''(x)\|_\infty}_{\leq C}.$$

L'errore è quindi un infinitesimo di ordine 1 rispetto ad h.

Per stimare l'errore della differenza finita centrata, supponiamo $f \in C^3((a,b))$ e sviluppiamo f in serie di Taylor fino all'ordine 3:

$$f(\overline{x}+h) = f(\overline{x}) + hf'(\overline{x}) + \frac{h^2}{2}f''(\overline{x}) + \frac{h^3}{6}f'''(\xi), \quad \xi \in (\overline{x}, \overline{x}+h) \tag{6.8}$$
$$f(\overline{x}-h) = f(\overline{x}) - hf'(\overline{x}) + \frac{h^2}{2}f''(\overline{x}) - \frac{h^3}{6}f'''(\eta), \quad \eta \in (\overline{x}-h, \overline{x}) \tag{6.9}$$

sottraiamo (6.8) e (6.9) membro a membro:

$$f(\overline{x}+h) - f(\overline{x}-h) = 2hf'(\overline{x}) + \frac{h^3}{6}[f'''(\xi) + f'''(\eta)].$$

Dividendo per $2h$:

$$\underbrace{\frac{f(\overline{x}+h) - f(\overline{x}-h)}{2h}}_{(\delta f)(\overline{x})} - f'(\overline{x}) = \frac{h^2}{12}[f'''(\xi) + f'''(\eta)]$$
$$(\delta f)(\overline{x}) - f'(\overline{x}) = \frac{h^2}{12}[f'''(\xi) + f'''(\eta)]$$
$$E(\overline{x}) = \frac{h^2}{12}[f'''(\xi) + f'''(\eta)].$$

Prendendo la norma infinito:

$$\|f'''(\xi)\|_\infty \leq \max_{x\in(a,b)} |f'''(x)| =: \|f'''(x)\|_\infty,$$
$$\|f'''(\eta)\|_\infty \leq \max_{x\in(a,b)} |f'''(x)| =: \|f'''(x)\|_\infty,$$
$$\Rightarrow |E(\overline{x})| \leq \frac{h^2}{6}\|f'''(x)\|_\infty.$$

Abbiamo guadagnato un ulteriore ordine di infinitesimo con le differenze centrate. Più l'ordine di infinitesimo è alto, migliore è l'approssimazione, poiché $h \ll 1$ e quindi tende a zero aumentando il suo esponente.

Osservazione.

Supponiamo si voglia approssimare

$$f'(x_i), \quad i = 0, \ldots, n, \quad x_i = a + ih.$$

- Per approssimarla nei punti (nodi) interni $x_1 \dots x_{n-1}$ possiamo usare le differenze finite in avanti, all'indietro o centrate, a piacere.
- Per approssimarla in x_0 possiamo usare solo le differenze finite in avanti:

$$f'(x_0) \approx \frac{f(x_1) - f(x_2)}{h}.$$

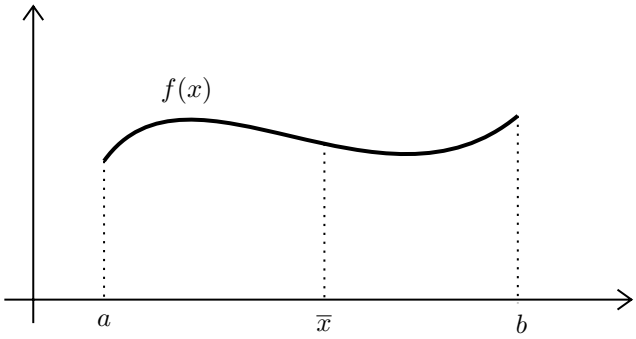
- Per approssimarla in in x_n possiamo usare solo le differenze finite all'indietro:

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{h}.$$

Si ha quindi meno accuratezza agli estremi. Introducendo le **differenze finite generalizzate**, facendo riferimento a sviluppi di Taylor di ordine superiore, è possibile raggiungere l'ordine di infinitesimo desiderato.

6.2 Approssimazione della derivata seconda

Sia $f \in C^2([a,b])$. Vogliamo approssimare $f''(\overline{x})$, $x \in (a,b)$, utilizzando, come prima, gli sviluppi di Taylor.



Fissiamo $h > 0$ e scriviamo, supponendo $f \in C^4([a,b])$:

$$f(\overline{x}+h) = f(\overline{x}) + hf'(\overline{x}) + \frac{h^2}{2}f''(\overline{x}) + \frac{h^3}{6}f'''(\overline{x}) + \frac{h^4}{24}f^{(\mathrm{iv})}(\xi)$$
$$f(\overline{x}-h) = f(\overline{x}) - hf'(\overline{x}) + \frac{h^2}{2}f''(\overline{x}) - \frac{h^3}{6}f'''(\overline{x}) + \frac{h^4}{24}f^{(\mathrm{iv})}(\eta).$$

con $\xi \in (\overline{x}, \overline{x}+h)$ e $\eta \in (\overline{x}-h, \overline{x})$. Sommiamo ora membro a membro:

$$f(\overline{x}+h) + f(\overline{x}-h) = 2f(\overline{x}) + h^2 f''(\overline{x}) + \frac{h^4}{24}\left[f^{(\mathrm{iv})}(\xi) + f^{(\mathrm{iv})}(\eta)\right].$$

Isolando $f''(\overline{x})$:

$$f''(\overline{x}) = \frac{f(\overline{x}+h) + f(\overline{x}-h) - 2f(\overline{x})}{h^2} - \frac{h^2}{24}\left[f^{(\mathrm{iv})}(\xi) + f^{(\mathrm{iv})}(\eta)\right].$$

Definiamo l'approssimazione di f'' troncando gli infinitesimi di ordine superiore ad 1:

$$f''(\overline{x}) \approx \frac{f(\overline{x}+h) + f(\overline{x}-h) - 2f(\overline{x})}{h^2}.$$

L'errore di approssimazione è quindi:

$$E(\overline{x}) = -\frac{h^2}{24}\left[f^{(\mathrm{iv})}(\xi) + f^{(\mathrm{iv})}(\eta)\right].$$

Passando alla norma infinito, otteniamo:

$$|E(\overline{x})| \leq -\frac{h^2}{12}\left\|f^{(\mathrm{iv})}(x)\right\|_\infty.$$

Si noti che questa espressione vale solo per i nodi interni.

Capitolo 7

Risoluzione di Equazioni Differenziali Ordinarie

In questo capitolo ci dedicheremo alle EDO, **equazioni differenziali ordinarie**, ovvero equazioni dalla forma generale:

$$F\left(x, y, y', y'', \cdots, y^{(n)}\right) = 0,$$

dove quindi sono coinvolte le derivate di y, la funzione incognita da determinare, ed eventualmente la variabile x, cioè l'argomento della y.

Queste equazioni sono importantissime, perché consentono di descrivere in modo molto accurato tantissimi fenomeni fisici, anche se in realtà questi si descrivono più propriamente con le EDP, **equazioni alle derivate parziali**. Nelle EDP non c'è dipendenza da una sola variabile, ma da più variabili, quindi naturalmente si potrà derivare la funzione incognita rispetto a ciascuna delle variabili, eventualmente più volte. Tipicamente quando la dipendenza è da una sola variabile essa si può interpretare come un tempo o uno spazio. Quando però si vuole studiare l'evoluzione nel tempo di un sistema in 3D è chiaro che servono $3 + 1$ variabili, quindi una EDP. Un aspetto interessante è che a volte la risoluzione delle EDP può essere ridotta a una EDO, per cui anche lo studio che affronteremo adesso sarà fondamentale per la modellistica fisica e le simulazioni numeriche.

Esempio. Supponiamo di avere una popolazione di batteri in un ambiente limitato nel quale non possono convivere più di B batteri contemporaneamente. Supponiamo inoltre che all'istante in cui inizia l'esperimento, il numero di batteri sia $y_0 \ll B$ e che il fattore di crescita dei batteri sia pari a una costante $r > 0$. Come cambia nel tempo il numero di batteri?

Per scrivere un modello matematico di questo problema, dobbiamo anzitutto definire la quantità di interesse: sia $y(t)$ il numero di batteri all'istante t.

I dati sono:

- $y(0) = y_0$, la quantità di batteri all'istante iniziale $t = 0$;
- r, il fattore di crescita;
- B, il numero massimo di batteri che possono coesistere.

Supponiamo $y \in C^1$: se a un certo punto $y(t) = B$, la derivata di $y(t)$ si deve annullare in quel punto.

Scriviamo ora il modello:

$$\begin{cases} \frac{\mathrm{d}y(t)}{\mathrm{d}t} = r\ y(t)\left(1 - \frac{y(t)}{B}\right) \\ y(0) = y_0. \end{cases}$$

Esso è un esempio di **equazione funzionale**, in quanto i suoi elementi, inclusa l'incognita $y(t)$, sono funzioni e non quantità algebriche. In particolare, è un'*equazione differenziale ordinaria* (EDO), poiché presenta operazioni di derivazione. *Ordinaria* significa che non presenta derivate parziali. L'*ordine* di tale equazione è invece il più alto grado di derivazione presente. In particolare, ci concentreremo sulle equazioni differenziali di primo ordine.

Supponiamo ora di avere due popolazioni di batteri distinte e in competizione tra loro. Indichiamo con $y_1(t)$ e $y_2(t)$ il numero di batteri rispettivamente della prima e della seconda popolazione. In questo caso il modello matematico diventa:

$$\begin{cases} \frac{dy_1}{\mathrm{d}t} = r_1 y_1(1 - b_1 y_1 - d_2 y_2) \\ \frac{dy_2}{\mathrm{d}t} = r_2 y_2(1 - b_2 y_2 - d_1 y_1) \\ y_1(0) = y_1^0 \\ y_2(0) = y_2^0 \end{cases}$$

dove

- r_1, r_2 sono i fattori di crescita delle due popolazioni
- b_1, b_2 sono fattori legati alla disponibilità di nutrienti, e quindi alla sopravvivenza delle singole popolazioni
- d_1, d_2 sono fattori che governano le interazioni, cioè la competizione, tra le due popolazioni.

La soluzione è della forma

$$\mathbf{y}(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix}.$$

7.1 Problema di Cauchy

In generale, il problema matematico modello che vogliamo studiare si può così formulare: Sia $I = [t_0, t_0 + T]$ e $f : I \times \mathbb{R} \to \mathbb{R}$, vogliamo trovare $y(t) : I \to \mathbb{R}$ tale

che

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(t_0) = y_0 \end{cases} \qquad t \in I \tag{PC}$$

dove $y_0 \in \mathbb{R}$ è assegnato.

Osservazione. Se la funzione f è continua rispetto a t, possiamo integrare la prima equazione di (PC) tra t_0 e t ottenendo la seguente formulazione:

$$y(t) = y_0 + \int_{t_0}^{t} f(\tau, y(\tau))\mathrm{d}\tau.$$

Ci interessa conoscere:

1. se (PC) ammette una sola soluzione;
2. la regolarità di questa soluzione;
3. se la soluzione sia stabile rispetto ai dati, ovvero se dipenda con continuità da essi.

Definizione 7.1 — Funzione lipschitziana. Una funzione $g : I \to \mathbb{R}$ è detta lipschitziana se esiste una costante L tale che

$$|g(x) - g(y)| \leq L|x - y|, \quad \forall x, y \in I.$$

Inoltre la costante migliore, cioè la più bassa possibile che rispetti la disuguaglianza, viene detta costante di Lipschitz.

Teorema 7.2 — Esistenza della soluzione del (PC)**.** Supponiamo che la funzione $f(\cdot, \cdot)$ sia continua rispetto a entrambe le variabili, e che sia lipschitziana rispetto al secondo argomento. Allora la soluzione del Problema di Cauchy (PC) esiste ed è unica, e inoltre $y \in C^1(I)$.

Il teorema risponde ai primi due punti. Al fine di studiare il terzo punto, ovvero la stabilità di (PC), consideriamo il seguente problema perturbato:

$$(\widetilde{\text{PC}}) \quad \begin{cases} z'(t) = f(t, z(t)) + \delta(t) \\ z(0) = y_0 + \delta_0 \end{cases} \qquad t \in I$$

dove $\delta_0 \in \mathbb{R}$ e $\delta(t)$ è una funzione continua in I. Vogliamo caratterizzare la sensibilità di $z(t)$ alle perturbazioni δ_0 e $\delta(t)$. Questo è il concetto di stabilità secondo Liapunov.

Definizione 7.3 — Stabilità secondo Liapunov. Sia $I = [t_0, t_0 + T]$ un intervallo limitato ($T < +\infty$). Diciamo che (PC) è stabile secondo Liapunov se

per ogni perturbazione $(\delta_0, \delta(t))$ tale che

$$|\delta_0| < \varepsilon, \quad |\delta(t)| < \varepsilon, \quad \forall t \in I, \quad \varepsilon > 0,$$

con ε sufficientemente piccolo da garantire che $(\widetilde{\text{PC}})$ ammetta una e una sola soluzione, allora esiste $c > 0$, indipendente da ε, tale che

$$|y(t) - z(t)| < c\,\varepsilon, \quad \forall t \in I. \tag{7.1}$$

Definizione 7.4 — Stabilità asintotica. Sia I un intervallo superiormente illimitato. Si dice che (PC) è asintoticamente stabile se vale (7.1) e inoltre

$$\lim_{t\to\infty} |y(t) - z(t)| = 0$$

purché $|\delta(t)| \to 0$ per $t \to \infty$.

Enunciamo ora un lemma utile nella dimostrazione di risultati di stabilità.

Lemma 7.5 — Gronwall. Sia p una funzione integrabile e non negativa sull'intervallo $[t_0, t_0 + T]$ e siano g e φ due funzioni continue su $[t_0, t_0 + T]$, con g non decrescente. Allora

$$\varphi(t) \le g(t) + \int_{t_0}^{t} p(s)\varphi(s)\mathrm{d}s \quad \Rightarrow \quad \varphi(t) \le g(t)\exp\left(\int_{t_0}^{t} p(s)\mathrm{d}s\right).$$

Teorema 7.6 — Stabilità. Sia $f(\cdot,\cdot)$ lipschitziana rispetto al secondo argomento. Allora (PC) è asintoticamente stabile.

Dimostrazione. Sia $y(t)$ la soluzione di (PC) e sia $z(t)$ la soluzione di $(\widetilde{\text{PC}})$. Definiamo $w(t) = z(t) - y(t)$. Si ha, con $t \in I$:

$$\begin{cases} w'(t) = f(t, z(t)) + \delta(t) - f(t, y(t)) \\ w(t_0) = \cancel{y_0} + \delta_0 - \cancel{y_0}. \end{cases}$$

Integriamo entrambi i membri della prima equazione tra t_0 e t:

$$\begin{aligned} \int_{t_0}^{t} w'(s)\mathrm{d}s &= \int_{t_0}^{t} [f(s, z(s)) + \delta(s) - f(s, y(s))]\mathrm{d}s \\ w(t) - w(t_0) &= \int_{t_0}^{t} [f(s, z(s)) - f(s, y(s))]\mathrm{d}s + \int_{t_0}^{t} \delta(s)\mathrm{d}s \\ w(t) &= \delta_0 + \int_{t_0}^{t} [f(s, z(s)) - f(s, y(s))]\mathrm{d}s + \int_{t_0}^{t} \delta(s)\mathrm{d}s \end{aligned}$$

Passiamo al modulo e otteniamo:

$$|w(t)| \leq |\delta_0| + \int_{t_0}^{t} |f(s, z(s)) - f(s, y(s))| \mathrm{d}s + \int_{t_0}^{t} |\delta(s)| \mathrm{d}s.$$

Ricordando che f è lipschitziana per ipotesi:

$$|w(t)| \leq |\delta_0| + \int_{t_0}^{t} L|z(s) - y(s)| \mathrm{d}s + \int_{t_0}^{t} |\delta(s)| \mathrm{d}s.$$

Considerando che per definizione $z(s) - y(s) =: w(s)$:

$$|w(t)| \leq |\delta_0| + L \int_{t_0}^{t} |w(s)| \mathrm{d}s + \int_{t_0}^{t} |\delta(s)| \mathrm{d}s,$$

quindi se $|\delta_0| < \varepsilon$ e $|\delta(s)| < \varepsilon$, $\forall t \in I$, possiamo riscriverlo come

$$\begin{aligned} |w(t)| &\leq \varepsilon + L \int_{t_0}^{t} |w(s)| \mathrm{d}s + \int_{t_0}^{t} \varepsilon \mathrm{d}s \\ &\leq \varepsilon(1 + |t - t_0|) + L \int_{t_0}^{t} |w(s)| \mathrm{d}s. \end{aligned}$$

Usando il lemma di Gronwall con $g(t) = \varepsilon(1 + |t - t_0|)$ e $p(t) = L$, si ottiene:

$$\begin{aligned} |w(t)| &\leq \varepsilon(1 + |t - t_0|) \exp\left(\int_{t_0}^{t} L \mathrm{d}s \right) \\ &\leq \varepsilon(1 + |t - t_0|) e^{L|t - t_0|}. \end{aligned}$$

Poiché $|t - t_0| \leq T$, si ha:

$$|w(t)| \leq \varepsilon(1 + T) e^{LT}.$$

Definendo $c \coloneqq (1 + T)e^{LT}$, allora:

$$|w(t)| \leq c\, \varepsilon.$$

■

Teorema 7.7 — Buona posizione. Sotto le ipotesi del teorema 7.2 il (PC) è ben posto, cioè esiste un'unica soluzione $y(t) \in C^1(I)$ che dipende con continuità dai dati, ovvero è stabile secondo Liapunov.

Osservazioni.

- Dalla dimostrazione del teorema 7.6, abbiamo mostrato che la costante di stabilità
$$c = (1 + T)e^{LT}$$
cresce esponenzialmente all'aumentare di T, l'ampiezza dell'intervallo, ovvero il tempo di osservazione.
- Per costruire i metodi numerici lavoreremo solo con (PC) che sono ben posti.

7.2 Metodi numerici a un passo

Analizziamo ora quattro esempi di **metodi ad un passo**. Ciò significa che, per calcolare la derivata a un certo t_{n+1} utilizziamo solo informazioni che dipendono dal passo t_n. Esistono anche metodi a più passi, o multistep, dove sfruttiamo più informazioni precedenti anziché solo quella immediatamente precedente. Essi saranno l'oggetto della sezione 7.5.

Sia $I = (t_0, t_0 + T)$, con $0 < T < +\infty$, e sia data la funzione $f(\cdot, \cdot) : I \times \mathbb{R} \to \mathbb{R}$. Ricordiamo la formulazione del modello di interesse, il problema di Cauchy:

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(t_0) = y_0 \end{cases} \qquad t \in I. \tag{PC}$$

Per risolvere numericamente il problema, seguiremo la seguente procedura:

1. Discretizziamo l'intervallo temporale I: fissiamo il passo di integrazione temporale $h > 0$ e costruiamo i nodi di discretizzazione

   $$t_n = t_0 + nh, \quad n = 0, 1, \ldots, N_h$$

   dove N_h è il massimo intero tale che $t_{N_h} = t_0 + N_h h \leq t_0 + T$.

2. Costruiamo un'approssimazione di $y(t_n)$ per ogni nodo t_n. Anzitutto, (PC) soddisfa

   $$\begin{cases} y'(t_n) = f(t_n, y(t_n)) \\ y(t_0) = y_0 \end{cases} \qquad t \in I.$$

   Definiamo

   $$u_n \approx y(t_n), \quad n = 0, 1, \ldots, N_h$$

   e riscriviamo il (PC) approssimando numericamente la derivata. Il modo che utilizziamo per approssimarla dà origine a diversi metodi numerici.

7.2.1 Metodo di Eulero Esplicito

Analizziamo il metodo di **Eulero esplicito** (o **Eulero in avanti**), che abbrevieremo con (EE). Approssimando la derivata con la formula delle differenze finite in avanti (vedi (6.3)) otteniamo:

$$\frac{u_{n+1} - u_n}{h} = f(t_n, u_n), \quad n = 0, 1, \ldots, N_h - 1.$$

Dato che $u_0 = y(t_0) = y_0$, calcoliamo gli u_n nel seguente modo:

$$u_0 \to u_1 = u_0 + hf(t_0, u_0)$$
$$u_1 \to u_2 = u_1 + hf(t_1, u_1)$$

$$\vdots$$
$$u_n \to u_{n+1} = u_n + hf(t_n, u_n).$$

Pertanto otteniamo, per $n = 0, 1, \ldots, N_h - 1$:

$$\begin{cases} u_{n+1} = u_n + hf(t_n, u_n) \\ u_0 = y_0. \end{cases} \tag{EE}$$

7.2.2 Metodo di Eulero Implicito

Analizziamo il metodo di **Eulero implicito** (o **Eulero all'indietro**), che abbrevieremo con (EI). Approssimiamo ora la derivata con le differenze finite all'indietro (vedi (6.4)). Con conti analoghi troviamo, per $n = 0, 1, \ldots, N_h - 1$:

$$\begin{cases} u_{n+1} = u_n + hf(t_{n+1}, u_{n+1}) \\ u_0 = y_0. \end{cases} \tag{EI}$$

NB. Per calcolare u_{n+1} ad ogni passo bisognerà quindi risolvere un'equazione non lineare, a meno che $f(\cdot, \cdot)$ non sia lineare nel secondo argomento.

7.2.3 Metodo di Crank–Nicolson

Analizziamo il metodo di **Crank-Nicolson** (CN). Scriviamo la media aritmetica dei metodi (EE) ed (EI), sommando membro a membro:

$$\begin{cases} \frac{1}{2}u_{n+1} + \frac{1}{2}u_{n+1} = \frac{1}{2}u_n + \frac{1}{2}hf(t_{n+1}, u_{n+1}) + \frac{1}{2}u_n + \frac{1}{2}hf(t_n, u_n) \\ u_0 = y_0 \end{cases}$$

da cui si ottiene la forma del metodo di Crank-Nicolson:

$$\begin{cases} u_{n+1} = u_n + \frac{h}{2}[f(t_{n+1}, u_{n+1}) + f(t_n, u_n)] \\ u_0 = y_0 \end{cases} \tag{CN}$$

per $n = 0, 1, \ldots, N_h - 1$.

Notiamo che è un metodo implicito, in quanto compare u_{n+1} da entrambe le parti dell'uguaglianza. Ciò significa che non abbiamo già *pronto e impacchettato* il suo valore, ma dovremo risolvere l'equazione implicita per trovarlo.

Osservazione. Il metodo di Crank-Nicolson si può ricavare anche partendo dalla formulazione integrale del (PC):

$$y(t) = y_0 + \int_{t_0}^{t} f(\tau, y(\tau))\mathrm{d}\tau, \quad t \in I.$$

Scriviamo la formulazione integrale anche per i sottointervalli (t_n, t_{n+1}):

$$\int_{t_n}^{t_{n+1}} y'(\tau)\mathrm{d}\tau = \int_{t_n}^{t_{n+1}} f(\tau, y(\tau))\mathrm{d}\tau$$
$$y(t_{n+1}) - y(t_n) = \int_{t_n}^{t_{n+1}} f(\tau, y(\tau))\mathrm{d}\tau$$
$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(\tau, y(\tau))\mathrm{d}\tau$$

per poi approssimare l'integrale con la formula dei trapezi:

$$\begin{cases} u_{n+1} = u_n + \frac{h}{2}[f(t_{n+1}, u_{n+1}) + f(t_n, u_n)] \\ u_0 = y_0 \end{cases} \qquad n = 0, 1, \ldots, N_h - 1.$$

7.2.4 Metodo di Heun

Analizziamo il metodo di **Heun**, che abbrevieremo con (H). Ripendiamo il metodo di (CN):

$$\begin{cases} u_{n+1} = u_n + \frac{h}{2}[f(t_{n+1}, u_{n+1}) + f(t_n, u_n)] \\ u_0 = y_0. \end{cases} \tag{CN}$$

Nel membro destro, sostituiamo a u_{n+1} una sua stima $\hat{u}_{n+1}$ che siamo in grado di calcolare. Otteniamo quindi, per $n = 0, 1, \ldots, N_h - 1$:

$$\begin{cases} \hat{u}_{n+1} = u_n + hf(t_n, u_n) \\ u_{n+1} = u_n + \frac{h}{2}[f(t_{n+1}, \hat{u}_{n+1}) + f(t_n, u_n)] \\ u_0 = y_0. \end{cases} \tag{H}$$

In questo modo, il metodo è diventato esplicito dato che non compare u_{n+1}.

7.3 Analisi dei metodi a un passo

Studiamo ora l'efficacia dei metodi a un passo e la loro convergenza. Cerchiamo inoltre di capire quali metodi funzionano meglio per quali specifiche situazioni.

7.3.1 Consistenza

Osserviamo che ciascuno dei metodi visti può essere scritto nella seguente forma generale:

$$\begin{cases} u_{n+1} = u_n + h\Phi[t_n, u_n, f(t_n, u_n); h], \quad n = 0, 1, \ldots N_h - 1 \\ u_0 = y_0 \end{cases} \tag{7.2}$$

dove $\Phi(\cdot, \cdot, \cdot; \cdot)$ è detta *funzione di incremento.*

Esempi.

- Nel caso di (EE):

$$\Phi[t_n, u_n, f(t_n, u_n); h] = f(t_n, u_n).$$

- Nel caso di (H):

$$\Phi[t_n, u_n, f(t_n, u_n); h] = \frac{1}{2}[f(t_{n+1}, u_n + hf(t_n, u_{n+1})) + f(t_n, u_n)].$$

Si ha che la soluzione esatta non soddisfa esattamente la soluzione numerica, ma è presente un residuo, una quantità che, se possibile, vogliamo far tendere a zero.

$$\begin{cases} y(t_{n+1}) = y(t_n) + h\Phi(t_n, y(t_n), f(t_n, y(t_n)); h) + \varepsilon_{n+1} \\ y(t_0) = y_0. \end{cases}$$

Chiamo ε_{n+1} il **residuo** che si genera all'istante t_{n+1}. Esso ha la forma

$$\varepsilon_{n+1} = h\ \tau_{n+1}(h).$$

La quantità $\tau_{n+1}(h)$ è l'**errore di troncamento locale**. Definiamo anche l'**errore di troncamento globale** come segue:

$$\tau(h) = \max_{0 \le n \le N_h - 1} |\tau_{n+1}(h)|.$$

Questi errori dipendono dal troncamento effettuato nell'approssimazione della funzione con gli sviluppi di Taylor.

Definizione 7.8. Un metodo della forma (7.2) è detto **consistente** se

$$\lim_{h \to 0} \tau(h) = 0.$$

Inoltre diciamo che il metodo della forma (7.2) ha ordine p se $\tau(h) = O(h^p)$ per $h \to 0$.

7.3.2 Zero-stabilità

Se il problema subisce una piccola perturbazione, cosa sappiamo dire della differenza tra le soluzioni numeriche con e senza perturbazione?

Consideriamo il metodo generale della forma (7.2) e perturbiamolo:

$$\begin{cases} z_{n+1} = z_n + h\Phi(t_n, z_n, f(t_n, z_n); h) + \delta_{n+1} \\ z_0 = y_0 + \delta_0 \end{cases} \qquad \forall n = 0, 1, \dots N_h - 1$$

dove $\delta_{n+1}, n = 0, \dots, N_h - 1$ e δ_0 sono le perturbazioni.

Definizione 7.9 — Zero-stabilità. Il metodo numerico della forma (7.2) è 0-stabile se esistono $h_0 > 0, C > 0, \varepsilon_0 > 0$ tali che $\forall h \in (0, h_0]$ e $\forall \varepsilon \in (0, \varepsilon_0]$, se $|\delta_n| \leq \varepsilon, 0 \leq n \leq N_h$, allora

$$\left| u_n^{(h)} - z_n^{(h)} \right| \leq C\varepsilon, \quad n = 0, \ldots, N_h.$$

Il nome zero-stabilità deriva dal fatto che se le perturbazioni distano meno di ε, le soluzioni sono controllate a meno di una costante C che non dipende da h, per $h \to 0$. Essa è una proprietà specifica del metodo numerico e non del problema di Cauchy (il quale è sempre stabile, grazie alla lipschitzianità di f).

La zero-stabilità studia il comportamento della soluzione in *intervalli limitati* per $h \to 0$.

7.3.3 Convergenza

Definizione 7.10 — Convergenza. Diciamo che un metodo è convergente se

$$|y(t_n) - u_n| \leq C(h), \quad \forall n = 0, \ldots, N_h$$

dove $C(h)$ è un infinitesimo rispetto ad h. In tal caso diciamo che il metodo è convergente con ordine p se $C(h) = O(h^p)$.

Teorema 7.11. Consideriamo un metodo della forma (7.2) che sia consistente. Allora

$$\text{convergenza} \Leftrightarrow \text{zero-stabilità}.$$

7.3.4 Convergenza di Eulero Esplicito

Riportiamo nel dettaglio l'analisi di convergenza per il metodo di Eulero Esplicito. Per ogni $n = 0, \ldots, N_h$ scriviamo l'errore come:

$$e_{n+1} = y(t_{n+1}) - u_{n+1}.$$

Aggiungiamo e sottraiamo $\tilde{u}_{n+1}$:

$$e_{n+1} = \underbrace{(y(t_{n+1}) - \tilde{u}_{n+1})}_{\text{errore di consistenza}} + \underbrace{(\tilde{u}_{n+1} - u_{n+1})}_{\text{effetto memoria}},$$

essendo

$$\tilde{u}_{n+1} = y(t_n) + hf(t_n, y(t_n)) \tag{7.3}$$

la soluzione ottenuta applicando un passo del metodo di Eulero Esplicito a partire dal dato iniziale y_n. Stiamo cercando di mantenere solo l'errore dovuto all'approssimazione della derivata e non all'effetto memoria.

Dobbiamo stimare separatamente questi due errori.

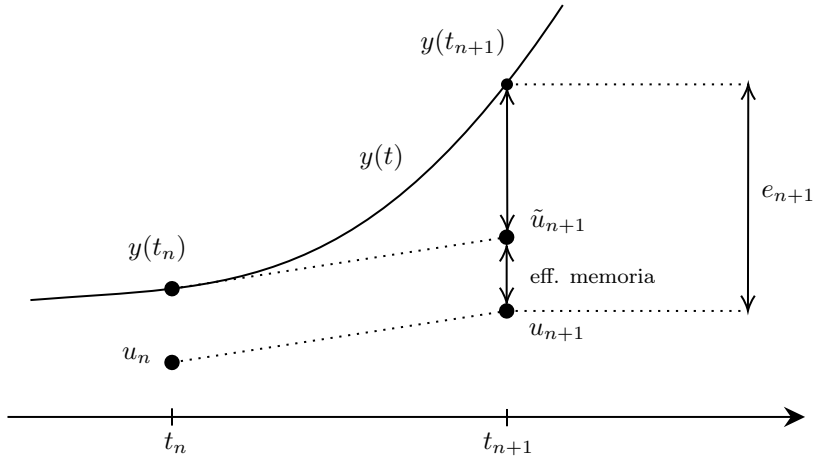


Figura 7.1: Intuizione grafica del ruolo dell'errore di consistenza e dell'effetto memoria nell'analisi di convergenza del metodo di Eulero Esplicito.

1. Stimiamo l'errore di consistenza $y(t_{n+1}) - \tilde{u}_{n+1}$.
   Scriviamo lo sviluppo di Taylor di y:

$$y(t_{n+1}) = y(t_n) + hy'(t_n) + \frac{h^2}{2}y''(\xi), \quad \xi \in (t_n, t_{n+1})$$

   e sostituiamola insieme a (7.3):

$$\begin{aligned} y(t_{n+1}) - \tilde{u}_{n+1} &= \cancel{y(t_n)} + \cancel{hf(t_n, y(t_n))} + \frac{h^2}{2}y''(\xi) - \cancel{y(t_n)} - \cancel{hf(t_n, y(t_n))} \\ &= \frac{h^2}{2}y''(\xi) = h\tau_{n+1}(h). \end{aligned}$$

2. Stimiamo l'errore dovuto all'effetto memoria, $\tilde{u}_{n+1} - u_{n+1}$:

$$\begin{aligned} \tilde{u}_{n+1} &= y(t_n) + hf(t_n, y(t_n)) \\ u_{n+1} &= u_n + hf(t_n, u_n) \\ \tilde{u}_{n+1} - u_{n+1} &= \underbrace{y(t_n) - u_n}_{e_n} + h[f(t_n, y(t_n)) - f(t_n, u_n)] && \text{(sottraiamo)} \\ |\tilde{u}_{n+1} - u_{n+1}| &\le |e_n| + h|f(t_n, y(t_n)) - f(t_n, u_n)| && \text{(modulo)} \\ &\le |e_n| + hL\underbrace{|y(t_n) - u_n|}_{e_n} && \text{(lipschitzianità)} \\ &\le (1 + hL)|e_n|. \end{aligned}$$

Mettendo insieme le due stime troviamo:

$$|e_{n+1}| \leq h\tau_{n+1}(h) + (1+hL)|e_n|$$

che possiamo riscrivere come:

$$\begin{aligned}
|e_{n+1}| &\leq h\tau(h) + (1+hL)|e_n| \\
&\leq h\tau(h) + (1+hL)[h\tau(h) + (1+hL)|e_{n-1}|] \\
&\vdots \ (e_0 = 0) \\
&\leq h\tau(h)\left[1 + (1+hL) + (1+hL)^2 + \cdots + (1+hL)^n\right] \\
&\leq h\tau(h)\sum_{k=0}^{n}(1+hL)^k
\end{aligned}$$

ricordando che $\tau(h) = \frac{h}{2}\|y''(t)\|_\infty$. Sfruttiamo ora l'espressione della serie geometrica

$$\sum_{k=0}^{n} x^k = \frac{1-x^{k+1}}{1-x},$$

pertanto abbiamo che

$$\begin{aligned}
|e_{n+1}| &\leq h\tau(h)\left[\frac{1-(1+hL)^{n+1}}{1-(1+hL)}\right] \\
&\leq \frac{\tau(h)}{L}\left[-1 + (1+hL)^{n+1}\right] \\
&\leq \frac{\left[e^{(n+1)hL} - 1\right]}{L}\tau(h) \\
&\leq \frac{e^{TL}}{L}\tau(h),
\end{aligned}$$

allora il metodo converge perché $\tau(h) = \frac{h}{2}\|y''(t)\|_\infty$ e:

$$|e_{n+1}| \leq Ch, \quad C = \frac{e^{TL}}{2L}\|y''(t)\|_\infty, \quad \forall n = 0, \ldots, N_h - 1.$$

L'ordine di convergenza è 1. ■

Riassumiamo senza dimostrare i risultati di convergenza di tutti e quattro i metodi studiati.

Teorema 7.12 — Convergenza di (EE), (EI), (CN), (H). Sia $y \in C^2(I)$ la soluzione del (PC). Allora

$$\begin{aligned}
\max_{n=0,\ldots,N_h} \left|y(t_n) - u_n^{EE}\right| &\leq C_{EE}h \\
\max_{n=0,\ldots,N_h} \left|y(t_n) - u_n^{EI}\right| &\leq C_{EI}h
\end{aligned}$$

dove $C_{EE} = C_{EE}(\|y''(t)\|_\infty, T) > 0$ e $C_{EI} = C_{EI}(\|y''(t)\|_\infty, T) > 0$. Quindi i metodi di (EE) e (EI) convergono con ordine 1 rispetto ad h.

Se invece $y \in C^3(I)$, allora

$$\max_{n=0,\dots,N_h} \left|y(t_n) - u_n^{CN}\right| \leq C_{CN} h^2$$
$$\max_{n=0,\dots,N_h} \left|y(t_n) - u_n^{H}\right| \leq C_{H} h^2$$

dove $C_{CN} = C_{CN}(\|y'''(t)\|_\infty, T) > 0$ e $C_H = C_H(\|y'''(t)\|_\infty, T) > 0$. Quindi i metodi di (CN) e (H) convergono con ordine 2 rispetto ad h.

7.3.5 Assoluta stabilità

Studiamo il comportamento dei metodi quando h è fissato e $t_n \to \infty$: in tal caso, desideriamo che la soluzione numerica u_n si mantenga *vicina* a $y(t_n)$. Questa caratteristica è nota come **assoluta stabilità**, o **$\mathcal{A}$-stabilità**.

Per analizzare la stabilità su intervalli illimitati, considereremo il seguente *problema modello*:

$$\begin{cases} y'(t) = \lambda y(t), \quad t \in (0, +\infty) \\ y(0) = 1 \end{cases} \tag{PMod}$$

con $\lambda \in \mathbb{C}$, la cui soluzione di è $y(t) = e^{\lambda t}$. Inoltre se $\Re(\lambda) < 0$, si ha che

$$\lim_{t \to +\infty} |y(t)| = 0.$$

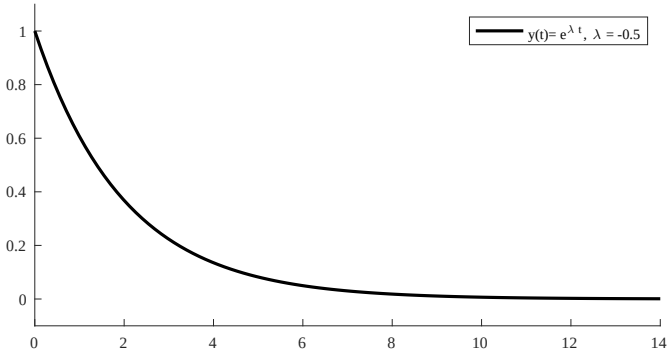


Figura 7.2: Comportamento di (PMod)

Vorremmo dunque che anche la soluzione numerica tenda a zero, in modo da catturare il comportamento asintotico della soluzione vera.

Definizione 7.13 — Assoluta stabilità. Diciamo che un metodo numerico per l'approssimazione di (PMod) è assolutamente stabile se

$$|u_n| \to 0 \quad \text{per} \quad t_n \to +\infty. \tag{7.4}$$

Osservazione. Per h fissato, la soluzione numerica u_n dipende da h e da λ. Possiamo quindi dare la seguente definizione.

Definizione 7.14 — Regione di assoluta stabilità. Definiamo $\mathcal{A}$ la regione di assoluta stabilità di un metodo numerico il seguente sottoinsieme del piano complesso:

$$\mathcal{A} = \{z = h\lambda \in \mathbb{C} \text{ tali che (7.13) è soddisfatta}\}.$$

NB. Poiché $h > 0$ e $\Re(\lambda) < 0$ per ipotesi, sicuramente $\mathcal{A} \subseteq \mathbb{C}^-$.

Definizione 7.15 — $\mathcal{A}$-stabilità. Un metodo numerico è $\mathcal{A}$-stabile se $\mathcal{A} \cap \mathbb{C}^- = \mathbb{C}^-$, ovvero se per qualsiasi $\lambda \in \mathbb{C}^-$ la condizione (7.13) è soddisfatta incondizionatamente rispetto ad h.

7.3.6 Stabilità di Eulero Esplicito

Il metodo di Eulero esplicito applicato al problema modello (PMod) diventa:

$$\begin{cases} u_{n+1} = u_n + h \underbrace{\lambda u_n}_{f(t_n,u_n)} & n = 0, 1, 2, \dots \\ u_0 = 1. \end{cases}$$

Quindi otteniamo $u_{n+1} = (1 + h\lambda)u_n$, cioè, ricorsivamente:

$$\begin{aligned} u_{n+1} &= (1 + h\lambda)u_n \\ &= (1 + h\lambda)(1 + h\lambda)u_{n-1} \\ &\vdots \\ &= (1 + h\lambda)^{n+1}u_0 \\ &= (1 + h\lambda)^{n+1}. \end{aligned}$$

Abbiamo quindi mostrato che il metodo di (EE) utilizzato per approssimare (PMod) ha la seguente forma:

$$\begin{cases} u_n = (1 + h\lambda)^n & n = 0, 1, 2, \dots \\ u_0 = 1. \end{cases}$$

Dalla definizione di assoluta stabilità si ha che:

$$\lim_{t_n \to +\infty} |u_n| = 0 \quad \Leftrightarrow \quad \lim_{t_n \to +\infty} |(1 + h\lambda)^n| = 0 \Leftrightarrow |1 + h\lambda| < 1 \tag{7.5}$$

$$\Leftrightarrow \quad h\lambda \in \mathbb{C}^- \text{ e } 0 < h < \frac{-2\Re(\lambda)}{|\lambda|^2}. \tag{7.6}$$

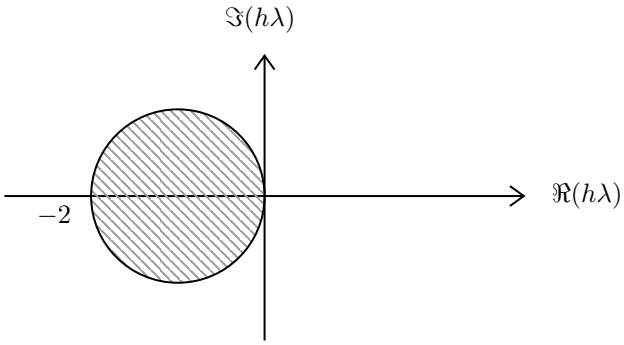


Figura 7.3: Regione di stabilità di (EE). Si notino in particolare gli assi che sono la parte reale e immaginaria di $h\lambda$.

Ovvero abbiamo mostrato che (EE) è assolutamente stabile solo sotto queste condizioni. Si dice quindi che il metodo di (EE) è *condizionatamente assolutamente stabile*.

Osservazione. Se $\lambda \in \mathbb{R}, \lambda < 0$ allora (7.6) diventa $h < 2/|\lambda|$.

7.3.7 Stabilità di Eulero Implicito

Discretizziamo ora (PMod) con il metodo di (EI):

$$\begin{cases} u_{n+1} = u_n + h\lambda u_{n+1} & n = 0, 1, 2, \dots \\ u_0 = 1 \end{cases}$$

da cui otteniamo $u_0 = 1$ e $(1 - h\lambda)u_{n+1} = u_n$ e quindi:

$$\begin{cases} u_{n+1} = \frac{1}{(1-h\lambda)} u_n & n = 0, 1, 2, \dots \\ u_0 = 1, \end{cases}$$

ragionando ricorsivamente come prima $u_{n+1} = \frac{1}{(1-h\lambda)^{n+1}}$. Dunque:

$$\lim_{t_n \to +\infty} |u_n| = 0 \Leftrightarrow \lim_{t_n \to +\infty} \left| \frac{1}{(1 - h\lambda)^n} \right| = 0 \Leftrightarrow \left| \frac{1}{(1 - h\lambda)^n} \right| < 1 \Leftrightarrow |1 - h\lambda| > 1.$$

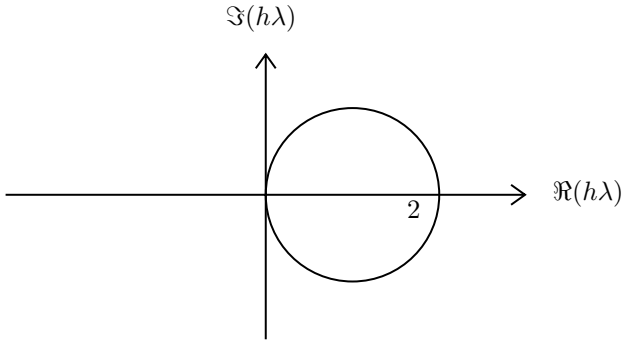


Figura 7.4: Regione di stabilità di (EI). L'intersezione tra il complementare del cerchio e il semipiano di sinistra è proprio il semipiano di sinistra, da cui l'assoluta stabilità.

Il metodo di (EI) è assolutamente stabile senza condizioni su h perché ci interessa solo la parte negativa del piano complesso, essendo $h > 0$ e $\Re(\lambda) < 0$. In altre parole, il metodo di (EI) è $\mathcal{A}$-stabile.

Ecco perché in certe occasioni, quando il dominio dell'equazione differenziale è molto ampio, i metodi impliciti sono più convenienti, pur essendo più costosi.

7.3.8 Stabilità di Crank-Nicolson e Heun

Discretizzando (PMod) con il metodo di (CN) otteniamo:

$$\begin{cases} u_{n+1} = \left[\frac{\left(1+\frac{h\lambda}{2}\right)}{\left(1-\frac{h\lambda}{2}\right)}\right]^{n+1} n \geq 0 \\ u_0 = 1 \end{cases} \tag{CN'}$$

che si rivela quindi $\mathcal{A}$-stabile. Invece per (H) otteniamo:

$$\begin{cases} u_{n+1} = \left[1 + h\lambda + \frac{(h\lambda)^2}{2}\right]^{n+1} n \geq 0 \\ u_0 = 1. \end{cases} \tag{H'}$$

Esso è dunque condizionatamente assolutamente stabile. La regione di stabilità per (H) è rappresenta in figura 7.5.

Osservazioni.

1. Si può dimostrare che un metodo esplicito non può essere $\mathcal{A}$-stabile, cioè tutti i metodi espliciti sono condizionatamente assolutamente stabili: esiste sempre una qualche condizione su $h\lambda$. Come spesso succede, si ha quindi un trade-off tra efficienza computazionale (metodi espliciti) e stabilità dell'approssimazione (metodi impliciti).

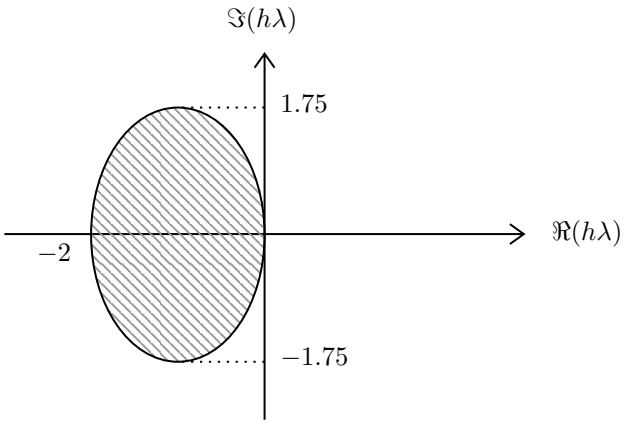


Figura 7.5: Regione di stabilità per (H).

2. Non tutti i metodi impliciti sono $\mathcal{A}$-stabili.

7.3.9 Tabella riassuntiva

	Consistenza	Zero-stabilità	Ordine di convergenza	Assoluta stabilità
(EE)	sì	sì	h	condiz. ass. stabile
(EI)	sì	sì	h	$\mathcal{A}$-stabile
(CN)	sì	sì	h^2	$\mathcal{A}$-stabile
(H)	sì	sì	h^2	condiz. ass. stabile

Nel seguito vedremo due macro-famiglie di metodi numerici di ordine elevato per EDO:

- **Metodi di Runge-Kutta** (a 1 passo): per aumentare l'ordine di convergenza si utilizzano delle valutazioni aggiuntive di f tra t_n e t_{n+1}.
- **Metodi multistep** (a p passi): l'ordine di convergenza è legato al numero di passi utilizzati.

7.4 Metodi di Runge-Kutta

Partiamo ancora una volta dal problema di Cauchy (PC). I metodi di Runge-Kutta hanno la seguente forma:

$$\begin{cases} u_{n+1} = u_n + h \sum_{i=1}^{s} b_i K_i & \text{con } n \geq 0 \\ u_0 = y_0, \end{cases} \tag{7.7}$$

dove s è detto **numero di stadi** del metodo, ed è legato all'ordine del metodo.

I coefficienti K_i sono le valutazioni non solo in t_n e t_{n+1} ma anche in punti intermedi:

$$K_i = f\left(t_n + c_i h, u_n + h\sum_{j=1}^{s} a_{ij}K_j\right), \quad i = 1, \dots, s. \tag{7.8}$$

I coefficienti $\{a_{ij}\}_{i,j=1,\dots,s}$, $\{b_i\}_{i=1,\dots,s}$ e $\{c_i\}_{i=1,\dots,s}$ sono memorizzati nell'*array di Butcher* e caratterizzano univocamente il particolare metodo di RK utilizzato:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array} \quad \text{o} \quad \begin{array}{c|c} \mathbf{c} & A \\ \hline & \mathbf{b}^T \end{array} \tag{7.9}$$

Come suggerito dalla notazione, i vincoli applicati a questi coefficienti sono:

$$c_i = \sum_{j=1}^{s} a_{ij}, \quad \sum_{i=1}^{s} b_i = 1, \quad i = 1, \dots, s. \tag{7.10}$$

Vedremo nel teorema 7.18 che la condizione su b serve a garantire un'importante proprietà del metodo.

7.4.1 Classificazione dei metodi Runge-Kutta

I metodi di RK si possono classificare nel seguente modo:

- **RK espliciti** : $a_{ij} = 0 \ \forall j \geq i$). K_i può essere calcolato a partire da $K_1, K_2, \dots, K_{i-1}$.

$$\begin{array}{c|cccc} c_1 & 0 & 0 & \cdots & 0 \\ c_2 & a_{21} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & 0 \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

- **RK semi-impliciti** : $a_{ij} = 0 \ \forall j > i$). K_i dipende non linearmente solo da sé stesso. Abbiamo un sistema di s equazioni non-lineari disaccoppiate in $K_1, \dots, k_s$.

$$\begin{array}{c|cccc} c_1 & a_{11} & 0 & \cdots & 0 \\ c_2 & a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

- **RK impliciti**. Non ho restrizioni su a_{ij}, abbiamo un sistema di s equazioni non-lineari in $k_1, \dots, k_s$.

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array}$$

Costruiamo un metodo RK *esplicito* a $s = 2$ stadi. Partiamo dalla forma generale:

$$\begin{cases} u_{n+1} = u_n + h(b_1 K_1 + b_2 K_2) \\ u_0 = y_0, \end{cases}$$

dove

$$\begin{aligned} K_1 &= f(t_n + c_1 h, u_n + h(a_{11} K_1 + a_{12} K_2)) \\ K_2 &= f(t_n + c_2 h, u_n + h(a_{21} K_1 + a_{22} K_2)). \end{aligned} \tag{7.11}$$

L'array di Butcher risulta:

$$\begin{array}{c|cc} c_1 & a_{11} & a_{12} \\ c_2 & a_{21} & a_{22} \\ \hline & b_1 & b_2 \end{array}$$

Poiché vogliamo costruire un metodo esplicito, dobbiamo imporre $a_{ij} = 0, \forall j \geq i$, ovvero

$$a_{11} = a_{12} = a_{22} = 0.$$

Quindi l'array da calcolare diventa:

$$\begin{array}{c|cc} c_1 & 0 & 0 \\ c_2 & a_{21} & 0 \\ \hline & b_1 & b_2 \end{array}$$

A questo punto sfruttiamo le ipotesi (7.4):

$$\sum_{i=1}^{s} b_i = 1 \Rightarrow b_1 + b_2 = 1$$

$$c_i = \sum_{j=1}^{s} a_{ij} \Rightarrow \begin{cases} c_1 = 0 \\ c_2 = a_{21}. \end{cases}$$

Allora l'array diventa:

$$\begin{array}{c|cc} 0 & 0 & 0 \\ c_2 & c_2 & 0 \\ \hline & b_1 & b_2 \end{array} \quad \text{con} \quad b_1 + b_2 = 1.$$

L'idea per calcolare tali coefficienti è quella di sviluppare in serie di Taylor la soluzione numerica e quella esatta, e imporre l'uguaglianza tra i primi termini di questi sviluppi. Riprendiamo l'espressione (7.11) e il metodo, che diventano:

$$\begin{aligned} K_1 &= f(t_n, u_n) \\ K_2 &= f(t_n + c_2 h, u_n + h c_2 K_1) \\ u_{n+1} &= u_n + h(b_1 K_1 + b_2 K_2). \end{aligned}$$

Per calcolare esplicitamente c_2, b_1, b_2 supponiamo di conoscere la soluzione esatta all'istante t_n: $y(t_n) = y_n$. Sviluppiamo in serie di Taylor K_2 in un intorno di t_n, arrestandoci al secondo ordine:

$$\begin{aligned} K_2 &= f(t_n + c_2 h, u_n + h c_2 K_1) \\ &= f(t_n, y_n) + c_2 h (f_{n,t} + K_1 f_{n,y}) + O\left(h^2\right) \end{aligned}$$

dove abbiamo usato la notazione

$$f_{n,t} = \frac{\partial f(t,y)}{\partial t}|_{t=t_n, y=y_n} \quad \text{e} \quad f_{n,y} = \frac{\partial f(t,y)}{\partial y}|_{t=t_n, y=y_n},$$

che indicano le derivate parziali di f rispetto a t e y (rispettivamente), e valutata nel punto (t_n, y_n).

Sostituiamo lo sviluppo nel metodo RK:

$$\begin{aligned} u^\star_{n+1} &= u_n + h(b_1 K_1 + b_2 K_2) \\ &= y_n + h\left(b_1 f(t_n, y_n) + b_2\left(f(t_n, y_n) + c_2 h (f_{n,t} + K_1 f_{n,y}) + O\left(h^2\right)\right)\right) \\ &= y_n + h\left(b_1 f(t_n, y_n) + b_2\left(f(t_n, y_n) + c_2 h (f_{n,t} + f(t_n, y_n) f_{n,y}) + O\left(h^2\right)\right)\right). \end{aligned} \tag{7.12}$$

Analogamente, se sviluppiamo la soluzione esatta in un intorno di t_n:

$$\begin{aligned} y_{n+1} &= y_n + h y'_n + \frac{h^2}{2} y''_n + O\left(h^3\right) \\ &= y_n + h f(t_n, y_n) + \frac{h^2}{2}[f_{n,t} + f(t_n, y_n) f_{n,y}] + O\left(h^3\right), \end{aligned} \tag{7.13}$$

ricordando che (7.12) è la soluzione ottenuta con un metodo RK partendo dalla soluzione esatta.

Adesso imponiamo che i termini (7.12) e (7.13) coincidano, così l'errore di troncamento locale sarà dell'ordine di h^2:

$$\begin{cases} b_1 + b_2 = 1 \\ b_2 c_2 = \frac{1}{2}. \end{cases}$$

Quindi se queste due condizioni sono soddisfatte, si ha $\tau_n(h) = O\left(h^2\right)$.

Alla fine otteniamo una famiglia di metodi RK a 2 stadi espliciti, il cui array di Butcher è

$$\begin{array}{c|cc} 0 & 0 & 0 \\ c_2 & c_2 & 0 \\ \hline & b_1 & b_2 \end{array} \qquad \begin{cases} b_1 + b_2 = 1 \\ b_2 c_2 = \frac{1}{2} \end{cases} \qquad \begin{cases} u_{n+1} = u_n + h(b_1 K_1 + b_2 K_2) \\ K_1 = f(t_n, u_n) \\ K_2 = f(t_n + c_2 h, u_n + h c_2 K_1). \end{cases}$$

Una scelta frequente per i coefficienti è

$$b_1 = b_2 = \frac{1}{2}, \quad c_2 = 1 \quad \Rightarrow \quad \begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

quindi il metodo numerico diventa:

$$\begin{cases} u_{n+1} = u_n + \frac{1}{2} h(K_1 + K_2) \\ K_1 = f(t_n, u_n) \\ K_2 = f(t_n + h, u_n + h K_1) = f(t_{n+1}, u_n + h f(t_n, u_n)) \end{cases}$$

Questo è il metodo (H)[1].

Esempio. Vediamo un metodo RK a 4 stadi esplicito: $u_{n+1} = u_n + h(b_1 K_1 + b_2 K_2 + b_3 K_3 + b_4 K_4)$ con

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} \quad \Rightarrow \quad \begin{cases} u_{n+1} = u_n + \frac{1}{6} h(K_1 + 2K_2 + 2K_3 + K_4) \\ K_1 = f(t_n, u_n) \\ K_2 = f\left(t_n + \frac{h}{2}, u_n + \frac{h}{2} K_1\right) \\ K_3 = f\left(t_n + \frac{h}{2}, u_n + \frac{h}{2} K_2\right) \\ K_4 = f(t_{n+1}, u_n + h K_3) \end{cases}$$

questo è un metodo consistente e 0-stabile, quindi anche convergente.

7.4.2 Consistenza di un metodo RK a s stadi

Definizione 7.16 — Errore di troncamento locale. Definiamo l'**errore di troncamento locale** $\tau_{n+1}(h)$ nell'istante temporale t_{n+1} come segue:

$$h\tau_{n+1}(h) \coloneqq y(t_{n+1}) - y(t_n) - h \sum_{i=1}^{s} b_i K_i.$$

[1] Sotto mentite spoglie.

Definizione 7.17 — Consistenza. Diciamo che il metodo RK è consistente se l'**errore di troncamento globale** tende a zero:

$$\tau(h) = \max_n |\tau_n(h)| \xrightarrow{h\to 0} 0.$$

Diciamo inoltre che l'errore di troncamento globale è di ordine p, con $p \geq 1$, se $\tau(h) = O(h^p)$ per $h \to 0$.

Teorema 7.18. Un metodo RK a s stadi è consistente se e solo se

$$\sum_{i=1}^{s} b_i = 1.$$

Inoltre, poiché sono metodi a un passo, la consistenza implica la zero-stabilità e quindi la convergenza.

Teorema 7.19. Un metodo RK esplicito a s stadi non può avere ordine maggiore di s. Inoltre, non esistono metodi RK espliciti a s stadi con ordine s, per $s \geq 5$. Il legame tra le due proprietà è riassunto nella seguente tabella:

ordine richiesto	1	2	3	4	5	6	7	8
numero di stadi necessario s	1	2	3	4	6	7	9	11

NB. Nella pratica non si usano quasi mai più di 4 stadi.

7.4.3 Assoluta stabilità dei metodi RK

Ricordiamo il Problema Modello (PMod) utilizzato per lo studio dell'assoluta stabilità:

$$\begin{cases} y'(t) = \lambda y(t) \\ y(0) = 1 \end{cases} \qquad t > 0,\ \lambda \in \mathbb{C},\ \Re(\lambda) < 0. \tag{PMod}$$

Approssimando (PMod) con un metodo RK otteniamo:

$$\begin{cases} u_{n+1} = u_n + h \sum_{i=1}^{s} b_i K_i \\ u_0 = 1 \end{cases} \qquad \text{con } K_i = \lambda \left(u_n + h \sum_{j=1}^{s} a_{ij} K_j \right). \tag{7.14}$$

Riscriviamo (7.14) in forma compatta:

$$\mathbf{K} = \begin{pmatrix} K_1 \\ K_2 \\ \vdots \\ K_s \end{pmatrix} \quad \mathbf{1} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_s \end{pmatrix} \quad \begin{cases} u_{n+1} = u_n + h\mathbf{b}^T\mathbf{K} \\ u_0 = 1. \end{cases}$$

Inoltre possiamo esprimere $\mathbf{K}$ come

$$\begin{aligned}\mathbf{K} &= \lambda(u_n\mathbf{1} + hA\mathbf{K})\\ \mathbf{K} &= \lambda u_n\mathbf{1} + \lambda hA\mathbf{K}\\ \mathbf{K} - \lambda hA\mathbf{K} &= \lambda u_n\mathbf{1}\\ \mathbf{K}(I - \lambda hA) &= \lambda u_n\mathbf{1}\\ \mathbf{K} &= (I - \lambda hA)^{-1}\lambda u_n\mathbf{1}\end{aligned}$$

e quindi

$$u_{n+1} = \left[1 + h\lambda\mathbf{b}^T(I - h\lambda A)^{-1}\mathbf{1}\right] u_n = R(h\lambda)u_n$$

avendo denotato con $R(h\lambda)$ la cosiddetta **funzione di stabilità**.

Definizione 7.20 — Assoluta stabilità. Un metodo RK è assolutamente stabile se e solo se

$$|R(h\lambda)| < 1.$$

Notiamo che tale condizione implica che $|u_{n+1}| \to 0$ per $n \to \infty$. La sua **regione di assoluta stabilità** è:

$$\mathcal{A} = \{z = h\lambda \in \mathbb{C} \text{ tali che } |R(h\lambda)| < 1\}.$$

Un metodo RK si dice $\mathcal{A}$-stabile se $A \cap \mathbb{C}^- = \mathbb{C}^-$.

In figura 7.6 si possono osservare alcune regioni di stabilità per metodi Runge-Kutta.

Osservazione. Se il metodo RK è esplicito, allora A è una matrice triangolare inferiore con elementi nulli sulla diagonale, ovvero $a_{ij} = 0\ \forall j \geq i$. In particolare per i metodi RK espliciti a s stadi, con $s = 1, 2, 3, 4$, si può dimostrare che:

$$R(h\lambda) = 1 + h\lambda + \frac{1}{2}(h\lambda)^2 + \cdots + \frac{1}{s!}(h\lambda)^s.$$

7.5 Metodi multistep

Fin'ora abbiamo sempre calcolato la soluzione usando le informazioni sulla derivata abbinate alla soluzione nota o calcolata negli istanti immediatamente adiacenti. Spingiamoci oltre e cerchiamo di capire come sfruttare l'informazione calcolata non solo all'istante precedente, ma a *più* istanti precedenti.

Definizione 7.21. Un metodo si dice a q passi, con $q \geq 1$, se $\forall n \geq q - 1$, si ha che u_{n+1} dipende da u_{n+1-j} con $j = 1, \ldots, q$, ma non da valori u_k con $k < n + 1 - q$.

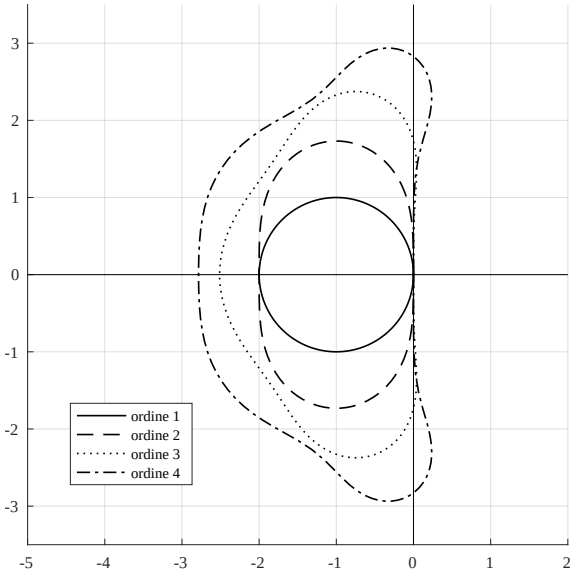


Figura 7.6: Regioni di assoluta stabilità per i metodi RK espliciti a s stadi con $s = 1, \dots, 4$

Un metodo multistep a q passi calcola quindi u_{n+1} usando le informazioni disponibili nei q istanti precedenti a quello attuale: $t_n, t_{n-1}, \dots, t_{n-q+1}$.

La forma generale di un **metodo multistep lineare** (LMS) a $p+1$ passi, con $p \geq 0$, è:

$$u_{n+1} = \sum_{j=0}^{p} a_j u_{n-j} + h \sum_{j=0}^{p} b_j f_{n-j} + h b_{-1} f_{n+1}, \quad n = p, p+1, \dots \tag{7.15}$$

dove i coefficienti $a_j, b_j \in \mathbb{R}$ caratterizzano univocamente lo schema e sono tali che $a_p \neq 0$ o $b_p \neq 0$, e $f_n = f(t_n, u_n)$.

Osservazione.

- se $b_{-1} = 0$, si ha un metodo multistep esplicito
- se $b_{-1} \neq 0$, si ha un metodo multistep implicito
- se $p = 0$, abbiamo i metodi a un passo.

Esempi.

- Un noto metodo a due passi esplicito può essere ad esempio ottenuto utilizzando l'approssimazione centrata della derivata prima: si trova il

cosiddetto **metodo del punto medio**:

$$u_{n+1} = u_{n-1} + 2hf_n, \quad n \geq 1.$$

- Uno schema implicito a due passi è invece fornito dal **metodo di Simpson**, ottenuto a partire dalla forma integrale sull'intervallo (t_{n-1}, t_{n+1}) ed utilizzando la formula di Cavalieri-Simpson:

$$u_{n+1} = u_{n-1} + \frac{h}{6}[f_{n-1} + 4f_n + f_{n+1}], \quad n \geq 1.$$

Osserviamo ora che è possibile riformulare (7.15) come

$$\sum_{s=0}^{p+1} \alpha_s u_{n+s} = h \sum_{s=0}^{p+1} \beta_s f(t_{n+s}, u_{n+s}), \quad n = 0, 1, \ldots, N_h - (p+1) \tag{7.16}$$

avendo posto $\alpha_{p+1} = 1$, $\alpha_s = -a_{p-s}$ per $s = 0, \ldots, p$ e $\beta_s = b_{p-s}$ per $s = 0, \ldots, p+1$.

Definizione 7.22 — Errore di troncamento locale. Un metodo della forma (7.15) ha il seguente errore di troncamento locale:

$$h\tau_{n+1}(h) = y(t_{n+1}) - \left[\sum_{j=0}^{p} a_j y(t_{n-j}) + h \sum_{j=-1}^{p} b_j y'(t_{n-j}) \right], \quad n \geq p.$$

Definizione 7.23 — Consistenza. Un metodo multistep è consistente se

$$\tau(h) = \max_n |\tau_n(h)| \xrightarrow{h \to 0} 0.$$

Inoltre se $\tau(h) = O(h^q)$, per qualche $q \geq 1$, allora il metodo si dirà di ordine q.

Esempi di metodi multistep:

- Metodi di Adams, che sono sviluppati a partire dalla formulazione integrale:
  - **Metodi di Adams-Bashforth** (AB) (espliciti)
  - **Metodi di Adams-Moulton** (AM) (impliciti)
- **Metodi BDF**, che approssimano la derivata su $p+1$ nodi.

Affrontiamo nel dettaglio i metodi di Adams. Come già visto per il metodo di Crank-Nicholson (approssimato mediante la formula del trapezio), il nostro problema differenziale (PC) può essere scritto equivalentemente con una formulazione integrale:

$$\begin{cases} y'(t) = f(t, y(t)) \\ y(t_0) = y_0 \end{cases} \quad \Leftrightarrow \quad y(t) = y_0 + \int_{t_0}^{t} f(s, y(s)) \mathrm{d}s.$$

In termini numerici, si ha quindi che:

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(s, y(s))\mathrm{d}s. \tag{7.17}$$

Si noti che l'intervallo di integrazione è da un generico t_n al suo istante successivo, diversamente dal modello continuo in cui integriamo dal tempo iniziale fissato a t generico. Per costruire un metodo di Adams si parte dalla formulazione integrale (7.17). Come è stato detto, i metodi multistep tengono conto delle informazioni ottenute ai $p+1$ passi precedenti, in questo caso le utilizziamo per valutare la f nei vari istanti temporali: denotiamo quindi $f_n = f(t_n, u_n)$. A questo punto abbiamo una serie di nodi in cui è nota f, costruiamo il polinomio interpolante in tali punti per f e nell'integrale della formulazione (7.17) usiamo il polinomio anziché f.

Se siamo nel caso di metodi *espliciti* a $(p+1)$ passi il polinomio interpolante passa per i nodi da t_{n-p} a t_n, che sono $(p+1)$ nodi, quindi il polinomio interpolante avrà grado p:

$$u_{n+1} = u_n + \int_{t_n}^{t_{n+1}} \Pi_p(t)\mathrm{d}t.$$

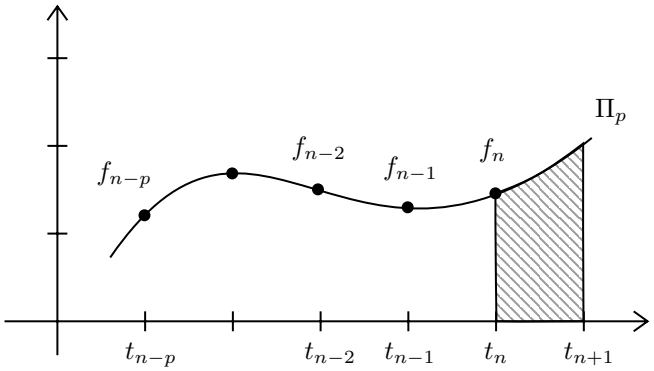


Figura 7.7: Intuizione grafica dei metodi di Adams espliciti.

Se siamo nel caso di metodi *impliciti* a $(p+1)$ passi il polinomio interpolante passa per i nodi da t_{n-p} a t_{n+1}, che sono $(p+2)$ nodi, quindi il polinomio interpolante avrà grado $p+1$:

$$u_{n+1} = u_n + \int_{t_n}^{t_{n+1}} \Pi_{p+1}(t)\mathrm{d}t.$$

La sostanziale differenza è che, come si nota in figura 7.8, il polinomio passa anche per l'istante t_{n+1}. Questo fa si che quando si andrà a determinare il polinomio

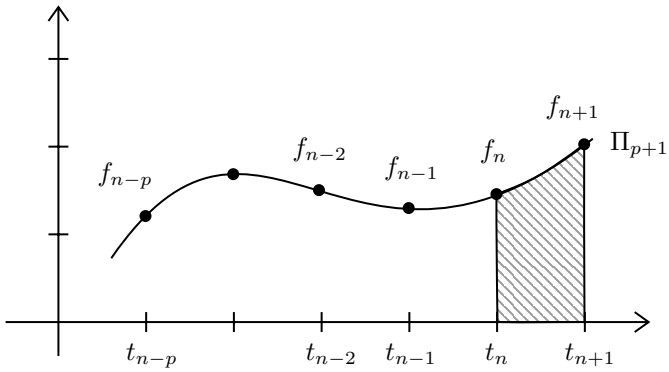


Figura 7.8: Intuizione grafica dei metodi di Adams impliciti.

$\Pi_{p+1}(t)$, esso a sua volta dipenderà da u_{n+1}, da cui il fatto che è un metodo implicito.

Si noti anche che nei due casi esaminati si fanno sempre $(p+1)$ passi indietro, ma nei metodi impliciti usiamo un nodo in più. Questa fatica ulteriore è tuttavia compensata da un ordine di convergenza in più rispetto ai metodi espliciti, come si vedrà in sezione 7.5.1.

La formula generale di un metodo di Adams è quindi

$$u_{n+1} = u_n + h \sum_{j=-1}^{p} b_j f(t_{n-j}, u_{n-j}) \tag{7.18}$$

in cui

- se $b_{-1} = 0$, stiamo interpolando su $p+1$ nodi, ovvero $t_n, t_{n-1}, \ldots, t_{n-p}$, e otteniamo i metodi di Adams espliciti (Adams-Bashforth);
- se $b_{-1} \neq 0$, stiamo interpolando su $p+2$ nodi, ovvero $t_{n+1}, t_n, t_{n-1}, \ldots, t_{n-p}$, e otteniamo i metodi di Adams impliciti (Adams-Moulton).

Esempi.

- AB (esplicito) con $p = 1$, cioè a 2 passi:

$$u_{n+1} = u_n + \int_{t_n}^{t_{n+1}} \Pi_1 f(s) \mathrm{d}s.$$

Per $p = 1$, il polinomio interpolatore nei nodi t_{n-1} e t_n è dato da:

$$\Pi_1 f(t) = f(t_n, u_n) + (t - t_n) \frac{f(t_{n-1}, u_{n-1}) - f(t_n, u_n)}{t_{n-1} - t_n}.$$

Valutiamo il polinomio nei due nodi:

$$\begin{aligned}\Pi_1 f(t_n) &= f_n \\ \Pi_1 f(t_{n+1}) &= f(t_n, u_n) + \underbrace{(t_{n+1} - t_n)}_{h} \frac{f(t_{n-1}, u_{n-1}) - f(t_n, u_n)}{-h} \\ &= f(t_n, u_n) - [f(t_{n-1}, u_{n-1}) - f(t_n, u_n)] \\ &= 2f(t_n, u_n) - f(t_{n-1}, u_{n-1}).\end{aligned}$$

Si ottiene quindi:

$$\begin{aligned}\int_{t_n}^{t_{n+1}} \Pi_1 f(s)\mathrm{d}s &\overset{\text{trapezi}}{=} \frac{h}{2}[\Pi_1 f(t_n) + \Pi_1 f(t_{n+1})] \\ &= \frac{h}{2}[3f(t_n, u_n) - f(t_{n-1}, u_{n-1})].\end{aligned}$$

Si ottiene pertanto lo schema AB a due passi:

$$u_{n+1} = u_n + \frac{h}{2}[3f(t_n, u_n) - f(t_{n-1}, u_{n-1})].$$

Si può dimostrare che esso è uno schema con ordine di convergenza 2.

- AM (implicito) con $p = 1$, cioè a 2 passi:

$$u_{n+1} = u_n + \int_{t_n}^{t_{n+1}} \Pi_2 f(s)\mathrm{d}s.$$

Approssimiamo f con $\Pi_2 f$, cioè il polinomio che interpola f nei nodi t_{n-1}, t_n, t_{n+1}. Svolgendo i conti si ottiene:

$$u_{n+1} = u_n + \frac{h}{12}[5f(t_{n+1}, u_{n+1}) + 8f(t_n, u_n) - f(t_{n-1}, u_{n-1})]$$

e si può dimostrare che è uno schema con ordine di convergenza 3.

L'espressione esplicita di AB e AM con $p = 0$ (che sono, rispettivamente, i metodi (EE) e (CN)) è lasciata al lettore come esercizio.

Facciamo ora un'importante osservazione. Supponendo di considerare un metodo di Adams con $p = 2$, $b_{-1} = 0$ per fissare le idee. La forma di tale metodo risulta:

$$u_{n+1} = u_n + h[b_0 f_n + b_1 f_{n-1} + b_2 f_{n-2}]$$

dove f_n dipende da u_n, f_{n-1} da u_{n-1} e f_{n-2} da u_{n-2}.

Il primo passo ammissibile è con $n = 2$, infatti se consideriamo $n = 0, 1$ intervengono u_{-1}, u_{-2} che non hanno senso, sarebbero a sinistra della nostra origine dei tempi. Partendo quindi da $n = 2$, è noto il valore di u_0, ma non sono

noti u_1, u_2. Per costruire quei due valori non possiamo usare il metodo multistep, ma dovremo affidarci ad altri metodi, ad esempio un metodo Runge-Kutta. Dato che il metodo di Adams in esame è esplicito con $p = 2$, esso avrà ordine 3, come si vedrà nella sezione 7.5.1; la scelta del metodo RK dovrà essere fatta in modo da non *sporcare* l'ordine complessivo, quindi ci servirà almeno di ordine 3.

Un ultimo esempio di metodi multistep sono i BDF (Backward Differentiation Formula). L'idea è approssimare $y'(t_{n+1})$ con la derivata del polinomio di approssimazione di grado $p+1$ costruito interpolando la funzione nei $p+2$ nodi $t_{n+1}, t_n, t_{n-1}, \ldots, t_{n-p}$, $p \geq 0$. Si ottiene che:

$$u_{n+1} = \sum_{j=0}^{p} a_j u_{n-j} + h b_{-1} f(t_{n+1}, u_{n+1}), \quad b_{-1} \neq 0.$$

7.5.1 Analisi dei metodi multistep

In generale:

- AB a $p+1$ passi ha ordine $p+1$;
- AM a $p+1$ passi ha ordine $p+2$.

Teorema 7.24 — Consistenza. Un metodo multistep è consistente se e solo se i coefficienti $\{a_j\}$ e $\{b_j\}$ soddisfano le seguenti condizioni:

$$\sum_{j=0}^{p} a_j = 1, \quad -\sum_{j=0}^{p} j a_j + \sum_{j=-1}^{p} b_j = 1. \tag{7.19}$$

Nel caso dei metodi di Adams la consistenza si ha se:

$$\sum_{j=-1}^{p} b_j = 1. \tag{7.20}$$

Invitiamo quindi il lettore a verificare negli esempi precedenti che tale condizione sia soddisfatta.

Teorema 7.25 — Ordine del metodo. Se la soluzione $y \in C^{q+1}(I), q \geq 1$, allora il metodo multistep è di ordine q se e solo se vale (7.19) e inoltre:

$$\sum_{j=0}^{p} (-j)^i a_j + i \sum_{j=-1}^{p} (-j)^{i-1} b_j = 1, \quad i = 2, \ldots, q.$$

Nel caso dei metodi di Adams, come per la consistenza, bisogna considerare nulli

i coefficienti a_j, quindi l'ordine risulta q se vale (7.20) e inoltre:

$$i \sum_{j=-1}^{p} (-j)^{i-1} b_j = 1, \quad i = 2, \ldots, q. \tag{7.21}$$

7.6 Sistemi di EDO

Consideriamo un sistema di m equazioni differenziali:

$$\begin{cases} y_1'(t) = f_1(t, y_1(t), y_2(t), \ldots, y_m(t)) \\ y_2'(t) = f_2(t, y_1(t), y_2(t), \ldots, y_m(t)) \\ \vdots \\ y_m'(t) = f_m(t, y_1(t), y_2(t), \ldots, y_m(t)) \\ y_1(0) = y_{1,0} \\ y_2(0) = y_{2,0} \\ \vdots \\ y_m(0) = y_{m,0} \end{cases}$$

Usando una notazione vettoriale, il sistema si può scrivere anche come:

$$\begin{cases} \mathbf{y}'(t) = \mathbf{F}(t, \mathbf{y}) \\ \mathbf{y}(0) = \mathbf{y_0}. \end{cases}$$

Cerchiamo la funzione $\mathbf{y}(t) : I \to \mathbb{R}^m$ che soddisfi il sistema, con $\mathbf{y_0} \in \mathbb{R}^m$, $\mathbf{F}(t, \mathbf{y}) : I \times \mathbb{R}^m \to \mathbb{R}^m$.

Teorema 7.26 — Esistenza e unicità della soluzione. Sia $\mathbf{F}(t, \mathbf{y}) : I \times \mathbb{R}^m \to \mathbb{R}^m$ una funzione continua su $I \times \mathbb{R}^m$, con $I = [t_0, t_0 + T]$. Se $\exists L > 0$ tale che

$$\|\mathbf{F}(t, \mathbf{z}) - \mathbf{F}(t, \mathbf{w})\| \leq L \|\mathbf{z} - \mathbf{w}\|, \ \forall (t, \mathbf{z}), (t, \mathbf{w}) \in I \times \mathbb{R}^m$$

allora per ogni dato iniziale $\mathbf{y_0} \in \mathbb{R}^m$ esiste un'unica soluzione $\mathbf{y}(t)$ del problema, ed essa è continua e differenziabile.

Osservazione. Tutti i metodi descritti possono essere estesi al caso di sistemi di equazioni differenziali. Per esempio, il metodo (EE) diventa, dato $\mathbf{u}_0 = \mathbf{y}(t_0) \in \mathbb{R}^m$:

$$\mathbf{u}_{n+1} = \mathbf{u}_n + h\mathbf{F}(t_n, \mathbf{u}_n), \quad n = 0, 1, 2, \ldots$$

Osservazione. Per l'analisi di assoluta stabilità consideriamo il seguente problema modello, simile al caso unidimensionale (PMod):

$$\begin{cases} \mathbf{y}'(t) = A\mathbf{y}(t) \\ \mathbf{y}(0) = \mathbf{1} \end{cases} \tag{7.22}$$

con $A \in \mathbb{R}^{m\times m}$. Se la matrice A ha gli autovalori $\lambda_1, \lambda_2, \ldots, \lambda_m$, allora la soluzione di (7.22) può essere scritta come:

$$\mathbf{y}(t) = \sum_{j=1}^{m} c_j e^{\lambda_j t}\mathbf{v}_j$$

dove $c_1, c_2, \ldots, c_m \in \mathbb{R}$ e $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m$ sono gli autovettori di A.

NB. Il metodo numerico è stabile per tempi lunghi, ovvero per $t \to \infty$, se $\|\mathbf{y}(t)\| \xrightarrow{t\to\infty} 0$. Tale condizione è soddisfatta se $\Re(\lambda_j) < 0 \ \forall j = 1, \ldots, n$.

NB. Se gli autovalori sono distinti, possiamo diagonalizzare A nel seguente modo:

$$\Lambda = Q^{-1}AQ$$

dove $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m)$. Possiamo dunque riscrivere (7.22) come segue:

$$\begin{cases} \mathbf{z}' = \Lambda\mathbf{z} \\ \mathbf{z}(0) = \mathbf{z_0} \end{cases}$$

il quale diventa il nostro problema modello per lo studio dell'assoluta stabilità.

7.7 Equazioni differenziali del secondo ordine

Sia data una EDO del secondo ordine, ovvero della forma

$$\begin{cases} y''(t) = f(t, y(t), y'(t)) \\ y(t_0) = y_0 \\ y'(t_0) = y_1. \end{cases}$$

Riscriviamola come un sistema di EDO del primo ordine, introducendo delle variabili ausiliarie:

$$\begin{cases} w_1(t) = y(t) \\ w_2(t) = y'(t). \end{cases}$$

Allora:

$$\begin{cases} w_2'(t) = f(t, w_1(t), w_2(t)) \\ w_1'(t) = w_2(t) \\ w_1(t_0) = y_0 \\ w_2(t_0) = y_1. \end{cases}$$

Definendo $\mathbf{w}(t) = [w_1(t), w_2(t)]^T$, risulta anche che $\mathbf{w}(t_0) = [y_0, y_1]^T = \mathbf{w_0}$, e definendo

$$\mathbf{F}(t, \mathbf{w}) = \begin{bmatrix} 0 \\ f(t, w_1, w_2) \end{bmatrix},$$

otteniamo

$$\begin{cases} \mathbf{w}'(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{w}(t) + \mathbf{F}(t, \mathbf{w}) \\ \mathbf{w}(t_0) = \mathbf{w_0} \end{cases}$$

che diventa un'equazione del primo ordine. In questo modo, possiamo applicare i metodi visti in precedenza anche a un'equazione di ordine più alto del primo.

Capitolo 8

Equazioni e sistemi non lineari

La complessità di un problema, o il comportamento fisico di un sistema, può essere tale da non poter essere catturata da un modello lineare. Si deve quindi ricorrere a modelli non lineari, che sono di più difficile trattazione.

Esempio. Supponiamo di avere un gas a una temperatura T soggetto a una pressione p. Vogliamo calcolare il suo volume V. L'equazione dei gas è:

$$\left[p + a\left(\frac{N}{V}\right)^2\right](V - Nb) = kNT$$

dove a, b sono costanti che dipendono dal gas, N è il numero di molecole, e k è costante di Boltzmann. Per trovare V dobbiamo dunque risolvere un'equazione non lineare.

È anche possibile che un'equazione non lineare emerga dall'applicazione di un metodo numerico implicito, come abbiamo visto nelle pagine precedenti.

Esempio. Consideriamo l'equazione differenziale

$$\begin{cases} y'(t) = \frac{y^2(t)}{t},\ t > 1 \\ y(1) = \dots \end{cases}$$

Discretizzando questo problema con il metodo di (EI) otteniamo:

$$u_{n+1} = u_n + h\left[\frac{(u_{n+1})^2}{t_{n+1}}\right], \quad n = 0, 1, 2, \dots$$

Ad ogni passo temporale dobbiamo quindi risolvere un'equazione non lineare.

La risoluzione di equazioni non lineari avviene attraverso l'individuazione degli zeri di opportune funzioni. Data una funzione f reale di variabile reale, vogliamo trovare, *se esistono*, i suoi zeri, ovvero cerchiamo α tale che $f(\alpha) = 0$.

8.1 Metodo di bisezione

Per iniziare, ricordiamo un risultato di Analisi I.

Teorema 8.1 — di Bolzano. Sia $f : [a, b] \to \mathbb{R}$ continua e tale che $f(a)f(b) < 0$. Allora esiste almeno un punto $\alpha \in (a, b)$ tale che $f(\alpha) = 0$.

NB. È importante che l'intervallo sia chiuso e limitato e ricordarsi che il teorema garantisce l'esistenza di *almeno* uno zero, ma non ne fornisce il numero esatto.

Per applicare numericamente questo risultato, costruiamo dei metodi iterativi. Scegliamo $x^{(0)} \in [a, b]$ e costruiamo una successione

$$x^{(1)}, x^{(2)}, \dots, x^{(k)} \quad \text{tale che} \quad \lim_{k\to\infty} x^{(k)} = \alpha.$$

Operativamente, procediamo con gli stessi passi della dimostrazione del teorema di Bolzano, definiamo:

$$a^{(0)} = a, b^{(0)} = b, I^{(0)} = \left[a^{(0)}, b^{(0)}\right], x^{(0)} = \frac{a^{(0)} + b^{(0)}}{2}.$$

Si presentano tre casi possibili:

- Se $f\left(x^{(0)}\right) = 0$ abbiamo finito, $\alpha = x^{(0)}$.
- Se $f\left(a^{(0)}\right) f\left(x^{(0)}\right) < 0$ allora $\alpha \in \left(a^{(0)}, x^{(0)}\right)$ quindi ridefiniamo:

$$a^{(1)} = a^{(0)}, b^{(1)} = x^{(0)}, I^{(1)} = \left(a^{(1)}, b^{(1)}\right), x^{(1)} = \frac{a^{(1)} + b^{(1)}}{2}.$$

- Se $f\left(x^{(0)}\right) f\left(b^{(0)}\right) < 0$ allora $\alpha \in \left(x^{(0)}, b^{(0)}\right)$ quindi ridefiniamo:

$$a^{(1)} = x^{(0)}, b^{(1)} = b^{(0)}, I^{(1)} = \left(a^{(1)}, b^{(1)}\right), x^{(1)} = \frac{a^{(1)} + b^{(1)}}{2}.$$

Questo procedimento ricorsivo diventa l'**algoritmo di bisezione**.

Algoritmo 19: Algoritmo di bisezione

```
1  inizializza a^(0) = a
2  inizializza b^(0) = b
3  inizializza x^(0) = (a^(0) + b^(0))/2
4  for k = 0, 1, 2, ... do
5      if f(x^(k)) = 0 then
6          α = x^(k)
7          termina algoritmo
8      else if f(a^(k)) f(x^(k)) < 0 then
9          a^(k+1) = a^(k)
10         b^(k+1) = x^(k)
11         x^(k+1) = (a^(k+1) + b^(k+1))/2
12     else if f(x^(k)) f(b^(k)) < 0 then
13         a^(k+1) = x^(k)
14         b^(k+1) = b^(k)
15         x^(k+1) = (a^(k+1) + b^(k+1))/2
16     end
17     if criterio di arresto then
18         termina algoritmo
19     end
20 end
```

Osservazioni.

1. Ad ogni passo k, si ha che $\alpha \in I^{(k)} = \left[a^{(k)}, b^{(k)}\right]$. Inoltre:

$$\left|I^{(k)}\right| = \left|b^{(k)} - a^{(k)}\right| = \frac{1}{2}\left|b^{(k-1)} - a^{(k-1)}\right| = \ldots = \frac{1}{2^k}(b-a).$$

   Quindi l'errore $e^{(k)} = x^{(k)} - \alpha$ può essere stimato come

$$\left|e^{(k)}\right| = \left|x^{(k)} - \alpha\right| \leq \frac{1}{2}\left(b^{(k)} - a^{(k)}\right) = \frac{1}{2}I^{(k)} = \frac{1}{2^{k+1}}(b-a),$$

   pertanto:

$$0 \leq \left|e^{(k)}\right| \leq \frac{1}{2^{k+1}}(b-a) \Rightarrow \lim_{k\to\infty}\left|e^{(k)}\right| = 0.$$

2. Supponiamo di voler calcolare l'errore a meno di una precisione ε, cioè garantire che $\left|e^{(k)}\right| \leq \varepsilon$. Calcoliamo l'iterazione $k_{\min}$ prima della quale non possiamo interrompere il metodo:

$$\left|e^{(k)}\right| \leq \frac{1}{2^{k+1}}(b-a) \leq \varepsilon$$

$$\begin{aligned}
\frac{1}{2^{k+1}} &\leq \frac{\varepsilon}{b-a} \\
\log_2 \frac{1}{2^{k+1}} &\leq \log_2 \frac{\varepsilon}{b-a} \\
-(k+1) &\leq \log_2 \frac{\varepsilon}{b-a} \\
k+1 &\geq -\log_2 \frac{\varepsilon}{b-a} \\
k &\geq -\log_2 \frac{\varepsilon}{b-a} - 1 \\
k &\geq \log_2 \frac{b-a}{\varepsilon} - 1,
\end{aligned}$$

ovvero:

$$k_{\min} = \left\lceil \log_2 \frac{b-a}{\varepsilon} - 1 \right\rceil$$

Analizziamo la figura 8.1: si può notare come al passo 0 la soluzione sia molto vicina allo zero della funzione. Al passo 1, in cui eliminiamo l'intervallo di sinistra, ci allontaniamo notevolmente rispetto all'iterazione precedente.

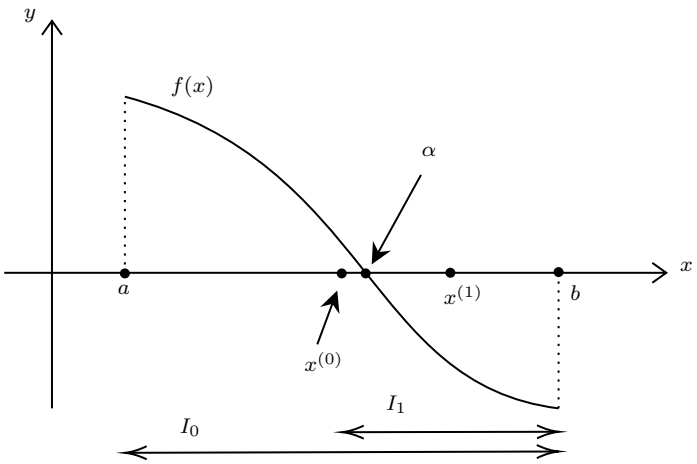


Figura 8.1: L'errore col metodo di bisezione non tende a zero in modo monotono.

In generale, la successione degli errori $e^{(k)}$ generata dal metodo di bisezione non converge a zero *monotonicamente*.

Definizione 8.2 — Ordine di convergenza. Sia $\left\{x^{(k)}\right\}_{k\geq 0}$ la successione di approssimazioni di α generata da un metodo numerico. Diciamo che la

successione converge ad α con ordine p, con ≥ 1, se esiste $C > 0$ tale che

$$\frac{\left|x^{(k+1)} - \alpha\right|}{\left|x^{(k)} - \alpha\right|^p} \leq C, \quad \forall k \geq k_0, \quad k_0 \in \mathbb{Z}_+ \cup \{0\}.$$

Se $p = 1$, per avere convergenza deve essere $C < 1$, e in questo caso C è detto **fattore di convergenza**.

Vogliamo costruire dei metodi numerici che garantiscano convergenza secondo questa definizione.

8.2 Approccio geometrico per l'approssimazione di radici

L'idea è di fare uno sviluppo di Taylor nell'intorno del punto x:

$$0 = f(\alpha) = f(x + (\alpha - x)) = f(x) + (\alpha - x)f'(\xi).$$

Se pensiamo la soluzione α come l'iterata successiva $x^{(k+1)}$ e x come il punto nel quale il metodo si trova attualmente, otteniamo la struttura del metodo iterativo:

$$f\left(x^{(k)}\right) + \left(x^{(k+1)} - x^{(k)}\right) f'(\xi) = 0.$$

Si noti che il punto ξ non è noto, altrimenti basterebbe un solo passo, calcolando la derivata della funzione in ξ e arrivando direttamente ad α. Per questa ragione, sostituiamo il termine $f'(\xi)$ con $q^{(k)}$, la cui scelta caratterizzerà il metodo numerico e determinerà la velocità con cui esso arriverà ad α:

$$\begin{aligned} f\left(x^{(k)}\right) + \left(x^{(k+1)} - x^{(k)}\right) q^{(k)} &= 0 \\ \left(x^{(k+1)} - x^{(k)}\right) q^{(k)} &= -f\left(x^{(k)}\right) \\ x^{(k+1)} - x^{(k)} &= -\frac{f\left(x^{(k)}\right)}{q^{(k)}} \\ x^{(k+1)} &= x^{(k)} - \frac{f\left(x^{(k)}\right)}{q^{(k)}}. \end{aligned}$$

La forma generale è quindi:

$$x^{(k+1)} = x^{(k)} - \frac{1}{q^{(k)}} f\left(x^{(k)}\right), \quad \forall k \geq 0. \tag{8.1}$$

Vediamo ora tre metodi per approssimare le radici:

- **Metodo di Newton**: Supponiamo $f \in C^1(a,b)$ e $f'(x) \neq 0, \forall x \in (a,b)$, scegliamo
$$q^{(k)} = f'\left(x^{(k)}\right), \quad \forall k \geq 0,$$
allora (8.1) diventa:
$$x^{(k+1)} = x^{(k)} - \frac{f\left(x^{(k)}\right)}{f'\left(x^{(k)}\right)}, \quad \forall k \geq 0.$$

- **Metodo delle secanti**: Per il metodo delle secanti, scegliamo invece
$$q^{(k)} = \frac{f\left(x^{(k)}\right) - f\left(x^{(k-1)}\right)}{x^{(k)} - x^{(k-1)}}, \quad \forall k \geq 1,$$
allora (8.1) diventa:
$$x^{(k+1)} = x^{(k)} - \frac{x^{(k)} - x^{(k-1)}}{f\left(x^{(k)}\right) - f\left(x^{(k-1)}\right)} f\left(x^{(k)}\right), \quad \forall k \geq 1.$$

- **Metodo delle corde**: Il metodo delle corde usa:
$$q^{(k)} = q = \frac{f(b) - f(a)}{b - a}, \quad \forall k \geq 0,$$
quindi (8.1) diventa:
$$x^{(k+1)} = x^{(k)} - \frac{b-a}{f(b) - f(a)} f\left(x^{(k)}\right), \quad \forall k \geq 0.$$

NB. I metodi descritti convergono *localmente* rispetto alla definizione che ci siamo dati, ovvero è importante che il dato iniziale $x^{(0)}$ sia scelto ragionevolmente vicino allo zero esatto della funzione. Diversamente dai metodi iterativi per sistemi lineari, qui la scelta del dato iniziale è importante.

8.3 Metodo delle iterazioni di punto fisso

Vediamo un modo più generale di vedere gli algoritmi per equazioni non lineari. A questo scopo, sia α tale che $f(\alpha) = 0$. Se definiamo:
$$\Phi(x) \coloneqq x - f(x), \tag{8.2}$$
allora risulta che α è un **punto fisso** di $\Phi(x)$, infatti:
$$\Phi(\alpha) = \alpha - \underbrace{f(\alpha)}_{=0} = \alpha,$$

quindi cercare gli zeri di f è equivalente a cercare i punti fissi della cosiddetta **funzione di iterazione** $\Phi(x)$.

L'idea è di calcolare ogni iterata come la funzione di iterazione valutata all'iterata precedente:

$$x^{(k+1)} = \Phi\big(x^{(k)}\big), \quad k \geq 0, \tag{8.3}$$

e si prosegue fino a convergenza.

L'algoritmo è pertanto il seguente.

Algoritmo 20: Algoritmo delle iterazioni di punto fisso.

```
1 inizializza x^(0)
2 for k = 0, 1, ... do
3 |   x^(k+1) = Φ(x^(k))
4 |   if criterio di arresto then
5 |   |   termina algoritmo
6 |   end
7 end
```

L'approccio visivo si può osservare in figura 8.2.

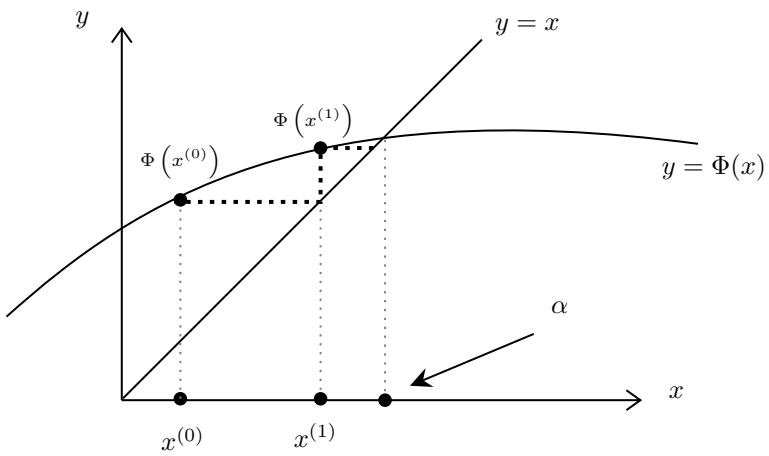


Figura 8.2: Intuizione geometrica del metodo delle iterazioni di punto fisso.

È tuttavia facile pensare a dei controesempi in cui la funzione Φ non porta a convergenza, come in figura 8.3. Tale problema sarà risolto nel teorema 8.4.

Osservazione. Il metodo delle corde e il metodo di Newton possono essere scritti come metodi di iterazione di punto fisso:

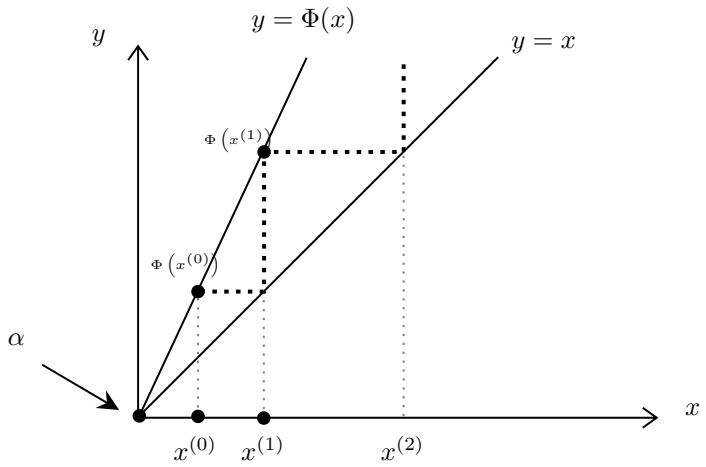


Figura 8.3: Esempio in cui il metodo di punto fisso non porta a convergenza.

- Metodo delle corde:

$$x^{(0)}, \quad x^{(k+1)} = \Phi_{\text{corde}}\left(x^{(k)}\right) \quad \text{con} \quad \Phi_{\text{corde}}(x) = x - \frac{b-a}{f(b)-f(a)} f(x).$$

- Metodo di Newton:

$$x^{(k+1)} = \Phi_{\text{Newton}}\left(x^{(k)}\right) \quad \text{con} \quad \Phi_{\text{Newton}}(x) = x - \frac{f(x)}{f'(x)}.$$

Teorema 8.3 — Convergenza delle iterazioni di punto fisso. Consideriamo la successione delle iterazioni di punto fisso (8.3).

Supponiamo che Φ sia continua in $[a, b]$ e sia tale che $\Phi(x) \in [a, b]$ per ogni $x \in [a, b]$; allora esiste almeno un punto fisso $\alpha \in [a, b]$.

Se supponiamo inoltre che Φ sia una contrazione, cioè che:

$$\exists L < 1 \text{ t.c. } |\Phi(x_1) - \Phi(x_2)| \leq L|x_1 - x_2| \quad \forall x_1, x_2 \in [a, b].$$

Allora il punto fisso α è unico e la successione (8.3) converge ad α, per qualunque scelta del dato iniziale $x^{(0)} \in [a, b]$.

Teorema 8.4 — di Ostrowski. Sia α un punto fisso di una funzione Φ continua e derivabile con continuità in un opportuno intorno I di α. Se risulta $|\Phi'(\alpha)| < 1$, allora esiste $\delta > 0$ in tale che la successione $\left\{x^{(k)}\right\}$ converge ad α,

per ogni $x^{(0)}$ per cui si abbia $\left|x^{(0)} - \alpha\right| < \delta$. Inoltre:

$$\lim_{k\to\infty} \frac{x^{(k+1)} - \alpha}{x^{(k)} - \alpha} = \Phi'(\alpha).$$

La quantità $|\Phi'(\alpha)|$ è detta **fattore asintotico di convergenza** e, in analogia con il caso dei metodi iterativi per la risoluzione di sistemi lineari, si definisce la **velocità asintotica di convergenza** come segue:

$$R = -\log(|\Phi'(\alpha)|).$$

Ricapitolando:

- se $|\Phi'(\alpha)| < 1$ il metodo è localmente convergente. In particolare:
  - se $\Phi'(\alpha) > 0$ le iterate si avvicinano ad α in maniera monotona;
  - se $\Phi'(\alpha) < 0$ le iterate si avvicinano ad α oscillando intorno ad α;
- se $|\Phi'(\alpha)| > 1$ il metodo è divergente;
- se $|\Phi'(\alpha)| = 1$ non è possibile trarre conclusioni dal teorema.

Vediamo ora un altro risultato notevole sui metodi di punto fisso.

Teorema 8.5. Se $\Phi \in C^{p+1}(I)$ per un opportuno intorno I di α e per un intero $p \geq 1$, e se $\Phi^{(i)}(\alpha) = 0$ per $i = 1, \ldots, p$ mentre $\Phi^{(p+1)}(\alpha) \neq 0$, allora il metodo di punto fisso con funzione di iterazione Φ ha ordine $p+1$ e risulta inoltre che

$$\lim_{k\to\infty} \frac{x^{(k+1)} - \alpha}{\left(x^{(k)} - \alpha\right)^{p+1}} = \frac{\Phi^{(p+1)}(\alpha)}{(p+1)!}.$$

8.3.1 Convergenza del metodo delle corde

Ricordiamo che per il metodo delle corde si ha che:

$$\Phi_{\text{corde}}(x) = x - \frac{b-a}{f(b) - f(a)} f(x).$$

Osserviamo che se $f \in C^1([a,b])$ allora $\Phi_{\text{corde}} \in C^1([a,b])$. Inoltre:

$$\Phi'_{\text{corde}}(x) = 1 - \frac{b-a}{f(b) - f(a)} f'(x).$$

Ora, se α è lo zero di f (α è il punto fisso di Φ_{corde}), vorremmo usare il teorema di Ostrowski. Verifichiamo dove esso sia applicabile:

$$|\Phi'_{\text{corde}}(\alpha)| = \left|1 - \frac{b-a}{f(b) - f(a)} f'(\alpha)\right| = \begin{cases} \text{se } f'(\alpha) = 0 \Rightarrow |\Phi'_{\text{corde}}(\alpha)| = 1 \text{ (no)} \\ \text{se } f'(\alpha) \neq 0 \Rightarrow |\Phi'_{\text{corde}}(\alpha)| < 1 \text{ (sì)}. \end{cases}$$

Studiamo quindi la disequazione che permette di applicare il teorema:

$$\left|1 - \frac{b-a}{f(b)-f(a)} f'(\alpha)\right| < 1$$
$$\Updownarrow$$
$$-1 < 1 - \frac{b-a}{f(b)-f(a)} f'(\alpha) < 1$$
$$\Updownarrow$$
$$\begin{cases} \dfrac{b-a}{f(b)-f(a)} f'(\alpha) > 0 \\ \dfrac{b-a}{f(b)-f(a)} f'(\alpha) < 2 \end{cases}$$
$$\Updownarrow$$
$$\begin{cases} \dfrac{b-a}{f(b)-f(a)} \text{ e } f'(\alpha) \text{ concordi} \\ b-a < \dfrac{2[f(b)-f(a)]}{f'(\alpha)}. \end{cases}$$

Se queste due condizioni sono soddisfatte contemporaneamente, per il teorema di Ostrowski il metodo delle corde converge. Se non dovessero essere soddisfatte, possiamo tentare di fare alcuni passi col metodo di bisezione finché queste condizioni non si verificano.

8.3.2 Convergenza del metodo di Newton

Ricordiamo l'espressione del metodo di Newton:

$$\Phi_{\text{Newton}}(x) = x - \frac{f(x)}{f'(x)}.$$

Si presentano due casi: o α è uno zero con molteplicità 1 per f (detto zero semplice), oppure ha molteplicità maggiore di 1. Affrontiamo questi due casi separatamente. Supponiamo inizialmente che α sia uno zero semplice, cioè $f(\alpha) = 0$, $f'(\alpha) \neq 0$. In questo caso:

$$\Phi'_{\text{Newton}}(x) = 1 - \frac{[f'(x)]^2 - f(x)f''(x)}{[f'(x)]^2}$$

e dunque:

$$\Phi'_{\text{Newton}}(\alpha) = 1 - \frac{[f'(\alpha)]^2 - f(\alpha)f''(\alpha)}{\underbrace{[f'(\alpha)]^2}_{\text{sicuramente } \neq 0}} = 1 - 1 - \frac{\overbrace{f(\alpha)}^{=0} f''(\alpha)}{[f'(\alpha)]^2} = 0.$$

Calcoliamo una derivata successiva per determinare l'ordine di convergenza:

$$\begin{aligned}
\Phi''_{\text{Newton}}(x) &= \frac{d}{dx}\left[1 - \frac{[f']^2 - ff''}{[f']^2}\right] \\
&= 0 - \left(\frac{[2f'f'' - (f'f'' + ff''')]\left[[f']^2\right] - \left[[f']^2 - ff''\right][2f'f'']}{[f']^4}\right) \\
&= -\frac{\left[2[f']^3 f'' - [f']^3 f'' - [f']^2 ff'''\right] - \left[2[f']^3 f'' - 2ff'[f'']^2\right]}{[f']^4} \\
&= -\frac{\cancel{2[f']^3 f''} - [f']^3 f'' - [f']^2 ff''' \cancel{-2[f']^3 f''} + 2ff'[f'']^2}{[f']^4} \\
&= \frac{[f']^3 f'' + [f']^2 ff''' - 2ff'[f'']^2}{[f']^4} \\
&= \frac{[f']^2 f'' + f'ff''' - 2f[f'']^2}{[f']^3} \\
&= \frac{[f'(x)]^2 f''(x) + f(x)f'(x)f'''(x) - 2f(x)[f''(x)]^2}{[f'(x)]^3}.
\end{aligned}$$

Pertanto:

$$\begin{aligned}
\Phi''_{\text{Newton}}(\alpha) &= \frac{[f'(\alpha)]^2 f''(\alpha) + \overbrace{f(\alpha)}^{=0} f'(\alpha)f'''(\alpha) - 2\overbrace{f(\alpha)}^{=0}[f''(\alpha)]^2}{[f'(\alpha)]^3} \\
&= \frac{[f'(\alpha)]^2 f''(\alpha)}{[f'(\alpha)]^3} \\
&= \frac{f''(\alpha)}{f'(\alpha)} \neq 0.
\end{aligned}$$

In questo caso il metodo di Newton converge localmente con ordine 2.

Supponiamo ora invece che α sia uno zero con molteplicità $m > 1$, cioè:

$$f(\alpha) = 0, \quad f'(\alpha) = 0, \quad f''(\alpha) = 0, \quad \ldots, \quad f^{(m-1)}(\alpha) = 0, \quad f^{(m)}(\alpha) \neq 0.$$

In questo caso si può dimostrare che:

$$\Phi'_{\text{Newton}}(\alpha) = 1 - \frac{1}{m} \neq 0,$$

ovvero che il metodo converge localmente con ordine 1. Abbiamo perso un ordine, ma possiamo ripristinarlo utilizzando il **metodo di Newton modificato** (NM):

$$x^{(k+1)} = x^{(k)} - m\frac{f\left(x^{(k)}\right)}{f'\left(x^{(k)}\right)}.$$

In particolare:

$$\Phi_{\mathrm{NM}}(x) = x - m\frac{f(x)}{f'(x)} \quad \Rightarrow \quad \Phi'_{\mathrm{NM}}(x) = 1 - m\left[\frac{[f'(x)]^2 - f(x)f''(x)}{[f'(x)]^2}\right]$$

dove m è la molteplicità di α. Ovviamente occorrono dei modi per stimare questa molteplicità, ma essi esulano dall'ambito di questa trattazione.

8.4 Criteri di arresto

Sia $\left\{x^{(k)}\right\}_{k\geq 0}$ la successione di approssimazioni generata da uno dei metodi iterativi illustrati. Consideriamo l'errore $e^{(k)} = \alpha - x^{(k)}, \forall k$. Supponiamo $f \in C^1(I_\alpha)$, con I_α intorno di α.

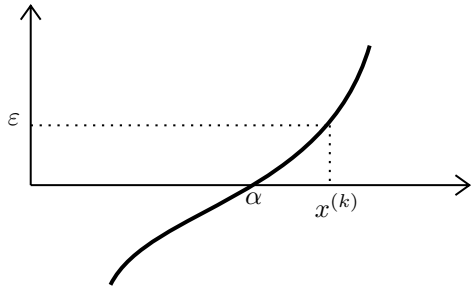
8.4.1 Controllo del residuo

Fissiamo una tolleranza $\varepsilon > 0$. Terminiamo il ciclo dell'algoritmo quando

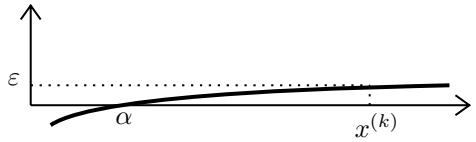
$$\left|f\left(x^{(k)}\right)\right| \leq \varepsilon.$$

In particolare, si può dimostrare che

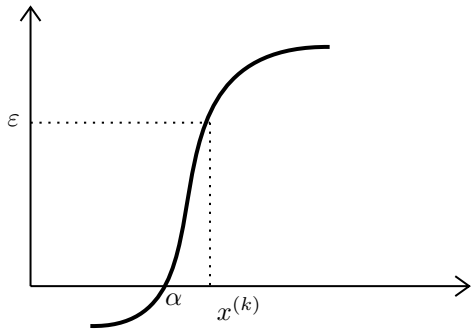
1. Se $|f'(\alpha)| \approx 1$ allora $\left|e^{(k)}\right| \approx \varepsilon$ e quindi il **criterio di arresto è affidabile**.



2. Se $|f'(\alpha)| \ll 1$ allora $\left|e^{(k)}\right| \gg \varepsilon$ e quindi il **criterio di arresto è inaffidabile**. Si noti in figura come l'errore sia molto piccolo, ma la soluzione sia ancora molto lontana dallo zero.



3. Se $|f'(\alpha)| \gg 1$ allora $\left|e^{(k)}\right| \ll \varepsilon$ e quindi il **criterio di arresto è troppo stringente**. Si noti in figura il contrario rispetto al punto precedente: l'errore è molto grande anche se la soluzione è molto vicina.



8.4.2 Controllo sull'incremento

Fissando la tolleranza $\varepsilon > 0$, terminiamo il ciclo quando

$$\left| x^{(k+1)} - x^{(k)} \right| < \varepsilon.$$

Grazie a uno sviluppo di Taylor possiamo scrivere

$$e^{(k+1)} = \alpha - x^{(k+1)} = \Phi(\alpha) - \Phi\left(x^{(k)}\right) = \Phi'\left(\xi^{(k)}\right) e^{(k)}$$

allora

$$x^{(k+1)} - x^{(k)} = e^{(k)} - e^{(k+1)} = \left(1 - \Phi'\left(\xi^{(k)}\right)\right) e^{(k)}.$$

Dato che $\xi^{(k)}$ sta convergendo ad α possiamo assumere che $\Phi'\left(\xi^{(k)}\right) \approx \Phi'(\alpha)$, da cui

$$e^{(k)} = \frac{1}{1 - \Phi'(\alpha)} \left(x^{(k+1)} - x^{(k)}\right). \tag{8.4}$$

Quindi:

- se $\Phi'(\alpha) \approx 1$ allora il criterio è *inaffidabile*;
- per i metodi del secondo ordine, se $\Phi'(\alpha) = 0$ allora il criterio è *affidabile*;
- continua ad essere soddisfacente se $-1 < \Phi'(\alpha) < 0$.

8.5 Sistemi di equazioni non lineari

Vediamo l'estensione vettoriale dello stesso tipo di problema visto nelle pagine precedenti. Sia assegnata $\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^n$: vogliamo trovare $\mathbf{x}^\star \in \mathbb{R}^n$ tale che $\mathbf{F}(\mathbf{x}^\star) = \mathbf{0}$.

Ricordiamo che la **matrice jacobiana** associata a $\mathbf{F}$ e valutata nel punto $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ è data da:

$$(J_{\mathbf{F}}(\mathbf{x}))_{ij} = \frac{\partial F_i}{\partial x_j}(\mathbf{x}), \quad i, j = 1, \ldots, n.$$

Supponiamo anche che $\mathbf{F}$ sia una funzione continua e con derivate parziali positive. Si ha il seguente algoritmo di punto fisso:

Algoritmo 21: Algoritmo di Newton per sistemi non lineari.

```
1 inizializza x^(0)
2 for k = 0, 1, 2, ... do
3 |   risolvi J_F(x^(k)) δx^(k) = −F(x^(k))
4 |   aggiorna x^(k+1) = x^(k) + δx^(k)
5 |   if criterio di arresto then
6 |   |   termina algoritmo
7 |   end
8 end
```

Osservazione. Esistono molte varianti di questo algoritmo, ad esempio basate su opportune approssimazioni di $J_{\mathbf{F}}\left(\mathbf{x}^{(k)}\right)$ oppure sull'aggiornamento di $J_{\mathbf{F}}\left(\mathbf{x}^{(k)}\right)$ solo una volta ogni k^* iterazioni, con $k^* > 1$.

Teorema 8.6 — Convergenza di Newton per sistemi non lineari. Sia $\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^n$, $\mathbf{F} \in C^1(D)$, dove D è un sottoinsieme aperto convesso di $\mathbb{R}^n$ che contiene $\mathbf{x}^\star$. Supponiamo inoltre che $J_{\mathbf{F}}(\mathbf{x}^\star)$ sia invertibile, e che esistano delle costanti positive R, C, L tali che $\left\|J_{\mathbf{F}}^{-1}(\mathbf{x}^\star)\right\| \leq C$ e

$$\|J_{\mathbf{F}}(\mathbf{z}) - J_{\mathbf{F}}(\mathbf{y})\| \leq L\|\mathbf{z} - \mathbf{y}\|, \quad \forall \mathbf{z}, \mathbf{y} \in B(\mathbf{x}^\star, R)$$

avendo indicato con lo stesso simbolo $\|\cdot\|$ una norma vettoriale ed una norma matriciale consistenti[a]. Esiste allora $r > 0$ tale che, per ogni $\mathbf{x}^{(0)} \in B(\mathbf{x}^\star, r)$ il metodo di Newton è univocamente definito, converge a $\mathbf{x}^\star$ ed è tale che

$$\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^\star\right\| \leq CL\left\|\mathbf{x}^{(k)} - \mathbf{x}^\star\right\|^2.$$

[a] Una norma matriciale $\|\cdot\|$ su $\mathbb{R}^{m\times n}$ è consistente con una norma vettoriale $\|\cdot\|_a$ su $\mathbb{R}^n$ e $\|\cdot\|_b$ su $\mathbb{R}^m$ se $\|A\mathbf{x}\|_b \leq \|A\|\|\mathbf{x}\|_a$ per tutte le matrici $A \in \mathbb{R}^{m\times n}$ e i vettori $\mathbf{x} \in \mathbb{R}^n$. Tutte le norme indotte, cioè quelle che consideriamo in questo corso, sono consistenti per definizione.

Appendice A

Richiami di algebra lineare

Il testo richiede la conoscenza di basilari concetti dall'algebra lineare. In questa appendice presentiamo alcuni utili richiami alla materia.

A.1 Vettori e spazi vettoriali

Definizione A.1 — Spazio vettoriale. Chiamiamo spazio vettoriale V su un campo K, solitamente con $K = \mathbb{R}$, un insieme non vuoto i cui elementi, detti vettori, sono dotati di un operatore di addizione $+ : V \times V \to V$ e un operatore di moltiplicazione per scalare $\cdot : K \times V \to V$. Questi operatori devono possedere le seguenti proprietà, per ogni scelta dei vettori $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{v}, \mathbf{w} \in V$ coinvolti:

- l'addizione è associativa, ovvero $\mathbf{a} + (\mathbf{b} + \mathbf{c}) = (\mathbf{a} + \mathbf{b}) + \mathbf{c}$, e commutativa, cioè $\mathbf{a} + \mathbf{b} = \mathbf{b} + \mathbf{a}$;
- esiste un vettore nullo $\mathbf{0} \in V$ tale che $\mathbf{v} + \mathbf{0} = \mathbf{v}$;
- $0 \cdot \mathbf{v} = \mathbf{0}$ e $1 \cdot \mathbf{v} = \mathbf{v}$;
- per ogni elemento $\mathbf{v}$ di V esiste ed appartiene a V il suo opposto $(-\mathbf{v})$, ovvero che sia tale che $\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$;
- vale la proprietà distributiva, ovvero $(\alpha + \beta)\mathbf{v} = \alpha\mathbf{v} + \beta\mathbf{v}$ e $\alpha(\mathbf{v} + \mathbf{w}) = \alpha\mathbf{v} + \alpha\mathbf{w}$ per ogni scelta degli scalari $\alpha, \beta \in K$.

Notiamo che queste proprietà sono simili a quelle possedute dai numeri scalari con cui normalmente si opera.

I vettori sono quindi *rappresentabili* come n-uple ordinate di numeri. Ai fini di

questo corso è utile vederli proprio come *colonne*, denotate da:

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}, \quad \mathbf{v} \in V.$$

Definizione A.2. Un insieme di vettori $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$ si dice linearmente indipendente se vale la seguente implicazione, con $\alpha_1, \ldots, \alpha_n \in K$:

$$\alpha_1 \mathbf{v}_1 + \ldots + \alpha_n \mathbf{v}_n = \mathbf{0} \quad \Rightarrow \quad \alpha_1, \ldots, \alpha_n = 0.$$

Definizione A.3 — Span. Dato V uno spazio vettoriale su K e un insieme di vettori di V: $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$, si dice *span* o *spazio generato da vettori* l'insieme di tutte le loro possibili combinazioni lineari, e si indica con:

$$\operatorname{span}(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n) \coloneqq \{a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \ldots + a_n\mathbf{v}_n \text{ con } a_1, a_2, \ldots, a_n \in K\}.$$

Per esempio, si può facilmente mostrare che i vettori $[1, 0]$ e $[2, 0]$, che sono elementi di $\mathbb{R}^2$, generano $\mathbb{R}$, essendo linearmente *dipendenti*. Invece, i vettori $[2, 0]$ e $[4, 5]$ generano $\mathbb{R}^2$, in quanto sono linearmente *indipendenti*.

Definizione A.4 — Prodotto scalare. Dato uno spazio vettoriale V su $\mathbb{R}$, un *prodotto scalare* in V è un operatore binario $(\cdot, \cdot) : V \times V \to \mathbb{R}$ tale che, per ogni $\mathbf{v}, \mathbf{w}, \mathbf{x} \in V$ e per ogni $\alpha, \beta \in \mathbb{R}$:

- $(\alpha\mathbf{v} + \beta\mathbf{w}, \mathbf{x}) = \alpha(\mathbf{v}, \mathbf{x}) + \beta(\mathbf{w}, \mathbf{x})$, cioè si ha *linearità* in entrambi gli operandi;
- $(\mathbf{x}, \mathbf{x}) \geq 0$;
- $(\mathbf{x}, \mathbf{x}) = 0$ se e solo se $\mathbf{x} = \mathbf{0}$.

Definizione A.5 — Prodotto scalare canonico in $\mathbb{R}^n$. Tra i vari prodotti scalari possibili per $\mathbb{R}^n$, quello definito *canonico* è il prodotto scalare euclideo. Esso è un operatore binario denotato dal simbolo "$\cdot$", e definito su $\mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ nel seguente modo:

$$\mathbf{u} \cdot \mathbf{v} \coloneqq u_1 v_1 + u_2 v_2 + \ldots + u_n v_n.$$

Nel seguito quando ci riferiremo al prodotto scalare in $\mathbb{R}^n$ indicheremo sempre quello canonico.

Definizione A.6 — Ortogonalità. Due vettori $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ sono ortogonali se

$$(\mathbf{x}, \mathbf{y}) = 0.$$

Definizione A.7 — Norma euclidea. Si dice norma euclidea di un vettore $\mathbf{x} \in \mathbb{R}^n$ l'operatore $\|\cdot\|_2 : \mathbb{R}^n \to \mathbb{R}$ tale che:

$$\|\mathbf{x}\|_2 \coloneqq \sqrt{x_1^2 + x_2^2 + \ldots + x_n^2}.$$

Definizione A.8 — Norma indotta. Dato un prodotto scalare $(\cdot, \cdot)$, la norma indotta da tale prodotto scalare è definita come:

$$\|\mathbf{x}\| \coloneqq \sqrt{(\mathbf{x}, \mathbf{x})}.$$

Essa rispetta tutte le proprietà richieste ad un operatore norma.

Per esempio, la norma euclidea è la norma indotta dal prodotto scalare canonico in $\mathbb{R}^n$:

$$\|\mathbf{x}\|_2 = \sqrt{\mathbf{x} \cdot \mathbf{x}}.$$

A.2 Matrici

Definizione A.9 — Matrice. Una matrice A di dimensioni $m \times n$ sul campo K è una tabella di mn elementi di K. I suoi elementi saranno indicati con la notazione (a_{ij}), dove i è l'indice della riga dell'elemento e j quello della sua colonna, con $1 \leq i \leq m$ e $1 \leq j \leq n$. Una matrice si rappresenta nel seguente modo:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

Per indicare una matrice su $\mathbb{R}$ di m righe e n colonne si dirà $A \in \mathbb{R}^{m \times n}$.

Definizione A.10 — Prodotto tra matrici. Date due matrici $A \in \mathbb{R}^{m \times p}$ e $B \in \mathbb{R}^{p \times n}$ (inclusi i casi in cui m, n e/o p siano uguali a 1), si dice prodotto righe per colonne, o semplicemente prodotto tra le due matrici, la matrice $C \in \mathbb{R}^{m \times n}$

definita per componenti da:

$$c_{ik} = \sum_{j=1}^{p} a_{ij} b_{jk}, \quad \forall i = 1, \ldots, m, \ \forall k = 1, \ldots, n.$$

In altre parole, la componente ik del risultato è il prodotto scalare canonico della riga i di A e della colonna k di B.

Evidenziamo che il numero p di colonne di A e di righe di B devono essere uguali. La definizione è ancora valida se una o entrambe le matrici coinvolte sono in realtà vettori ("orizzontali" oppure "verticali"), i quali possono essere visti come casi particolari di matrici con una dimensione unitaria.
Esempio. Consideriamo:

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad B = \begin{bmatrix} x & y & z \\ u & v & w \end{bmatrix},$$

il loro prodotto è

$$AB = \begin{bmatrix} ax + bu & ay + bv & az + bw \\ cx + du & cy + dv & cz + dw \end{bmatrix}.$$

Osserviamo che in generale $AB \neq BA$, infatti:

$$A = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \quad \Rightarrow \quad AB = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \neq \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} = BA.$$

Presentiamo ora una serie di comuni classificazioni per le matrici.

Definizione A.11 — Matrice quadrata. Una matrice $m \times n$ si dice *quadrata* se $m = n$, cioè se il numero di righe è uguale al numero di colonne.

Definizione A.12 — Matrice invertibile. Data una matrice quadrata, si dice *invertibile* se esiste una matrice B con lo stesso numero di righe e colonne tale che:

$$BA = AB = I,$$

dove I è la *matrice identità*, delle stesse dimensioni, costruita come $(I)_{ij} = \delta_{ij}$, cioè 1 se $i = j$ e 0 se $i \neq j$. La matrice inversa di A si indica con la notazione A^{-1}.

Definizione A.13 — Matrice trasposta. Data una matrice $A \in \mathbb{R}^{m \times n}$, si definisce la sua *matrice trasposta* la matrice $B \in \mathbb{R}^{n \times m}$ tale che $b_{ij} = a_{ji}$ per ogni indice i, j, ovvero ottenuta "ruotando" la matrice scambiando righe e colonne ma mantenendone l'ordine.

È facile dimostrare le seguenti proprietà della trasposta:

$$\left(A^T\right)^T = A, \quad (A+B)^T = A^T + B^T,$$
$$(AB)^T = B^T A^T, \quad (\alpha A)^T = \alpha A^T.$$

Inoltre, se A è invertibile si ha:

$$\left(A^T\right)^{-1} = \left(A^{-1}\right)^T =: A^{-T}.$$

Definizione A.14. Data una matrice quadrata $A \in \mathbb{R}^{n\times n}$:

- essa è **simmetrica** se $A = A^T$, cioè se $a_{ij} = a_{ji}$ per ogni $i, j = 1, \ldots, n$;
- essa è **diagonale** se $a_{ij} = 0$ per ogni $i \neq j$. La *matrice identità* è quindi un caso particolare di matrice diagonale;
- essa è **triangolare superiore** (risp. **inferiore**) se $a_{ij} = 0$ per ogni $i > j$ (risp. $i < j$).

Esempi. Ecco alcuni esempi delle tipologie di matrici appena presentate:

- simmetrica,
$$\begin{bmatrix} 3 & 2 & 17 & 34 \\ 2 & 5 & 1 & 8 \\ 17 & 1 & 9 & e \\ 34 & 8 & e & 22 \end{bmatrix}$$
- diagonale,
$$\begin{bmatrix} 13 & 0 & 0 & 0 \\ 0 & \pi & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 9 \end{bmatrix}$$
- triangolare superiore,
$$\begin{bmatrix} 1 & 4 & 0 & 11 \\ 0 & 29 & 2 & 15 \\ 0 & 0 & 6 & 7 \\ 0 & 0 & 0 & 5 \end{bmatrix}$$
- triangolare inferiore.
$$\begin{bmatrix} 5 & 0 & 0 & 0 \\ 7 & 14 & 0 & 0 \\ 0 & 8 & 80 & 0 \\ 4 & \pi & 9 & 66 \end{bmatrix}$$

Le matrici quadrate hanno due grandezze scalari fondamentali ad esse associate: il determinante e la traccia.

Definizione A.15 — Determinante. Data una matrice quadrata $A \in \mathbb{R}^{n\times n}$, si definisce *determinante* di A la seguente quantità scalare:

$$\det A \coloneqq \sum_{\sigma\in S_n} \operatorname{sgn}(\sigma) a_{1\sigma(1)} \dots a_{n\sigma(n)},$$

dove S_n indica l'insieme delle $n!$ permutazioni di $\{1,\dots,n\}$, e $\operatorname{sgn}(\sigma)$ indica il segno della permutazione, cioè 1 (risp. -1) se il numero di scambi con cui si ottiene è pari (risp. dispari).

Teorema A.16 — Sviluppo di Laplace. Consideriamo una matrice quadrata $A \in \mathbb{R}^{n\times n}$ e una sua sottomatrice $C_{ij} \in \mathbb{R}^{(n-1)\times(n-1)}$ ottenuta rimuovendo la i-esima riga e la j-esima colonna. Allora:

$$\det A = \sum_{j=1}^{n} (-1)^{i+j} a_{ij} \det(C_{ij}).$$

Nel caso molto comune di $n = 2$ il determinante si trova come:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad \det(A) = a_{11}a_{22} - a_{12}a_{21}$$

Teorema A.17. Date $A, B \in \mathbb{R}^{n\times n}$, valgono le seguenti proprietà:

$$\det(AB) = \det(A)\det(B), \quad \det\left(A^{-1}\right) = \frac{1}{\det(A)},$$
$$\det(A) = \det\left(A^T\right), \quad \det(\alpha A) = \alpha^n \det(A), \quad \forall \alpha \in K.$$

La prima proprietà è nota come teorema di Binet.

Definizione A.18 — Traccia. Si definisce *traccia* di una matrice quadrata $A \in \mathbb{R}^{n\times n}$ la somma degli elementi sulla diagonale:

$$\operatorname{tr}(A) \coloneqq \sum_{i=1}^{n} a_{ii}.$$

Esistono, infine, diversi spazi vettoriali associati ad una matrice.

Definizione A.19 — Rango. Si dice *rango* (in inglese *rank*) di una matrice A la dimensione dello spazio generato dalle sue colonne. Esso viene indicato

con $\text{Rank}(A)$.

Si può dimostrare che si ottiene lo stesso spazio se si usa lo spazio generato dalle righe invece che dalle colonne.

Definizione A.20 — Nucleo e immagine. Data una matrice $A \in \mathbb{R}^{n \times n}$, si dice *nucleo* o *kernel* di A il seguente spazio:

$$\text{Ker}(A) \coloneqq \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{0}\}$$

e *immagine* di A il seguente spazio:

$$\text{Im}(A) \coloneqq \{A\mathbf{x} : \mathbf{x} \in \mathbb{R}^n\}.$$

Teorema A.21. Valgono le seguenti proprietà:

$$\text{Rank}(A) = \text{Rank}\left(A^T\right), \quad \text{Rank}(A) = \dim(\text{Im}(A)),$$
$$\dim(\text{Im}(A)) + \dim(\text{Ker}(A)) = n.$$

$\dim(\text{Ker}(A))$ viene anche detta *nullità* della matrice. L'ultima proprietà porta pertanto il nome di teorema di **nullità più rango**.

Teorema A.22. Data $A \in \mathbb{R}^{n \times n}$, le seguenti affermazioni sono equivalenti:

- A è invertibile;
- $\det(A) \neq 0$;
- $\text{Ker}(A) = \{\mathbf{0}\}$;
- $\text{Rank}(A) = n$;
- Le righe (e le colonne) di A sono linearmente indipendenti.

A.2.1 Autovalori e autovettori

Le matrici quadrate posseggono un insieme di valori scalari particolarmente interessanti per l'analisi numerica, gli autovalori, e dei vettori ad essi associati, gli autovettori.

Definizione A.23 — Autovalore e autovettore. Consideriamo una matrice $A \in \mathbb{R}^{n \times n}$. Si dicono *autovalori* quei $\lambda \in \mathbb{R}$ tali che

$$A\mathbf{x} = \lambda\mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^n.$$

L'insieme degli autovalori è chiamato **spettro** di A, indicato con $\sigma(A)$.

È facile verificare che gli autovalori di una matrice A possono essere determinati calcolando le radici del suo **polinomio caratteristico**:

$$p_A(\lambda) = \det(A - \lambda I) = 0.$$

Questo è al più un polinomio di grado n, per cui vi saranno al massimo n autovalori distinti. Inoltre gli $\mathbf{x}$ che soddisfano $A\mathbf{x} = \lambda\mathbf{x}$ si dicono *autovettori*.

Teorema A.24. Una matrice $A \in \mathbb{R}^{n\times n}$ e la sua trasposta hanno gli stessi autovalori.

Teorema A.25. Gli autovalori di qualsiasi matrice quadrata A sono legati al suo determinante e traccia nel seguente modo:

$$\det(A) = \prod_{i=1}^{n} \lambda_i, \quad \mathrm{tr}(A) = \sum_{i=1}^{n} \lambda_i.$$

Definizione A.26 — Matrice definita positiva/negativa. Una matrice quadrata $A \in \mathbb{R}^{n\times n}$ si dice definita positiva (risp. negativa) se $\mathbf{x}^T A\mathbf{x} > 0$ (risp. < 0) per ogni $\mathbf{x} \neq \mathbf{0}$, $\mathbf{x} \in \mathbb{R}^n$.

Teorema A.27. Sia A una matrice quadrata di dimensione n. Le seguenti affermazioni sono equivalenti:

- A è *definita positiva* (*negativa*);
- tutti gli autovalori di A sono positivi (negativi);
- **Criterio di Sylvester**: tutti i determinanti dei minori Nord-Ovest[a] sono positivi (tutti i determinanti dei minori Nord-Ovest di ordine dispari sono negativi e i determinanti dei minori di testa di ordine pari sono positivi).

[a] Le sottomatrici di ordine k ottenute eliminando le ultime $(n-k)$ righe e colonne, con $0 \leq k \leq n-1$.

Definizione A.28. Due matrici A, B sono **simili** se esiste una matrice P invertibile, detta di passaggio, tale che:

$$P^{-1}AP = B.$$

Definizione A.29. Una matrice A è **diagonalizzabile** se è simile ad una matrice diagonale, cioè se esiste P invertibile tale che:

$$P^{-1}AP = \Delta,$$

con Δ matrice diagonale.

Teorema A.30. Una matrice quadrata di dimensione n è diagonalizzabile se e solo se ammette n autovettori linearmente indipendenti. Inoltre, una matrice reale simmetrica è sempre diagonalizzabile.

Definizione A.31 — Matrice ortogonale. Una matrice $U \in \mathbb{R}^{n\times n}$ è *ortogonale* se:

$$U^T U = I = UU^T.$$

Teorema A.32. Se $U \in \mathbb{R}^{n\times n}$ è una matrice quadrata ortogonale, allora:

- $\det(U) = \pm 1$;
- conseguentemente, U è invertibile, e in particolare $U^{-1} = U^T$;
- U^T è ortogonale;
- preserva il prodotto scalare canonico, nel senso che $(U\mathbf{x}, U\mathbf{y}) = (\mathbf{x}, \mathbf{y})$, con $\mathbf{x}, \mathbf{y} \in \mathbb{R}$. Infatti $(U\mathbf{x}, U\mathbf{y}) = \big(\mathbf{x}, U^T U\mathbf{y}\big) = (\mathbf{x}, \mathbf{y})$;
- preserva anche la norma euclidea $\|U\mathbf{x}\|_2 = \|\mathbf{x}\|_2$. Infatti $\|U\mathbf{x}\|_2^2 = (U\mathbf{x}, U\mathbf{x}) = (\mathbf{x}, \mathbf{x}) = \|\mathbf{x}\|_2^2$.

Elenco degli algoritmi

Indice analitico