

Appunti del corso di esercitazioni di Calcolo Scientifico a.a. 2015-2016 “Problemi di Vibrazioni”

Dario A. Bini

19 ottobre 2015

1 Introduzione

Lo studio delle vibrazioni di sistemi di particelle è un tipico problema in cui il ruolo giocato dagli autovalori e dagli autovettori è particolarmente rilevante. In questo contesto i metodi numerici per il calcolo di autovalori e autovettori hanno una importanza determinante.

Molte applicazioni del calcolo scientifico si riconducono allo studio di problemi di vibrazioni. Giusto per fare qualche esempio: minimizzare le vibrazioni delle rotaie nei treni ad alta velocità, minimizzare il rumore che si genera dentro l'abitacolo delle auto a causa delle vibrazioni del motore e degli effetti aerodinamici, la progettazione dei ponti in modo che le frequenze a cui vengono sollecitati (cammino dei pedoni, traffico stradale, raffiche di vento) non coincidano con frequenze di risonanza del ponte stesso.

È significativo il caso del Millennium Bridge di Londra che fu chiuso a tre soli giorni dall'apertura poiché oscillava in modo pericoloso nel momento in cui le persone vi camminavano sopra, vedi:

[http://en.wikipedia.org/wiki/Millennium_Bridge_\(London\)](http://en.wikipedia.org/wiki/Millennium_Bridge_(London))

Infatti le frequenze laterali di oscillazione del ponte erano vicine alla frequenza del passo umano. Fu successivamente modificato inserendo degli ammortizzatori che aggiungevano forze resistenti. Per maggiori dettagli sulle applicazioni dei problemi di autovalori si veda il lavoro di Tisseur e Meerbergen “The quadratic eigenvalue problem”

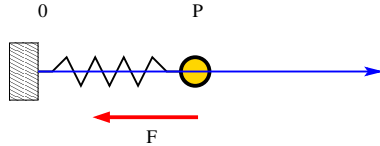
http://eprints.ma.man.ac.uk/466/01/covered/MIMS_ep2006_256.pdf

dove vengono toccati sistemi dinamici discreti, meccanica dei fluidi, acustica, trattamento dei segnali digitali. Applicazioni allo studio di vibrazione delle rotaie sono mostrati in

<http://math.cts.nthu.edu.tw/Mathematics/preprints/prep2007-1-005.pdf>

2 Il caso di un punto materiale

Studiamo il problema nel caso di una sola particella, cioè di un punto materiale P di massa m che si può muovere lungo una retta ed è soggetto a una forza elastica $F = -kOP$ di centro O e costante elastica $k > 0$, e a una forza resistente $R = -\theta v$, di attrito, proporzionale alla velocità v del punto con costante $\theta \geq 0$.



L'equazione di moto per questo semplice modello, ricavata dalla equazione generale $ma = F$, è

$$ma = -kOP - \theta v, \quad (1)$$

dove a è l'accelerazione del punto P . Indicando con $x = x(t)$ l'ascissa del punto P al tempo t si ottiene l'equazione differenziale

$$mx'' + \theta x' + kx = 0 \quad (2)$$

che, completata con le condizioni iniziali $x(0) = x_0$, $x'(0) = v_0$, individua una soluzione che costituisce l'espressione del moto del punto materiale p . Nell'ipotesi in cui θ e k siano costanti rispetto a t , l'insieme delle soluzioni dell'equazione differenziale si può esprimere in modo semplice mediante le soluzioni di un'equazione algebrica.

Cerchiamo soluzioni del tipo $x(t) = e^{\lambda t}$. Vale $x'(t) = \lambda e^{\lambda t}$, $x''(t) = \lambda^2 e^{\lambda t}$. Per cui sostituendo nell'equazione (2) si ottiene

$$m\lambda^2 e^{\lambda t} + \theta \lambda e^{\lambda t} + k e^{\lambda t} = 0,$$

da cui, essendo $e^{\lambda t} \neq 0$ si ottiene che una soluzione del tipo $e^{\lambda t}$ esiste se e solo se λ è soluzione dell'equazione algebrica

$$m\lambda^2 + \theta\lambda + k = 0.$$

Si hanno tre casi:

1. Se l'equazione ha due soluzioni distinte λ_1, λ_2 allora tutte le soluzioni di (2) si possono scrivere come

$$x(t) = \gamma_1 e^{\lambda_1 t} + \gamma_2 e^{\lambda_2 t}.$$

2. Se l'equazione ha una soluzione doppia $\lambda_1 = \lambda_2$ allora tutte le soluzioni di (2) si possono scrivere come

$$x(t) = (\gamma_1 + \gamma_2 t) e^{\lambda_1 t}.$$

3. Se l'equazione ha una coppia di soluzioni complesse coniugate $\lambda_{1/2} = \alpha \pm \beta i$ allora $e^{\lambda_{1/2}t} = e^{t(\alpha \pm \beta i)} = e^{\alpha t}(\cos(\beta t) \pm i \sin(\beta t))$. Per cui prendendo somma e differenza delle due soluzioni otteniamo una nuova base reale per lo spazio lineare delle soluzioni formata da $e^{\alpha t} \cos(\beta t)$ e $e^{\alpha t} \sin(\beta t)$. Otteniamo quindi per la soluzione generale l'espressione:

$$x(t) = e^{\alpha t}(\gamma_1 \cos(\beta t) + \gamma_2 \sin(\beta t)).$$

cioè il punto oscilla attorno ad O con frequenza $\beta/(2\pi)$ e ampiezza $e^{\alpha t}$. Le oscillazioni si smorzano se $\alpha < 0$, si amplificano se $\alpha > 0$.

In ogni caso l'insieme delle soluzioni costituisce uno spazio lineare di dimensione 2 e le costanti γ_1, γ_2 vengono determinate imponendo le due condizioni iniziali.

Poiché nel nostro caso è

$$\lambda_{1/2} = (-\theta \pm \sqrt{\theta^2 - 4km})/(2m)$$

si ha che, se $\theta^2 - 4km > 0$, cioè se la forza resistente domina la forza elastica, le soluzioni dell'equazione algebrica sono reali e negative. Siamo quindi nel caso 1. In questo caso il punto P converge a O senza oscillare.

Se invece $\theta^2 - 4km < 0$, cioè la forza resistente è dominata dalla forza della molla, le soluzioni sono oscillanti attorno ad O ma l'ampiezza delle oscillazioni converge a zero (oscillazioni smorzate) essendo $\alpha < 0$. La frequenza delle oscillazioni è data da $\frac{1}{2\pi} \sqrt{k/m - \theta^2/(4m^2)}$. Siamo quindi nel caso 3).

Il caso 2) si incontra quando $\theta^2 - 4km = 0$ e vale $\lambda_{1/2} = -\theta/(2m)$ e le soluzioni sono del tipo $e^{-t \frac{\theta}{2m}}(\gamma_1 + \gamma_2 t)$.

Nel caso di un'equazione non omogenea

$$mx'' + \theta x' + kx + f = 0$$

dove f è una funzione di t , l'insieme di tutte le soluzioni si ottiene sommando una qualsiasi soluzione della non omogenea con tutte le soluzioni dell'omogenea. Ad esempio, se f è costante allora una soluzione della non omogenea è data da $-f/k$.

Possiamo simulare il suono generato dalle vibrazioni di un punto materiale come quello descritto, registrando i valori numerici della funzione $e^{\alpha t}(\gamma_1 \cos \beta t + \gamma_2 \sin \beta t)$ per particolari valori delle costanti che intervengono e per un intervallo di tempo fissato. Il file con i valori numerici deve poi essere trasformato in un file di tipo `wav` "suonabile" col comando "play" disponibile nelle *release* di Linux. In ambiente Octave è possibile salvare un segnale memorizzato in un vettore `v` scrivendo i contenuti in un file di tipo `wav` col comando

```
wavwrite(v, 16000, 'nomefile')
```

Il valore 16000 è il sample rate cioè la frequenza di campionamento. Quindi un vettore di 16000 componenti produce un suono della durata di un secondo. È possibile scegliere sample rate diversi, ad esempio 8000 per ingombrare meno la memoria ma con un conseguente peggioramento della qualità del suono. Per

default il comando `wavwrite` utilizza 16 bit (2 byte) per registrare il valore di ogni singolo campione. Anche questo valore, detto bps (bit per sample), può essere modificato. Ad esempio

```
wavwrite(v, 16000, 8, 'nomefile')
```

salva il suono con un solo byte per campione.

Se vogliamo far suonare per la durata di un secondo una nota La dell'ottava centrale, che ha una frequenza di 440 Hz, basta creare il file `la.wav` coi comandi

```
t = [1:16000]/16000;
x = sin(440*2*pi*t);
wavwrite(x, 16000, 'la.wav')
```

e poi 'suonare' il file da un terminale linux col comando `play la.wav`.

Se vogliamo creare un suono stereofonico con un canale sinistro e un canale destro in generale diversi allora dobbiamo usare due vettori colonna `sx` e `dx`, creare la matrice `z=[x,y]` e creare il file col comando

```
wavwrite(z, 16000, 'nomefile')
```

Nel caso in cui il punto materiale sia vincolato a stare su un piano, e quindi P sia individuato da due coordinate (x, y) , l'equazione (1) riscritta nelle due componenti x e y dà origine a due equazioni differenziali

$$mx'' + \alpha x' + kx = 0,$$

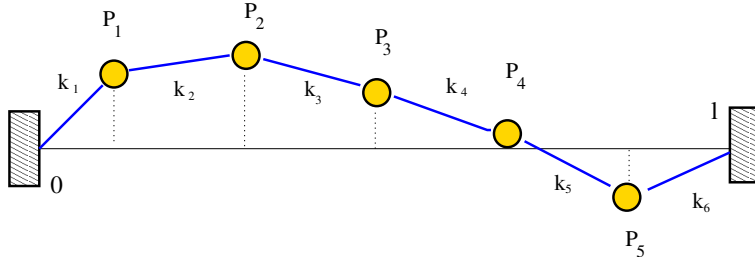
$$my'' + \alpha y' + ky = 0.$$

Analogamente nel caso di un punto libero di muoversi nello spazio abbiamo tre equazioni differenziali. Le equazioni sono indipendenti nel senso che possono essere risolte separatamente e le loro soluzioni determinano le coordinate al tempo t del punto materiale.

In questi casi è facile dare una espressione esplicita della soluzione esprimendo separatamente le componenti in termini di esponenziali e di funzioni trigonometriche. Il problema diventa più complesso per sistemi formati da più punti materiali.

3 Un sistema formato da più particelle

Supponiamo ora di avere n punti materiali P_1, P_2, \dots, P_n di massa m_1, m_2, \dots, m_n , che possono muoversi liberamente nel piano cartesiano. Assegnamo le coordinate ad ogni punto $P_i = (x_i, y_i)$ e definiamo $P_0 = (0, 0)$, $P_{n+1} = (\ell, 0)$ indipendenti dal tempo. Supponiamo che ogni punto sia soggetto ad una forza elastica e ad una forza resistente. Più precisamente il punto P_j è soggetto alla forza elastica $F_j^{(1)} = -k_j(P_j - P_{j-1})$ esercitata dal punto P_{j-1} e alla forza elastica $F_j^{(2)} = -k_{j+1}(P_j - P_{j+1})$ esercitata dal punto P_{j+1} , e alla forza resistente $R_j = -\theta_j v_j$, dove $\theta_j \geq 0$ e v_j è la velocità di P_j . Inoltre i punti P_1 e P_n sono soggetti alle forze elastiche rispettivamente $-k_1(P_1 - P_0)$ e $-k_{n+1}(P_n - P_{n+1})$. La figura seguente mostra la situazione con $n = 5$:



In questo caso, posto a_j l'accelerazione di P_j , le equazioni di moto del sistema di punti materiali sono

$$a_j m_j = -k_j(P_j - P_{j-1}) - k_{j+1}(P_j - P_{j+1}) - \theta v_j, \quad j = 1, \dots, n.$$

Proiettando le equazioni lungo l'asse delle x e delle y si ottengono i sistemi

$$\begin{aligned} m_j x_j'' + \theta x_j' - k_j x_{j-1} + (k_j + k_{j+1})x_j - k_{j+1}x_{j+1} &= 0, \\ m_j y_j'' + \theta y_j' - k_j y_{j-1} + (k_j + k_{j+1})y_j - k_{j+1}y_{j+1} &= 0, \end{aligned} \quad j = 1, \dots, n.$$

Questi sistemi possono essere riscritti in forma matriciale nel seguente modo. Posto $M = \text{diag}(m_1, \dots, m_n)$, K tale che $(K)_{i,i} = k_i + k_{i+1}$, $i = 1, \dots, n$, $(K)_{i,i+1} = (K)_{i+1,i} = -k_{i+1}$, $i = 1, \dots, n-1$, $R = \text{diag}(\theta_1, \dots, \theta_n)$, $\mathbf{x} = (x_1, \dots, x_n)$, $\mathbf{y} = (y_1, \dots, y_n)$, vale

$$M\mathbf{y}'' + R\mathbf{y}' + K\mathbf{y} = 0 \quad (3)$$

con le condizioni iniziali

$$\mathbf{y}(0) = \mathbf{y}_0, \quad \mathbf{y}'(0) = \mathbf{v}_0,$$

e le condizioni ai bordi $y_0(t) = y_{n+1}(t) = 0$ per ogni t . Analogamente per l'altro sistema

$$M\mathbf{x}'' + R\mathbf{x}' + K\mathbf{x} = 0 \quad (4)$$

con le condizioni iniziali

$$\mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{x}'(0) = \mathbf{x}_0,$$

e le condizioni ai bordi $x_0(t) = 0$, $x_{n+1}(t) = \ell$ per ogni t . Nel caso volessimo introdurre la forza peso allora le equazioni per le y_i cambiano in

$$M\mathbf{y}'' + R\mathbf{y}' + K\mathbf{y} + \mathbf{g} = 0$$

dove \mathbf{g} è il vettore di componenti $(-gm_i)$, g accelerazione di gravità.

Si osservi che K è matrice tridiagonale, inoltre se le costanti elastiche k_i sono non nulle la matrice è irriducibile.

Si osservi che, analogamente al caso monodimensionale, possiamo cercare soluzioni del tipo $\mathbf{y}(t) = \mathbf{w}e^{\lambda t}$, dell'equazione omogenea dove però stavolta

$\mathbf{w} \in \mathbb{R}^n$ è un vettore di costanti. Poichè $\mathbf{y}'(t) = \lambda \mathbf{w} e^{\lambda t}$, $\mathbf{y}''(t) = \lambda^2 \mathbf{w} e^{\lambda t}$, sostituendo nell'equazione differenziale si ottiene

$$(\lambda^2 M + \lambda R + K)\mathbf{w} = 0.$$

Esistono allora soluzioni del tipo ipotizzato se solo se λ risolve l'equazione $\det(\lambda^2 M + \lambda R + K) = 0$ e \mathbf{w} sta nel nucleo di tale matrice. Il problema del calcolo delle soluzioni λ, \mathbf{w} è noto come problema quadratico agli autovalori ed è ampiamente studiato in letteratura, si veda ad esempio l'articolo di Tisseur e Meerbergen già citato.

Un modo di risolverlo è quello di ricondurlo ad un problema standard agli autovalori. Noi faremo questa trasformazione direttamente sull'equazione differenziale (3) in modo da trasformarla in una equazione del primo ordine.

3.1 Riduzione di un sistema dal secondo ordine al primo ordine.

È possibile trasformare il sistema di equazioni differenziali del secondo ordine (3) in un sistema di equazioni differenziali del primo ordine nel modo seguente.

Sia $\mathbf{w}^T = (\mathbf{y}^T, \mathbf{y}'^T)$ il vettore di $2n$ componenti ottenuto giustapponendo il vettore delle velocità \mathbf{y}' al vettore degli spostamenti \mathbf{y} , per cui $\mathbf{w}'^T = (\mathbf{y}'^T, \mathbf{y}''^T)$. Vale allora

$$\mathbf{w}' = A\mathbf{w}, \quad A = \begin{bmatrix} 0 & I \\ -M^{-1}K & -M^{-1}R \end{bmatrix} \quad (5)$$

e le condizioni iniziali diventano

$$\mathbf{w}(0) = \mathbf{w}_0, \quad \mathbf{w}_0^T = (\mathbf{y}_0^T, \mathbf{y}'_0^T).$$

Il sistema di equazioni differenziali scritto nella forma (5) può essere facilmente risolto se siamo in grado di calcolare gli autovalori e gli autovettori della matrice A . Supponiamo per semplicità che A sia simile ad una matrice diagonale D , cioè $A = SDS^{-1}$, dove $D = \text{diag}(d_1, \dots, d_n)$ e S ha per colonne gli autovettori $\mathbf{s}_1, \dots, \mathbf{s}_n$ corrispondenti agli autovalori d_1, \dots, d_n di A .

Il sistema allora diventa $\mathbf{w}' = SDS^{-1}\mathbf{w}$, e posto $\mathbf{u} = S^{-1}\mathbf{w}$ si ottiene $\mathbf{u}' = D\mathbf{u}$. Cioè, in componenti si ha $u'_j = d_j u_j$, $j = 1, \dots, n$, da cui $u_j(t) = c_j e^{td_j}$. La soluzione risulta allora

$$\mathbf{w} = S\mathbf{u} = S \text{diag}(e^{td_i})\mathbf{c}, \quad \mathbf{c} = (c_i) \quad (6)$$

dove c_1, \dots, c_{2n} sono costanti che vengono determinate dalle condizioni iniziali

$$\mathbf{w}(0) = \begin{bmatrix} \mathbf{y}(0) \\ \mathbf{y}'(0) \end{bmatrix},$$

cioè

$$\mathbf{c} = S^{-1} \begin{bmatrix} \mathbf{y}(0) \\ \mathbf{y}'(0) \end{bmatrix}. \quad (7)$$

La soluzione allora è data da

$$\mathbf{w} = S \text{diag}(e^{td_i}) S^{-1} \begin{bmatrix} \mathbf{y}(0) \\ \mathbf{y}'(0) \end{bmatrix}$$

Allora diventa facile scrivere una function che calcola i valori di \mathbf{w} istante per istante e da in output le ordinate di una particolare componente, ad esempio l'ordinata del punto di mezzo o l'ordinata del primo punto.

Se campioniamo la vibrazione con un sample rate di 16000 Hz e se denotiamo con s il numero dei secondi in cui vogliamo studiare la vibrazione dobbiamo svolgere un campionamento agli istanti $t_i = i/16000$ per $i = 1, 2, \dots, s * 16000$.

Alla funzione daremo in input il vettore delle masse m_1, \dots, m_n , il vettore delle costanti elastiche k_1, k_2, \dots, k_{m+1} , il vettore dei coefficienti di attrito a_1, \dots, a_n , il vettore con posizione iniziale dei punti, il vettore delle velocità iniziali, la durata s in secondi e la coordinata intera del punto che vogliamo analizzare.

Il programma che implementa l'algoritmo è mostrato nel listing 1.

4 Sperimentazione

Riportiamo ora i risultati di alcune sperimentazioni fatte col nostro modello matematico. Consideriamo il caso di una corda discreta costituita da $n = 40$ masse uguali tra di loro con il valore $1/(1000n)$ con costanti elastiche uguali tra di loro pari a 4.0×10^4 e coefficienti di attrito pari a 10^{-4} . Si studiano due condizioni iniziali:

- le velocità sono nulle e gli spostamenti sono $y_i = i/q$, $i = 1, \dots, q$, $y_i = 1 - (i - q)/(n + 1 - q)$, per $i = q + 1, \dots, n$, dove q è un intero. Questa è la simulazione di una corda pizzicata nel punto q come nel caso del clavicembalo.
- Si considera invece il caso iniziale di spostamenti nulli e di velocità date da $y'_i = 10^4 * \exp(-0.1 * (i - q)^2)$ che simula il caso della corda colpita da un martelletto nel punto q come nel caso del pianoforte.

In entrambi i casi studiamo la vibrazione del secondo punto della corda, e la vibrazione del punto centrale.

I dati della corda pizzicata si ottengono in questo modo

```
n = 40;
q = 2;
y = [[1:q]/q, 1-[1:n-q]/(n+1-q)];
v = 0;
```

mentre i dati iniziali della corda colpita da un martelletto sono

```
n = 40;
q = 2;
p = zeros(1,n);
v = 1.e4*exp(-0.1*[1-q:n-2].^2);
```

Listing 1: Function vibrazioni

```

function [vib, vel] = vibrazioni(m, k, a, p, v, s, x)
% function vib = vibrazioni(m, k, a, p, v, s, x)
% calcola la vibrazione di una corda elastica discreta formata da n
% particelle p(1),...p(n) di massa m(1),...,m(n) soggette a forze di
% attrito di coefficienti a(1),...a(n), e a forze elastiche di costante
% elastica k(1),...,k(n+1) dove k(i) e' la costante della forza che
% agisce
% tra il punto p(i-1) e p(i) per i=1,...n+1, dove p(0) e p(n+1) sono i
% punti
% estremi della corda che sono assunti fissi
% i punti p(i) hanno ascissa costante uguale a i*h con h=L/(n+1)
% e L e' la lunghezza della corda. Mentre le ordinate variano col tempo
% In INPUT: m e' il vettore delle masse
% k e' il vettore delle costanti elastiche
% a e' il vettore delle costanti di attrito
% p e' il vettore delle ordinate dei punti all'istante t=0
% v e' il vettore delle velocita' all'istante t=0
% s e' il numero di secondi della simulazione
% x e' la coordinata intera del punto di cui si registrano le vibrazioni
% In OUTPUT: vib e' il vettore delle ordinate del punto p(x) al variare
% del tempo
% vel e' il vettore con le velocita' del punto p(x)

n=length(m);
% costruisco la matrice A
K = diag(k(1:n) + k(2:n+1)) - diag(k(2:n),1) - diag(k(2:n),-1);
A = zeros(2*n);
A(1:n,n+1:2*n) = eye(n);
A(n+1:2*n,n+1:2*n) = diag(-a./m);
A(n+1:2*n,1:n) = diag(-1./m)*K;

% calcolo autovalori e autovettori di A
[eigvec, eigval] = eig(A);
eigval = diag(eigval);

% formo il vettore delle condizioni iniziali
if size(p)(1)==1
    p = p';
end
if size(v)(1)==1
    v = v';
end
y = [p;v];
c = eigvec\y;

% calcolo e campiono il segnale
nt = round(s*16000);
deltat = 1/16000;
vib = zeros(nt,1); vel = vib;
for j=1:nt
    t = j*deltat;
    z = c.* exp(t*eigval);
    vib(j) = eigvec(x,:)*z;
    vel(j) = eigvec(n+x,:)*z;
end
vib = real(vib);
vel = real(vel);

```

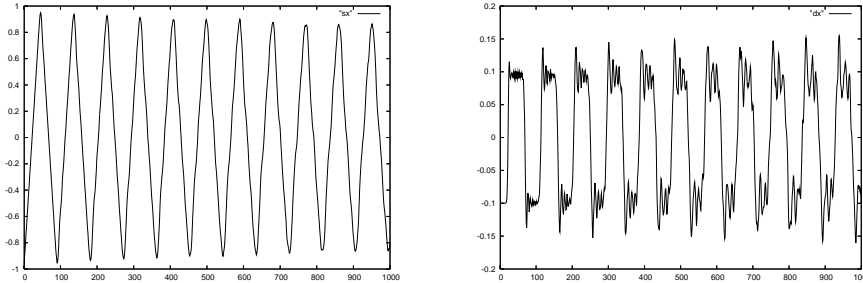



Figura 1: Grafici del primo quarantesimo di secondo della vibrazione di una corda pizzicata al centro. A sinistra oscillazioni rilevate al centro della corda; a destra oscillazioni vicino ad un estremo

I valori delle masse, costanti elastiche e coefficienti di attrito si assegnano così

```
m = ones(n,1)/(1000*n);
k = 4.e4*ones(n+1,1);
a = 1.e-4*ones(n, 1);
```

I 10 valori delle frequenze di vibrazione più basse (parte immaginaria degli autovalori) rapportate alla frequenza più bassa sono

- 1
- 1.998532681218886
- 2.994132168992735
- 3.9853373370719316
- 4.970693465860291
- 5.948754412924554
- 6.918084740483228
- 7.877261823379432
- 8.824877937402999
- 9.759542325489589

Si osservi che tali rapporti sono spesso vicini a numeri interi.

Caso a) Con $q = n/2$. I grafici dei due segnali relativi al primo quarantesimo di secondo (1000 campioni). sono riportati nella figura 1

Si noti come nella oscillazione del punto vicino all'estremità (figura di destra) siano evidenti componenti in alta frequenza che non compaiono nel caso della oscillazione del punto centrale.

Con $q = 2$ I grafici dei due segnali relativi al primo quarantesimo di secondo (1000 campioni). sono riportati nella figura 2

Osservate la presenza di una maggior quantità di componenti in alta frequenza sia nella figura di destra che di sinistra: pizzicare un corda in prossimità

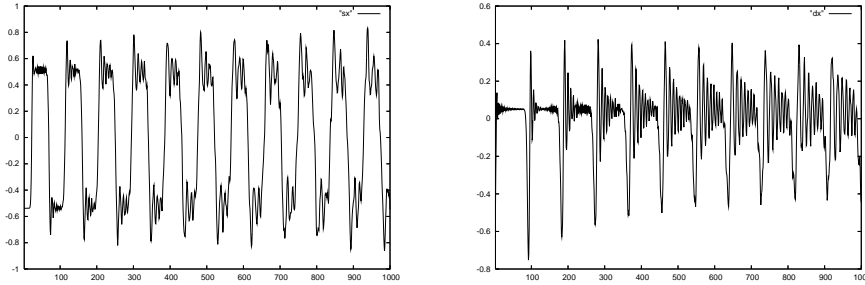


Figura 2: Grafici del primo quarantesimo di secondo della vibrazione di una corda pizzicata vicino ad un bordo. A sinistra oscillazioni rilevate al centro della corda; a destra oscillazioni vicino ad un estremo

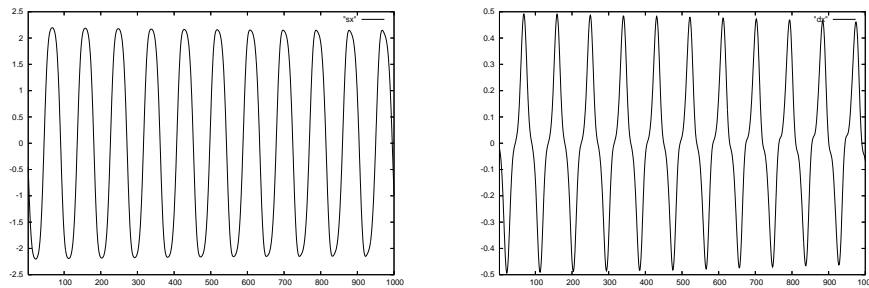


Figura 3: Simulazione della corda colpita da un martelletto al centro. A sinistra oscillazioni rilevate al centro della corda; a destra oscillazioni vicino ad un estremo

di un suo estremo dà un suono più argentino, sia se questo viene rilevato in prossimità di un estremo della corda (figura di destra) sia che venga rilevato nel centro (figura a sinistra).

Caso b) Con $q = n/2$ i grafici dei due canali sono riportati nella figura 3

Si noti in questo caso l'assenza di componenti in alta frequenza sui due canali. La corda non è pizzicata ma è idealmente percossa da un martelletto nella parte centrale

Con $q = 2$, cioè se il punto di percussione è vicino al bordo, i grafici dei due canali sono riportati nella figura

Anche in questo caso ci sono maggiori componenti in alta frequenza.

Per studiare le varie componenti in frequenza presenti nella vibrazione del punto selezionato della corda possiamo calcolare la trasformata discreta di Fourier del vettore \mathbf{vib} che ci dà lo spostamento del punto al variare del tempo. Ad

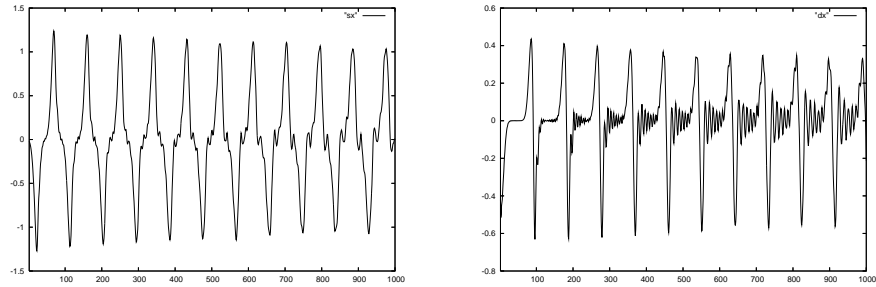
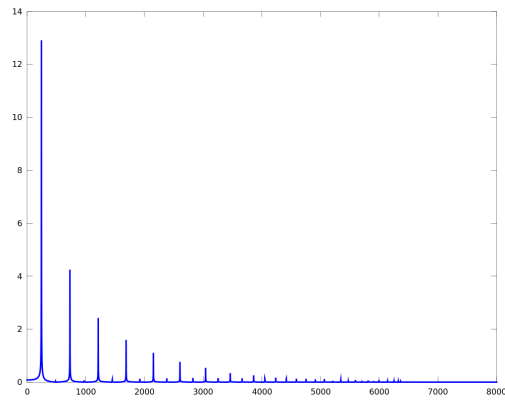


Figura 4: Simulazione della corda colpita da un martelletto vicino a un bordo. A sinistra oscillazioni rilevate al centro della corda; a destra oscillazioni vicino ad un estremo

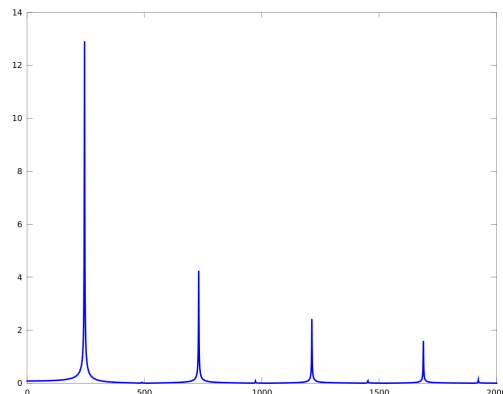
esempio scrivendo

```
f vib = fft(vib);
plot(abs(f vib(1:end/2)), 'LineWidth', 3);
```

si ottiene il grafico seguente



Ingrandendo la parte più a sinistra si ha



Si possono notare dei picchi dove le ampiezze sono elevate in corrispondenza solo di alcune frequenze specifiche. La più bassa di queste è la fondamentale, le altre sono le armoniche.

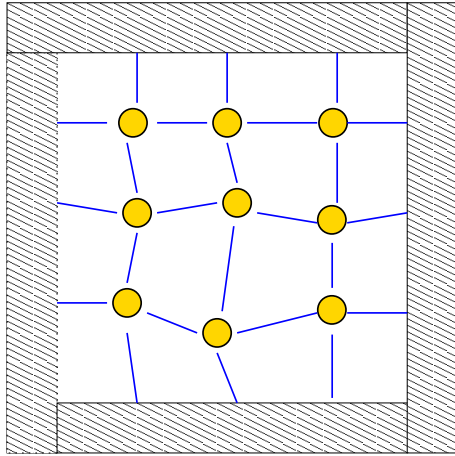
Vediamo ora alcuni esercizi.

Esercizio 1. La simulazione fatta con le condizioni iniziali descrive il comportamento di una corda pizzicata (clavicembalo). Si provino altre condizioni iniziali, ad esempio, per simulare la vibrazione di una corda colpita da un martelletto (tipo pianoforte) si ponga $\mathbf{y}_0 = \mathbf{0}$ e \mathbf{y}_0' di componenti $1000 * \exp(-(i - n/2)^2)$. Si provi a vedere come suona una corda in cui un punto subisce una forza resistente superiore rispetto agli altri punti. Si provi a simulare una corda con una distribuzione di massa non uniforme

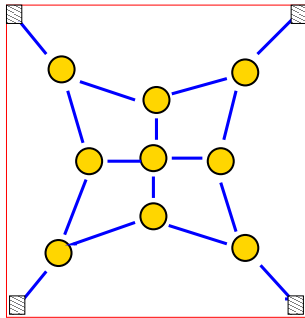
Esercizio 2. Si adatti il programma al caso del calcolo dei x_i . L'unica differenza rispetto al caso delle y_i è che le condizioni al bordo non sono omogenee essendo $x_{n+1} = \ell$. Si provi a registrare, nei file dx e sx anziché i valori di un particolare punto y_k , la media dei valori in un intorno di un particolare punto.

Esercizio 3. Si provino a costruire le matrici di elasticità K nel caso di un sistema di punti n^2 $p_{i,j}$, $i, j = 1, \dots, n$ liberi di muoversi nel piano in cui il punto $P_{i,j}$ di massa $m_{i,j}$ è soggetto alle quattro forze elastiche esercitate dai punti "contigui" $p_{i,j\pm 1}$, $p_{i\pm 1,j}$, come in figura.

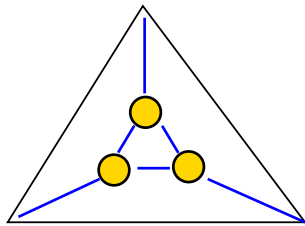
(Suggerimento: si definiscano le matrici $X = (x_{i,j})$, $Y = (y_{i,j})$ in cui $(x_{i,j}, y_{i,j})$ sono le coordinate del punto $P_{i,j}$; si definiscano anche le matrici K_x e K_y delle costanti elastiche delle forze "orizzontali" e "verticali").



Esercizio 4. Si consideri un problema analogo al precedente dove solo i punti sui vertici sono soggetti a forze elastiche dirette verso i vertici di un quadrato. (Suggerimento: modificare il problema precedente ponendo uguali a zero alcuni elementi delle matrici K_x e K_y)



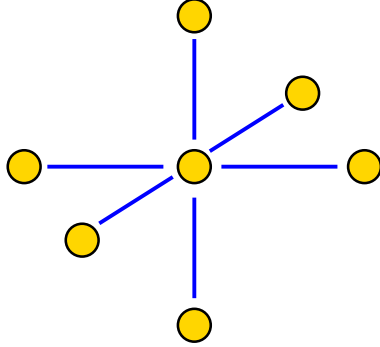
Esercizio 5. Si consideri il caso in figura



Esercizio 6. Si scrivano le equazioni di moto per un sistema come quelli dei tre casi precedenti dove i punti sono liberi di muoversi nello spazio.

Esercizio 7. Si scrivano le equazioni di moto di un sistema di n^3 punti $P_{i,j,k}$ liberi di muoversi nello spazio in cui il punto $P_{i,j,k}$ è soggetto alle 6 forze

elastiche esercitate dai punti contigui $P_{i\pm 1,j,k}$, $p_{i,j\pm 1,k}$, $p_{i,j,k\pm 1}$.



5 Un approccio diverso: l'esponenziale di matrice

Per risolvere il problema $\mathbf{w}' = A\mathbf{w}$ si consideri la seguente funzione di matrice

$$\exp(A) = \sum_{i=0}^{\infty} \frac{A^i}{i!}$$

che estende la funzione esponenziale al caso di matrici. Si osservi che la funzione è ben definita poichè la serie è convergente per ogni matrice quadrata A . Vale inoltre

$$\frac{d \exp(tA)}{dt} = A \exp(tA)$$

per cui possiamo descrivere le soluzioni dell'equazione differenziale $\mathbf{w}' = A\mathbf{w}$ come

$$\mathbf{w}(t) = \exp(tA)\mathbf{w}_0$$

dove $\mathbf{w}(0) = \mathbf{w}_0$.

La simulazione delle vibrazioni di una corda basate sul calcolo dell'esponenziale di una matrice è più onerosa poichè per ogni valore di t occorre calcolare l'esponenziale. Ciò ha un costo maggiore che non calcolare il valore di y_k una volta noti gli autovalori e autovettori di A .

La function che implementa questo approccio è riportata nel listing 2

6 Sistemi continui: l'equazione delle onde

I problemi trattati nelle sezioni precedenti hanno riguardato lo studio di vibrazioni di sistemi formati da un numero finito di punti materiali. Nel caso si considerino sistemi continui quali una corda vincolata a due estremità o

Listing 2: Risoluzione mediante l'uso dell'esponenziale di matrice

```

function [vib, vel] = vibrazioni_exp(m, k, a, p, v, s, x)
% function vib = vibrazioni(m, k, a, p, v, s, x)
% calcola la vibrazione di una corda elastica discreta
% usando l'esponenziale di matrice dato dalla funzione expm

n=length(m);
% costruisco la matrice A
K = diag(k(1:n) + k(2:n+1)) - diag(k(2:n),1) - diag(k(2:n),-1);
A = zeros(2*n);
A(1:n,n+1:2*n) = eye(n);
A(n+1:2*n,n+1:2*n) = diag(-a./m);
A(n+1:2*n,1:n) = diag(-1./m)*K;

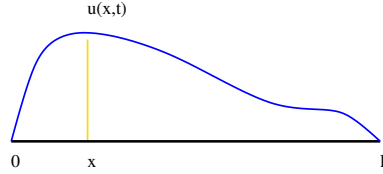
% formo il vettore delle condizioni iniziali
if size(p)(1)==1
    p = p';
end
if size(v)(1)==1
    v = v';
end
y = [p;v];

% calcolo e campiono il segnale
nt = round(s*16000);
deltat = 1/16000;
vib = zeros(nt,1); vel = vib;
for j=1:nt
    t = j*deltat;
    z = expm(t*A)*y;
    vib(j) = z(x);
    vel(j) = z(x+n);
end

```

una membrana elastica vincolata a un certo bordo dato da una linea chiusa, il modello matematico da considerare è formato da equazioni differenziali.

Consideriamo il problema di una corda elastica vibrante sottesa tra due estremi di coordinate $(0, 0)$ e $(\ell, 0)$ descritta come descritta in figura



dove denotiamo con $u(x, t)$ l'ordinata del suo punto di ascissa x al tempo t . Il moto dei punti di questa corda è governato dalla seguente equazione differenziale nota come *equazione delle onde*

$$\frac{\partial^2}{\partial t^2} u(x, t) - \sigma \frac{\partial^2}{\partial x^2} u(x, t) = 0 \quad (8)$$

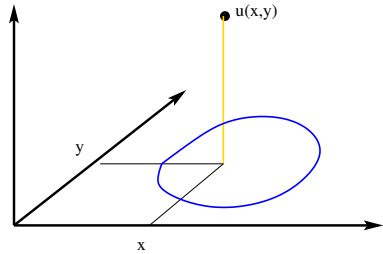
con $\sigma = Tg/w$, dove T è la tensione della corda, g è l'accelerazione di gravità e w è il peso per unità di lunghezza (C. Gerald, P.O. Wheatly, "Applied Numerical Analysis", Pearson, Addison-Wesley, 2004).

Tale equazione viene complementata con le seguenti condizioni aggiuntive

$$\begin{aligned} u(x, 0) = p(x), \quad u'(x, 0) = v(x), \quad x \in [0, \ell] & \quad \text{Condizioni iniziali} \\ u(0, t) = 0, \quad u(\ell, t) = 0 & \quad \text{Condizioni al bordo} \end{aligned} \quad (9)$$

La corda viene quindi abbandonata al tempo $t = 0$ in posizione $p(x)$ con velocità iniziale $v(x)$.

Nel caso bidimensionale di una membrana elastica vincolata a stare su un contorno chiuso Γ che racchiude un insieme aperto $\Omega \subset \mathbb{R}^2$, come mostrato in figura



l'equazione diventa

$$\frac{\partial^2}{\partial t^2} u(x, y, t) - \sigma \left(\frac{\partial^2}{\partial x^2} u(x, y, t) + \frac{\partial^2}{\partial y^2} u(x, y, t) \right) = 0, \quad (x, y) \in \Omega \quad (10)$$

con le condizioni aggiuntive

$$\begin{aligned} u(x, y, 0) = p(x, y), \quad u'(x, y, 0) = v(x), \quad (x, y) \in \Omega & \quad \text{Condizioni iniziali} \\ u(x, y, t) = 0, \quad (x, y) \in \Gamma & \quad \text{Condizioni al bordo} \end{aligned} \quad (11)$$

Siamo interessati alle soluzioni in cui tutti i punti oscillano con la stessa frequenza, cioè soluzioni del tipo $u(x, t) = v(x) \cos \omega t$, $v(0) = v(\ell) = 0$. Sostituendo questa espressione in (8), poiché $(\cos \omega t)'' = -\omega^2 \cos \omega t$, si ottiene

$$\sigma v''(x) = -\lambda v(x), \quad \lambda = \omega^2 / \sigma \quad (12)$$

Esistono diversi modi di approssimare la soluzione dell'equazione delle onde. Uno di questi si basa sul metodo delle differenze finite in cui si approssima l'operatore derivata seconda con opportuni rapporti incrementali. Per questo occorre assumere che la soluzione $v(x)$ sia di classe $C^4[0, \ell]$.

Si fissi un intero $n > 0$, si ponga $h = \ell / (n+1)$ e si definiscano i punti $x_i = ih$, $i = 0, 1, \dots, n+1$. Si considerino i seguenti sviluppi in serie:

$$\begin{aligned} v(x_{i+1}) &= v(x_i) + hv'(x_i) + \frac{h^2}{2}v''(x_i) + \frac{h^3}{3!}v'''(x_i) + \frac{h^4}{4!}v^{(4)}(\xi_i) \\ v(x_{i-1}) &= v(x_i) - hv'(x_i) + \frac{h^2}{2}v''(x_i) - \frac{h^3}{3!}v'''(x_i) + \frac{h^4}{4!}v^{(4)}(\eta_i) \end{aligned}$$

dove $\xi_i \in (x_i, x_{i+1})$, $\eta_i \in (x_{i-1}, x_i)$. Sommando entrambi i membri delle due equazioni e ricavando $v''(x_i)$ si ha

$$v''(x_i) = \frac{1}{h^2}(v_{i-1} - 2v_i + v_{i+1}) - h^2\tau_i$$

dove si è posto $v_i = v(x_i)$, $\tau_i = \frac{h^3}{4!}v^{(4)}(\xi_i) + \frac{h^3}{4!}v^{(4)}(\eta_i)$. Si osservi che, poiché $v \in C^4[0, \ell]$, risulta $|v^{(4)}(x)| \leq M$ per una costante $M > 0$, per cui $|\tau_i| \leq \frac{1}{12}M$.

Possiamo allora restringere il nostro interesse al calcolo dei soli v_i . Abbiamo la seguente equazione

$$\frac{1}{h^2}(v_{i-1} - 2v_i + v_{i+1}) = -\lambda v_i + h^2\tau_i, \quad i = 1, 2, \dots, n. \quad (13)$$

In forma matriciale abbiamo

$$\frac{1}{h^2}A\mathbf{v} = \lambda\mathbf{v} - h^2\boldsymbol{\tau} \quad (14)$$

con $A = \text{tridiag}(-1, 2, -1)$, $\mathbf{v} = (v_i)$, $\boldsymbol{\tau} = (\tau_i)$.

Poiché non conosciamo i valori di τ_i e poiché $h \rightarrow 0$ è ragionevole considerare il problema agli autovalori

$$A\mathbf{w} = \mu\mathbf{w}. \quad (15)$$

Dobbiamo però dimostrare che le soluzioni di (15) sono buone approssimazioni delle soluzioni del problema originale (14).

Per questo ci è di aiuto il teorema di Bauer-Fike.

Riscriviamo la (14) nella forma

$$\left(\frac{1}{h^2}A + \frac{h^2}{\|\mathbf{v}\|_2^2}\boldsymbol{\tau}\mathbf{v}^T\right)\mathbf{v} = \lambda\mathbf{v} \quad (16)$$

e applichiamo il teorema di Bauer-Fike alle matrici $B = \frac{1}{h^2}A$ e $B + F$ con $F = \frac{h^2}{\|\mathbf{v}\|_2^2}\boldsymbol{\tau}\mathbf{v}^T$.

Teorema 1 (Bauer-Fike) *Siano H, B, F matrici $n \times n$ tali che $H = B + F$. Supponiamo che B sia diagonalizzabile, cioè $B = SDS^{-1}$ con D matrice diagonale. Sia inoltre $\|\cdot\|$ una norma assoluta. Allora, per ogni autovalore λ di H esiste un autovalore μ di B tale che*

$$|\lambda - \mu| \leq \|F\| \cdot \|S\| \cdot \|S^{-1}\|$$

Un'ulteriore analisi ci permette di determinare una maggiorazione dell'errore sugli autovettori.

Teorema 2 *Nelle ipotesi del teorema di Bauer-Fike, sia $\mathbf{u} \in \mathbb{R}^n$ tale che, $(B + F)\mathbf{u} = \lambda\mathbf{u}$. Allora esistono un autovettore \mathbf{v} e un autovalore μ di B tali che $B\mathbf{v} = \mu\mathbf{v}$ e*

$$\frac{\|\mathbf{v} - \mathbf{u}\|}{\|\mathbf{u}\|} \leq \|(B - \mu I)^+\| \cdot (\|F\| + |\lambda - \mu|) \leq \|(B - \mu I)^+\| \cdot \|F\| (1 + \|S\| \cdot \|S^{-1}\|)$$

dove $(B - \mu I)^+$ è l'inversa generalizzata di $B - \mu I$ e $S^{-1}BS = D$ è diagonale. In particolare, se B è simmetrica risulta

$$\frac{\|\mathbf{v} - \mathbf{u}\|_2}{\|\mathbf{u}\|_2} \leq \frac{2}{\min_{t \in \sigma(B), t \neq \mu} |t - \mu|} \|F\|_2.$$

dove $\sigma(B)$ è l'insieme degli autovalori di B .

Dim. Vale

$$(B + F - \lambda I)\mathbf{u} = 0$$

$$B\mathbf{v} = \mu\mathbf{v}$$

Sottraendo entrambi i membri delle precedenti equazioni si ottiene

$$B(\mathbf{u} - \mathbf{v}) + F\mathbf{u} + (\lambda - \mu)\mathbf{u} - \mu(\mathbf{u} - \mathbf{v}) = 0.$$

Da cui

$$(B - \mu I)(\mathbf{u} - \mathbf{v}) = -F\mathbf{u} - (\lambda - \mu)\mathbf{u}$$

Quindi $F\mathbf{u} + (\lambda - \mu)\mathbf{u}$ sta nell'immagine di $B - \mu I$ ed esiste una soluzione $\mathbf{u} - \mathbf{v}$ di minima norma tale che

$$\mathbf{u} - \mathbf{v} = -(B - \mu I)^+(F + (\lambda - \mu)I)\mathbf{u}$$

inoltre vale

$$\|\mathbf{u} - \mathbf{v}\| \leq \|(B - \mu I)^+\| (\|F\| + |\lambda - \mu|) \|\mathbf{u}\|.$$

Da cui la prima parte della tesi. Se B è simmetrica, con autovalori β_i , si ha $\|S\|_2 \cdot \|S^{-1}\|_2 = 1$, per cui la tesi discende dal fatto che i valori singolari di $(B - \mu I)^+$, per definizione di inversa generalizzata, sono $1/|\beta_i - \mu|$ se $\beta_i - \mu \neq 0$ e 0 altrimenti. \square

Nel caso in cui la matrice B è $(1/h^2)\text{trid}(-1, 2, -1)$, e $F = \boldsymbol{\tau}\boldsymbol{v}^T$, risulta $\|F\| = \frac{h^2}{\|\boldsymbol{v}\|^2} \|\boldsymbol{\tau}\| \cdot \|\boldsymbol{v}\| = h^2 \|\boldsymbol{\tau}\| / \|\boldsymbol{v}\|$, per cui

$$|\lambda - \mu| \leq h^2 \frac{\|\boldsymbol{\tau}\|}{\|\boldsymbol{v}\|}. \quad (17)$$

Si osserva che il quoziente $\frac{\|\boldsymbol{\tau}\|}{\|\boldsymbol{v}\|}$ per $n \rightarrow \infty$ converge a

$$\int_0^\ell 2(v''(x))^2 dx / \int_0^\ell v(x)^2 dx$$

che ha un valore finito.

Inoltre, poiché gli autovalori di B sono $\beta_i = (n+1)^2(2 - 2\cos \pi i/(n+1)) = i^2\pi^2 + O(h^2)$, se $\mu = \beta_j$ allora $1/\min_{i \neq j} |\beta_i - \mu| = 1/(\pi^2(2i-1)) + O(h^2)$ che non dipende da n . Per cui

$$\frac{\|\boldsymbol{w} - \boldsymbol{v}\|_2}{\|\boldsymbol{v}\|_2} \leq h^2 \frac{2}{\pi^2(2i-1)} \int_0^\ell 2(v''(x))^2 dx / \int_0^\ell v(x)^2 dx + O(h^2)$$

6.1 Caso bidimensionale

Nel caso della vibrazione di una membrana vincolata al bordo di un quadrato di lato ℓ l'equazione (13) diventa

$$\frac{1}{h^2}(v_{i-1,j} + v_{i,j-1} - 4v_{i,j} + v_{i,j+1} + v_{i+1,j}) = -\lambda v_{i,j} + h^2 \tau_{i,j}, \quad i, j = 1, 2, \dots, n, \quad (18)$$

dove $v_{i,j} = v(x_i, y_j)$, $x_i = \ell * i/(n+1)$, $y_j = \ell * j/(n+1)$. Il problema agli autovalori da risolvere è allora

$$-\frac{1}{h^2} A \boldsymbol{w} = \mu \boldsymbol{w}. \quad (19)$$

dove la matrice $-A$ è la matrice tridiagonale a blocchi $n \times n$ con blocchi sopra diagonali e sotto diagonali uguali a $-I$ e blocchi diagonali uguali a

$$\begin{bmatrix} 4 & -1 & & & \\ -1 & \ddots & \ddots & & \\ & \ddots & \ddots & -1 & \\ & & & -1 & 4 \end{bmatrix}.$$

7 Calcolo dei modi di vibrazione

Nelle applicazioni hanno maggior interesse gli autovalori più piccoli ed è importante poter calcolare gli autovettori corrispondenti che approssimano le autofunzioni del problema continuo. Questi autovettori sono chiamati i modi di

vibrazione. In questa sezione ci occupiamo di questo calcolo per quanto riguarda il problema bidimensionale (19).

Assumiamo di avere un dominio rettangolare discretizzato con un reticolo di $m \times n$ punti interni, di modo che la matrice in (19) sia $m \times m$ a blocchi con blocchi $n \times n$. Supponiamo che $h = 1/(\max(m, n) + 1)$ e occupiamoci del calcolo degli autovalori di A che alla fine verranno moltiplicati per $1/h^2$. Più precisamente, dato un intero k molto più piccolo di n e m , il nostro scopo è calcolare i più piccoli k autovalori di A . Si osserva che, poiché i metodi numerici disponibili, tipo il metodo delle potenze e le sue varianti, calcolano i k più grandi autovalori di A è necessario fare opportune manipolazioni. Siano per questo

$$\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{mn}$$

gli autovalori di A .

Una prima possibilità consiste nell'applicare il metodo delle potenze e le sue varianti alla matrice $B = A^{-1}$ che ha autovalori

$$\mu_{mn} \leq \mu_{mn-1} \leq \dots \leq \mu_1, \quad \mu_i = \lambda_{mn-i+1}^{-1}.$$

Il metodo delle potenze e le sue varianti, approssimano gli autovalori di modulo più grande, cioè μ_1, \dots, μ_k che forniscono gli autovalori desiderati $\lambda_i = \lambda_{mn-i+1}^{-1}$ per $i = 1, 2, \dots, k$. Per fare questo occorre avere un risolutore efficiente del sistema lineare $Ax = b$. La velocità di convergenza dipende dal rapporto λ_1/λ_k .

Una seconda possibilità consiste nel trasformare gli autovalori evitando di ricorrere a inversioni di matrici. Per questo si osservi che, per il teorema di Gerschgorin risulta $0 < \lambda_1, \lambda_{mn} < 8$, per cui la matrice $B = 8I - A$ ha autovalori

$$\mu_{mn} \leq \mu_{mn-1} \leq \dots \leq \mu_1$$

dove $\mu_i = 8 - \lambda_i$. Quindi, applicando il metodo delle potenze e le sue varianti a B si calcolano μ_1, \dots, μ_k e quindi $\lambda_1, \dots, \lambda_k$. Si osserva che la matrice B ha la stessa struttura tridiagonale a blocchi di A dove i blocchi sopra e sotto diagonali sono matrici identiche e il blocco diagonale è la matrice tridiagonale con elementi diagonali uguali a 4 e sopra sotto diagonali uguali a 1. Si osserva ancora che se un vettore \mathbf{v} di mn componenti viene organizzato come matrice $m \times n$ $V = (v_{i,j})$, allora il prodotto $\mathbf{u} = A\mathbf{v}$ è tale che

$$u_{i,j} = \frac{1}{h^2}(v_{i-1,j} + v_{i,j-1} + 4v_{i,j} + v_{i,j+1} + v_{i+1,j}) \quad (20)$$

Questo approccio ha il vantaggio di non richiedere la risoluzione di sistemi lineari ma ha lo svantaggio che il rapporto μ_k/μ_1 che determina la velocità di convergenza del metodo delle potenze è prossimo a 1. Questo comporta un elevato numero di iterazioni.

7.1 Iterazione dei sottospazi (generalizzazione del metodo delle potenze)

In questa sezione assumiamo che il problema da risolvere sia quello di calcolare i k autovalori più grandi di una matrice A di dimensione $n \times n$ reale simmetrica, definita positiva.

Il metodo delle iterazioni dei sottospazi, detto anche delle iterazioni ortogonali, è una generalizzazione del metodo delle potenze in cui, anziché costruire una successione di vettori che converge all'autovettore dominante della matrice, si costruisce una successione di sottospazi, tutti di dimensione k che converge al sottospazio invariante di A generato dai primi k autovettori.

Ricordiamo prima il metodo delle potenze nella forma:

$$\begin{aligned} \mathbf{y}^{(\nu)} &= A\mathbf{x}^{(\nu)} \\ \mathbf{x}^{(\nu+1)} &= \mathbf{y}^{(\nu)} / \|\mathbf{y}^{(\nu)}\| \quad \nu = 0, 1, 2, \dots \\ \sigma_\nu &= y_i^{(\nu)} / x_i^{(\nu)}, \text{ dove } x_i^{(\nu)} \neq 0 \end{aligned}$$

Si dimostra facilmente che $|\sigma_\nu - \lambda_1| = O(|\lambda_2/\lambda_1|^\nu)$, dove λ_1 è l'autovalore di massimo modulo di A e si assume che $|\lambda_2| < |\lambda_1|$.

Nel caso in cui A sia simmetrica si ottiene una migliore convergenza con la seguente variante

$$\begin{aligned} \mathbf{y}^{(\nu)} &= A\mathbf{x}^{(\nu)} \\ \mathbf{x}^{(\nu+1)} &= \mathbf{y}^{(\nu)} / \|\mathbf{y}^{(\nu)}\|_2 \quad \nu = 0, 1, 2, \dots \\ \sigma_\nu &= \mathbf{y}^{(\nu)T} \mathbf{x}^{(\nu)} \end{aligned}$$

Infatti si può dimostrare che $|\sigma_\nu - \lambda_1| = O(|\lambda_2/\lambda_1|^{2\nu})$.

Il metodo delle iterazioni dei sottospazi viene dato in due versioni che generalizzano, in qualche forma, le due versioni date del metodo delle potenze.

Data la matrice $n \times k$ $X^{(0)}$ con colonne ortogonali, la prima versione del metodo è la seguente

$$\begin{aligned} Y^{(\nu)} &= AX^{(\nu)} \\ Y^{(\nu)} &= Q^{(\nu)}R^{(\nu)} \quad \nu = 0, 1, 2, \dots \\ X^{(\nu+1)} &= Q^{(\nu)} \end{aligned}$$

dove $Y^{(\nu)} = Q^{(\nu)}R^{(\nu)}$ è la fattorizzazione QR della matrice $Y^{(\nu)}$. Si dimostra che per quasi ogni scelta di $X^{(0)}$, se $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k > \lambda_{k+1}$ allora

$$|r_{i,i}^{(\nu)} - \lambda_i| \leq \gamma(|\lambda_{i+1}/\lambda_i|^\nu + |\lambda_i/\lambda_{i-1}|^\nu)$$

dove se $i = 1$ il termine $|\lambda_i/\lambda_{i-1}|^\nu$ non compare.

L'inconveniente di questo metodo è che per avere convergenza occorre la stretta separazione di tutti gli autovalori da λ_1 a λ_{k+1} .

Questo inconveniente viene rimosso se si adotta la seconda variante:

$$\begin{aligned}
Y^{(\nu)} &= AX^{(\nu)} \\
Y^{(\nu)} &= Q^{(\nu)}R^{(\nu)} \\
B^{(\nu)} &= Q^{(\nu)T}AQ^{(\nu)} \quad \nu = 0, 1, 2, \dots \\
B^{(\nu)} &= U^{(\nu)}D^{(\nu)}U^{(\nu)T} \\
X^{(\nu+1)} &= Q^{(\nu)}U^{(\nu)}
\end{aligned} \tag{21}$$

Si osserva che le colonne della matrice $X^{(\nu)}$ generata nelle due varianti generano lo stesso sottospazio vettoriale, cioè quello generato dalle colonne di $A^\nu X^{(0)}$. Le due basi però sono differenti, e nel caso della seconda variante si può dimostrare che per quasi ogni scelta di $X^{(0)}$ esiste una costante $\gamma > 0$ tale che

$$|d_{i,i}^{(\nu)} - \lambda_i| \leq \gamma(|\lambda_{k+1}/\lambda_i|^\nu).$$

Dalla relazione precedente segue che la convergenza della seconda variante è migliore e che è garantita dalla condizione più debole $\lambda_i > \lambda_{k+1}$.

7.2 Implementazione

L'implementazione del metodo delle iterazioni dei sottospazi nella forma (21), sia applicato ad A^{-1} che applicato a $8I - A$, si realizza agevolmente in Octave. Occorre dire che Octave dispone del comando `eigs` che permette di calcolare solo alcuni autovalori e autovettori di una matrice densa o sparsa e per questo implementa in modo efficiente metodi basati sull'iterazione delle potenze e dei sottospazi di Krylov. In questa sezione realizziamo le varie possibilità implementative.

Consideriamo inizialmente il problema del calcolo dei più piccoli k autovalori, e dei corrispondenti autovettori, della matrice \mathcal{A} tridiagonale a blocchi $n \times n$ con blocchi diagonali $m \times m$ dati da $\text{trid}_m(-4, 1, -4)$ e blocchi sopra/sotto diagonali dati da $-I_m$. Questa matrice si può scrivere come $I_n \otimes H_m + H_n \otimes I_n$ con $H_m = \text{trid}_m(-1, 2, -1)$ ed ha autovalori e autovettori esprimibili esplicitamente. Infatti per gli autovalori vale $\lambda_{i,j} = 4 - 2 \cos \frac{\pi i}{m+1} - 2 \cos \frac{\pi j}{n+1}$, $i = 1, \dots, m$, $j = 1, \dots, n$. Mentre per gli autovettori corrispondenti $v^{(i,j)} = (v_{h,\ell}^{(i,j)})$ vale $v_{k,\ell}^{(i,j)} = \frac{2}{\sqrt{(n+1)(m+1)}} \sin \pi i k m + 1 \sin \frac{\pi j \ell}{n+1}$. Andiamo calcolare numericamente queste quantità, poi estenderemo il calcolo ad altri domini per i quali non è possibile dare una espressione esplicita degli autovalori e autovettori.

Partiamo dalla possibilità più semplice: quella dell'applicazione del comando `eigs`. Questo comando ha varie modalità di applicazione che si possono trovare sul manuale di Octave o dando il comando `help eigs`. A noi basta dare il comando

```
[evec, eval] = eigs(a, k, 'sa');
```

dove `a` è la matrice di cui calcolare gli autovalori, `'sa'` significa algebraically smallest cioè i più piccoli autovalori (non in valore assoluto), `evec` è la variabile che contiene gli autovettori, `eval` è la variabile che contiene gli autovalori.

Per costruire la matrice con $m = 20$ e $n = 30$ procediamo così:

```

m = 20; n = 30;
Hn = 2*eye(n) - diag(ones(n-1,1),-1) - diag(ones(n-1,1),1);
Hn = sparse(Hn);
Hm = 2*eye(m) - diag(ones(m-1,1),-1) - diag(ones(m-1,1),1);
Hm = sparse(Hm);
A = kron(eye(m),Hn) + kron(Hm,eye(n));

```

Dove abbiamo usato il comando `kron` per costruire il prodotto di Kronecker e il comando `sparse` per usare matrici in modalità sparsa.

Qui sotto riportiamo i tempi richiesti per vari valori di $n = m$ per il calcolo dei 10 autovalori più piccoli e dei corrispondenti autovettori usando un PC con CPU i3.

$m = n$	sec
20	0.01
40	0.06
80	0.5
160	6*

dove per $m = n = 160$ viene riportato il messaggio

Only 2 of the 10 requested eigenvalues converged

Ciò significa che la function di Octave non è stata in grado di calcolare tutti e 10 gli autovalori col numero prefissato di iterazioni usate dalla function. In uscita vengono comunque fornite approssimazioni di 10 autovalori/autovettori.

Per poter tracciare il grafico di ciascun autovettore dobbiamo trasformare l'autovettore dato come mn -upla in una matrice di dimensioni $m \times n$. Per fare questo usiamo il comando `reshape` nel modo seguente. Se ad esempio, vogliamo tracciare il quinto autovettore scriviamo

```

v = evec(:,5);
v = reshape(v, n, m);
mesh(v, 'LineWidth',2)

```

L'opzione `LineWidth` serve a tracciare linee del grafico un po' più spesse.

Per una migliore descrizione dell'autofunzione ricordiamoci che essa è nulla sul bordo. Allora per questo ricopiamo l'autovettore in una matrice più grande con valori nulli sulla prima e ultima riga e colonna. Facciamo fare questo a una funzione che si chiama `tracciabene`

```

function tracciabene(v,m,n)
v = reshape(v, n, m);
w = zeros(m+2,n+2);
w(2:m+1,2:n+1) = v;
mesh(w, 'LineWidth',2)

```

Adesso forniamo un'implementazione del metodo di iterazione dei sottospazi nella variante (21) applicata alla matrice $8I - A$. Ad ogni passo del metodo

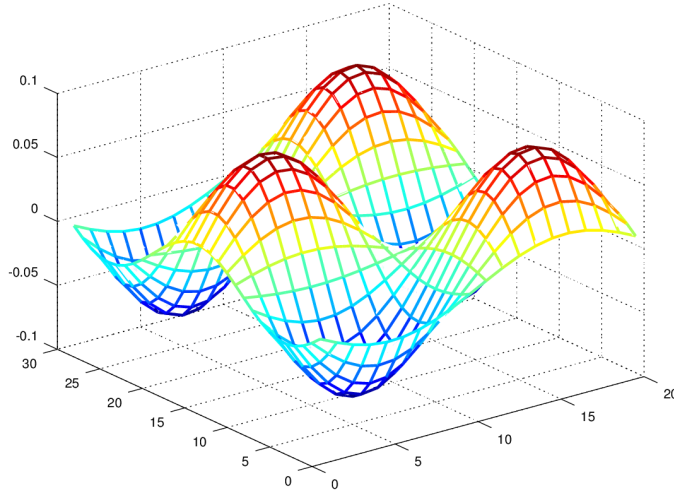


Figura 5: Quinto autovettore con $m = 20$, $n = 30$.

dobbiamo moltiplicare la matrice $8I - A$ per un vettore. Forniamo due versioni. La prima, in cui il prodotto matrice vettore lo lasciamo svolgere ad Octave traendo vantaggio dalla sparsità della matrice, la seconda in cui implementiamo questo prodotto per conto nostro.

La prima versione è riportata nella function del listing 3

Il numero di iterazioni impiegate con $m = 20$, $n = 20$, $k = 10$ per arrivare ad un errore inferiore a 10^{-13} è pari a 3389 per un tempo di CPU di 8.7 secondi. Con $m = n = 40$ si sale a oltre 10.000 iterazioni con un tempo superiore ai 1500 secondi.

La versione in cui il prodotto matrice vettore viene calcolato usando la struttura specifica della matrice si basa sulla function riportata nel listing 4

La function che implementa le iterazioni sui sottospazi va modificata sostituendo all'istruzione $y=A*x$ l'istruzione

```
y = prodotto(m, n, k, x);
```

e similmente per il calcolo di $A*q$. La function ottenuta è riportata nel listing 5.

Adesso implementiamo la versione in cui l'iterazione dei sottospazi è applicata alla matrice A^{-1} . Il listing 3 verrà modificato sostituendo semplicemente l'istruzione $y=A*x$; con $y = A \setminus x$; e l'istruzione $b=q'*(A*q)$; con $b=q'*A \setminus q$;

Il numero di iterazioni e i tempi di CPU che otteniamo in questo caso sono riportati di seguito

$n = m$	it	sec
20	195	0.4
40	198	26.6
80	195	398

Listing 3: Iterazione dei sottospazi, prima versione

```
function [evec, eval] = sottospazi1(A, k, itmax)
N = size(A)(1)
x = rand(N,k);
[q r] = qr(x); x=q(:,1:k); % ortogonalizzo x
dd = zeros(k,1);
for it=1:itmax
    y = A*x;
    [q, r] = qr(y); q=q(:,1:k); % fattorizzazione QR
    b=q'*(A*q);
    [u, d] = eig(b);
    [ds, s] = sort(diag(d)); % permuto gli autovalori e gli autovettori
    u = u(:,s);
    err = norm(dd-ds,'inf');
    dd = ds;
    disp([it, err])
    if err<1.e-13
        break
    end
    x = q*u;
end
eval = dd;
evec = x;
if err >= 1.e-13
    disp('raggiunto il limite massimo di iterazioni')
end
```

Listing 4: Calcolo del prodotto matrice vettore

```
function y = prodotto1(m, n, k, x)
% function y = prodotto1(m, n, k, x)
% calcola il prodotto y della matrice trid_m(I,H,I) col vettore x mxnk
% dove H=trid_n(1,4,1)
X = reshape (x, n, m, k);
Y = 4*X;
Y(1:end-1, :, :) = Y(1:end-1, :, :) + X(2:end, :, :);
Y(2:end, :, :) = Y(2:end, :, :) + X(1:end-1, :, :);
Y(:, 1:end-1, :) = Y(:, 1:end-1, :) + X(:, 2:end, :);
Y(:, 2:end, :) = Y(:, 2:end, :) + X(:, 1:end-1, :);
y = reshape(Y, m*n, k);
```

Listing 5: Iterazione dei sottospazi seconda versione

```

function [evec, eval] = sottospazi2(m, n, k, itmax)
% function [evec, eval] = sottospazi2(m, m, k, itmax)
% applica itmax iterazioni dei sottospazi alla matrice A per calcolare
% i primi k autovalori e autovettori
% si ferma se la differenza tra gli autovalori in due iterate successive
% e' minore di 1.e-13
% Per calcolare il prodotto matrice vettore usa la funcon prodotto
N = m*n;
x = rand(N,k);
[q r] = qr(x); x=q(:,1:k); % ortogonalizzo x
dd = zeros(k,1);
for it=1:itmax
    y=prodottol(m,n,k,x);
    [q, r] = qr(y); q=q(:,1:k); % fattorizzazione QR
    y=prodottol(m,n,k,q);
    b = q'*y;
    b = (b+b')/2;
    [u, d] = eig(b);
    [ds, s] = sort(diag(d)); % permuto gli autovalori e gli autovettori
    u = u(:,s);
    err = norm(dd-ds,'inf');
    dd = ds;
    disp([it, err])
    if err<1.e-13
        break
    end
    x = q*u;
end
eval = dd;
evec = x;
if err >= 1.e-13
    disp('raggiunto il limite massimo di iterazioni')
end

```

Listing 6: Function per tracciare le linee modali

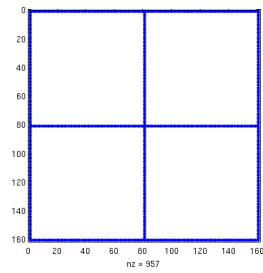
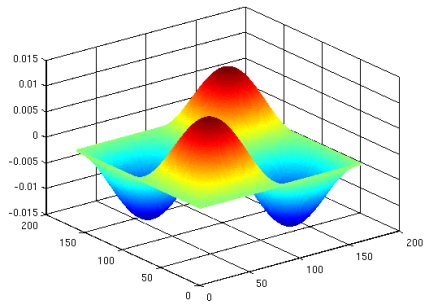
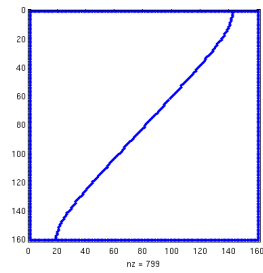
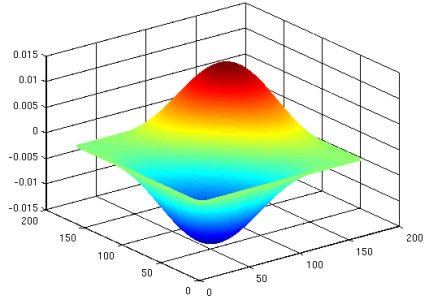
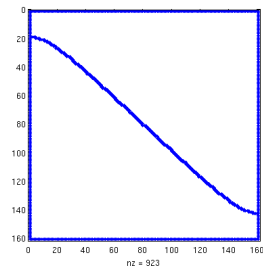
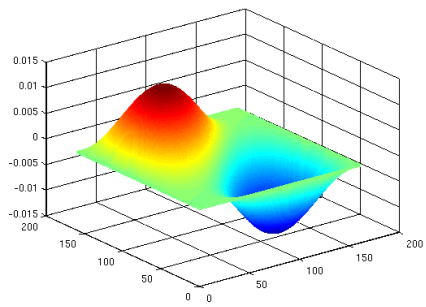
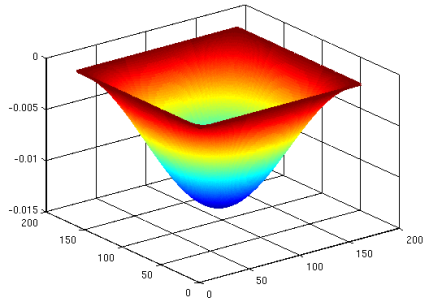
```
function L = traccialinee(v, m, n)
w = reshape(v, n, m);
u = zeros(m+2,n+2);
u(2:m+1,2:n+1) = w;
r = u(1:m+1,:).*u(2:m+2,:); rr=r(1:m+1,1:n+1);
c = u(:,1:n+1).*u(:,2:n+2); cc=c(1:m+1,1:n+1);
L = (rr<=0) + (cc<=0);
spy(L)
```

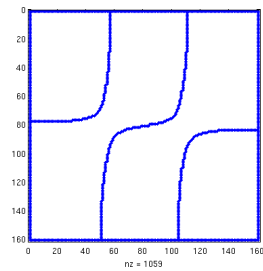
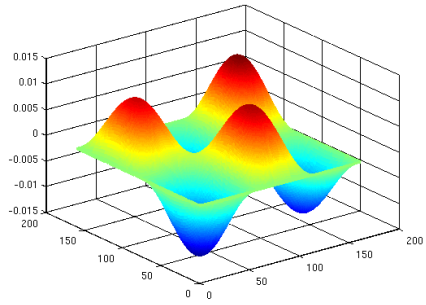
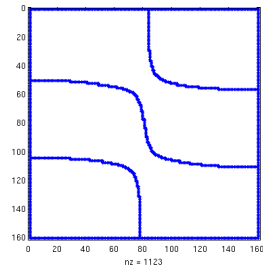
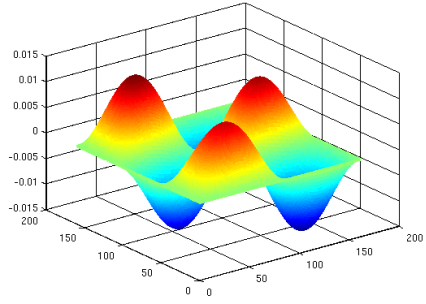
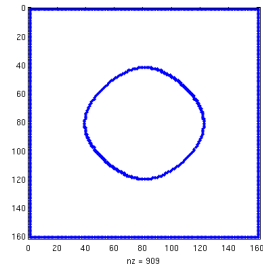
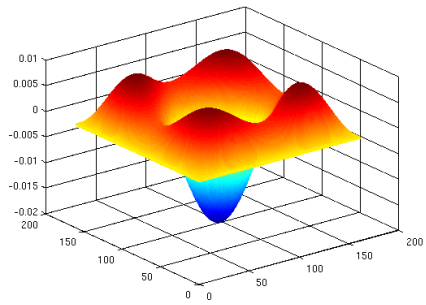
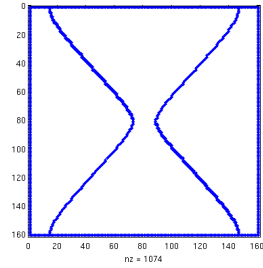
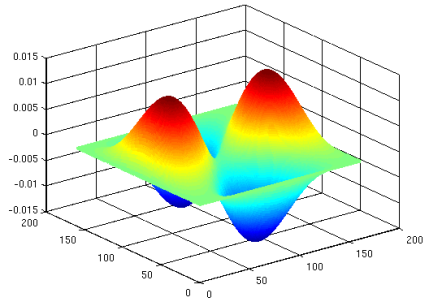
Se il reticolo della discretizzazione è abbastanza fitto possiamo calcolare le linee modali costituite da quei punti della membrana che non oscillano. Per evidenziare questi punti basta scrivere `spy(abs(w)<1.e-4)` dove `w` è l'autovettore considerato. Un modo più preciso per tracciare le linee modali è quello di usare la function `traccialinee` riportata nel listig 6

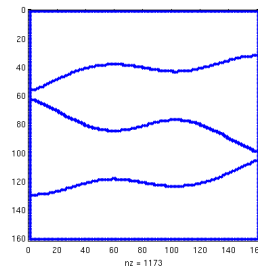
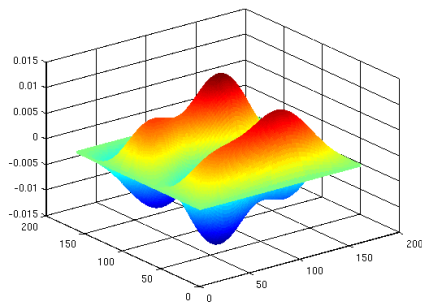
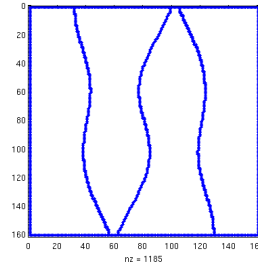
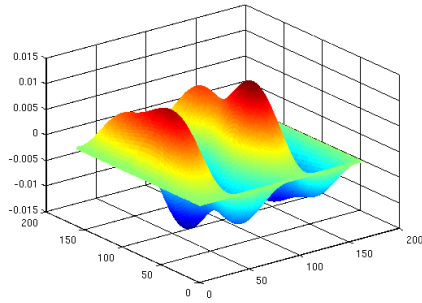
Riportiamo i modi di vibrazione ottenuti con $m = n = 160$ e le corrispondenti linee modali. In questo caso le approssimazioni degli autovalori sono

```
7.6149001e-04
1.9035801e-03
1.9035801e-03
3.0456701e-03
3.8065803e-03
3.8065803e-03
4.9486703e-03
4.9486703e-03
6.4697662e-03
6.4697662e-03
```

Si noti che alcuni autovalori sono doppi, per cui combinando linearmente i due autovettori corrispondenti si ottengono infiniti autovalori e infinite linee modali.







7.3 Altri domini

Per poter implementare il metodo dei sottospazi relativamente ad altri domini si può procedere nel modo seguente. Supponiamo di inserire il nostro dominio dentro un rettangolo che lo include. Introduciamo un reticolo di punti sul rettangolo e per definire il dominio e la sua frontiera costruiamo una matrice che chiamiamo `dominio` in cui l'elemento di posto (i, j) vale 1 se il punto è interno al dominio e vale 0 altrimenti. Ad esempio, la matrice

```

00000000000000000000
01111111111111111110
00111111111111111100
00011111111111111000
00001111111111111000
dominio= 00000111111111110000
00000011111111100000
00000001111111000000
00000000111110000000
00000000011110000000
00000000001110000000
00000000000100000000
00000000000000000000

```

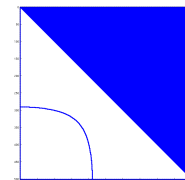
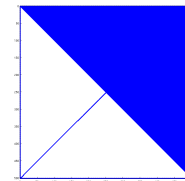
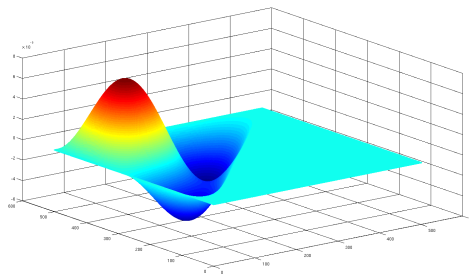
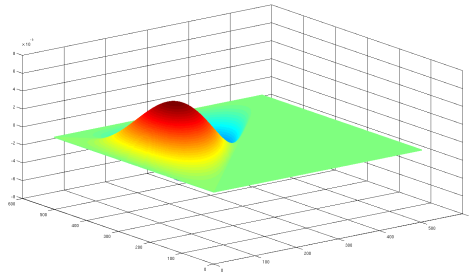
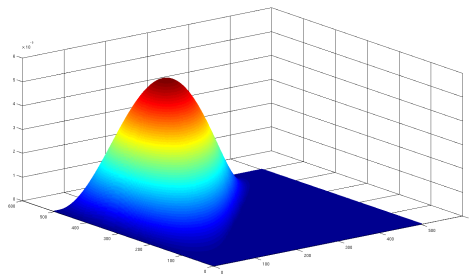
definisce un dominio triangolare. A questo punto, poiché i valori dei punti esterni al dominio in ogni autovettore sono forzati a zero, il programma per il

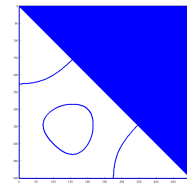
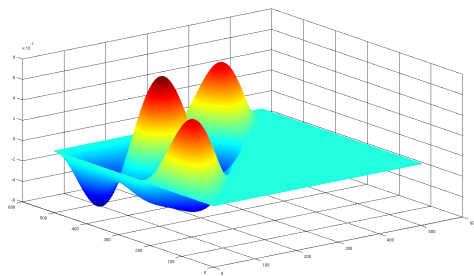
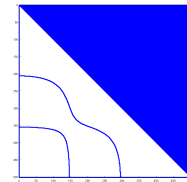
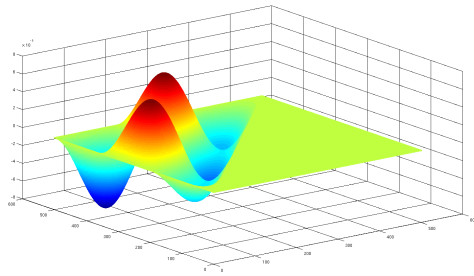
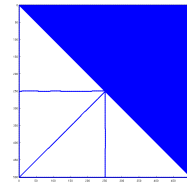
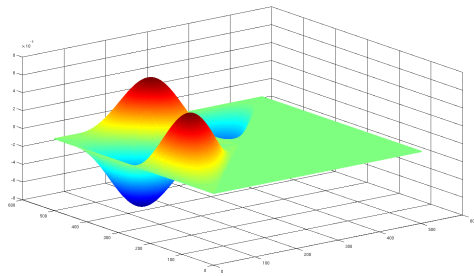
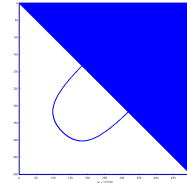
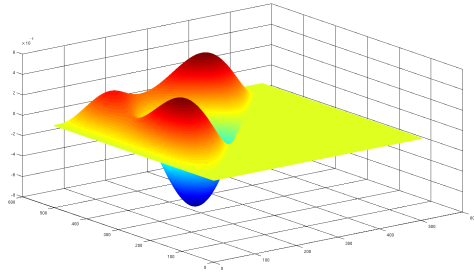
calcolo del prodotto matrice vettore andrà modificato facendo seguire al calcolo di ogni occorrenza del prodotto y di A e di x l'istruzione $y = y .* \text{dominio}$

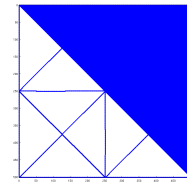
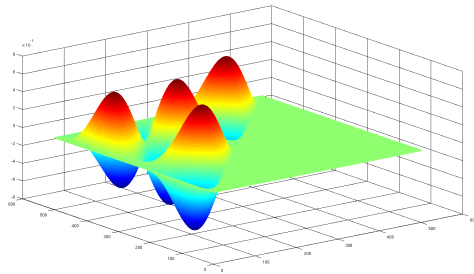
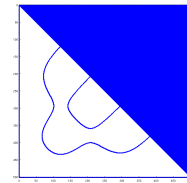
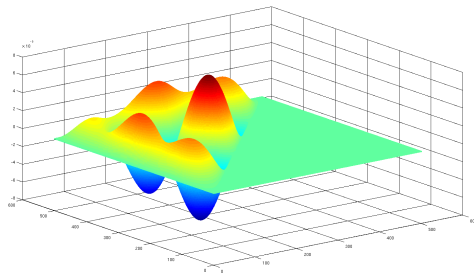
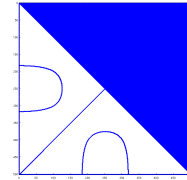
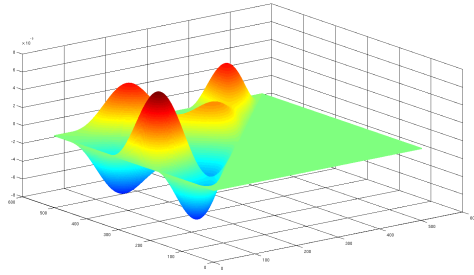
Per brevità non riportiamo i programmi modificati di conseguenza.

Un comando semplice comando di Octave per costruire un dominio a forma di triangolo rettangolo isoscele, metà di un quadrato di lato n è dato da $\text{dominio} = \text{tril}(\text{ones}(n));$

Di seguito si riportano i primi 10 modi di vibrazione ottenuti con $n = m = 500$ nel caso di un dominio dato da un triangolo rettangolo isoscele, assieme alle linee modali. Il tutto è stato calcolato con 1000 passi del metodo di Lanczos descritto nel prossimo paragrafo.







8 Il metodo di Lanczos

Un metodo poco costoso, adatto per calcolare efficientemente gli autovalori estremi dello spettro è il metodo di Lanczos. Esso genera una successione di matrici $k \times k$ tridiagonali T_k i cui autovalori più grandi e più piccoli convergono rapidamente agli autovalori più grandi e più piccoli di A . L'idea è quella di costruire una matrice ortogonale Q che tridiagonalizzi la matrice A . Cioè sia tale che

$$AQ = QT$$

dove T è la matrice tridiagonale simmetrica con elementi diagonali α_i , $i = 1, \dots, n$ e sopra diagonali β_i , $i = 1, 2, \dots, n - 1$. Scrivendo la relazione di sopra

in funzione delle colonne \mathbf{q}_i di Q si ottiene

$$\begin{aligned} A\mathbf{q}_1 &= \alpha_1\mathbf{q}_1 + \beta_1\mathbf{q}_2 \\ A\mathbf{q}_i &= \beta_{i-1}\mathbf{q}_{i-1} + \alpha_i\mathbf{q}_i + \beta_i\mathbf{q}_{i+1}, \quad i = 2, \dots, n-1 \\ A\mathbf{q}_n &= \beta_{n-1}\mathbf{q}_{n-1} + \alpha_n\mathbf{q}_n \end{aligned}$$

Il calcolo dei vettori \mathbf{q}_i e degli scalari α_i, β_i procede sfruttando l'ortogonalità dei \mathbf{q}_i con le seguenti formule

$$\begin{aligned} \alpha_1 &= \mathbf{q}_1^T A \mathbf{q}_1, \quad \beta_1 = \|(A - \alpha_1 I) \mathbf{q}_1\|_2, \quad \mathbf{q}_2 = \frac{1}{\beta_1} (A - \alpha_1 I) \mathbf{q}_1 \\ \alpha_i &= \mathbf{q}_i^T A \mathbf{q}_i, \quad \mathbf{y}_i = (A - \alpha_i I) \mathbf{q}_i - \beta_{i-1} \mathbf{q}_{i-1}, \quad \beta_i = \|\mathbf{y}_i\|_2, \quad \mathbf{q}_{i+1} = \frac{1}{\beta_i} \mathbf{y}_i \end{aligned}$$

assumendo di partire con un vettore \mathbf{q}_1 tale che $\|\mathbf{q}_1\|_2 = 1$. Il metodo presenta problemi di stabilità numerica nel caso in cui si presenta cancellazione (β_i piccolo) nel calcolo di β_i e di \mathbf{q}_{i+1} .

Una implementazione del metodo di Lanczos è riportata nei listings 7.

Per valutare l'errore a posteriori si usa il seguente risultato

Teorema 3 *Sia A una matrice normale $n \times n$, $v \in \mathbb{C}^n$ un vettore non nullo e $\sigma \in \mathbb{C}$. Esiste un autovalore λ di A tale che $|\sigma - \lambda| \leq \|(A - \sigma I)v\|_2 / \|v\|_2$. Se A è invertibile esiste un autovalore λ di A tale che $|(\sigma - \lambda)/\lambda| \leq \|(\sigma A^{-1} - I)v\|_2 / \|v\|_2$.*

Dim. Dimostriamo più in generale che se $f(x)$ è una funzione definita nello spettro di A allora esiste un autovalore λ di A tale che $|f(\lambda)| \leq \|f(A)v\|_2 / \|v\|_2$. Per funzione di matrice $f(A)$ si intende la matrice $f(A) = Qf(D)Q^H$ dove $A = QDQ^H$. La tesi poi segue scegliendo $f(x) = x - \sigma$ e $f(x) = (x - \sigma)/x$. Poiché A è normale esiste Q unitaria tale che $D = Q^H A Q$ è diagonale. Allora vale $f(A) = Qf(D)Q^H$ per cui $\|f(A)v\|_2 = \|f(D)Q^H v\|_2$. Inoltre, poiché $\|Q^H v\|_2 = \|v\|_2$ ne segue che $\|f(A)v\|_2 / \|v\|_2 = \|f(D)w\|_2 / \|w\|_2$, con $w = Q^H v$. Quindi, $\|f(A)v\|_2 / \|v\|_2 = (\sum_{i=1}^n |f(\lambda_i)^2 w_i^2|)^{1/2} / (\sum_{i=1}^n |w_i|^2)^{1/2} \geq \min |f(\lambda_i)|$. \square

Se σ è una approssimazione di un autovalore e v è una approssimazione dell'autovettore corrispondente allora la quantità $\|(A - \sigma I)v\|_2$, presumibilmente piccola, fornisce una stima a posteriori dell'errore di approssimazione. Questo risultato viene applicato dentro la subroutine `lanczos` per selezionare quegli autovalori che hanno raggiunto la precisione voluta.

Riferimenti bibliografici

P. Arbenz, D. Kressner, "Lecture Notes on Solving Large Scale Eigenvalue Problems", <http://people.inf.ethz.ch/arbenz/ewp/Lnotes/>

D. Bini, M. Capovani, O. Menchi, "Metodi Numerici per l'Algebra Lineare", Zanichelli, Bologna.

B. N. Parlett, The Symmetric Eigenvalue Problem, Prentice Hall, Englewood Cliffs, NJ, 1980. (Republished by SIAM, Philadelphia, 1998.).

Listing 7: Function per il calcolo dei modi di vibrazione col metodo di Lanczos

```

function [evec, eval, cont] = lanczos(A, maxit)
% function [evec, eval] = lanczos
% Calcola i modi di vibrazione di una membrana a supporto rettangolare
% col metodo di Lanczos
n = size(A,1);
q1 = rand(n,1)-0.5;
q1 = q1/norm(q1);
aq = A*q1;
a(1) = aq'*q1;
y = aq - a(1)*q1;
b(1) = norm(y);
q2 = y/b(1);
u(:,1) = q1;
for k=2:maxit
    aq = A*q2;
    u(:,k) = q2;
    a(k) = q2'*aq;
    y = aq - a(k)*q2 - b(k-1)*q1;
    b(k) = norm(y);
    q3=y/b(k);
    q1=q2;
    q2=q3;
end
% calcolo autovalori e autovettori
trid = diag(a(1:maxit))+diag(b(1:maxit-1),1)+diag(b(1:maxit-1),-1);
[v, ei] = eig(trid);
ei=diag(ei);
u = u*v;
% seleziono le autocopie approssimate e stimo l'errore
cont=0;
for k=1:maxit
    aq = A*u(:,k);
    aq = aq - ei(k)*u(:,k);
    err=norm(aq);
    if err<1.d-8
        cont=cont+1;
        evec(:,cont)=u(:,k);
        eval(cont)=ei(k);
        disp([cont,err,eig(k)])
    end
end
end

```