

機械振動論

固有振動と振動モード

本事例では、板バネを解析対象として、数値計算（シミュレーション）と固有値問題を解くことにより振動解析を行っています。

実際の振動は振動モードと呼ばれる特定パターンが複数組み合わせられますが、各振動モードによる振動に分けて解析を行うことでその現象を捉え易くすることが出来ます。そこで、本事例では、アニメーションを活用した解析結果の可視化も取り入れています。

▼ 板バネの振動

このセクションでは、板バネの振動シミュレーションを行っています。

Maple カーネルを初期化します：

```
[> restart:
```

板バネを 30 分割してモデル化します：

```
[> n:=30:
```

分割したそれぞれの板に関して運動方程式を生成します。
一方の端は固定しないため最後の板を除き同じパターンの運動方程式となります。

$x[j]$ は各板の高さを表し、一階微分は速度、二階微分は加速度となります。

for 文により、各板について運動方程式を生成します：

```
[> for j to n-1 do  
    eq[j]:= m*diff(x[j](t),t$2)+k*(x[j](t)-x[j-1](t))-k*(x  
    [j+1](t)-x[j](t));  
end:
```

端(30番目)の板は隣がないので別に定義します：

```
[> eq[30]:=m*diff(x[30](t),t$2)+k*(x[30](t)-x[29](t));  

$$eq_{30} := 500 \left( \frac{d^2}{dt^2} x_{30}(t) \right) + 1.0 \cdot 10^7 x_{30}(t) - 1.0 \cdot 10^7 x_{29}(t) \quad (1.1)$$

```

方程式の集合を生成します：

```
[> eq:= {seq(eq[j],j=1..n)}:
```

各板の質量 m とバネ定数 k を定義します：

・ 各板の質量 m

```
[> m:=500:
```

・ バネ定数 k

```
[> k:=1.0e7:
```

板バネの元は固定にします。x[0]は常に0です:

```
[ > x[0]:=t->0;
                                      $x_0 := t \mapsto 0$ 
]
```

(1.2)

初期値の設定をします。

以降 for 文処理のため、空集合を生成しておきます:

```
[ > init:={};
```

30番目(端)以外については、初期高さおよび初速は全て 0 にします。
初期高さを x[j]、初速を D(x[j]) としています。

for 文により上記初期条件を定義します:

```
[ > for j to n-1 do
    init:=init union {x[j](0)=0,D(x[j])(0)=0};
end;
```

30番目の板については、初期高さは 0、初速は 0.1m/s (単位系MKS) とします。
上記初期条件の定義および x1~x29 番目の板バネの初期条件を1つの集合にします:

```
[ > init:=init union {x[30](0)=0,D(x[30])(0)=0.1};
```

サンプリングの個数 mm とシミュレーション時間 TT、出力の刻み幅 d を設定します。

・サンプリングの個数 mm

```
[ > mm:=50;
```

・シミュレーション時間 TT

```
[ > TT:=1;
```

・出力の刻み幅 d

```
[ > d:=TT/mm;
```

$$d := \frac{1}{50}$$

(1.3)

サンプリングのセットを Array 型として生成します:

```
[ > samp:=Array([seq(j*d,j=0..mm)]);
```

```

                                     1 .. 51 Array
                                     Data Type: anything
                                     Storage: rectangular
                                     Order: Fortran_order

```

(1.4)

運動方程式の連立微分方程式系について、数値解を求めます。

ここでは、dsolve コマンドに、output=samp のオプションを指定することで、各サンプリング点における独立変数の数値解を求めます：

```
> ans:=dsolve(equ union init,{seq(x[j](t),j=1..n)},numeric,
output=samp);
```

$$ans := \left[\begin{array}{l} \left[\begin{array}{l} 1 .. 61 \text{ Array} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{array} \right] \\ \\ \left[\begin{array}{l} 51 \times 61 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{array} \right] \end{array} \right] \quad (1.5)$$

次の各独立変数の数値解が求められます：

```
> seq(ans[1,1][j],j=1..2*n+1);
```

$$t, x_1(t), \frac{d}{dt} x_1(t), x_2(t), \frac{d}{dt} x_2(t), x_3(t), \frac{d}{dt} x_3(t), x_4(t), \frac{d}{dt} x_4(t), x_5(t), \frac{d}{dt} x_5(t), \quad (1.6)$$

$$x_6(t), \frac{d}{dt} x_6(t), x_7(t), \frac{d}{dt} x_7(t), x_8(t), \frac{d}{dt} x_8(t), x_9(t), \frac{d}{dt} x_9(t), x_{10}(t),$$

$$\frac{d}{dt} x_{10}(t), x_{11}(t), \frac{d}{dt} x_{11}(t), x_{12}(t), \frac{d}{dt} x_{12}(t), x_{13}(t), \frac{d}{dt} x_{13}(t), x_{14}(t),$$

$$\frac{d}{dt} x_{14}(t), x_{15}(t), \frac{d}{dt} x_{15}(t), x_{16}(t), \frac{d}{dt} x_{16}(t), x_{17}(t), \frac{d}{dt} x_{17}(t), x_{18}(t),$$

$$\frac{d}{dt} x_{18}(t), x_{19}(t), \frac{d}{dt} x_{19}(t), x_{20}(t), \frac{d}{dt} x_{20}(t), x_{21}(t), \frac{d}{dt} x_{21}(t), x_{22}(t),$$

$$\frac{d}{dt} x_{22}(t), x_{23}(t), \frac{d}{dt} x_{23}(t), x_{24}(t), \frac{d}{dt} x_{24}(t), x_{25}(t), \frac{d}{dt} x_{25}(t), x_{26}(t),$$

$$\frac{d}{dt} x_{26}(t), x_{27}(t), \frac{d}{dt} x_{27}(t), x_{28}(t), \frac{d}{dt} x_{28}(t), x_{29}(t), \frac{d}{dt} x_{29}(t), x_{30}(t),$$

$$\frac{d}{dt} x_{30}(t)$$

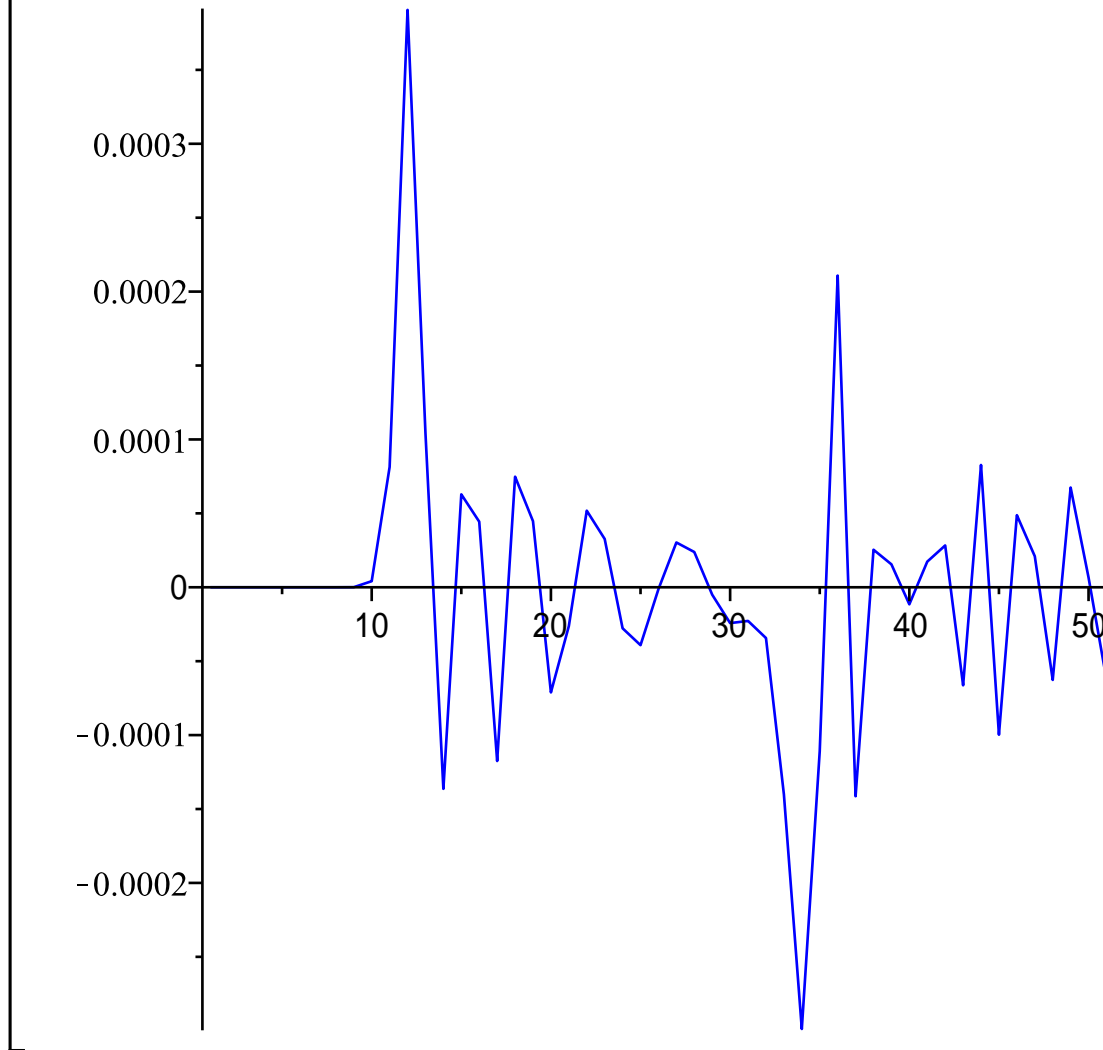
結果をプロットします。

プロットには、plots および plottools パッケージ内のコマンドを用いるため、2つのパッケージをロードします：

```
[> with(plots):with(plottools):
```

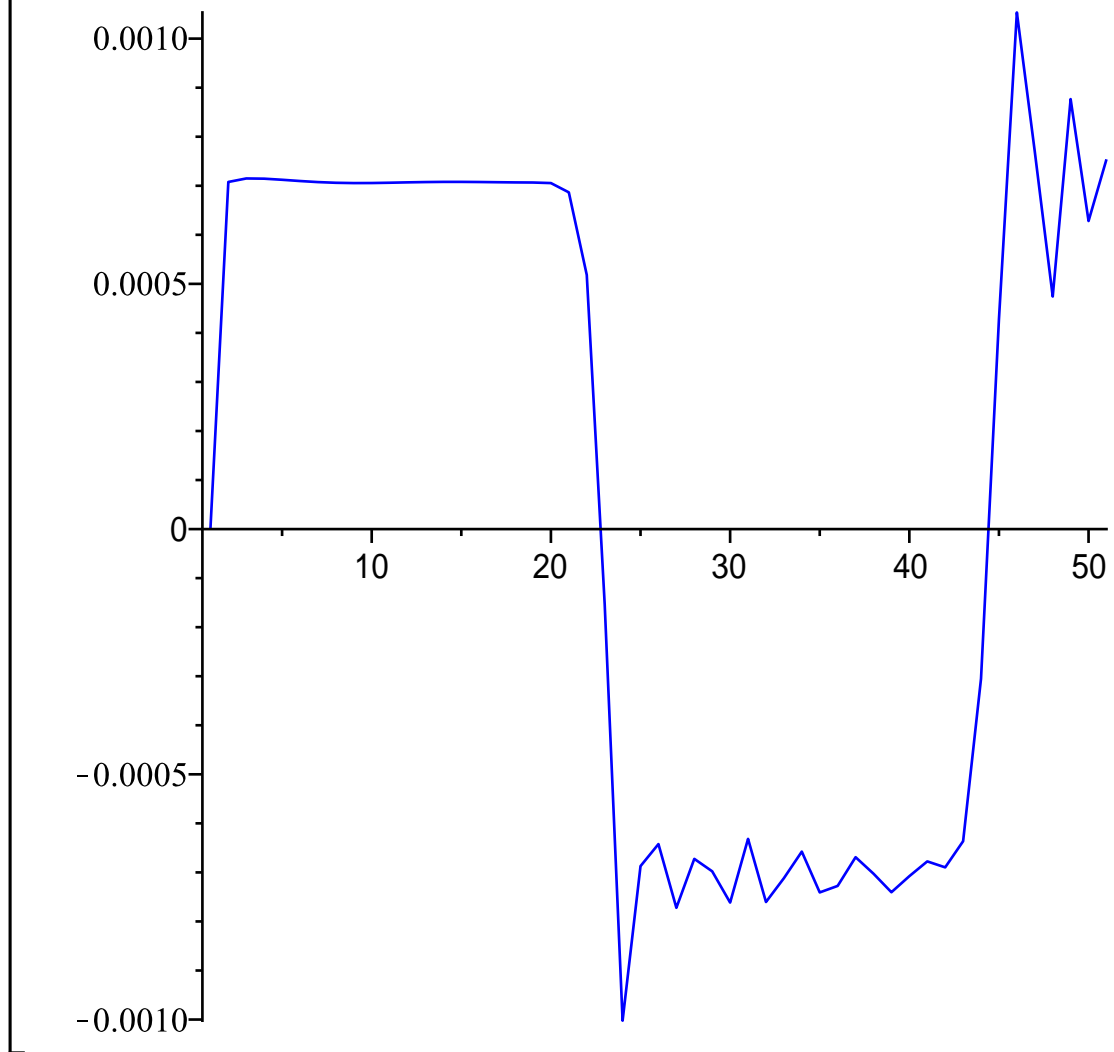
解(ans)の二番目は x[1] の時間に対する変化です:

```
> listplot([seq(ans[2,1][j,2],j=1..mm+1)],color=blue);
```



60番目は x[30] すなわち一番端(フリー)の時間に対する変化です :

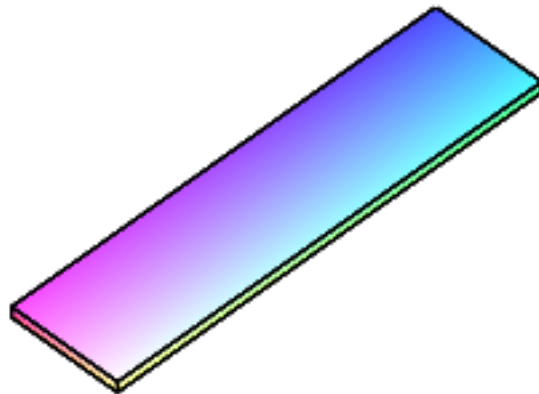
```
> listplot([seq(ans[2,1][j,60],j=1..mm+1)],color=blue);
```



板全体の動きをアニメーションとして再現します。

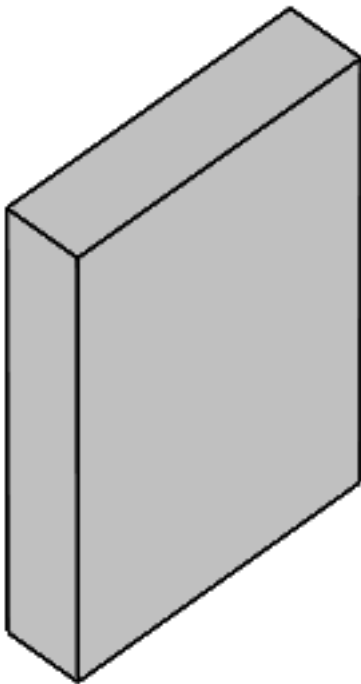
各板のオブジェクトを直方体で用意します:

```
> cb:=cuboid([0,0,0],[4,1,0.1]);  
> display(cb,scaling=constrained);
```



壁のオブジェクトを直方体で用意します:

```
> kabe:=cuboid([0,-1,-3],[4,0,3],color=gray):  
> display(kabe,scaling=constrained);
```



フレームを保存する変数 p を初期化します:


```
[> p:=[]:
```

時間変化の各フレームを作成します。変位は強調(1500倍)してあります:

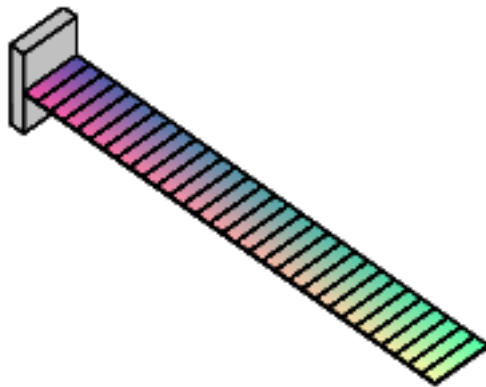
```
[> for j to mm+1 do  
  pp:=[kabe,cb]:  
  for i to n do  
    pp:=pp,translate(cb,0,i,1500*ans[2,1][j,2*i]):  
  end:  
  p:=p,display(pp):  
end:
```

アニメーション表示を行います:

アニメーションの再生には、グラフを選択し、コンテキストバーにある再生ボタン

 現在のフレーム をクリックします:

```
[> display(p,insequence=true,scaling=constrained);
```



固有値問題

この板バネの系の固有値問題を解き固有モードと振動スペクトラムを求めます。

これらの 30 個の運動方程式は行列形式で表現することができます。

$$[M] \cdot \{x''\} + [K] \cdot \{x\} = \{f\}$$

M を質量行列、K を剛性行列と呼びます。x を変位ベクトル、f を力ベクトルと呼びます。

このときすべての自由度が同位相で振動すると仮定すれば、以下の行列方程式となります。

$$\Omega^2 \cdot [M] \cdot X - [K] \cdot X = 0 \quad ([K] \cdot X = \lambda \cdot [M] \cdot X)$$

これを一般固有値問題($A \cdot x = \lambda \cdot B \cdot x$)と呼んでいます。ここで Ω は固有角振動数です。

[M] の逆行列を両辺にかければ、通常の標準固有値問題($A \cdot x = \lambda \cdot x$)となります。

固有値問題を解く為に線形代数パッケージを呼び出します：

```
[> with(LinearAlgebra):
```

剛体行列を生成します：

対角が質量の単純な対角行列です。

・出力結果の上でマウス左ボタンをダブルクリックすると行列を確認することができます

```
[> M:=DiagonalMatrix([seq(m,j=1..n)]);
```

$$M := \begin{bmatrix} 30 \times 30 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{bmatrix}$$

(2.1)

剛性行列を生成します。

対角要素を定義しておきます：

```
[> K:=DiagonalMatrix([seq(2*k,j=1..n)]):
```

上側と下側の subdiagonal (対角の1つ上、1つ下)を設定します：

```
> for j to n-1 do
  K[j,j+1]:=-k:
  K[j+1,j]:=-k:
end:
> K;
```

$$\left[\begin{array}{l} 30 \times 30 \text{ Matrix} \\ \text{Data Type: anything} \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{array} \right] \quad (2.2)$$

行列 K と M を与えて固有ベクトル(P)と固有値(W)を関数 Eigenvectors で求めます：

```
> (P,W):=Eigenvectors(K,M,output=['vectors','values']);
```

$$P, W := \left[\begin{array}{l} 30 \times 30 \text{ Matrix} \\ \text{Data Type: complex}_8 \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{array} \right], \left[\begin{array}{l} 1 \dots 30 \text{ Vector}_{\text{column}} \\ \text{Data Type: complex}_8 \\ \text{Storage: rectangular} \\ \text{Order: Fortran_order} \end{array} \right] \quad (2.3)$$

求められた固有値を確認します：

```
> seq([j,W[j]],j=1..n);
```

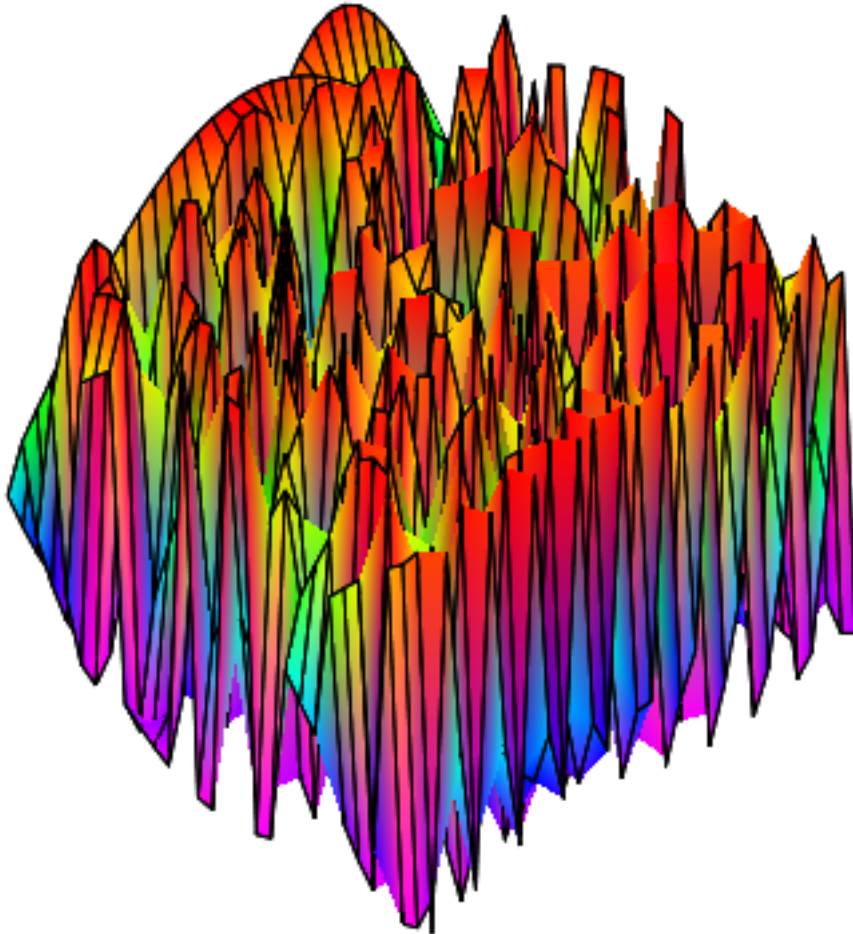
```
[1, 818.802349900224 + 0. I], [2, 205.227064324194 + 0. I], [3, 1834.42974399804 + 0. I], [4, 3241.68753519078 + 0. I], [5, 5026.13535421671 + 0. I], [6, 7169.46235170895 + 0. I], [7, 9649.67509228837 + 0. I], [8, 12441.3232369725 + 0. I], [9, 15515.7606980935 + 0. I], [10, 18841.4395869215 + 0. I], [11, 22384.2339376946 + 0. I], [12, 26107.7898862072 + 0. I], [13, 29973.8987096512 + 0. I], [14, 33942.8888998169 + 0. I], [15, 37974.0332464515 + 0. I], [16, 42025.9667535484 + 0. I], [17, 46057.1111001830 + 0. I], [18, 50026.1012903488 + 0. I], [19, 53892.2101137928 + 0. I], [20, 57974.7729356757 + 0. I], [21, 59181.1976500997 + 0. I], [22, 78165.5702560021 + 0. I], [23, 76758.3124648094 + 0. I], [24, 74973.8646457833 + 0. I], [25, 72830.5376482911 + 0. I], [26, 70350.3249077117 + 0. I], [27, 67558.6767630274 + 0. I], [28, 64484.2393019065 + 0. I], [29, 57615.7660623054 + 0. I], [30, 61158.5604130785 + 0. I]
```

複素数になっていますが、虚部はすべて 0 です。

実部を取り、固有ベクトル(実部)をプロットします：

・この行列は固有モード行列あるいはモード行列と呼ばれます

```
> matrixplot(map(Re,P),shading=zhue);
```

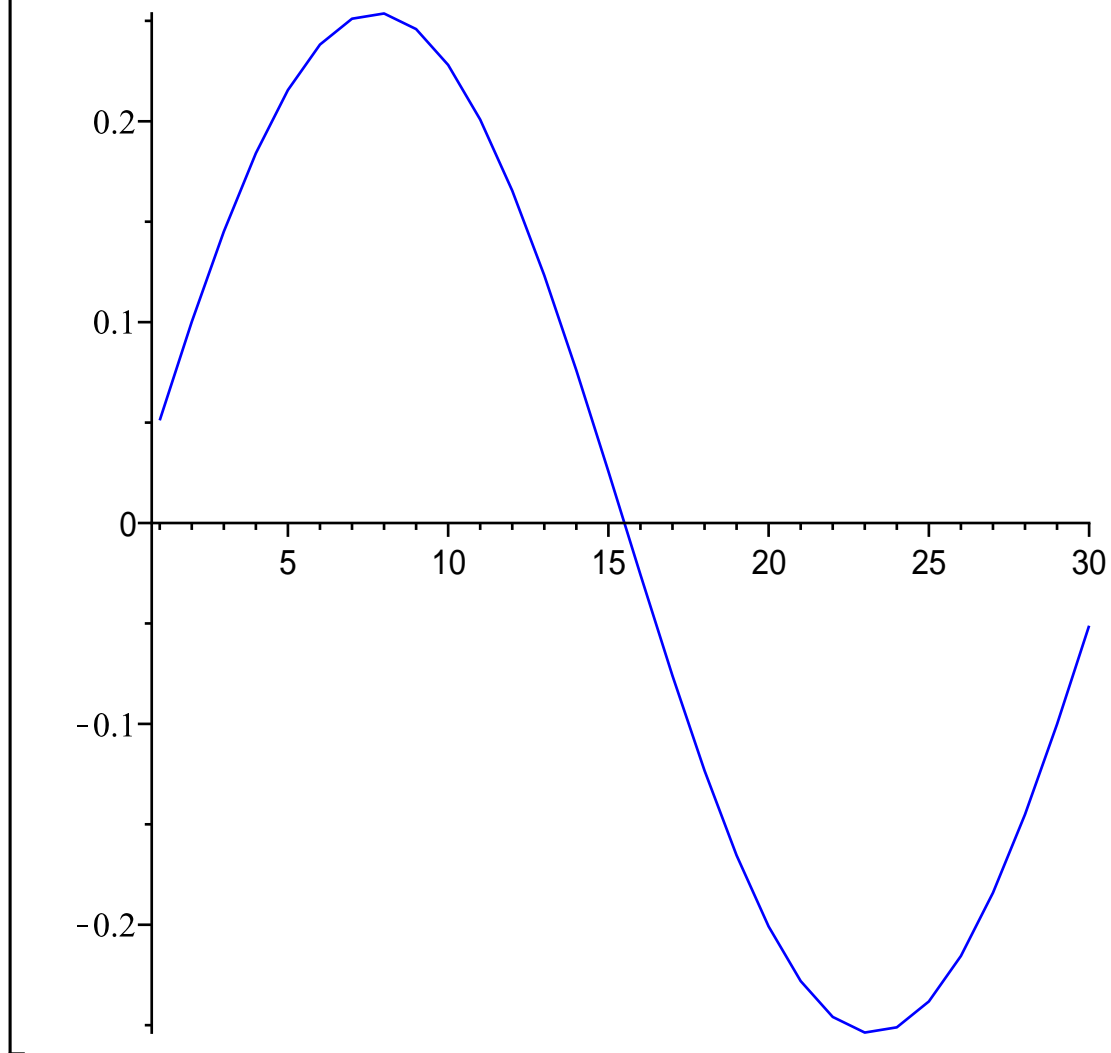


周波数がだんだん高くなっています。

最初の振動モード(最も低周波)をプロットします:

・このモードは一次モードあるいは基本モードとも呼ばれます

```
> listplot([seq(Re(P[j,1]),j=1..n)],color=blue);
```



周波数変化のアニメーションを表示します。

for 文処理のために、フレームを保存する変数を空にしておきます：

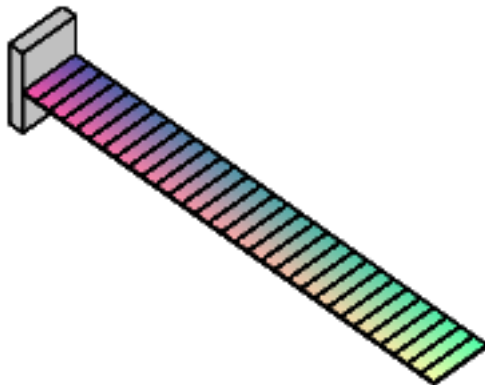
```
[> p:=[]]:
```

時間変化の各フレームを生成します:

```
[> for j from 0 to mm do  
  pp:=[kabe,cb]:  
  for i to n do  
    pp:=pp,translate(cb,0,i,10*Re(P[i,1])*sin(2*Pi*j/mm)):  
  end:  
  p:=p,display(pp):  
end:
```

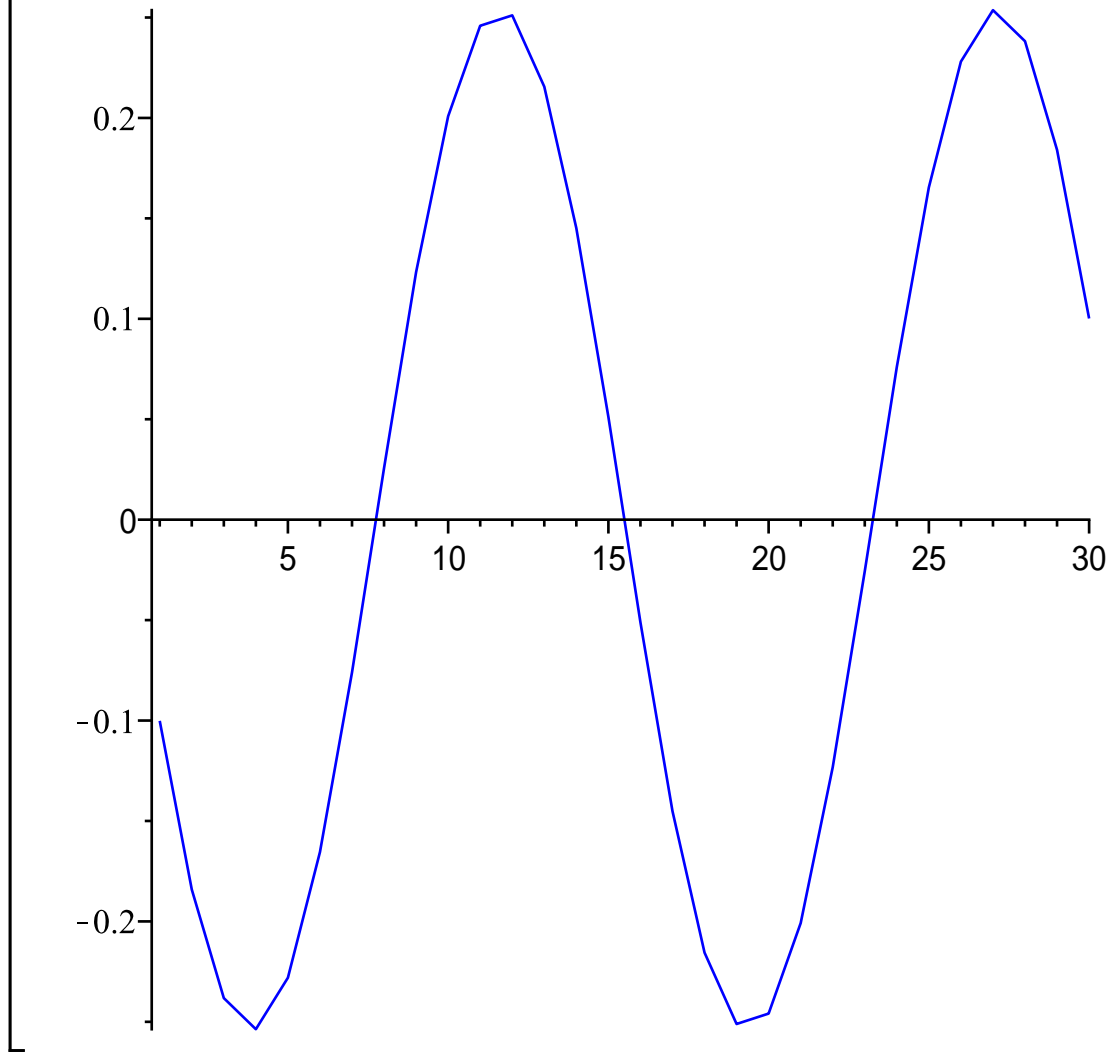
アニメーションを表示します：

```
[> display(p,insequence=true,scaling=constrained);
```



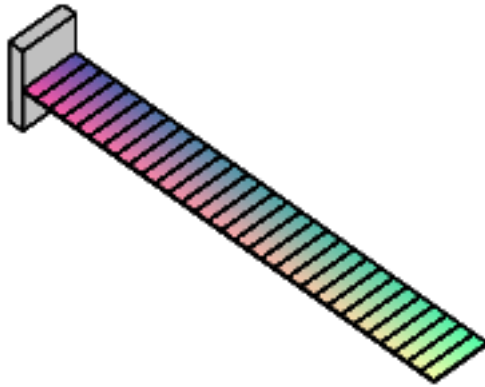
2番目の振動モードをプロットします：

```
> listplot([seq(Re(P[j,4]),j=1..n)],color=blue);
```



アニメーションを表示します:

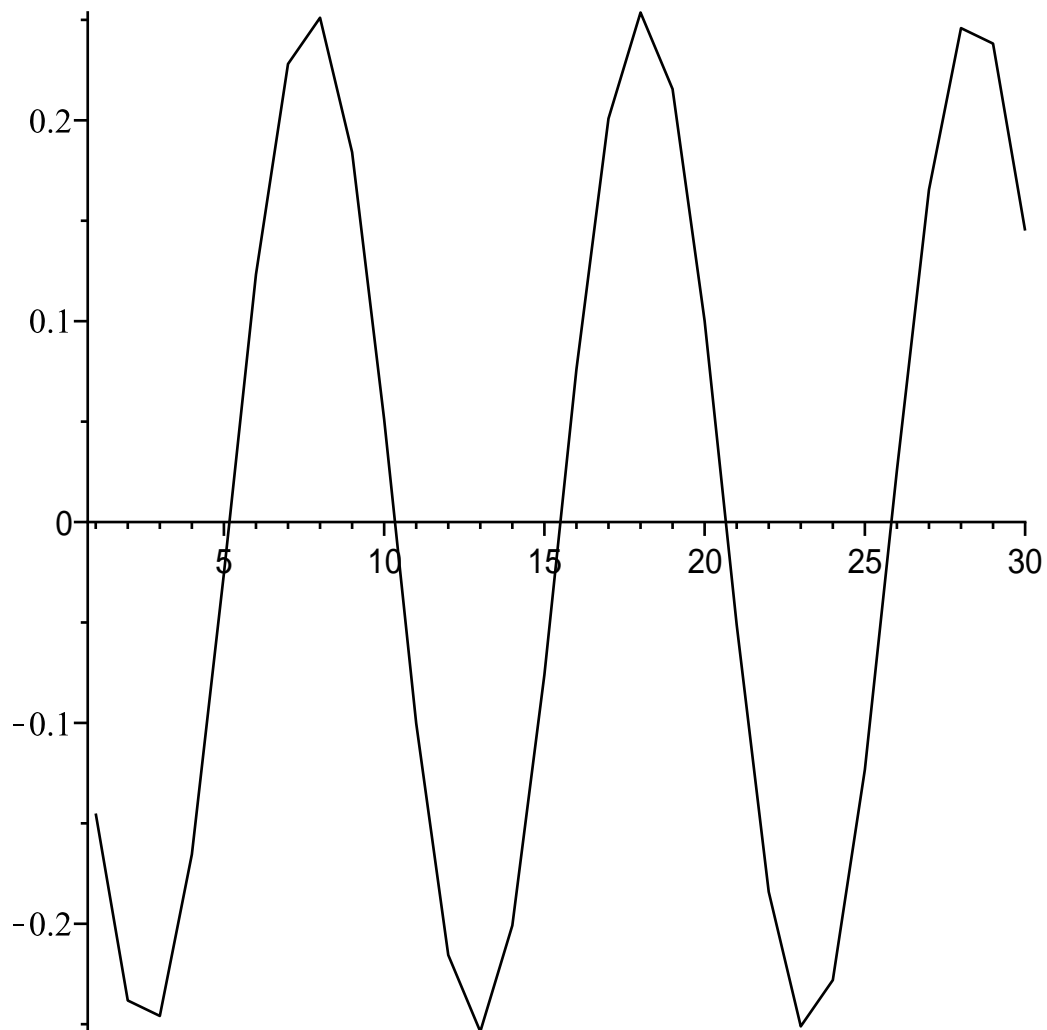
```
> p:=[]:  
> for j from 0 to mm do  
  pp:=[kabe,cb]:  
  for i to n do  
    pp:=pp,translate(cb,0,i,10*Re(P[i,4])*sin(2*Pi*j/mm)):  
  end:  
  p:=p,display(pp):  
end:  
> display(p,insequence=true,scaling=constrained);
```



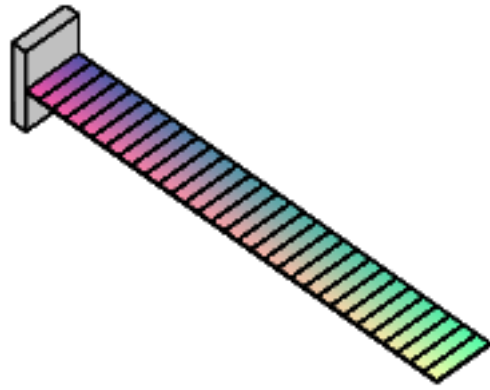
同様な方法で、3番目、4番目、5番目の振動モードのアニメーションを生成します。

3番目の振動モード：

```
> listplot([seq(Re(P[j,6]),j=1..n)]);
```

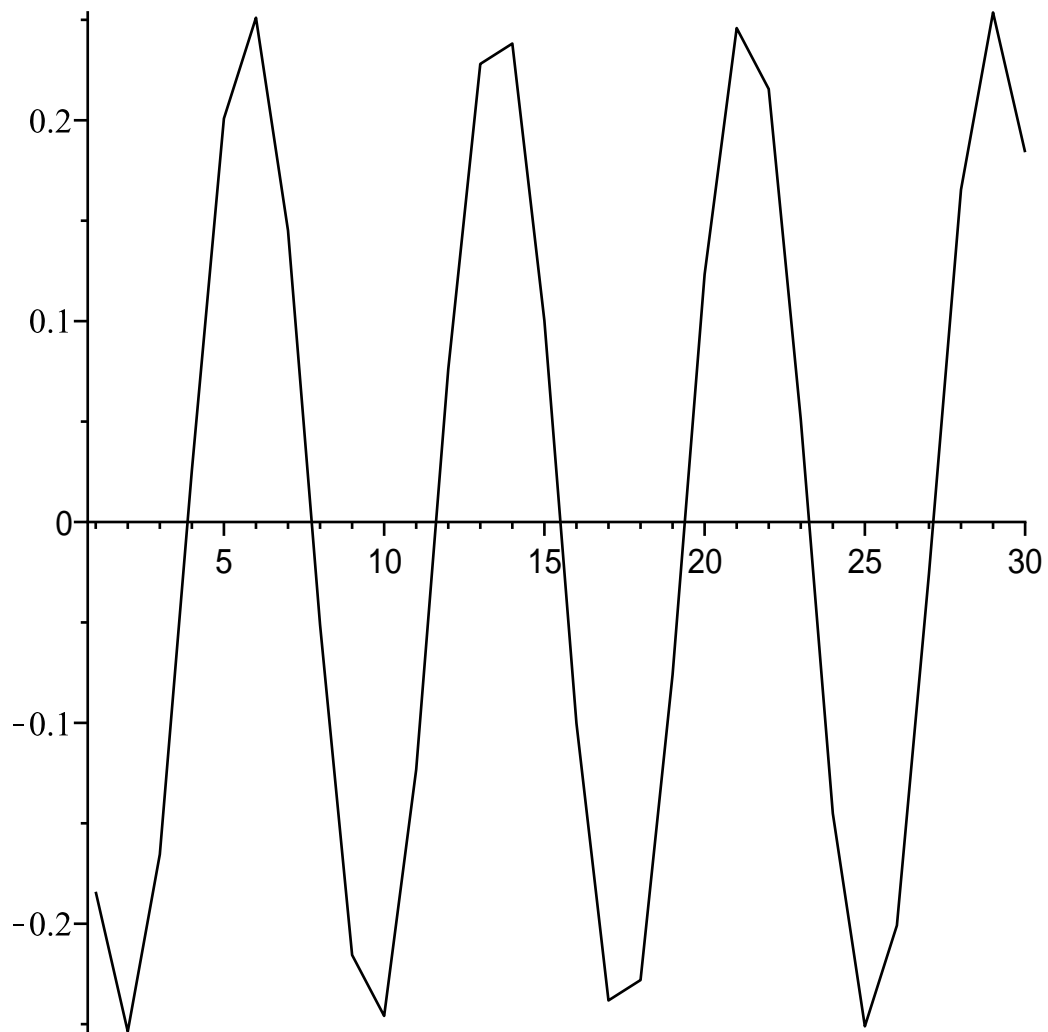


```
> p:=[]:  
> for j from 0 to mm do  
  pp:=[kabe,cb]:  
  for i to n do  
    pp:=pp,translate(cb,0,i,6*Re(P[i,6])*sin(2*Pi*j/mm)):  
  end:  
  p:=p,display(pp):  
end:  
> display(p,insequence=true,scaling=constrained);
```

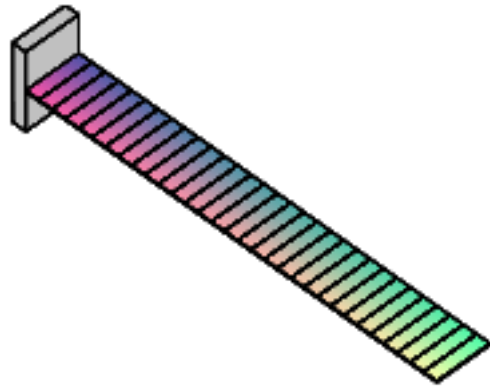



4番目の振動モード:

```
> listplot([seq(Re(P[j,8]),j=1..n)]);
```

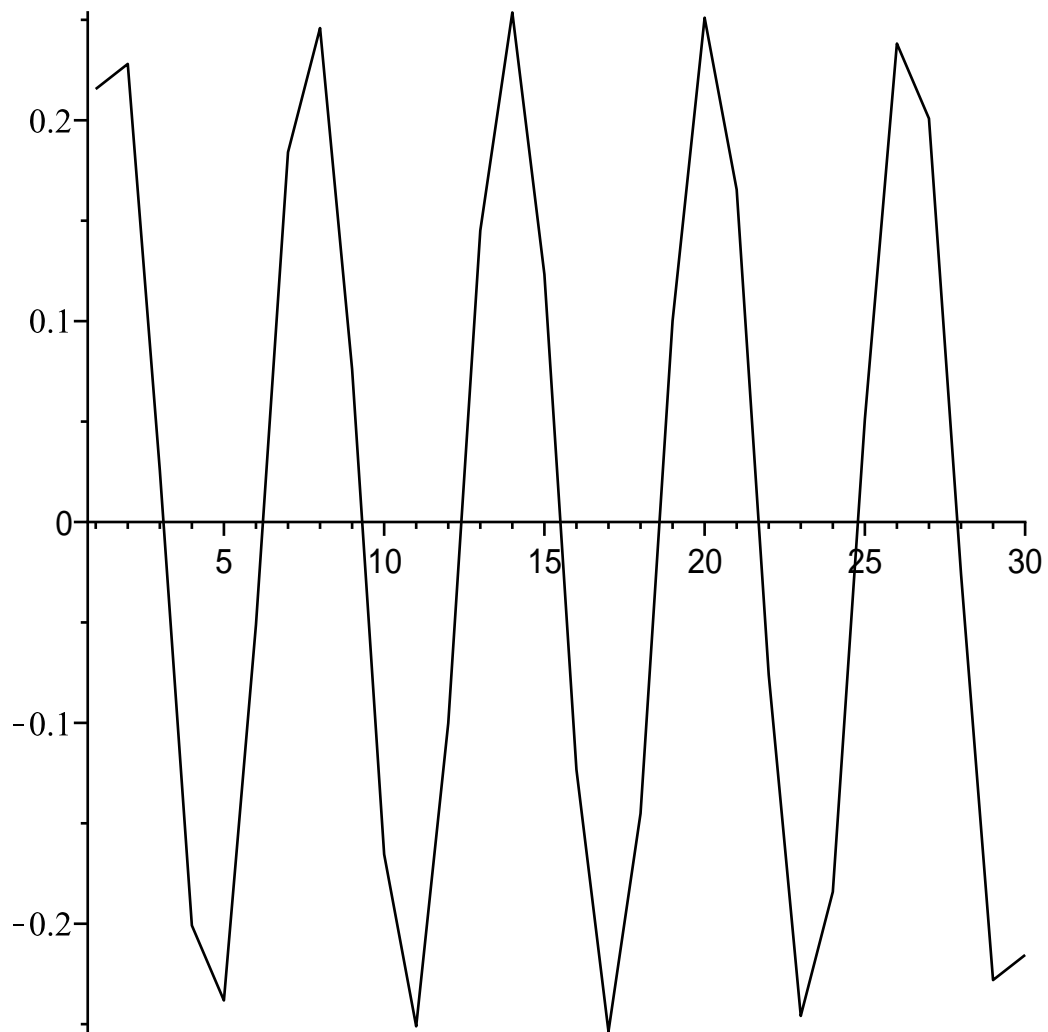


```
> p:=[]:  
> for j from 0 to mm do  
  pp:=[kabe,cb]:  
  for i to n do  
    pp:=pp,translate(cb,0,i,3*Re(P[i,8])*sin(2*Pi*j/mm)):  
  end:  
  p:=p,display(pp):  
end:  
> display(p,insequence=true,scaling=constrained);
```

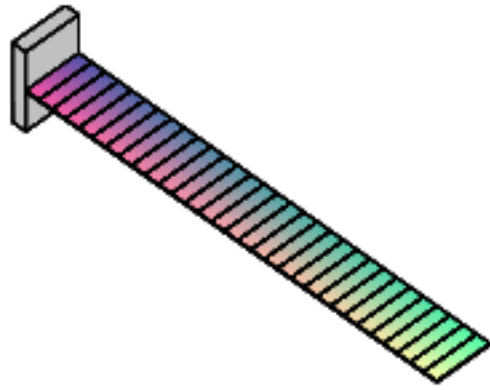


5番目の振動モード:

```
> listplot([seq(Re(P[j,10]),j=1..n)]);
```

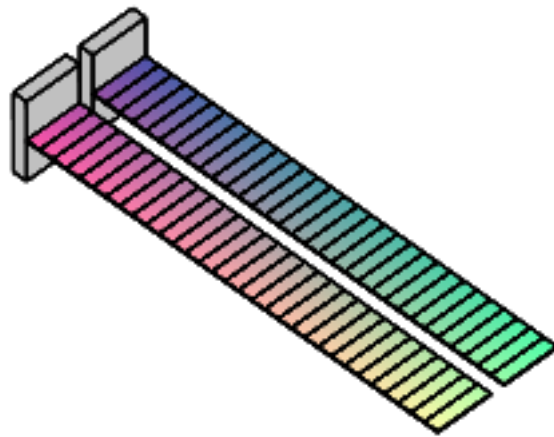


```
> p:=[];  
> for j from 0 to mm do  
  pp:=[kabe,cb]:  
  for i to n do  
    pp:=pp,translate(cb,0,i,3*Re(P[i,10])*sin(2*Pi*j/mm));  
  end:  
  p:=p,display(pp):  
end:  
> display(p,insequence=true,scaling=constrained);
```



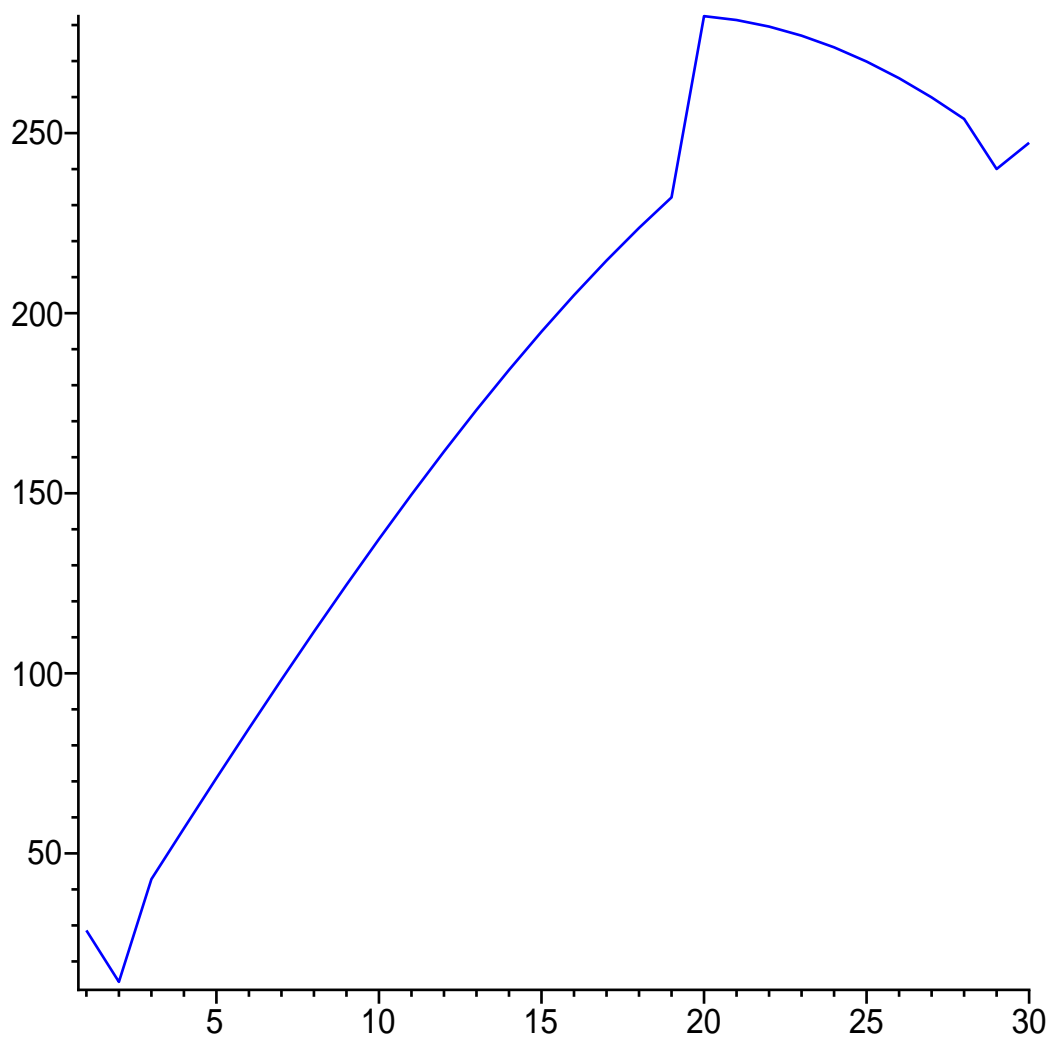
2番目と3番目の振動モードについて、同時にアニメーションを表示します：

```
> p:=[]:  
> for j from 0 to mm do  
  pp:=[kabe,cb]:  
  pp:=pp,translate(kabe,5,0,0):  
  pp:=pp,translate(cb,5,0,0):  
  for i to n do  
    pp:=pp,translate(cb,0,i,6*Re(P[i,4])*sin(2*Pi*j/mm)):  
    pp:=pp,translate(cb,5,i,6*Re(P[i,6])*sin(2*Pi*j/mm)):  
  end:  
  p:=p,display(pp):  
end:  
> display(p,insequence=true,scaling=constrained);
```



各モードの固有角振動(スペクトラム： Ω)をプロットします：

```
> listplot([seq(sqrt(Re(W[j])),j=1..n)],color=blue);
```



Copyright © 2011 Cybernet Systems Co., Ltd. All rights reserved