古典型Cartan行列の対角化

- Author: 黒木玄
- Date: 2019-04-09～2019-05-11, 2023-01-14
-
-

古典型Cartan行列の固有ベクトルの表が検索しても容易に見付けることができなかったので, その表を作ることにした. このノートでは, 有限型の古典型Cartan行列とアフィン古典型の一般Cartan行列の固有ベクトルを求める. 古典Cartan行列の特性多項式とChebyshev多項式の関係に関するよく知られた結果も解説する.

A_∞ 型Cartan行列は1次元格子 $\mathbb{Z}$ 上の離散Laplacianであり, 有限サイズの古典型Cartan行列は適当な境界条件を課すことによって得られる.

目次

In [1]:

```julia
using Plots
default(fmt=:png, tickfontsize=6, titlefontsize=8)
using LinearAlgebra
using SymPy: SymPy, sympy, @syms, @vars, simplify, PI, oo
const ChebyshevT = sympy.chebyshevt_poly
const ChebyshevU = sympy.chebyshevu_poly;
```

1 古典型の場合

1.1 古典型Cartan行列と隣接行列の定義

A_n, C_n, B_n, D_n 型の**Cartan行列**とはそれぞれ以下の形の $n\times n$ 行列のことである.

A_n 型:

$$
\begin{bmatrix}
2 & -1 & & & \\
-1 & 2 & -1 & & \\
 & -1 & 2 & \ddots & \\
 & & \ddots & \ddots & -1 \\
 & & & -1 & 2
\end{bmatrix}.
$$

C_n 型:

$$\begin{bmatrix} 2 & -2 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{bmatrix}.$$

B_n 型:

$$\begin{bmatrix} 2 & -1 & & & \\ -2 & 2 & -1 & & \\ & -1 & 2 & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{bmatrix}.$$

D_n 型:

$$\begin{bmatrix} 2 & 0 & -1 & & & \\ 0 & 2 & -1 & & & \\ -1 & -1 & 2 & -1 & & \\ & & -1 & 2 & \ddots & \\ & & & \ddots & \ddots & -1 \\ & & & & -1 & 2 \end{bmatrix}.$$

Cartan行列はどれも $2E - A$ (E は単位行列) の形をしている. そのとき A を**隣接行列**と呼ぶ. Cartan行列と隣接行列の固有ベクトルは一致し, Cartan行列の固有値は 2 から隣接行列の固有値を引いたものになる. したがって, Cartan行列の固有値と固有ベクトルを求めるためには, 対応する隣接行列のそれらを求めればよい.

上における C_n, B_n, D_n 型のCartan行列の表示において, 行列の成分の位置を表す番号 $1, 2, \dots, n$ を反転させればよく見る表示に戻る. 以下では1次元格子 $\mathbb{Z}$ の 0 を境界条件を最初に設定する場所として選ぶので上のような表示を採用した.

1.2 計算の仕方の方針

我々は A_n, C_n, B_n, D_n 型の隣接行列の固有値と固有ベクトルを求めたい. しかし, 隣接行列の特性多項式を計算して, 固有値を計算して, 固有ベクトルを計算する経路を採用すると, 答えを知らないと非常に面倒な計算が必要になる.

そこで我々は, まず, 次の A_∞ 型の両無限隣接行列の固有ベクトルを構成し, 境界条件を設定した結果として, $A_{\infty/2}$, C_∞ 型の半無限隣接行列の固有ベクトルを構成し, C_∞ 型の半無限隣接行列の固有ベクトルの最初の成分を半分にすることによって, B_∞ を重複させることによって, D_∞ 型の半無限隣接行列の固有ベクトルを構成する.

そして, 2つ目の境界条件を設定することによって, A_n, C_n, B_n, D_n 型隣接行列の固有ベクトルを得る.

1.3 無限隣接行列の定義

A_∞, $A_{\infty/2}$, C_∞, B_∞, D_∞ 型の隣接行列は以下のように定義される.

A_∞ 型の隣接行列とは次の $\mathbb{Z} \times \mathbb{Z}$ 行列のことである:

$$\left[\delta_{i,j-1} + \delta_{i,j-1}\right]_{i,j\in\mathbb{Z}} = \begin{bmatrix} \ddots & \ddots & & & & \\ \ddots & 0 & 1 & & & \\ & 1 & 0 & 1 & & \\ & & 1 & 0 & 1 & \\ & & & 1 & 0 & \ddots \\ & & & & \ddots & \ddots \end{bmatrix}.$$

$A_{\infty/2}$ 型の隣接行列とは次の $\{1, 2, \dots\} \times \{1, 2, \dots\}$ 行列のことである:

$$\left[\delta_{i,j-1} + \delta_{i,j-1}\right]_{i,j>0} = \begin{bmatrix} 0 & 1 & & & \\ 1 & 0 & 1 & & \\ & 1 & 0 & 1 & \\ & & 1 & 0 & \ddots \\ & & & \ddots & \ddots \end{bmatrix}.$$

C_∞ 型の隣接行列とは次の $\{0, 1, 2, \dots\} \times \{0, 1, 2, \dots\}$ 行列のことである:

$$\begin{bmatrix} 0 & 2 & & & \\ 1 & 0 & 1 & & \\ & 1 & 0 & 1 & \\ & & 1 & 0 & \ddots \\ & & & \ddots & \ddots \end{bmatrix}.$$

B_∞ 型の隣接行列とは次の $\{0, 1, 2, \dots\} \times \{0, 1, 2, \dots\}$ 行列のことである:

$$\begin{bmatrix} 0 & 1 & & & \\ 2 & 0 & 1 & & \\ & 1 & 0 & 1 & \\ & & 1 & 0 & \ddots \\ & & & \ddots & \ddots \end{bmatrix}.$$

D_∞ 型の隣接行列とは次の $\{0', 0, 1, 2, \dots\} \times \{0', 0, 1, 2, \dots\}$ 行列のことである:

$$\begin{bmatrix} 0 & 0 & 1 & & & \\ 0 & 0 & 1 & & & \\ 1 & 1 & 0 & 1 & & \\ & & 1 & 0 & 1 & \\ & & & 1 & 0 & \ddots \\ & & & & \ddots & \ddots \end{bmatrix}.$$

1.4 A_∞ 型

A は A_∞ 型隣接行列($\mathbb{Z} \times \mathbb{Z}$ 行列)であるとする:

$$A = \begin{bmatrix} \ddots & \ddots & & & & \\ \ddots & 0 & 1 & & & \\ & 1 & 0 & 1 & & \\ & & 1 & 0 & 1 & \\ & & & 1 & 0 & \ddots \\ & & & & \ddots & \ddots \end{bmatrix}.$$

ベクトル $v = [x_j]_{j\in\mathbb{Z}} \neq 0$ が A の固有値 α の固有ベクトルであるための必要十分条件は

$$x_{j-1} + x_{j+1} = \alpha x_j \quad (j \in \mathbb{Z})$$

が成立することである. この定数係数の線形漸化式の解空間は常に2次元になるので, A の固有空間の次元も常に2次元になる.

$\alpha \neq \pm 2$ のとき, z に関する2次方程式 $z^2 - \alpha z - 1 = 0$ は異なる2つの解を持ち, その片方を z とするともう一方は z^{-1} になり, $z \neq \pm 1$ となる. 逆に, $z \neq \pm 1$ のとき, $\alpha = z + z^{-1}$ とおくと, $\alpha \neq \pm 2$ となる. (例: $\alpha = 0$ のとき, $z^{\pm 1} = \pm i$.)

ゆえに, A の固有値 $\alpha = z + z^{-1} \neq \pm 2$ の固有空間の基底として, $x_j = z^j, z^{-j}$ が取れる:

$$z^{j-1} + z^{j+1} = (z + z^{-1}) z^j.$$

A の固有値 $\alpha = \pm 2$ の固有空間の基底として, $x_j = (\pm 1)^j, j(\pm 1)^{j-1}$ が取れる:

$$(j-1)(\pm 1)^{j-2} + (j+1)(\pm 1)^j = ((j-1)(\pm 1) + (j+1)(\pm 1))(\pm 1)^{j-1} = \pm 2j(\pm 1)^{j-1}.$$

1.5 A_∞ 型Cartan行列と $\mathbb{R}$ 上の正値Laplacianの関係

$\mathbb{R}$ 上の正値Laplacianとは $-(d/dt)^2$ のことである. $f(t)$ が C^2 級ならば

$$f(t \pm h) - f(t) = \pm f'(t)h + \frac{1}{2} f''(t)h^2 + o(h^2)$$

が成立している. ゆえに

$$-f(t-h) + 2f(t) - f(t+h) = -f''(t)h^2 + o(h^2)$$

なので

$$\lim_{h\to 0} \frac{-f(t-h) + 2f(t) - f(t+h)}{h^2} = -\left(\frac{d}{dt}\right)^2 f(t).$$

ゆえに, $f(t)$ の値を $\mathbb{Z}$ 上に制限して得られる数列 $x_k = f(k)$ ($k \in \mathbb{Z}$) を考えるとき, 無限次元ベクトル $[x_k]_{k\in\mathbb{Z}}$ を

$$[-x_{k-1} + 2x_k - x_{k+1}]_{k\in\mathbb{Z}}$$

に対応させる線形変換は正値Laplacianの離散化とみなされる. その線形変換を表現する行列は A_∞ 型Cartan行列に一致する.

$f(0) = 0$ という条件を課すことは, $x_0 = 0$ という条件を課すことに対応しており, そのとき A_∞ 型Cartan行列によって無限次元ベクトル $[x_k]_{k\in\mathbb{Z}}$ をうつすと, うつした先のベクトルの第1成分は

$$2x_1 - x_2$$

になり, x_0 が見掛け上見えなくなるので, A_∞ 型Cartan行列は $[x_k]_{k=1}^\infty$ にも自然に作用できるようになる. その表現行列が $A_{\infty/2}$ 型Cartan行列になる. このように, 境界条件を設定することによって, A_∞ 型Cartan行列から, サイズの小さな別のCartan行列が得られる. 詳しくは以下の解説を参照せよ.

1.6 $A_{\infty/2}$ 型

A は A_∞ 型の両無限隣接行列であるとし, A_+ は $A_{\infty/2}$ 型の片無限隣接行列であるとする:

$$
A = \begin{bmatrix}
\ddots & \ddots & & & & \\
\ddots & 0 & 1 & & & \\
& 1 & 0 & 1 & & \\
& & 1 & 0 & 1 & \\
& & & 1 & 0 & \ddots \\
& & & & \ddots & \ddots
\end{bmatrix}, \quad
A_+ = \begin{bmatrix}
0 & 1 & & & \\
1 & 0 & 1 & & \\
& 1 & 0 & 1 & \\
& & 1 & 0 & \ddots \\
& & & \ddots & \ddots
\end{bmatrix}
$$

ベクトル $v = [x_j]_{j\in\mathbb{Z}}$ が A の固有値 α の固有ベクトルのとき, $x_0 = 0$ ならば, ベクトル $v_+ = [x_j]_{j=1}^\infty$ は A_+ の固有値 α の固有ベクトルになる.

ゆえに, A_+ の固有値 $\alpha = z + z^{-1} \neq \pm 2$ $(z \neq \pm 1)$ の固有空間の基底として $\left[z^j - z^{-j}\right]_{j=0}^\infty$ が取れ, A_+ の固有値 $\alpha = \pm 2$ の固有空間の基底として $\left[j(\pm 1)^{j-1}\right]_{j=0}^\infty$ が取れる.

注意: 以上で課した x_j 達に関する条件 $x_0 = 0$ は奇函数の条件 $x_{-j} = -x_j$ と同値である. □

1.7 A_n 型

A_n 型の隣接行列は次の形の $n \times n$ 行列になる.

$$
A = \begin{bmatrix}
0 & 1 & & & \\
1 & 0 & 1 & & \\
& 1 & 0 & \ddots & \\
& & \ddots & \ddots & 1 \\
& & & 1 & 0
\end{bmatrix}.
$$

$A_{\infty/2}$ 型の隣接行列の固有値 $\alpha = z + z^{-1} \neq \pm 1$ の固有ベクトル $[z^k - z^{-k}]_{k=1}^\infty$, $z \neq \pm 1$ において, $z^{n+1} - z^{-(n+1)} = 0$ が成立しているとき, ゆえに特に $z = e^{i\cdot j\pi/(n+1)}$, $j = 1, 2, \dots, n$ のとき, ベクトル

$$
[z^k - z^{-k}]_{k=1}^n = 2i\left[\sin\frac{kj\pi}{n+1}\right]_{k=1}^n
$$

は A_n 型の隣接行列 A の固有値

$$
z + z^{-1} = 2\cos\frac{j\pi}{n+1}
$$

の固有ベクトルになる. そこで θ_j を次のように定める:

$$
\theta_j = \frac{j\pi}{n+1}.
$$

このとき, A_n 型の隣接行列 A の互いに異なる n 個の固有値と対応する固有ベクトルとして以下が取れる:

$$
\alpha_j = 2\cos\theta_j, \qquad v_j = \begin{bmatrix} \sin(1\theta_j) \\ \sin(2\theta_j) \\ \vdots \\ \sin(n\theta_j) \end{bmatrix} \qquad (j = 1, 2, \dots, n).
$$

以上の結果を直接確認したい場合には, $\sin 0\theta_j = \sin(n+1)\theta_j = 0$ と $\sin 2\theta_j = 2\cos\theta_j \sin\theta_j$ および, $\sin((k\pm 1)\theta) = \cos\theta\sin(k\theta) \pm \sin\theta\cos(k\theta)$ より

$$
\sin((k-1)\theta) + \sin((k+1)\theta) = 2\cos\theta\ \sin(k\theta)
$$

となることなどに注意せよ.

In [2]:
```julia
function adjacent_matrix_of_type_A(n)
    @assert n ≥ 2
    SymTridiagonal(zeros(Int,n), ones(Int,n-1))
end

function eigenvectors_of_type_A(n)
    V = zeros(n, n)
    for j in 1:n
        θ_j = (j*π)/(n+1)
        for i in 1:n
            V[i,j] = 2sin(i*θ_j)
        end
    end
    V
end

function eigenvalues_of_type_A(n)
    α = zeros(n)
    for j in 1:n
        θ_j = j/(n+1)*π
        α[j] = 2cos(θ_j)
    end
    α
end
```

Out[2]:
```
eigenvalues_of_type_A (generic function with 1 method)
```

In [3]:

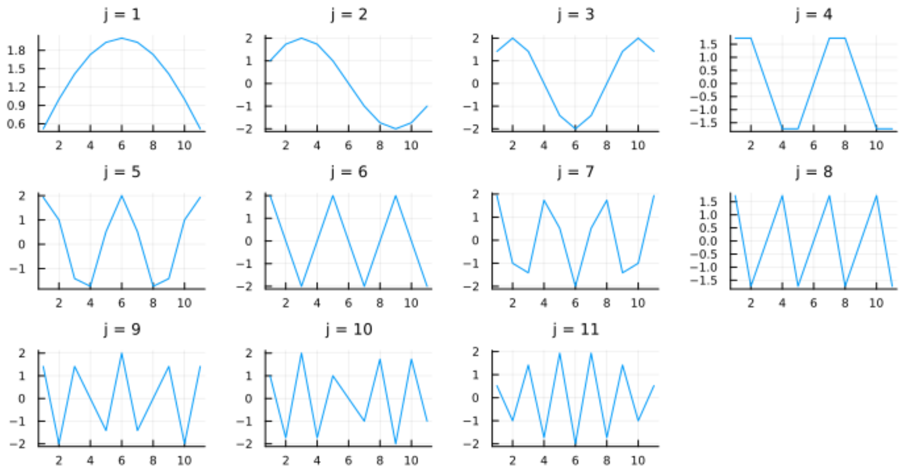
```julia
n = 11
A = adjacent_matrix_of_type_A(n)
V = eigenvectors_of_type_A(n)
α = eigenvalues_of_type_A(n)
display(A)
display(round.(V\(A*V), digits=1))
display(round.(diag(V\(A*V))', digits=2))
display(round.(α', digits=2))

PP = []
for j in 1:n
    P = plot(V[:,j], title="j = $j", legend=false)
    push!(PP, P)
end
plot(PP..., size=(700, 360), legend=false) ▷ display
```

```
11×11 SymTridiagonal{Int64, Vector{Int64}}:
 0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0
```

```
11×11 Matrix{Float64}:
  1.9  -0.0   0.0   0.0   0.0   0.0  -0.0   0.0  -0.0   0.0  -0.0
  0.0   1.7  -0.0  -0.0   0.0  -0.0   0.0  -0.0  -0.0   0.0   0.0
 -0.0   0.0   1.4   0.0   0.0   0.0   0.0   0.0   0.0  -0.0  -0.0
 -0.0  -0.0   0.0   1.0   0.0  -0.0   0.0  -0.0   0.0   0.0   0.0
 -0.0   0.0  -0.0   0.0   0.5   0.0  -0.0   0.0  -0.0  -0.0  -0.0
  0.0  -0.0   0.0  -0.0   0.0   0.0   0.0  -0.0   0.0   0.0   0.0
 -0.0   0.0  -0.0   0.0   0.0   0.0  -0.5   0.0  -0.0  -0.0  -0.0
 -0.0   0.0   0.0  -0.0  -0.0  -0.0   0.0  -1.0   0.0   0.0   0.0
  0.0   0.0  -0.0   0.0   0.0   0.0   0.0   0.0  -1.4  -0.0  -0.0
 -0.0  -0.0   0.0  -0.0  -0.0  -0.0  -0.0  -0.0  -0.0  -1.7   0.0
  0.0  -0.0   0.0   0.0  -0.0   0.0   0.0   0.0   0.0   0.0  -1.9
```

```
1×11 Matrix{Float64}:
 1.93  1.73  1.41  1.0  0.52  0.0  -0.52  -1.0  -1.41  -1.73  -1.93
```

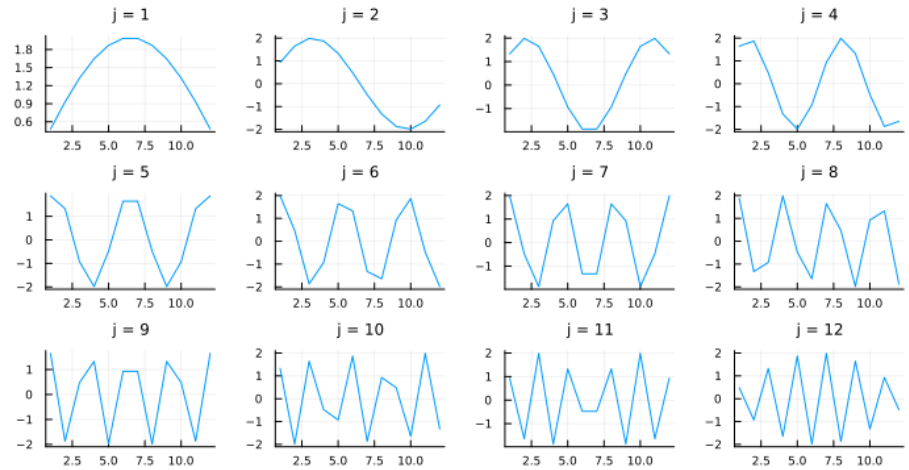
```
1×11 Matrix{Float64}:
 1.93  1.73  1.41  1.0  0.52  0.0  -0.52  -1.0  -1.41  -1.73  -1.93
```



In [4]:

```julia
n = 12
A = adjacent_matrix_of_type_A(n)
V = eigenvectors_of_type_A(n)
α = eigenvalues_of_type_A(n)
display(A)
display(round.(V\(A*V), digits=1))
display(round.(diag(V\(A*V))', digits=2))
display(round.(α', digits=2))

PP = []
for j in 1:n
    P = plot(V[:,j], title="j = $j", legend=false)
    push!(PP, P)
end
plot(PP..., size=(700, 360), legend=false) |> display
```

```
12×12 SymTridiagonal{Int64, Vector{Int64}}:
 0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0

12×12 Matrix{Float64}:
  1.9   0.0   0.0  -0.0  -0.0   0.0  -0.0  -0.0  -0.0  -0.0  -0.0  -0.0
  0.0   1.8  -0.0  -0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
 -0.0   0.0   1.5   0.0   0.0  -0.0  -0.0  -0.0  -0.0  -0.0  -0.0   0.0
 -0.0  -0.0  -0.0   1.1   0.0   0.0   0.0  -0.0   0.0   0.0   0.0   0.0
 -0.0   0.0  -0.0   0.0   0.7  -0.0  -0.0   0.0   0.0  -0.0  -0.0   0.0
  0.0  -0.0   0.0  -0.0  -0.0   0.2   0.0  -0.0  -0.0   0.0   0.0  -0.0
  0.0   0.0   0.0   0.0   0.0  -0.0  -0.2   0.0  -0.0  -0.0  -0.0   0.0
  0.0  -0.0   0.0  -0.0   0.0   0.0   0.0  -0.7   0.0   0.0   0.0  -0.0
 -0.0   0.0  -0.0   0.0  -0.0  -0.0  -0.0   0.0  -1.1  -0.0  -0.0   0.0
  0.0  -0.0   0.0   0.0  -0.0  -0.0   0.0  -0.0   0.0  -1.5   0.0  -0.0
 -0.0   0.0  -0.0  -0.0   0.0   0.0  -0.0   0.0   0.0  -0.0  -1.8  -0.0
  0.0  -0.0   0.0   0.0   0.0  -0.0  -0.0  -0.0  -0.0  -0.0   0.0  -1.9

1×12 Matrix{Float64}:
 1.94  1.77  1.5  1.14  0.71  0.24  -0.24  -0.71  -1.14  -1.5  -1.77  -1.94

1×12 Matrix{Float64}:
 1.94  1.77  1.5  1.14  0.71  0.24  -0.24  -0.71  -1.14  -1.5  -1.77  -1.94
```



1.8 C_∞ 型

ベクトル $v = [x_j]_{j\in\mathbb{Z}} \neq 0$ が A_∞ 型隣接行列の固有値 α の固有ベクトルになるための必要十分条件は

$$x_{j-1} + x_{j+1} = \alpha x_j$$

なので, もしも $x_{-j} = x_j$ が成立しているならば,

$$2x_1 = \alpha x_0$$

となるので, ベクトル $[x_j]_{j=0}^\infty$ は C_∞ 型隣接行列の固有値 α の固有ベクトルになる.

ゆえに, C_∞ 型の隣接行列の固有値 $\alpha = z + z^{-1} \neq \pm 2$ $(z \neq \pm 1)$の固有空間の基底として $\left[z^j + z^{-j}\right]_{j=0}^\infty$ が取れ, 固有値 $\alpha = \pm 2$ の固有空間の基底として $\left[(\pm 1)^j\right]_{j=0}^\infty$ が取れる.

1.9 C_n 型

C_n 型の隣接行列は次の形の $n \times n$ 行列になる:

$$
A = \begin{bmatrix} 0 & 2 & & & \\ 1 & 0 & 1 & & \\ & 1 & 0 & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & 0 \end{bmatrix}.
$$

以下ではこれを $\{0, 1, \cdots, n-1\} \times \{0, 1, \cdots, n-1\}$ 行列とみなす.

C_∞ 型の隣接行列の固有ベクトル $\left[z^k + z^{-k}\right]_{k=0}^{\infty}$ について, $z^n + z^{-n} = 0$ が成立しているとき, ゆえに特に $z = e^{i\cdot(2j+1)\pi/(2n)}$, $j = 0, 1, \dots, n-1$ のとき, ベクトル

$$
\left[z^k + z^{-k}\right]_{k=0}^{n-1} = 2\left[\cos\frac{k(2j+1)\pi}{2n}\right]_{k=0}^{n-1}
$$

は C_n 型隣接行列の固有値

$$
z + z^{-1} = 2\cos\frac{(2j+1)\pi}{2n}
$$

の固有ベクトルである. そこで, θ_j を次のように定める:

$$
\theta_j = \frac{(2j+1)\pi}{2n}.
$$

このとき, C_n 型の隣接行列の互いに異なる n 個の固有値と対応する固有ベクトルとして以下が取れる:

$$
\alpha_j = 2\cos\theta_j, \qquad v_j = \begin{bmatrix} \cos(0\theta_j) \\ \cos(1\theta_j) \\ \cos(2\theta_j) \\ \vdots \\ \cos((n-1)\theta_j) \end{bmatrix} \quad (j = 0, 1, \dots, n-1).
$$

以上の結果を直接確認したい場合には, $\cos((k\pm 1)\theta) = \cos\theta\cos(k\theta) \mp \sin\theta\sin(k\theta)$ より

$$
\cos((k-1)\theta) + \cos((k+1)\theta) = 2\cos\theta\ \cos(k\theta)
$$

となることなどに注意せよ.

In [5]:

```julia
function adjacent_matrix_of_type_C(n)
    @assert n ≥ 2
    Tridiagonal(ones(Int,n-1), zeros(Int,n), [2; ones(Int,n-2)])
end

function eigenvectors_of_type_C(n)
    V = zeros(n, n)
    for j in 1:n
        θ_j = (2j-1)*π/(2n)
        for i in 1:n
            V[i,j] = cos((i-1)*θ_j)
        end
    end
    V
end

function eigenvalues_of_type_C(n)
    α = zeros(n)
    for j in 1:n
        θ_j = (2j-1)/n*π/2
        α[j] = 2cos(θ_j)
    end
    α
end
```

Out[5]:

```
eigenvalues_of_type_C (generic function with 1 method)
```

In [6]:

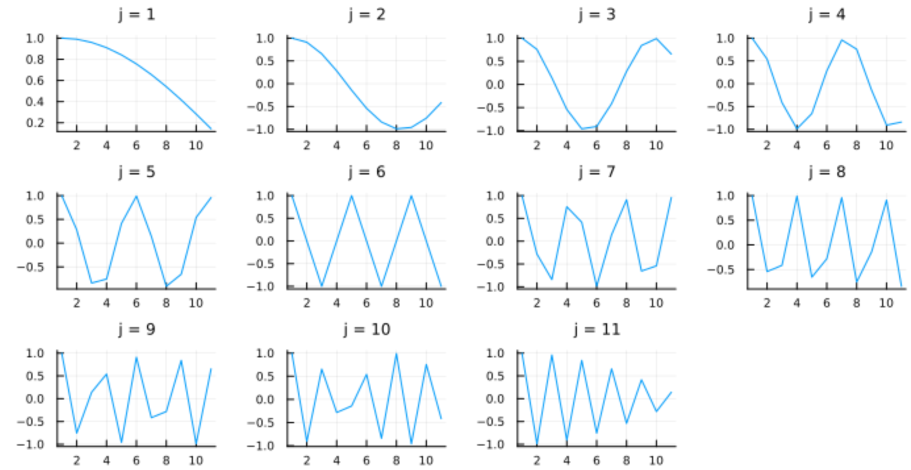
```julia
n = 11
A = adjacent_matrix_of_type_C(n)
V = eigenvectors_of_type_C(n)
α = eigenvalues_of_type_C(n)
display(A)
display(round.(V\(A*V), digits=1))
display(round.(diag(V\(A*V))', digits=2))
display(round.(α', digits=2))

PP = []
for j in 1:n
    P = plot(V[:,j], title="j = $j", legend=false)
    push!(PP, P)
end
plot(PP..., size=(700, 360), legend=false) |> display
```

```
11×11 Tridiagonal{Int64, Vector{Int64}}:
 0  2  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0
```

```
11×11 Matrix{Float64}:
  2.0   0.0  -0.0   0.0  -0.0  -0.0   0.0   0.0  -0.0   0.0   0.0
  0.0   1.8   0.0  -0.0  -0.0  -0.0  -0.0  -0.0   0.0  -0.0  -0.0
  0.0   0.0   1.5   0.0  -0.0   0.0   0.0   0.0   0.0   0.0  -0.0
  0.0  -0.0  -0.0   1.1   0.0  -0.0  -0.0  -0.0   0.0  -0.0   0.0
  0.0  -0.0  -0.0   0.0   0.6   0.0   0.0   0.0  -0.0   0.0   0.0
 -0.0  -0.0  -0.0  -0.0   0.0  -0.0  -0.0  -0.0   0.0  -0.0   0.0
 -0.0   0.0   0.0   0.0  -0.0   0.0  -0.6   0.0  -0.0   0.0  -0.0
 -0.0  -0.0  -0.0  -0.0   0.0  -0.0  -0.0  -1.1   0.0  -0.0   0.0
  0.0  -0.0  -0.0   0.0   0.0   0.0   0.0   0.0  -1.5   0.0   0.0
 -0.0  -0.0   0.0  -0.0   0.0  -0.0  -0.0  -0.0   0.0  -1.8  -0.0
  0.0   0.0  -0.0   0.0  -0.0  -0.0   0.0   0.0   0.0   0.0  -2.0
```

```
1×11 Matrix{Float64}:
 1.98  1.82  1.51  1.08  0.56  -0.0  -0.56  -1.08  -1.51  -1.82  -1.98
```

```
1×11 Matrix{Float64}:
 1.98  1.82  1.51  1.08  0.56  0.0  -0.56  -1.08  -1.51  -1.82  -1.98
```



In [7]:

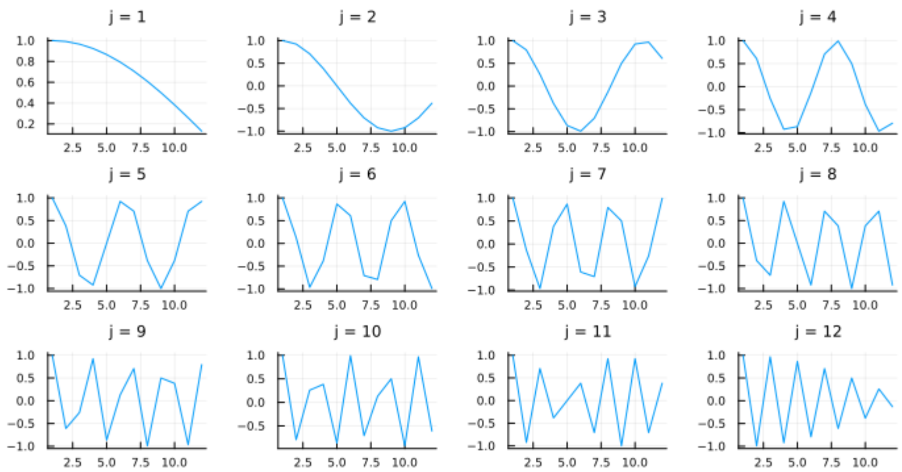
```julia
n = 12
A = adjacent_matrix_of_type_C(n)
V = eigenvectors_of_type_C(n)
α = eigenvalues_of_type_C(n)
display(A)
display(round.(V\(A*V), digits=1))
display(round.(diag(V\(A*V))', digits=2))
display(round.(α', digits=2))

PP = []
for j in 1:n
    P = plot(V[:,j], title="j = $j", legend=false)
    push!(PP, P)
end
plot(PP..., size=(700, 360), legend=false) |> display
```

```
12×12 Tridiagonal{Int64, Vector{Int64}}:
 0  2  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0
```

```
12×12 Matrix{Float64}:
  2.0  -0.0   0.0   0.0  -0.0   0.0   0.0   0.0  -0.0   0.0   0.0  -0.0
 -0.0   1.8   0.0  -0.0   0.0  -0.0   0.0  -0.0   0.0   0.0  -0.0   0.0
 -0.0  -0.0   1.6  -0.0  -0.0   0.0  -0.0  -0.0  -0.0  -0.0   0.0  -0.0
  0.0  -0.0  -0.0   1.2   0.0  -0.0   0.0  -0.0   0.0  -0.0   0.0   0.0
 -0.0   0.0   0.0   0.0   0.8   0.0   0.0   0.0  -0.0   0.0   0.0   0.0
  0.0  -0.0  -0.0   0.0   0.0   0.3   0.0  -0.0  -0.0  -0.0   0.0   0.0
  0.0   0.0   0.0  -0.0  -0.0   0.0  -0.3   0.0   0.0   0.0  -0.0  -0.0
  0.0  -0.0  -0.0   0.0   0.0  -0.0   0.0  -0.8  -0.0  -0.0   0.0  -0.0
  0.0   0.0   0.0   0.0  -0.0   0.0  -0.0   0.0  -1.2   0.0  -0.0   0.0
 -0.0  -0.0  -0.0  -0.0   0.0   0.0  -0.0  -0.0  -0.0  -1.6  -0.0  -0.0
  0.0   0.0   0.0   0.0   0.0  -0.0   0.0   0.0   0.0  -0.0  -1.8   0.0
 -0.0  -0.0   0.0  -0.0  -0.0  -0.0  -0.0  -0.0  -0.0  -0.0  -0.0  -2.0
```

```
1×12 Matrix{Float64}:
 1.98  1.85  1.59  1.22  0.77  0.26  -0.26  -0.77  -1.22  -1.59  -1.85  -1.98
```

```
1×12 Matrix{Float64}:
 1.98  1.85  1.59  1.22  0.77  0.26  -0.26  -0.77  -1.22  -1.59  -1.85  -1.98
```



1.10 B_∞ 型

ベクトル $v = [x_k]_{k=0}^{\infty} \neq 0$ が C_∞ 型隣接行列の固有値 α の固有ベクトルであることの必要十分条件は

$$2x_1 = \alpha x_0,\ x_0 + x_2 = \alpha x_1,\ x_1 + x_3 = \alpha x_2,\ x_2 + x_4 = \alpha x_3,\ \dots$$

が成立することであり, これは

$$x_1 = \alpha \frac{x_0}{2},\ 2\frac{x_0}{2} + x_2 = \alpha x_1,\ x_1 + x_3 = \alpha x_2,\ x_2 + x_4 = \alpha x_3,\ \dots$$

と同値であり, v の最初の成分を半分にしたものは B_n 型隣接行列の固有値になっていることがわかる.

ゆえに, B_∞ 型の隣接行列の固有値 $\alpha = z + z^{-1} \neq \pm 2$ $(z \neq \pm 1)$の固有空間の基底として

$$\begin{bmatrix} 1 \\ z + z^{-1} \\ z^2 + z^{-2} \\ z^3 + z^{-3} \\ \vdots \end{bmatrix}$$

が取れ, 固有値 $\alpha = \pm 2$ の固有空間の基底として

$$\begin{bmatrix} 1/2 \\ -1 \\ 1 \\ -1 \\ 1 \\ \vdots \end{bmatrix}$$

が取れる.

1.11 B_n 型

B_n 型の隣接行列は次の形の $n \times n$ 行列になる:

$$A = \begin{bmatrix} 0 & 1 & & & \\ 2 & 0 & 1 & & \\ & 1 & 0 & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & 0 \end{bmatrix}.$$

以下ではこれを $\{0, 1, \cdots, n-1\} \times \{0, 1, \cdots, n-1\}$ 行列とみなす.

θ_j を次のように定める:

$$\theta_j = \frac{(2j+1)\pi}{2n}.$$

前節の結果を使うと, B_n 型の隣接行列の互いに異なる n 個の固有値と対応する固有ベクトルとして以下が取れることがわかる:

$$\alpha_j = 2\cos\theta_j, \quad v_j = \begin{bmatrix} 1/2 \\ \cos(1\theta_j) \\ \cos(2\theta_j) \\ \vdots \\ \cos((n-1)\theta_j) \end{bmatrix} \quad (j = 0, 1, \dots, n-1).$$

この固有ベクトルは C_n 型の場合の固有ベクトルの第1成分を半分にしたものになっている. $v = [x_k]_{k=0}^{n-1} \neq 0$ が C_n 型の隣接行列の固有値 α の固有ベクトルであることの必要十分条件は,

$$2x_1 = \alpha x_0, \ x_0 + x_2 = \alpha x_1, \ \dots, \ x_{n-3} + x_{n-1} = \alpha x_{n-2}, \ x_{n-2} = \alpha x_{n-1}$$

が成立することであり, これは

$$x_1 = \alpha\frac{x_0}{2}, \ 2\frac{x_0}{2} + x_2 = \alpha x_1, \ \dots, \ x_{n-3} + x_{n-1} = \alpha x_{n-2}, \ x_{n-2} = \alpha x_{n-1}$$

と同値であるので, v の最初の成分を半分にしたものは B_n 型の隣接行列の固有ベクトルになっていることがわかる.

In [8]:

```
function adjacent_matrix_of_type_B(n)
    @assert n ≥ 2
    Tridiagonal([2; ones(Int,n-2)], zeros(Int,n), ones(Int,n-1))
end

function eigenvectors_of_type_B(n)
    V = zeros(n, n)
    for j in 1:n
        θ_j = (2j-1)*π/(2n)
        V[1,j] = 1/2
        for i in 2:n
            V[i,j] = cos((i-1)*θ_j)
        end
    end
    V
end

function eigenvalues_of_type_B(n)
    α = zeros(n)
    for j in 1:n
        θ_j = (2j-1)/n*π/2
        α[j] = 2cos(θ_j)
    end
    α
end
```

Out[8]:

```
eigenvalues_of_type_B (generic function with 1 method)
```

In [9]:

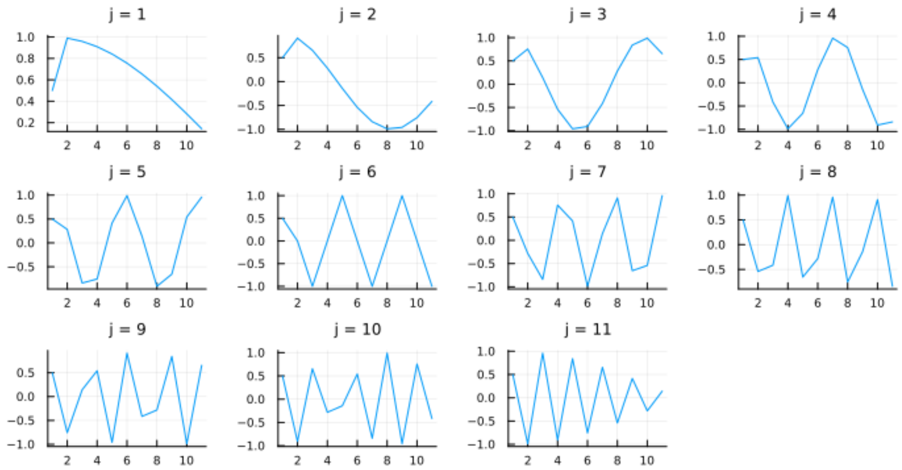
```julia
n = 11
A = adjacent_matrix_of_type_B(n)
V = eigenvectors_of_type_B(n)
α = eigenvalues_of_type_B(n)
display(A)
display(round.(V\(A*V), digits=1))
display(round.(diag(V\(A*V))', digits=2))
display(round.(α', digits=2))

PP = []
for j in 1:n
    P = plot(V[:,j], title="j = $j", legend=false)
    push!(PP, P)
end
plot(PP..., size=(700, 360), legend=false) |> display
```

```
11×11 Tridiagonal{Int64, Vector{Int64}}:
 0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 2  0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0
```

```
11×11 Matrix{Float64}:
  2.0  -0.0   0.0   0.0   0.0  -0.0   0.0   0.0  -0.0   0.0   0.0
  0.0   1.8  -0.0  -0.0  -0.0  -0.0  -0.0  -0.0   0.0  -0.0  -0.0
  0.0   0.0   1.5   0.0  -0.0   0.0   0.0   0.0   0.0   0.0  -0.0
 -0.0  -0.0   0.0   1.1   0.0  -0.0  -0.0  -0.0   0.0  -0.0   0.0
 -0.0   0.0  -0.0   0.0   0.6   0.0   0.0   0.0  -0.0   0.0   0.0
 -0.0  -0.0  -0.0  -0.0   0.0  -0.0  -0.0  -0.0   0.0  -0.0   0.0
 -0.0   0.0   0.0   0.0  -0.0   0.0  -0.6   0.0  -0.0   0.0  -0.0
 -0.0  -0.0  -0.0  -0.0   0.0  -0.0  -0.0  -1.1   0.0  -0.0   0.0
  0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0  -1.5   0.0   0.0
 -0.0  -0.0   0.0  -0.0   0.0  -0.0  -0.0  -0.0  -0.0  -1.8   0.0
  0.0   0.0  -0.0   0.0  -0.0  -0.0   0.0   0.0   0.0   0.0  -2.0
```

```
1×11 Matrix{Float64}:
 1.98  1.82  1.51  1.08  0.56  -0.0  -0.56  -1.08  -1.51  -1.82  -1.98
```

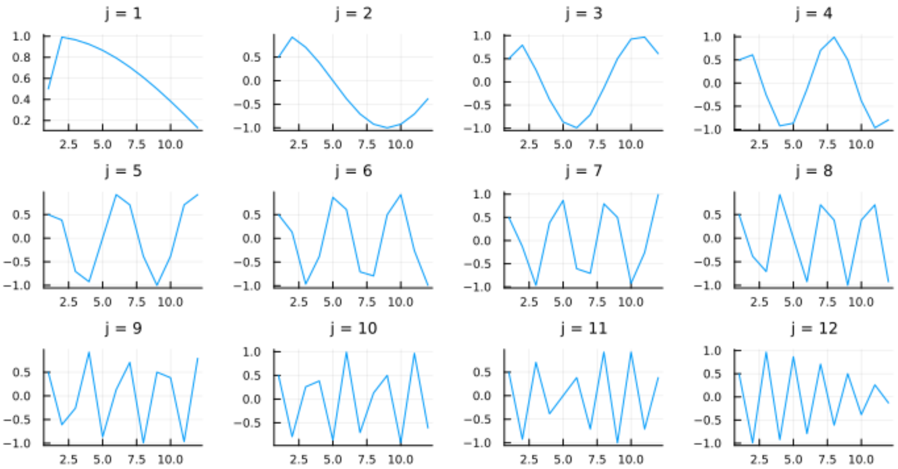
```
1×11 Matrix{Float64}:
 1.98  1.82  1.51  1.08  0.56  0.0  -0.56  -1.08  -1.51  -1.82  -1.98
```



In [10]:

```julia
n = 12
A = adjacent_matrix_of_type_B(n)
V = eigenvectors_of_type_B(n)
α = eigenvalues_of_type_B(n)
display(A)
display(round.(V\(A*V), digits=1))
display(round.(diag(V\(A*V))', digits=2))
display(round.(α', digits=2))

PP = []
for j in 1:n
    P = plot(V[:,j], title="j = $j", legend=false)
    push!(PP, P)
end
plot(PP..., size=(700, 360), legend=false) |> display
```

```
12×12 Tridiagonal{Int64, Vector{Int64}}:
 0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 2  0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1  ⋅
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0  1
 ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  ⋅  1  0

12×12 Matrix{Float64}:
  2.0  -0.0   0.0   0.0  -0.0   0.0   0.0   0.0  -0.0   0.0   0.0  -0.0
 -0.0   1.8  -0.0  -0.0   0.0  -0.0   0.0  -0.0   0.0  -0.0  -0.0  -0.0
 -0.0   0.0   1.6  -0.0  -0.0   0.0  -0.0   0.0  -0.0   0.0   0.0  -0.0
  0.0  -0.0   0.0   1.2   0.0  -0.0   0.0  -0.0   0.0  -0.0   0.0   0.0
  0.0   0.0   0.0   0.0   0.8   0.0   0.0   0.0  -0.0   0.0   0.0  -0.0
  0.0  -0.0  -0.0   0.0   0.0   0.3   0.0  -0.0  -0.0  -0.0   0.0   0.0
  0.0   0.0   0.0  -0.0  -0.0   0.0  -0.3   0.0  -0.0   0.0  -0.0   0.0
  0.0  -0.0  -0.0   0.0   0.0  -0.0  -0.0  -0.8  -0.0  -0.0  -0.0  -0.0
  0.0   0.0   0.0  -0.0  -0.0   0.0  -0.0   0.0  -1.2   0.0   0.0   0.0
 -0.0  -0.0  -0.0  -0.0   0.0   0.0  -0.0  -0.0  -0.0  -1.6   0.0  -0.0
  0.0   0.0   0.0   0.0   0.0  -0.0   0.0   0.0  -0.0   0.0  -1.8  -0.0
 -0.0  -0.0   0.0  -0.0  -0.0  -0.0  -0.0  -0.0   0.0   0.0   0.0  -2.0

1×12 Matrix{Float64}:
 1.98  1.85  1.59  1.22  0.77  0.26  -0.26  -0.77  -1.22  -1.59  -1.85  -1.98

1×12 Matrix{Float64}:
 1.98  1.85  1.59  1.22  0.77  0.26  -0.26  -0.77  -1.22  -1.59  -1.85  -1.98
```



1.12 D_∞ 型

ベクトル $v = [x_k]_{k=0',0,1,2,\dots} \neq 0$ が D_∞ 型隣接行列の固有値 α の固有ベクトルになるための必要十分条件は

$$x_1 = \alpha x_{0'},\ x_1 = \alpha x_0,\ x_{0'} + x_0 + x_2 = \alpha x_1,\ x_1 + x_3 = \alpha x_2,\ x_2 + x_4 = \alpha x_3,\ \dots$$

が成立することである. 一方, ベクトル $w = [y_k]_{k=0}^\infty \neq 0$ が B_∞ 型隣接行列の固有値 α の固有ベクトルになるための必要十分条件は,

$$y_1 = \alpha y_0,\ 2y_0 + y_2 = \alpha y_1,\ y_1 + y_3 = \alpha y_2,\ y_2 + y_4 = \alpha y_3,\ \dots$$

が成立することである. B_∞ 型隣接行列の固有値 α の固有ベクトル $w = [y_k]_{k=0}^\infty \neq 0$ に対して, $x_{0'} = y_0, x_k = y_k$ とおくと, ベクトル $v = [x_k]_{k=0',0,1,2,\dots} \neq 0$ は D_∞ 型隣接行列の固有値 α の固有ベクトルになる.

$$x_1 = \alpha\frac{x_0}{2},\ 2\frac{x_0}{2} + x_2 = \alpha x_1,\ x_1 + x_3 = \alpha x_2,\ x_2 + x_4 = \alpha x_3,\ \dots$$

と同値であり, v の最初の成分を半分にしたものは B_n 型隣接行列の固有値になっていることがわかる.

ゆえに, D_∞ 型の隣接行列の固有値 $\alpha = z + z^{-1} \neq \pm 2$ $(z \neq \pm 1)$の固有空間の基底として

$$\begin{bmatrix} 1 \\ 1 \\ z+z^{-1} \\ z^2+z^{-2} \\ z^3+z^{-3} \\ \vdots \end{bmatrix}$$

が取れ, 固有値 $\alpha = \pm 2$ の固有空間の基底として

$$\begin{bmatrix} 1/2 \\ 1/2 \\ -1 \\ 1 \\ -1 \\ 1 \\ \vdots \end{bmatrix}$$

が取れる.

以上のようにして作られた D_∞ 型隣接行列の固有ベクトルでは最初の2つの成分が等しくなる. そうではない固有値 0 の固有ベクトルを

$$v = \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \\ \vdots \end{bmatrix} = \left[\delta_{k,0'} - \delta_{k,0}\right]_{k=0',0,1,2,\ldots}$$

によって作ることができる.

1.13 D_{n+1} 型

次の形の $(n+1)\times(n+1)$ 行列を D_{n+1} 型の**隣接行列**と呼ぶ:

$$A = \begin{bmatrix} 0 & 0 & 1 & & & \\ 0 & 0 & 1 & & & \\ 1 & 1 & 0 & 1 & & \\ & & 1 & 0 & \ddots & \\ & & & \ddots & \ddots & 1 \\ & & & & 1 & 0 \end{bmatrix}.$$

θ_j を次のように定める:

$$\theta_j = \frac{(2j+1)\pi}{2n}.$$

このとき, 前節の結果より, D_{n+1} 型の隣接行列の固有値と対応する固有ベクトルとして以下が取れることがわかる:

$$\begin{aligned} \alpha_j &= 2\cos\theta_j, & v_j &= \begin{bmatrix} 1/2 \\ 1/2 \\ \cos(1\theta_j) \\ \cos(2\theta_j) \\ \vdots \\ \cos((n-1)\theta_j) \end{bmatrix} \quad (j = 0, 1, \ldots, n-1), \\ \alpha_n &= 0, & v_n &= \begin{bmatrix} 1 \\ -1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \end{aligned}$$

$v_0, v_1, \ldots, v_{n-1}$ は B_n 型隣接行列の固有ベクトルの第1成分を重複させたものに等しい. そのことから $v_0, v_1, \ldots, v_{n-1}$ が D_{n+1} 型隣接行列の固有ベクトルであることがわかる. v_n が D_{n+1} 型隣接行列の固有値 0 の固有ベクトルであることは自明である.

$n+1$ が偶数のとき, 0 は D_{n+1} 型隣接行列の固有値として2重に重複している. $n+1$ が奇数のとき, $n+1$ 個の固有値 α_j は互いに異なる.

In [11]:
```julia
function adjacent_matrix_of_type_D(n)
    @assert n ≥ 3
    G = zeros(Int, n, n)
    G[1,3] = 1
    for i in 2:n-1
        G[i,i+1] = 1
    end
    Symmetric(G)
end

function eigenvectors_of_type_D(n)
    V = zeros(n, n)
    for j in 1:n-1
        θ_j = (2j-1)*π/(2*(n-1))
        V[1,j] = V[2,j] = 1/2
        for i in 3:n
            V[i,j] = cos((i-2)*θ_j)
        end
    end
    V[1,n] = 1
    V[2,n] = -1
    V
end

function eigenvalues_of_type_D(n)
    α = zeros(n)
    for j in 1:n-1
        θ_j = (2j-1)/(n-1)*π/2
        α[j] = 2cos(θ_j)
    end
    α[n] = 0
    α
end
```

Out[11]:
```
eigenvalues_of_type_D (generic function with 1 method)
```

In [12]:

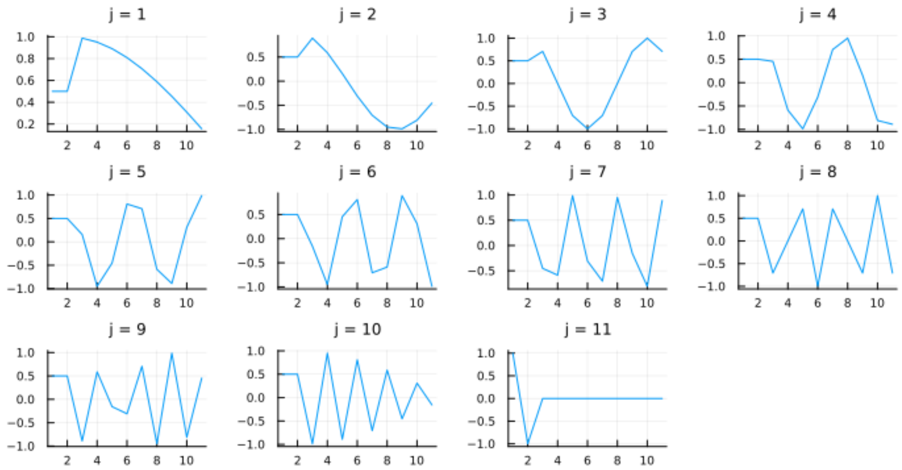
```julia
n = 11
A = adjacent_matrix_of_type_D(n)
V = eigenvectors_of_type_D(n)
α = eigenvalues_of_type_D(n)
display(A)
display(round.(V\(A*V), digits=1))
display(round.(diag(V\(A*V))', digits=2))
display(round.(α', digits=2))

PP = []
for j in 1:n
    P = plot(V[:,j], title="j = $j", legend=false)
    push!(PP, P)
end
plot(PP..., size=(700, 360), legend=false) |> display
```

```
11×11 Symmetric{Int64, Matrix{Int64}}:
 0  0  1  0  0  0  0  0  0  0  0
 0  0  1  0  0  0  0  0  0  0  0
 1  1  0  1  0  0  0  0  0  0  0
 0  0  1  0  1  0  0  0  0  0  0
 0  0  0  1  0  1  0  0  0  0  0
 0  0  0  0  1  0  1  0  0  0  0
 0  0  0  0  0  1  0  1  0  0  0
 0  0  0  0  0  0  1  0  1  0  0
 0  0  0  0  0  0  0  1  0  1  0
 0  0  0  0  0  0  0  0  1  0  1
 0  0  0  0  0  0  0  0  0  1  0
```

```
11×11 Matrix{Float64}:
  2.0  -0.0  -0.0  -0.0  -0.0  -0.0   0.0  -0.0  -0.0   0.0   0.0
  0.0   1.8  -0.0  -0.0   0.0  -0.0  -0.0  -0.0   0.0  -0.0  -0.0
  0.0   0.0   1.4   0.0  -0.0   0.0   0.0   0.0  -0.0  -0.0  -0.0
 -0.0   0.0   0.0   0.9   0.0  -0.0   0.0  -0.0  -0.0   0.0  -0.0
 -0.0   0.0  -0.0   0.0   0.3   0.0  -0.0  -0.0  -0.0   0.0  -0.0
  0.0  -0.0   0.0  -0.0   0.0  -0.3   0.0   0.0   0.0  -0.0  -0.0
  0.0   0.0   0.0   0.0  -0.0   0.0  -0.9   0.0   0.0   0.0  -0.0
  0.0  -0.0  -0.0  -0.0  -0.0  -0.0   0.0  -1.4   0.0  -0.0   0.0
 -0.0   0.0   0.0   0.0   0.0   0.0  -0.0  -0.0  -1.8  -0.0   0.0
  0.0  -0.0  -0.0  -0.0  -0.0  -0.0   0.0  -0.0   0.0  -2.0   0.0
  0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
```

```
1×11 Matrix{Float64}:
 1.98  1.78  1.41  0.91  0.31  -0.31  -0.91  -1.41  -1.78  -1.98  0.0
```

```
1×11 Matrix{Float64}:
 1.98  1.78  1.41  0.91  0.31  -0.31  -0.91  -1.41  -1.78  -1.98  0.0
```



In [13]:

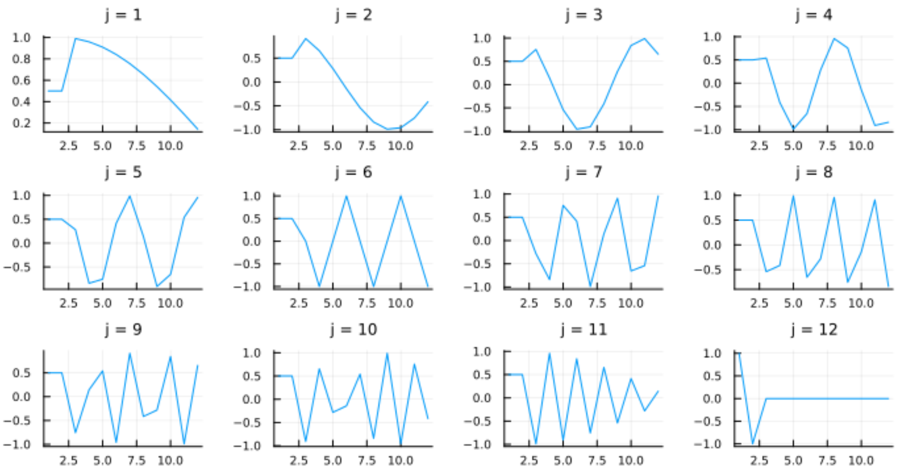
```
n = 12
A = adjacent_matrix_of_type_D(n)
V = eigenvectors_of_type_D(n)
α = eigenvalues_of_type_D(n)
display(A)
display(round.(V\(A*V), digits=1))
display(round.(diag(V\(A*V))', digits=2))
display(round.(α', digits=2))

PP = []
for j in 1:n
    P = plot(V[:,j], title="j = $j", legend=false)
    push!(PP, P)
end
plot(PP..., size=(700, 360), legend=false) ▷ display
```

```
12×12 Symmetric{Int64, Matrix{Int64}}:
 0  0  1  0  0  0  0  0  0  0  0  0
 0  0  1  0  0  0  0  0  0  0  0  0
 1  1  0  1  0  0  0  0  0  0  0  0
 0  0  1  0  1  0  0  0  0  0  0  0
 0  0  0  1  0  1  0  0  0  0  0  0
 0  0  0  0  1  0  1  0  0  0  0  0
 0  0  0  0  0  1  0  1  0  0  0  0
 0  0  0  0  0  0  1  0  1  0  0  0
 0  0  0  0  0  0  0  1  0  1  0  0
 0  0  0  0  0  0  0  0  1  0  1  0
 0  0  0  0  0  0  0  0  0  1  0  1
 0  0  0  0  0  0  0  0  0  0  1  0
```

```
12×12 Matrix{Float64}:
  2.0  -0.0   0.0   0.0   0.0  -0.0   0.0   0.0  -0.0   0.0   0.0   0.0
  0.0   1.8  -0.0  -0.0  -0.0  -0.0  -0.0  -0.0   0.0  -0.0  -0.0  -0.0
  0.0   0.0   1.5   0.0  -0.0   0.0   0.0   0.0   0.0   0.0  -0.0  -0.0
 -0.0  -0.0   0.0   1.1   0.0  -0.0  -0.0  -0.0   0.0  -0.0   0.0  -0.0
 -0.0   0.0  -0.0   0.0   0.6   0.0   0.0   0.0  -0.0   0.0  -0.0  -0.0
 -0.0  -0.0  -0.0  -0.0   0.0   0.0  -0.0  -0.0   0.0  -0.0   0.0  -0.0
 -0.0   0.0   0.0   0.0  -0.0   0.0  -0.6   0.0  -0.0   0.0  -0.0   0.0
 -0.0  -0.0   0.0  -0.0   0.0  -0.0  -0.0  -1.1   0.0  -0.0   0.0  -0.0
  0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0  -1.5   0.0   0.0   0.0
 -0.0  -0.0   0.0  -0.0   0.0  -0.0  -0.0  -0.0  -0.0  -1.8  -0.0  -0.0
  0.0   0.0  -0.0   0.0  -0.0  -0.0   0.0   0.0   0.0  -0.0  -2.0   0.0
 -0.0  -0.0  -0.0  -0.0  -0.0  -0.0  -0.0  -0.0  -0.0  -0.0  -0.0  -0.0
```

```
1×12 Matrix{Float64}:
 1.98  1.82  1.51  1.08  0.56  0.0  -0.56  -1.08  -1.51  -1.82  -1.98  -0.0
```

```
1×12 Matrix{Float64}:
 1.98  1.82  1.51  1.08  0.56  0.0  -0.56  -1.08  -1.51  -1.82  -1.98  0.0
```



2 古典アフィン型の場合

2.1 $A_{n-1}^{(1)}$ 型

$n \geqq 3$ と仮定する. $A_{n-1}^{(1)}$ 型の隣接行列とは次の形の $n \times n$ 行列のことであると定める:

$$\begin{bmatrix} 0 & 1 & & & 1 \\ 1 & 0 & 1 & & \\ & 1 & 0 & \ddots & \\ & & \ddots & \ddots & 1 \\ 1 & & & 1 & 0 \end{bmatrix}.$$

以下ではこの行列を $\{0, 1, \dots, n-1\} \times \{0, 1, \dots, n-1\}$ 行列とみなす.

ベクトル $[x_k]_{k \in \mathbb{Z}} \neq 0$ が A_∞ 型隣接行列の固有値 α の固有ベクトルのとき, $[x_k]_{k=0}^{n-1}$ が $A_n^{(1)}$ 型の隣接行列の固有ベクトルになるための必要十分条件は周期境界条件 $x_{k+n} = x_k$ が成立していることである. このことと, A_∞ 型隣接行列の固有値・固有ベクトルに関する結果より以下が得られる. ζ を

$$\zeta = e^{i\theta} = e^{\frac{2\pi i}{n}}$$

と定める. このとき, $A_{n-1}^{(1)}$ 型隣接行列は以下の固有値 α_j と対応する固有ベクトル v_j を持つ:

$$\alpha_j = \zeta^j + \zeta^{-j} = 2\cos\frac{2\pi j}{n}, \quad v_j = \left[\zeta^{jk}\right]_{k=0}^{n-1}.$$

ここで $j = 0, 1, \dots, n-1$ である.

$\alpha_{n-j} = \alpha_j$ となっていることに注意せよ. n が奇数のとき, $n = 2m+1$ とおくと,

$$\alpha_1 = \alpha_{2m},\ \alpha_2 = \alpha_{2m-1},\ \dots,\ \alpha_m = \alpha_{m+1}$$

であり, n が偶数のとき, $n = 2m$ とおくと,

$$\alpha_1 = \alpha_{2m-1},\ \alpha_2 = \alpha_{2m-2},\ \dots,\ \alpha_{m-1} = \alpha_{m+1}.$$

2.2 $C_n^{(1)}$ 型

$C_n^{(1)}$ 型の隣接行列とは次の形の $(n+1)\times(n+1)$ 行列のことであると定める:

$$\begin{bmatrix} 0 & 2 & & & & \\ 1 & 0 & 1 & & & \\ & 1 & 0 & \ddots & & \\ & & \ddots & \ddots & 1 & \\ & & & 1 & 0 & 1 \\ & & & & 2 & 0 \end{bmatrix}.$$

以下ではこの行列を $\{0, 1, \dots, n\} \times \{0, 1, \dots, n\}$ 行列とみなす.

ベクトル $v = [x_k]_{k=0}^{2n-1}$ が $A_{2n-1}^{(1)}$ 型隣接行列の固有値 α の固有ベクトルであるとする. そのとき, さらに $x_{2n-1} = x_1$ かつ $x_{n+1} = x_{n-1}$ ならば, ベクトル $[x_k]_{k=0}^{n}$ は $C_n^{(1)}$ 型隣接行列の固有値 α の固有ベクトルになる.

ゆえに, $C_n^{(1)}$ 型隣接行列は以下の固有値 α_j と対応する固有ベクトル v_j を持つ:

$$\alpha_j = 2\cos\frac{j\pi}{n}, \quad v_j = \left[\cos\frac{kj\pi}{n}\right]_{k=0}^{n}.$$

2.3 $A_{2n+2}^{(2)}$ 型

$A_{2n+2}^{(2)}$ 型の隣接行列とは次の形の $(n+1)\times(n+1)$ 行列のことであると定める:

$$\begin{bmatrix} 0 & 1 & & & & \\ 2 & 0 & 1 & & & \\ & 1 & 0 & \ddots & & \\ & & \ddots & \ddots & 1 & \\ & & & 1 & 0 & 1 \\ & & & & 2 & 0 \end{bmatrix}.$$

以下ではこの行列を $\{0, 1, \dots, n\} \times \{0, 1, \dots, n\}$ 行列とみなす.

ベクトル $v = [x_k]_{k=0}^{n}$ が $C_n^{(1)}$ 型隣接行列の固有値 α の固有ベクトルならば, v の最初の成分 x_0 を半分にしたものは $A_{2n+2}^{(2)}$ 型隣接行列の固有値 α の固有ベクトルになる.

2.4 $D_{n+2}^{(2)}$ 型

$D_{n+2}^{(2)}$ 型の隣接行列とは次の形の $(n+1)\times(n+1)$ 行列のことであると定める:

$$\begin{bmatrix} 0 & 1 & & & & \\ 2 & 0 & 1 & & & \\ & 1 & 0 & \ddots & & \\ & & \ddots & \ddots & 1 & \\ & & & 1 & 0 & 2 \\ & & & & 1 & 0 \end{bmatrix}.$$

以下ではこの行列を $\{0, 1, \dots, n\} \times \{0, 1, \dots, n\}$ 行列とみなす.

ベクトル $v = [x_k]_{k=0}^{n}$ が $C_n^{(1)}$ 型隣接行列の固有値 α の固有ベクトルならば, v の最初の成分 x_0 と最後の成分 x_n を半分にしたものは $D_{n+2}^{(2)}$ 型隣接行列の固有値 α の固有ベクトルになる.

In [14]:
```julia
n = 5
A = Matrix(Tridiagonal([ones(Int,n-1);2], zeros(Int, n+1), [2;ones(Int,n-1)]))
display(A)
eigA = eigen(A)
display(round.(eigA.values, digits=3))
V = eigA.vectors
V = V/Diagonal(V[1,:])
display(round.(V, digits=2))
```

```
6×6 Matrix{Int64}:
 0  2  0  0  0  0
 1  0  1  0  0  0
 0  1  0  1  0  0
 0  0  1  0  1  0
 0  0  0  1  0  1
 0  0  0  0  2  0

6-element Vector{Float64}:
 -2.0
 -1.618
 -0.618
  0.618
  1.618
  2.0

6×6 Matrix{Float64}:
  1.0   1.0    1.0    1.0    1.0   1.0
 -1.0  -0.81  -0.31   0.31   0.81  1.0
  1.0   0.31  -0.81  -0.81   0.31  1.0
 -1.0   0.31   0.81  -0.81  -0.31  1.0
  1.0  -0.81   0.31   0.31  -0.81  1.0
 -1.0   1.0   -1.0    1.0   -1.0   1.0
```

In [15]:
```julia
n = 5
A = Matrix(Tridiagonal([2; ones(Int,n-2); 2], zeros(Int, n+1), ones(Int,n)))
display(A)
eigA = eigen(A)
display(round.(eigA.values, digits=3))
V = eigA.vectors
V = V/Diagonal(V[1,:])/2
display(round.(V, digits=2))
```

```
6×6 Matrix{Int64}:
 0  1  0  0  0  0
 2  0  1  0  0  0
 0  1  0  1  0  0
 0  0  1  0  1  0
 0  0  0  1  0  1
 0  0  0  0  2  0

6-element Vector{Float64}:
 -2.0
 -1.618
 -0.618
  0.618
  1.618
  2.0

6×6 Matrix{Float64}:
  0.5   0.5    0.5    0.5    0.5   0.5
 -1.0  -0.81  -0.31   0.31   0.81  1.0
  1.0   0.31  -0.81  -0.81   0.31  1.0
 -1.0   0.31   0.81  -0.81  -0.31  1.0
  1.0  -0.81   0.31   0.31  -0.81  1.0
 -1.0   1.0   -1.0    1.0   -1.0   1.0
```

In [16]:

```julia
n = 5
A = Matrix(Tridiagonal([2;ones(Int,n-1)], zeros(Int, n+1), [ones(Int,n-1);2]))
display(A)
eigA = eigen(A)
display(round.(eigA.values, digits=3))
V = eigA.vectors
V = V/Diagonal(V[1,:])/2
display(round.(V, digits=2))
```

```
6×6 Matrix{Int64}:
 0  1  0  0  0  0
 2  0  1  0  0  0
 0  1  0  1  0  0
 0  0  1  0  1  0
 0  0  0  1  0  2
 0  0  0  0  1  0

6-element Vector{Float64}:
 -2.0
 -1.618
 -0.618
  0.618
  1.618
  2.0

6×6 Matrix{Float64}:
  0.5   0.5    0.5    0.5    0.5   0.5
 -1.0  -0.81  -0.31   0.31   0.81  1.0
  1.0   0.31  -0.81  -0.81   0.31  1.0
 -1.0   0.31   0.81  -0.81  -0.31  1.0
  1.0  -0.81   0.31   0.31  -0.81  1.0
 -0.5   0.5   -0.5    0.5   -0.5   0.5
```

2.5 $A^{(2)}_{2n+3}$ 型

$A^{(2)}_{2n+3}$ 型の隣接行列とは次の形の $(n+2)\times(n+2)$ 行列のことであると定める:

$$
\begin{bmatrix}
0 & 0 & 1 & & & & \\
0 & 0 & 1 & & & & \\
1 & 1 & 0 & 1 & & & \\
& & 1 & 0 & \ddots & & \\
& & & \ddots & \ddots & 1 & \\
& & & & 1 & 0 & 1 \\
& & & & & 2 & 0
\end{bmatrix}.
$$

以下ではこの行列を $\{0', 0, 1, \dots, n\} \times \{0', 0, 1, \dots, n\}$ 行列とみなす.

ベクトル $v = [x_k]_{k=0}^n$ が $C_n^{(1)}$ 型隣接行列の固有値 α の固有ベクトルならば, v の最初の成分 x_0 を半分にしたものを重複させて得られるベクトル

$$
\begin{bmatrix}
x_0/2 \\
x_0/2 \\
x_1 \\
\vdots \\
x_n
\end{bmatrix}
$$

は $A^{(2)}_{2n+3}$ 型の隣接行列の固有値 α の固有ベクトルになる. その他に

$$
\begin{bmatrix}
1 \\
-1 \\
0 \\
\vdots \\
0
\end{bmatrix}
$$

は $A^{(2)}_{2n+3}$ 型の隣接行列の固有値 0 の固有ベクトルになる.

2.6 $B^{(1)}_{n+1}$ 型

$B^{(1)}_{n+1}$ 型の隣接行列とは次の形の $(n+2)\times(n+2)$ 行列のことであると定める:

$$
\begin{bmatrix}
0 & 0 & 1 & & & & \\
0 & 0 & 1 & & & & \\
1 & 1 & 0 & 1 & & & \\
& & 1 & 0 & \ddots & & \\
& & & \ddots & \ddots & 1 & \\
& & & & 1 & 0 & 2 \\
& & & & & 1 & 0
\end{bmatrix}.
$$

以下ではこの行列を $\{0', 0, 1, \dots, n\} \times \{0', 0, 1, \dots, n\}$ 行列とみなす.

ベクトル $v = [x_k]_{k=0}^n$ が $C_n^{(1)}$ 型隣接行列の固有値 α の固有ベクトルならば, v の最初の成分 x_0 を半分にしたものを重複させ, 最後の成分 x_n を半分にして得られるベクトル

$$\begin{bmatrix} x_0/2 \\ x_0/2 \\ x_1 \\ \vdots \\ x_{n-1} \\ x_n/2 \end{bmatrix}$$

は $B_{n+1}^{(1)}$ 型の隣接行列の固有値 α の固有ベクトルになる. その他に

$$\begin{bmatrix} 1 \\ -1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

は $B_{n+1}^{(1)}$ 型の隣接行列の固有値 0 の固有ベクトルになる.

2.7 $D_{n+2}^{(1)}$ 型

$D_{n+2}^{(1)}$ 型の隣接行列とは次の形の $(n+3)\times(n+3)$ 行列のことであると定める:

$$\begin{bmatrix} 0 & 0 & 1 & & & & & \\ 0 & 0 & 1 & & & & & \\ 1 & 1 & 0 & 1 & & & & \\ & & 1 & 0 & \ddots & & & \\ & & & \ddots & \ddots & 1 & & \\ & & & & 1 & 0 & 1 & 1 \\ & & & & & 1 & 0 & 0 \\ & & & & & 1 & 0 & 0 \end{bmatrix}.$$

以下ではこの行列を $\{0', 0, 1, \dots, n, n'\} \times \{0', 0, 1, \dots, n, n'\}$ 行列とみなす.

ベクトル $v = [x_k]_{k=0}^n$ が $C_n^{(1)}$ 型隣接行列の固有値 α の固有ベクトルならば, v の最初の成分 x_0 を半分にしたものを重複させ, 最後の成分 x_n を半分にして重複して得られるベクトル

$$\begin{bmatrix} x_0/2 \\ x_0/2 \\ x_1 \\ \vdots \\ x_{n-1} \\ x_n/2 \\ x_n/2 \end{bmatrix}$$

は $D_{n+2}^{(1)}$ 型の隣接行列の固有値 α の固有ベクトルになる. その他に

$$\begin{bmatrix} 1 \\ -1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ -1 \end{bmatrix},$$

は $B_{n+1}^{(1)}$ 型の隣接行列の固有値 0 の固有ベクトルになる.

3 Chebyshev多項式

3.1 Chebyshev多項式の定義

Chebyshev多項式 (チェビシェフ多項式) の話は本質的に三角函数の n 倍角の公式の話に過ぎない. そこでまず三角函数の n 倍角の公式を母函数の方法を使って計算してみよう.

$\theta, t \in \mathbb{R}$, $|t| < 1$ であるとする. このとき,

$$\sum_{n=0}^{\infty} e^{in\theta} t^n = \frac{1}{1 - e^{i\theta} t} = \frac{1 - e^{-i\theta} t}{(1 - e^{i\theta} t)(1 - e^{-i\theta} t)} = \frac{(1 - t\cos\theta) + it\sin\theta}{1 - 2t\cos\theta + t^2}.$$

ゆえに, 両辺の実部と虚部を比較することによって次を得る:

$$\sum_{n=0}^{\infty} t^n \cos(n\theta) = \frac{1 - t\cos\theta}{1 - 2t\cos\theta + t^2},$$
$$\sum_{n=1}^{\infty} t^n \sin(n\theta) = \frac{t\sin\theta}{1 - 2t\cos\theta + t^2}.$$

後者の式の両辺を $t\sin\theta$ で割れば次が得られる:

$$\sum_{n=0}^{\infty} t^n \frac{\sin((n+1)\theta)}{\sin\theta} = \frac{1}{1 - 2t\cos\theta + t^2}.$$

ゆえに, t についてべき級数展開することによって, x の多項式 $T_n(x)$, $U_n(x)$ を

$$\frac{1 - xt}{1 - 2xt + t^2} = \sum_{n=0}^{\infty} T_n(x) t^n, \qquad \frac{1}{1 - 2xt + t^2} = \sum_{n=0}^{\infty} U_n(x) t^n$$

と定義すると,

$$\cos(n\theta) = T_n(\cos\theta), \qquad \frac{\sin((n+1)\theta)}{\sin\theta} = U_n(\cos\theta)$$

が得らえる. $T_n(x)$ を第1種Chebyshev多項式と呼び, $U_n(x)$ を第2種Chebyshev多項式と呼ぶ.

3.2 Chebyshev多項式の具体形

$$\begin{aligned}
\frac{1}{1 - 2xt + t^2} &= \sum_{m=0}^{\infty} (2xt - t^2)^m \\
&= \sum_{m=0}^{\infty} \sum_{j=0}^{m} (-1)^j \binom{m}{j} 2^{m-j} x^{m-j} t^{m+j} \\
&= \sum_{n=0}^{\infty} \left(\sum_{0 \leqq j \leqq n/2} (-1)^j \binom{n-j}{j} 2^{n-2j} x^{n-2j} \right) t^n \\
\frac{1 - xt}{1 - 2xt + t^2} &= (1 - xt) \frac{1}{1 - 2xt + t^2} \\
&= \sum_{n=0}^{\infty} \left(\sum_{0 \leqq j \leqq n/2} (-1)^j \binom{n-j}{j} 2^{n-2j} x^{n-2j} \right) t^n \\
&+ \sum_{n=0}^{\infty} \left(\sum_{0 \leqq j \leqq n/2} (-1)^{j+1} \binom{n-j}{j} 2^{n-2j} x^{n-2j+1} \right) t^{n+1} \\
&= 1 + \sum_{n=1}^{\infty} \left(\sum_{0 \leqq j \leqq n/2} (-1)^j \binom{n-j}{j} 2^{n-2j} x^{n-2j} \right) t^n \\
&+ \sum_{n=1}^{\infty} \left(\sum_{0 \leqq j \leqq (n-1)/2} (-1)^{j+1} \binom{n-j-1}{j} 2^{n-2j-1} x^{n-2j} \right) t^n \\
&= 1 + \sum_{n=1}^{\infty} \left(\frac{n}{2} \sum_{0 \leqq j \leqq n/2} \frac{(-1)^j}{n-j} \binom{n-j}{j} 2^{n-2j} x^{n-2j} \right) t^n
\end{aligned}$$

より, $T_0(x) = U_0(x) = 1$ でかつ $n > 0$ のとき,

$$T_n(x) = \frac{n}{2} \sum_{0 \leqq j \leqq n/2} \frac{(-1)^j}{n-j} \binom{n-j}{j} 2^{n-2j} x^{n-2j},$$
$$U_n(x) = \sum_{0 \leqq j \leqq n/2} (-1)^j \binom{n-j}{j} 2^{n-2j} x^{n-2j}.$$

$n > 0$ のとき, $T_n(x)$, $U_n(x)$ はそれぞれ最高次の係数が 2^{n-1}, 2^n の多項式であり, それらの函数としての偶奇と n の偶奇は一致する.

In [17]:

```julia
function MyChebyshevT(n, x)
    T = typeof(x)
    (n/T(2)) * sum((-1)^j/T(n-j)*binomial(n-j,j)*(2x)^(n-2j) for j in 0:n÷2)
end

@vars x
N = 10
[MyChebyshevT(n, x) for n in 1:N] |> display
[ChebyshevT(n, x) for n in 1:N] |> display
```

$$\begin{bmatrix} x \\ 2x^2 - 1 \\ 4x^3 - 3x \\ 8x^4 - 8x^2 + 1 \\ 16x^5 - 20x^3 + 5x \\ 32x^6 - 48x^4 + 18x^2 - 1 \\ 64x^7 - 112x^5 + 56x^3 - 7x \\ 128x^8 - 256x^6 + 160x^4 - 32x^2 + 1 \\ 256x^9 - 576x^7 + 432x^5 - 120x^3 + 9x \\ 512x^{10} - 1280x^8 + 1120x^6 - 400x^4 + 50x^2 - 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ 2x^2 - 1 \\ 4x^3 - 3x \\ 8x^4 - 8x^2 + 1 \\ 16x^5 - 20x^3 + 5x \\ 32x^6 - 48x^4 + 18x^2 - 1 \\ 64x^7 - 112x^5 + 56x^3 - 7x \\ 128x^8 - 256x^6 + 160x^4 - 32x^2 + 1 \\ 256x^9 - 576x^7 + 432x^5 - 120x^3 + 9x \\ 512x^{10} - 1280x^8 + 1120x^6 - 400x^4 + 50x^2 - 1 \end{bmatrix}$$

In [18]:

```julia
function MyChebyshevU(n, x)
    sum((-1)^j*binomial(n-j,j)*(2x)^(n-2j) for j in 0:n÷2)
end

@vars x
N = 10
[MyChebyshevU(n, x) for n in 1:N] |> display
[ChebyshevU(n, x) for n in 1:N] |> display
```

$$\begin{bmatrix} 2x \\ 4x^2 - 1 \\ 8x^3 - 4x \\ 16x^4 - 12x^2 + 1 \\ 32x^5 - 32x^3 + 6x \\ 64x^6 - 80x^4 + 24x^2 - 1 \\ 128x^7 - 192x^5 + 80x^3 - 8x \\ 256x^8 - 448x^6 + 240x^4 - 40x^2 + 1 \\ 512x^9 - 1024x^7 + 672x^5 - 160x^3 + 10x \\ 1024x^{10} - 2304x^8 + 1792x^6 - 560x^4 + 60x^2 - 1 \end{bmatrix}$$

$$\begin{bmatrix} 2x \\ 4x^2 - 1 \\ 8x^3 - 4x \\ 16x^4 - 12x^2 + 1 \\ 32x^5 - 32x^3 + 6x \\ 64x^6 - 80x^4 + 24x^2 - 1 \\ 128x^7 - 192x^5 + 80x^3 - 8x \\ 256x^8 - 448x^6 + 240x^4 - 40x^2 + 1 \\ 512x^9 - 1024x^7 + 672x^5 - 160x^3 + 10x \\ 1024x^{10} - 2304x^8 + 1792x^6 - 560x^4 + 60x^2 - 1 \end{bmatrix}$$

3.3 Chebyshev多項式の因数分解と隣接行列の特性多項式の関係

3.3.1 第1種Chebyshev多項式の場合

θ_j 達を

$$\theta_j = \frac{(2j+1)\pi}{2n} \quad (j = 0, 1, \dots, n-1)$$

と定めると, $\cos(n\theta_j) = 0$ となり, $\cos\theta_j$ は互いに異なる. ゆえに $\cos(n\theta) = T_n(\cos\theta)$ かつ $T_n(x)$ が最高次の係数が 2^{n-1} の n 次多項式であることより,

$$T_n(x) = 2^{n-1}\prod_{j=0}^{n-1}\left(x - \cos\theta_j\right) = 2^{n-1}\prod_{j=0}^{n-1}\left(x + \cos\theta_j\right)$$

を得る. この公式の後者の等号は $\cos\theta_{n-1-j} = \cos(\pi - \theta_j) = -\cos\theta_j$ から得られる.

一方, C_n 型の $n \times n$ の隣接行列

$$\begin{bmatrix} 0 & 2 & & & \\ 1 & 0 & 1 & & \\ & 1 & 0 & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & 0 \end{bmatrix}$$

の固有値の全体は $2\cos\theta_j$ $(j = 0, 1, \dots, n-1)$ だったので,

$$\begin{vmatrix} 2x & 2 & & & \\ 1 & 2x & 1 & & \\ & 1 & 2x & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & 2x \end{vmatrix} = \begin{vmatrix} 2x & -2 & & & \\ -1 & 2x & -1 & & \\ & -1 & 2x & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2x \end{vmatrix} = 2T_n(x)$$

左辺は C_n 型隣接行列の特性多項式 A の -1 倍の特性多項式に $2x$ を代入して得られる $|2xE + A|$ であり, A と $-A$ の固有値の全体は一致するので, それは A の特性多項式に $2x$ を代入したもの $|2xE - A|$ に等しいことがわかり, 零点と最高次の係数の一致によって2つ目の等号も成立する. この公式の両辺を 2 で割ると,

$$\begin{vmatrix} x & 1 & & & \\ 1 & 2x & 1 & & \\ & 1 & 2x & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & 2x \end{vmatrix} = T_n(x)$$

も得られる. B_n 型隣接行列は C_n 型隣接行列の転置に等しく, 行列式は転置で不変なので, 以上と同じ結果が B_n 型隣接行列についても得られる. D_{n+1} 型の $(n+1) \times (n+1)$ の隣接行列について同様にすると以下の公式も得られる:

$$\begin{vmatrix} 2x & 0 & 1 & & & \\ 0 & 2x & 1 & & & \\ 1 & 1 & 2x & 1 & & \\ & & 1 & 2x & \ddots & \\ & & & \ddots & \ddots & 1 \\ & & & & 1 & 2x \end{vmatrix} = 4xT_n(x).$$

In [19]:

```julia
function charmat_C(n,x)
    @assert n ≥ 1
    if n ≥ 2
        Diagonal(fill(2x, n)) + adjacent_matrix_of_type_C(n)
    else
        hcat([2x])
    end
end

@vars x
charmat_C(5,x) ▷ display

N = 10
[sympy.det(charmat_C(n,x)).expand() for n in 1:N] ▷ display
[2ChebyshevT(n,x) for n in 1:N] ▷ display
```

$$\begin{bmatrix} 2x & 2 & 0 & 0 & 0 \\ 1 & 2x & 1 & 0 & 0 \\ 0 & 1 & 2x & 1 & 0 \\ 0 & 0 & 1 & 2x & 1 \\ 0 & 0 & 0 & 1 & 2x \end{bmatrix}$$

$$\begin{bmatrix} 2x \\ 4x^2 - 2 \\ 8x^3 - 6x \\ 16x^4 - 16x^2 + 2 \\ 32x^5 - 40x^3 + 10x \\ 64x^6 - 96x^4 + 36x^2 - 2 \\ 128x^7 - 224x^5 + 112x^3 - 14x \\ 256x^8 - 512x^6 + 320x^4 - 64x^2 + 2 \\ 512x^9 - 1152x^7 + 864x^5 - 240x^3 + 18x \\ 1024x^{10} - 2560x^8 + 2240x^6 - 800x^4 + 100x^2 - 2 \end{bmatrix}$$

$$\begin{bmatrix} 2x \\ 4x^2 - 2 \\ 8x^3 - 6x \\ 16x^4 - 16x^2 + 2 \\ 32x^5 - 40x^3 + 10x \\ 64x^6 - 96x^4 + 36x^2 - 2 \\ 128x^7 - 224x^5 + 112x^3 - 14x \\ 256x^8 - 512x^6 + 320x^4 - 64x^2 + 2 \\ 512x^9 - 1152x^7 + 864x^5 - 240x^3 + 18x \\ 1024x^{10} - 2560x^8 + 2240x^6 - 800x^4 + 100x^2 - 2 \end{bmatrix}$$

In [20]:

```
function charmat_halfC(n,x)
    @assert n ≥ 1
    if n ≥ 2
        Diagonal([x;fill(2x, n-1)]) + adjacent_matrix_of_type_A(n)
    else
        hcat([x])
    end
end

@vars x
charmat_halfC(5,x) ▷ display

N = 10
[sympy.det(charmat_halfC(n,x)).expand() for n in 1:N] ▷ display
[ChebyshevT(n,x) for n in 1:N] ▷ display
```

$$\begin{bmatrix} x & 1 & 0 & 0 & 0 \\ 1 & 2x & 1 & 0 & 0 \\ 0 & 1 & 2x & 1 & 0 \\ 0 & 0 & 1 & 2x & 1 \\ 0 & 0 & 0 & 1 & 2x \end{bmatrix}$$

$$\begin{bmatrix} x \\ 2x^2 - 1 \\ 4x^3 - 3x \\ 8x^4 - 8x^2 + 1 \\ 16x^5 - 20x^3 + 5x \\ 32x^6 - 48x^4 + 18x^2 - 1 \\ 64x^7 - 112x^5 + 56x^3 - 7x \\ 128x^8 - 256x^6 + 160x^4 - 32x^2 + 1 \\ 256x^9 - 576x^7 + 432x^5 - 120x^3 + 9x \\ 512x^{10} - 1280x^8 + 1120x^6 - 400x^4 + 50x^2 - 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ 2x^2 - 1 \\ 4x^3 - 3x \\ 8x^4 - 8x^2 + 1 \\ 16x^5 - 20x^3 + 5x \\ 32x^6 - 48x^4 + 18x^2 - 1 \\ 64x^7 - 112x^5 + 56x^3 - 7x \\ 128x^8 - 256x^6 + 160x^4 - 32x^2 + 1 \\ 256x^9 - 576x^7 + 432x^5 - 120x^3 + 9x \\ 512x^{10} - 1280x^8 + 1120x^6 - 400x^4 + 50x^2 - 1 \end{bmatrix}$$

In [21]:
```
function charmat_D(n,x)
    @assert n ≥ 2
    if n ≥ 3
        Diagonal(fill(2x, n)) + adjacent_matrix_of_type_D(n)
    else
        Diagonal(fill(2x, 2))
    end
end

@vars x
charmat_D(5,x) ▷ display

N = 10
[sympy.det(charmat_D(n+1,x)).simplify().expand() for n in 1:N] ▷ display
[(4x*ChebyshevT(n,x)).expand() for n in 1:N] ▷ display
```

$$\begin{bmatrix} 2x & 0 & 1 & 0 & 0 \\ 0 & 2x & 1 & 0 & 0 \\ 1 & 1 & 2x & 1 & 0 \\ 0 & 0 & 1 & 2x & 1 \\ 0 & 0 & 0 & 1 & 2x \end{bmatrix}$$

$$\begin{bmatrix} 4x^2 \\ 8x^3 - 4x \\ 16x^4 - 12x^2 \\ 32x^5 - 32x^3 + 4x \\ 64x^6 - 80x^4 + 20x^2 \\ 128x^7 - 192x^5 + 72x^3 - 4x \\ 256x^8 - 448x^6 + 224x^4 - 28x^2 \\ 512x^9 - 1024x^7 + 640x^5 - 128x^3 + 4x \\ 1024x^{10} - 2304x^8 + 1728x^6 - 480x^4 + 36x^2 \\ 2048x^{11} - 5120x^9 + 4480x^7 - 1600x^5 + 200x^3 - 4x \end{bmatrix}$$

$$\begin{bmatrix} 4x^2 \\ 8x^3 - 4x \\ 16x^4 - 12x^2 \\ 32x^5 - 32x^3 + 4x \\ 64x^6 - 80x^4 + 20x^2 \\ 128x^7 - 192x^5 + 72x^3 - 4x \\ 256x^8 - 448x^6 + 224x^4 - 28x^2 \\ 512x^9 - 1024x^7 + 640x^5 - 128x^3 + 4x \\ 1024x^{10} - 2304x^8 + 1728x^6 - 480x^4 + 36x^2 \\ 2048x^{11} - 5120x^9 + 4480x^7 - 1600x^5 + 200x^3 - 4x \end{bmatrix}$$

3.3.2 第2種Chebyshev多項式の場合

θ_j 達を

$$\theta_j = \frac{j\pi}{n+1} \quad (j = 1, 2, \dots, n)$$

と定めると, $\sin((n+1)\theta_j) = 0$ となり, $\cos\theta_j$ は互いに異なる. ゆえに $\sin((n+1)\theta)/\sin\theta = U_n(\cos\theta)$ かつ $U_n(x)$ が最高次の係数が 2^n の n 次多項式であることより,

$$U_n(x) = 2^n \prod_{j=1}^{n} \left(x - \cos\theta_j\right) = 2^n \prod_{j=1}^{n} \left(x + \cos\theta_j\right)$$

を得る. この公式の後者の等号は $\cos\theta_{n+1-j} = \cos(\pi - \theta_j) = -\cos\theta_j$ から得られる.

一方, A_n 型隣接行列

$$\begin{bmatrix} 0 & 1 & & & \\ 1 & 0 & 1 & & \\ & 1 & 0 & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & 0 \end{bmatrix}$$

の固有値の全体は $2\cos\theta_j$ $(j = 1, 2, \dots, n)$ だったので, 最高次の係数と零点の一致によって

$$\begin{vmatrix} 2x & 1 & & & \\ 1 & 2x & 1 & & \\ & 1 & 2x & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & 2x \end{vmatrix} = \begin{vmatrix} 2x & -1 & & & \\ -1 & 2x & -1 & & \\ & -1 & 2x & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2x \end{vmatrix} = U_n(x)$$

が得られる.

In [22]:

```julia
function charmat_A(n,x)
    if n ≥ 2
        Diagonal(fill(2x, n)) + adjacent_matrix_of_type_A(n)
    else
        hcat([2x])
    end
end

@vars x
charmat_A(5,x) ▷ display

N = 10
[sympy.det(charmat_A(n,x)).simplify().expand() for n in 2:N] ▷ display
[ChebyshevU(n,x) for n in 2:N] ▷ display
```

$$\begin{bmatrix} 2x & 1 & 0 & 0 & 0 \\ 1 & 2x & 1 & 0 & 0 \\ 0 & 1 & 2x & 1 & 0 \\ 0 & 0 & 1 & 2x & 1 \\ 0 & 0 & 0 & 1 & 2x \end{bmatrix}$$

$$\begin{bmatrix} 4x^2 - 1 \\ 8x^3 - 4x \\ 16x^4 - 12x^2 + 1 \\ 32x^5 - 32x^3 + 6x \\ 64x^6 - 80x^4 + 24x^2 - 1 \\ 128x^7 - 192x^5 + 80x^3 - 8x \\ 256x^8 - 448x^6 + 240x^4 - 40x^2 + 1 \\ 512x^9 - 1024x^7 + 672x^5 - 160x^3 + 10x \\ 1024x^{10} - 2304x^8 + 1792x^6 - 560x^4 + 60x^2 - 1 \end{bmatrix}$$

$$\begin{bmatrix} 4x^2 - 1 \\ 8x^3 - 4x \\ 16x^4 - 12x^2 + 1 \\ 32x^5 - 32x^3 + 6x \\ 64x^6 - 80x^4 + 24x^2 - 1 \\ 128x^7 - 192x^5 + 80x^3 - 8x \\ 256x^8 - 448x^6 + 240x^4 - 40x^2 + 1 \\ 512x^9 - 1024x^7 + 672x^5 - 160x^3 + 10x \\ 1024x^{10} - 2304x^8 + 1792x^6 - 560x^4 + 60x^2 - 1 \end{bmatrix}$$

3.4 三角函数の無限積表示

3.4.1 cos の無限積表示

n は偶数であるとし, $n = 2m$ とおくと, 上の方で示した結果より,

$$\begin{aligned} \cos(2mx) &= 2^{2m-1} \prod_{j=0}^{m-1} \left(\cos^2 x - \cos^2 \frac{(2j+1)\pi}{4m} \right) \\ &= (-1)^m 2^{2m-1} \prod_{j=0}^{m-1} \left(\sin^2 x - \sin^2 \frac{(2j+1)\pi}{4m} \right). \end{aligned}$$

この等式の両辺を $x = 0$ とおいた場合の等式の両辺で割ると次が得られる:

$$\cos(2mx) = \prod_{j=0}^{m-1} \left(1 - \frac{\sin^2 x}{\sin^2 \frac{(2j+1)\pi}{4m}} \right).$$

これに $x = \dfrac{\pi s}{4m}$ を代入すると,

$$\cos \frac{\pi s}{2} = \prod_{j=0}^{m-1} \left(1 - \frac{\sin^2 \frac{\pi s}{4m}}{\sin^2 \frac{(2j+1)\pi}{4m}} \right).$$

この等式の $m \to \infty$ の極限で次が得られる:

$$\cos \frac{\pi s}{2} = \prod_{j=0}^{\infty} \left(1 - \frac{s^2}{(2j+1)^2} \right) = \left(1 - \frac{s^2}{1^2} \right) \left(1 - \frac{s^2}{3^2} \right) \left(1 - \frac{s^2}{5^2} \right) \cdots$$

3.4.2 sin の無限積表示

n は偶数であるとし, $n = 2m$ とおくと, 上の方で示した結果より,

$$
\begin{aligned}
\frac{\sin((2m+1)x)}{\sin x} &= 2^{2m} \prod_{j=1}^{m} \left(\cos^2 x - \cos^2 \frac{j\pi}{2m+1} \right) \\
&= (-1)^m 2^{2m} \prod_{j=1}^{m} \left(\sin^2 x - \sin^2 \frac{j\pi}{2m+1} \right).
\end{aligned}
$$

この等式の $x \to 0$ での極限の両辺でこの等式の両辺をそれぞれ割ると次が得られる:

$$
\frac{\sin((2m+1)x)}{(2m+1)\sin x} = \prod_{j=1}^{m} \left(1 - \frac{\sin^2 x}{\sin^2 \frac{j\pi}{2m+1}} \right).
$$

これに $x = \frac{\pi s}{2m+1}$ を代入すると,

$$
\frac{\sin(\pi s)}{(2m+1)\sin \frac{\pi s}{2m+1}} = \prod_{j=1}^{m} \left(1 - \frac{\sin^2 \frac{\pi s}{2m+1}}{\sin^2 \frac{j\pi}{2m+1}} \right).
$$

この等式の $m \to \infty$ での極限で次が得られる:

$$
\frac{\sin(\pi s)}{\pi s} = \prod_{j=1}^{\infty} \left(1 - \frac{s^2}{j^2} \right) = \left(1 - \frac{s^2}{1^2} \right) \left(1 - \frac{s^2}{2^2} \right) \left(1 - \frac{s^2}{3^2} \right) \cdots
$$