UNIVERSITÀ CATTOLICA DEL SACRO CUORE

Facoltà di Scienze Matematiche, Fisiche e Naturali

ANALISI NUMERICA

Docenti: **Prof. Maurizio Paolini, Dr. Francesco Ballarin**

Questa dispensa è stata scritta, in LaTeX, da Mattia Garatti e raccoglie gli appunti delle lezioni del corso di Analisi Numerica tenuto dal Prof. Maurizio Paolini e dal Dr. Francesco Ballarin nell'anno accademico 2021/2022.

Realizzare una dispensa è un'operazione complessa, che richiede numerosi controlli. L'esperienza suggerisce che è praticamente impossibile realizzare un'opera priva di errori. Saremo quindi grati ai lettori che vorranno segnalare eventuali errori al seguente indirizzo di posta elettronica:

mattiagaratti@gmail.com

aggiungendo in copia conoscenza i docenti del corso in modo che aggiornino la versione disponibile su Blackboard.

Si ringraziano inoltre le seguenti persone, che hanno contribuito con integrazioni o correzioni apportate successivamente alla prima edizione:

- Francesco Ballarin, a.a. 2022/2023: aggiunta collegamenti con le esercitazioni.

Prima edizione 13 maggio 2022
Ultimo aggiornamento 11 ottobre 2022

Indice

Capitolo 0

L'analisi numerica

L'analisi numerica, o calcolo numerico, oppure ancora calcolo scientifico, si occupa di studiare problemi in cui si cerca una soluzione approssimata perché non è possibile determinarne analiticamente una esatta. Non si accettano quindi in questo ambito le risposte tipiche dell'Analisi Matematica (es. "la soluzione esiste"). Le due discipline sono comunque legate tra loro, da qui la somiglianza nel nome. Lo strumento usato per arrivare alla soluzione approssimata in analisi numerica è il calcolatore.

0.1 Calcolo numerico e calcolo formale

Il calcolo formale, o simbolico, può essere effettuato mediante diversi ambienti di lavoro come DERIVE, macsyma, maxima, gap, macaulay, magma oppure mathematica. In tutti questi casi parliamo di Computer Algebra System (CAS).

Per fare un esempio, calcolare la derivata prima della funzione $f(x) = \sin x$ rientra nel calcolo simbolico.

Esempi di ambienti di calcolo numerico sono invece MATLAB, octave oppure scilab.

Se volessimo pensare ad un problema risolubile mediante il calcolo numerico, e quindi le tecniche dell'Analisi Numerica, potremmo osservare il seguente.

(0.1.1) Esempio *Risolvere l'equazione*

$$x^5 - x = 1$$

Come possiamo osservare questa equazione è di quinto grado perciò non esiste una formula risolvente per radicali. Possiamo però cercare una soluzione approssimata.

Se volessimo pensare ad altri esempi di problemi risolubili mediante tecniche di analisi numerica potremmo citare le previsioni meteorologiche, analisi statica e dinamica, il comportamento di materiali innovativi, ecc...

Collegamento con le esercitazioni Esistono ambienti in cui sono disponibili sia strumenti di calcolo numerico sia strumenti di calcolo simbolico. Pur essendo il calcolo numerico l'obiettivo principale di questo corso, nelle esercitazioni useremo spesso anche alcune funzionalità di calcolo simbolico. Grazie all'organizzazione del linguaggio `Python` in librerie sarà immediatamente chiaro in quali casi ci avvarremo del calcolo numerico (tipicamente, con la libreria `numpy`) e in quali invece useremo calcolo simbolico (tipicamente, mediante la libreria `sympy`).

0.2 Fasi risolutive di un problema fisico

L'analisi numerica si occupa in particolare del secondo e terzo passaggio. Premettiamo già che a volte è necessario iterare queste fasi "aggiustando il tiro" per arrivare alla soluzione.

1) Modellizzazione Matematica[1]

L'uomo, nella sua indagine della natura si imbatte spesso in un **problema fisico** che vuole risolvere. Per fare ciò deve prima formalizzarlo mediante il linguaggio della matematica, rendendolo così un **problema matematico**. Per fare ciò si opera un'**idealizzazione** del problema[2]. Dall'idealizzazione nascono degli errori difficilmente quantificabili. Altri errori che nascono in questa fase sono gli **errori di misura**, questi invece quantificabili.

In generale possiamo rappresentarlo come

$$P(x, d) = 0$$

ovvero un'espressione matematica che lega le **incognite** x ai **dati del problema** d. Tuttavia può capitare, e nell'ambito dell'analisi numerica questi sono i casi di interesse, che il problema matematico non abbia una soluzione esplicita.

2) Discretizzazione/Troncamento/Passaggio in dimensione finita

Non avendo una soluzione esplicita il problema matematico viene trasformato in un **problema numerico**[3] che invece ha soluzione esplicita:

$$P_h(x_h, d_h) = 0$$

con $h > 0$ e "piccolo", $P_h \neq P$, $x_h \neq x$ e $d_h \neq d$. Una volta scelto h abbiamo quindi un'approssimazione dei dati e quindi una soluzione che non sarà la soluzione esatta, ma un'approssimazione di essa; viene perciò detta **soluzione approssimata**. Osserviamo che il legame stesso tra incognite e dati cambia. In questa fase si introducono nuovi errori.

Le principali caratteristiche di una soluzione sono le seguenti:

- **convergenza**, quando h tende a zero la soluzione approssimata tende alla soluzione esatta;
- **costo computazionale**, il tempo di calcolo necessario, e la memoria, a produrre la soluzione;
- **consistenza/stabilità**.

3) Individuazione di un algoritmo risolutivo

In questo passaggio si individua uno schema rigoroso di passaggi, una "formula risolvente", per arrivare alla soluzione approssimata.

In questa fase nascono **errori di arrotondamento**, rounding error, dovuti al fatto che il calcolatore non è in grado di eseguire le operazioni elementari in maniera esatta sui numeri reali in quanto $\mathbb{R}$ è infinito. In alcuni contesti possono essere disastrosi, in questo caso l'algoritmo si dice **instabile**.

[1] Se ne occupa la Fisica Matematica.
[2] Lo si "semplifica" perché troppo complesso.
[3] Spesso è un sistema lineare.

4) Implementazione ed esecuzione

Una volta ideato un'algoritmo risolutivo esso viene implementato mediante un linguaggio di programmazione: viene così prodotto un **codice di calcolo** che se eseguito produce la **soluzione numerica approssimata**.

In questa fase possono nascere **errori di programmazione**; in passato durante l'esecuzione potevano nascere errori, al giorno d'oggi possiamo affermare che non ci siano errori di esecuzioni.

(0.2.1) Esempio "Consideriamo un filo elastico sottoposto ad uno sforzo. Calcoliamo la deformazione del filo."

Questo è un esempio di problema fisico, ambito di studi della fisica matematica. Le idealizzazioni fatte sul problema sono le seguenti: elasticità lineare, filo sottile, piccola deformazione, ecc...

Mediante la modellizzazione si arriva a un problema matematico di questo tipo:

$$\begin{cases} -u''(x) = f(x)\ ,\ x \in (a,b) \\ u(a) = u(b) = 0 \end{cases}$$

Utilizzando la ***tecnica delle differenze finite*** *convertiamo il problema matematico in un problema numerico.*

1. *griglia, suddividiamo l'intervallo (a,b) in intervalli più piccoli; dato $n \in \mathbb{N}$, si costruisce cioè una suddivisione dell'intervallo di ampiezza $h = \dfrac{b-a}{n}$. I nodi risulteranno essere i seguenti*
$$a = x_0 < x_1 < \cdots < x_n = b$$
con $x_i = a + ih$.

2. *collocazione, detto $u_i := u(x_i)$, collochiamo l'equazione differenziale sulla griglia costruita al passaggio precedente:*
$$\begin{cases} -u_i'' = f(x_i)\ ,\ i = 1, \ldots, n-1 =: \tilde{n} \\ u_0 = u_n = 0 \end{cases}$$

3. *approssimazione, mediante lo sviluppo di Taylor approssimiamo il valore di $-u''(x)$:*
$$u(x+h) = u(x) + hu' + \frac{h^2}{2}u''(x) + \frac{h^3}{6}u'''(x) + O(h^4)$$
$$u(x-h) = u(x) - hu' + \frac{h^2}{2}u''(x) - \frac{h^3}{6}u'''(x) + O(h^4)$$
da cui, sommando entrambi i membri otteniamo
$$u(x+h) + u(x-h) = 2u(x) + h^2u''(x) + O(h^4)$$
allora risulta
$$\frac{2u(x) - u(x+h) - u(x-h)}{h^2} = -u''(x) + O(h^2).$$

Possiamo quindi scrivere

$$-u''(x) \approx \frac{2u(x) - u(x+h) - u(x-h)}{h^2}$$

In questo modo abbiamo trovato un'approssimazione di $-u''(x)$ in cui sono coinvolti solo i valori nodali, abbiamo perciò discretizzato il problema. Possiamo ora scrivere

$$2u_i - u_{i-1} - u_{i+1} \approx h^2 f(x_i) =: b_i \ , \ i = 1, ..., \tilde{n}$$

Scrivendo $=$ al posto di $\approx$, avremo delle soluzioni approssimate, diverse da quelle esatte che indichiamo con U_i

$$2U_i - U_{i-1} - U_{i+1} = b_i \ , \ i = 1, \dots, \tilde{n}$$

Ciò che abbiamo trovato è un sistema lineare di $\tilde{n}$ equazioni in $\tilde{n}$ incognite

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

con A la matrice dei coefficienti, $\mathbf{U}$ il vettore di componenti U_i e $\mathbf{b}$ il vettore di componenti b_i. Rappresentiamo la matrice A:

$$\begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 2 & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix}$$

Si osserva che la matrice è simmetrica, definita positiva e tridiagonale[4]*, perciò è non singolare quindi il sistema ammette una e una sola soluzione, cioè $\exists!$ $\mathbf{U}$. Notiamo anche che detta n la dimensione di A, $n = O(\frac{1}{h})$*[5]*, perciò al diminuire di h aumenta la dimensione della matrice.*

Un modo comodo per risolvere un sistema di questo tipo è l'eliminazione di Gauss che in questo caso ha un costo computazionale c.c. $= O(n)$.

A questo livello rimane una domanda: riusciamo a stimare l'errore commesso?

Analizziamo brevemente un secondo esempio di problema fisico.

(0.2.2) Esempio *Consideriamo un membrana elastica, bloccata su un contorno piano, sottoposta ad un carico. Calcoliamo la deformazione.*

Il problema è simile al precedente, solo che ora siamo in D-2. Il problema matematico che nasce dalla modellizzazione stavolta è il seguente:

$$\begin{cases} - \triangle u(x,y) = f(x,y) \ , \ (x,y) \in \Omega \\ u|_{\partial\Omega} = 0 \end{cases}$$

[4] Torneremo più avanti su queste definizioni.
[5] Quindi è proporzionale a $\frac{1}{h}$.

Supponiamo $\Omega = (0,1) \times (0,1)$, cioè che il dominio sia un quadrato. Sia $n \in \mathbb{N}$ e $h = \frac{1}{n}$. Potendo operare la seguente approssimazione

$$-\triangle u(x,y) \approx \frac{4u(x,y) - u(x-h,y) - u(x+h,y) - u(x,y-h) - u(x,y+h)}{h^2}$$

risulta, operando analogamente a prima,

$$4U_i - U_{i-1} - U_{i+1} - U_{i-\tilde{n}} - U_{i+\tilde{n}} = b_i \ , \ i = 1, \ldots, \tilde{n}^2 =: N.$$

Anche qui arriviamo ad avere un sistema lineare $N \times N$, in cui la matrice A gode di alcune proprietà: in particolare essa è simmetrica, definita positiva e sparsa (ma non tridiagonale); in questo caso parliamo di matrice a banda. Si può risolvere con l'eliminazione di Gauss ma qui risulta più efficace un metodo iterativo.

Notiamo quindi che problemi all'apparenza simili possono avere algoritmi risolutivi molto diversi tra loro.

Collegamento con le esercitazioni L'Esercizio 01.1 introduce come memorizzare vettori e matrici in `Python` mediante la libreria `numpy`. La matrice A e il vettore $\mathbf{b}$ riportati nell'Esempio (0.2.1) possono essere quindi effettivamente memorizzati in un ambiente di calcolo. Si veda l'Homework 02.2.

Capitolo 1

Gli errori

1.1 Errore assoluto e relativo

(1.1.1) Definizione *Dato un valore esatto x e $\tilde{x} \approx x$ una sua approssimazione, possiamo definire l'errore*

$$e := x - \tilde{x}.$$

(1.1.2) Definizione *Dato un valore esatto x e $\tilde{x} \approx x$ una sua approssimazione, definiamo* ***errore assoluto****, ε_{abs}, ed* ***errore relativo****, ε_{rel}*[1]*:*

$$\varepsilon_{abs}(\tilde{x}) = |x - \tilde{x}| \ , \ x \in \mathbb{R}$$

$$\varepsilon_{rel}(\tilde{x}) = \frac{|x - \tilde{x}|}{|x|} \ , \ x \in \mathbb{R} \setminus \{0\} .$$

(1.1.3) Definizione *Dato un valore esatto x e $\tilde{x} \approx x$ una sua approssimazione, definiamo* ***maggiorazione dell'errore*** *una stima scritta in una di queste due modalità:*

$$\varepsilon_{abs}(\tilde{x}) \leq C_1$$

$$\varepsilon_{rel}(\tilde{x}) \leq C_2.$$

Rispettivamente avremo una maggiorazione dell'errore assoluto o una maggiorazione dell'errore relativo.

(1.1.4) Osservazione *L'errore assoluto è una quantità dimensionale, cioè possiede un'unità di misura. $\varepsilon_{REL}(\tilde{x})$ è invece adimensionale perciò preferibile rispetto a $\varepsilon_{ABS}(\tilde{x})$ perché dipende, per questo motivo, meno dal contesto.*

[1] Spesso è indicato in termini percentuali.

1.2 Proprietà del problema matematico

Dato un problema matematico $P(x, d) = 0$, possiamo individuare tre proprietà fondamentali:

- **buona posizione**, se $x = f(d)$ con f continua, cioè se esiste un'unica soluzione in dipendenza continua dai dati;
- "**stabilità**"[2], se $\tilde{d} \approx d \implies \tilde{x} = f(\tilde{d}) \approx x$, se $\exists K > 0 : \varepsilon_{REL}(\tilde{x}) \leq K\varepsilon_{REL}(\tilde{d})$
- **buon condizionamento**, se[3] K è non troppo grande.

Nell'ambito dell'analisi numerica possiamo risolvere problemi matematici che hanno queste tre proprietà.

Collegamento con le esercitazioni Non tutti i problemi matematici soddisfano queste proprietà. In particolare, l'Esercizio 03.1 mostra come la ricerca degli zeri di un polinomio possa essere un problema mal condizionato.

1.3 Indice di condizionamento

(1.3.1) Teorema *Siano x il dato reale ed y la soluzione reale di un problema matematico $y = f(x)$. Allora*

$$K_f(x) \approx \left| \frac{xf'(x)}{f(x)} \right|.$$

Dimostrazione. Sia $\tilde{x} \approx x : \tilde{x} = x + \delta$ allora avremo $\tilde{y} \approx y : \tilde{y} = f(\tilde{x})$. Usando lo sviluppo di Taylor:

$$\tilde{y} = f(x + \delta) \approx f(x) + \delta f'(x) = y + \delta f'(x)$$

allora,

$$\tilde{y} - y \approx \delta f'(x)$$

perciò $\varepsilon_{abs}(\tilde{y}) = |y - \tilde{y}| \approx |f'(x)|\varepsilon_{abs}(\tilde{x})$ e quindi

$$K_f^{abs} \approx |f'(x))|.$$

Ora essendo $\varepsilon_{rel}(\tilde{y}) = \dfrac{|y - \tilde{y}|}{|y|}$ possiamo scrivere

$$\varepsilon_{rel}(\tilde{y}) \approx \frac{|f'(x)|\varepsilon_{abs}(\tilde{x})}{|f(x)|}$$

e quindi moltiplicando e dividendo per $|x|$

$$\varepsilon_{rel}(\tilde{y}) \approx \frac{|f'(x)|\varepsilon_{abs}(\tilde{x})}{|f(x)|} \cdot \frac{|x|}{|x|}$$

[2] Torneremo sul concetto di stabilità più avanti.

[3] Il numero K è detto *numero di condizionamento* o anche *indice di condizionamento.*

tuttavia, essendo $\frac{\varepsilon_{abs}(\tilde{x})}{|x|} = \varepsilon_{rel}(\tilde{x})$, possiamo scrivere

$$\varepsilon_{rel}(\tilde{y}) \approx \left|\frac{xf'(x)}{f(x)}\right|\varepsilon_{rel}(\tilde{x}).$$

Possiamo quindi concludere che

$$K_f(x) := K_f^{rel} \approx \left|\frac{xf'(x)}{f(x)}\right|. \blacksquare$$

(1.3.2) Esempio *Calcoliamo l'indice di condizionamento dei seguenti problemi:*

1. $f(x) = \sqrt{x}$;
2. $f(x) = \ln x\ ,\ x > 0$;

1. *Iniziamo determinando la derivata prima della funzione in esame, supponendo* $x > 0$[4]*:*

$$f'(x) = \frac{1}{2\sqrt{x}}$$

Ricordando la formula del teorema precedente risulta $K_f(x) \approx \left|\frac{x\frac{1}{2\sqrt{x}}}{\sqrt{x}}\right| = \frac{1}{2}$*. Concludiamo quindi dicendo che la radice quadrata è ben condizionata perché* K_f *non è troppo grande. Inoltre notiamo che non dipende dal dato del problema.*

2. *Iniziamo determinando la derivata prima della funzione in esame:*

$$f'(x) = \frac{1}{x}$$

Ricordando la formula del teorema precedente risulta $K_f(x) \approx \left|\frac{x\frac{1}{x}}{\ln x}\right| = \frac{1}{\ln x}$.

Concludiamo quindi dicendo che il logaritmo naturale è mal condizionato per $x \approx 1$.

(1.3.3) Teorema *Siano* x, y *i dati reali ed* z *la soluzione reale di un problema matematico* $z = f(x, y)$*. Allora*

$$K_f(x, y) \approx \left|\frac{x\frac{\partial f}{\partial x}}{f(x, y)}\right| + \left|\frac{y\frac{\partial f}{\partial y}}{f(x, y)}\right|.$$

Dimostrazione. Omettiamo la dimostrazione. ■

(1.3.4) Teorema *Siano* x *il dato reale ed* y *la soluzione reale di un problema matematico* $y = f(x)$*. Sia inoltre* z *la soluzione reale del problema matematico* $z = g(y)$*. Allora, detto* $h = g \circ f$*,*

$$K_h \approx K_f \cdot K_g.$$

[4]Verrebbe naturale usare $\geq$ tuttavia siccome lavoriamo in ambito relativo, dobbiamo anche supporre $x \neq 0$.

Dimostrazione.

$$K_h \approx \left|\frac{xh'(x)}{h(x)}\right| = \left|\frac{xf'(x)g'(y)}{g(y)}\right|$$

moltiplicando e dividendo per $|y|$

$$\left|\frac{xh'(x)}{h(x)}\right| = \left|\frac{xf'(x)g'(y)}{g(y)}\right| \cdot \left|\frac{y}{y}\right| = \left|\frac{xf'(x)}{f(x)}\right| \cdot \left|\frac{yg'(y)}{g(y)}\right|$$

da cui la tesi. ■

(1.3.5) Osservazione *Notiamo che in generale l'indice di condizionamento dipende dal regime del problema matematico, cioè dipende dai dati.*

Collegamento con le esercitazioni L'Esercizio 03.2 e l'Homework 03.1 discutono il calcolo dell'indice di condizionamento di alcune funzioni $\mathbb{R} \to \mathbb{R}$ o $\mathbb{R}^2 \to \mathbb{R}$.

Indice di condizionamento delle operazioni elementari

Moltiplicazione $f(x, y) = x \cdot y$

Essendo $\frac{\partial f}{\partial x} = y$ e $\frac{\partial f}{\partial y} = x$, risulta $K_\cdot \approx 2$.

Quindi è indipendente dal regime e piccolo, deduciamo che l'operazione è ben condizionata.

Divisione $f(x, y) = \frac{x}{y}$

Essendo $\frac{\partial f}{\partial x} = \frac{1}{y}$ e $\frac{\partial f}{\partial y} = -\frac{x}{y^2}$, risulta $K_\div \approx 2$.

Quindi è indipendente dal regime e piccolo, deduciamo che l'operazione è ben condizionata.

Sottrazione $f(x, y) = x - y$

Essendo $\frac{\partial f}{\partial x} = 1$ e $\frac{\partial f}{\partial y} = -1$, risulta $K_- \approx \dfrac{|x| + |y|}{|x - y|}$.

Quindi $K_- >> 1$ se $x \approx y$, da cui deduciamo che l'operazione può essere mal condizionata. L'errore che ne deriva è detto **errore di cancellazione**.

Analogamente possiamo calcolare l'indice di condizionamento dell'**addizione** e noteremo che è molto grande se $x \approx -y$.

(1.3.6) Esempio *Consideriamo l'equazione*

$$x^2 - 2px + q = 0$$

con $p, q > 0$, $p^2 - q > 0$, $p^2 >> q$. Sotto queste ipotesi l'equazione avrà due soluzioni reali positive. Consideriamo il problema matematico di determinare la minore, che indicheremo con x_1.

In questo caso osserviamo innanzitutto che il problema matematico coincide con il problema numerico e che esiste una formula risolvente

$$x_1 = p - \sqrt{p^2 - q}.$$

Da questa formula discende un algoritmo risolutivo in 4 operazioni elementari che può essere così schematizzato:

1. $p \longrightarrow p^2$ *(moltiplicazione)*
2. $\triangle = p^2 - q$ *(sottrazione)*
3. $\triangle \longrightarrow \sqrt{\triangle}$ *(radice quadrata)*
4. $p - \sqrt{\triangle}$ *(sottrazione)*

Siccome sappiamo che $p = \dfrac{x_1 + x_2}{2}$ *e* $q = x_1 \cdot x_2$ *analizziamo l'algoritmo che abbiamo costruito. Posto* $x_1 = \frac{1}{3}$ *e* $x_2 = 10^8$*, otteniamo* $p \approx 50000000,17$ *e* $q \approx 33333333,33$*. Utilizziamo questi due valori di* p *e* q *nell'algoritmo e cerchiamo* x_1*:*

1. $p^2 \approx 2,500000017 \cdot 10^{15}$
2. $p^2 - q \approx 2,499999984 \cdot 10^{15}$
3. $\sqrt{\triangle} \approx 49999999,84$
4. $x_1 \approx 0,333335$

Usando l'algoritmo che abbiamo ideato risultano corrette solo le prime 5 cifre dopo la virgola, anche se ci saremmo aspettati almeno 10 cifre esatte.

Calcoliamo l'indice di condizionamento di questo problema $x_1 = f(p,q) = p - \sqrt{p^2 - q}$*. Essendo*

$$\frac{\partial f}{\partial p} = 1 - \frac{p}{\sqrt{p^2 - q}} = -\frac{p - \sqrt{p^2 - q}}{\sqrt{p^2 - q}}$$

e

$$\frac{\partial f}{\partial q} = \frac{1}{2\sqrt{p^2 - q}}$$

risulta

$$\left|\frac{p\frac{\partial f}{\partial p}}{f(p,q)}\right| = \left|\frac{p(p - \sqrt{p^2 - q})}{(p - \sqrt{p^2 - q})\sqrt{p^2 - q}}\right| = \frac{1}{\sqrt{1 - \frac{q}{p^2}}}$$

$$\left|\frac{q\frac{\partial f}{\partial q}}{f(p,q)}\right| = \left|\frac{q}{2(p - \sqrt{p^2 - q})\sqrt{p^2 - q}}\right| = \left|\frac{q}{2(p - \sqrt{p^2 - q})\sqrt{p^2 - q}}\right| \cdot \left|\frac{p + \sqrt{p^2 - q}}{p + \sqrt{p^2 - q}}\right| = \frac{p + \sqrt{p^2 - q}}{2\sqrt{p^2 - q}}$$

essendo $p^2 >> q$*,*

$$\left|\frac{q\frac{\partial f}{\partial q}}{f(p,q)}\right| \leq \frac{p}{\sqrt{p^2 - q}} = \frac{1}{\sqrt{1 - \frac{q}{p^2}}}.$$

Risulta quindi che $K_f \approx \dfrac{2}{\sqrt{1 - \frac{q}{p^2}}}$*.*

Essendo in un regime in cui $p^2 >> q$*,* $\frac{q}{p^2} \approx 0$ *e quindi* $K_f \approx 2$ *e perciò il problema matematico risulta ben condizionato. L'origine dell'errore deve essere un'altra, analizziamo un diverso algoritmo per comprendere quale sia.*

$$x_1 = (p - \sqrt{p^2 - q}) \cdot \frac{p + \sqrt{p^2 - q}}{p + \sqrt{p^2 - q}} = \frac{q}{p + \sqrt{p^2 - q}}$$

Rimaneggiando la formula risolvente siamo arrivati ad una sua versione equivalente che però ci fornisce un algoritmo differente, in 5 operazioni elementari

1. $p \longrightarrow p^2$ *(moltiplicazione)*
2. $\triangle = p^2 - q$ *(sottrazione)*
3. $\triangle \longrightarrow \sqrt{\triangle}$ *(radice quadrata)*
4. $D = p + \sqrt{\triangle}$ *(addizione)*
5. $\frac{q}{D}$ *(divisione)*

Mediante questo algoritmo si arriva alla seguente soluzione: $x_1 \approx 0,3333333333$. *Si vede fin da subito che il risultato è molto più accurato.*

Collegamento con le esercitazioni L'Homework 03.2 discute il calcolo dell'indice di condizionamento di una equazione polinomiale di grado ≥ 2. Procedendo in modo simile, l'Homework 03.3 considera il calcolo dell'indice di condizionamento delle radici dell'equazione di Lambert, un esempio di equazione non lineare e non polinomiale.

Risulta chiaro che i due algoritmi dell'Esempio (1.3.6), pur operando con gli stessi dati, producano soluzioni differenti. L'errore deriva dalla rappresentazione dei numeri reali all'interno del calcolatore.

1.4 Rappresentazione floating-point

Come ben sappiamo i numeri reali sono infiniti, tuttavia il calcolatore possiede una memoria finita perciò non è possibile rappresentarli tutti. Si è scelto di rappresentare un sottoinsieme di $\mathbb{R}$ detto **insieme dei numeri macchina**. Quando il risultato di un'operazione tra due numeri macchina non è un numero macchina il calcolatore lo approssima generando un errore.

(1.4.1) Teorema (di rappresentazione) *Sia* $\beta \in \mathbb{N}$, *pari, una base*[5]. *Siano* $0, 1, \ldots, \beta - 1$ *le cifre in base* β. *Valgono i seguenti fatti:*

- *se* $x \in [0, 1)$, $\exists!$[6] *sequenza infinita di cifre in base* β , $d_1, \ldots, d_n, \ldots$ *tale che*

$$x = d_1\beta^{-1} + \cdots + d_n\beta^{-n} + \cdots = \sum_{n=1}^{\infty} d_n\beta^{-n}.$$

  Si usa scrivere $x = (.d_1 d_2 \ldots d_n \ldots)_\beta$.

[5] Scelte tipiche sono $2, 8, 10, 12, 16, 20, 40, 60$.

[6] Diciamo unica perché non consideriamo rappresentazioni in cui da un certo punto in poi tutte le cifre sono $\beta - 1$.

- *se* $x \geq 1$, $\exists!\, e \in \mathbb{N} \,:\, \hat{x} := \frac{x}{\beta^e} < 1$ *e* $\frac{x}{\beta^{e-1}} \geq 1$ *tale che* $\hat{x} = (.d_1 \dots d_n \dots)_\beta$ *e quindi*

$$x = (.d_1 \dots d_n \dots)_\beta \cdot \beta^e.$$

Chiamiamo $(.d_1 \dots d_n \dots)_\beta$ ***mantissa*** *ed* e ***esponente***[7].

Dimostrazione. Omettiamo la dimostrazione. ■

(1.4.2) Definizione *Una rappresentazione si dice* ***normalizzata*** *se* $d_1 \neq 0$.

(1.4.3) Osservazione *Se* $\hat{x} \geq \frac{1}{\beta}$ *allora* $d_1 \neq 0$ *e quindi la rappresentazione è normalizzata.*

(1.4.4) Teorema *Sia* $\beta \in \mathbb{N}$, *pari, una base. Siano* $0, 1, \dots, \beta - 1$ *le cifre in base* β. *Se* $x \in \left(0, \frac{1}{\beta}\right)$, $\exists!\, e < 0 \in \mathbb{Z} \,:\, \hat{x} := \frac{x}{\beta^e} \geq \frac{1}{\beta}$ *e* $\hat{x} \in \left[\frac{1}{\beta}, 1\right)$. *Risulta inoltre che* $\hat{x}$ *ha una rappresentazione normalizzata ed essendo* $x = (.d_1 \dots d_n \dots)_\beta \cdot \beta^e$ *anche* x *ha una rappresentazione normalizzata.*

Dimostrazione. Omettiamo la dimostrazione. ■

Possiamo riassumere i casi visti in questi teoremi di rappresentazione in questo modo

(1.4.5) Teorema *Se* $x \in \mathbb{R} \setminus \{0\}$, $\exists!$ *segno* $\in \{+, -\}$, *un esponente* $e \in \mathbb{Z}$ *ed* $d_1, \dots, d_n, \dots$ *cifre in base* β *con* $d_1 \neq 0$*:*

$$x = \pm(.d_1 \dots d_n \dots)_\beta \cdot \beta^e$$

ovvero ogni numero non nullo ha una rappresentazione normalizzata.

Dimostrazione. Omettiamo la dimostrazione. ■

(1.4.6) Osservazione *In questa rappresentazione ci sono due "fonti" di* ∞*:*

1. $e \in \mathbb{Z}$ *che è un insieme infinito;*
2. *le cifre sono una sequenza infinita.*

(1.4.7) Definizione *Dati una base* β, *una precisione* $p \in \mathbb{N}$ *e due numeri* $L, U \in \mathbb{Z}$ *tali che* $L < U$ *e* $L \approx -U$, *chiamiamo* $\{\beta, p, L, U\}$ *un* ***sistema floating-point****.*

(1.4.8) Definizione *Dato un sistema floating-point* $\{\beta, p, L, U\}$, *chiamiamo* $M \subset \mathbb{R} \setminus \{0\}$ *l'insieme dei numeri macchina.*

Dato $x = (.d_1 \dots d_n \dots)_\beta \cdot \beta^e$ positivo, $x \in M$ se valgono le seguenti due proprietà[8]:

- $L \leq e \leq U$;
- $\forall n > p$, $d_n = 0$ quindi $x = \pm(.d_1 \dots d_p)_\beta \cdot \beta^e$.

[7] Chiamiamo questa scrittura **rappresentazione in base β di x.**

[8] M si addensa vicino a 0 e si dirada allontanandosi.

Avendo definito in questo modo M, l'errore di rappresentazione sarà uniforme per tutti i numeri. Inoltre, ci accorgiamo subito che i numeri macchina hanno una mantissa composta da p cifre.

Supponiamo ora che x non sia un numero macchina, cioè che $x \notin M$. Siamo di fronte a due possibili scenari:

- $e > U$ oppure $e < L$, in questa situazione siamo di fronte ad un **errore di overflow/underflow**;
- $L \leq e \leq U$ ma per $n > p$, $d_n \neq 0$

Nei sistemi moderni il primo caso non si verifica più, perciò immagineremo che l'errore di overflow/underflow non si verifichi mai. Diverso è il discorso per la seconda casistica, che in un calcolatore è abbastanza frequente. Per ovviare a ciò si approssima x ad un numero macchina, cioè si cerca $\bar{x} \in M \,:\, \bar{x} = fl(x)$ con $\bar{x} \approx x$. Esistono due strategie per produrre $\bar{x}$:

- **chopping** detta anche approssimazione per difetto, in cui $\bar{x} = (.d_1 \dots d_p)_\beta \cdot \beta^e$;

  Volendo effettuare una stima dell'errore possiamo così procedere, sapendo che $\bar{x} \leq x$

  $$\varepsilon_{abs}(\bar{x}) = x - \bar{x} = (.d_1 \dots d_n \dots)_\beta \cdot \beta^e - (.d_1 \dots d_p)_\beta \cdot \beta^e = (.\underbrace{0 \dots 0}_{p \text{ volte}} d_{p+1} \dots)_\beta \cdot \beta^e$$

  normalizzando ora la rappresentazione dell'errore assoluto otteniamo

  $$\varepsilon_{abs}(\bar{x}) = \underbrace{(.d_{p+1} \dots)_\beta}_{<1} \cdot \beta^{e-p} < \beta^{e-p}$$

- **rounding**[9], in cui se $d_{p+1} < \frac{\beta}{2}$ si esegue un chopping, mentre se $d_{p+1} \geq \frac{\beta}{2}$ si approssima per eccesso, cioè al numero macchina successivo.

  Volendo effettuare una stima dell'errore possiamo così procedere, consapevoli che l'errore assoluto con la strategia di rounding sarà nel caso peggiore pari alla metà dell'errore assoluto commesso con la strategia di chopping, ovvero

  $$\varepsilon_{abs}(fl(x)) \leq \frac{1}{2}\beta^{e-p}.$$

  Stimiamo l'errore relativo

  $$\varepsilon_{rel}(fl(x)) = \frac{|x - fl(x)|}{|x|} \leq \frac{1}{|x|} \cdot \frac{1}{2}\beta^{e-p}.$$

  Siccome per semplicità possiamo supporre $x > 0$,

  $$\varepsilon_{rel}(fl(x)) \leq \frac{1}{x} \cdot \frac{1}{2}\beta^{e-p}$$

  ed essendo $x = (.d_1 \dots d_p \dots)_\beta \cdot \beta^e$ normalizzato, $d_1 \neq 0$ e in particolare $d_1 > 0$, allora

[9] I processori moderni utilizzano questa tecnica.

$$x \geq (.100\ldots)_\beta \cdot \beta^e = \beta^{e-1}.$$

Possiamo quindi concludere che

$$\varepsilon_{rel}(fl(x)) \leq \frac{1}{2}\beta^{e-p} \cdot \beta^{1-e} = \frac{1}{2}\beta^{1-p}.$$

In definitiva l'errore relativo non dipende da x. Possiamo poi chiamare $\varepsilon_M := \frac{1}{2}\beta^{1-p}$ **errore macchina**; ε_M è uniforme.

Collegamento con le esercitazioni L'Homework 04.1 contiene una discussione sulla distribuzione dei numeri macchina sulla retta reale, ed in particolare del fatto che essi *non* sono equispaziati. L'Homework 04.2 invita a studiare una semplice rappresentazione macchina nel caso $\beta = 3$.

1.5 Lo standard IEEE-754

I calcolatori rappresentano i numeri reali secondo uno standard definito dall'Institute of Electrical and Electronics Engineers: lo IEEE-754.

Vediamo tre possibili varianti:

- **single precision**: sia un sistema floating-point così definito $\{2, 24, L, U\}$ con $U \approx 128$. Con un sistema di questo tipo sono richiesti 4 Byte di memoria per rappresentare un singolo numero macchina: 1 Byte è riservato per l'esponente, 1 bit per il segno ed i restanti 23 bit vengono utilizzati per rappresentare[10] le 24 cifre in base 2. L'errore macchina con questa rappresentazione sarà $\varepsilon_M = 2^{-24} \approx 10^{-6}$.

- **double precision**: sia un sistema floating-point così definito $\{2, 53, L, U\}$ con $U \approx 1024$. Con un sistema di questo tipo sono richiesti 8 Byte di memoria per rappresentare un singolo numero macchina: 11 bit sono riservati per l'esponente, 1 bit per il segno ed i restanti 52 bit vengono utilizzati per rappresentare le 53 cifre in base 2. L'errore macchina con questa rappresentazione sarà $\varepsilon_M = 2^{-53} \approx 10^{-15}$.

- **half precision**: senza entrare troppo nel dettaglio, con il sistema floating-point relativo sono richiesti 2 Byte di memoria per rappresentare un singolo numero macchina: 5 bit sono riservati per l'esponente, 1 bit per il segno ed i restanti 10 bit vengono utilizzati per rappresentare le 11 cifre in base 2.

Collegamento con le esercitazioni Le tre varianti double, single e half precision sono confrontate nell'Esercizio 04.1.

[10] Potrebbe a prima vista sembrare che ci sia un bit in meno, tuttavia la prima cifra, d_1, essendo la rappresentazione normalizzata, è diversa da zero, perciò essendo in base 2 non può essere se non $d_1 = 1$ per ogni numero; in virtù di questo non è necessario memorizzare la prima cifra e perciò lo spazio disponibile è sufficiente.

1.6 Operazioni elementari all'interno del calcolatore

Non esiste un insieme ben definito di operazioni elementari: possiamo però pensare che sicuramente questo insieme contiene le quattro operazioni aritmetiche, che sono operazioni binarie, e poi alcune in una variabile come la radice quadrata, il logaritmo, eccetera.

Chiamiamo la generica operazione elementare *op*; se l'operazione è binaria possiamo rappresentarla come una funzione

$$op : \mathbb{R} \times \mathbb{R} \to \mathbb{R}.$$

Nel calcolatore operiamo però con i numeri macchina perciò necessitiamo di una versione approssimata dell'operazione

$$\tilde{op} : M \times M \to M.$$

(1.6.1) Definizione *Dati due numeri macchina $\bar{x}, \bar{y} \in M$, il risultato di un'operazione binaria approssimata*[11] *$\tilde{op}$ viene così definito*[12]

$$\bar{x}\ \tilde{op}\ \bar{y} := fl(\bar{x}\ op\ \bar{y}) \in M.$$

(1.6.2) Esempio *Volendo fare un esempio di un'operazione approssimata possiamo prendere $\bar{x}, \bar{y} \in M$, $\bar{x} \oplus \bar{y} = fl(\bar{x} + \bar{y})$.*

Siano $\bar{x}, \bar{y} \in M$ ed *op* un'operazione elementare. Stimiamo l'errore commesso approssimandola:

$$\varepsilon_{rel}(\bar{x}\ \tilde{op}\ \bar{y}) = \frac{|\bar{x}\ op\ \bar{y} - \bar{x}\ \tilde{op}\ \bar{y}|}{|\bar{x}\ op\ \bar{y}|}$$

detto $z = \bar{x}\ op\ \bar{y}$, risulta che $fl(z) = \bar{x}\ \tilde{op}\ \bar{y}$, perciò possiamo riscrivere la precedente in questo modo

$$\frac{|z - fl(z)|}{|z|}$$

e maggiorarla con l'errore macchina come abbiamo stimato nel paragrafo precedente, quindi

$$\varepsilon_{rel}(\bar{x}\ \tilde{op}\ \bar{y}) \leq \varepsilon_M.$$

Ogni volta che un calcolatore esegue un'operazione elementare, avvengono due fatti:

1. viene propagato l'errore presente nei dati,tipicamente amplificandolo; si ottiene così un **errore propagato**.

2. viene generato un nuovo errore, comunque maggiorato dall'errore macchina, chiamato **errore generato**.

[11] In maniera analoga si definisce l'approssimazione di un'operazione in una variabile; se $op : \mathbb{R} \to \mathbb{R}$ e $x \in \mathbb{R}$ avremo che $\tilde{op}(\bar{x}) = fl(op(\bar{x}))$.

[12] Dallo standard dell'IEEE.

Supponiamo ora di avere un'espressione scritta in operazioni elementari che descrive l'algoritmo risolutivo di un problema matematico. Nel calcolatore tutte le operazioni che compaiono nell'espressione, vengono sostituite dalla loro versione approssimata tramite una funzione flt

$$\text{expr} \to flt(\text{expr})$$

(1.6.3) Esempio *Ricollegandoci all'esempio 2.10, data* $x_1 = p - \sqrt{p^2 - q}$*, la sua versione approssimata sarà* $flt(\bar{p} - \sqrt{\bar{p}^2 - \bar{q}}) = \bar{p} \ominus {}^*\sqrt{\bar{p} \odot \bar{p} \ominus \bar{q}} \neq flt\left(\dfrac{\bar{q}}{\bar{p} + \sqrt{\bar{p}^2 - \bar{q}}}\right)$*, siccome la versione approssimata di espressioni equivalenti non è equivalente.*

(1.6.4) Definizione *Chiamiamo* ***trasformazione residua*** *ogni singolo contributo all'errore di un'espressione matematica dovuto ad un'approssimazione legata ad un* ε*. L'errore complessivo di un'espressione sarà minore o uguale della somma delle trasformazioni residue presenti.*

(1.6.5) Definizione *Dato un algoritmo per la risoluzione di un problema matematico, chiamiamo* K_{alg} *l'****indice di condizionamento dell'algoritmo****.*

(1.6.6) Osservazione *Combinando la definizione 2.22 con la definizione 2.23 è chiaro che* K_{alg} *è pari alla somma delle trasformazioni residue relative all'espressione matematica dell'algoritmo.*

(1.6.7) Definizione *Un algoritmo relativo ad un problema matematico si dice* ***instabile*** *se l'indice di condizionamento dell'algoritmo è molto maggiore*[13] *dell'indice di condizionamento legato alla trasformazione residua del problema matematico.*

In altre parole la stabilità di un algoritmo è la misura della sensibilità dell'algoritmo rispetto all'errore di arrotondamento.

(1.6.8) Esempio *Consideriamo il problema matematico* $f(x) = \ln(1 + x)$ *in un regime con* $|x| << 1$*.*

Si tratta di un problema matematico in cui compaiono due operazioni elementari, il logaritmo e la somma. Siccome $1 \in M$ *qualunque sia la rappresentazione scelta, possiamo pensare alla somma come ad un'operazione in una variabile con un parametro macchina.*

Dato un valore esatto x*, per calcolare* $f(x)$ *dovrei prima sommare* 1 *e poi calcolare il logaritmo di* $1 + x$*: otterrei* $z = \ln(1 + x)$*.*

Per eseguire questo algoritmo con un calcolatore devo dapprima approssimare x *ad* $\bar{x}$*, commettendo un errore che posso stimare con l'errore macchina. A questo punto dovrei sommare a* $\bar{x}$ *il numero macchina* 1 *e poi fare il logaritmo del risultato, ottenendo* $\ln(1 + \bar{x})$*. L'errore commesso utilizzando* $\ln(1 + \bar{x})$ *al posto di* z *è maggiorabile con* $K_f \varepsilon_M$*, intendendo con* K_f *l'indice di condizionamento del problema matematico* $f(x)$*.*

Tuttavia nel calcolatore le operazioni vengono eseguite nella loro versione approssimata, perciò al posto di $1 + \bar{x}$ *avremo* $1 \oplus \bar{x}$*, commettendo un errore maggiorabile con l'errore macchina, e quindi* $\ln(1 \oplus \bar{x})$*. L'errore commesso utilizzando* $\ln(1 \oplus \bar{x})$ *al posto di* $\ln(1 + \bar{x})$

[13] In generale per come è definito K_{alg} comunque è sempre maggiore dell'indice di condizionamento del problema matematico (indicabile con K_f).

è maggiorabile con $K_{ln}\varepsilon_M$, intendendo con K_{ln} l'indice di condizionamento del logaritmo naturale.

Ricordiamo però che anche il logaritmo viene eseguito in versione approssimata e perciò avremo $\ln^*(1 \oplus \bar{x}) = \bar{z}$, commettendo un errore maggiorabile con l'errore macchina, ovvero l'indice di condizionamento in questo caso è quello dell'identità $K_{id} = 1$.

Riassumendo abbiamo tre trasformazioni residue legate all'algoritmo risolutivo:

1. *tutta l'espressione è una trasformazione residua impropria con indice di condizionamento K_f;*
2. *il logaritmo è una trasformazione residua propria con indice di condizionamento K_{ln};*
3. *l'identità finale è una trasformazione residua impropria con indice di condizionamento $K_{id} = 1$.*

Perciò avremo $K_{alg} = K_f + K_{ln} + K_{id}$.
Essendo ora

$$K_f \approx \left| \frac{x \frac{1}{1+x}}{\ln(1+x)} \right|$$

in un regime in cui $x << 1$, sviluppando anche il logaritmo con l'espansione di Taylor otteniamo $K_f \approx 1$; risulta invece

$$K_{ln} \approx \left| \frac{(1+x)\frac{1}{1+x}}{\ln(1+x)} \right| = \frac{1}{|\ln(1+x)|} >> 1$$

Quindi concludiamo che questo algoritmo è instabile.

Volendo analizzare invece un diverso algoritmo per la risoluzione dello stesso problema matematico potremmo invece ottenere risultati diversi: ad esempio se usassimo l'algoritmo

$$\ln(1+x) = \frac{x \ln(1+x)}{(1+x) - 1}$$

quest'ultimo risulterebbe un algoritmo stabile.

Volendo riprendere l'esempio (1.3.6), possiamo ora dire che il primo algoritmo utilizzato è instabile in quanto incappa nell'errore di cancellazione, mentre il secondo è stabile in quanto la somma di numeri uguali non è affetta da cancellazione. Risulta quindi ora chiaro perchè uno è preferibile all'altro.

Collegamento con le esercitazioni L'Esercizio 04.2 e gli Homework 04.3 e 04.4 contengono ulteriori semplici esempi di due funzioni che, pur essendo matematicamente equivalenti, restituiscono valutazioni diverse in aritmetica macchina per via di errori di cancellazione.

(1.6.9) Definizione *Chiamiamo **costo computazionale**, c.c., il numero di operazioni di un algoritmo, considerando solo le operazioni lunghe (quindi non addizione e sottrazione).*

(1.6.10) Definizione *Chiamiamo **flops** il numero di operazioni floating point.*

Capitolo 2

I sistemi lineari

Anche problemi che apparentemente non hanno niente a che vedere con l'algebra lineare alla fine quando vengono approssimati per essere risolti con un calcolatore richiedono la risoluzione di un sistema lineare. Saper risolvere sistemi lineare è quindi centrale nell'ambito dell'analisi numerica.

2.1 Alcuni richiami di algebra lineare

(2.1.1) Definizione *Sia $A \in Mat_{m\times p}$ e $B \in Mat_{p\times n}$. Chiamiamo **prodotto righe per colonne** l'operazione che ha come risultato una matrice $C \in Mat_{m\times n}$ e tale che $C = AB$ dove*

$$c_{ij} = \sum_{k=1}^{p} a_{i,k} b_{k,j},$$

cioè il prodotto scalare tra la i-esima riga della matrice A e la j-esima colonna della matrice B.

(2.1.2) Osservazione *Il prodotto righe per colonne tra una matrice ed un vettore colonna compatibile è un vettore colonna.*

(2.1.3) Definizione *Una matrice $A \in Mat_n := Mat_{n\times n}$ si chiama **matrice quadrata**.*

Esiste una particolare matrice quadrata, detta **matrice identica**, caratterizzata dall'avere tutti 1 sulla diagonale principale ed 0 altrove. Possiamo rappresentarla così

$$\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 \end{pmatrix}$$

La matrice identica risulta essere l'elemento neutro del prodotto righe per colonne.

(2.1.4) Definizione *Una matrice $T \in Mat_n$ si dice **triangolare superiore** (risp. **inferiore**) se gli elementi sotto (risp. sopra) la diagonale principale sono nulli.*

(2.1.5) Definizione *Chiamiamo* ***determinante*** *la funzione*

$$det() : \begin{cases} Mat_n \to \mathbb{R} \\ A \to det(A) = \sum_{\sigma \in S_n} \left(sgn(\sigma) \cdot \prod_{i=1}^{n} a_{i,\sigma(i)} \right) \end{cases}$$

che possiede le seguenti proprietà:

- ***antisimmetria****, cioè scambiando due righe (risp. colonne) il segno della funzione cambia;*
- ***multilinearità*** *rispetto alle colonne (risp. righe), cioè se pensiamo il determinante come funzione delle colonne (risp. righe), se consideriamo una colonna (risp. riga) che è combinazione lineare di due altre colonne (risp. righe) allora il determinante è combinazione lineare dei determinanti delle matrici che corrispondono alle colonne (risp. righe) che stiamo utilizzando per la combinazione lineare;*
- $det(Id) = 1$.

(2.1.6) Definizione *Una matrice $A \in Mat_n$ si dice* ***non singolare*** *se $det(A) \neq 0$.*

(2.1.7) Definizione *Una matrice $A \in Mat_n$ si dice* ***invertibile*** *se è non singolare. Chiamiamo A^{-1} la sua inversa.*

(2.1.8) Proposizione *Il determinante di una matrice T triangolare superiore (risp. inferiore) risulta essere*

$$det(T) = \prod_{i=1}^{n} t_{i,i} = t_{1,1} \cdot ... \cdot t_{n,n}$$

Dimostrazione. Dimostriamo innanzitutto che per matrici triangolari superiori risulta che se $\forall i, \sigma(i) \geq i$ allora $\sigma = Id$. Risulta infatti che $\sigma(n) = n$, $\sigma(n-1) = n oppure \sigma(n-1) = n-1$ ma essendo già assegnato n al passaggio precedente e ricordando che le permutazioni sono biunivoche può solo essere il secondo caso. Procedendo in questo modo è facile vedere che $\sigma = Id$. Da ciò segue che, ricordando la definizione di determinante,

$$\sum_{\sigma = Id} \prod t_{i,\sigma(i)} = \prod_{i=1}^{n} t_{i,i}$$

In maniera simile possiamo dimostrare la tesi per il caso di matrici triangolati inferiori. ■

(2.1.9) Teorema *Una matrice $T \in Mat_n$, triangolare, è invertibile se e solo se $\forall i, t_{i,i} \neq 0$.*

Dimostrazione. Basta combinare la definizione la definizione (2.1.6) con la proposizione (2.1.8). ■

(2.1.10) Definizione *Una matrice B si dice* ***convergente*** *se*

$$\lim_{k\to\infty} B^k = 0.$$

(2.1.11) Proposizione *Una matrice B è convergente se e solo se $\rho(B) < 1$* [1].

Dimostrazione. Omettiamo la dimostrazione. ■

(2.1.12) Osservazione *La norma di una matrice è sempre maggiore o uguale al suo raggio spettrale*

$$||B|| \geq \rho(B).$$

(2.1.13) Proposizione *Se $||B|| < 1$ allora la matrice B è convergente.*

Dimostrazione. Dalla precedente osservazione risulta banalmente che $\rho(B) < 1$ e quindi la matrice è convergente. ■

(2.1.14) Definizione *Chiamiamo* ***sistema lineare di*** m ***equazioni in*** n ***incognite*** *un sistema di equazioni di primo grado lineari del tipo*

$$\begin{cases} a_{1,1}x_1 + \cdots + a_{1,n}x_n = b_1 \\ \qquad\qquad \vdots \\ a_{m,1}x_1 + \cdots + a_{m,n}x_n = b_m \end{cases}$$

In sintesi, detta $A \in Mat_{m,n}$ la matrice dei coefficienti, $\mathbf{x} \in Mat_{n,1} = \mathbb{R}^n$ il vettore colonna delle incognite e $\mathbf{b} \in Mat_{m,1} = \mathbb{R}^m$ il vettore colonna dei termini noti

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

2.2 Condizionamento di una matrice

(2.2.1) Definizione *Una* ***norma*** *in $\mathbb{R}^n$ è una funzione scalare $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ con le seguenti proprietà:*

- $\|\mathbf{x}\| \geq 0, \qquad \forall \mathbf{x} \in \mathbb{R}^n$;
- $\|\mathbf{x}\| = 0$ *se e solo se* $\mathbf{x} = \mathbf{0}$;
- $\|\alpha\mathbf{x}\| = |\alpha| \, \|\mathbf{x}\|, \qquad \forall \alpha \in \mathbb{R}, \mathbf{x} \in \mathbb{R}^n$;
- $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|, \qquad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

[1] Il raggio spettrale di una matrice è il più grande valore assoluto degli autovalori della matrice.

(2.2.2) Definizione *Due norme* $\|\cdot\|, \|\cdot\|_*$ *in* $\mathbb{R}^n$ *sono* ***equivalenti*** *se esistono due costanti reali* $c_1, c_2 > 0$ *tali che*

$$c_1 \|\mathbf{x}\| \leq \|\mathbf{x}\|_* \leq c_2 \|\mathbf{x}\|, \qquad \forall \mathbf{x} \in \mathbb{R}^n.$$

(2.2.3) Definizione *Chiamiamo* ***norma-p*** *di un vettore, la seguente norma*

$$\|\mathbf{x}\|_p = \begin{cases} \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}} , \ p \in (0, \infty) \\ \max\limits_{i=1,\dots,n} |x_i| \ , \ p = \infty \end{cases}$$

(2.2.4) Definizione *Una* ***norma matriciale*** *in* $Mat_{m,n}$ *è una funzione scalare* $\|\cdot\|_\square : Mat_{m,n} \to \mathbb{R}$ *con le seguenti proprietà:*

- $\|A\|_\square \geq 0, \qquad \forall A \in Mat_{m\times n}$;
- $\|A\|_\square = 0$ *se e solo se* $A = \mathbf{0}$;
- $\|\alpha A\|_\square = |\alpha| \|A\|_\square, \qquad \forall \alpha \in \mathbb{R}, A \in Mat_{m\times n}$;
- $\|A + B\|_\square \leq \|A\|_\square + \|B\|_\square, \qquad \forall A, B \in Mat_{m\times n}$.

Inoltre una norma matriciale è detta ***sub-moltiplicativa*** *se*

$$\|AB\|_\square \leq \|A\|_\square \|B\|_\square, \qquad \forall A, B \in Mat_{m\times n}$$

e ***compatibile con una norma vettoriale*** *se*

$$\|A\mathbf{x}\| \leq \|A\|_\square \|\mathbf{x}\|, \qquad \forall A \in Mat_{m\times n}, \mathbf{x} \in \mathbb{R}^n.$$

Una norma compatibile è tipicamente denotata con lo stesso simbolo della norma vettoriale, omettendo il suffisso.

(2.2.5) Definizione *Sia una norma vettoriale* $\|\cdot\|$ *on* $\mathbb{R}^n$. *La corrispondente* ***norma matriciale indotta*** *in* $\mathbb{R}^{n\times n}$ *è così definita*

$$\|A\| = \sup_{\mathbf{x}\neq\mathbf{0}} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|}.$$

La norma indotta è per definizione compatibile con la corrispondente norma vettoriale. Sia ora una matrice quadrata $A \in Mat_n$: $A = [a_{ij}]_{i,j=1}^n$. La norma matriciale indotta per i casi $p = 1, 2, \infty$ può essere così scritta

$$\|A\|_1 = \max_{j=1,\dots,n} \sum_{i=1}^{n} |a_{ij}|$$

$$\|A\|_2 = \sqrt{\lambda_{\max}(A^T A)}$$

$$\|A\|_\infty = \max_{i=1,\dots,n} \sum_{j=1}^{n} |a_{ij}|$$

(2.2.6) Definizione *Data una norma matriciale indotta $\|\cdot\|$ in Mat_n l'**indice di condizionamento di una matrice** non singolare A è definito come*

$$K(A) = \|A\| \left\|A^{-1}\right\|.$$

Quando è definito usando una norma p di una matrice, l'indice di condizionamento solitamente si indica con $K_p(\cdot)$.

Collegamento con le esercitazioni

- L'Esercizio 05.1 mostra come sia possibile quantificare il valore delle costanti c_1, c_2 nella Definizione (2.2.2) per alcune coppie $\|\cdot\|, \|\cdot\|_*$ molto comuni. Grazie a ciò sarà anche possibile stabilire una nozione di equivalenza tra $K(\cdot)$ e $K_*(\cdot)$, definiti rispettivamente come gli indici di condizionamento rispetto alle norme indotte da $\|\cdot\|$ e $\|\cdot\|_*$.
- L'Esercizio 05.2 mostra un esempio di una matrice mal condizionata, la matrice di Hilbert.
- Le norme p sono molto utilizzate, ma altre norme sono certamente possibili. L'Homework 05.2 è relativo alla costruzione di una norma vettoriale che non rientra nei casi delle norme p, mentre l'Homework 05.1 introduce la norma di Frobenius, una norma matriciale che non è una norma indotta da una norma vettoriale.
- I concetti di determinante e di condizionamento di una matrice sono chiaramente diversi. L'Homework 05.4 mostra come costruire una famiglia di matrici le quali, pur avendo determinante unitario, hanno numero di condizionamento arbitrariamente grande.

2.3 Sistemi lineari quadrati

Nella nostra trattazione ci concentreremo su sistemi lineari quadrati, cioè sistemi lineare in cui la matrice associata è una matrice quadrata. Problemi in cui A non è quadrata, cioè detti m il numero di righe ed n il numero di colonne, $m \neq n$, sono **mal posti** e perciò non possiamo risolverli. Ci limiteremo al caso $m = n$.

(2.3.1) Teorema (di Rouché-Capelli) *Sia $A \in Mat_n$ la matrice dei coefficienti di un sistema lineare $A\mathbf{x} = \mathbf{b}$. $det(A) \neq 0$, cioè la matrice è non singolare, se e solo se $\exists!$ soluzione del sistema lineare.*

Dimostrazione. Omettiamo la dimostrazione. ∎

(2.3.2) Lemma *Se B è una matrice convergente allora valgono i seguenti fatti:*

1. *$Id - B$ è non singolare;*
2. *$(Id - B)^{-1} = Id + B + B^2 + \cdots + B^k + \cdots = \sum_{k=0}^{\infty} B^k$;*

3. se $||B|| < 1$ *allora* $||(Id - B)^{-1}|| \leq \sum_{k=0}^{\infty} ||B||^k = \frac{1}{1 - ||B||}$.

Dimostrazione. Dimostriamo i primi due fatti, il terzo è una banale conseguenza.

1. Per assurdo sia $(Id-B)$ singolare, allora considerando il sistema lineare $(Id-B)\mathbf{x} = 0$, esisterà una soluzione non nulla e quindi potremo scrivere

$$B\mathbf{x} = \mathbf{x}$$

cioè 1 è un autovalore della matrice. Ma allora $\rho(B) \geq 1$, assurdo.

2.

$$(Id-B)(I+B+B^2+\cdots+B^k) = Id+B+B^2+\cdots+B^k-(B+B^2+\cdots+B^{k+1}) = Id-B^{k+1}$$

per $k \to \infty$, $Id - B^{k+1}$ tende a Id essendo B convergente e quindi

$$(Id - B)^{-1} = I + B + B^2 + \cdots + B^k + \ldots$$

da cui la tesi. ■

Sistemi lineari triangolari

Consideriamo un generico sistema lineare triangolare superiore

$$\begin{cases} t_{1,1}x_1 + t_{1,2}x_2 + \cdots + t_{1,n}x_n = b_1 \\ \qquad\quad t_{2,2}x_2 + \cdots + t_{2,n}x_n = b_2 \\ \qquad\qquad\qquad\quad \vdots \\ \qquad\qquad\qquad\quad t_{n,n}x_n = b_n \end{cases}$$

Possiamo ricavare dall'ultima equazione x_n

$$x_n = \frac{b_n}{t_{n,n}}$$

Trovato x_n si può risalire un'equazione alla volta e trovare tutte le incognite. Questo algoritmo è detto **algoritmo di sostituzione all'indietro**. Di seguito riportiamo lo pseudo-codice:

Per $i = n, n-1, \ldots, 1$:

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} t_{i,j}x_j}{t_{i,i}}$$

Notiamo che per $i = n$ la sommatoria è vuota e restituisce 0 perciò il codice è coerente. Un'altra cosa da osservare è che questo algoritmo funziona se $\forall i, t_{i,i} \neq 0$, cioè se la matrice triangolare associata è invertibile.

Calcoliamo il costo computazionale dell'algoritmo: guardando lo pseudo-codice che abbiamo scritto abbiamo $\frac{n(n-1)}{2}$ moltiplicazioni (e altrettante somme e sottrazioni) ed n divisioni; allora

$$c.c. = \frac{1}{2}n^2 + O(n) \approx \frac{1}{2}n^2$$

ed invece flops ≈ 2 *c.c.*

Notiamo subito che il costo computazionale corrisponde circa al numero di termini non nulli della matrice: ciò indica che questo algoritmo è ottimale e non possiamo sperare di trovarne uno migliore.

In maniera simile possiamo trattare il caso di un sistema lineare triangolare inferiore: in questo caso l'algoritmo sarà detto **algoritmo di sostituzione in avanti**.

Collegamento con le esercitazioni L'Esercizio 06.1 discute una implementazione degli algoritmi di sostituzione, e una animazione grafica del processo di sostituzione.

2.4 L'algoritmo di eliminazione di Gauss

L'algoritmo di eliminazione di Gauss non restituisce la soluzione di un sistema lineare, bensì trasforma un generico sistema lineare quadrato in un sistema triangolare che potrà poi essere risolto mediante l'algoritmo visto in precedenza. In altre parole, dato un sistema lineare quadrato $A\mathbf{x} = \mathbf{b}$, l'algoritmo permette di passare a sistemi equivalenti[2] via via diversi fino ad arrivare ad un sistema triangolare; se $dim(A) = n$

$$A\mathbf{x} = \mathbf{b} \to A^{(2)}\mathbf{x} = \mathbf{b}^{(2)} \to \cdots \to A^n\mathbf{x} = \mathbf{b}^n$$

con A^n triangolare superiore.

Vediamo la sua implementazione più basilare. Al generico passo k, la matrice $A^{(k)}$ avrà la seguente struttura

$$\begin{pmatrix} a_{1,1} & a_{1,2} & \dots & \dots & \dots & a_{1,n} \\ 0 & \ddots & a_{2,k} & \ddots & \ddots & \vdots \\ \vdots & \ddots & a_{k,k} & \ddots & \ddots & \vdots \\ \vdots & \vdots & a_{k+1,k} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & a_{n,k} & \dots & \dots & a_{n,n} \end{pmatrix}$$

L'algoritmo azzererà gli elementi della k-esima colonna che stanno al di sotto della diagonale principale. Volendo scrivere lo pseudo-codice, risulta

per $k = 1, \dots, n-1$:
  per $i = k+1, \dots, n$:
    $m_{i,k} = \frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}}$
    $b_i^{(k+1)} = b_i^{(k)} - m_{i,k} b_k^{(k)}$
    per $j = k+1$[3], $...n$:

[2] Per ottenere un sistema equivalente si sottrae ad un'equazione un multiplo di un'altra.

[3] Si potrebbe inizialmente pensare che j debba andare da 1 a n, eppure se ci pensiamo un attimo questo sarebbe uno spreco di tempo di calcolo: le colonne prime della k-esima sono già state azzerate, perciò non è necessario azzerarle nuovamente; inoltre per come abbiamo definito $m_{i,k}$ non è necessario scandire la colonna k-esima perché risulta sicuramente 0.

$$a_{i,j}^{(k+1)} = a_{i,j}^{(k)} - m_{i,k} a_{k,j}^{(k)}$$

Ci accorgiamo subito che deve essere $\forall k, a_{k,k}^{(k)}$[4] $\neq 0$, affinché l'algoritmo funzioni. Tuttavia non è sufficiente supporre che la matrice sia non singolare per avere ciò: di conseguenza non possiamo sapere in anticipo se l'algoritmo funzionerà o meno. L'algoritmo può essere instabile, coinvolge un gran numero di operazioni ed ognuna genera un errore; mediante alcune **strategie pivotali** si può migliorare la situazione.

Calcoliamo il costo computazionale dell'algoritmo: partiamo dal costo del singolo passo k; fissato i abbiamo $(n-k+1)$ moltiplicazioni ed 1 divisione. Allora il singolo passo costa $(n-k+2)(n-k)$. In definitiva il costo complessivo è

$$\sum_{k=1}^{n-1} (n-k+2)(n-k)$$

e riscrivendo la sommatoria chiamando $n-k=k'$, consapevoli che questo è un indice muto e che quindi può benissimo essere chiamato k,

$$\sum_{k=1}^{n-1} k(k+2) = \sum_{k=1}^{n-1} k^2 + 2\sum_{k=1}^{n-1} k = \frac{n(n-1)(2n-2)}{6} + 2\frac{n(n-1)}{2} = \frac{1}{3}n^3 + O(n^2) \approx \frac{1}{3}n^3.$$

Il numero di flops sarà circa il doppio, cioè $\frac{2}{3}n^3$.

Collegamento con le esercitazioni L'Esercizio 06.1 discute una implementazione dell'eliminazione di Gauss (senza pivoting), e una animazione grafica del processo di eliminazione.

Strategie pivotali

(2.4.1) Proposizione *Se* $\det A \neq 0$ *allora esiste* $i_0 \geq k : a_{i_0,k}^{(k)} \neq 0$.

Dimostrazione. Per assurdo supponiamo che $\forall i \geq k \ : \ a_{i,k}^{(k)} = 0$. Consideriamo la matrice B formata dalle prime $k-1$ righe e k colonne della matrice $A^{(k)}$. Le colonne di B saranno vettori di $\mathbb{R}^{k-1}$ e quindi saranno linearmente dipendenti. Allora se consideriamo le prime k colonne di $A^{(k)}$ saranno linearmente dipendenti,perché siamo aggiungendo semplicemente degli zeri e quindi la combinazione lineare nulla a coefficienti non nulli delle colonne di B avrà gli stessi coefficienti della combinazione delle prime k colonne di $A^{(k)}$. Da ciò deriva che la matrice $A^{(k)}$ è singolare. Siccome l'eliminazione di Gauss non cambia il determinante ciò è assurdo. ■

Pivoting Parziale Prima del passo k, cerco $i_0 \geq k$: $\left|a_{i_0,k}^{(k)}\right| = \max_{i \geq k} \left|a_{i,k}^{(k)}\right|$[5]. Il pivoting parziale consiste nello scambiare l'equazione k-esima con la i_0-esima (compreso il termine noto). Ciò produce un sistema equivalente e in più mi permette di avere un elemento diverso da zero nella posizione pivotale evitando problemi con l'algoritmo di eliminazione di Gauss.

[4] Detti elementi pivotali.

[5] Avremmo potuto scrivere anche, più semplicemente $a_{i_0,k}^{(k)} \neq 0$. Tuttavia per motivi di stabilità questa è una scelta migliore.

(2.4.2) Osservazione $|m_{i,k}| \leq 1$, *ottimo risultato in termini di stabilità.*

(2.4.3) Osservazione *Il costo computazionale aggiuntivo del pivoting parziale rispetto al costo dell'algoritmo è trascurabile*

$$c.c.(\text{pivoting parziale}) = O(n^2) << \frac{1}{3}n^3$$

con $O(n^2)$ *il numero di confronti effettuati durante il pivoting parziale.*

Pivoting completo Prima del passo k, cerco l'elemento più grande in modulo in tutta la *regione attiva*, cioè la parte di matrice dove effettivamente, al passo k agisce l'algoritmo di eliminazione di Gauss. In altre parole cerco $i_0, j_0 \geq k : \left|a^{(k)}_{i_0,j_0}\right| = \max_{i,j\geq k}\left|a^{(k)}_{i,j}\right|$. Il pivoting completo consiste poi nello scambiare oltre alle righe, come nel caso del pivoting completo, anche le colonne della matrice.

(2.4.4) Osservazione *Il pivoting completo non produce un sistema equivalente perché cambia l'ordine delle incognite, ciò quindi porterà ad un diverso ordine delle soluzioni. Per ovviare al problema bisogna tenere traccia degli scambi e, una volta determinate le soluzioni, applicare gli scambi all'inverso.*

(2.4.5) Osservazione *Il costo computazionale aggiuntivo del pivoting completo rispetto al costo dell'algoritmo non è trascurabile, c.c.* $= O(n^3)$.

(2.4.6) Osservazione *Utilizzare il pivoting completo migliora la stabilità dell'algoritmo.*

Nelle situazioni concrete la strategia più utilizzata è comunque il pivoting parziale.

L'utilizzo di strategie di pivoting a volte può essere negativo: vedremo che non è possibile ottenere la fattorizzazione LU ad esempio. Risulta quindi utile capire se ci sono situazioni in cui è possibile evitare di utilizzare strategie di pivoting. Analizziamo ora due classi di matrici per cui non è necessario ricorrere a queste strategie:

- **matrici simmetriche definite positive**, cioè matrici A in cui si verificano questi due fatti:
  - $A = A^T$;
  - $\forall\, \mathbf{x} \in \mathbb{R}^n \setminus \{\mathbf{0}\} : \mathbf{x}^T A\mathbf{x} > 0$;
- **matrici fortemente diagonalizzate**, cioè matrici A in cui $\forall\, i\ ,\ |a_{i,i}| > \sum_{j\neq i}|a_{i,j}|$.

(2.4.7) Proposizione *Le matrici simmetriche definite positive e le matrici fortemente diagonalizzate sono non singolari.*

Dimostrazione. Sia A una matrice simmetrica definita positiva, supponiamo per assurdo che $\det A = 0$, allora, il sistema lineare omogeneo $A\mathbf{x} = \mathbf{0}$ non avrà un'unica soluzione. Allora esisterà $\mathbf{x} \neq \mathbf{0}$ soluzione del sistema. Allora $\mathbf{x}^T A\mathbf{x} = 0$, assurdo.

Sia ora una matrice A fortemente diagonalizzata, supponiamo per assurdo che $\det A = 0$ e consideriamo $A\mathbf{x} = \mathbf{0}$. Allora $\sum_{j=1}^{n} a_{i,j}x_j = 0$. Allora $-a_{i,i}x_i = \sum_{j\neq i} a_{i,j}x_j$. Sia $i_0 : |x_{i_0}| = \max_{i=1}^{n}|x_i| = ||x||_\infty > 0$ in quanto $\mathbf{x}$ è non nullo. Abbiamo quindi

$$|a_{i_0,i_0}|||x||_\infty \leq \sum_{j\neq i_0} |a_{i_0,j}||x_j| \leq \sum_{j\neq i_0} |a_{i_0,j}|||x||_\infty$$

perciò

$$|a_{i_0,i_0}|||x||_\infty \leq ||x||_\infty \sum_{j\neq i_0} |a_{i_0,j}|$$

e quindi

$$|a_{i_0,i_0}| \leq \sum_{j\neq i_0} |a_{i_0,j}|$$

che è assurdo perché la matrice è fortemente diagonalizzata. ■

(2.4.8) Proposizione *Se una matrice A è simmetrica definita positiva (risp. fortemente diagonalizzata) allora il minore principale di A di dimensione $k \times k$, detto A_k è simmetrico definito positivo (risp. fortemente diagonalizzato).*

Dimostrazione. La dimostrazione è banale. ■

(2.4.9) Teorema (di Sylvester) $\forall\, k, a_{k,k}^{(k)} \neq 0$ *se e solo se* $\forall\, k\ ,\ \det A_k \neq 0$.

Dimostrazione. Risulta che $\det A_k = \det A_k^{(k)}$ e $A_k^{(k)}$ è triangolare superiore. Ricordando come si calcola il determinante di una matrice triangolare superiore otteniamo la tesi. ■

Collegamento con le esercitazioni

- L'Esercizio 06.2 discute una implementazione dell'eliminazione di Gauss (con strategie pivotali), e una animazione grafica del processo di eliminazione.
- L'Esercizio 06.3 mostra un esempio concreto in cui il risultato di stabilità commentato nell'Osservazione (2.4.2) porta ad avere una soluzione notevolmente più accurata di un sistema lineare quando si utilizzi, in aritmetica macchina, una strategia di pivoting parziale per un sistema lineare dipendente da un parametro ε. Tuttavia, gli Homework 06.3 e 06.4 mostrano come, specialmente in casi estramamente patologici, nemmeno le strategie pivotali possano superare le limitazioni intrinseche dell'aritmetica macchina.
- È possibile ottenere una dimostrazione del fatto che strategie pivotali non sono necessarie nel caso di matrici simmetriche definite positive combinando gli Homework 06.1 e 06.2.

Casi particolari

Analizziamo alcuni casi di matrici con particolare struttura.

Matrici sparse sono matrici con un numero di elementi diversi da 0 molto minore del totale degli elementi, n^2. In generale l'eliminazione di Gauss rovina le matrici sparse: si verifica un fenomeno di riempimento, **fill-in**, secondo cui elementi che al passo k erano nulli al passo $k+1$ risultano diversi da 0. In generale è quindi sconsigliabile utilizzare l'eliminazione di Gauss con matrici sparse generiche.

Matrici a banda p, q sono matrici sparse con q diagonali superiori e p diagonali inferiori non vuote e in genere $p, q << n$. Cioè del tipo

$$\begin{pmatrix} * & * & \dots & * & 0 & \dots & 0 \\ * & \ddots & \ddots & & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & & \ddots & 0 \\ * & & \ddots & \ddots & \ddots & & * \\ 0 & \ddots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & & \ddots & \ddots & * \\ 0 & \dots & 0 & * & \dots & * & * \end{pmatrix}$$

Con queste matrici, l'eliminazione di Gauss, senza strategie pivotali, funziona bene: cioè $\forall\, k, A^{(k)}$ sarà a banda p, q. La struttura rimane quindi invariata. Ciò si verifica perché in questo caso la regione attiva è una matrice con p colonne e q righe.

Il costo computazionale al passo k è circa $p \cdot q$ e quindi, *c.c.* $\approx n \cdot p \cdot q$. Nella nostra ipotesi che $p, q << n$ risulta che *c.c.* $<< n^3$.

Un caso notevole di matrici a banda è quello di **matrici tridiagonali**, ovvero matrici a banda con $p = q = 1$. In questo caso *c.c.* $= 3n$.

Collegamento con le esercitazioni L'Esercizio 07.3 mostra un esempio concreto di una matrice che è affetta dal fenomeno di fill-in. Per lo stesso esempio, l'Homework 07.4 discute un semplice modo di evitare che tale fenomeno accada.

2.5 Fattorizzazioni LU

(2.5.1) Definizione *Diciamo che LU è una* ***fattorizzazione LU*** *di una data matrice $A \in Mat_n$, non singolare, se*

- *L è triangolare inferiore;*
- *U è triangolare superiore;*
- *$A = LU$.*

Il prossimo teorema garantisce l'esistenza di una fattorizzazione LU a partire dall'algoritmo di eliminazione di Gauss.

(2.5.2) Teorema ***(di esistenza di una fattorizzazione LU)*** *Se $A^{(n)}$ è la matrice triangolare superiore risultate dall'applicazione dell'algoritmo di eliminazione di Gauss ad una matrice A. Sia la matrice triangolare inferiore dei moltiplicatori $\hat{M} \in Mat_{k+1,n-1}$ così costituita*

$$\hat{M} = \begin{pmatrix} 1 & & & \\ & \ddots & & 0 \\ & m_{i,j} & \ddots & \\ & & & \ddots \\ & & & & 1 \end{pmatrix}.$$

Allora risulta $\hat{M}A^{(n)} = A$.

Dimostrazione. $\forall i, \forall j$ risulta che, ricordando la definizione di prodotto righe per colonne,

$$(\hat{M}A^{(n)})_{i,j} = \sum_{k=1}^{n} \hat{m}_{i,k} a_{k,j}^{(n)}$$

essendo poi $\hat{M}$ triangolare inferiore, i termini della sommatoria dal $i+1$-esimo in poi sono nulli perché nullo è $\hat{m}_{i,k}$, perciò

$$\sum_{k=1}^{n} \hat{m}_{i,k} a_{k,j}^{(n)} = \sum_{k=1}^{i} \hat{m}_{i,k} a_{k,j}^{(n)} = \hat{m}_{i,i} a_{i,j}^{(n)} + \sum_{k<i} \hat{m}_{i,k} a_{k,j}^{(n)}$$

ora però $\hat{m}_{i,i} = 1$, per come abbiamo definito la matrice, e $a_{i,j}^{(n)} = a_{i,j}^{(i)}$, siccome durante l'eliminazione di Gauss le righe i-esime, dopo il passo i non vengono più modificate; inoltre se $k < i$, $\hat{m}_{i,k}$ sta sotto la diagonale e quindi $\hat{m}_{i,k} = m_{i,k}$. Perciò

$$(\hat{M}A^{(n)})_{i,j} = a_{i,j}^{(i)} + \sum_{k<i} m_{i,k} a_{k,j}^{(n)}$$

ora, procedendo in modo analogo a prima, $a_{k,j}^{(n)} = a_{k,j}^{(k)}$ e quindi, ricordando come opera l'algoritmo di eliminazione di Gauss,

$$(\hat{M}A^{(n)})_{i,j} = a_{i,j}^{(i)} + \sum_{k<i} m_{i,k} a_{k,j}^{(k)} = a_{i,j}^{(i)} + \sum_{k<i} (a_{i,j}^{(k)} - a_{i,j}^{(k+1)}).$$

La sommatoria che deriva è telescopica e quindi

$$(\hat{M}A^{(n)})_{i,j} = a_{i,j}^{(i)} + a_{i,j}^{(1)} - a_{i,j}^{(i)} = a_{i,j}^{(1)}.$$

In definitiva risulta quindi $\hat{M}A^{(n)} = A$. ■

(2.5.3) Osservazione *Se avessimo operato delle strategie pivotali durante l'applicazione dell'algoritmo di eliminazione di Gauss, non saremmo arrivati al risultato del teorema precedente. Quindi l'esistenza di una fattorizzazione LU derivata dall'eliminazione di Gauss è subordinata al non utilizzo di strategie pivotali.*

(2.5.4) Teorema (di "unicità" della fattorizzazione LU) *La fattorizzazione LU di una matrice* $A \in Mat_n$*, non singolare, è unica a meno di una matrice diagonale D.*

Dimostrazione. Se D è una matrice diagonale non singolare e $LU = A$ una fattorizzazione LU di A, risulta

$$A = LU = L \cdot Id \cdot U = L \cdot DD^{-1} \cdot U = (LD) \cdot (D^{-1}U)$$

e, sapendo che il prodotto di una matrice triangolare superiore (risp. inferiore) per una matrice diagonale da ancora una matrice triangolare superiore (risp. inferiore), dette $\tilde{L} := LD$, triangolare inferiore, ed $\tilde{U} := D^{-1}U$, possiamo scrivere una nuova fattorizzazione LU di A come

$$A = \tilde{L}\tilde{U}.$$

Ora se $A = L_1U_1 = L_2U_2$, per la regola di Binet e per la legge di annullamento del prodotto risulta che tutti i determinanti delle matrici L_1, U_1, L_2, U_2 sono non nulli, essendo A non singolare. Le matrici sono quindi invertibili.

Risulta

$$L_1U_1 = L_2U_2$$
$$L_1^{-1} \cdot L_1U_1.U_2^{-1} = L_1^{-1} \cdot L_2U_2 \cdot U_2^{-1}$$
$$U_1.U_2^{-1} = L_1^{-1} \cdot L_2.$$

Sapendo che l'inversa di una matrice triangolare superiore (risp. inferiore) è ancora triangolare superiore (risp. inferiore) e che il prodotto di matrici triangolari superiori (risp. inferiori) è ancora triangolare superiore (risp. inferiore), possiamo porre $D := U_1.U_2^{-1} = L_1^{-1} \cdot L_2$, siccome l'unico modo che due matrici triangolari non singolari, una superiore e l'altra inferiore, siano uguali è che esse siano diagonali.

Perciò

$$L_2 = L_1 \cdot D \ , \ U_2 = D^{-1} \cdot U_1$$

e quindi $L_2U_2 = L_1 \cdot DD^{-1} \cdot U_1$. ■

(2.5.5) Osservazione *Una fattorizzazione LU è utile per risolvere un sistema lineare. Se $A = LU$, allora abbiamo*

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

da cui detto $\mathbf{y} = U\mathbf{x}$, posso costruire due sistemi lineari

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

che se risolti in questo ordine mi danno la soluzione del sistema lineare di partenza $A\mathbf{x} = \mathbf{b}$.

Ricordando che L è una matrice triangolare inferiore e che U è triangolare superiore per risolvere questi due sistemi è necessario applicare semplicemente l'algoritmo di sostituzione in avanti seguito dall'algoritmo di sostituzione all'indietro. Questi due algoritmi hanno entrambi costo computazionale di circa $\frac{1}{2}n^2$ e quindi complessivamente abbiamo

$$c.c._{tot} \approx n^2.$$[6]

[6]Dal punto di vista del sistema lineare conoscere una sua fattorizzazione LU è equivalente a conoscere la matrice inversa di A, siccome anche $A^{-1}\mathbf{b}$ ha un costo computazionale di circa n^2.

Sembrerebbe questo un risultato miracoloso, tuttavia ricordiamo che se la fattorizzazione non è nota, calcolarla con l'algoritmo di eliminazione di Gauss ha un costo computazionale di circa $\frac{1}{3}n^3$. Inoltre non abbiamo la certezza,per matrici generiche senza strategie pivotali, di poter utilizzare l'algoritmo ed utilizzandole sicuramente non possiamo ottenere una fattorizzazione.

Collegamento con le esercitazioni

- L'Esercizio 07.1 implementa e applica la fattorizzazione LU, e ne discute il costo computazionale in caso di risoluzione di uno o più sistemi lineari. Lo stesso esercizio, e anche l'Homework 07.1, introducono possibili varianti con pivoting.
- L'Homework 07.2 estende la definizione di fattorizzazione LU a matrici a blocchi, introducendo il concetto di complemento di Schur.

Fattorizzazione di Choleski

Se A è una matrice simmetrica definita positiva sono sicuro che esisterà almeno una fattorizzazione LU di A perché posso sicuramente applicare l'eliminazione di Gauss anche senza strategie pivotali. Chiamiamo L la matrice triangolare inferiore prodotta dall'eliminazione di Gauss e U la corrispondente triangolare superiore.

Conviene poi costruire anche una matrice diagonale D che ha sulla diagonale principale gli elementi della diagonale principale di U. Detta $\tilde{U} = D^{-1}U$, risulta

$$\tilde{U} = \begin{pmatrix} 1 & * & \dots & * \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & * \\ 0 & \dots & 0 & 1 \end{pmatrix}$$

essendo poi A simmetrica,

$$A = A^T = (LD\tilde{U})^T = \tilde{U}^T D^T L^T = \tilde{U}^T D L^T$$

otteniamo quindi una fattorizzazione LU di A, essendo $\tilde{U}^T$ triangolare inferiore e (DL^T) triangolare superiore.

Per l'unicità della fattorizzazione LU esisterà una matrice diagonale che lega LU e $\tilde{U}^T \cdot (DL^T$ ma ricordando che L e $\tilde{U}^T$ hanno la stessa diagonale principale la matrice che lega le fattorizzazioni non può che essere la matrice identica. Allora

$$L = \tilde{U}^T$$

cioè

$$\tilde{U} = L^T.$$

Otteniamo così la **prima forma della Fattorizzazione di Choleski**

$$A = L \cdot D \cdot L^T.$$

Prima di giungere alla seconda forma premettiamo la seguente proposizione

(2.5.6) Proposizione *Essendo A definita positiva, $\forall\, i, d_{i,i} > 0$.*

Dimostrazione. Essendo $A = L \cdot D \cdot L^T$, sapendo che $\mathbf{x}^T L \cdot D \cdot L^T \mathbf{x} > 0$ per ogni vettore non nullo, posto

$$\mathbf{x} = L^{-T} e_i$$

con e_i l'i-esimo vettore della base canonica, risulta

$$e_i^T D e_i > 0.$$

Essendo $d_{i,i} = e_i^T D e_i$, otteniamo la tesi. ■

Consideriamo ora una matrice E diagonale così costruita

$$E = \begin{pmatrix} \pm\sqrt{d_{1,1}} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \pm\sqrt{d_{n,n}} \end{pmatrix}$$

allora $D = E^2 = E \cdot E^T$.

Possiamo quindi scrivere

$$A = L \cdot D \cdot L^T = L \cdot E \cdot E^T \cdot L^T = (LE) \cdot (E^T L^T) = (LE) \cdot (LE)^T.$$

Detta $\tilde{L} := LE$, matrice triangolare inferiore, otteniamo la **seconda forma della Fattorizzazione di Choleski**

$$A = \tilde{L} \cdot \tilde{L}^T.$$

(2.5.7) Osservazione *Possiamo applicare la fattorizzazione di Choleski alla risoluzione di un sistema lineare. Se $A = LDL^T$, allora abbiamo*

$$LDL^T \mathbf{x} = \mathbf{b}$$

da cui detto $\mathbf{z} = L^T \mathbf{x}$ e $\mathbf{y} = D\mathbf{z}$ posso costruire tre sistemi lineari

$$\begin{cases} L\mathbf{y} = \mathbf{b} \\ D\mathbf{z} = \mathbf{y} \\ L^T \mathbf{x} = \mathbf{z} \end{cases}$$

che se risolti in questo ordine mi danno la soluzione del sistema lineare di partenza $A\mathbf{x} = \mathbf{b}$. Notiamo subito che il secondo sistema lineare è un sistema addirittura diagonale. Il costo computazionale per la risoluzione dei tre sistema è di circa n^2.

Siccome esiste un'altra strada oltre all'applicazione dell'eliminazione di Gauss per ottenere la fattorizzazione di Choleski, detta **metodo compatto**, di costo computazionale circa $\frac{1}{6}n^3$, diventa molto conveniente[7] risolvere i sistemi lineari le cui matrici sono simmetriche definite positive in questo modo.

Collegamento con le esercitazioni

- L'Esercizio 07.2 implementa e applica entrambe le forme della fattorizzazione di Choleski.
- L'Homework 07.3 propone alcune matrici e chiede di determinare per quali di esse siano applicabili la fattoriazzione LU, la fattorizzazione di Choleski, o entrambe.

[7] Addirittura si ha un guadagno anche in termini di occupazione di memoria.

2.6 Condizionamento di un sistema lineare

Sia dato il seguente problema matematico:

Risolvere $A\mathbf{x} = \mathbf{b}$, dati A e $\mathbf{b}$ e sapendo che $\det A \neq 0$.

Vogliamo determinare l'indice di condizionamento di questo problema. Cominciamo da un caso preliminare: supponiamo che l'errore sui dati sia circoscritto al solo vettore dei termini noti, cioè

$$\mathbf{b} \approx \bar{\mathbf{b}} = \mathbf{b} + \delta\mathbf{b}$$

avremo quindi $\mathbf{x} \approx \bar{\mathbf{x}} = \mathbf{x} + \delta\mathbf{x}$. Quello che risolveremo sarà $A\bar{\mathbf{x}} = \bar{\mathbf{b}}$ e quindi possiamo scrivere

$$A(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b} + \delta\mathbf{b}$$

da cui, con semplici manipolazioni algebriche, ricordando che $A\mathbf{x} = \mathbf{b}$

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

ma siccome A è non singolare esisterà la sua inversa e perciò

$$\delta\mathbf{x} = A^{-1}\delta\mathbf{b}.$$

Ora,

$$\varepsilon_{abs}(\bar{\mathbf{x}}) = ||\delta\mathbf{x}|| = ||A^{-1}\delta\mathbf{b}|| \leq ||A^{-1}|| \cdot ||\delta\mathbf{b}|| = ||A^{-1}||\varepsilon_{abs}(\bar{\mathbf{b}})$$

L'indice di condizionamento assoluto sarà quindi $K^{abs} = ||A^{-1}||$.

$$\varepsilon_{rel}(\bar{\mathbf{x}}) = \frac{\varepsilon_{abs}(\bar{\mathbf{x}})}{||\mathbf{x}||} \leq \frac{||A^{-1}||\varepsilon_{abs}(\bar{\mathbf{b}})}{||\mathbf{x}||}$$

considerando che $\mathbf{b} = A\mathbf{x}$, normando i due membri otteniamo

$$||\mathbf{b}|| = ||A\mathbf{x}|| \leq ||A|| \cdot ||\mathbf{x}||$$

e quindi $\frac{1}{||\mathbf{x}||} \leq \frac{||A||}{||\mathbf{b}||}$. Sostituendo nella precedente relazione otteniamo

$$\varepsilon_{rel}(\bar{\mathbf{x}}) \leq ||A|| \cdot ||A^{-1}||\varepsilon_{rel}(\bar{\mathbf{b}}).$$

L'indice di condizionamento relativo sarà quindi $K = ||A|| \cdot ||A^{-1}|| = K(A)$, indice di condizionamento della matrice A.

Consideriamo ora il caso generale in cui l'errore è sia sui termini noti che sui coefficienti delle incognite. Avremo

$$A \approx \bar{A} = A + \delta A\ ,\ \mathbf{b} \approx \bar{\mathbf{b}} = \mathbf{b} + \delta\mathbf{b}$$

ipotizziamo inoltre che $||\delta A|| < \frac{1}{||A^{-1}||}$.

Avremo, analogamente al caso preliminare,

$$(A + \delta A)(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b} + \delta\mathbf{b}$$

perciò,

$$(A+\delta A)\delta\mathbf{x} + A\mathbf{x} + \delta A\ \mathbf{x} = \mathbf{b} + \delta\mathbf{b}$$

da cui

$$(A+\delta A)\delta\mathbf{x} + \delta A\ \mathbf{x} = \delta\mathbf{b}$$

e quindi

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

Potendo scrivere $A+\delta A = A(Id + A^{-1}\delta A)$, e ponendo $B := -A^{-1}\delta A$, risulta $A+\delta A = A(I-B)$.

Calcoliamo la norma della matrice B,

$$||B|| = ||A^{-1}\delta A|| \leq ||A^{-1}|| \cdot ||\delta A|| < 1$$

perciò la matrice B è convergente e quindi dal lemma sulle matrici convergenti ricaviamo che $Id - B$ è non singolare. Allora

$$(A+\delta A)^{-1} = (Id-B)^{-1}A^{-1}$$

da cui

$$\delta\mathbf{x} = (Id-B)^{-1}A^{-1}(\delta\mathbf{b} - \delta A\ \mathbf{x})$$

normando ambo i membri otteniamo

$$||\delta\mathbf{x}|| \leq ||(Id-B)^{-1}|| \cdot ||A^{-1}|| \cdot (||\delta\mathbf{b}|| + ||\delta A||||\mathbf{x}||) \leq \frac{||A^{-1}||}{1 - ||A^{-1}|| \cdot ||\delta A||}(||\delta\mathbf{b}|| + ||\delta A||||\mathbf{x}||).$$

Possiamo quindi stimare l'errore relativo

$$\varepsilon_{rel}(\bar{\mathbf{x}}) \leq \frac{1}{||\mathbf{x}||} \cdot \frac{||A^{-1}||}{1 - ||A^{-1}|| \cdot ||\delta A||}(||\delta\mathbf{b}|| + ||\delta A|| \cdot ||\mathbf{x}||) = \frac{||A^{-1}||}{1 - ||A^{-1}|| \cdot ||\delta A||}\left(\frac{||\delta\mathbf{b}||}{||\mathbf{x}||} + \frac{||\delta A|| \cdot ||\mathbf{x}||}{||\mathbf{x}||}\right)$$

analogamente al caso precedente possiamo stimare $\frac{1}{||\mathbf{x}||} \leq \frac{||A||}{||\mathbf{b}||}$ e quindi

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

essendo $\frac{||\delta A||}{||A||} = \varepsilon_{rel}(A)$, possiamo scrivere

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

Siccome $\varepsilon_{rel}(\bar{\mathbf{b}}) + \varepsilon_{rel}(\bar{A}) = \varepsilon_{rel}(\bar{\mathbf{b}}, \bar{A})$, in definitiva abbiamo che l'indice di condizionamento del problema viene così stimato

$$K \leq \frac{||A||||A^{-1}||}{1 - ||A^{-1}|| \cdot ||\delta A||}$$

e per piccoli errori sulla matrice, è facile vedere che $K \approx K(A)$.

Scritto da Mattia Garatti

(2.6.1) Osservazione *Se richiedessimo che* $||\delta A|| \leq \frac{1}{2||A^{-1}||}$ *potremmo fare un'ulteriore stima*

$$||\delta A|| \cdot ||A^{-1}|| \leq \frac{1}{2}$$

allora

$$\frac{||A||||A^{-1}||}{1 - ||A^{-1}|| \cdot ||\delta A||} \leq 2K(A).$$

Collegamento con le esercitazioni L'Homework 05.3 richiede una applicazione dei risultati di questa sezione per uno specifico sistema lineare.

2.7 Metodi iterativi classici

Gli algoritmi di sostituzione in avanti e all'indietro sono metodo diretti per arrivare alla soluzione di un sistema lineare. Essi non sono l'unico modo per produrre una soluzione.

Esiste una categoria di metodi, detti **metodi iterativi**, in cui la soluzione si ottiene mediante un processo infinito di approssimazioni: in altre parole la soluzione esatta $\mathbf{x}$ sarà

$$\mathbf{x} = \lim_{k \to \infty} \mathbf{x}^{(k)}$$

dove $\mathbf{x}^{(k)} \approx \mathbf{x}$ è un'approssimazione della soluzione esatta.

Non è tuttavia detto a priori che ci sia garanzia che l'approssimazione al passo $k+1$ sia migliore di quella al passo k. Parliamo quindi di **metodo convergente** se il limite della successione delle approssimazioni esiste finito.

Risulta chiaro che abbiamo un problema a livello procedurale: il calcolatore non può effettuare un numero infinito di approssimazioni successive. Entra dunque in gioco il concetto di **test di arresto**: in base alla verifica o meno di questo il calcolatore saprà di fermarsi quando è abbastanza vicino alla soluzione esatta, in simboli

$$\text{STOP se } \varepsilon(\mathbf{x}^{(k)}) \leq \sigma$$

con σ la soglia di arresto. Ovviamente il test di arresto incide sul numero di iterazioni effettuate e quindi in particolare sul costo computazionale. Per misurare il costo computazionale di un metodo iterativo in generale possiamo procedere in questo modo, moltiplicando il costo del singolo passo per il numero γ di iterazioni

$$c.c. = c.c.(k) * \gamma$$

Nell'ambito della risoluzione di sistemi lineari esistono più famiglie di metodi iterativi. Analizziamo in questo corso i metodi iterativi classici. Questi metodi si basano su una decomposizione della matrice A in una differenza di due matrici N e M, cioè

$$A = N - M$$

in cui N deve essere una matrice *facilmente invertibile*[8].

Il sistema lineare $A\mathbf{x} = \mathbf{b}$ diventerà quindi

[8] Significa che è facile risolvere un sistema lineare che ha come matrice associata N.

$$N\mathbf{x} - M\mathbf{x} = \mathbf{b}$$

quindi,

$$N\mathbf{x} = \mathbf{b} + M\mathbf{x}$$

ed essendo N invertibile,

$$\mathbf{x} = N^{-1}(M\mathbf{x} + \mathbf{b})$$

ottenendo

$$\mathbf{x} = N^{-1}M\mathbf{x} + N^{-1}\mathbf{b}$$

che è un **problema di punto fisso**[9]. Se quindi $\mathbf{x} = \Phi(\mathbf{x})$, con Φ una funzione che manda vettori in vettori, stiamo cercando un vettore che venga lasciato invariato da Φ.

Siano ora $B := N^{-1}M$, la **matrice di iterazione**, e $\mathbf{t} := N^{-1}\mathbf{b}$, otteniamo

$$\mathbf{x} = B\mathbf{x} + \mathbf{t}$$

Per risolvere un problema di punto fisso il suggerimento è di iterare la funzione Φ: scegliamo un vettore di innesco $\mathbf{x}^{(0)} \in \mathbb{R}^n$. Per $k = 0, 1, ...$ costruiamo

$$\mathbf{x}^{(k+1)} = \Phi(\mathbf{x}^{(k)}) = B\mathbf{x}^{(k)} + \mathbf{t}$$

ottenendo procedendo a ritroso

$$N\mathbf{x}^{(k+1)} = M\mathbf{x}^{(k)} + \mathbf{b}.$$

(2.7.1) Definizione *Un metodo iterativo si dice **consistente** se è convergente e il limite della successione delle approssimazioni tende alla soluzione esatta.*

(2.7.2) Teorema *Un metodo iterativo converge se e solo se la matrice di iterazione è convergente.*

Dimostrazione. L'errore al passo k è $\varepsilon^{(k)} = \mathbf{x} - \mathbf{x}^{(k)}$. Allora un metodo converge se e solo se $\varepsilon^{(k)}$ tende a 0.

Essendo $\mathbf{x} = B\mathbf{x} + \mathbf{t}$ e $\mathbf{x}^{(k+1)} = B\mathbf{x}^{(k)} + \mathbf{t}$, facendo la differenza membro a membro otteniamo

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

Ricordando che $\varepsilon^{(k+1)} = \mathbf{x} - \mathbf{x}^{(k+1)}$ e $\varepsilon^{(k)} = \mathbf{x} - \mathbf{x}^{(k)}$ risulta

$$\varepsilon^{(k+1)} = B\varepsilon^{(k)} = B \cdot B\varepsilon^{(k-1)}$$

e cioè

$$\varepsilon^{(k)} = B^k\varepsilon^{(0)}$$

Abbiamo quindi ottenuto che $\varepsilon^{(k)}$ tende a 0 se e solo se B^k tende a 0, dunque la tesi. ∎

[9] In generale un problema di punto fisso è un problema del tipo $f(x)$ con soluzione x.

Vediamo tre esempi di metodi iterativi classici:

- il metodo di **Jacobi**, che si presta ad essere utilizzato molto efficacemente in architetture parallele;
- il metodo di **Gauss-Seidel**, più efficente del metodo di Jacobi ma non adatto ad architetture parallele;
- la famiglia di metodi **SOR** (Successive Over-Relaxation).

Tutti e tre si basano su una decomposizione della matrice A in una somma di tre matrici particolari: una matrice triangolare inferiore L, contenente gli elementi di A che stanno sotto la diagonale principale, una matrice triangolare superiore U, contenente gli elementi di A che stanno sopra la diagonale principale ed una matrice diagonale D contenente gli elementi della diagonale principale di A; perciò

$$A = L + D + U.$$

Per quanto riguarda il metodo di Jacobi si pone

$$N = D \text{ e } M = -(L + U)$$

e quindi possiamo schematizzare l'algoritmo in questo modo

$$x_i^{(k+1)} = \frac{b_i - \sum_{j<i} a_{i,j} x_j^{(k)} - \sum_{j>i} a_{i,j} x_j^{(k)}}{a_{i,i}}$$

ottenendo come costo computazionale della singola iterazione $c.c.(k) \approx n^2$.

Per quanto riguarda il metodo di Gauss-Seidel si pone invece

$$N = (L + D) \text{ e } M = -U$$

e quindi possiamo schematizzare l'algoritmo in questo modo

$$x_i^{(k+1)} = \frac{b_i - \sum_{j<i} a_{i,j} x_j^{(k+1)} - \sum_{j>i} a_{i,j} x_j^{(k)}}{a_{i,i}}$$[10]

ottenendo come costo computazionale della singola iterazione $c.c.(k) \approx n^2$.

(2.7.3) Osservazione *Se A è una matrice sparsa $c.c.(k) << n^2$ perché vengono risparmiate le operazioni in cui l'elemento della matrice è 0.*

(2.7.4) Teorema *Se A è una matrice fortemente diagonalizzata allora il metodo di Jacobi converge.*

Dimostrazione. Se la matrice di iterazione è $B_J = -D^{-1}(L+U)$ allora risulta $||B_J||_\infty < 1$, da cui la tesi. ■

(2.7.5) Teorema *Se A è una matrice fortemente diagonalizzata allora il metodo di Gauss-Seidel converge.*

[10] Potrebbe essere strano vedere $x_j^{(k+1)}$ a secondo membro, tuttavia questo funziona perché sono componenti già calcolate in quanto $j < i$.

Dimostrazione. Siccome $||B_{G-S}||_\infty \leq ||B_J||_\infty$, otteniamo la tesi. ■

(2.7.6) Lemma (di Kahan) *Date due matrici $M, N \in Mat_n$, se $N - M$ è simmetrica definita positiva e $N + M$ è definita positiva allora N è non singolare e $\rho(N^{-1}M) < 1$.*

Dimostrazione. Omettiamo la dimostrazione. ■

(2.7.7) Teorema *Se A è una matrice simmetrica definita positiva allora il metodo di Gauss-Seidel converge.*

Dimostrazione. Sia $A = N - M$ con $N = L + D$ e $M = -U$. Essendo A simmetrica poi risulta $U = L^T$ e quindi abbiamo

$$N + M = L + D - L^T.$$

Risulta che $\mathbf{x}^T D\mathbf{x} > 0$ per la proposizione 3.38. Invece essendo $L - L^T$ antisimmetrica, $\mathbf{x}^T(L - L^T)\mathbf{x} = 0$[11]. Di conseguenza

$$\mathbf{x}^T(L + D - L^T)\mathbf{x} = \mathbf{x}^T D\mathbf{x} + \mathbf{x}^T(L - L^T)\mathbf{x} > 0.$$

La tesi discende allora dal lemma di Kahan. ■

Metodi SOR La famiglia di metodi SOR cerca di migliorare ulteriormente l'algoritmo di Gauss-Seidel

$$(L + D)\mathbf{x}^{(k+1)} = -U\mathbf{x}^{(k)} + \mathbf{b}$$

che possiamo riscrivere così

$$D\mathbf{x}^{(k+1)} = \mathbf{b} - U\mathbf{x}^{(k)} - L\mathbf{x}^{(k+1)}.$$

Sia $\omega \in \mathbb{R}$, introduciamo un vettore intermedio $\mathbf{z}$

$$\mathbf{z} = D^{-1}(\mathbf{b} - L\mathbf{x}^{(k+1)} - U\mathbf{x}^{(k)})$$

e poi calcoliamo $\mathbf{x}^{(k+1)}$ facendo una media pesata tra $\mathbf{x}^{(k)}$ e $\mathbf{z}$. Si ottiene quindi

$$\begin{cases} z_i = \dfrac{b_i - \sum_{j<i} a_{i,j}x_j^{(k+1)} - \sum_{j>i} a_{i,j}x_j^{(k)}}{a_{i,i}} \\ x_i^{(k+1)} = \omega z_i + (1 - \omega)x_i^{(k)} \end{cases}$$

Al variare di ω mi muovo sulla retta passante per $\mathbf{x}^{(k)}$ e $\mathbf{z}$: in particolare se $\omega > 1$ si verifica un'estrapolazione che accelera quindi il metodo di Gauss-Seidel, mentre se $\omega < 1$ si verifica un'interpolazione che invece lo rallenta. Il caso $\omega = 0$ non fornisce un metodo convergente perché l'algoritmo non modifica nemmeno il vettore d'innesco; il caso $\omega = 1$ coincide con il metodo di Gauss-Seidel.

[11] Considerando $\alpha = \mathbf{x}^T(L - L^T)\mathbf{x}$ come una matrice di Mat_1, possiamo facilmente dire che α è simmetrica ma allora $\alpha = \alpha^T$, e quindi $\mathbf{x}^T(L - L^T)\mathbf{x} = (\mathbf{x}^T(L - L^T)\mathbf{x})^T = -\mathbf{x}^T(L - L^T)\mathbf{x} = -\alpha$. Di conseguenza $\alpha = 0$.

Per come è definita, la famiglia di metodi SOR, potrebbe sembrare non essere un metodo classico, invece lo è infatti è possibile scriverlo con l'usuale struttura con decomposizione della matrice A. Introduciamo innanzitutto $\mu = \frac{1}{\omega}$, supponendo quindi $\omega \neq 0$. Moltiplichiamo a sinistra la seconda equazione della definizione per μD, ottenendo

$$\mu D\mathbf{x}^{(k+1)} = D\mathbf{z} + (\mu - 1)D\mathbf{x}^{(k)}$$

ora per come è stato costruito $\mathbf{z}$

$$\mu D\mathbf{x}^{(k+1)} = \mathbf{b} - L\mathbf{x}^{(k+1)} - U\mathbf{x}^{(k)} + (\mu - 1)D\mathbf{x}^{(k)}$$

$$L\mathbf{x}^{(k+1)} + \mu D\mathbf{x}^{(k+1)} = \mathbf{b} - ((1 - \mu)D + U)\mathbf{x}^{(k)}$$

$$(L + \mu D)\mathbf{x}^{(k+1)} = \mathbf{b} - ((1 - \mu)D + U)\mathbf{x}^{(k)}$$

quindi ponendo $N_\omega = L + \mu D$ e $M_\omega = -((1-\mu)D + U)$ otteniamo la classica decomposizione della matrice A essendo

$$N_\omega - M_\omega = L + \mu D + (1 - \mu)D + U = L + D + U = A$$

(2.7.8) Teorema *Se $\omega \notin (0, 2)$ allora il metodo SOR corrispondente non converge.*

Dimostrazione. Calcoliamo il valore assoluto del determinante della matrice di iterazione

$$|\det B_\omega| = \frac{|\det M_\omega|}{|\det N_\omega|} = \frac{|1 - \mu|^n \cdot |\det D|}{|\mu|^n |\det D|} = \left|\frac{1}{\mu} - 1\right|^n = |\omega - 1|^n \geq 1.$$

Siccome il modulo del determinante è maggiore o uguale a 1 allora deve esistere[12] un autovalore che ha modulo maggiore o uguale ad 1 e quindi il raggio spettrale sarà maggiore o uguale ad 1 e quindi la matrice d'iterazione non sarà convergente, da cui la tesi. ■

(2.7.9) Teorema *Se A è una matrice simmetrica definita positiva e $\omega \in (0, 2)$ allora il metodo SOR corrispondente converge.*

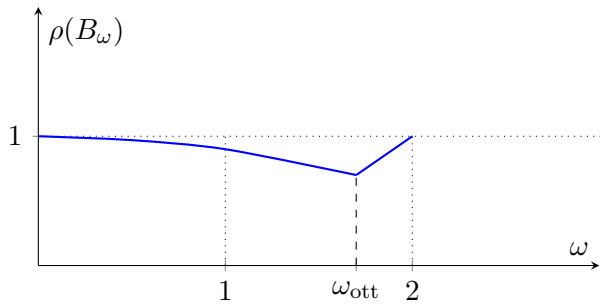
Dimostrazione. Innanzitutto essendo $A = N_\omega - M_\omega$, per ipotesi abbiamo che $N_\omega - M_\omega$ è simmetrica definita positiva. Inoltre

$$N_\omega + M_\omega = L + \mu D - (U + (1 - \mu)D) = (2\mu - 1)D + L - L^T.$$

Ora, ricordando che per ogni vettore non nullo $\mathbf{x}$ risulta $\mathbf{x}^T(L - L^T)\mathbf{x} = 0$ e osservando che $(2\mu - 1)D$ è simmetrica e che $\forall\, \omega \in (0, 2)$ risulta $\frac{2}{\omega} - 1 > 0$ possiamo dire che $N_\omega + M_\omega$ sia definita positiva. La tesi discende dal lemma di Kahan. ■

Di seguito è mostrato un grafico che mostra l'andamento del raggio spettrale della matrice d'iterazione al variare di ω, il valore ω_{ott} indica il valore ottimale di ω per ottenere il metodo SOR più efficiente.

[12] Discende dal fatto che il determinante è il prodotto di tutti gli autovalori.



Purtroppo in generale non c'è una formula per calcolare ω_{ott}: si procede per tentativi utilizzando dei problemi modello.

Collegamento con le esercitazioni

- Gli Esercizi 08.1 e 08.2 verificano le condizioni di convergenza dei metodi di Jacobi, Gauss-Seidel e SOR. In particolare, l'Esercizio 08.1 utilizza la condizione necessaria e sufficiente, mentre nell'Esercizio 08.2 basta verificare condizioni sufficienti.
- L'Esercizio 08.2 e l'Homework 08.1 discutono una implementazione dei metodi di Jacobi e Gauss-Seidel, scegliendo uno specifico criterio d'arresto.
- L'Homework 08.2 propone ulteriori condizioni sufficienti per la convergenza di un metodo iterativo.
- L'Homework 08.3 discute un caso in cui il metodo di Jacobi calcola in un numero finito di iterazioni la soluzione esatta di un sistema lineare molto particolare, cioè si comporta come un metodo diretto.
- L'Homework 08.4 propone un metodo iterativo che non corrisponde né a Jacobi, né a Gauss-Sediel, né a SOR.

Capitolo 3

Equazioni non lineari

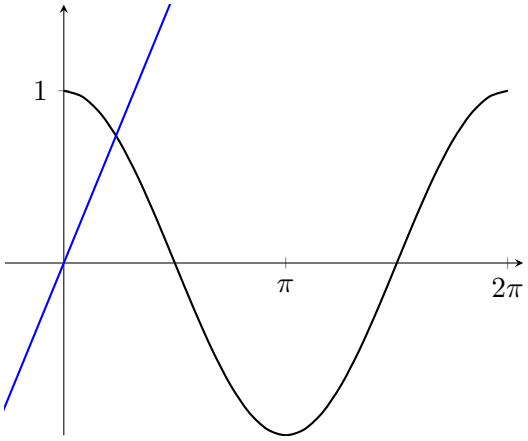
In questo capitolo vogliamo analizzare alcuni metodi per la risoluzione di equazioni non lineari. Premettiamo subito che non esistono formule risolutive in generale per affrontare questo tipo di equazioni: ricorreremo pertanto a metodi iterativi.

3.1 Alcuni richiami di analisi matematica

(3.1.1) Esempio *Consideriamo l'equazione*

$$\cos x = x$$

Visualizziamo graficamente questa equazione



Risolvere questa equazione corrisponde a trovare gli zeri della funzione continua

$$f : \begin{cases} [a,b] \to \mathbb{R} \\ x \mapsto \cos x - x \end{cases}$$

Osserviamo che a priori, quando trattiamo un'equazione non lineare, non abbiamo garanzie sull'esistenza della soluzione e nemmeno sulla sua unicità.

(3.1.2) Teorema (di esistenza degli zeri) *Siano $a, b \in \mathbb{R}$ con $a < b$ e sia $f : [a,b] \to \mathbb{R}$ una funzione continua. Supponiamo $f(a) \cdot f(b) \leq 0$. Allora esiste $c \in]a,b[$ tale che $f(c) = 0$.*

Dimostrazione. Omettiamo la dimostrazione in quanto già presentata in corsi precedenti. ■

Questo teorema garantisce, sotto l'ipotesi che la funzione cambi segno agli estremi, che esista almeno una soluzione ad un'equazione non lineare descrivibile come funzione continua. Tuttavia la soluzione potrebbe ancora non essere unica.

Purtroppo nessun metodo ci permette di trovare tutte le soluzioni di un'equazione non lineare.

3.2 Metodo di bisezione

Il metodo di bisezione è il più semplice ma efficace metodo per la risoluzione di equazioni non lineari. Consideriamo un'equazione descrivibile tramite una funzione che soddisfa le ipotesi del teorema 3.1.2.

Prendiamo innanzitutto il centro dell'intervallo $[a, b]$:

$$c = \frac{a+b}{2}$$

Ora se $f(a) \cdot f(c) \leq 0$, definiamo un nuovo intervallo $[a_1, b_1] := [a, c]$; altrimenti, se ciò non accade, dovrà essere sicuramente[1] $f(c) \cdot f(b) \leq 0$ e quindi definiremo $[a_1, b_1] := [c, b]$. L'idea intuitivamente è di iterare questo processo. Volendo scrivere un algoritmo, a partire da un intervallo iniziale $[a_0, b_0] := [a, b]$, otteniamo

per $k = 0, 1, \ldots$:
    $c_k = \frac{a_k + b_k}{2}$;
    se $f(a_k) \cdot f(c_k) \leq 0$:
        $[a_{k+1}, b_{k+1}] = [a_k, c_k]$;
    altrimenti:
        $[a_{k+1}, b_{k+1}] = [c_k, b_k]$;

Ovviamente, siccome questo algoritmo presuppone un processo infinito, ci sarà di mezzo un test di arresto: interromperemo le iterazioni quando troveremo un intorno abbastanza piccolo che contiene la soluzione.

Notiamo subito che il metodo di bisezione non produce un'approssimazione della soluzione. Esso è infatti un metodo di tipo **enclosure**, cioè produce una successione di intervalli di ampiezza sempre minore, uno contenuto nell'altro, in cui si trova la soluzione esatta α:

- $[a_{k+1}, b_{k+1}] \subseteq [a_k, b_k]$;
- $\alpha \in [a_k, b_k] \ \forall\, k$;
- l'ampiezza dell'intervallo all'aumentare delle iterazioni tende a 0, cioè

$$\lim_{k \to \infty} b_k - a_k = 0.$$

[1] Perché per ipotesi abbiamo che $f(a) \cdot f(b) \leq 0$ quindi l'intersezione con l'asse delle ascisse se non è prima del centro dell'intervallo sarà sicuramente dopo.

Come approssimazione della soluzione esatta al passo k la scelta più naturale è quella del centro dell'intervallo $[a_k, b_k]$

$$\alpha \approx c_k = \frac{a_k + b_k}{2}.$$

Una stima dell'errore assoluto sarà data dalla semi-ampiezza dell'intervallo

$$\varepsilon_{abs}(c_k) \leq \frac{b_k - a_k}{2}.$$

(3.2.1) Osservazione *Possiamo stimare l'errore assoluto al passo k fin dall'inizio, perché l'ampiezza dell'intervallo si dimezza ad ogni iterazione*

$$\varepsilon_{abs}^{(k)} \leq \frac{1}{2^{k+1}}(b-a).$$

Un semplice ma efficace metodo per capire dopo quante iterazioni terminare l'algoritmo è il seguente: impongo che l'errore assoluto scenda sotto una certa soglia $\sigma > 0$, cioè

$$\frac{1}{2^{k+1}}(b-a) \leq \sigma$$

dopo semplici calcoli arriviamo a

$$k \geq \log_2 \frac{b-a}{2\sigma}.$$

Quindi possiamo affermare che dopo $\log_2 \frac{b-a}{2\sigma}$ passi, otterremo un'approssimazione sufficientemente corretta, rispetto alla soglia che noi abbiamo imposto, della soluzione esatta.

Collegamento con le esercitazioni L'Esercizio 09.1 contiene una applicazione del metodo di bisezione.

3.3 Velocità di convergenza di un metodo iterativo

In questa sezione introduciamo alcuni concetti fondamentali relativi alla velocità di convergenza di un metodo iterativo in generale.

(3.3.1) Definizione *Diciamo che una successione (x_k) di approssimazioni, di una soluzione esatta α, prodotta da un processo iterativo* ***converge linearmente****, oppure converge con ordine $p = 1$[2], se esiste $c \in (0,1)$ tale che per ogni k*

$$\varepsilon_{k+1} \leq c \cdot \varepsilon_k$$

con $\varepsilon_k = |\alpha - x_k|$ l'errore assoluto.

(3.3.2) Osservazione *Osserviamo che sotto le ipotesi della definizione* (3.3.1) *il metodo iterativo converge perché l'errore assoluto tende a* 0.

[2]Il numero p indica la velocità di convergenza, cioè quanto velocemente converge un metodo iterativo.

(3.3.3) Definizione *Sia $p > 1$. Diciamo che una successione (x_k) di approssimazioni, di una soluzione esatta α, prodotta da un processo iterativo* ***converge con ordine*** *p, se esiste $c > 0$, tale che per ogni k*

$$\varepsilon_{k+1} \leq c \cdot \varepsilon_k^p.$$

(3.3.4) Proposizione *Abbiamo garanzia di convergenza per un metodo iterativo che soddisfa le ipotesi della definizione* (3.3.3) *se l'errore iniziale ε_0 è sufficientemente piccolo.*

Dimostrazione. Consideriamo solamente il caso $p = 2$, che viene detto **convergenza quadratica**[3]. Introduciamo il residuo $r_k = c \cdot \varepsilon_k$. Risulta

$$r_{k+1} = c\varepsilon_{k+1} \leq c \cdot c\varepsilon_k^2 = r_k^2$$

quindi

$$r_k \leq r_{k-1}^2 \leq r_{k-2}^{2^2} \leq \cdots \leq r_0^{2^k}$$

perciò se $r_0 < 1$ allora risulta $\lim_{k\to\infty} r_k = 0$. Da qui possiamo quindi dire che se l'errore iniziale è più piccolo di $1/c$ abbiamo la convergenza del metodo, cioè se $\varepsilon_0 < \frac{1}{c}$ allora $\lim_{k\to\infty} \varepsilon_k = 0$. ■

Introdotti questi concetti possiamo analizzare il metodo di Bisezione descritto nella sezione precedente.

(3.3.5) Proposizione *Il metodo di Bisezione converge linearmente rispetto alla semi-ampiezza dell'intervallo $\varepsilon_{abs}^{(k)}$.*

Dimostrazione. Se $r_k = \frac{b-a}{2^{k+1}}$ è una stima dell'errore, ho che $\varepsilon_k \leq r_k$. Quindi

$$r_{k+1} \leq \frac{1}{2} r_k$$

cioè se al posto dell'errore utilizzo una stima dell'errore il metodo di bisezione soddisfa le ipotesi della definizione (3.3.1). ■

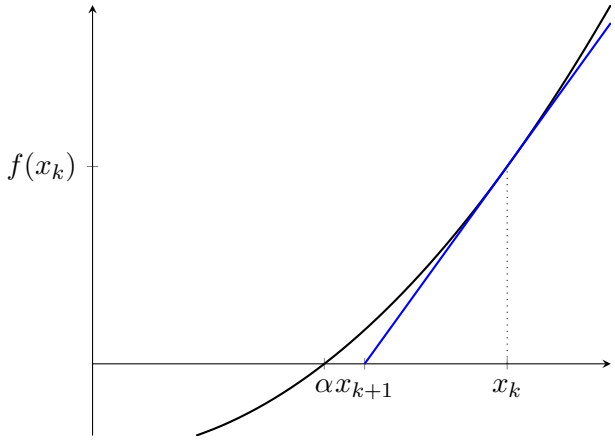
Collegamento con le esercitazioni L'Esercizio 09.1 mostra che il metodo di bisezione converge in modo monotono (e lineare) rispetto alla semi-ampiezza del k-esimo intervallo, ma che la convergenza del residuo $f(c_k)$ in generale non è affatto monotona.

3.4 Metodo di Newton - Raphson

Il metodo di Newton-Raphson, detto anche **metodo delle tangenti**, è un metodo iterativo molto usato perché di semplice implementazione; richiede tuttavia di essere in grado di scrivere l'espressione analitica della derivata della funzione di cui cerchiamo gli zeri.

L'idea alla base è la seguente: costruiamo la retta tangente alla funzione nel punto x_k e prendiamo x_{k+1} come l'intersezione di questa retta con l'asse delle ascisse. Visualizziamo graficamente

[3] Cioè il numero di cifre corrette raddoppia ad ogni passo.



Per fare ciò serve che la funzione sia almeno di classe C^1 con derivata non nulla. Sviluppando con Taylor la funzione posso ottenere la retta tangente, infatti

$$f(x) \approx s(x) = f(x_k) + (x - x_k)f'(x_k)$$

per come abbiamo scelto di prendere x_{k+1} avremo che

$$s(x_{k+1}) = 0$$

e quindi con semplici manipolazioni algebriche

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

Volendo scrivere un algoritmo, dato un valore di innesco x_0, otteniamo

per $k = 0, 1, \ldots$:
$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$;

(3.4.1) Teorema *Il metodo di Newton converge quadraticamente se la funzione f di cui cerco una radice è di classe C^2, la radice α cercata è semplice, e il valore di innesco x_0 è sufficientemente vicino ad α.*

Dimostrazione. Sviluppando mediante la formula di Taylor con il resto di Lagrange otteniamo

$$f(\alpha) = f(x_k) + (\alpha - x_k)f'(x_k) + \frac{1}{2}(\alpha - x_k)^2 f''(\xi_k)$$

con $\xi_k \in (\alpha, x_k)$ oppure $\xi_k \in (x_k, \alpha)$. Siccome $f(\alpha) = 0$, dividendo per $f'(x_k)$ otteniamo

$$-\frac{f(x_k)}{f'(x_k)} = \alpha - x_k + \frac{1}{2}(\alpha - x_k)^2 \frac{f''(\xi_k)}{f'(x_k)}$$

$$x_k - \frac{f(x_k)}{f'(x_k)} = \alpha + \frac{1}{2}(\alpha - x_k)^2 \frac{f''(\xi_k)}{f'(x_k)}$$

essendo il primo membro proprio x_{k+1}

$$\alpha - x_{k+1} = -\frac{1}{2}(\alpha - x_k)^2 \frac{f''(\xi_k)}{f'(x_k)}$$

e quindi in definitiva

$$\alpha - x_{k+1} = -\frac{1}{2}\frac{f''(\xi_k)}{f'(x_k)}\varepsilon_k^2.$$

Posto ora $c = \frac{\max\limits_{x\in(a,b)} |f''(x)|}{2 \min\limits_{x\in(a,b)} |f'(x)|} > 0$, possiamo scrivere

$$\varepsilon_{k+1} \leq c \cdot \varepsilon_k^2$$

da cui la tesi. ■

(3.4.2) Osservazione *A priori non abbiamo garanzia di convergenza globale. Se questo metodo converge, siccome ha convergenza di ordine $p = 2$, allora converge più rapidamente del metodo di bisezione, che ha ordine $p = 1$.*

Anche nel caso del metodo di Newton servirà introdurre un test di arresto: una scelta ottimale è quella di guardare la differenza tra due iterate consecutive.

Collegamento con le esercitazioni

- L'Esercizio 09.2 mostra un esempio per il quale le ipotesi del Teorema (3.4.1) sono valide, e contiene una verifica numerica della convergenza quadratica del metodo di Newton in tale caso.
- L'Esercizio 09.2 considera anche un esempio per il quale l'ipotesi di radice semplice nel Teorema (3.4.1) non è verificata, e mostra numericamente che in tale caso l'ordine di convergenza deteriora a lineare.
- L'Homework 09.3 introduce un metodo, dovuto ad Halley, che gode di convergenza cubica.

3.5 Metodo delle secanti

In molti casi non è possibile esprime analiticamente f', perciò il metodo precedente non è applicabile. Possiamo però approssimare la derivata prima con un rapporto incrementale andando così ad ottenere il cosiddetto metodo delle secanti.

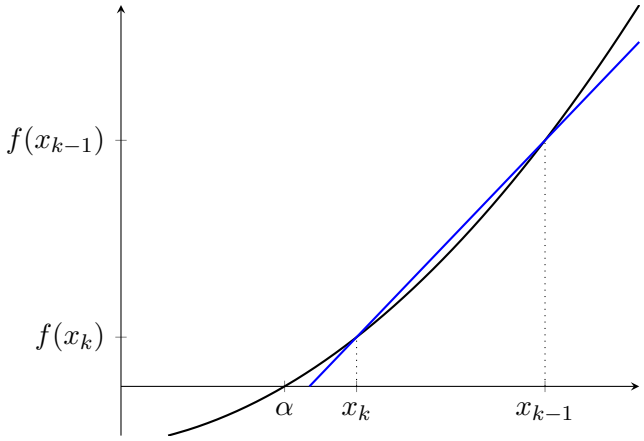
L'idea è la seguente: prendiamo la seguente approssimazione

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}.$$

Allora risulta, in modo analogo al metodo di Newton, che

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} \cdot f(x_k)$$

Visualizziamo graficamente



Vediamo subito dal grafico il motivo del nome: la retta che viene costruita è la secante alla funzione nei punti di ascissa x_k e x_{k-1}.

(3.5.1) Teorema *Il metodo delle secanti ha ordine di convergenza* $p = \frac{1+\sqrt{5}}{2}$.[4]

Dimostrazione. Omettiamo la dimostrazione. ■

Per il test di arresto si usa la stessa tecnica del metodo di Newton.

Collegamento con le esercitazioni L'Homework 09.4 richiede l'implementazione del metodo delle secanti e di altri metodi strettamente collegati, detti metodi Quasi-Newton.

3.6 Processo di iterazione funzionale

Premettiamo che in questa sezione non parliamo di un singolo metodo ma di una famiglia di metodi.

Sono la conseguenza naturale di un problema scritto sotto la forma di un *problema di punto fisso*, cioè

$$x = \varphi(x)$$

con $\varphi : [a, b] \to [a, b]$. Se φ è continua avremo la garanzia che esiste almeno un punto fisso, conformemente al teorema di esistenza degli zeri.

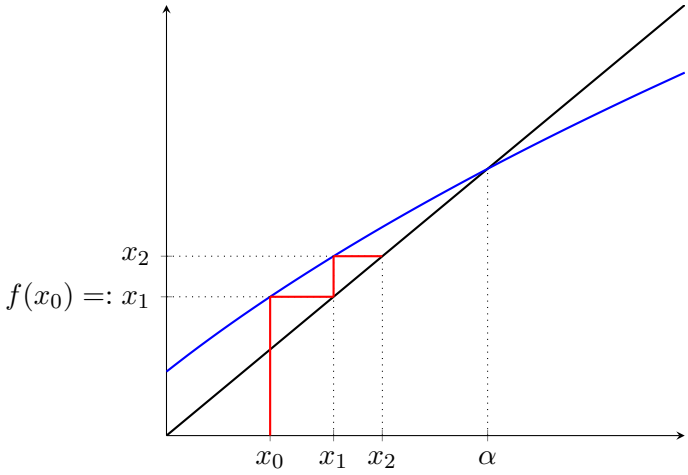
Un approccio molto naturale è utilizzare la funzione φ stessa per migliorare l'approssimazione di α, con α un punto fisso. Da ciò abbiamo quindi che l'algoritmo ricercato può essere scritto in questo modo, preso un valore d'innesco x_0

per $k = 0, 1, \ldots$:
$\quad x_{k+1} = \varphi(x_k)$;

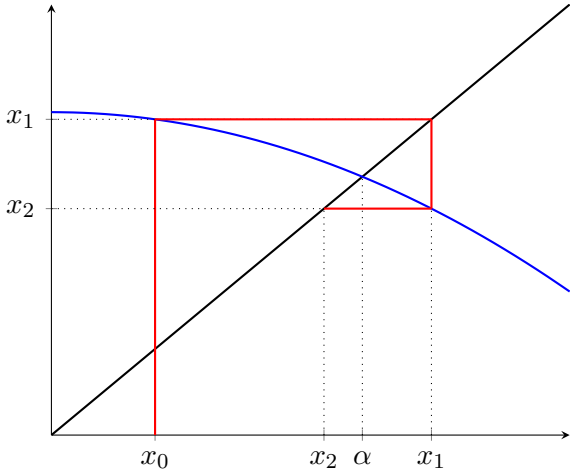
In generale non abbiamo garanzie di convergenza. Visualizziamo graficamente come varia la convergenza di questa famiglia di metodi al variare di $\varphi'(\alpha)$:

- se $0 < \varphi'(\alpha) < 1$, abbiamo una **convergenza monotona** (andamento a scalini)

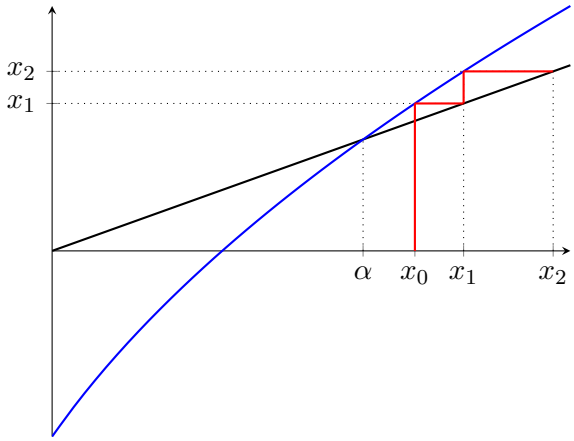
[4] La sezione aurea.



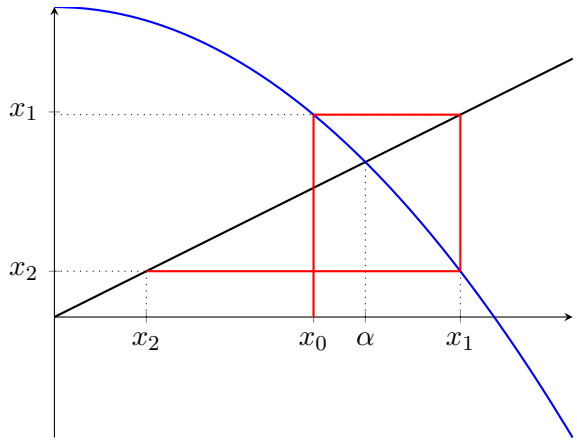
- se $-1 < \varphi'(\alpha) < 0$, abbiamo una **convergenza alternante** (andamento a spirale)



- se $\varphi'(\alpha) > 1$, abbiamo una **divergenza monotona** (andamento a scalini)



- se $\varphi'(\alpha) < -1$, abbiamo una **divergenza alternante** (andamento a spirale)



Possiamo dire quindi che abbiamo garanzia di convergenza per $|\varphi'(\alpha)| < 1$ e se l'errore iniziale è sufficientemente piccolo. In particolare ho poi convergenza quadratica se $\varphi'(\alpha) = 0$ e lineare se $\varphi'(\alpha) \neq 0$.

(3.6.1) Proposizione *Se φ è una contrazione, ovvero se è lipschitziana e la costante di lipschitz minore di 1, allora il processo di iterazione converge alla soluzione esatta.*

Dimostrazione. Se α è la soluzione esatta del problema di punto fisso

$$\alpha = \varphi(\alpha)$$

allora

$$|\alpha - x_{k+1}| = |\varphi(\alpha) - \varphi(x_k)| \leq L|\alpha - x_k|$$

ed essendo $L < 1$, il problema di punto fisso converge alla soluzione esatta. ■

Collegamento con le esercitazioni

- L'Esercizio 09.3 e l'Homework 09.1 discutono alcuni metodi di iterazione funzionale.
- L'Homework 09.3 riconduce la valutazione della radice quadrata di un numero ad un metodo di iterazione funzionale che richieda solamente la valutazione di operazioni elementari.

3.7 Metodo delle successioni di Sturm per equazioni polinomiali

Il metodo delle successioni di Sturm permette di trovare tutti gli zeri reali di un polinomio appartenenti ad un intervallo $[a, b] \subseteq \mathbb{R}$. Se $P \in \mathbb{P}_n$, l'insieme dei polinomi di grado n, allora sappiamo che P ha esattamente n zeri, contati con la relativa molteplicità. Considereremo polinomi a zeri semplici.

(3.7.1) Definizione *Siano le funzioni $f_0, \ldots, f_m : [a, b] \to \mathbb{R}$. Diciamo che esse formano una **Successione di Sturm** se*

(a) *f_m ha segno costante in $[a,b]$,*

(b) *per ogni $i \in [1,m[$, se $f_i(\xi)=0$ per qualche $\xi \in [a,b]$ allora $f_{i-1}(\xi)\cdot f_{i+1}(\xi)<0$,*

(c) *se $f_0(\xi)=0$ allora $f_0'(\xi)\cdot f_1(\xi)<0$.*

Sia $f_0 \in \mathbb{P}_n$ un polinomio avente solo zeri semplici in $[a,b]$. Poniamo, conformemente alla proprietà (c) della definizione 3.7.1[5]

$$f_1 := -f_0'.$$

Effettuiamo ora la divisione euclidea tra f_0 ed f_1. Otteniamo

$$f_0 = f_1 \cdot q + r.$$

Ora se $r=0$ allora f_0, f_1 è la successione di Sturm desiderata ed abbiamo concluso; altrimenti definiamo $f_2 = -r$ e iteriamo fino a che $r=0$.

Sia $x \in [a,b]$ tale che $f_0(x) \neq 0$. Calcoliamo l'immagine di x attraverso ogni funzione della successione di Sturm e contiamo le variazioni di segno (escludiamo gli zeri). Poniamo $\theta(x)$ il numero di variazioni di segno. Vale il seguente

(3.7.2) Teorema (di Sturm) *Sia $f(x)$ un polinomio a coefficienti reali ed $a,b \in \mathbb{R}$ tali che $f(a), f(b) \neq 0$. Allora $\theta(b)-\theta(a)$ è il numero di zeri reali di $f(x)$ in $[a,b]$.*

Dimostrazione. Sia $x \in [a,b]$, $\theta(x)$ varia al variare di x tra a e b solo se avviene un cambio di segno in una delle funzioni della successione di Sturm.

Conformemente alla (b) della definizione (3.7.1), per $1 \leq i < m$ se per un qualche $\xi \in (a,b)$ vale $f_i(\xi)=0$ avremo che $f_{i-1}(\xi)f_{i+1}(\xi)<0$. Allora in un intorno di ξ deve accadere che

x	$f_{i-1}(x)$	$f_i(x)$	$f_{i+1}(x)$
$\xi+\varepsilon$	$+$	$\pm$	$-$
ξ	$+$	0	$-$
$\xi-\varepsilon$	$+$	$\pm$	$-$

oppure

x	$f_{i-1}(x)$	$f_i(x)$	$f_{i+1}(x)$
$\xi+\varepsilon$	$-$	$\pm$	$+$
ξ	$-$	0	$+$
$\xi-\varepsilon$	$-$	$\pm$	$+$

in ogni caso passando per ξ non cambia il numero di variazioni di segno.

Invece se ξ è una radice di f_0 avremo

x	$f_0(x)$	$f_1(x)$
$\xi+\varepsilon$	$+$	$+$
ξ	0	$+$
$\xi-\varepsilon$	$-$	$+$

oppure

x	$f_0(x)$	$f_1(x)$
$\xi+\varepsilon$	$-$	$-$
ξ	0	$-$
$\xi-\varepsilon$	$+$	$-$

in ogni caso passando per ξ il numero di variazioni di segno aumenta. La combinazione di queste due osservazioni fornisce la tesi. ■

[5] Infatti se $f_0(\xi)=0$ allora $f_1'(\xi) \neq 0$.

Dal teorema (3.7.2), risulta definita una funzione

$$\sigma_{f_0} : \begin{cases} [a,b] \times [a,b] \to \mathbb{N} \\ (c,d) \mapsto \sigma_{f_0}(c,d) := \theta(d) - \theta(c) \end{cases}^{6}.$$

Altro non è che la funzione che conta gli zeri reali di $f_0 : [a,b] \to \mathbb{R}$ contenuti nell'intervallo $[c,d] \subseteq [a,b]$. In particolare $\sigma_{f_0}(a,b)$ è il numero di zeri reali di f in $[a,b]$.

(3.7.3) Osservazione *Ottenuto il numero di zeri reali basta trovare $I_1, \ldots, I_n$ intervalli disgiunti in $[a,b]$ ciascuno contenente uno ed un solo zero di f_0 usando il metodo di bisezione.*

(3.7.4) Esempio *Verifichiamo con un esempio il funzionamento del metodo. Consideriamo*

$$f_0(x) = x^3 - 6x^2 + 11x - 6.$$

ed applichiamo il metodo delle successioni di Sturm. Poniamo

$$f_1(x) = -f_0'(x) = -3x^2 + 12x - 11.$$

Effettuiamo la divisione euclidea tra $f_0(x)$ e $f_1(x)$; troviamo

$$q(x) = -\frac{1}{3}x + \frac{2}{3} \qquad r(x) = -\frac{2}{3}x + \frac{4}{3}$$

perciò poniamo

$$f_2(x) = -r(x) = \frac{2}{3}x - \frac{4}{3}.$$

Analogamente otteniamo

$$f_3(x) = -1.$$

Osserviamo che non serve effettuare la divisione successiva in quanto il prossimo resto sarà sicuramente 0 e quindi abbiamo trovato la successione di Sturm $f_0, \ldots, f_3$. Cerchiamo ad esempio il numero di zeri reali contenuti nell'intervallo $[0,5]$

x	$f_0(x)$	$f_1(x)$	$f_2(x)$	$f_3(x)$
5	24	-26	2	-1
	$+$	$-$	$+$	$-$
0	-6	-11	$-\frac{4}{3}$	-1
	$-$	$-$	$-$	$-$

Quindi risulta $\sigma_{f_0}(0,5) = 3-0 = 3$ e quindi in questo intervallo ho 3 zeri reali. Suddividiamo ora l'intervallo $[0,5]$ in tre intervalli disgiunti, ciascuno dei quali contenga una soluzione.

Il centro dell'intervallo $[0,5]$ è $C_{[0,5]} = \frac{5-0}{2} = \frac{5}{2}$; risulta che $\theta(\frac{5}{2}) = 2$ allora nell'intervallo $\left[0, \frac{5}{2}\right]$ ho due radici reali mentre in $\left[\frac{5}{2}, 5\right]$ ne ho una. Ripetendo il ragionamento per l'intervallo $\left[0, \frac{5}{2}\right]$ si ottiene in conclusione

[6] Ovviamente deve essere $c < d$.

$$I_1 = \left[0, \frac{5}{4}\right] \qquad I_2 = \left[\frac{5}{4}, \frac{5}{2}\right] \qquad I_3 = \left[\frac{5}{2}, 5\right].$$

Osserviamo che $f_0(x)$ *è scomponibile in fattori*

$$f_0(x) = (x-1)(x-2)(x-3)$$

ed ha quindi radici $x_1 = 1, x_2 = 2$ *e* $x_3 = 3$ *che sono contenute rispettivamente in* I_1, I_2 *e* I_3*. Il metodo risulta quindi verificato.*

Capitolo 4

Interpolazione polinomiale

4.1 Il polinomio interpolante

L'interpolazione polinomiale prevede l'approssimazione di una funzione tramite un polinomio. L'obiettivo è poter avere una semplice approssimazione di una funzione molto complessa: rientra quindi nell'ambito più generale dell'approssimazione di funzioni.

Considereremo in questo capitolo funzioni reali di variabile reale f, almeno continue, definite su un intervallo $[a, b]$.

(4.1.1) Definizione *Sia $n \in \mathbb{N}$ e siano $x_0, \ldots, x_n$[1] dei punti distinti nell'intervallo $[a, b]$. Data la funzione $f : [a, b] \to \mathbb{R}$, chiamiamo **polinomio interpolante** il polinomio $\tilde{f} \in \mathbb{P}_n$ tale che $\tilde{f}(x_i) = f(x_i)$ per $i = 0, \ldots, n$. Diciamo, in particolare, che $\tilde{f}$ interpola f nei nodi.*

Prima di enunciare un importante teorema sulle funzioni interpolanti, premettiamo questa proprietà.

(4.1.2) Proposizione *Sia $P \in \mathbb{P}_n$. Se P ha un numero di radici maggiore di n, allora P è il polinomio nullo.*

Dimostrazione. Omettiamo la dimostrazione. ■

(4.1.3) Teorema *Sia $f : [a, b] \to \mathbb{R}$, allora esiste uno ed un solo polinomio interpolante.*

Dimostrazione. Dimostriamo innanzitutto l'unicità. Siano $p_1, p_2 \in \mathbb{P}_n$ due polinomi interpolanti di f. Sia $q := p_1 - p_2 \in \mathbb{P}_n$. Risulta

$$q(x_i) = p_1(x_i) - p_2(x_i) = f(x_i) - f(x_i) = 0.$$

Allora il polinomio q ha come radici i nodi di interpolazione. Ma i nodi sono $n + 1$ allora q è il polinomio nullo e quindi

$$p_1 = p_2.$$

Per dimostrare l'esistenza costruiamo il polinomio di interpolazione nella forma di Lagrange.

[1] I cosiddetti **nodi di interpolazione**: devono essere distinti, ma non necessariamente ordinati o equi-spaziati. Notiamo che i nodi sono $n + 1$.

1. Per $i = 0, \dots, n$, prendiamo
$$l_i = (x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)$$
polinomio di grado n. Vediamo subito, dal fatto di aver supposto i nodi distinti, che nel nodo x_j,
$$l(x_j) = \begin{cases} 0 \ , \ i \neq j \\ \neq 0 \ , \ i = j \end{cases} \quad .$$

2. Costruiamo i polinomi di Lagrange
$$L_i := \frac{1}{l_i(x_i)} l_i,$$
osserviamo che sono polinomi di grado n che fanno 1 nei nodi e zero altrove[2].

3. Costruiamo $\tilde{f}$ tramite una combinazione lineare
$$\tilde{f} := \sum_{i=0}^{n} f(x_i) L_i(x).$$

È evidente che $\tilde{f} \in \mathbb{P}_n$. Verifichiamo che interpola f nei nodi:
$$\tilde{f}(x_i) = \sum_{j=0}^{n} f(x_j) L_j(x_i)$$
ma dal fatto che i polinomi di Lagrange sono non nulli solo per $i = j$, risulta
$$\tilde{f}(x_i) = f(x_i) L_i(x_i) = f(x_i)$$
da cui la tesi. ■

(4.1.4) Definizione *Sia $f : [a, b] \to \mathbb{R}$. Sia $\tilde{f} \in \mathbb{P}_n$ il polinomio interpolante negli $x_0, \dots, x_n$ nodi. Chiamiamo* ***differenza divisa*** *di f nei nodi $x_0, \dots, x_n$ il coefficiente del termine di grado n di $\tilde{f}$ e lo indichiamo con $f[x_0, \dots, x_n]$. Chiamiamo poi* ***ordine*** *di una differenza divisa il numeri di nodi diminuito di 1.*

(4.1.5) Osservazione *La precedente definizione produce le seguenti osservazioni:*

- $f[x_0] = f(x_0)$*;*
- $f[x_0, \dots, x_n]$ *non dipende dall'ordine dei nodi;*
- *per una buona definizione i nodi devono essere distinti.*

- L'Esercizio 11.2 discute l'implementazione dell'interpolazione polinomiale in forma di Lagrange.
- L'Esercizio 12.2 mostra un caso limite in cui i nodi di interpolazione non sono più distinti, noto come interpolazione di Hermite.

[2] Quindi sono delle delta di Kronecker:
$$L_i(x_j) = \begin{cases} 1 \ , \ i = j \\ 0 \ , \ i \neq j \end{cases} = \delta_{i,j}.$$

4.2 Forme polinomiali

Sia $p \in \mathbb{P}_n$. Vediamo altre possibili modalità di rappresentare il polinomio p oltre alla forma di Lagrange

Forma di potenze [3]

$$p = \sum_{i=0}^{n} a_i x^i$$

Forma traslata [4] sia $\bar{x} \in \mathbb{R}$,

$$p = b_0 + b_1(x - \bar{x}) + \dots b_n(x - \bar{x})^n$$

Forma Fattorizzata

$$p = c(x - x_1) \dots (x - x_n)$$

Forma di Horner [5] variante della forma di potenze

$$p = a_0 + x(a_1 + x(a_2 + \dots (a_{n-1} - xa_n) \dots))$$

Forma di Newton

Supponiamo di avere $p_n \in \mathbb{P}_n$ e di voler costruire $p_{n+1} \in \mathbb{P}_{n+1}$ aggiungendo un nuovo nodo di interpolazione x_{n+1}. In sostanza avremo quindi che p_{n+1} interpola la funzione anche in x_{n+1}. Introduciamo innanzitutto il polinomio $q := p_{n+1} - p_n \in \mathbb{P}_{n+1}$. Per ogni nodo $x_0, \dots, x_n$, risulta

$$q(x_i) = p_{n+1}(x_i) - p_n(x_i) = f(x_i) - f(x_i) = 0$$

ovvero sono note tutte le radici di q perciò possiamo scriverlo in forma fattorizzata

$$q(x) = \gamma(x - x_0) \dots (x - x_n) = \gamma x^n + \dots \text{[6]}.$$

Perciò avremo che

$$p_{n+1}(x) = q(x) + p_n(x) = \gamma x^n + \dots$$

ovvero

$$\gamma = f[x_0, \dots, x_{n+1}].$$

Riassumendo possiamo scrivere

$$p_{n+1}(x) = p_n(x) + f[x_0, \dots, x_{n+1}](x - x_0) \dots (x - x_n).\text{[7]}$$

Ora, se p_{n-1} interpola in $x_0, \dots, x_{n-1}$, allora

[3] Valutare in un punto un polinomio scritto in questa forma richiede circa $2n$ moltiplicazioni ed n addizioni.

[4] Il polinomio di Taylor si scrive in questa forma.

[5] Valutare in un punto un polinomio scritto in questa forma richiede circa n moltiplicazioni ed n addizioni.

[6] Termini di grado inferiore.

[7] Attenzione: al momento non abbiamo un meccanismo per calcolare le differenze divise.

$$p_n(x) = p_{n-1}(x) + f[x_0, \dots, x_n](x - x_0) \dots (x - x_{n-1}).$$

Ripetendo il ragionamento a ritroso fino a p_0 otteniamo

$$p_n(x) = p_0(x) + f[x_0, x_1](x - x_0) + \dots + f[x_0, \dots, x_n](x - x_0) \dots (x - x_{n-1})$$

e siccome p_0 interpola in x_0 ed è un polinomio costante abbiamo che $p_0(x) = f(x_0) = f[x_0]$; otteniamo così il polinomio interpolante in **forma di Newton**[8]

$$p_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + \dots + f[x_0, \dots, x_n](x - x_0) \dots (x - x_{n-1})$$

(4.2.1) Teorema *Vale la seguente formula per il calcolo delle differenze divise*

$$f[x_0, \dots, x_n] = \frac{f[x_0, \dots, x_{n-1}] - f[x_1, \dots, x_n]}{x_0 - x_n}.$$

Dimostrazione. Sia $p_{n-2} \in \mathbb{P}_{n-2}$. Possiamo costruire p_{n-1} in due modi:

- $p_{n-1}^{(1)} \in \mathbb{P}_{n-1}$ che interpola in $x_1, \dots, x_{n-1}$ ed x_0;
- $p_{n-1}^{(2)} \in \mathbb{P}_{n-1}$ che interpola in $x_1, \dots, x_{n-1}$ ed x_n;

quindi possiamo costruire in due modi p_n

$$p_n = p_{n-2} + f[x_1, \dots, x_{n-1}, x_0](x - x_1) \dots (x - x_{n-1}) + f[x_1, \dots, x_{n-1}, x_0, x_n](x - x_1) \dots (x - x_0)$$

e

$$p_n = p_{n-2} + f[x_1, \dots, x_{n-1}, x_n](x - x_1) \dots (x - x_{n-1}) + f[x_1, \dots, x_{n-1}, x_n, x_0](x - x_1) \dots (x - x_n)$$

ma per l'unicità del polinomio interpolante, uguagliando i secondi membri otteniamo

$$\begin{aligned}(x - x_1) \dots (x - x_{n-1}) \left[f[x_1, \dots, x_{n-1}, x_0] + f[x_1, \dots, x_{n-1}, x_0, x_n](x - x_0) \right] = \\ (x - x_1) \dots (x - x_{n-1}) \left[f[x_1, \dots, x_{n-1}, x_n] + f[x_1, \dots, x_{n-1}, x_n, x_0](x - x_n) \right]\end{aligned}$$

supponendo ora che x non sia un nodo posso semplificare l'espressione raggiungendo la forma

$$f[x_0, \dots, x_{n-1}] - f[x_1, \dots, x_n] = f[x_0, \dots, x_n](x - x_n) - f[x_0, \dots, x_n](x - x_0)$$

da cui con semplici manipolazioni algebriche si ottiene la tesi. ∎

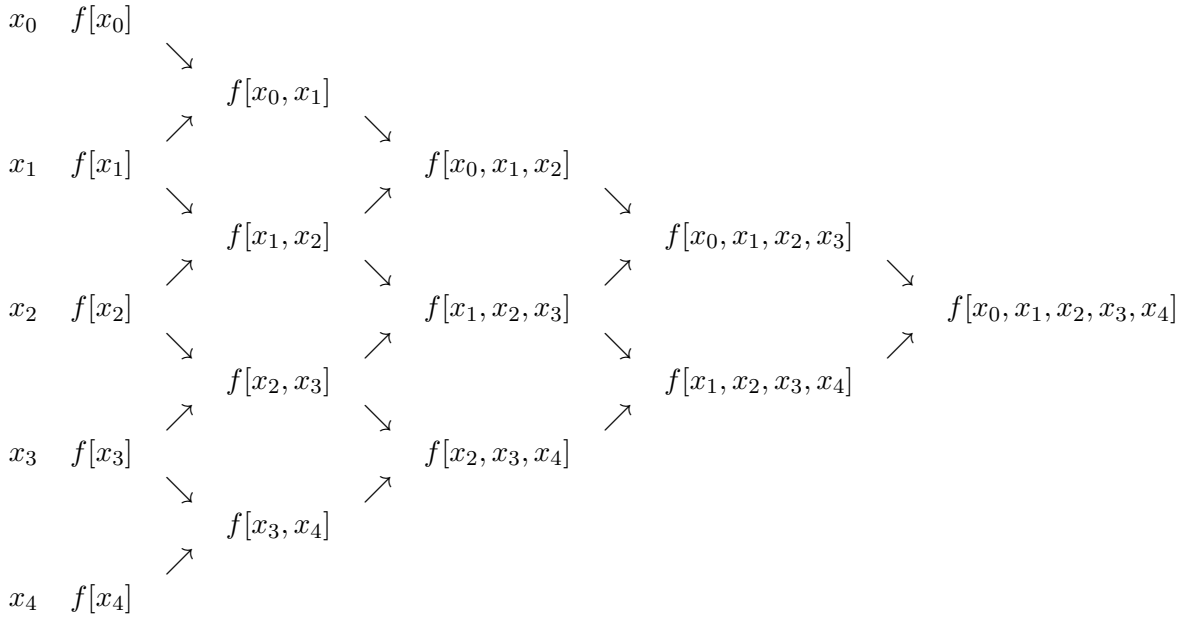
[8] Da notare l'interessante somiglianza con la forma traslata. Per assurdo se tutti i nodi coincidessero avremmo proprio l'uguaglianza delle due forme, cosa ovviamente impossibile per definizione di nodi.

(4.2.2) Osservazione *Nel caso di due nodi abbiamo*

$$f[x_0, x_1] = \frac{f[x_0] - f[x_1]}{x_0 - x_1} = \frac{f(x_0) - f(x_1)}{x_0 - x_1} = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

che è chiaramente un rapporto incrementale.

(4.2.3) Esempio *Sia $n = 5$. Visualizziamo il calcolo della differenza divisa di ordine 4.*



Collegamento con le esercitazioni

- L'Esercizio 12.1 e gli Homework 12.1, 12.2 e 12.4 mostrano ulteriori proprietà delle differenze divise.
- L'Esercizio 12.3 mostra l'implementazione del polinomio interpolante in forma di Newton, e come utilizzare tale forma nel caso in cui venga aggiunto un ulteriore nodo di interpolazione.
- L'Homework 11.2 mostra come utilizzare la forma di potenze porti tipicamente ad un algoritmo mal condizionato.
- L'Homework 12.3 richiede l'implementazione della forma di Horner per la valutazione di un polinomio interpolante.

4.3 Stima dell'errore

(4.3.1) Teorema *Sia $f \in C^n(a, b) \cap C^0[a, b]$ allora esiste $\xi \in I(x_0, \ldots, x_n)$[9] tale che*

$$f[x_0, \ldots, x_n] = \frac{f^{(n)}(\xi)}{n!}$$

[9] L'intervallo aperto $(\min\{x_0, \ldots, x_n\}, \max\{x_0, \ldots, x_n\})$.

Dimostrazione. Sia $E(x) := f(x) - p_n(x) \in C^n$ la funzione errore. Per semplicità assumiamo che i nodi siano ordinati[10]

Per ogni $i = 1, \dots, n$, per il Teorema di Rolle esiste $\xi_i \in (x_{i-1}, x_i)$ tale che $E'(\xi_i) = 0$. Ripetiamo il ragionamento per ogni $E^{(i)}$ fino a $E^{(n-1)}$: troviamo così $\xi \in (x_0, x_n)$ tale che $E^{(n)}(\xi) = 0$. Ora risulta

$$E^{(n)}(x) = f^{(n)}(x) - p_n^{(n)}(x)$$

e $p_n^{(n)}(x)$ è un polinomio costante della forma

$$p_n^{(n)}(x) = n! \cdot f[x_0, \dots, x_n].$$

Quindi possiamo scrivere

$$E^{(n)}(x) = f^{(n)}(x) - n! \cdot f[x_0, \dots, x_n]$$

e valutando la seguente espressione in ξ otteniamo la tesi. ■

(4.3.2) Teorema *Se* $x \notin \{x_0, \dots, x_n\}$

$$E(x) = f[x_0, \dots, x_n, x]\omega(x)$$

con $\omega(x) = (x - x_0) \dots (x - x_n) \in \mathbb{P}_n$.

Dimostrazione. Sia $\bar{x}$, un nuovo nodo. Allora

$$p_{n+1} = p_n + f[x_0, \dots, x_n, \bar{x}]\omega(x)$$

valutando la seguente espressione in $\bar{x}$ otteniamo

$$f(\bar{x}) = p_n + f[x_0, \dots, x_n, \bar{x}]\omega(\bar{x})$$

ed essendo $E(\bar{x}) = f(\bar{x}) - p_n$ otteniamo la tesi. ■

(4.3.3) Teorema *Se* $f \in C^{n+1}(a, b)$ *esiste* $\xi \in I(x_0, \dots, x_n, x)$ *tale che*

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

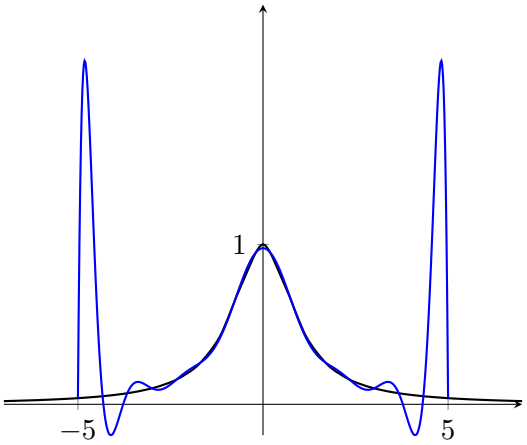
Dimostrazione. Omettiamo la dimostrazione. ■

(4.3.4) Osservazione *Possiamo fare le seguenti osservazioni:*

- *Il teorema precedente vale anche se* x *non è uno dei nodi;*
- *Se* x *è interno all'intervallo dei nodi si parla di vera e propria interpolazione, altrimenti di estrapolazione;*
- *L'errore ha un andamento simile ad* ω.

[10] O comunque ordiniamoli cambiando gli indici.

(4.3.5) Esempio (Controesempio di Runge) *Sia* $f(x) = \frac{1}{1+x^2}$.
Interpoliamo utilizzando nodi equi-spaziati e visualizziamo il risultato dell'interpolazione



Il caso a nodi equi-spaziati presenta un grosso problema vicino agli estremi dell'intervallo di interpolazione. Infatti ω *tende ad oscillare sempre di più all'avvicinarsi ai nodi estremi e di conseguenza l'errore aumenta. Per ovviare al problema possiamo operare alcune strategie:*

(0 *Interpolare con pochi punti)*

1 *Non pretendere l'interpolazione e tenere basso il grado; ad esempio con tecniche come i MINIMI QUADRATI, OTTIMA APPROSSIMAZIONE, . . . ;*

2 *Non prendere nodi equi-spaziati ma addensarli verso gli estremi;*[11]

3 *Non usare i polinomi ma altri tipi di funzioni; ad esempio mediante SPLINE, polinomi trigonometrici, . . .*

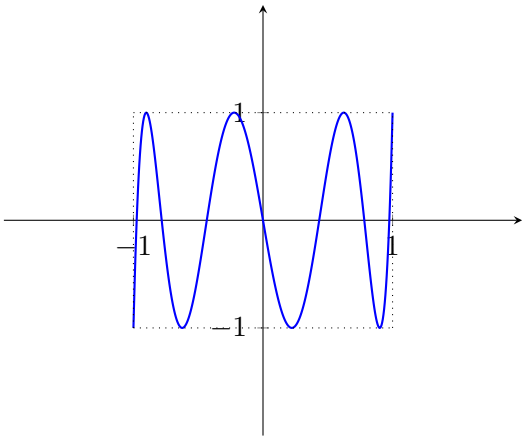
Collegamento con le esercitazioni

- L'Esercizio 11.1 discute una nozione di stabilità del problema di interpolazione rispetto a perturbazioni sui dati, introducendo la cosiddetta costante di Lebesgue.
- L'Esercizio 11.1 introduce il polinomio di ottima approssimazione in una specifica norma, e stabilisce una stima per quantificare quanto il polinomio interpolante sia "peggiore" di quello di ottima approssimazione.
- L'Esercizio 11.2 implementa il controesempio di Runge presentato nell'Esempio (4.3.5). L'Homework 11.3 propone di utilizzare la seconda strategia individuata nell'Esempio (4.3.5) per ovviare al problema manifestato dal controesempio.
- L'Homework 11.4 mostra un altro caso patologico per un problema di interpolazione, in cui il polinomio interpolante è identicamente nullo nonostante la funzione non lo sia.
- L'Homework 11.1 richiede di dimostrare un corollario del Teorema (4.3.3) per lo specifico caso di nodi equispaziati.

[11] I cosiddetti **Nodi di Chebyschev**.

4.4 Polinomi di Chebyschev

I polinomi di Chebyschev sono una particolare famiglia di polinomi ortogonali. Di seguito un esempio



(4.4.1) Definizione *Siano $f, g : [-1,1] \to \mathbb{R}$ due applicazioni. Sia $\omega : [-1,1] \to]0, +\infty]$ con le seguenti proprietà:*

- $\omega(x) \geq \omega(0) > 0$;
- ω *è integrabile in senso improprio, ovvero*

$$\int_{-1}^{1} \omega(x)dx < +\infty.$$

Chiamiamo ***prodotto scalare di Chebyschev*** $L^2_\omega(-1,1)$ *la funzione così definita*

$$(f|g)_{L^2_\omega(-1,1)} = \int_{-1}^{1} f(x)g(x)\omega(x)dx.$$

In particolare $L^2_\omega(-1,1)$ risulta effettivamente un prodotto scalare sullo spazio vettoriale delle funzioni $f : [-1,1] \to \mathbb{R}$.

(4.4.2) Definizione *Sia $n \in \mathbb{N}$. Chiamiamo* ***polinomio di Chebyschev*** *la funzione*

$$T_n : \begin{cases} [-1,1] \to \mathbb{R} \\ x \mapsto \cos(n\theta) \end{cases}$$

dove $\theta := \arccos x$.

(4.4.3) Osservazione *Notiamo subito che $T_0 \equiv 1 \in \mathbb{P}_0$ e $T_1 = x \in \mathbb{P}_1 \setminus \mathbb{P}_0$.*

(4.4.4) Proposizione *Valgono le seguenti proprietà:*

(*a*) ***ricorrenza tripla****, ovvero*

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x);$$

(*b*) $T_n \in \mathbb{P}_n \setminus \mathbb{P}_{n-1}$, *ovvero è un polinomio di grado* n;

(*c*) *se* $n \geq 1$ *allora* $T_n(x) = 2^{n-1}x^n + \ldots$;

(*d*) ***equi-oscillazione***, *ovvero* $|T_n(x)| \leq 1$ *e vale* ± 1 *in* $n+1$ *punti distinti della forma* $\cos\frac{k\pi}{n}$ *per* $k = 0, \ldots, n$;

(*e*) $T_n(x)$ *possiede* n *radici distinte, dette* ***nodi di Chebyschev***, *in* $]-1,1[$ *della forma*

$$x_k = \cos\frac{\frac{\pi}{2} + k\pi}{n}\ ,\ k = 0, \ldots, n-1;$$

(*f*) ***ortogonalità***, *ovvero posto* $\omega(x)$[12]$= \frac{1}{\sqrt{1-x^2}}$, *risulta*

$$(T_i|T_j)_{L^2_\omega(-1,1)} = \begin{cases} 0\ ,\ i \neq j \\ \neq 0\ ,\ i = j \end{cases}\quad .$$

Dimostrazione.

(a) Siccome

$$T_{n+1}(x) = \cos(\theta + n\theta) = cos\theta\cos n\theta - \sin\theta\sin n\theta$$

e

$$T_{n-1}(x) = \cos(\theta - n\theta) = cos\theta\cos n\theta + \sin\theta\sin n\theta$$

sommando ambo i membri otteniamo

$$T_{n+1}(x) + T_{n-1}(x) = 2\cos\theta\cos n\theta = 2xT_n(x).$$

(b) Discende direttamente dalla (a).

(c) Discende direttamente dalla (a).

(d) Innanzitutto per le proprietà della funzione coseno è banale che $|T_n(x)| \leq 1$; inoltre

$$|T_n(x)| = 1 \Leftrightarrow n\arccos x = k\pi\ ,\ k \in \mathbb{Z}.$$

Osserviamo subito che in realtà $k \in [0, n]$ siccome compare l'arcocoseno. In definitiva abbiamo quindi

$$x_k = \cos\frac{k\pi}{n}.$$

(e) Analogamente a quanto fatto al passo (d), ricordando che

$$|T_n(x)| = 0 \Leftrightarrow n\arccos x = \frac{\pi}{2} + k\pi\ ,\ k \in \mathbb{Z}$$

e che comparendo l'arcocoseno in realtà $k \in [0, n-1]$ otteniamo

$$x_k = \cos\frac{\frac{\pi}{2} + k\pi}{n}.$$[13]

[12] Il cosiddetto **peso di Chebyschev**.

[13] Notiamo che gli zeri si addensano agli estremi dell'intervallo $[-1, 1]$.

(f) Calcoliamo

$$\int_{-1}^{1} T_i(x)T_j(x)\frac{1}{\sqrt{1-x^2}}dx$$

effettuiamo un cambio di variabile $\theta = \arccos x$, ovvero $x = \cos\theta$

$$\int_0^{\pi} \cos i\theta \cos j\theta \frac{\sin\theta}{\sqrt{1-\cos^2\theta}}d\theta = \int_0^{\pi} \cos i\theta \cos j\theta d\theta$$

applicando la formula di Werner per il coseno otteniamo

$$\int_0^{\pi} \frac{1}{2}\left\{\cos\left[(i+j)\,\theta\right] + \cos\left[(i-j)\,\theta\right]\right\}d\theta = \begin{cases} 0\ ,\ i \neq j \\ \pi\ ,\ i = j = 0 \\ \frac{\pi}{2}\ ,\ i = j \neq 0 \end{cases}.^{14}$$ ■

(4.4.5) Proposizione *Se ψ_k è un polinomio di grado esattamente k allora*

$$(\psi_0, \ldots, \psi_n)$$

è una base dello spazio $\mathbb{P}_n$.

Dimostrazione. Omettiamo la dimostrazione. ■

(4.4.6) Teorema (Processo di ortogonalizzazione di Gram-Schmidt) *Sia (ψ_k) una base nello spazio $\mathbb{P}_n$. Allora posso costruire una base (φ_k) ortogonale.*

Dimostrazione. Procediamo per induzione su k, ponendo innanzitutto $\varphi_0 = \psi_0$. Definiamo

$$\varphi_1 := \psi_1 - \alpha\varphi_0$$

con α un coefficiente che rende ortogonali i due vettori, ovvero $(\varphi_1, \varphi_0) = 0$; per linearità del prodotto scalare possiamo scrivere

$$(\varphi_1, \varphi_0) = (\psi_1, \varphi_0) - \alpha\,(\varphi_0, \varphi_0) = 0$$

da cui

$$\alpha = \frac{(\psi_1, \varphi_0)}{(\varphi_0, \varphi_0)}$$

Supponiamo ora di avere costruito

$$\{\varphi_0, \ldots, \varphi_k\}\,.$$

Definiamo

[14] Abbiamo omesso un semplice passaggio analitico. Infatti

$$\int_0^{\pi} \cos k\theta d\theta = \begin{cases} \frac{1}{k}\left[\sin k\theta\right]_0^{\pi} = 0\ ,\ k \neq 0 \\ \pi\ ,\ k = 0 \end{cases}.$$

$$\varphi_{k+1} := \psi_{k+1} - \sum_{j=0}^{k} \lambda_j \varphi_j$$

imponendo che per ogni $i \in [0, \dots, k]$ risulti $(\varphi_{k+1}, \varphi_i) = 0$; sempre per la linearità del prodotto scalare possiamo scrivere

$$(\varphi_{k+1}, \varphi_i) = (\psi_{k+1}, \varphi_i) - \sum_{j=0}^{k} \lambda_j (\varphi_j, \varphi_i) = 0$$

ma siccome per ipotesi induttiva i φ_i sono ortogonali tra loro, la sommatoria ha un unico termine non nullo, ovvero

$$(\psi_{k+1}, \varphi_i) - \lambda_i (\varphi_i, \varphi_i) = 0$$

per cui in definitiva

$$\lambda_i = \frac{(\psi_{k+1}, \varphi_i)}{(\varphi_i, \varphi_i)}. \blacksquare$$

(4.4.7) Osservazione *Possiamo fare due osservazioni in merito a quanto abbiamo appena dimostrato:*

- $(\varphi_k)_0^n$*, base ortogonale di* $\mathbb{P}_n$ *con* $\varphi_k \in \mathbb{P}_k \backslash \mathbb{P}_{k-1}$ *è unica a meno di fattori moltiplicativi non nulli;*
- *se consideriamo il prodotto scalare di Chebyschev* $(f, g)_{L^2_\omega(-1,1)}$*,* $(T_k)_0^n$ *è l'"unica" base ortogonale.*

Collegamento con le esercitazioni

- L'Esercizio 13.1 mostra una implementazione dei polinomi di Chebyschev.
- L'Esercizio 13.2 e gli Homework 13.1 e 13.2 dimostrano ulteriori proprietà dei polinomi di Chebyschev.

4.5 Polinomi di Legendre

(4.5.1) Definizione *Consideriamo l'intervallo* $]-1, 1[$*. Prendiamo*

$$\omega(x) = 1$$

e di conseguenza

$$(f, g)_{L^2(-1,1)} = \int_{-1}^{-1} f(x) g(x) dx.$$

Chiamiamo $(f, g)_{L^2(-1,1)}$ ***prodotto scalare di Legendre****.*

Scritto da Mattia Garatti

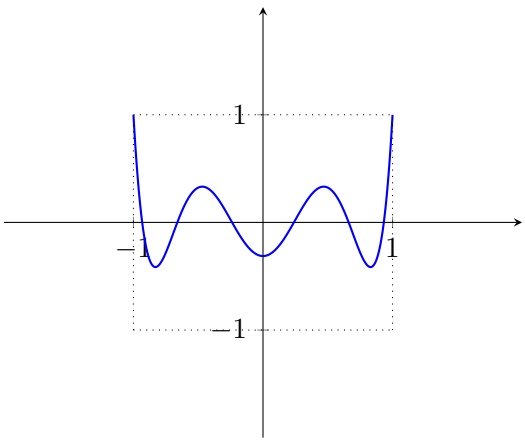
(4.5.2) Definizione *Chiamiamo* ***polinomi di Legendre*** *una famiglia di polinomi* $\{P_k\}$ *costruiti in questo modo:*

$$P_0(x) \equiv 0$$
$$P_1(x) = x$$
$$\vdots$$
$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x).$$

In generale hanno la forma

$$P_n(x) = (2^n \cdot n!)^{-1} \frac{\mathrm{d}^n}{\mathrm{d}x^n} \left[(x^2-1)^n\right].$$

Di seguito un esempio di un polinomio di Legendre



(4.5.3) Teorema $\{P_k\}$ *sono ortogonali rispetto al prodotto scalare di Legendre.*

Dimostrazione. Omettiamo la dimostrazione. ■

Collegamento con le esercitazioni

- L'Esercizio 13.1 mostra una implementazione dei polinomi di Legendre.
- Dato un generico peso, l'Homework 13.3 mostra come costruire una famiglia di polinomi che siano ortogonali rispetto a tale peso. Alcune famiglie notevoli di polinomi ortogonali che possono essere ottenute in tal modo sono riportate nell'Homework 13.4.

Capitolo 5

Minimi quadrati

5.1 I limiti dell'interpolazione polinomiale

Iniziamo il capitolo con alcuni esempi per chiarire l'importanza di quanto affronteremo.

(5.1.1) Esempio *Si consideri l'intervallo* $I = [-5, 5]$*, e la funzione*

$$r : \begin{cases} I \to \mathbb{R} \\ x \mapsto r(x) = \frac{1}{1+x^2} \end{cases}$$

come nell'esempio (4.3.5)*. Consideriamo il problema di determinare il polinomio interpolante* $\Pi_n(r) \in \mathbb{P}_n$ *utilizzando* $n+1$ *nodi di interpolazione equi-spaziati.*

Il valore di n *compare con due ruoli:*

- *in relazione al numero di "dati" del problema di interpolazione, che sono gli* $n+1$ *nodi di interpolazione (equi-spaziati, nel nostro caso),*
- *in relazione al grado del polinomio interpolante* $\Pi_n(r)$ *o, in modo equivalente, alla dimensione dello spazio* $\mathbb{P}_n$ *in cui cerchiamo la soluzione del problema di interpolazione.*

In un problema di interpolazione il numero dei dati, gli $n+1$ *nodi, ci vincola per definizione di interpolazione, a ricercare il polinomio interpolante in uno spazio* $\mathbb{P}_n$ *ma al crescere di* n *questo vincolo può deteriorare la qualità del polinomio interpolante, specialmente se i nodi sono scelti male come già abbiamo potuto osservare.*

In questo capitolo vedremo come è possibile rendere indipendente il numero $m+1$ dei dati del problema dalla dimensione dello spazio $\mathbb{P}_n$ in cui se ne cerca la soluzione.[1]

5.2 Il caso discreto

Considereremo un intervallo $I = [a, b]$, una funzione continua $f : I \to \mathbb{R}$, $n \in \mathbb{N}$, $\{x_0, \ldots, x_m\}$, $m+1$ nodi distinti in I e le corrispondenti immagini $\{y_0, \ldots, y_m\}$.

[1] Dovremo ovviamente cambiare la definizione del problema, perché richiedere che valgano le condizioni di interpolazione per $m \neq n$ potrebbe portare a definire un problema di interpolazione per il quale non esiste soluzione oppure la soluzione non è unica.

(5.2.1) Definizione *Siano $g, h : I \to \mathbb{R}$ ed $m \geq n \in \mathbb{N}$. Chiamiamo* ***prodotto scalare discreto***

$$(g|h)_m = \sum_{i=0}^{m} g(x_i)h(x_i).$$

In particolare questo prodotto scalare indurrà la corrispondente ***norma discreta*** *così definita*

$$||g||_m = \sqrt{(g|g)_m} = \sqrt{\sum_{i=0}^{m} g(x_i)^2}.$$

(5.2.2) Definizione *Si consideri l'intervallo $I = [a, b]$ e una funzione continua $f : I \to \mathbb{R}$. Sia $\Phi_n(f) \in \mathbb{P}_n$ tale che*

$$||\Phi_n(f) - f||_m^2 \leq ||p_n - f||_m^2 \ , \ \forall \ p_n \in \mathbb{P}_n.$$

Chiamiamo $\Phi_n(f)$ ***migliore approssimazione*** *di f nello spazio $\mathbb{P}_n$ rispetto alla norma $||\cdot||_m$.*

(5.2.3) Osservazione *Problemi ai minimi quadrati nel discreto sono molto utilizzati in vari ambiti, ad esempio:*

- ***data fitting*** *in scienza dei dati, dove le coppie $\{(x_0, y_0), ..., (x_m, y_m)\}$ corrispondono ad esempio a dati sperimentali;*
- ***problemi di regressione*** *in statistica, ad esempio il caso della regressione lineare che corrisponde alla scelta $n = 1$.*

Collegamento con le esercitazioni

- L'Esercizio 14.1 discute un caso di problema ai minimi quadrati nel discreto.
- L'Homework 14.1 quantifica la robustezza del polinomio di migliore approssimazione in applicazioni di data fitting a dati sperimentali affetti da rumore.

5.3 Il caso continuo

Il caso continuo è studiato specialmente nell'ambito di problemi di migliore approssimazione in spazi di Hilbert.

(5.3.1) Definizione *Dato un intervallo $I = [a, b]$, una funzione $f : I \to \mathbb{R}$ continua e una funzione peso $\omega : I \to]0, +\infty]$ tale che*

(a) *esiste $\omega_0 > 0$ tale che per ogni $x \in I$ risulta $\omega(x) \geq \omega_0$;*

(b) *$\int_I \omega(x)dx < +\infty$.*

Si consideri il prodotto scalare pesato

$$(f|g)_\omega = \int_I f(x)g(x)\omega(x)dx$$

e la corrispondente norma indotta

$$||g||_\omega^2 = (g|g)_\omega = \int_I g(x)^2\omega(x)dx.$$

Definiamo ***migliore approssimazione*** *di f nello spazio $\mathbb{P}_n$ rispetto alla norma $||\cdot||_\omega$ il polinomio $\Phi_n(f) \in \mathbb{P}_n$ tale che*

$$||\Phi_n(f) - f||_\omega^2 \leq ||p_n - f||_\omega^2 \ , \ \forall\, p_n \in \mathbb{P}_n.$$

Collegamento con le esercitazioni L'Esercizio 14.2 e l'Homework 14.2 discutono un caso di problema ai minimi quadrati nel continuo.

5.4 Proprietà di ortogonalità

In questa sezione quando omettiamo i pedici dai simboli di prodotto scalare e norma intendiamo che i risultati valgono sia nel caso discreto che in quello continuo.

(5.4.1) Definizione *Due funzioni $f, g : I \to \mathbb{R}$ continue sono ortogonali rispetto al prodotto scalare $(\ |\ )$ se*

$$(f|g) = 0.$$

Denotiamo tale relazione con $f \perp g$.

(5.4.2) Definizione *Dato un intervallo $I = [a, b]$, una funzione $f : I \to \mathbb{R}$ continua e un prodotto scalare $(\ |\ ) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ diciamo che $p_n^* \in \mathbb{P}_n$ verifica la* ***proprietà di ortogonalità*** *se*

$$f - p_n^* \perp \mathbb{P}_n$$

ovvero $f - p_n^*$ è ortogonale ad ogni polinomio $p \in \mathbb{P}_n$.

(5.4.3) Teorema *Siano un intervallo $I = [a, b]$, una funzione $f : I \to \mathbb{R}$ continua e un prodotto scalare $(\ |\ ) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$. Sono fatti equivalenti:*

(*a*) *p_n^* verifica la proprietà di ortogonalità;*

(*b*) *p_n^* è migliore approssimazione ai minimi quadrati.*

Dimostrazione.

$(a) \Rightarrow (b)$ Assumiamo che p_n^* soddisfi la proprietà di ortogonalità, cioè per ogni $p_n \in \mathbb{P}_n$ valga $f - p_n^* \perp p_n$. Calcoliamo

$$\begin{aligned} ||f - p_n||^2 = ||(f - p_n^*) + (p_n^* - p_n)||^2 = \\ = ((f - p_n^*) + (p_n^* - p_n)|(f - p_n^*) + (p_n^* - p_n)) = \\ = ||f - p_n^*||^2 + 2\,(f - p_n^*|p_n^* - p_n) + ||p_n^* - p_n||^2. \end{aligned}$$

Ora siccome vale la proprietà di ortogonalità il prodotto scalare centrale è nullo quindi abbiamo

$$||f - p_n||^2 = ||f - p_n^*||^2 + ||p_n^* - p_n||^2 \geq ||f - p_n^*||^2$$

da cui possiamo dire che p_n^* è una soluzione del problema dei minimi quadrati. In realtà essa è anche unica perché supponendo esista $p_n^\bigstar \in \mathbb{P}_n$, un'altra soluzione del problema dei minimi quadrati, con passaggi analoghi ai precedenti possiamo scrivere

$$||f - p_n^\bigstar||^2 = ||f - p_n^*||^2 + ||p_n^\bigstar - p_n||^2$$

e quindi deve essere per forza

$$||p_n^\bigstar - p_n||^2 = 0$$

da cui l'unicità.

$(b) \Rightarrow (a)$ Supponiamo che p_n^* sia soluzione del problema dei minimi quadrati. Per ogni $p_n \in \mathbb{P}_n$ definiamo la funzione

$$\varphi : \begin{cases} \mathbb{R} \to \mathbb{R}^+ \\ t \mapsto \varphi(t) = ||f - (p_n^* + t \cdot p_n)||^2 \end{cases}.$$

Siccome $\varphi(0) = ||f - p_n^*||^2$ e per ipotesi p_n^* è soluzione del problema dei minimi quadrati abbiamo che $t = 0$ è un punto di minimo per la funzione φ, ed essendo essa di classe C^∞ possiamo scrivere

$$\varphi'(0) = 0.$$

Ora siccome

$$\varphi(t) = ||f - p_n^*||^2 - 2t\,(f - p_n^*|p_n) + t^2||p_n||^2$$

derivando otteniamo

$$\varphi'(t) = -2\,(f - p_n^*|p_n) + 2t||p_n||^2.$$

In conclusione quindi

$$0 = \varphi'(0) = -2\,(f - p_n^*|p_n)$$

e perciò per l'arbitrarietà di p_n vale la proprietà di ortogonalità. ■

Scritto da Mattia Garatti

(5.4.4) Teorema *Siano un intervallo $I = [a, b]$, una funzione $f : I \to \mathbb{R}$ continua e un prodotto scalare $(\ |\ ) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$. Si fissi una base per $\mathbb{P}_n$ in modo che*

$$\mathbb{P}_n = \langle \varphi_0, \ldots, \varphi_n \rangle.$$

Sono fatti equivalenti:

(*a*) *p_n^* verifica la proprietà di ortogonalità;*

(*b*) *detti c_i i coefficienti del vettore $\mathbf{c} \in \mathbb{R}^{n+1}$, soluzione del sistema lineare*

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

dove $A \in Mat_{n+1}$ tale che $a_{i,j} = (\varphi_i|\varphi_j)$ e $\mathbf{b} \in \mathbb{R}^{n+1}$ tale che $b_i = (f|\varphi_i)$ risulta

$$p_n^* = \sum_{i=0}^{n} c_i \varphi_i.$$

Dimostrazione.

(*a*) $\Rightarrow$ (*b*) Sia p_n^* soddisfacente la proprietà di ortogonalità e $(\varphi_0, \ldots, \varphi_n)$ una base di $\mathbb{P}_n$. Ogni polinomio di $\mathbb{P}_n$ è combinazione lineare di certi φ_i e quindi per linearità del prodotto scalare la proprietà di ortogonalità può essere riscritta rispetto alla base scelta

$$(f - p_n^*|\varphi_i) = 0\ ,\ \forall\, i = 0, \ldots, n.$$

Se ora prendiamo $c_0, \ldots, c_n$ tali che

$$p_n^* = \sum_{j=0}^{n} c_j \varphi_j$$

otteniamo

$$\left(f - \sum_{j=0}^{n} c_j \varphi_j | \varphi_i \right) = 0\ ,\ \forall\, i = 0, \ldots, n$$

ovvero per la linearità del prodotto scalare

$$(f|\varphi_i) = \sum_{j=0}^{n} c_j\, (\varphi_i|\varphi_j)\ \ ,\ \forall\, i = 0, \ldots, n.$$

Siano ora $A \in Mat_{n+1}$ tale che $a_{i,j} = (\varphi_i|\varphi_j)$ e $\mathbf{b} \in \mathbb{R}^{n+1}$ tale che $b_i = (f|\varphi_j)$. Possiamo riscrivere la relazione nel seguente modo

$$\sum_{j=0}^{n} c_j A_{i,j} = b_i\ ,\ \forall\, i = 0, \ldots, n$$

Scritto da Mattia Garatti

e prendendo un vettore $\mathbf{c} = (c_0, \ldots, c_n)^t \in \mathbb{R}^{n+1}$ otteniamo

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

$(b) \Rightarrow (a)$ È sufficiente ripercorrere a ritroso i passaggi del punto precedente. ■

(5.4.5) Proposizione *Sia $A \in Mat_{n+1}$ la matrice del sistema lineare equivalente al problema ai minimi quadrati. Allora A è una matrice simmetrica definita positiva.*

Dimostrazione. Innanzitutto la matrice A è simmetrica per costruzione.
Sia poi $\mathbf{x} \in \mathbb{R}^{n+1} \setminus \{0\}$,

$$\mathbf{x}^T A\mathbf{x} = \sum_{i=0}^{n}\sum_{j=0}^{n} x_i A_{ij} x_j = \sum_{i=0}^{n}\sum_{j=0}^{n} x_i \left(\varphi_j|\varphi_i\right) x_j = \sum_{i=0}^{n}\sum_{j=0}^{n} x_i \left(\varphi_i|\varphi_j\right) x_j$$

da cui per bilinearità del prodotto scalare

$$\mathbf{x}^T A\mathbf{x} = \left(\sum_{i=0}^{n} x_i\varphi_i | \sum_{j=0}^{n} x_j\varphi_j\right).$$

I coefficienti x_i definiscono quindi univocamente un polinomio $e := \sum_{i=0}^{n} x_i\varphi_i$ grazie al quale possiamo concludere che

$$\mathbf{x}^T A\mathbf{x} = (e|e) = ||e||^2$$

e, siccome $\mathbf{x} \neq \mathbf{0}$, $e \neq 0$ da cui la definita positività. ■

(5.4.6) Osservazione *Siamo liberi di scegliere la base, purché generi lo spazio $\mathbb{P}_n$. La matrice A è simmetrica definita positiva a prescindere da quale sia la scelta della base. Tuttavia, il numero di condizionamento invece dipende dalla scelta della base:*

- *una scelta sbagliata, ad esempio la base dei monomi, porterà ad una matrice A molto mal-condizionata se n non è troppo piccolo;*
- *una scelta opportuna della base, ad esempio una base ortogonale, porterà ad una matrice A ben condizionata e anche altre comode proprietà, come il fatto di essere diagonale.*

Collegamento con le esercitazioni

- Entrambi gli Esercizi 14.1 e 14.2 confronteranno diverse scelte per la base dello spazio $\mathbb{P}_n$, in linea con l'Osservazione (5.4.6).
- L'Homework 14.3 mostra alcuni semplici casi di migliore approssimazione in spazi di Banach, anziché di Hilbert.

5.5 Retta di regressione lineare

Affrontiamo ora un esempio in cui il problema ai minimi quadrati può essere risolto a mano, senza l'ausilio di algoritmi. In generale comunque si possono usare i metodi visti nel capitolo sui sistemi lineari, facendo attenzione al tipo di matrice che di volta in volta si troverà.

(5.5.1) Esempio *Consideriamo il caso discreto, per un generico intervallo I ed i seguenti dati:*

- *grado massimo del polinomio $n = 1$;*
- *$m + 1$ nodi distinti $x_0, \dots, x_m$ con $x_0 \in I$, e le corrispondenti valutazioni $y_0, \dots, y_m$.*

La retta di regressione lineare[2] *è una retta*

$$p_1^*(x) = sx + q.$$

Posto

$$\bar{x} := \frac{\sum_{k=0}^{m} x_k}{m+1}$$

la media aritmetica dei dati, e $\bar{y}$ la corrispondente media aritmetica delle valutazioni, troviamo i coefficienti s e q.

Consideriamo la base dei monomi di $\mathbb{P}_1$, ovvero

$$(1, x).$$

La matrice del sistema lineare sarà la seguente

$$A = \begin{pmatrix} m+1 & (m+1)\bar{x} \\ (m+1)\bar{x} & (m+1)\overline{x^2} \end{pmatrix}$$

ed il vettore dei termini noti

$$\mathbf{b} = \begin{pmatrix} (m+1)\bar{y} \\ (m+1)\overline{xy} \end{pmatrix}.$$

Potendo scrivere $p_1^*(x) = q \cdot 1 + s \cdot x$, ovvero come combinazione lineare di vettori della base scelta, s e q saranno la soluzione del sistema lineare

$$\begin{pmatrix} 1 & \bar{x} \\ \bar{x} & \overline{x^2} \end{pmatrix} \begin{pmatrix} q \\ s \end{pmatrix} = \begin{pmatrix} \bar{y} \\ \overline{xy} \end{pmatrix}$$

ovvero

$$s = \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}^2}\ ,\ q = \bar{y} - \bar{x}\frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}^2}.$$[3]

Collegamento con le esercitazioni L'Homework 14.4 introduce una versione non lineare di regressione utilizzata spesso in statistica.

[2] Denominazione utilizzata in statistica per indicare la soluzione di questo problema ai minimi quadrati nel discreto.

[3] Ricordiamo che la media aritmetica del prodotto è diversa dal prodotto delle medie aritmetiche.

Capitolo 6

Integrazione numerica

6.1 Formule di quadratura

Vogliamo ora affrontare il problema di calcolare l'integrale di una funzione continua definita su un intervallo. Ovvero data

$$f : [a, b] \to \mathbb{R}$$

vogliamo determinare

$$I(f) := \int_a^b f(x)\, dx.$$

L'idea principale è approssimare $I(f)$ mediante **formule di quadrature**, ovvero formule del tipo

$$\tilde{I}(f) = \sum_{i=1}^{k} w_i f(x_i)$$

dove $\tilde{I}(f) \approx I(f)$, gli w_i sono k **pesi** e gli x_i sono k punti distinti dell'intervallo $[a, b]$[1] detti **nodi**.

(6.1.1) Definizione *Data $\tilde{I}(f) \approx I(f)$ ed $r \in \mathbb{N}$ diciamo che $\tilde{I}(f)$ ha **grado di precisione*** almeno *m se per ogni $f \in \mathbb{P}_m$ risulta che $\tilde{I}(f) = I(f)$. Diremo in particolare che il grado di precisione è* esattamente *m se esiste $f \in \mathbb{P}_{m+1}$ tale che $\tilde{I}(f) \neq I(f)$.*

(6.1.2) Esempio (Formula dei trapezi composita) *Sia $n \in \mathbb{N}$, dividiamo l'intervallo $[a, b]$ in n intervalli di ampiezza $h = \frac{b-a}{n}$ in modo da ottenere $n+1$ nodi del tipo $x_i = a + ih$ per $i = 0, \ldots, n$. Consideriamo il segmento che congiunge due nodi consecutivi ed i segmenti proiezione dei punti del grafico di ascissa i nodi. Otteniamo n trapezi la cui somma delle aree approssima dal basso l'area sottesa alla curva, ovvero*

$$\tilde{I}(f) = \frac{h}{2}\left[f(a) + 2f(x_1) + \cdots + 2f(x_{n-1}) + f(b)\right]$$

ovvero ponendo $w_0 = w_n = \frac{h}{2}$ e $w_i = h$ per $i = 1, \ldots, n-1$, otteniamo la formula di quadratura nella forma vista sopra.

[1] A rigore non è tuttavia obbligatorio che tutti i nodi siano punti dell'intervallo.

Determiniamo il grado di precisione: se $f \in \mathbb{P}_1$ allora f è una retta e sicuramente per costruzione la formula dei trapezi composita restituirà l'integrale esatto. Allora il grado di precisione è sicuramente almeno 1. *Sia ora però $f = (x-a)(x-b)$ un polinomio di secondo grado. In questo caso l'integrazione approssimata sarà inferiore a quella esatta quindi possiamo concludere che il grado di precisione è esattamente 1.*

Collegamento con le esercitazioni Nell'Esempio (6.1.2) abbiamo determinato il grado di precisione di una formula di quadratura ottenuta fissando $n+1$ nodi e $n+1$ pesi. Al contrario, nell'Esercizio 15.1 determiniamo, se possibile, gli $n+1$ pesi affinché una formula di quadratura associata ad $n+1$ nodi dati abbia un grado di precisione da noi desiderato. Un procedimento simile può essere seguito anche nel caso di integrali pesati, come nell'Homework 15.3.

6.2 Formule di quadratura interpolatorie

Vediamo ora una tecnica di costruzione di formule di quadratura basata sull'utilizzo delle tecniche di interpolazione polinomiale.

Sia un'applicazione $f : [a,b] \to \mathbb{R}$. Fissiamo $n \in \mathbb{N}$, prendiamo $n+1$ nodi distinti nell'intervallo $[a,b]$. Allora esisterà un'unico polinomio $\tilde{f} \in \mathbb{P}_n$ che interpola f nei nodi. Definiamo

$$\tilde{I}(f) := I(\tilde{f})$$

ovvero approssimiamo l'integrale di f mediante l'integrale del polinomio interpolante di f.

Se ora scriviamo $\tilde{f}$ in forma di Lagrange

$$\tilde{f} = \sum_{i=0}^{n} f(x_i) L_i(x)$$

perciò dalla linearità dell'integrale

$$\tilde{I}(f) = \int_a^b \sum_{i=0}^{n} f(x_i) L_i(x)\, dx = \sum_{i=0}^{n} f(x_i) \int_a^b L_i(x)\, dx$$

dove ponendo $w_i := \int_a^b L_i(x)\, dx$ otteniamo la formula di quadratura interpolatoria di f.

(6.2.1) Teorema *Una formula di quadratura ad $n+1$ nodi è interpolatoria se e solo se il suo grado di precisione è almeno pari ad n.*

Dimostrazione. Supponiamo che la formula di quadratura sia interpolatoria. Se $f \in \mathbb{P}_n$ allora f interpola se stessa, ovvero $\tilde{f} \equiv f$, da cui

$$\tilde{I}(f) = I(\tilde{f}) = I(f)$$

per cui possiamo concludere che il grado di precisione è almeno n.

Viceversa, supponiamo che, detto m il grado di precisione, si abbia $m \geq n$. Tra i polinomi di grado n ci sono i $L_i(x)$, quindi essi vengono integrati esattamente, ovvero

$$\tilde{I}(L_i) = I(L_i) = \int_a^b L_i(x)\, dx$$

ma d'altra parte

$$\tilde{I}(L_i) = \sum_{j=0}^{n} w_j L_i(x_j) = w_i L_i(x_i) = w_i$$

allora $w_i = \int_a^b L_i(x)\,dx$, ovvero la formula di quadratura è interpolatoria. ■

(6.2.2) Osservazione *Per formule di quadratura interpolatorie può accadere che $m > n$.*

(6.2.3) Proposizione *Se nodi e pesi sono simmetrici rispetto al centro dell'intervallo $[a, b]$ allora il grado di precisione è dispari.*

Dimostrazione. Supponiamo per assurdo che m sia pari. Consideriamo $\bar{f} = (x - c)^{m+1}$ dove $c = \frac{a+b}{2}$ è il centro dell'intervallo $[a, b]$. Facilmente si osserva che

$$I(\bar{f}) = 0.$$

Risulta anche che

$$\tilde{I}(\bar{f}) = 0$$

siccome i nodi sono disposti simmetricamente rispetto al centro e si bilanciano a coppie.

Allora $\bar{f}$ è integrabile esattamente, ma siccome possiamo scrivere il generico polinomio di $\mathbb{P}_{m+1}$ come

$$p(x) = \alpha \bar{f} + \kappa(x)$$

con $\kappa(x)$ un polinomio di grado m risulta che tutti i polinomi di grado $m + 1$ vengono integrati esattamente, che è assurdo. ■

6.3 Formule di Newton - Cotes

Sono una famiglia di formule interpolatorie a nodi equi-spaziati. Si distinguono in chiuse ed aperte.

Formule chiuse

Sia $n \geq 1 \in \mathbb{N}$, suddividiamo l'intervallo $[a, b]$ in n sotto-intervalli di ampiezza $h = \frac{b-a}{n}$. I nodi saranno del tipo

$$x_i = a + ih\ ,\ i = 0, \ldots, n.$$

Una volta scelti i nodi costruisco la relativa formula interpolatoria.

Scritto da Mattia Garatti

Formule aperte

Sia $n \geq 0 \in \mathbb{N}$, suddividiamo l'intervallo $[a, b]$ in $n + 2$ sotto-intervalli di ampiezza $h = \frac{b-a}{n+2}$. I nodi saranno del tipo

$$x_i = a + (i + 1)h \ , \ i = 0, \ldots, n.$$

Una volta scelti i nodi costruisco la relativa formula interpolatoria.

(6.3.1) Teorema *Data $\tilde{I}$ una generica formula di Newton-Cotes ed n il numero di nodi, valgono i seguenti fatti:*

(*a*) *il grado di precisione vale n se n è dispari; vale $n + 1$ se n è pari;*

(*b*) *detto $\varepsilon(f) := I(f) - \tilde{I}(f)$, l'errore di integrazione, esiste $\xi \in [a, b]$ tale che*

$$\varepsilon(f) = \begin{cases} K_n \frac{f^{(n+1)}(\xi)}{(n+1)!} \cdot h^{n+2} \ , \ n \text{ dispari} \\ K_n \frac{f^{(n+2)}(\xi)}{(n+2)!} \cdot h^{n+3} \ , \ n \text{ pari} \end{cases}$$

dove

$$K_n = \begin{cases} \int_0^n \prod_{i=0}^{n} (t - i) \, dt \ , \ n \text{ dispari e } \tilde{I} \text{ chiusa} \\ \int_{-1}^{n+1} \prod_{i=0}^{n} (t - i) \, dt \ , \ n \text{ dispari e } \tilde{I} \text{ aperta} \\ \int_0^n t \prod_{i=0}^{n} (t - i) \, dt \ , \ n \text{ pari e } \tilde{I} \text{ chiusa} \\ \int_{-1}^{n+1} t \prod_{i=0}^{n} (t - i) \, dt \ , \ n \text{ pari e } \tilde{I} \text{ aperta} \end{cases} .$$

Dimostrazione. Omettiamo la dimostrazione. ∎

(6.3.2) Osservazione *Nel teorema* (6.3.1)*, la* (*a*) *è una conseguenza della* (*b*)*.*

(6.3.3) Esempio (Formula dei rettangoli semplice) *Corrisponde alla scelta di $n = 0$ nelle formule di Newton-Cotes aperte. Avremo due sotto-intervalli di ampiezza $h = \frac{b-a}{2}$ ed un nodo x_0 corrispondente al centro dell'intervallo.*

$$\tilde{I}(f) = (b - a)f(x_0) = 2hf(x_0).$$

Grado di precisione *Sicuramente $m \geq 0$. Siccome x_0 è il centro dell'intervallo $m \geq 1$ in quanto tutte le rette vengono interpolate esattamente.*

Sia $\bar{f}(x) = (x - x_0)^2$. Il suo integrale approssimato risulta nullo mentre l'integrale esatto è banalmente strettamente positivo perciò $m = 1$.

Errore di integrazione

$$K_0 = \int_{-1}^{1} t^2 \, dt = \frac{2}{3}$$

perciò abbiamo

$$\varepsilon(f) = \frac{2}{3} \frac{f''(\xi)}{2} h^3 = \frac{1}{3} h^3 f''(\xi).$$

Scritto da Mattia Garatti

(6.3.4) Esempio (Formula dei trapezi semplice) *Corrisponde alla scelta di $n = 1$ nelle formule di Newton-Cotes chiuse. Avremo $h = b - a$ e due nodi x_0, x_1 corrispondenti agli estremi dell'intervallo.*

$$\tilde{I}(f) = \frac{(f(a) + f(b)) \cdot h}{2} = \frac{h}{2}(f(x_0) + f(x_1)).$$

Grado di precisione *Sicuramente $m \geq 1$, infatti le rette vengono integrate esattamente. Sia $\bar{f}(x) = (x - a)(x - b)$. Essa ha integrale esatto strettamente positivo ed integrale approssimato nullo allora $m = 1$.*

Errore di integrazione

$$K_1 = \int_0^1 t(t-1)\, dt = -\frac{1}{6}$$

perciò abbiamo

$$\varepsilon(f) = -\frac{1}{12}h^3 f''(\xi).$$

(6.3.5) Esempio (Formula di Simpson semplice) *Corrisponde alla scelta di $n = 2$ nelle formule di Newton-Cotes chiuse. Avremo $h = \frac{b-a}{2}$ e tre nodi x_0, x_1, x_2 corrispondenti agli estremi dell'intervallo ed al centro.*

Sappiamo dal teorema (6.2.1) *che $m \geq 2$ allora per ogni $p(x) \in \mathbb{P}_2$ l'integrazione risulterà esatta. Consideriamo tre polinomi di $\mathbb{P}_2$*

$$f_1(x) = 1\ ,\ f_2(x) = x - x_1\ ,\ f_3(x) = (x - x_1)^2$$

questi dovranno essere integrati esattamente, cioè

$$\begin{cases} I(f_1) = \tilde{I}(f_1) \\ I(f_2) = \tilde{I}(f_2) \\ I(f_3) = \tilde{I}(f_3) \end{cases} . \tag{6.3.6}$$

Analizziamo i tre polinomi:

- *f_1: l'integrale esatto è banalmente $I(f_1) = 2h$ mentre l'integrale approssimato è $\tilde{I}(f_1) = \omega_0 + \omega_1 + \omega_2$;*
- *f_2: l'integrale esatto è banalmente $I(f_2) = 0$ mentre l'integrale approssimato è $\tilde{I}(f_2) = -h\omega_0 + h\omega_2$;*
- *f_3: l'integrale esatto è $I(f_3) = \frac{2}{3}h^3$ mentre l'integrale approssimato è $\tilde{I}(f_3) = h^2\omega_0 + h^2\omega_2$.*

Svolgendo i conti del sistema (6.3.6) *otteniamo quindi*

$$\begin{cases} \omega_0 = \frac{h}{3} \\ \omega_1 = \frac{4}{3}h \\ \omega_2 = \frac{h}{3} \end{cases}$$

da cui

$$\tilde{I}(f) = \frac{h}{3}\left[f(x_0) + 4f(x_1) + f(x_2)\right].$$

Grado di precisione *Osserviamo senza dimostrazione che* $m = 3$.

Errore di integrazione

$$K_2 = \int_0^2 t^2(t-1)(t-2)\,dt = -\frac{4}{15}$$

perciò abbiamo

$$\varepsilon(f) = -\frac{1}{90}h^5 f^{(4)}(\xi).$$

(6.3.7) Osservazione *Apparentemente sembrerebbe che l'errore di integrazione segua una potenza di* h. *Tuttavia a questo livello non possiamo cambiare* h *senza cambiare l'intervallo. È possibile un'estensione, le cosiddette* ***formule composite****.*

L'idea è la seguente: suddividiamo preliminarmente l'intervallo $[a, b]$ *in sotto-intervalli e successivamente utilizziamo su ogni sotto-intervallo la corrispondente formula semplice.*

In questo modo è possibile far tendere h *a* 0 *mantenendo fisso l'intervallo. Inoltre l'errore di integrazione si accumula perché sto usando la formula semplice tante volte ed il numero di suddivisioni preliminari è inversamente proporzionale all'ampiezza* h.[2]

(6.3.8) Osservazione *Non vengono utilizzate formule di Newton-Cotes con* n *grande, infatti siccome stiamo utilizzando un polinomio interpolante, per valori alti di* n *il valore assoluto dei pesi diventa fuori controllo, iniziano anche a comparire pesi negativi, ed incappiamo nell'errore di cancellazione. Siamo quindi lontani dall'avere un risultato di convergenza all'aumentare di* n.

Collegamento con le esercitazioni

- Nell'Esercizio 15.2 e nell'Homework 15.4 implementiamo la formule dei rettangoli, dei trapezi e di Simpson, semplici o composite.
- La stima dell'errore nel Teorema (6.3.1) coinvolge derivate di ordine superiore della funzione integranda. Tale stima può essere estesa al caso delle formule composite, sommando la stima dell'errore su tutti i sotto-intervalli. L'Homework 15.1 chiede di valutare l'errore di integrazione per alcune particolari funzioni integrande che non rispettano le ipotesi di regolarità riportate nel Teorema (6.3.1).
- L'Homework 15.2 mostra che la regola dei trapezi composita è esatta non solo per tutte le funzioni affini, ma anche per tutte le funzioni che godono di una particolare proprietà di simmetria.

[2] In parole povere, nella versione composita l'errore ha una potenza di h in meno rispetto alla corrispondente semplice. Ad esempio per la formula di Simpson composita vale

$$\varepsilon(f) = c \cdot h^4 f^{(4)}(\xi).$$

6.4 Formule di Gauss-Legendre

Se scegliamo i nodi e vogliamo il massimo grado di precisione possibile siamo obbligati a utilizzare una formula interpolatoria come visto nel teorema (6.2.1).

Vediamo ora come è possibile effettuare un'integrazione approssimata non scegliendo preventivamente i nodi, ma assegnandoli in modo da ottenere il più alto grado di precisione possibile.

(6.4.1) Teorema *Sia una formula di quadratura generica*[3]

$$\tilde{I}(f) = \sum_{i=0}^{n} f(x_i) w_i.$$

Allora $m \leq 2n + 1$.

Dimostrazione. Sia $\omega(x) = (x - x_0) \dots (x - x_n) \in \mathbb{P}_{n+1} \setminus \mathbb{P}_n$, evidentemente l'integrale approssimato di questa funzione è zero, mentre non possiamo dire nulla dell'integrale esatto.

Poniamo dunque $f(x) := [\omega(x)]^2$. L'integrale approssimato continuerà ad essere nullo mentre l'integrale esatto sarà sicuramente strettamente positivo. Siccome $f(x) \in \mathbb{P}_{2n+2} \setminus \mathbb{P}_{2n+1}$, sicuramente $m < 2n + 2$ ovvero $m \leq 2n + 1$, cioè la tesi. ∎

Il seguente teorema definisce le formule di Gauss-Legendre.

(6.4.2) Teorema *Siano una formula di quadratura generica*

$$\tilde{I}(f) = \sum_{i=0}^{n} f(x_i) w_i$$

e l'intervallo $]-1, 1[$. *Sono fatti equivalenti:*

(*a*) $\tilde{I}(f)$ *ha grado di precisione* $m = 2n + 1$*;*

(*b*) $\tilde{I}(f)$ *è interpolatoria e i nodi sono gli zeri del polinomio di Legendre* $P_{n+1}(x) \in \mathbb{P}_{n+1}$.

Dimostrazione.

(*a*) $\Rightarrow$ (*b*) Siccome $2n + 1 > n + 1$ allora $m > n + 1$ e quindi $\tilde{I}(f)$ è interpolatoria per il teorema (6.2.1).

Sia $p(x) \in \mathbb{P}_n$ ed $f(x) := p(x)\omega(x) \in \mathbb{P}_{2n+1}$ con $\omega(x) = (x - x_0) \dots (x - x_n)$. Risulta che

$$\tilde{I}(f) = \sum_{i=0}^{n} p(x_i)\omega(x_i) w_i = 0$$

e

$$I(f) = \int_{-1}^{1} \omega(x) p(x)\, dx = (\omega | p)_{L^2(-1,1)}.$$

[3] Lasciamo quindi liberi sia i nodi che i pesi.

Dall'arbitrarietà di $p(x)$ e dal fatto che $f(x)$ viene integrata esattamente otteniamo

$$(\omega|p)_{L^2(-1,1)} = 0 \ , \ \forall p(x) \in \mathbb{P}_n$$

ma siccome i polinomi di Lagrange sono tutti ortogonali tra loro rispetto al prodotto scalare di Lagrange, deve essere

$$\omega(x) = \alpha P_{n+1}$$

per un certo $\alpha \in \mathbb{R}$ e perciò i due polinomi hanno gli stessi zeri.

$(b) \Rightarrow (a)$ Sia $p(x) \in \mathbb{P}_{2n+1}$. Effettuiamo la divisione euclidea tra $p(x)$ e $\omega(x) = (x - x_0) \dots (x - x_n)$ e otteniamo

$$p(x) = \omega(x) \cdot q(x) + r(x)$$

dove $q(x) \in \mathbb{P}_n \setminus \mathbb{P}_{n-1}$ e $r(x) \in \mathbb{P}_n$ perciò il resto verrà integrato esattamente e siccome i polinomi di Legendre sono definiti a meno di una costante moltiplicativa[4]

$$\tilde{I}(p) = \sum_{i=0}^{n} [\omega(x_i)q(x_i) + r(x_i)] = \tilde{I}(r)$$

e

$$I(p) = \int_{-1}^{1} [\omega(x)q(x) + r(x)]\, dx = \int_{-1}^{1} [\alpha P_{n+1} q(x) + r(x)]\, dx = \int_{-1}^{1} r(x)\, dx = I(r)$$

possiamo concludere che $\tilde{I}(p) = I(p)$ ossia che $p(x)$ viene integrato esattamente. Il grado di precisione sarà quindi, per l'arbitrarietà di $p(x)$, almeno $2n+1$. Combinando questo risultato con il teorema (6.4.1) otteniamo la (a). ■

(6.4.3) Proposizione *In una formula di Gauss-Legendre, per ogni valore di n, risulta che*

$$w_i > 0 \ , \ \forall \ i.$$

Dimostrazione. Siccome

$$w_i = \int_{-1}^{1} L_i(x)\, dx$$

ed $L_i(x) \in \mathbb{P}_n$, quindi vengono integrati esattamente, possiamo scrivere, ricordando che $L_i(x_j) = \delta_{i,j}$,

$$w_i = \sum_{j=0}^{n} L_i(x_i) w_j = \sum_{j=0}^{n} [L_i(x_j)]^2 w_j$$

perciò

$$w_i = \tilde{I}([L_i(x)]^2)$$

[4]Quindi $\omega(x) = \alpha P_{n+1}$.

ma $[L_i(x)]^2 \in \mathbb{P}_{2n}$ e quindi viene integrato esattamente. Quindi

$$w_i = \int_{-1}^{1} [L_i(x)]^2 \, dx > 0$$

da cui la tesi. ■

(6.4.4) Osservazione *Dalla proposizione* (6.4.3) *osserviamo che le formule di Gauss-Legendre possono essere usate anche per valori di n molto grandi non presentando lo stesso problema delle formule di Newton-Cotes.*

Vale anche un risultato di convergenza: per $n \to +\infty$ *risulta che* $\tilde{I}(f) \to I(f)$.

Collegamento con le esercitazioni

- I calcoli nella dimostrazione della Proposizione (6.4.3) non possono essere utilizzati direttamente per implementare numericamente le formule di Gauss-Legendre, in quanto il calcolo di w_i richiederebbe il calcolo di un integrale! Tramite una serie di lemmi, nell'Esercizio 16.1 deriveremo una formula alternativa, e più pratica, per w_i.
- L'Esercizio 16.2 e l'Homework 16.1 contengono una implementazione delle formule di Gauss-Legendre, semplici o composite.
- L'Homework 16.2 discute ulteriori proprietà di ortogonalità associate a formule di quadratura Gaussiane.
- Il Teorema (6.4.2) definisce le formule di quadratura di Gauss-Legendre e ne stabilisce il grado di precisione $m = 2n + 1$. L'Homework 16.3 propone una generalizzazione di tale risultato che permetta, fissato $\widetilde{m} = n + k$ per $0 < k \leq n + 1$, di ottenere una formula di quadratura con grado di precisione $\widetilde{m}$.
- I nodi di Gauss-Legendre non contengono gli estremi dell'intervallo di integrazione. L'Homework 16.4 introduce una altra famiglia di formule gaussiane, note come formule di Gauss-Lobatto, che raggiungono il grado di precisione massimo nel caso in cui i nodi di integrazione siano vincolati a contenere gli estremi dell'intervallo di integrazione.

Capitolo 7

Problema di Cauchy

In questo capitolo ci occupiamo di determinare numericamente la soluzione di un'equazione differenziale ordinaria noti i valori iniziali.

7.1 Alcuni richiami di analisi matematica

Cominciamo dal caso scalare. Cerchiamo $y(x) : [a, b[\to \mathbb{R}$ tale che, assegnata una certa $f : [a, b[\times \mathbb{R} \to \mathbb{R}$, si abbia che

$$y'(x) = f(x, y(x)).$$

In generale questa equazione ha infinite soluzioni.

(7.1.1) Teorema *Sia $f : [a, b[\times \mathbb{R} \to \mathbb{R}$ una funzione continua nella variabile x e tale che esiste $c > 0$ tale che per ogni $\xi \in [a, b[$, dati $y_1, y_2 \in \mathbb{R}$*

$$|f(\xi, y_1) - f(\xi, y_2)| \leq c \cdot |y_1 - y_2|$$

ovvero che f sia uniformemente lipschitziana nella variabile y. Allora il problema di Cauchy

$$\begin{cases} y'(x) = f(x, y(x)) \\ y(a) = Y_0 \end{cases}$$

ha una ed una sola soluzione che indichiamo con $Y(x)$.

Dimostrazione. Omettiamo la dimostrazione. ■

Analogamente possiamo definire il caso vettoriale.

(7.1.2) Osservazione *Il problema di Cauchy definito nel teorema* (7.1.1) *è ben posto.*

Collegamento con le esercitazioni L'Homework 18.3 propone un metodo per riportare la soluzione di una equazione del tipo $y''(x) = f(x, y(x))$ a quello di un sistema di problemi di Cauchy.

7.2 Metodo di Eulero

Innanzitutto dobbiamo dire che la soluzione del problema di Cauchy è una funzione definita in un intervallo, e non è pensabile che un calcolatore possa contenere tutte le valutazioni della funzione in quell'intervallo.

Ciò che si fa è concentrare l'attenzione su una griglia di punti[1], che si immagina abbastanza fitta: posto quindi $h > 0$, avremo

$$x_n = a + n \cdot h$$

gli infiniti nodi che si possono costruire a partire dal punto $a = x_0$. Detto poi

$$N(h) = \lfloor \frac{b-a}{h} \rfloor$$

avremo in particolare

$$x_n = a + n \cdot h \ , \ n = 0, \dots, N(h).$$

Quello che faremo sarà andare a cercare la soluzione nei nodi, ovvero

$$Y_n = Y(x_n)$$

imponendo quindi che

$$Y_n' = f(x_n, Y_n) \ , \ n = 1, \dots, N(h)^2$$

Tuttavia non possiamo scrivere Y_n' perché richiederebbe la conoscenza della funzione Y in un intorno di x_n. Approssimiamo quindi la derivata con un rapporto incrementale

$$Y_n' \approx \frac{Y(x_n + h) - Y_n}{h} = \frac{Y_{n+1} - Y_n}{h}$$

quindi

$$Y_{n+1} \approx Y_n + hf(x_n, Y_n) \ , \ n = 0, \dots, N(h)$$

e se consideriamo un'approssimazione della soluzione esatta y possiamo scrivere

$$y_{n+1} = y_n + h \cdot f(x_n, y_n) \ , \ n = 0, \dots, N(h)$$

ottenendo così il **metodo di Eulero esplicito**.

Cerchiamo di quantificare l'errore che commettiamo utilizzando questo metodo. Sviluppando con Taylor, resto di Lagrange, la soluzione esatta possiamo scrivere

$$Y_{n+1} = Y(x_n + h) = Y_n + hY_n' + \frac{h^2}{2} Y''(\xi)$$

ed ovviamente ciò richiede una regolarità di tipo C^2 sulla soluzione. A questo punto detto

$$T_{n+1} := \frac{h^2}{2} Y''(\xi)$$

[1] Detti nodi, che per noi saranno per semplicità equi-spaziati.

[2] Sarebbe da 0, ma lì è vero per come è definito il problema di Cauchy, quindi non serve imporlo.

l'**errore di troncamento**[3], otteniamo

$$Y_{n+1} = Y_n + hY'_n + T_{n+1}$$

oppure, detto $\tau_{n+1}(h, Y) := \frac{1}{h}T_{n+1}$

$$Y_{n+1} = Y_n + hY'_n + h \cdot \tau_{n+1}.$$

Analizziamo quindi l'andamento dell'errore. Innanzitutto per $n = 0$

$$Y_1 = Y_0 + hf(x_0, Y_0) + h\tau_1$$

e

$$y_1 = Y_0 + hf(x_0, Y_0)$$

quindi

$$\varepsilon_1 = |Y_1 - y_1| = h\tau_1.$$

Invece per un generico n abbiamo che l'errore ε_n sarà dato dalla somma dell'errore generato ai passi precedenti che viene propagato e di un nuovo errore generato a questo passo in modo analogo.

(7.2.1) Proposizione *Posto* $\varepsilon = \max_n |Y_n - y_n|$ *l'errore globale, esiste* $c > 0$ *tale che*

$$\varepsilon \leq c \cdot \tau(h, Y)$$

dove $\tau(h, Y) = \max_n |\tau_n|$. *In particolare quindi il metodo di Eulero esplicito ha ordine 1.*

Dimostrazione. Innanzitutto $\varepsilon_n = |Y_n - y_n|$. Inoltre

$$Y_{n+1} = Y_n + hf(x_n, Y_n) + h\tau_{n+1}$$

e

$$y_{n+1} = y_n + hf(x_n, y_n)$$

quindi

$$\varepsilon_{n+1} = |\varepsilon_n + h(f(x_n, Y_n) - f(x_n, y_n)) + h\tau_{n+1}|.$$

Ora

$$\varepsilon_{n+1} \leq \varepsilon_n + h|f(x_n, Y_n) - f(x_n, y_n)| + h\tau$$

e per la lipschitziana della funzione f, esiste $L > 0$ tale che

$$\varepsilon_{n+1} \leq (1 + hL)\varepsilon_n + h\tau.$$

Adesso considerando il generico passo n, possiamo scrivere

[3] Notiamo che è proporzionale ad h^2.

$\varepsilon_n \leq (1+hL)\varepsilon_{n-1}+h\tau \leq (1+hL)\left[(1+hL)\varepsilon_{n-2}+h\tau\right]+h\tau = (1+hL)^2\varepsilon_{n-2}+(1+(1+hL))h\tau$

e procedendo a ritroso otteniamo

$$\varepsilon_n \leq (1+hL)^n\varepsilon_0 + h\tau\cdot\sum_{i=0}^{n-1}(1+hL)^i = (1+hL)^n\varepsilon_0 + h\tau\frac{(1+hL)^n-1}{(1+hL)-1}$$

e quindi

$$\varepsilon_n \leq (1+hL)^n\varepsilon_0 + \frac{(1+hL)^n}{L}\tau.$$

Ma ora ricordando che pero ogni x, $1+x \leq e^x$ e quindi se $x \geq -1$ $(1+x)^n \leq e^{nx}$ otteniamo

$$\varepsilon_n \leq e^{hnL}\varepsilon_0 + \frac{1}{L}e^{hnL}\tau$$

ma ricordando che $hn \leq b-a$ otteniamo

$$\varepsilon_n \leq e^{L(b-a)}\varepsilon_0 + \frac{1}{L}e^{L(b-a)}\tau$$

che è un'espressione che non dipende più da n e quindi possiamo scrivere, ricordando che $\varepsilon_0 = 0$

$$\varepsilon \leq c\cdot\tau$$

dove abbiamo posto $c = \frac{1}{L}e^{L(b-a)}$. ■

Collegamento con le esercitazioni L'Esercizio 17.1 discute una implementazione del metodo di Eulero esplicito, ed una applicazione della Proposizione (7.2.1) grazie al quale, stabilito un errore massimo desiderato, si stabilisca un valore per il passo h in modo che l'errore compiuto dal metodo di Eulero sia sicuramente al di sotto di tale soglia.

7.3 Metodi espliciti ed impliciti

Consideriamo il problema

$$\int_{x_n}^{x_{n+1}} Y'(x)\,dx = \int_{x_n}^{x_{n+1}} f(x,Y(x))\,dx$$

il primo membro possiamo semplicemente riscriverlo come $Y_{n+1}-Y_n$ mentre per il secondo potremmo approssimare l'integrale con una formula di quadratura.

Metodo di Eulero esplicito Usando la formula dei rettangoli sinistri, che ha grado di precisione $m=0$, si ottiene

$$Y_{n+1}-Y_n \approx hf(x_n,Y_n)$$

ovvero il metodo di Eulero esplicito. È un metodo ad 1 stadio [4].

[4] Chiamiamo **numero di stadi** il numero di valutazioni di f al singolo passo.

Metodo di Eulero implicito Usando la formula dei rettangoli destri, che ha grado di precisione $m = 0$, si ottiene

$$Y_{n+1} - Y_n \approx hf(x_{n+1}, Y_{n+1})$$

ovvero

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1})$$

il cosiddetto metodo di Eulero implicito. Questo è un metodo di ordine 1 ed è un problema implicito risolubile mediante una delle tecniche per le equazioni non lineari viste in precedenza. È ad 1 passo.

Metodo di Crank - Nicolson o dei trapezi Usando la formula dei trapezi, che ha grado di precisione $m = 1$, si ottiene

$$y_{n+1} = y_n + \frac{h}{2}(f(x_n, y_n) + f(x_{n+1}, y_{n+1}))$$

un metodo implicito di ordine 2 ad 1 passo.

Consideriamo invece il problema

$$\int_{x_{n-1}}^{x_{n+1}} Y'(x)\, dx = \int_{x_{n-1}}^{x_{n+1}} f(x, Y(x))\, dx$$

il primo membro possiamo semplicemente riscriverlo come $Y_{n+1} - Y_n$ mentre per il secondo potremmo approssimare l'integrale con una formula di quadratura.

Metodo mid-point Usando la formula dei rettangoli, che ha grado di precisione $m = 1$, si ottiene

$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n)$$

un metodo esplicito di ordine 2 a 2 passi ed 1 stadio. Questo metodo ha purtroppo problemi di stabilità non essendo relativamente stabile.

Collegamento con le esercitazioni

- Gli Esercizi 17.2 e 17.3, insieme agli Homework 17.1 e 17.2, contengono una implementazione del metodo di Eulero Implicito.
- L'Homework 17.3 richiede di studiare una famiglia di metodi, del quale il metodo di Crank-Nicolson è un caso particolare.
- L'Homework 17.4 chiede di determinare condizioni sul passo h in modo che la soluzione di un problema modello rimanga limitata.

7.4 Metodi Predictor-Corrector

Consideriamo il metodo dei trapezi

$$y_{n+1} = y_n + \frac{h}{2}(f(x_n, y_n) + f(x_{n+1}, y_{n+1}))$$

e poniamo

$$f_n := f(x_n, y_n)$$
$$\Phi(y_{n+1}) := y_n + \frac{h}{2}(f_n + f(x_{n+1}, y_{n+1}))$$

otteniamo la seguente scrittura

$$y_{n+1} = \Phi(y_{n+1})$$

corrispondente ad un problema di punto fisso risolvibile mediante un processo di iterazione funzionale. Chiameremo la soluzione $y_{n+1}^{(c)}$ **corrector** e avremo

$$y_{n+1}^{(c)} = y_n + \frac{h}{2}\left(f_n + f(x_{n+1}, y_{n+1}^{(c)})\right).$$

Il processo iterativo può essere così descritto: per n fissato[5] prendiamo un valore di innesco $y_{n+1}^{(0)} =: y_{n+1}^{(p)}$

per $k = 0, 1, \ldots$:
$y_{n+1}^{(k+1)} = y_n + \frac{h}{2}\left(f_n + f(x_{n+1}, y_{n+1}^{(k)})\right)$;

Per la scelta del valore di innesco utilizziamo un metodo esplicito che chiameremo **predictor**. In questo caso usiamo il metodo di Eulero esplicito; avremo quindi

$$y_{n+1}^{(p)} = y_n + hf_n.$$

(7.4.1) Proposizione *Il processo iterativo che porta alla soluzione approssimata del problema di punto fisso Φ converge se h è sufficientemente piccolo.*

Dimostrazione. Chiamiamo innanzitutto L la costante di lipschitz di f. Possiamo scrivere

$$\begin{aligned}
|y_{n+1}^{(c)} - y_{n+1}^{(k+1)}| &= \left| y_n + \frac{h}{2}\left(f_n + f(x_{n+1}, y_{n+1}^{(c)})\right) - y_n - \frac{h}{2}\left(f_n + f(x_{n+1}, y_{n+1}^{(k)})\right)\right| = \\
&= \left|\frac{h}{2}\left(f(x_{n+1}, y_{n+1}^{(c)}) - f(x_{n+1}, y_{n+1}^{(k)})\right)\right| \leq \\
&\leq \frac{hL}{2}|y_{n+1}^{(c)} - y_{n+1}^{(k)}|
\end{aligned}$$

quindi per la proposizione (3.6.1) del capitolo 3 avremo convergenza se

$$\frac{hL}{2} < 1.$$

Ricordando ora che L è una costante, otteniamo la tesi. ∎

[5] Facendo variare n troviamo i nuovi punti in cui passa la soluzione. Facendo variare k, miglioriamo l'approssimazione.

(7.4.2) Proposizione *Eseguendo una iterazione del processo iterativo Φ, si ottiene un metodo di ordine 2, cioè l'errore di troncamento è*

$$T_{n+1}(h, Y) = o(h^3),$$

Dimostrazione. Per la disuguaglianza triangolare del valore assoluto,

$$|Y_{n+1} - y_{n+1}^{(k)}| \leq |Y_{n+1} - y_{n+1}^{(c)}| + |y_{n+1}^{(c)} - y_{n+1}^{(k)}|$$

ma il primo termine è l'errore di troncamento del metodo dei trapezi, ovvero

$$|Y_{n+1} - y_{n+1}^{(c)}| = T_{n+1}^{(c)}(h, Y) = o(h^3)$$

quindi, ripetendo induttivamente i passaggi della dimostrazione della proposizione (7.4.1) otteniamo

$$\begin{aligned} |Y_{n+1} - y_{n+1}^{(k)}| &\leq o(h^3) + \left(\frac{hL}{2}\right)^k |y_{n+1}^{(c)} - y_{n+1}^{(p)}| \leq \\ &\leq o(h^3) + \left(\frac{hL}{2}\right)^k |y_{n+1}^{(c)} - Y_{n+1}| + \left(\frac{hL}{2}\right)^k |Y_{n+1} - y_{n+1}^{(p)}| \end{aligned}$$

dove nell'ultimo passaggio abbiamo utilizzato nuovamente la disuguaglianza triangolare del valore assoluto. Ora quindi, analogamente a sopra otteniamo

$$|Y_{n+1} - y_{n+1}^{(k)}| \leq o(h^3) + \left(\frac{hL}{2}\right)^k o(h^3) + \left(\frac{hL}{2}\right)^k T_{n+1}^{(p)}(h, Y)$$

ma per il predictor abbiamo utilizzato il metodo di Eulero esplicito, perciò

$$|Y_{n+1} - y_{n+1}^{(k)}| \leq o(h^3) + \left(\frac{hL}{2}\right)^k o(h^3) + \left(\frac{hL}{2}\right)^k o(h^2)$$

siccome da questa scrittura è chiaro che ad ogni iterata l'esponente di h del terzo termine aumenta di 1, basta una sola iterazione per ottenere ordine di convergenza 2. ■

(7.4.3) Osservazione *Quello che abbiamo descritto, in generale, è noto come metodo* ***P.E.C.E.*** *(Predictor - Evaluator - Corrector - Evaluator)*[6]*. In particolare, nel nostro caso siamo partiti dal metodo dei trapezi, è noto come metodo di Heun.*

Possiamo riassumere l'algoritmo in questo modo

al passo n:
ho y_n ed il corrispondente f_n;
calcolo $y_{n+1}^{(p)}$;
calcolo $f(x_{n+1}, y_{n+1}^{(p)})$;
calcolo mediante il processo iterativo $y_{n+1} \approx y_{n+1}^{(c)}$;
calcolo $f(x_{n+1}, y_{n+1})$;

[6]Noi in questo caso abbiamo fatto una sola iterazione del processo di iterazione funzionale, ammettendo di fare k iterazione quello che si ottiene è un metodo di questo tipo $P.(E.C.)^k E$.

Il metodo di Heun rientra quindi nella famiglia dei metodi predictor-corrector: si tratta inoltre di un metodo di ordine 2 della famiglia di Runge-Kutta, **RK2**

$$\begin{cases} y_{n+1}^{(p)} = y_n + f_n \\ y_{n+1} = y_n + \frac{h}{2}\left(f_n + f(x_{n+1}, y_{n+1}^{(p)})\right) \end{cases}$$

è un metodo a 2 stadi ed 1 passo.

7.5 Metodi Runge - Kutta

I metodi predictor-corrector che abbiamo analizzato nella sezione precedente, rientrano nella più generale famiglia dei metodi di Runge - Kutta.

(7.5.1) Osservazione *Il metodo di Eulero corrisponde a **RK1**.*

Analizziamo ora il metodo di Runge - Kutta di ordine 4, **RK4**. Consideriamo il problema[7]

$$\int_{x_n}^{x_{n+1}} Y'(x)\,dx = \int_{x_n}^{x_{n+1}} f(x, Y(x))\,dx$$

il primo membro possiamo semplicemente riscriverlo come $Y_{n+1} - Y_n$ mentre per il secondo potremmo approssimare l'integrale con la formula di Simpson, ottenendo

$$Y_{n+1} = Y_n + \frac{h}{6}\left(Y_n' + 4Y'(x_n + \frac{h}{2}) + Y_{n+1}'\right)$$

dove in questo caso h è l'ampiezza dell'intervallo $[a, b[$. Utilizzando ora altri metodi numerici cerchiamo della approssimazioni per le due quantità che non ci sono note, ovvero $Y'(x_n + \frac{h}{2})$ e Y_{n+1}'.[8]

Consideriamo

$$k_1 := f_n$$

utilizziamo quindi, con mezzo passo, la formula dei rettangoli sinistri, ovvero il metodo di Eulero esplicito, per costruire una prima approssimazione di f in $x_n + \frac{h}{2}$,

$$k_2 := f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1)$$

ora, utilizzando k_2 come approssimazione del valore di f in $x_n + \frac{h}{2}$, utilizziamo la formula dei rettangoli destri, ovvero il metodo di Eulero implicito, per costruire una seconda approssimazione di f in $x_n + \frac{h}{2}$

$$k_3 := f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2)$$

infine, mediante la formula dei rettangoli costruiamo un'approssimazione di Y_{n+1}'

$$k_4 := f(x_{n+1}, y_n + hk_3).$$

[7] Integriamo tra x_n ed x_{n+1} perché stiamo costruendo un metodo ad 1 passo.

[8] Se avessimo i valori esatti il metodo sarebbe di ordine 4, siccome per il metodo di Simpson $m = 3$. Dobbiamo quindi approssimare in modo da non perdere questa proprietà.

Risulta così definito il metodo di Runge - Kutta di ordine 4

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

dove abbiamo usato la media aritmetica tra k_2 e k_3 come approssimazione di $Y'(x_n + \frac{h}{2})$.

(7.5.2) Proposizione *Fino all'ordine* 4 *compreso, i metodi di Runge-Kutta hanno ordine uguale al numero di stadi. A partire dal metodo di Runge-Kutta-Fehlberg quattro cinque,* ***RKF45****, il numero di stadi è superiore. In particolare* ***RKF45*** *ha ordine* 5 *e* 6 *stadi.*

Collegamento con le esercitazioni L'Esercizio 18.1 e l'Homework 18.1 mostrano una applicazione di metodi RK.

7.6 Metodi Multi-Step

Analizziamo ora la famiglia dei metodi a più passi, o multi-step.

(7.6.1) Definizione *Dato $p \geq 0$, un metodo lineare si dice a $p+1$ passi, se possiamo scriverlo nella forma*

$$y_{n+1} = \sum_{i=0}^{p} \alpha_i y_{n-i} + h \sum_{i=-1}^{p} \beta_i f(x_{n-i}, y_{n-i}).$$

(7.6.2) Osservazione *Come già preannunciato, il metodo mid-point è a* 2 *passi, infatti corrisponde alla scelta di $p = 1$, $\alpha_0 = 0$, $\alpha_1 = 1$, $\beta_{-1} = 0$, $\beta_0 = 2$ e $\beta_1 = 0$ nella definizione* (7.6.1).

(7.6.3) Proposizione *Un metodo lineare a $p+1$ passi è implicito se e solo se $\beta_{-1} \neq 0$.*

Dimostrazione. Omettiamo la dimostrazione. ■

(7.6.4) Osservazione *Il metodo di Heun ed il metodo* ***RK4*** *non sono metodi lineari, perciò banalmente non possono rientrare in questa categorizzazione. Al contrario i metodi di Eulero ed il metodo dei trapezi sono metodi lineari ad* 1 *passo:*

- *con la scelta di $p = 0$, $\alpha_0 = 1$, $\beta_{-1} = 0$, $\beta_0 = 1$ otteniamo il metodo di Eulero esplicito;*
- *con la scelta di $p = 0$, $\alpha_0 = 1$, $\beta_{-1} = \frac{1}{2}$, $\beta_0 = \frac{1}{2}$ otteniamo il metodo dei trapezi;*
- *con la scelta di $p = 0$, $\alpha_0 = 1$, $\beta_{-1} = 1$, $\beta_0 = 0$ otteniamo il metodo di Eulero implicito.*

(7.6.5) Osservazione *Anche se nella definizione* (7.6.1)*, compaiono più valutazioni della funzione f, i metodi così definiti sono tutti ad* 1 *stadio, infatti le valutazioni che compaiono sono fatte tutte, eccetto una, a iterazioni di n precedenti.*

(7.6.6) Osservazione *I metodi multi-step sembrano a prima vista molto comodi, per via di una formula algebrica chiara che li definisce. Tuttavia presentano almeno due complicazioni:*

- *per quanto riguarda il bootstrap, l'avvio del metodo, a rigore avremmo bisogno delle prime $p+1$ valutazioni della funzione, ovvero servono $y_0, \dots, y_p$ come valori di innesco, che però in generale non possediamo. Per ovviare a questo problema possiamo utilizzare un metodo ad un passo, di ordine comparabile al metodo multi-step che stiamo implementando per produrre le valutazioni necessarie;*
- *in generale, quando si utilizza un metodo numerico, il passo h non è fisso, ma bisogna adattarlo ad ogni valutazione in base alla regolarità di f. Cambiare h in un metodo multi-step può essere complicato: se voglio farlo devo fermare il calcolo e ripartire dall'inizio.*

Condizioni algebriche

Analizziamo in generale la famiglia dei metodi multi-step per andare a determinare delle condizioni per cui si abbia la convergenza.

(7.6.7) Definizione *Definiamo errore di troncamento per un metodo multi-step lineare*

$$T_{n+1}(h, Y) := Y_{n+1} - \sum_{i=0}^{p} \alpha_i Y_{n-j} + h \sum_{i=-1}^{p} \beta_i Y'(x_{n-j}).$$

Indichiamo poi con $\tau_{n+1}(h, Y) := \frac{1}{h} T_{n+1}(h, Y)$.

(7.6.8) Definizione *Diciamo che un metodo multi-step è* ***convergente****, se*

$$\max_{n \le N(h)} |Y_n - y_n| \to 0$$

per $h \to 0$.

(7.6.9) Definizione *Diciamo che un metodo multi-step è* ***consistente*** *, se*

$$\tau_{n+1}(h, Y) \to 0$$

per $h \to 0$.

(7.6.10) Osservazione *La consistenza è condizione necessaria alla convergenza.*

(7.6.11) Definizione *Diciamo che un metodo multi-step ha ordine q se*

$$\tau_{n+1}(h, Y) = o(h^q)$$

(7.6.12) Teorema *Un metodo multi-step lineare è consistente se e solo se*

$$\sum_{j=0}^{p} \alpha_j = 1$$

e

$$\sum_{j=-1}^{p} \beta_j - \sum_{j=0}^{p} j\alpha_j = 1.$$

Dimostrazione. La tesi è equivalente a supporre che $T_{n+1}(h, Y) \equiv 0$ per ogni $Y \in \mathbb{P}_1$. Prendiamo come base

$$B = (1, x - x_n)\,.$$

Per quanto riguarda $Y(x) = 1$, avremo che per ogni j vale che $Y_{n-j} = 1$ e perciò

$$T_{n+1}(h, 1) = 0$$

se e solo se, per la definizione (7.6.7)

$$1 - \sum_{j=0}^{p} \alpha_j \cdot 1 = 0$$

ovvero

$$\sum_{j=0}^{p} \alpha_j = 1.$$

Se invece consideriamo $Y(x) = x - x_n$, risulta che

$$Y_{n-j} = -hj$$

e

$$Y'(x_{n-j}) = 1$$

perciò sempre dalla definizione (7.6.7) abbiamo che

$$h - h\sum_{j=0}^{p} \alpha_j \cdot j - h \sum_{j=-1}^{p} \beta_j \cdot 1 = 0$$

ovvero

$$\sum_{j=-1}^{p} \beta_j - \sum_{j=0}^{p} j\alpha_j = 1$$

ovvero la tesi. ■

(7.6.13) Teorema *Un metodo multi-step lineare è ha ordine q se e solo se è consistente e*

$$\sum_{j=0}^{p} j^k \alpha_j - k \sum_{j=-1}^{p} j^{k-1}\beta_j = (-1)^k\ ,\ k = 0, \ldots, q.$$

Dimostrazione. Omettiamo la dimostrazione. ■

(7.6.14) Definizione *Siano $\bar{y_0}, \ldots, \bar{y_p}$ dei valori di innesco di un metodo multi-step che produce una soluzione numerica y_n e $\bar{z_0}, \ldots, \bar{z_p}$ degli altri valori di innesco per lo stesso metodo che quindi produrrà z_n. Diciamo che il metodo multi-step è* ***(zero) stabile*** *se esiste $c \in \mathbb{R}$ tale che*

$$\max_{n\leq N(h)} |y_n - z_n| \leq c \cdot \max_{i\leq p} |\bar{y}_i - \bar{z}_i|.$$

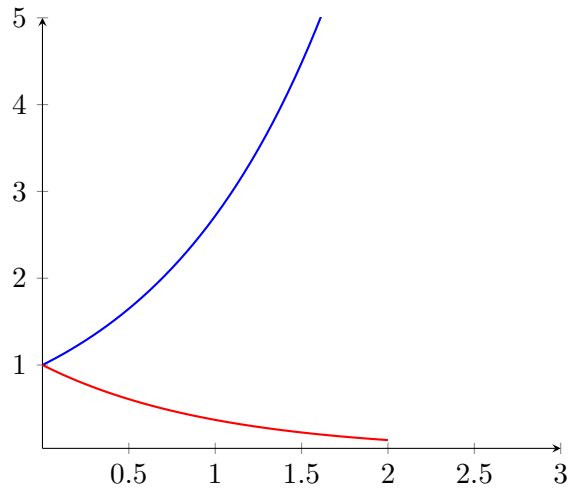
Collegamento con le esercitazioni L'Esercizio 18.2 e l'Homework 18.2 mostrano una applicazione di metodi multi-step.

Un esempio notevole

Consideriamo un problema prototipo. Sia $\lambda \in \mathbb{C}$ e

$$\begin{cases} y'(x) = \lambda y(x) \ , x \in [0, +\infty) \\ y(0) = 1 \end{cases} .$$

Questo problema ha soluzione esatta determinabile analiticamente, $Y(x) = e^{\lambda x}$, il cui andamento dipende da λ come mostrato in figura, in blu un caso in cui $\mathfrak{Re}(\lambda) > 0$ ed in rosso un caso in cui $\mathfrak{Re}(\lambda) < 0$.



Particolarmente interessanti sono poi i valori di λ tali che $\mathfrak{Re}(\lambda) << 0$: in questo caso si parla di problemi **stiff**[9].

Fissiamo dunque $h > 0$ e prendiamo i nodi $x_n = nh$. Qui la soluzione esatta sarà $Y_n = e^{\lambda x_n} = e^{\lambda nh}$. Poniamo per comodità poi $s := h\lambda$ e otteniamo

$$Y_n = e^{ns}.$$

Cerchiamo invece la soluzione numerica nei nodi ovvero y_n

$$y_{n+1} = \sum_{i=0}^{p} \alpha_i y_{n-i} + h\lambda \sum_{i=-1}^{p} \beta_i y_{n-i}.$$

[9] In generale un problema è stiff se la derivata parziale rispetto a y è molto negativa.

La precedente scrittura esprime un legame lineare tra $p+2$ valori consecutivi $y_{n+1}, \dots, y_{n-p}$: è quella che viene chiamata **equazione alle differenze**[10]. Cerchiamo dunque $p+1$ soluzioni della forma

$$y_n = r^n,$$

perciò abbiamo

$$r^{n+1} - \sum_{i=0}^{p} \alpha_i r^{n-i} - h\lambda \sum_{i=-1}^{p} \beta_i r^{n-i} = 0$$

da cui dividendo per il fattore comune ai tre addendi r^{n-p}

$$r^{p+1} - \sum_{i=0}^{p} \alpha_i r^{p-i} - h\lambda \sum_{i=-1}^{p} \beta_i r^{p-i} = 0.$$

Poniamo ora

$$\rho(r) := r^{p+1} - \sum_{i=0}^{p} \alpha_i r^{p-i}$$

e

$$\sigma(r) := \sum_{i=-1}^{p} \beta_i r^{p-i}.$$

Evidentemente $\rho(r), \sigma(r) \in \mathbb{P}_{p+1}$ e detto $\rho_{h\lambda}(r) := \rho(r) - h\lambda\sigma(r)$ anche quest'ultimo sta in $\mathbb{P}_{p+1}$. Riassumendo le notazioni abbiamo che

$$\rho_{h\lambda}(r) = 0$$

ovvero

$$\rho(r) - h\lambda\sigma(r) = 0.$$

In definitiva questi passaggi ci portano a dire che ho una soluzione dell'equazione alle differenze del tipo cercato ogni qual volta r è una radice di $\rho_{h\lambda}(r)$. Supponiamo, per semplicità, che tutte le radici siano semplici. Chiamiamo le radici che stiamo cercando

$$r_i(h\lambda)\ ,\ i = 0, \dots, p$$

e la generica soluzione numerica sarà combinazione lineare[11] di queste, ovvero

$$y_n = \sum_{j=0}^{p} \eta_j \left(r_j(h\lambda)\right)^n .$$

[10] Si tratta di un'equazione che lega tra loro i termini di una successione. Trovare una soluzione significa trovare una successione che verifica tale legame. L'insieme delle soluzioni è uno spazio vettoriale di dimensione $p+1$.

[11] Si potrebbe inoltre dimostrare che, per $h \to 0$, il vettore η tende ad un vettore che ha prima componente 1 e restanti 0.

Noi però conosciamo i valori di innesco del metodo, quindi dobbiamo imporre che $y_i = \bar{y}_i$ per $i = 0, \dots, p$. Ora però se $h \to 0$, le radici di $\rho_{h\lambda}(r)$ tendono alle radici di $\rho(r)$ che chiamiamo $r_0, \dots, r_p$.

Ora, se stiamo usando un metodo consistente, una radice di $\rho(r)$ la possediamo, infatti

$$\rho(1) = 1 - \sum_{i=0}^{p} \alpha_i = 0$$

dalla prima condizione algebrica di consistenza. Numeriamo quindi le radici in modo che $r_0 = 1$: questa viene detta radice principale.

Riportiamo di seguito una proposizione utile per il seguito

(7.6.15) Proposizione *Per metodi multi-step consistenti:*

1. *se* $\rho(1) = 1$ *allora* $r_0 = 1$*;*

2. $r_0(s) = 1 + s + o(s)$.

Dimostrazione. Iniziamo dalla seconda implicazione: ci basta dimostrare che la derivata di $r_0(s)$ rispetto ad s è 1 (quello scritto nella tesi è uno sviluppo di Taylor). Siccome $r_0(s)$ è una radice risulta che

$$0 = \rho(r_0(s)) - s\sigma(r_0(s))$$

relazione che possiamo derivare rispetto ad s ottenendo

$$0 = \rho'(r_0(s))\frac{dr_0(s)}{ds} - \sigma(r_0(s)) - s\sigma'(r_0(s))\frac{dr_0(s)}{ds}$$

da cui ponendo $s = 0$

$$0 = \rho'(1)\frac{dr_0(s)}{ds}|_{s=0} - \sigma(1)$$

quindi

$$\frac{dr_0(s)}{ds}|_{s=0} = \frac{\sigma(1)}{\rho'(1)}.$$

Ora, $\sigma(1) = \sum_{i=-1}^{p} \beta_i$ e invece

$$\begin{aligned}\rho'(1) &= p+1-\sum_{i=0}^{p-1}(p-i)\alpha_i = \\ &= p-p\sum_{i=0}^{p-1}\alpha_i - p\alpha_p + p\alpha_p + 1 + \sum_{i=0}^{p-1} i\alpha_i = \\ &= p-p\sum_{i=0}^{p}\alpha_i + p\alpha_p + 1 + \sum_{i=0}^{p-1} i\alpha_i = \\ &= p-p+p\alpha_p+1+\sum_{i=0}^{p-1} i\alpha_i = \\ &= \sum_{i=0}^{p-1} i\alpha_i + p\alpha_p + 1 = \\ &= \sum_{i=0}^{p} i\alpha_i + 1 = \sum_{i=-1}^{p}\beta_i\end{aligned}$$

da cui la tesi. La prima implicazione si può ottenere banalmente dalla seconda. ■

Ora, continuando la trattazione,

$$(r_0(s))^n \approx (1+s)^n = \left[(1+s)^{\frac{1}{s}}\right]^{ns} \approx e^{ns} = Y_n.$$

Perciò la radice principale approssima la soluzione esatta. Questo risultato non vale per le altre radici che vengono chiamate infatti **soluzioni parassite**: se

$$|r_j(s)| > |r_0(s)| \ , \ j > 0$$

incorriamo in problemi perché prima o poi prende il sopravvento la soluzione parassita e quindi non otteniamo più un risultato numerico valido.

(7.6.16) Definizione *Diciamo che un metodo multi-step verifica la* ***root-condition*** *se valgono i seguenti fatti:*

(*a*) $|r_j| \leq 1$ *per* $j = 0, \ldots, p$*;*

(*b*) *se* $|r_j| = 1$ *allora la radice è semplice.*

(7.6.17) Definizione *Diciamo che un metodo multi-step verifica la* ***strong root condition*** *se*

$$|r_j| < 1 \ \ j = 1, \ldots, p.$$

(7.6.18) Teorema *Se un metodo multi-step è consistente i seguenti fatti sono equivalenti:*

(*a*) *il metodo converge;*

(*b*) *il metodo verifica la root condition;*

(c) *il metodo è (zero) stabile.*

Dimostrazione. Omettiamo la dimostrazione. ■

(7.6.19) Definizione *Chiamiamo* ***regione di relativa stabilità*** *l'insieme*

$$R.R.S. := \{s = h\lambda : |r_j(s)| \leq |r_0(s)| \ , \ j = 1, \dots, p\} .$$

In particolare un metodo multi-step si dice ***relativamente stabile*** *se la sua regione di relativa stabilità contiene l'origine.*

(7.6.20) Osservazione *In generale la soluzione numerica è combinazione lineare della soluzione principale e delle soluzioni parassite*

$$\eta_0 \left(r_0(s)\right)^n + \sum_{j=1}^{p} \eta_j \left(r_j(s)\right)^n = \left(r_0(s)\right)^n \left[\eta_0 + \sum_{j=1}^{p} \eta_j \left(\frac{r_j(s)}{r_0(s)}\right)^n\right]$$

ma fuori dalla regione di relativa stabilità

$$\left(\frac{r_j(s)}{r_0(s)}\right)^n \to +\infty$$

cioè la soluzione parassita prende il sopravvento.

(7.6.21) Definizione *Se* $\mathfrak{Re}(\lambda) < 0$ *chiamiamo* ***regione di assoluta stabilità*** *l'insieme*

$$R.A.S. := \{h\lambda : |r_j(h\lambda)| < 1 \ , \ j = 1, \dots, p\}$$

(7.6.22) Osservazione *La definizione* (7.6.21) *è ben posta quando* $\mathfrak{Re}(\lambda) < 0$*: in questi casi infatti per* $x \to +\infty$ *la soluzione esatta tende a* 0*. Ciò che ci chiediamo è quindi se anche la soluzione numerica ha lo stesso comportamento qualitativo: cioò avviene se tutte le soluzioni parassite tendono a zero, ovvero ciò che abbiamo scritto.*

(7.6.23) Esempio *Consideriamo il metodo mid-point*

$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n).$$

Abbiamo

$$\rho(r) = r^2 - 1$$

e

$$\sigma(r) = 2r$$

Perciò $r_0 = 1$ *ed* $r_1 = -1$*, da cui possiamo dedurre che verifica la root condition ma non la strong root condition. Invece*

$$\rho_s(r) = r^2 - 2sr - 1$$

e quindi $r_{0,1}(s) = s \pm \sqrt{s^2+1}$*, ovvero sviluppando con Taylor, e limitandoci a soluzioni reali,*

$$r_0(h\lambda) = 1 + h\lambda + \frac{(h\lambda)^2}{2} + \dots$$

$$r_1(h\lambda) = -1 + h\lambda - \frac{(h\lambda)^2}{2} + \dots$$

vediamo subito quindi che se $\lambda > 0$*, tutto procede correttamente perché restiamo dentro la regione di relativa stabilità, invece se* $\lambda < 0$ *siamo fuori dalla regione di relativa stabilità. Siccome la relativa stabilità non dipende da* λ *possiamo quindi affermare che il metodo mid-point non è relativamente stabile.*

Riportiamo ora uno schema delle implicazioni valido per metodi multi-step consistenti:

$$\begin{array}{ccccc} \text{Convergenza} & \Leftrightarrow & \text{Root Condition} & \Leftrightarrow & \text{(zero) stabilità} \\ & & \Uparrow & & \Uparrow \\ & & \text{Strong Root Condition} & \Rightarrow & \text{Relativa Stabilità} \end{array}$$

Metodi di Adams

Un caso particolare di metodi multi-step è rappresentato dalla famiglia[12] di metodi di Adams. L'idea per costruire uno di questi metodi è la seguente: si considera il problema

$$Y_{n+1} = Y_n + \int_{x_n}^{x_{n+1}} f(x, Y(x))\, dx$$

e si utilizza una formula di quadratura interpolatoria per approssimare l'integrale; in base alla scelta dei nodi distinguiamo in due sotto-famiglie

- Metodi di Adams - Bashforth, **AB**, famiglia di metodi espliciti, utilizziamo come nodi $x_n, x_{n-1}, \dots, x_{n-p}$;
- Metodi di Adams - Moulton, **AM** , famiglia di metodi impliciti, , utilizziamo come nodi $x_{n+1}, x_n, x_{n-1}, \dots, x_{n-p}$.

Notiamo subito che i nodi sono quasi tutti esterni all'intervallo di integrazione, ma questo non è un problema.

(7.6.24) Osservazione *Per i metodi* ***AB****: con la scelta di* $p = 0$*, ho il metodo di Eulero esplicito.*

Per i metodi ***AM****: con la scelta* $p = 0$ *ho il metodo dei trapezi; volendo immaginare una formula per* $p = -1$*, il metodo è comunque ad un passo, non posso fare di meno, ho il metodo di Eulero implicito.*

(7.6.25) Osservazione *Riprendendo la definizione* (7.6.1)*, per ogni metodo di Adams, risulta che* $\alpha_0 = 1$*,* $\alpha_i = 0$ *per* $i = 1, \dots, p$ *e i coefficienti* β_i *vengono prodotti dall'integrazione numerica (sono i pesi a meno di un comune fattore moltiplicativo* h *che si può raccogliere).*

(7.6.26) Proposizione *Il grado di precisione delle formule di quadratura utilizzate nei metodi di Adams è il seguente:*

[12] Per ogni $p \in \mathbb{N}$ ne esistono due, uno esplicito ed uno implicito.

- ***AB**: $m = p$;*
- ***AM**: $m = p + 1$.*

*Allora se il metodo è **AB** ha ordine $p + 1$, se è **AM** ha ordine $p + 2$.*

Dimostrazione. Omettiamo la dimostrazione. ■

Possiamo già dire come implementare concretamente un metodo **AM**: utilizziamo una tecnica di tipo predictor-corrector, dove il predictor sarà un metodo **AB** corrispondente alla stessa scelta di p. Basta anche qui una sola iterazione del predictor.

Proprietà dei metodi di Adams Il generico metodo di Adams è

$$y_{n+1} = y_n + h \sum_{i=-1}^{p} \beta_i f_{n-i}$$

da cui $\rho(r) = r^{p+1} - r^p = r^p(r-1)$, da cui le radici sono $r_0 = 1$ e $r_j = 0$, per $j = 1, \dots, p$: ho quindi p radici multiple ma non hanno modulo unitario quindi non creano problemi. Possiamo quindi affermare che i metodi di Adams verificano sia la root condition sia la strong root condition. Pertanto siamo davanti a metodi convergenti, (zero) stabili e relativamente stabili.

Indice analitico