ルンゲ・クッタ法をつかう

演習課題 2 (6 月 13 日・14 日分)

距離 a だけ離れた剛体壁の間に質点を直線上に N 個配置し, それらを同じ長さ, 同じバネ定数 k のバネ $N+1$ 個をつかって図 1 のように連結する。j 番目の質点の質量を m_j とし, その位置は左の壁から x_j の距離にあるとする ($j = 1, 2, \cdots, N$)。静止状態 (安定状態) における質点の位置は

$$\overline{x}_j = ja/(N+1) \quad (j = 1, 2, \cdots, N) \tag{1}$$

である。いま質点の位置を

$$x_j(t) = \overline{x}_j + u_j(t) \quad (j = 1, 2, \cdots, N) \tag{2}$$

とおいて, あらためて変位 u_j を変数とみなすと, 運動方程式は

$$\begin{cases} m_1 \dfrac{\mathrm{d}^2 u_1}{\mathrm{d}t^2} = -k(2u_1 - u_2), \qquad m_N \dfrac{\mathrm{d}^2 u_N}{\mathrm{d}t^2} = -k(2u_N - u_{N-1}), \\ m_j \dfrac{\mathrm{d}^2 u_j}{\mathrm{d}t^2} = -k(2u_j - u_{j-1} - u_{j+1}) \quad (2 \le j \le N-1), \end{cases} \tag{3}$$

となる。長さの単位を a, 時間の単位を $\sqrt{m_1/k}$ にとって変数を無次元化すると, 静止状態での j 番目の質点の位置は $\overline{x}_j = j/(N+1)$ で, 方程式 (3) は

$$\begin{cases} \dfrac{\mathrm{d}^2 u_1}{\mathrm{d}t^2} = -c_1(2u_1 - u_2), \qquad \dfrac{\mathrm{d}^2 u_N}{\mathrm{d}t^2} = -c_N(2u_N - u_{N-1}), \\ \dfrac{\mathrm{d}^2 u_j}{\mathrm{d}t^2} = -c_j(2u_j - u_{j-1} - u_{j+1}) \quad (2 \le j \le N-1), \end{cases} \tag{4}$$

となる。ここで $c_j = m_1/m_j$ は j 番目の質点の質量の逆数に比例する無次元定数である (定義より $c_1 = 1$)。さらに j 番目の質点の無次元速度を $v_j = \mathrm{d}u_j/\mathrm{d}t$ と定義すれば, 結局

$$\begin{cases} \dfrac{\mathrm{d}u_j}{\mathrm{d}t} = v_j & (1 \le j \le N) \\ \dfrac{\mathrm{d}v_j}{\mathrm{d}t} = -c_j(2u_j - u_{j-1} - u_{j+1}) & (1 \le j \le N) \end{cases} \tag{5}$$

のような 1 階連立常微分方程式に帰着する (ただし $u_0 = u_{N+1} = 0$)。これをふまえて, 以下の演習課題 (a)–(e) をおこない, 結果を前回と同様に電子メールで報告せよ。しめきりは **6 月 18 日中**とする。

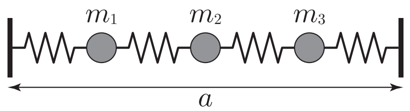

図 1: 連成振動子 ($N = 3$ の場合)。

(a) すべての質点の質量が等しいとおいて ($c_j = 1$), これら質点系の振動の時間発展を 4 次ルンゲ・クッタ法をもちいて計算し, 結果を適当な図またはムービーで表現せよ。たとえば $N = 10$, $\Delta_t = 0.1$, 積分時間は $t = 100$ までとし, 初期条件は, すべての質点は安定位置にあり ($u_j(0) = 0$), 速度は 1 番目の質点のみに初速度 $v_1(0) = \alpha$ を与え, それ以外は速度ゼロとしてみる ($v_j(0) = 0$, $j \ge 2$)。 質点の個数や初期条件などは各自適当に変えてもかまわない。

(b) 全エネルギー

$$E(t) = \sum_{j=1}^{N} \frac{v_j^2}{2c_j} + \frac{1}{2}\left[u_1^2 + u_N^2 + \sum_{j=2}^{N}(u_j - u_{j-1})^2\right]$$

の保存について議論せよ。たとえば誤差の相対値 $E(t)/E(0) - 1$ ($E(0) = \alpha^2/2$) は時間の関数としてどのようにふるまい, さらにそれは時間積分の刻み幅 Δ_t を変えたときどうなるかを調べる。

(c) 2 階の連立常微分方程式 (4) を

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}\boldsymbol{u} = A\boldsymbol{u} \tag{6}$$

のように行列・ベクトル表示する。ここで $\boldsymbol{u}(t)$ は質点の安定位置からの変位 u_j が順に並んだベクトルで, A は $N \times N$ の正方行列である。行列 A の固有値のうち絶対値最大のもの $\lambda_{\max}$ を物理数学的考察もしくは数値計算により求めよ (もちいた N の値も明示すること)。その結果をもとに, 1 階の連立常微分方程式 (5) を 4 次ルンゲ・クッタ法で安定に (どんなに時間が経っても変位が発散しないように) 時間積分するために必要な Δ_t の範囲を予想し, それを実際に数値的に確かめよ。

(d) この問題を, オイラー法や 2 次ルンゲ・クッタ法 (たとえば改良オイラー法) で解く場合と比べて, 今回のように, 4 次ルンゲ・クッタ法をもちいることのメリット（またはデメリット）について予想し, その根拠とともに簡単に述べよ（とくに数値解の精度や安定性, 実際に計算に要する時間などの観点から論ぜよ)。

(e) $j = 1$ 以外のある質点の質量が m_1 の s 倍である場合を考える (どの質点の質量を変えるかは各自適当に設定してよい。複数の質点の質量を変更してもよい)。$s = 1/10$, $s = 10$ の各場合について, 小問 (a)–(c) と同様の作業をおこなえ。

4 次ルンゲ・クッタ法

- 陽解法で精度がよく, 多くの微分方程式に適用可能な汎用性をもつ 4 次ルンゲ・クッタ (RK4) 法のアルゴリズムを図 2 にしめす。ただし解くべき微分方程式は

$$\frac{\mathrm{d}w}{\mathrm{d}t} = f(w), \qquad f \text{ は既知の関数},$$

で, 一般に w や f はベクトルであるとしている。このアルゴリズムが, 確かに (大域的に) 4 次精度をもっていることの証明は図 3 を参照のこと。

- とくに $f(w) = i\omega w$ であるような振動方程式のとき (ω は角周波数, i は虚数単位), RK4 法で時間ステップを 1 つ進めたときの数値解の変化率は, $x = \omega\Delta_t$ とおいて,

$$\frac{w_{n+1}}{w_n} = 1 - \frac{x^2}{2} + \frac{x^4}{24} + i\left(x - \frac{x^3}{6}\right) \equiv A\exp(i\Theta) \tag{7}$$

である。振幅 A が 1 以下であれば数値解は安定である。その条件は

$$A^2 = \left(1 - \tfrac{1}{2}x^2 + \tfrac{1}{24}x^4\right)^2 + \left(x - \tfrac{1}{6}x^3\right)^2 \le 1, \quad \text{すなわち} \quad \tfrac{1}{576}x^6(x^2 - 8) \le 0, \tag{8}$$

だから, 振動方程式に対する RK4 法の安定条件は

$$|\omega|\Delta_t \le 2\sqrt{2} \simeq 2.8284 \tag{9}$$

である。さらに $|x| \ll 1$ として $x = 0$ のまわりでテーラー展開すると,

$$A = 1 - \tfrac{1}{144}x^6 + \cdots, \quad \Theta = x\left(1 - \tfrac{1}{120}x^4 + \cdots\right), \tag{10}$$

すなわち振幅の局所誤差は $O(\Delta_t^6)$, 位相の局所誤差は $O(\Delta_t^4)$ であることがわかる (位相は遅れる)。

1. $\boldsymbol{p} \equiv \boldsymbol{w}_n$ とおいて, $\boldsymbol{k}_1 \equiv \boldsymbol{f}(\boldsymbol{p})$ を計算する。
2. $\boldsymbol{p} \equiv \boldsymbol{w}_n + \frac{1}{2}\Delta_t\,\boldsymbol{k}_1$ とおいて, $\boldsymbol{k}_2 \equiv \boldsymbol{f}(\boldsymbol{p})$ を計算する。
3. $\boldsymbol{p} \equiv \boldsymbol{w}_n + \frac{1}{2}\Delta_t\,\boldsymbol{k}_2$ とおいて, $\boldsymbol{k}_3 \equiv \boldsymbol{f}(\boldsymbol{p})$ を計算する。
4. $\boldsymbol{p} \equiv \boldsymbol{w}_n + \Delta_t\,\boldsymbol{k}_3$ とおいて, $\boldsymbol{k}_4 \equiv \boldsymbol{f}(\boldsymbol{p})$ を計算する。
5. $\boldsymbol{w}_{n+1} \equiv \boldsymbol{w}_n + \frac{1}{6}\Delta_t\,(\boldsymbol{k}_1 + 2\boldsymbol{k}_2 + 2\boldsymbol{k}_3 + \boldsymbol{k}_4)$ とする。時刻も進めてステップ 1 に戻る。

図 2: 微分方程式 $\mathrm{d}w/\mathrm{d}t = f(w)$ を 4 次ルンゲ・クッタ法で解くアルゴリズム。w_n は n 番目の時間ステップでの変数の値, Δ_t は時間ステップの幅。陽的なシンプソンの数値積分, とも解釈できる。

まずテーラー展開により, 補助変数 k_2, k_3, k_4 を $t = t_n$ での $f(w)$ (またはその導関数) の値で書き換える:

$$\begin{aligned}
k_2 &= f(w_n + \tfrac{1}{2}k_1\Delta_t) \\
&= f_n + \tfrac{1}{2}k_1\Delta_t f'_n + \tfrac{1}{8}k_1^2\Delta_t^2 f''_n + \tfrac{1}{48}k_1^3\Delta_t^3 f'''_n + \cdots \\
&= f_n + \tfrac{1}{2}f_n f'_n\Delta_t + \tfrac{1}{8}f_n^2 f''_n\Delta_t^2 + \tfrac{1}{48}f_n^3 f'''_n\Delta_t^3 + O(\Delta_t^4), \qquad (1)\\
k_3 &= f(w_n + \tfrac{1}{2}k_2\Delta_t) \\
&= f_n + \tfrac{1}{2}k_2\Delta_t f'_n + \tfrac{1}{8}k_2^2\Delta_t^2 f''_n + \tfrac{1}{48}k_2^3\Delta_t^3 f'''_n + \cdots \\
&= f_n + \tfrac{1}{2}\big(f_n + \tfrac{1}{2}f_n f'_n\Delta_t + \tfrac{1}{8}f_n^2 f''_n\Delta_t^2\big)f'_n\Delta_t + \tfrac{1}{8}\big(f_n + \tfrac{1}{2}f_n f'_n\Delta_t\big)^2 f''_n\Delta_t^2 + \tfrac{1}{48}f_n^3 f'''_n\Delta_t^3 + \cdots \\
&= f_n + \tfrac{1}{2}f_n f'_n\Delta_t + \Big[\tfrac{1}{4}f_n(f'_n)^2 + \tfrac{1}{8}f_n^2 f''_n\Big]\Delta_t^2 + \Big[\tfrac{3}{16}f_n^2 f'_n f''_n + \tfrac{1}{48}f_n^3 f'''_n\Big]\Delta_t^3 + O(\Delta_t^4), \qquad (2)\\
k_4 &= f(w_n + k_3\Delta_t) \\
&= f_n + k_3\Delta_t f'_n + \tfrac{1}{2}k_3^2\Delta_t^2 f''_n + \tfrac{1}{6}k_3^3\Delta_t^3 f'''_n + \cdots \\
&= f_n + \Big[f_n + \tfrac{1}{2}f_n f'_n\Delta_t + \tfrac{1}{4}f_n(f'_n)^2\Delta_t^2 + \tfrac{1}{8}f_n^2 f''_n\Delta_t^2\Big]f'_n\Delta_t \\
&\quad + \tfrac{1}{2}\big(f_n + \tfrac{1}{2}f_n f'_n\Delta_t\big)^2 f''_n\Delta_t^2 + \tfrac{1}{6}f_n^3 f'''_n\Delta_t^3 + \cdots \\
&= f_n + f_n f'_n\Delta_t + \Big[\tfrac{1}{2}f_n(f'_n)^2 + \tfrac{1}{2}f_n^2 f''_n\Big]\Delta_t^2 + \Big[\tfrac{1}{4}f_n(f'_n)^3 + \tfrac{5}{8}f_n^2 f'_n f''_n + \tfrac{1}{6}f_n^3 f'''_n\Big]\Delta_t^3 + O(\Delta_t^4). \qquad (3)
\end{aligned}$$

ただし f'_n ($f''_n, \cdots$) は $t = t_n$ における関数 f の 1 階 (2 階, $\cdots$) 導関数の値をあらわす。したがって, 4 次のルンゲ・クッタ法による推定値 $w_{n+1}^{(4)}$ は

$$\begin{aligned}
w_{n+1}^{(4)} &= w_n + \tfrac{1}{6}\Delta_t(k_1 + 2k_2 + 2k_3 + k_4) \\
&= w_n + f_n\Delta_t + \tfrac{1}{2}f_n f'_n\Delta_t^2 + \tfrac{1}{6}\Big(f_n(f'_n)^2 + f_n^2 f''_n\Big)\Delta_t^3 + \tfrac{1}{24}\Big(f_n(f'_n)^3 + 4f_n^2 f'_n f''_n + f_n^3 f'''_n\Big)\Delta_t^4 + O(\Delta_t^5)
\end{aligned} \qquad (4)$$

を満たす。ここで合成関数の微分公式により,

$$\begin{cases}
\dfrac{\mathrm{d}}{\mathrm{d}t}f(w(t)) = f'\dfrac{\mathrm{d}w}{\mathrm{d}t} = ff', \\
\dfrac{\mathrm{d}^2}{\mathrm{d}t^2}f(w(t)) = \dfrac{\mathrm{d}}{\mathrm{d}t}(ff') = f'\dfrac{\mathrm{d}w}{\mathrm{d}t}\cdot f' + f\cdot f''\dfrac{\mathrm{d}w}{\mathrm{d}t} = f(f')^2 + f^2 f'', \\
\dfrac{\mathrm{d}^3}{\mathrm{d}t^3}f(w(t)) = f'\dfrac{\mathrm{d}w}{\mathrm{d}t}\cdot(f')^2 + f\cdot 2f''\dfrac{\mathrm{d}w}{\mathrm{d}t} + 2ff'\dfrac{\mathrm{d}w}{\mathrm{d}t}\cdot f'' + f^2\cdot f'''\dfrac{\mathrm{d}w}{\mathrm{d}t} = f(f')^3 + 4f^2 f' f'' + f^3 f''',
\end{cases} \qquad (5)$$

がなりたつから, 結局

$$w_{n+1}^{(4)} = w_n + f_n\Delta_t + \frac{1}{2}\left(\frac{\mathrm{d}f}{\mathrm{d}t}\right)_{t_n}\Delta_t^2 + \frac{1}{6}\left(\frac{\mathrm{d}^2 f}{\mathrm{d}t^2}\right)_{t_n}\Delta_t^3 + \frac{1}{24}\left(\frac{\mathrm{d}^3 f}{\mathrm{d}t^3}\right)_{t_n}\Delta_t^4 + O(\Delta_t^5) \qquad (6)$$

である。これは厳密解 w_{n+1} の $t = t_n$ でのテーラー展開と, Δ_t^4 に比例する項まで完全に一致する (おわり)。

図 3: いわゆる 4 次ルンゲ・クッタ法の局所誤差がほんとうに $O(\Delta_t^5)$ であることの証明。

解法のヒントと補足

- 変数の無次元化は, 物理モデルを解析する上で重要である。いまの場合, $N-1$ 個の無次元パラメータ c_j ($2 \le j \le N$) を指定することによって, 系の基本的性質は決定される。いちいちバネ定数を変えた計算を何通りもやる必要はなく, 単に時間軸を調節すれば, 系の物理的記述は尽きるのである。

- 式 (5) を, 長さ $2N$ 元のベクトル変数に対する連立常微分方程式と考える。したがって長さ $2N$ の配列を定義して, それに変数を格納するのが自然である。プログラムの見た目のわかりやすさのため, 2 次元配列 `w(1:n, 1:2)` にしてやって,

  変位: $u_j(t) \iff$ `w(j, 1)`,　速度: $v_j(t) \iff$ `w(j, 2)`,

  のような対応関係を, プログラム全体にわたって「約束」する。あとは配列 `w(:, :)` が $2N$ 個の従属変数を格納していると思って, ルンゲ・クッタ法のアルゴリズム (図 2) を機械的にあてはめればよい。プログラム例 (図 4) を見よ。

- 運動方程式は, 2 階の微分方程式として

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}\boldsymbol{u} = A\boldsymbol{u}, \quad \boldsymbol{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{pmatrix}, \quad A = \begin{pmatrix} -2c_1 & c_1 & & \\ c_2 & -2c_2 & c_2 & \\ & \ddots & \ddots & \ddots \\ & & c_N & -2c_N \end{pmatrix},$$

  と書ける。三重対角行列 A の N 個の固有値 (すべて負の実数になるべきである。確認せよ) のうち, 時間積分の安定性に関わってくるのは, 絶対値最大の固有値 $\lambda_{\mathrm{max}} = -\omega_{\mathrm{max}}^2$ である。これに対応する 2 階常微分方程式は

$$\frac{\mathrm{d}^2 u'}{\mathrm{d}t^2} = -\omega_{\mathrm{max}}^2 u'$$

  で, 複素変数に拡張すれば, これは 1 階常微分方程式

$$\frac{\mathrm{d}u'}{\mathrm{d}t} = \pm i\omega_{\mathrm{max}} u'$$

  と等価である。すなわち運動方程式を 1 階の常微分方程式と見た場合の固有値は純虚数である。これと, 振動方程式に対するルンゲ・クッタ法の安定条件の結果 (9) とを見比べれば, Δ_t に関する制約条件が得られる。

- 行列 A の絶対値最大の固有値を求める, もっとも簡便な計算手法である「ベキ乗法」のアルゴリズムは以下のよう。

  1. 零ベクトルでない初期ベクトル $\boldsymbol{y}$ と固有値の初期値 $\lambda = 0$ を与える。
  2. ベクトル $\boldsymbol{y}$ の成分のうち絶対値最大のものを探し, それですべての成分を規格化する。
  3. ベクトル $\boldsymbol{y}_2 \equiv A\boldsymbol{y}$ を計算する。
  4. ベクトルの内積 $z \equiv \boldsymbol{y} \cdot \boldsymbol{y}$ を計算する。
  5. ベクトルの内積 $z_2 \equiv \boldsymbol{y}_2 \cdot \boldsymbol{y}$ を計算する。
  6. 絶対値 $|\lambda - z_2/z|$ があるしきい値 (たとえば 6 桁の精度で求めたかったら 10^{-6} とする) より小さくなったら終了。そうでないなら $\boldsymbol{y} \equiv \boldsymbol{y}_2$, $\lambda \equiv z_2/z$ とおいて, ステップ 2 に戻る。

  最終的に出力される λ が A の絶対値最大の固有値の近似値になり, $\boldsymbol{y}$ が固有ベクトルになる。

数値データの可視化に関するヒント

- すべての時間ステップでデータを書き出すと, ファイルが膨大になる恐れがあるので, あらかじめ適当に間引いておくのも一法である。データ出力の時間間隔を指定する方法と, 時間ステップのカウンタを用意しておく方法の 2 通りが考えられる (図 5)。

- プログラム例 (図 4, 5) のようにデータを出力したとすると, すべての質点の位置 $x_j(t) = \bar{x}_j + u_j(t)$ を時間の関数としてまとめてプロットするのは比較的たやすい。図 6 にシェルスクリプトの例をしめす。結果の例は図 7 にしめす。

- 振動のようすをムービーにするには, 等時間間隔ごとに質点の位置を絵にかいて, 最終的に得られるたくさんの画像ファイルを, なんらかの動画形式のファイルに変換してやればよい。たとえば図 8 にしめしたシェルスクリプトは, 連番の画像ファイルをたくさん生成するから, ImageMagick の animate コマンドで見たり, convert コマンドで GIF アニメをつくったりできるだろう。以下はその例:

```
sh> animate -delay 10 ex2_0[0-2][0-9].gif
sh> convert -delay 10 ex2_??[02468].gif ex2_all.gif
```

```
! n 個の質点からなる連成振動子の運動を 0 < t < tend の範囲で解く。長さ 2n の配列 w(1:n, 1:2) を定義して,
! j 番目の質点の変位を w(j, 1), 速度を w(j, 2) に格納する。質点の質量の逆数に比例する無次元パラメータは
! 配列 c(1:n) に格納する。初期条件は変位ゼロ, 速度は 1 番目の質点のみ vzero とする。
program coupled_oscillators
  implicit none
  integer(4), parameter :: n = 10
  integer(4) :: j
  real(8) :: t, dt
  real(8), dimension(n) :: c
  real(8), dimension(n, 2) :: w, p, k1, k2, k3, k4
  real(8), parameter :: vzero = 0.1d0, tend = 100.0d0
  ! 時間ステップの刻み幅 dt を標準入力から読み込む。
  read(*, *) dt
  ! 質量はみんな同じ。
  c(:) = 1.0d0
  ! 未知変数 w と時刻 t の初期化。
  w(:, 1) = 0.0d0
  w(1, 2) = vzero
  w(2:n, 2) = 0.0d0
  t = 0.0d0
  ! 時間積分の開始（無限 do loop)。
  do
    ! 時刻 t での質点の位置を書き出す。
    write(*, "(f12.7, 99e16.8)") t, (w(j, 1) + dble(j)/dble(n+1), j = 1, n)
    ! 時刻が tend を超えたら無限 do loop から脱出する。
    if ( t >= tend ) stop
    ! ４次ルンゲ・クッタ法。微分方程式を dw/dt = f(w) だと思って, 機械的にアルゴリズムを適用する。
    ! まず補助変数 k1 = f(p) を計算する。
    p(:, :) = w(:, :)
    call rhs (k1, p, c, n)
    ! 補助変数 k2 = f(p) を計算する。
    p(:, :) = w(:, :) + 0.5d0 * dt * k1(:, :)
    call rhs (k2, p, c, n)
    ! 補助変数 k3 = f(p) を計算する。
    p(:, :) = w(:, :) + 0.5d0 * dt * k2(:, :)
    call rhs (k3, p, c, n)
    ! 補助変数 k4 = f(p) を計算する。
    p(:, :) = w(:, :) + dt * k3(:, :)
    call rhs (k4, p, c, n)
    ! 未知変数 w と時刻 t を次の時間ステップでの値で上書きする。
    w(:, :) = w(:, :) + (dt/6.0d0) * (k1(:, :) + k4(:, :) + 2.d0*(k2(:, :) + k3(:, :)))
    t = t + dt
  end do
end
! 微分方程式 dw/dt = f(w) の右辺を計算するサブルーチン。
subroutine rhs (f, w, c, n)
  implicit none
  integer(4) :: n, j
  real(8), dimension(n) :: c
  real(8), dimension(n, 2) :: f, w
  do j = 1, n
    ! du_j/dt = v_j
    f(j, 1) = w(j, 2)
    ! dv_j/dt = -c_j (2u_j - u_{j-1} - u_{j+1})
    if ( j == 1 ) then
      f(j, 2) = - c(j) * (2.d0*w(j, 1) - w(j+1, 1))
    else if ( j == n ) then
      f(j, 2) = - c(j) * (2.d0*w(j, 1) - w(j-1, 1))
    else
      f(j, 2) = - c(j) * (2.d0*w(j, 1) - w(j-1, 1) - w(j+1, 1))
    end if
  end do
  return
end
```

図 4: 連成振動子の運動方程式を 4 次ルンゲ・クッタ法で解くプログラムの例。ルンゲ・クッタ法のアルゴリズムでは, 時間ステップを 1 段進めるのに, 方程式の右辺を 4 回も計算しなければならない。そこで方程式の右辺を計算する部分をサブルーチン (副プログラム) 化するのがよい。

```
real(8) :: t, t_out
real(8), parameter :: dt = 0.01d0, dt_out = 0.5d0
integer(4) :: it
integer(4), parameter :: nstep_out = 50
t = 0.0d0
t_out = 0.0d0
it = 0
do
  ! 方法（1）: 時間 dt_out おきに出力する
  if ( t >= t_out ) then
    write(10, "(f12.7, 99e16.8)") t, (w(j, 1) + dble(j)/dble(n+1), j = 1, n)
    t_out = t_out + dt_out
  end if
  ! 方法（2）: 積分の時間ステップ nstep_out おきに出力する
  if ( mod(it, nstep_out) == 0 ) then
    write(20, "(f12.7, 99e16.8)") t, (w(j, 1) + dble(j)/dble(n+1), j = 1, n)
  end if
  t = t + dt
  it = it + 1
end do
```

図 5: データを間引いて出力する方法。どちらの方法もほぼ同じ結果をもたらす。

```
#! /bin/sh
# 図 4 のプログラムの出力結果がファイル ex2.txt に保存してあると仮定する。
dat=ex2.txt
ps=ex2.eps
# gnuplot を起動。ちょっと横長の図にして、線の種類はぜんぶ同じにする。gnuplot の継続行はバックスラッシュ
# であるが、シェルの継続行と勘違いされてしまうので、バックスラッシュを 2 個重ねる。
gnuplot <<EOF
set terminal postscript eps
set output "${ps}"
set xrange [0:100]
set yrange [0:1]
set xlabel "time" font "Helvetica,24"
set ylabel "position" font "Helvetica,24"
set size ratio 0.5
unset key
plot '${dat}' using 1:2 with lines linetype 1, \\
     '${dat}' using 1:3 with lines linetype 1, \\
     '${dat}' using 1:4 with lines linetype 1, \\
     '${dat}' using 1:5 with lines linetype 1, \\
     '${dat}' using 1:6 with lines linetype 1, \\
     '${dat}' using 1:7 with lines linetype 1, \\
     '${dat}' using 1:8 with lines linetype 1, \\
     '${dat}' using 1:9 with lines linetype 1, \\
     '${dat}' using 1:10 with lines linetype 1, \\
     '${dat}' using 1:11 with lines linetype 1
EOF
```

図 6: 質点の位置の時間変化のようすをまとめてあらわす絵 (図 7 に似た絵) をかくシェルスクリプト。このスクリプトをファイル `hoge.sh` に保存し, `chmod u+x hoge.sh` などとすれば, そのファイルはコマンドとして実行可能になる。なお「Gnuplot のスクリプト」を出力するためのシェルスクリプトをつくれば, 質点の個数 N を増やしたり減らしたりしたときに, 描画作業を自動化できるだろう。各自工夫せよ。

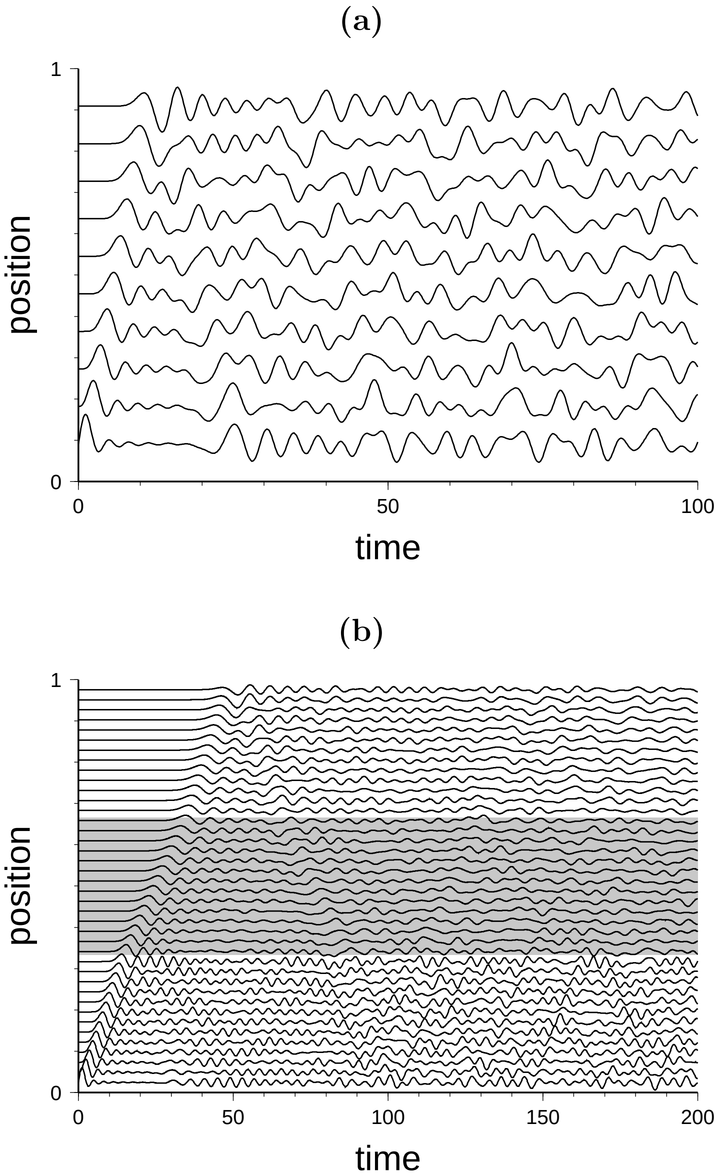

図 7: 連成振動子の運動。各質点の位置 $x_j(t) = \bar{x}_j + u_j(t)$ を時間の関数としてプロットした。(a) $N = 10$, $\alpha = 1/10$ ですべての質点の質量が等しい場合。 (b) $N = 40$, $\alpha = 1/20$ で, 静止位置が $1/3 \le \bar{x}_j \le 2/3$ (灰色の部分) である質点に対しては $c_j = 1/2$, それ以外では $c_j = 1$ とした場合。振動の位相が x 方向に「波」として伝わり, 「低速度層」で反射や屈折がおこっていることがわかる。N を大きくしていくと, 系のふるまいは, 弦の振動や音波の伝搬のそれに近づいていく。

```
#! /bin/sh
# 図 4 のプログラムの出力結果がファイル ex2.txt に保存してあると仮定する。
dat=ex2.txt
# 何ステップ分のデータがあるか数える（wc コマンドがなんたるかは man wc を見よ）。
nlines=‘cat ${dat} | wc -l‘
echo "${dat} contains ${nlines} lines."
m=‘expr ${nlines} - 1‘
# すべての時間ステップにおいて、質点の位置をプロットし、画像ファイル（GIF 形式）に保存する。
#「1 行おき」に絵をかきたい場合は seq 0 2 ${m} とする（seq コマンドがなんたるかは man seq を見よ）。
for i in ‘seq 0 ${m}‘
do
  # 画像ファイル名には連番の番号を付す。ただしあとの処理のことを考えて、連番号は 3 桁でそろえる。
  # そのために awk の printf 関数をつかって桁をそろえ、シェル変数 ii に代入する。
  ii=‘echo ${i} | awk '{printf("%03d", $1)}'‘
  img=ex2_${ii}.gif
  echo "Creating ${img} ..."
  # gnuplot を起動。横軸は質点の位置。縦軸は任意であるが座標ゼロのところに質点を置き、丸印であらわす。
  gnuplot <<EOF
set terminal gif
set output "${img}"
set xrange [0:1]
set yrange [-0.1:0.1]
set size ratio -1
set noytics
unset key
plot '${dat}' every 1:1:${i}:0:${i}:0 using 2:(0) with points pointtype 7 pointsize 1, \\
     '${dat}' every 1:1:${i}:0:${i}:0 using 3:(0) with points pointtype 7 pointsize 1, \\
     '${dat}' every 1:1:${i}:0:${i}:0 using 4:(0) with points pointtype 7 pointsize 1, \\
     '${dat}' every 1:1:${i}:0:${i}:0 using 5:(0) with points pointtype 7 pointsize 1, \\
     '${dat}' every 1:1:${i}:0:${i}:0 using 6:(0) with points pointtype 7 pointsize 1, \\
     '${dat}' every 1:1:${i}:0:${i}:0 using 7:(0) with points pointtype 7 pointsize 1, \\
     '${dat}' every 1:1:${i}:0:${i}:0 using 8:(0) with points pointtype 7 pointsize 1, \\
     '${dat}' every 1:1:${i}:0:${i}:0 using 9:(0) with points pointtype 7 pointsize 1, \\
     '${dat}' every 1:1:${i}:0:${i}:0 using 10:(0) with points pointtype 7 pointsize 1, \\
     '${dat}' every 1:1:${i}:0:${i}:0 using 11:(0) with points pointtype 7 pointsize 1
EOF
done
```

図 8: 質点の運動をアニメーション表示するために必要な画像ファイルを書き出すシェルスクリプトの例。各時間ステップでの質点の位置を, 連番の画像ファイルに書きこむ。Gnuplot で GIF 形式の出力ができなかったら, PostScript など他の形式のファイルに書き出して, あとで変換してもよい。