付表　主要な MATLAB 関数

一般的な MATLAB 関数

■ コマンドウィンドウの表示形式

関数名	使用例	説明
clc	clc	コマンドウィンドウの表示をクリア
format	format loose	空行を追加して表示 (標準の設定)
	format compact	余分な空行を抑制して表示
	format short	小数点以下 4 桁の short 型固定小数点で表示 (標準の設定)
	format short e	short 型の指数形式で表示
	format long	double 値の場合は小数点以下 15 桁，single 値の場合は小数点以下 7 桁の long 型固定小数点で表示
	format long e	long 型の指数形式で表示

■ コマンドウィンドウでの入出力

関数名	使用例	説明
disp	disp('text')	コマンドウィンドウに text を表示
pause	pause	キーボードから何か入力されるまで停止
input	y = input('text')	コマンドウィンドウに text を表示し，キーボードから入力された値を y に代入
fprintf	fprintf('text: %5.3f¥n',x)	コマンドウィンドウに表示 (C 言語の関数 printf と同様の使い方) • %d, %5d：整数　• %f, %5.3f：実数 • %e, %5.3e：指数形式の実数 • ¥n：改行　• ¥t：タブ

■ ワークスペース変数の保存と読み込み

関数名	使用例	説明
who	who	ワークスペース変数を表示
whos	whos	ワークスペース変数の名前，サイズ，型を表示
clear	clear	ワークスペース変数すべてを消去
	clear x y	ワークスペース変数のうち x, y のみを消去
save	save('filename') save filename	ワークスペース変数すべてを mat ファイル filename.mat に保存
	save('filename','x','y') save filename x y	ワークスペース変数のうち x, y のみを mat ファイル filename.mat に保存

関数名	使用例	説明
load	load('filename') load filename	mat ファイル filename.mat に保存されている変数すべてをワークスペースに読み込み
	load('filename','x','y') load filename x y	mat ファイル filename.mat に保存されている変数のうち x, y のみをワークスペースに読み込み
writematrix	writematrix(data,'filename.xlsx')	Excel ファイル "filename.xlsx" として data を保存
	writematrix(data,'filename.txt') writematrix(data,'filename')	テキストファイル "filename.txt" として data を保存
readmatrix	data = readmatrix('filename.xlsx')	Excel ファイル "filename.xlsx" からデータを読み込み，data として定義
	data = readmatrix('filename.txt') data = readmatrix('filename')	テキストファイル "filename.txt" からのデータの読み込み，data として定義

■ 基本的な数学関数

関数名	使用例	説明
sin	sin(x)	正弦関数 $\sin x$ (x [rad])
cos	cos(x)	余弦関数 $\cos x$ (x [rad])
tan	tan(x)	正接関数 $\tan x$ (x [rad])
asin	asin(x)	逆正弦関数 $\sin^{-1} x$ [rad]
acos	acos(x)	逆余弦関数 $\cos^{-1} x$ [rad]
atan	atan(x)	逆正接関数 $\theta = \tan^{-1} x$ [rad] $(-\pi/2 \leq \theta \leq \pi/2)$
atan2	atan2(y,x)	$\tan\theta = y/x$ となる θ [rad] $(-\pi \leq \theta \leq \pi)$
exp	exp(x)	指数関数 e^x
log	log(x)	自然対数関数 $\log_e x$, $\ln x$
log10	log10(x)	常用対数関数 $\log_{10} x$
sqrt	sqrt(x)	平方根 $\sqrt{x}$
real	real(x)	複素数 $x = a + jb$ の実部 $a = \mathrm{Re}[x]$
imag	imag(x)	複素数 $x = a + jb$ の虚部 $b = \mathrm{Im}[x]$
abs	abs(x)	実数 x の絶対値 $\lvert x\rvert$，複素数 $x = a + jb$ の大きさ $\lvert x\rvert = \sqrt{a^2 + b^2}$
angle	angle(x)	複素数 $x = a + jb$ の偏角 $\theta = \tan^{-1}(b/a)$ [rad] $(-\pi \leq \theta \leq \pi)$
mod	mod(m,n)	整数 m を整数 n で割ったときの余り

■ 等間隔のデータ列の生成

関数名	使用例	説明
linspace	`x = linspace(xmin,xmax,n)`	$x_{\min} \le x \le x_{\max}$ の範囲で等間隔に n 個のデータ x を生成
logspace	`w = logspace(dmin,dmax,n)`	$10^{d_{\min}} \le \omega = 10^{d} \le 10^{d_{\max}}$ の範囲で対数スケールで等間隔に n 個のデータ ω を生成 ($d_{\min} \le d \le d_{\max}$ の範囲で等間隔に n 個のデータ d を生成)

■ 基本的な行列の生成

関数名	使用例	説明
eye	`eye(n)`	$n \times n$ の単位行列 $\boldsymbol{I}$
zeros	`zeros(m,n)`	$m \times n$ の零行列 $\boldsymbol{O}$
diag	`D = diag([d1 d2 d3])`	対角行列 $\boldsymbol{D} = \mathrm{diag}\{\, d_1,\, d_2,\, \cdots,\, d_n \,\}$
blkdiag	`D = blkdiag(D1,D2,D3)`	ブロック対角行列 $\boldsymbol{D} = \text{block-diag}\{\, \boldsymbol{D}_1,\, \boldsymbol{D}_2,\, \cdots,\, \boldsymbol{D}_n \,\} := \begin{bmatrix} \boldsymbol{D}_1 & & \boldsymbol{0} \\ & \boldsymbol{D}_2 & \\ & & \ddots \\ \boldsymbol{0} & & \boldsymbol{D}_n \end{bmatrix}$

■ 基本的な行列の解析・操作

関数名	使用例	説明
size	`[m n] = size(A)`	$m \times n$ 行列 $\boldsymbol{A}$ のサイズ m, n
length	`n = length(x)`	ベクトル $\boldsymbol{x}$ の次元 n
	`N = length(A)`	$m \times n$ 行列 $\boldsymbol{A}$ の最大次元 $N = \begin{cases} n & (n \ge m) \\ m & (n \le m) \end{cases}$
inv	`inv(A)`	正方行列 $\boldsymbol{A}$ の逆行列 $\boldsymbol{A}^{-1}$
pinv	`pinv(A)`	$m \times n$ 行列 $\boldsymbol{A}$ の疑似逆行列 $\boldsymbol{A}^{+} := (\boldsymbol{A}^{\top}\boldsymbol{A})^{-1}\boldsymbol{A}^{\top}$ $(m \ge n)$ もしくは $\boldsymbol{A}^{+} := \boldsymbol{A}^{\top}(\boldsymbol{A}\boldsymbol{A}^{\top})^{-1}$ $(m \le n)$
eig	`p = eig(A)`	$n \times n$ 行列 $\boldsymbol{A}$ の固有値 p_i $(i = 1, 2, \ldots, n)$ を集約した縦ベクトル $\boldsymbol{p} = \begin{bmatrix} p_1 & p_2 & \cdots & p_n \end{bmatrix}^{\top}$
	`[V P] = eig(A)`	$n \times n$ 行列 $\boldsymbol{A}$ の固有値 p_i, 固有ベクトル $\boldsymbol{v}_i$ からなる $n \times n$ 行列 $\boldsymbol{P} = \mathrm{diag}\{\, p_1, p_2, \cdots, p_n \,\}$, $\boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n \end{bmatrix}$
rank	`rank(A)`	行列 $\boldsymbol{A}$ のランク (階数) $\mathrm{rank}\,\boldsymbol{A}$
det	`det(A)`	$n \times n$ 行列 $\boldsymbol{A}$ の行列式 $\det \boldsymbol{A}$ (もしくは $\lvert\boldsymbol{A}\rvert$)
poly	`c = poly(A)`	$n \times n$ 行列 $\boldsymbol{A}$ の特性多項式 $\lvert p\boldsymbol{I} - \boldsymbol{A}\rvert = a_n p^n + \cdots + a_1 p + a_0$ の係数を集約した横ベクトル $\boldsymbol{c} = \begin{bmatrix} a_n & \cdots & a_1 & a_0 \end{bmatrix}$

関数名	使用例	説明
roots	p = roots(c)	横ベクトル $\boldsymbol{c} = \begin{bmatrix} a_n & \cdots & a_1 & a_0 \end{bmatrix}$ の要素を係数とした n 次方程式 $$a_n p^n + \cdots + a_1 p + a_0 = 0$$ の解 $p = p_i$ を集約した縦ベクトル $\boldsymbol{p} = \begin{bmatrix} p_1 & p_2 & \cdots & p_n \end{bmatrix}^\top$
max	xmax = max(x)	n 次元ベクトル $\boldsymbol{x}$ の最大要素 $x_{\max}$
	[xmax imax] = max(x)	n 次元ベクトル $\boldsymbol{x}$ の最大要素 $x_{\max}$ ($\boldsymbol{x}$ の $i_{\max}$ 番目の要素)
min	xmin = min(x)	n 次元ベクトル $\boldsymbol{x}$ の最小要素 $x_{\min}$
	[xmin imin] = min(x)	n 次元ベクトル $\boldsymbol{x}$ の最小要素 $x_{\min}$ ($\boldsymbol{x}$ の $i_{\min}$ 番目の要素)

■ グラフの描画・操作

関数名	使用例	説明
figure	figure(i)	i 番目のフィギュアウィンドウを作成または指定
subplot	subplot(m,n,i)	フィギュアウィンドウを $m \times n$ に分割し，i 番目の場所を指定
close	close(i)	i 番目のフィギュアウィンドウを閉じる
	close all	すべてのフィギュアウィンドウを閉じる
plot	plot(x,y)	横軸を x，縦軸を y としたグラフの描画
	plot(x1,y1,x2,y2)	複数のグラフを描画
plot3	plot3(x,y,z)	3 次元グラフの描画
	plot3(x1,y1,z1,x2,y2,z2)	複数の 3 次元グラフを描画
semilogx	semilogx(x,y)	横軸を $\log_{10} x$，縦軸を y としたグラフの描画
	semilogx(x1,y1,x2,y2)	複数のグラフを描画
semilogy	semilogy(x,y)	横軸を x，縦軸を $\log_{10} y$ としたグラフの描画
	semilogy(x1,y1,x2,y2)	複数のグラフを描画
loglog	loglog(x,y)	横軸を $\log_{10} x$，縦軸を $\log_{10} y$ としたグラフの描画
	loglog(x1,y1,x2,y2)	複数のグラフを描画
title	title('text')	グラフの描画枠の上方にタイトル text を表示
xlabel	xlabel('text')	横軸にラベル text を表示
ylabel	ylabel('text')	縦軸にラベル text を表示
legend	legend('text')	凡例の表示
	legend('text1','text1')	複数のグラフの凡例を表示
xlim	xlim([xmin xmax])	横軸の範囲を $x_{\min} \leq x \leq x_{\max}$ に設定
ylim	ylim([ymin ymax])	縦軸の範囲を $y_{\min} \leq y \leq y_{\max}$ に設定
axis	axis([xmin xmax ymin ymax])	横軸の範囲を $x_{\min} \leq x \leq x_{\max}$，縦軸の範囲を $y_{\min} \leq y \leq y_{\max}$ に設定
	axis normal	枠を長方形にする (標準の設定)
	axis square	枠を正方形にする
	axis on	枠を表示 (標準の設定)
	axis off	枠を非表示 (グラフのみを表示)

関数名	使用例	説明
xticks	xticks(xmin:h:xmax)	横軸の目盛りを $x_{\min}$ から $x_{\max}$ まで h 刻みに設定
yticks	yticks(ymin:h:ymax)	縦軸の目盛りを $y_{\min}$ から $y_{\max}$ まで h 刻みに設定
grid	grid on	補助線の表示
	grid off	補助線の非表示
hold	hold on	グラフの保持 (グラフの重ね描きを許可)
	hold off	グラフの解放 (グラフの重ね描きを不許可)
clf	clf(i)	i 番目のフィギュアウィンドウのグラフを消去

■ 制御文

関数名	使用例	説明
if	if i > 0 　　fprintf('positive number¥n') elseif i < 0 　　fprintf('negative number¥n') else 　　fprintf('zero¥n') end	分岐処理
for	for i = 1:10 　　fprintf('i = %d¥n') end	指定した回数の反復処理

while	i = 1;
	while i <= 10
	　　fprintf('i = %d¥n')
	　　i = i + 1;
	end
		条件が true (真) の場合に反復

break	i = 1;
	while true
	　　if i > 5
	　　　　fprintf('i = %d¥n')
	　　　　break
	　　end
	　　i = i + 1;
	end
		反復処理を強制終了

数式処理における MATLAB 関数

関数名	使用例	説明
syms	`syms x y`	x, y を複素数のシンボリック変数として定義
	`syms x y real`	x, y を実数のシンボリック変数として定義
	`syms x y positive`	x, y を正数のシンボリック変数として定義
	`syms x y integer`	x, y を整数のシンボリック変数として定義
simplify	`simplify(fx)`	$f(x)$ を単純化
collect	`collect(fx)`	$f(x)$ をべき乗でまとめる
	`collect(fx,x)`	$f(x)$ を x に関するべき乗でまとめる
factor	`factor(fx)`	$f(x)$ を因数分解したときの因数
	`prod(factor(fx))`	$f(x)$ を因数分解
expand	`expand(fx)`	$f(x)$ の展開
subs	`subs(fx,x,a)`	$f(x)$ の x に a を代入 $\left(f(x)\big\vert_{x=a}\right)$
limit	`limit(fx,x,a)`	極限 $\lim_{x\to a} f(x)$
fplot	`fplot(fx)`	グラフの描画
	`fplot(fx,[xmin xmax])`	グラフの描画 (横軸の範囲を指定)
laplace	`fs = laplace(ft)`	$f(t)$ のラプラス変換 $f(s)=\mathcal{L}\left[f(t)\right]$
ilaplace	`ft = ilaplace(fs)`	$f(s)$ の逆ラプラス変換 $f(t)=\mathcal{L}^{-1}\left[f(s)\right]$
partfrac	`partfrac(fs,s)`	$f(s)$ を部分分数分解
diff	`diff(fx,x)`	$f'(x)$：$f(x)$ を x で微分
int	`int(fx,x)`	$\int f(x)\mathrm{d}x$：$f(x)$ を x で不定積分
	`int(fx,x,a,b)`	$\int_a^b f(x)\mathrm{d}x$：$f(x)$ を x で定積分
taylor	`taylor(fx)`	$f(x)$ の 5 次までのマクローリン展開
	`taylor(fx,x,'Order',n)`	$f(x)$ の n 次までのマクローリン展開
	`taylor(fx,x,a)`	$f(x)$ の $x=a$ における 5 次までのテイラー展開
	`taylor(fx,x,a,'Order',n)`	$f(x)$ の $x=a$ における n 次までのテイラー展開

制御工学に関連した MATLAB 関数

■ モデルの定義

関数名	使用例	説明
tf	`sys = tf(num,den)`	(T.1) 式の形式の伝達関数 $P(s)$ を定義
	`sys = tf(sys)`	(T.1) 式の形式の伝達関数 $P(s)$ に変換
	`s = tf('s')`	ラプラス演算子 s の定義
zpk	`sys = zpk(z,p,K)`	(T.2) 式の形式の伝達関数 $P(s)$ の定義
	`sys = zpk(sys)`	(T.2) 式の形式の伝達関数 $P(s)$ に変換
ss	`sys = ss(A,B,C,D)`	状態空間表現 (T.3) 式の定義
	`sys = ss(sys)`	状態空間表現 (T.3) 式に変換

$$P(s)=\frac{N(s)}{D(s)},\quad \begin{cases} N(s)=b_m s^m+\cdots+b_1 s+b_0 \\ D(s)=a_n s^n+\cdots+a_1 s+a_0 \end{cases} \Longrightarrow \begin{cases} \texttt{num = [bm } \cdots \texttt{ b1 b0]} \\ \texttt{den = [an } \cdots \texttt{ a1 a0]} \end{cases} \tag{T.1}$$

$$P(s)=\frac{k(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)} \Longrightarrow \begin{cases} \texttt{z = [z1 z2 } \cdots \texttt{ zm]} \\ \texttt{p = [p1 p2 } \cdots \texttt{ pn]} \end{cases} \tag{T.2}$$

$$\begin{cases} \dot{\boldsymbol{x}}(t)=\boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{B}\boldsymbol{u}(t) \\ \boldsymbol{y}(t)=\boldsymbol{C}\boldsymbol{x}(t)+\boldsymbol{D}\boldsymbol{u}(t) \end{cases} \tag{T.3}$$

■ モデルの解析

関数名	使用例	説明
tfdata	[num den] = tfdata(sys,'v')	伝達関数 $P(s)$ の分子 $N(s)$，分母 $D(s)$ を抽出
zpkdata	[z p k] = zpkdata(sys,'v')	伝達関数 $P(s)$ の零点 z_i，極 p_i，ゲイン k を抽出
pole	pole(sys)	伝達関数 $P(s)$ の極 p_i を抽出
zero	zero(sys)	伝達関数 $P(s)$ の零点 z_i を抽出
tzero	tzero(sys)	伝達関数 $P(s)$ の (不変) 零点 z_i を抽出 (多入力多出力系にも対応)
rlocus	rlocus(sys)	$1+kP(s)$ の根軌跡
ssdata	[A B C D] = ssdata(sys)	状態空間表現 (T.3) 式の係数行列 $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$ を抽出

■ モデルの結合

関数名	使用例	説明
*	sys = sys1*sys2	直列結合 $P(s)=P_1(s)P_2(s)$
+, -	sys = sys1 + sys2 - sys3	並列結合 $P(s)=P_1(s)+P_2(s)-P_3(s)$
feedback	sys = feedback(sys1,sys2)	フィードバック結合 $P(s)=\dfrac{P_1(s)}{1+P_1(s)P_2(s)}$
minreal	sys = minreal(sys)	伝達関数 $P(s)$ の極零相殺 (分母と分子の約分)

■ 時間応答

関数名	使用例	説明
residue	[k p] = residue(num,den)	部分分数分解 (T.4) 式
impulse	impulse(sys)	インパルス応答 $y(t)$ の描画 (時間指定なし)
	impulse(sys,t)	インパルス応答 $y(t)$ の描画 (時間指定あり)
	y = impulse(sys,t);	インパルス応答 $y(t)$ の計算
step	step(sys)	単位ステップ応答 $y(t)$ の描画 (時間指定なし)
	step(sys,t)	単位ステップ応答 $y(t)$ の描画 (時間指定あり)
	y = step(sys,t);	単位ステップ応答 $y(t)$ の計算

関数名	使用例	説明
stepinfo	S = stepinfo(sys)	単位ステップ応答 $y(t)$ の特性
	y = step(sys,t); S = stepinfo(y,t,yinf)	単位ステップ応答 $y(t)$ の特性 (yinf：$y_\infty = P(0)$)
lsim	lsim(sys,u,t)	入力 $u(t)$ に対する時間応答 $y(t)$ の描画
	y = impulse(sys,u,t);	入力 $u(t)$ に対する時間応答 $y(t)$ の計算

$$f(s) = \frac{N(s)}{D(s)} = \frac{k_1}{s-p_1} + \cdots + \frac{k_n}{s-p_n}, \quad \begin{cases} N(s) = b_m s^m + \cdots + b_1 s + b_0 \\ D(s) = a_n s^n + \cdots + a_1 s + a_0 \end{cases} \quad (n > m) \tag{T.4}$$

■ 周波数特性

関数名	使用例	説明
nyquist	nyquist(sys)	ナイキスト軌跡の描画
bode	bode(sys)	ボード線図の描画 (周波数指定なし)
	bode(sys,w)	ボード線図の描画 (周波数指定あり)
	[Gg Gp] = bode(sys,w);	ゲイン，位相差の計算
bodemag	bodemag(sys)	ゲイン線図の描画 (周波数指定なし)
	bodemag(sys,w)	ゲイン線図の描画 (周波数指定あり)
getPeakGain	[Mp wp] = getPeakGain(sys)	ピーク角周波数 ω_p，共振ピーク M_p の計算
margin	margin(sys)	ボード線図の描画と安定余裕の表示
	[invL Pm wpc wgc] = margin(sys) Gm = 20*log10(invL)	ゲイン余裕 G_m，位相余裕 P_m，位相交差角周波数 ω_pc，ゲイン交差角周波数 ω_gc の計算

■ PID コントローラの設計

関数名	使用例	説明
pidtune	[sysC info] = pidtune(sysP,type)	制御対象のモデル sysP に対し，形式を type とした PID コントローラの設計
	sysC = pidtune(sysP,type,wgc)	開ループ伝達関数のゲイン交差角周波数 ω_gc を指定
	sysC = pidtune(sysP,type,opts)	pidtuneOptions により位相余裕や，目標値追従と外乱抑制のバランスを設定
pidTuner	pidTuner(sysP)	制御対象のモデル sysP に対し，PID コントローラを視覚的に設計

- type には 'P', 'PI', 'PD', 'PDF' (不完全微分とした PD コントローラ), 'PID', 'PIDF' (不完全微分とした PID コントローラ) や 'PI-D', 'PI-DF' (不完全微分とした PI–D コントローラ), 'I-PD', 'I-PDF' (不完全微分とした I–PD コントローラ) などを設定する.
- pidtuneOptions の設定方法については文献 14) を参照されたい.

■ 状態空間表現に基づく解析

関数名	使用例	説明
initial	initial(sys,x0)	$\boldsymbol{x}(0)=\boldsymbol{x}_0$ に対する零入力応答 $y(t)$ の描画 (時間指定なし)
	initial(sys,x0,t)	$\boldsymbol{x}(0)=\boldsymbol{x}_0$ に対する零入力応答 $y(t)$ の描画 (時間指定あり)
	y = initial(sys,x0,t);	$\boldsymbol{x}(0)=\boldsymbol{x}_0$ に対する零入力応答 $y(t)$ の計算
ctrb	Vc = ctrb(A,B)	可制御性行列 $\boldsymbol{V}_{\mathrm{c}}=\begin{bmatrix}\boldsymbol{B} & \boldsymbol{AB} & \cdots & \boldsymbol{A}^{n-1}\boldsymbol{B}\end{bmatrix}$ の計算
obsv	Vo = obsv(A,C)	可制御性行列 $\boldsymbol{V}_{\mathrm{o}}=\begin{bmatrix}\boldsymbol{C}\\ \boldsymbol{CA}\\ \vdots\\ \boldsymbol{CA}^{n-1}\end{bmatrix}$ の計算

$$\begin{cases}\dot{\boldsymbol{x}}(t)=\boldsymbol{A}\boldsymbol{x}(t)\\ \boldsymbol{y}(t)=\boldsymbol{C}\boldsymbol{x}(t)\end{cases}\Longrightarrow\quad \texttt{sys = ss(A,[],C,[]);}$$

■ 状態空間表現に基づくコントローラ設計

関数名	使用例	説明
acker	K = - acker(A,B,p)	極配置法：1 入力 n 次系の制御対象に対し，$\boldsymbol{A}+\boldsymbol{BK}$ の固有値を $\boldsymbol{p}=\begin{bmatrix}p_1 & p_2 & \cdots & p_n\end{bmatrix}$ とする $u(t)=\boldsymbol{K}\boldsymbol{x}(t)$ を設計
place	K = - place(A,B,p)	極配置法：m 入力 n 次系の制御対象に対し，$\boldsymbol{A}+\boldsymbol{BK}$ の固有値を $\boldsymbol{p}=\begin{bmatrix}p_1 & p_2 & \cdots & p_n\end{bmatrix}$ とする $\boldsymbol{u}(t)=\boldsymbol{K}\boldsymbol{x}(t)$ を設計 (p_i の重複は m を超えてはならない)
lqr	K = - lqr(A,B,Q,R)	最適レギュレータ：評価関数 $J=\int_0^\infty(\boldsymbol{x}(t)^\top\boldsymbol{Q}\boldsymbol{x}(t)+\boldsymbol{u}(t)^\top\boldsymbol{R}\boldsymbol{u}(t))\mathrm{d}t$ を最小化する $u(t)=\boldsymbol{K}\boldsymbol{x}(t)$ を設計
care	P = care(A,B,Q,R)	リッカチ方程式 $\boldsymbol{PA}+\boldsymbol{A}^\top\boldsymbol{P}-\boldsymbol{P}\boldsymbol{B}R^{-1}\boldsymbol{B}^\top\boldsymbol{P}+\boldsymbol{Q}=\boldsymbol{O}$ の解 $\boldsymbol{P}=\boldsymbol{P}^\top>0$ を求める