

Esercizio Vib-2: Simulazione di un problema di
vibrazione bidimensionale
Corso di LSMC, a.a. 2019-2020

Cristian Soppio
559597

28 gennaio 2022

1 Descrizione del problema

Vogliamo studiare la propagazione di un'onda su una membrana elastica bidimensionale. Modellizziamo la membrana come un sistema di punti $p_{i,j}$, $i, j \leq n$ in \mathbb{R}^3 e supponiamo che il moto avvenga solo lungo una direzione, ad esempio la y .

Assumiamo anche che

1. le forze elastiche delle particelle immediatamente adiacenti, dove la costante elastica k è posta uguale per ogni punto;
2. una forza resistente, dipendente dalla velocità con coefficiente di resistenza θ , anch'esso uguale per ogni particella.

Vogliamo allora studiare questo modello modificandolo opportunamente per

1. ridurre la membrana a un dominio non quadrato, eliminando un sottoinsieme dei punti per simulare l'esistenza di più barriere contro la propagazione dell'onda;
2. simulare il colpo di un martelletto su un punto p fissato, e rappresentare la propagazione nel dominio scelto attraverso un'animazione;
3. studiare l'effetto della barriera sulla propagazione dell'onda rapportando l'energia cinetica del dominio nel caso in è protetto con l'energia nel caso senza protezioni e infine studiare il loro rapporto.

2 Descrizione della sperimentazione

Fissando $n = 40$ proteggiamo la parte di punti tra $(1 : 9, 1 : 40)$ posizionando tre barriere nei punti $(10 : 12, 1 : 10)$, $(10 : 12, 16 : 25)$ e $(10 : 12, 31 : 40)$.

Dopo aver implementato i dati in Matlab con lo Script1, con la function Bidim_movie ho creato l'animazione della membrana, colpita nel punto $P = (32, 20)$ per i primi due secondi di vibrazione per garantire la visione dell'onda.

Infine con la function bidim_energy troviamo come cambia l'energia dal caso protetto al caso non protetto, nei grafici troviamo questo studio e anche del rapporto tra le due energie.

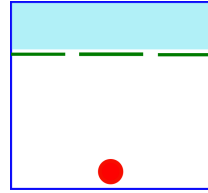


Figura 2.1: dominio

3 Script e function

Si riportano di seguito le function utilizzate nella sperimentazione.

Function 1

```
function bidim_movie(m, k, a, p, v, s, rate, dom)

% costruisco la matrice A
n = size(p,1);
S = 2*eye(n)- diag(ones(n-1,1),1)-diag(ones(n-1,1),-1);
K = kron(eye(n),S)+kron(S,eye(n));
% calcolo la sottomatrice relativa ai punti del dominio
ind = find(dom==1);
in = find(dom==0);
n2 = length(ind);
Kc = K(ind,ind);
A = zeros(2*n2);
A(1:n2,n2+1:2*n2) = eye(n2);
A(n2+1:2*n2,n2+1:2*n2) = -(a/m)*eye(n2);
A(n2+1:2*n2,1:n2) = -(k/m)*Kc;
% formo il vettore delle condizioni iniziali
p=vec(p); p = p(ind);
v=vec(v); v = v(ind);
y = [p;v];
% Simulazione
nt = round(s*rate);
deltat = 1/16000;
B = expm(deltat*A);
% traccio posizione iniziale
U = zeros(n+2,n+2);
```

```

w = zeros(n^2,1); w(ind) = y(1:n2);
w(in)=1;
U(2:n+1,2:n+1) = reshape(w,n,n);
% traccio il grafico della configurazione iniziale
mesh(U)
axis([0,n+2,0,n+2,-10,10])

disp('Premere un tasto per iniziare la simulazione')
% definiamo la variabile globale keep_drawing e poniamola uguale a true
global keep_drawing; keep_drawing = true;
% Generiamo il bottone 'Stop'
btn = uicontrol;
btn.String = 'Stop';
% La function on_btn_pressed e' definita in un file separato
btn.Callback = @on_btn_pressed;
pause
t=0;
while keep_drawing &&t<nt
    t=t+1;
    %for j=1:nt
    y = B*y;
    w = zeros(n^2,1); w(ind) = y(1:n2);
    w(in)=1;
    U(2:n+1,2:n+1) = reshape(w,n,n);
    % traccio il grafico della configurazione
    % all'istante generico
    mesh(U)
    axis([0,n+2,0,n+2,-10,10])
    drawnow;
    pause(0.0001)
end
return
end

```

La Function1 crea l'animazione della membrana elastica che è stata poi registrata e inserita nella sezione successiva. Abbiamo stato aggiunto inoltre un tasto di stop che permette di arrestare l'animazione a nostro piacimento.

Function 2

```

function Ecin=bidim_energy(m, k, a, p, v, s, rate, dom)
% calcola la vibrazione di una membrana elastica
% in un sottoinsieme di una griglia nxn, usando l'esponenziale di matrice
% dato dalla funzione expm, e restituisce l'energia
%della membrana stessa su tutto il sottoinsieme e su una sezione
%predefinita.
%Inoltre, traccia il grafico animato
% In INPUT

```

```

% m: massa di ciascuna particella (tutte masse uguali)
% k: costante elastica (tutte costanti uguali)
% a: costante di attrito (tutte costanti uguali)
% p, v: posizione e velocità iniziale
% s: numero di secondi
% rate: numero di fotogrammi al secondo
% dom: posizione delle barriere non incluse nel dominio (matrice
% nxn con 1 nei punti del dominio, 0 altrove)
% In OUTPUT
% Etot: energia totale della membrana
% Edom: energia della membrana sulla sezione da proteggere.

% costruisco la matrice A
n = size(p,1);
S = 2*eye(n)-diag(ones(n-1,1),1)-diag(ones(n-1,1),-1);
K = kron(eye(n),S)+kron(S,eye(n));
% calcolo la sottomatrice relativa ai punti del dominio
ind = find(dom==1);
in = find(dom==0);
n2 = length(ind);
Kc = K(ind,ind);
A = zeros(2*n2);
A(1:n2,n2+1:2*n2) = eye(n2);
A(n2+1:2*n2,n2+1:2*n2) = -(a/m)*eye(n2);
A(n2+1:2*n2,1:n2) = -(k/m)*Kc;
% formo il vettore delle condizioni iniziali
p=vec(p); p = p(ind);
v=vec(v); v = v(ind);
y = [p;v];
% Simulazione
nt = round(s*rate);
Ecin=0;
deltat = 1/16000;
B = expm(deltat*A);
% traccio posizione iniziale
U = zeros(n+2,n+2);
w = zeros(n^2,1); w(ind) = y(1:n2);
w(in)=1;
U(2:n+1,2:n+1) = reshape(w,n,n);
for j=1:100
y = B*y;
w = zeros(n^2,1);
w(ind) = y(1:n2);
w(in)=1;
U(2:n+1,2:n+1) = reshape(w,n,n);
%calcolo l'energia, cinetica nei punti del dominio,

```

```

%a ogni istante
w=ones(n^2,1);
w(ind)=y(n2+1:2*n2);
ZZ=reshape(w,n,n);
Ecin(j)=0.5*(sum(ZZ(1:10,:), 'all') .^2)*m;

end
end

```

La Function 2 ricava i dati dell'energia cinetica, applicata sia nel caso di protezioni, sia nel caso senza protezioni.

Si riportano ora gli script usati nella sperimentazione.

Script 1

```

n = 40;
m = 20/n^2;
k = 4.0e5;
a = 2.e-2;
p = zeros(n,n);
v = p;
qi = n-8; qj=n/2;
for i=-4:4
for j=-4:4
v(qi+i,qj+j) = 1.e4*exp(-0.07*(i^2+j^2));
end
end

dom = ones(n,n);
dom(10:12,1:12)=0% barriera a sinistra
dom(10:12,16:24)=0; %barriera centrale
dom(10:12,28:40)=0; % barriera a destra

Dom = ones(n,n);
Dom(10:12,1:15)=0; % barriera a sinistra
Dom(10:12,25:40)=0; % barriera a destra

s = 2;
rate = 16384;

```

Tale script1 crea il dominio nel caso di due barriere di protezione sia nel caso di sole due barriere (senza quella centrale).

4 Animazioni, grafici, tabelle e commenti

Per l'animazione:

Clicca qui sopra

Abbiamo registrato un'animazione di 15 secondi che ci mostra che comunque la barriera non protegge molto il dominio, infatti dopo pochi secondi il dominio inizia ad avere un'energia cinetica significativa.

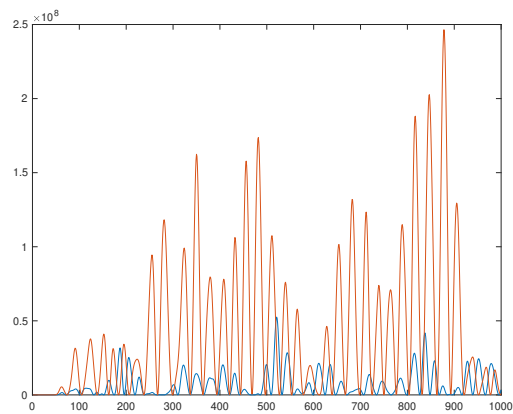


Figura 4.1: Grafico dell'energia cinetica con e senza protezione

In figura 4.1 abbiamo in blu i dati dell'energia del dominio con una barriera in più di protezione e in rosso i dati dell'energia senza una terza barriera. Possiamo osservare che l'uso di una terza barriera oltre a rallentare l'inizio delle vibrazioni, i valori rimangono notevolmente più bassi dell'energia.

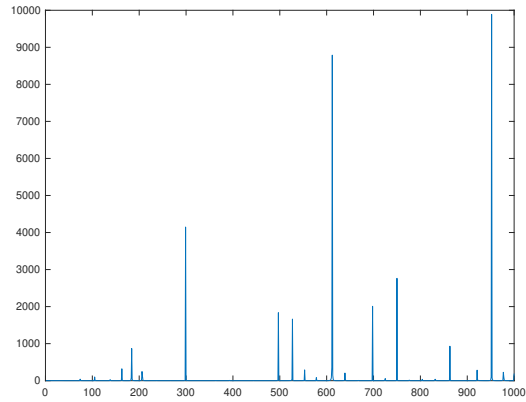


Figura 4.2: Grafico del rapporto dell'energie cinetiche

In figura 4.2 osserviamo il rapporto di

$$E_{con}/E_{senza}$$

nei punti del dominio e vediamo che le barriere proteggono molto tanto da far stare il rapporto molto vicino a zero, tranne in alcuni picchi che comunque abbiamo una bassa energia cinetica del dominio nel caso senza barriere.

Infine diamo i valori medi dell'energia con e senza protezioni

$$\frac{E_{con}}{E_{senza}} \left| \begin{array}{l} 6.869366437019439 \times 10^6 \\ 4.185492771983603 \times 10^7 \end{array} \right.$$

che mostrano comunque come l'energia è costantemente più bassa.