『多様体上の最適化理論』補足資料

佐藤 寛之

2024 年 1 月 27 日

I　本資料について

本資料では，書籍『多様体上の最適化理論』（以下，「書籍」と呼びます）の数値計算結果やそれを実現するためのコード等についての補足を行います．なお，書籍ならびに本資料に掲載した数値計算結果は，チップ：Apple M1 Max，メモリ：64GB，macOS：Ventura 13.4 の PC で MATLAB R2023a を用いて得たものです．なお，同じコードを実行した場合でも，計算環境によっては結果に若干の差異が生じる場合があります．

本資料の II 章では，多様体上の最適化のための MATLAB ライブラリである Manopt について，導入方法や使用方法を説明します．

また，書籍サポートページ

https://github.com/HiroyukiSatoLab/manifold-optimization-book

の直下のディレクトリ `chapter2`, `chapter9`, `chapter10` には，それぞれ書籍の 2 章，9 章，10 章の内容についてのサンプルコードを用意しています．本資料の III 章では，これらの解説を行います．

II　Manopt について

本章では，書籍の 1.5 節で紹介した多様体上の最適化のためのライブラリのうち，Nicolas Boumal 氏と Bamdev Mishra 氏により開発された Manopt を取り上げ，その詳細を解説します．

II.1　Manopt の導入方法

まず，Manopt のオフィシャルサイト

https://www.manopt.org/

から Manopt の最新版（2024 年 1 月 27 日現在のバージョンは 7.1）をダウンロードし，ZIP ファイルを解凍して `manopt` という名前のフォルダ（ディレクトリ）を得ます．続いて MATLAB を起動し，「現在のフォルダー」を先ほどのフォルダ `manopt` に変更します．そして，`importmanopt.m` を実行すると，パスの設定が自動的に行われます．これで準備は完了です．問題がなければ，`manopt` 直下のフォルダ `checkinstall` の中にある `basicexample.m` を実行することで，計算結果が表示されるとともに収束の様子を示すグラフが描画されるはずです．

II.2　Manopt のサンプルコード `basicexample.m` の説明

前節で紹介した `basicexample.m` は，Manopt のチュートリアルページ

https://www.manopt.org/tutorial.html

でも紹介されている球面上の最適化のサンプルコードです．これは，書籍の問題 (1.6) で $n = 1000$ として A を $-A$ と置き換えた S^{n-1} 上の関数 $f(\boldsymbol{x}) = -\boldsymbol{x}^\top A\boldsymbol{x}$ の最小化問題

$$\begin{aligned} &\text{目的関数：} f(\boldsymbol{x}) = -\boldsymbol{x}^\top A\boldsymbol{x} \to \text{最小} \\ &\text{制約条件：} \boldsymbol{x} \in S^{n-1} \end{aligned}$$

を，球面上の信頼領域法（書籍の 8.4.2 項）を用いて解く例です．

このサンプルコードは実行するだけで最適化を行えます．また，コード中の問題のサイズ n や行列 A を変更することも容易です．さらに他の問題も見据えて Manopt を使いこなすには，このサンプルコードの内容を具体的に見ておくと役立つので，以下でいくつかの箇所を抜粋して説明します．

Manopt での最適化は，対象とする最適化問題の情報からなる構造体 `problem`（名称は任意）を用意し，用いるアルゴリズムを指定すれば実行できます．サンプルコードの前半では構造体 `problem` のフィールド `M`，`cost`，`egrad` に必要な情報を格納しており，後半ではこの問題に対する信頼領域法を実行しています．

まず，

```
manifold = spherefactory(n);
problem.M = manifold;
```

により，problem.M に spherefactory(n) を代入しています．これで problem.M は球面 S^{n-1} 上の最適化に必要な情報をもった構造体となります．具体的には，S^{n-1} のリーマン計量（書籍の定義 6.1.1）を表す inner，接空間への直交射影（書籍の 6.3 節）を表す proj，レトラクション（書籍の定義 7.2.1）を表す retr などのフィールドをもちます．例えば，球面上の最適化計算に必要な写像であるレトラクションは一意的ではありません（書籍の例 7.2.3）が，書籍の式 (7.3) によるものがデフォルトとして設定されており，この時点でフィールド retr には対応する関数ハンドルが格納されています．なお，他の任意のレトラクションに相当する関数を用意し，その関数ハンドルを problem.M.retr に代入すれば，デフォルトのものとは異なるレトラクションを用いて最適化を行うこともできます．

次に，

```
problem.cost  = @(x) -x'*(A*x);
problem.egrad = @(x) -2*A*x;
```

により，解くべき最適化問題の目的関数とそのユークリッド勾配を指定しています．いま，目的関数は $f(\boldsymbol{x}) = -\boldsymbol{x}^\top A\boldsymbol{x}$ であり，これを $\mathbb{R}^n$ 上の関数と見た場合の通常の勾配，すなわちユークリッド勾配（書籍の付録 A.3.3 項）は $-2A\boldsymbol{x}$ なので，それらに対応する関数ハンドルを構造体 problem のそれぞれフィールド cost，egrad に代入しています．リーマン多様体（書籍の定義 6.1.1）としての球面 S^{n-1} 上の関数 f の勾配，すなわちリーマン勾配（書籍の定義 6.2.1）は，一般にはユークリッド勾配とは異なりますが，フィールド egrad でユークリッド勾配の式を指定しておけば，最適化の過程では自動的にリーマン勾配が正しく計算されるようになっています．ただし，あらかじめ理論的にリーマン勾配を導出しておくことで，計算の簡単化が行える場合もあります．このように計算効率を高めるためなどの理由により，手動でリーマン勾配を指定する場合は，対応する関数ハンドルを problem.grad に代入することで対応できます（具体例は III.5 節を参照）．勾配のみが必要な最適化手法（最急降下法や共役勾配法など）を用いる場合，これで準備は完了です．また，ヘシアンを用いる最適化手法（このサンプルコードで扱われている信頼領域法など）についても，ヘシアンを定義せずに実行することもできます．この場合は，ヘシアンは勾配の有限差分に基づき近似的に計算されるとともに，その旨の警告が表示されます．

一方，ヘシアンを具体的に定義する場合は，f を $\mathbb{R}^n$ 上の関数と見た場合のヘッセ行列が $-2A$ であり，これを $\boldsymbol{d} \in \mathbb{R}^n$ に作用させると $-2A\boldsymbol{d}$ となることに対応して，

```
problem.ehess = @(x, d) -2*A*d;
```

のように指定します．勾配の場合と同様に，リーマン多様体 S^{n-1} 上の関数としての f の

ヘシアン（書籍の定義 6.7.1）は，一般にはユークリッド空間のものとは異なりますが，ehess を指定しておけば最適化の過程では正しく計算されます．また，やはり計算の効率化の観点などから，手動でリーマン多様体上のヘシアンを problem.hess に指定することもできます．

最後に，

```
[x, xcost, info] = trustregions(problem);
```

とすることで，用いる最適化アルゴリズム（ここでは信頼領域法 trustregions）を指定し，上で用意した problem に対する最適化を実行しています．なお，trustregions のような最適化アルゴリズムを実行するための関数の入力引数としては，上記の problem のような解こうとする最適化問題の情報をもつ構造体に加えて，反復計算の初期点や，停止条件（目的関数の勾配のノルムがどれだけ小さくなれば計算を終了するかを表す値や，最大反復回数など）を与えることもできます（例えば III.1 節およびそこで説明しているコード AlgorithmComparison1.m を参照）．

このように，Manopt での多様体上の最適化は，解くべき最適化問題の情報を格納するための構造体 problem を用意し，そのフィールドに入れる値として，フィールド M に用いる多様体を，cost に目的関数を，grad にその（リーマン）勾配を（さらに必要ならば hess に（リーマン・）ヘシアンの作用を）指定し，最後に最適化アルゴリズムを指定するだけで容易に実行できます．また，上述の球面の例のように，ユークリッド空間の部分多様体上の最適化問題を解く場合には，構造体 problem のフィールド grad と hess に値（関数ハンドル）を代入する代わりに，フィールド egrad に目的関数のユークリッド勾配を，そして必要ならば ehess にヘッセ行列の作用を指定することでも，最適化計算を行えます．

しかしながら，以上の知識だけで Manopt を最大限に活用できるとは限りません．実際の問題に対する最適化でパフォーマンスの向上を図る場合や数値実験を行う場合に，Manopt の内部で設定されているパラメータをチューニングしたり，Manopt で用意されていない多様体を用いたり，用意されている多様体であっても異なるリーマン計量やレトラクションなどを指定したり，あるいは新しい最適化アルゴリズムを開発してそれを Manopt の枠組みで実装して実行したりするには，多様体上の最適化理論のより深い理解が必要不可欠です．書籍を通読することで，これらの実現のために必要な知識が得られるはずです．

II.3 Manopt で利用できる多様体と最適化アルゴリズムの例

II.1 節のようにして得られたフォルダ manopt の直下にある examples の中には多くのコード例が用意されており，これらを上手に活用するだけでも多くの問題を解くことができると期待されます．さらに，それ以外の問題にも対応できるよう，すでに多数の多様体や最適化アルゴリズムが実装されています．本節では，これらのうち書籍で扱っているものを中心に紹介します．

まず，多様体の情報をフィールドにもつ構造体を生成するための関数の一部を以下に紹介します．なお，いずれも入力としてサイズを指定するようになっているため，それもあわせて記載します．それぞれの多様体については書籍の 1.3 節（Δ_k については書籍の 1.4.3 項）を参照してください．ここで，$n, p \in \mathbb{N}$（$p \leq n$）とし，$G \in \mathrm{Sym}_{++}(n)$ とします．

- `spherefactory(n)`：球面 S^{n-1}[*1]
- `stiefelfactory(n,p)`：シュティーフェル多様体 $\mathrm{St}(p,n)$[*2]
- `stiefelgeneralizedfactory(n,p,G)`：一般化シュティーフェル多様体 $\mathrm{St}_G(p,n)$[*3]
- `grassmannfactory(n,p)`：グラスマン多様体 $\mathrm{Gr}(p,n)$[*4]
- `sympositivedefinitefactory(n)`：n 次正定値対称行列全体 $\mathrm{Sym}_{++}(n)$
- `multinomialfactory(n)`：各成分が正かつすべての成分の和が 1 であるような n 次列ベクトルの全体 $\Delta_n \coloneqq \{\boldsymbol{x} \in \mathbb{R}^n \mid \mathbf{1}_n^\top \boldsymbol{x} = 1, \boldsymbol{x} > \mathbf{0}\}$

ユークリッド空間 $\mathbb{R}^n$ や $\mathbb{R}^{m\times n}$ も多様体の一種であり，それぞれ `euclideanfactory(n)`，`euclideanfactory(m,n)` により対応する構造体を生成できます．これらを使えば，書籍の 2 章で紹介するユークリッド空間上の無制約最適化アルゴリズムも Manopt により実行できます（詳細は III.1 節を参照）．

続いて，Manopt で利用できる最適化アルゴリズムの一部を挙げます．

- `steepestdescent`：最急降下法（書籍の 8.1 節）
- `conjugategradient`：共役勾配法（書籍の 8.2 節）
- `rlbfgs`：（BFGS 公式に基づく記憶制限付き）準ニュートン法（書籍の 8.4.1 項）
- `trustregions`：信頼領域法（書籍の 8.4.2 項）
- `stochasticgradient`：確率的勾配降下法（書籍の 8.4.3 項）

なお，2024 年 1 月 27 日現在のバージョン 7.1 の Manopt ではニュートン法（書籍の 8.3 節）は実装されていません．ニュートン法の実装例は III.3 節を参照してください．

Manopt では，これらの他にも多数の多様体や最適化アルゴリズムが用意されています．詳細は Manopt のチュートリアルページ

https://www.manopt.org/tutorial.html

を参照してください．

[*1] 球面 S^{n-1} の多様体としての次元は $n-1$ ですが，S^{n-1} は $\mathbb{R}^n$ の部分多様体であるため，S^{n-1} の各点は $\mathbb{R}^n$ のベクトルとして表されます．$(n-1)$ 次元球面 S^{n-1} に対応する `spherefactory(n)` の引数 `n` が，次元とは一致しないことに注意してください．

[*2] ここで，$\mathrm{St}(p,n)$ の p，n と，`stiefelfactory(n,p)` の引数 `n`，`p` の順番が逆であることに注意してください．たとえば，$\mathbb{R}^{10\times 3}$ の部分多様体である $\mathrm{St}(3,10)$ に対応する構造体は，`stiefelfactory(10,3)` により定義できます．

[*3] シュティーフェル多様体の場合と同様に，引数 `n`，`p` の順番に注意してください．

[*4] やはり，引数 `n`，`p` の順番に注意してください．

III　サポートページのサンプルコードの説明

書籍の 2.5 節，9.2.1 項，9.2.2 項，9.3 節，9.4 節に掲載した数値計算結果を出力するためのサンプルコードを，書籍サポートページ

https://github.com/HiroyukiSatoLab/manifold-optimization-book

からダウンロードできます．また，サポートページには，書籍の 10.5 節で紹介する拡張ラグランジュ法を 10.6 節の非負主成分分析に対応する最適化問題に適用するためのサンプルコードも用意しています．本章では，これらのサンプルコードおよび関連事項について説明します．

III.1　ユークリッド空間上の種々の無制約最適化手法の比較（書籍 2.5 節）

サポートページのディレクトリ `chapter2` の `AlgorithmComparison1.m` は，書籍の (2.48) で定義される $\mathbb{R}^n$ 上の最適化問題に対し，ランダムに生成した初期点を用いて最急降下法，共役勾配法，準ニュートン法，信頼領域法を適用し結果を比較するコードであり，書籍の 2.5 節の表 2.1 と図 2.7 を出力します．また，このコードで初期点を $\boldsymbol{x}_0 = 300 \cdot \mathbf{1}_n$ に変更したものが `AlgorithmComparison2.m` であり，これを実行すると書籍の表 2.2 が出力されます．以下では `AlgorithmComparison1.m` について説明します．

`AlgorithmComparison1.m` の構造は II.2 節で説明した `basicexample.m` とほぼ同様です．

```
manifold = euclideanfactory(n);
problem.M = manifold;
```

により用いる多様体をユークリッド空間 $\mathbb{R}^n$ と設定し，

```
problem.cost  = @(x) 0.5 * x'*(A*x) - b'*x;
problem.egrad = @(x) A*x - b;
problem.ehess = @(x, d) A*d;
```

により，目的関数 $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top A\boldsymbol{x} - \boldsymbol{b}^\top\boldsymbol{x}$ とそのユークリッド勾配 $\operatorname{grad} f(\boldsymbol{x}) = A\boldsymbol{x} - \boldsymbol{b}$，またヘッセ行列による作用 $\operatorname{Hess} f(\boldsymbol{x})[\boldsymbol{d}] = A\boldsymbol{d}$ を定義しています．次に，

```
x0 = randn(n,1);
options.tolgradnorm = 1e-6;
```

では初期点 $\boldsymbol{x}_0$ をランダムに生成し，また，構造体 `options` のフィールド `tolgradnorm` を 10^{-6} と設定しています．そして，

```
[xSD, costSD, infoSD] = steepestdescent(problem, x0, options);
```

により最急降下法を実行しています．フィールド tolgradnorm をもつ構造体 options を入力とすることで，目的関数の勾配のノルムが options.tolgradnorm に設定された値 10^{-6} 未満になれば最急降下法の計算を終了します．出力 xSD は計算終了時点で得られた解であり，costSD は対応する目的関数値です．infoSD は計算に要した反復回数や計算時間，また各反復における目的関数値などの情報をもつ構造体です．最急降下法以外の最適化手法についても同様です．

III.2　グラスマン多様体上の種々の無制約最適化手法の比較（書籍 9.2.1 項）

書籍の (1.9) で定義されるグラスマン多様体上の最適化問題に対して，種々の最適化アルゴリズムを実行し，書籍の図 9.3 と表 9.1 を出力するためのコードが，サポートページのディレクトリ chapter9/section2 の GrassmannManifold.m です．このコードも前節までのコードと同様に記述されているため，新たに説明すべきことはほとんどありません．前節までで登場していなかった記述としては，構造体 options のフィールドとして

```
options.maxiter = 2000;
```

と設定していることが挙げられます．この構造体 options を用いて

```
[xSD, costSD, infoSD] = steepestdescent(problem, X0, options);
```

とすることで，アルゴリズムの停止条件として，最大反復回数を options.maxiter の値 2000 と設定して最急降下法を実行できます．すなわち，反復回数が 2000 回に達した場合には，目的関数の勾配のノルムが十分に小さくなっていなくとも計算を終了します．

III.3　シュティーフェル多様体の積多様体上のニュートン法（書籍 9.2.2 項）

書籍の (1.10) で定義される，2 つのシュティーフェル多様体の積多様体上の最適化問題に対してニュートン法と共役勾配法を実行し，書籍の表 9.2 を出力するためのコードが，サポートページのディレクトリ chapter9/section2 の StiefelManifold.m です．このコードのうち共役勾配法の実行については前節までの説明と同様であるため，ここではニュートン法について説明します．

2024 年 1 月 27 日現在のバージョン 7.1 の Manopt では，多様体上のニュートン法は実装されていません．そのため，多様体 $\mathcal{M}$ 上の点 x_k におけるニュートン方程式 $\operatorname{Hess} f(x_k)[\eta] = -\operatorname{grad} f(x_k)$ を $\eta \in T_{x_k}\mathcal{M}$ について解くためのコードが必要となります．そこで，ニュートン方程式の解法の一つとして書籍の 8.3.3 項で紹介しているアルゴリズム 8.4（一般の内積空間における線形方程式を解くための線形共役勾配法）を，リーマン多様体上の接空間における線形方程式の解法として実装したものが，サポートページのディレクトリ chapter9/section2 の GeneralLinearCG.m です．

```
[out, iter, time] = GeneralLinearCG(M, p, A, b, x0, tol);
```

により，リーマン多様体 $\mathcal{M}$ 上の点 p における接空間で定義された $x \in T_p\mathcal{M}$ についての線形方程式 $\mathcal{A}(x) = b$（ただし $\mathcal{A}$ は $T_p\mathcal{M}$ における対称変換）を，初期点を $x_0 \in T_p\mathcal{M}$ として解きます．この際，線形共役勾配法の反復により $x_1, x_2, \dots$ を計算し，ある $k \in \mathbb{N}_0$ について $\|\mathcal{A}(x_k) - b\|$ が `tol` 未満になったら，その時点での x_k を解 `out` として出力します．また，そのときの反復回数 k を `iter`，要した計算時間を `time` として，あわせて出力します．

さて，`StiefelManifold.m` では，用いる多様体を次のように定義しています．

```
M1 = stiefelfactory(m,p);
M2 = stiefelfactory(n,p);
problem.M = productmanifold(struct('U', M1, 'V', M2));
```

ここで，`productmanifold` は積多様体を構成するための関数であり，この箇所ではまず `M1`，`M2` をそれぞれ $\mathrm{St}(p,m)$，$\mathrm{St}(p,n)$ に対応する構造体とし，これらをフィールドの値にもつ構造体を `productmanifold` の入力とすることで，積多様体に対応する構造体を新たに作り出しています．そして，ニュートン法の反復を以下のように実装しています．

```
while normGrad >= 1e-6
  k = k + 1;
  Hess = @(D)problem.M.ehess2rhess(X,EGrad,problem.ehess(X,D),D);
  NegativeGrad = problem.M.lincomb(X, -1, Grad);
  [eta, iterCG(k), timeCG(k)] = ...
        GeneralLinearCG(problem.M, X, Hess, NegativeGrad, ...
                        problem.M.zerovec(X), 1e-6);
  X = problem.M.retr(X, eta);
  EGrad = problem.egrad(X);
  Grad = problem.M.egrad2rgrad(X, EGrad);
  normGrad = problem.M.norm(X, Grad);
end
```

ここではヘシアンの作用を表す関数ハンドルとして `Hess` を定義し，上述の関数 M ファイル `GeneralLinearCG.m` で定義した関数 `GeneralLinearCG` によりニュートン方程式を解いています．この際，線形共役勾配法の初期点は零ベクトルとし，ニュートン方程式の残差ノルムが 10^{-6} 未満となるような解を `eta` として出力しています．そして，得られた `eta` とレトラクション `problem.M.retr` により，次の点 `X` を計算しています．最後に `normGrad` として目的関数の勾配のノルムを計算しているのは，ニュートン法自体の反復の停止条件に用いるためです．`while` 文の条件からわかるように，目的関数の勾配のノルム `normGrad` が 10^{-6} 未満になったらニュートン法の計算を終了するようにしています．

III.4　一般化シュティーフェル多様体上のレトラクションの比較（書籍 9.3 節）

書籍の (9.17) で定義される，2 つの一般化シュティーフェル多様体の積多様体上の最適化問題に関連して，一般化シュティーフェル多様体上のレトラクションの計算時間についての比較を行い書籍の表 9.3 を出力するためのコードが，サポートページのディレクトリ `chapter9/section3` の `GeneralizedStiefelRetraction.m` です．このコードでは，

```
M = stiefelgeneralizedfactory(n,p,G);
X = M.rand();
eta = M.randvec(X);
```

として，$\mathrm{St}_G(p,n)$ 上の点 X と X における接ベクトル η をランダムに生成しています．コード中で Manopt を用いているのはこの部分だけで，他の箇所は書籍の式 (9.24), (9.25) をその通りに実装しています．詳細は書籍のノート 9.3.1 を参照してください．

III.5　正定値対称行列全体からなる多様体上の共役勾配法（書籍 9.4 節）

書籍の 9.4 節の混合正規分布モデルに関する最適化を行うためのコードが, サポートページのディレクトリ `chapter9/section4` の `SymmetricPositiveDefiniteManifold.m` です．このコードでは，積多様体を扱うために次の記述をしています．

```
Simplex = multinomialfactory(3);
Euclidean = euclideanfactory(2,1);
SymPositive = sympositivedefinitefactory(2);
M1 = Simplex;
M2 = powermanifold(Euclidean,3);
M3 = powermanifold(SymPositive,3);
problem.M = productmanifold(struct('w',M1,'mu',M2,'Sigma',M3));
```

`Simplex` および `M1` は重みベクトルが属する多様体 $\Delta_3 \coloneqq \{\boldsymbol{w} \in \mathbb{R}^3 \mid \mathbf{1}_3^\top \boldsymbol{w} = 1, \boldsymbol{w} > \mathbf{0}\}$, `Euclidean` は各平均ベクトルが属するユークリッド空間（平面）$\mathbb{R}^2$, `SymPositive` は各分散共分散行列が属する多様体 $\mathrm{Sym}_{++}(2)$ に対応する構造体です．また，`powermanifold` は入力された多様体の指定の個数分の積多様体に対応する構造体を生成します．したがって，`M2` は 3 つの `Euclidean` の積多様体 $(\mathbb{R}^2)^3$, `M3` は 3 つの `SymPositive` の積多様体 $\left(\mathrm{Sym}_{++}(2)\right)^3$ に対応する構造体です．そして，`problem.M` は，`M1`, `M2`, `M3` の積多様体 $\Delta_3 \times (\mathbb{R}^2)^3 \times \left(\mathrm{Sym}_{++}(2)\right)^3$ に対応する構造体です．コード `SymmetricPositiveDefiniteManifold.m` の他の箇所で Manopt を用いているのは，目的関数とその勾配を定義し共役勾配法を実行する箇所のみで，これらは前節までと同様です．なお，勾配の理論的な導出はやや複雑であり書籍では割愛しています．

以下では，書籍の 9.4 節の議論の補足として，目的関数

$$\begin{aligned} f(\Theta) &\coloneqq -\sum_{t=1}^{T} \log \left(\sum_{k=1}^{K} w_k p_n(\boldsymbol{x}_t; \boldsymbol{\mu}_k, \Sigma_k) \right) \\ &= -\sum_{t=1}^{T} \log \left(\sum_{k=1}^{K} \frac{w_k}{\sqrt{(2\pi)^n \det \Sigma_k}} \exp \left(-\frac{1}{2} (\boldsymbol{x}_t - \boldsymbol{\mu}_k)^\top \Sigma_k^{-1} (\boldsymbol{x}_t - \boldsymbol{\mu}_k) \right) \right) \end{aligned}$$

の勾配を計算します．ここで，$\Theta = (\boldsymbol{w}, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_K, \Sigma_1, \Sigma_2, \dots, \Sigma_K)$ です．$\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_K, \Sigma_1, \Sigma_2, \dots, \Sigma_K$ を固定したときの関数 $\boldsymbol{w} \mapsto f(\Theta)$ の勾配を $\operatorname{grad}_{\boldsymbol{w}} f(\Theta)$ とし，同様に $\operatorname{grad}_{\boldsymbol{\mu}_k} f(\Theta)$ や $\operatorname{grad}_{\Sigma_k}(\Theta)$ $(k = 1, 2, \dots, K)$ を定義すると，書籍の式 (9.10) より，積多様体上の関数 f の勾配について，

$$\begin{aligned} \operatorname{grad} f(\Theta) = \big(&\operatorname{grad}_{\boldsymbol{w}} f(\Theta), \operatorname{grad}_{\boldsymbol{\mu}_1} f(\Theta), \operatorname{grad}_{\boldsymbol{\mu}_2} f(\Theta), \dots, \operatorname{grad}_{\boldsymbol{\mu}_K} f(\Theta), \\ &\operatorname{grad}_{\Sigma_1} f(\Theta), \operatorname{grad}_{\Sigma_2} f(\Theta), \dots, \operatorname{grad}_{\Sigma_K} f(\Theta) \big) \end{aligned}$$

が成り立ちます．以下では $\operatorname{grad}_{\boldsymbol{w}} f(\Theta)$, $\operatorname{grad}_{\boldsymbol{\mu}_k} f(\Theta)$, $\operatorname{grad}_{\Sigma_k} f(\Theta)$ をそれぞれ求めます．

まず，$\operatorname{grad}_{\boldsymbol{w}} f(\Theta)$ を求めるために，$\mathbb{R}^K$ の開部分多様体 $\mathbb{R}^K_{++}$ 上で定義された関数 $\boldsymbol{w} \mapsto f(\Theta)$ を $f_0 \colon \mathbb{R}^K_{++} \to \mathbb{R}$ とおくと，f_0 のユークリッド勾配は

$$\operatorname{grad}^{\mathrm{E}} f_0(\boldsymbol{w}) = -\sum_{t=1}^{T} \frac{[p_n(\boldsymbol{x}_t; \boldsymbol{\mu}_1, \Sigma_1) \quad \cdots \quad p_n(\boldsymbol{x}_t; \boldsymbol{\mu}_K, \Sigma_K)]^\top}{\sum_{k=1}^{K} w_k p_n(\boldsymbol{x}_t; \boldsymbol{\mu}_k, \Sigma_k)}$$

です．書籍の 9.4 節のように，$\boldsymbol{w} = [w_k] \in \mathbb{R}^K_{++}$，$\boldsymbol{\xi} = [\xi_k]$, $\boldsymbol{\eta} = [\eta_k] \in T_{\boldsymbol{w}} \mathbb{R}^K_{++} = \mathbb{R}^K$ に対して $\langle \boldsymbol{\xi}, \boldsymbol{\eta} \rangle_{\boldsymbol{w}} = \sum_{k=1}^{K} \dfrac{\xi_k \eta_k}{w_k}$ により定まる $\mathbb{R}^K_{++}$ のリーマン計量を考えます．いま，$\operatorname{grad}^{\mathrm{E}} f_0(\boldsymbol{w}) = [g_k]$ とし，また一般に $\boldsymbol{u} = [u_k]$, $\boldsymbol{v} = [v_k] \in \mathbb{R}^K$ のアダマール積が $\boldsymbol{u} \odot \boldsymbol{v} \coloneqq [u_k v_k] \in \mathbb{R}^K$ と定義されることに注意すると，任意の $\boldsymbol{d} = [d_k] \in \mathbb{R}^K$ に対して，

$$\mathrm{D} f_0(\boldsymbol{w})[\boldsymbol{d}] = \operatorname{grad}^{\mathrm{E}} f_0(\boldsymbol{w})^\top \boldsymbol{d} = \sum_{k=1}^{K} g_k d_k = \sum_{k=1}^{K} \frac{(w_k g_k) d_k}{w_k} = \langle \boldsymbol{w} \odot \operatorname{grad}^{\mathrm{E}} f_0(\boldsymbol{w}), \boldsymbol{d} \rangle_{\boldsymbol{w}}$$

が成り立ちます．よって，リーマン多様体 $(\mathbb{R}^K_{++}, \langle \cdot, \cdot \rangle)$ 上の関数 f_0 のリーマン勾配は

$$\operatorname{grad} f_0(\boldsymbol{w}) = \boldsymbol{w} \odot \operatorname{grad}^{\mathrm{E}} f_0(\boldsymbol{w})$$

となります．さらに，f_0 を $\mathbb{R}^K_{++}$ のリーマン部分多様体 Δ_K に制限した関数 $f_0|_{\Delta_K}$ のリーマン勾配は，書籍の命題 6.3.1 より，直交射影 $P_{\boldsymbol{w}} \colon T_{\boldsymbol{w}} \mathbb{R}^K_{++} \to T_{\boldsymbol{w}} \Delta_K$ を用いて $\operatorname{grad} f_0|_{\Delta_K}(\boldsymbol{w}) = P_{\boldsymbol{w}}\big(\operatorname{grad} f_0(\boldsymbol{w}) \big)$ と表せます．このリーマン勾配が $\operatorname{grad}_{\boldsymbol{w}} f(\Theta)$ に他ならず，さらに書籍の 9.4 節から $P_{\boldsymbol{w}}(\boldsymbol{d}) = \boldsymbol{d} - (\mathbf{1}_K^\top \boldsymbol{d}) \boldsymbol{w}$ が成り立つことに注意すると，$\mathbf{1}_K^\top \big(\boldsymbol{w} \odot \operatorname{grad}^{\mathrm{E}} f_0(\boldsymbol{w}) \big) = \sum_{k=1}^{K} w_k g_k = \boldsymbol{w}^\top \operatorname{grad}^{\mathrm{E}} f_0(\boldsymbol{w}) = -\sum_{t=1}^{T} 1 = -T$ より，

$$\begin{aligned} \operatorname{grad}_{\boldsymbol{w}} f(\Theta) &= P_{\boldsymbol{w}}\big(\operatorname{grad} f_0(\boldsymbol{w}) \big) = \operatorname{grad} f_0(\boldsymbol{w}) - \big(\mathbf{1}_K^\top \operatorname{grad} f_0(\boldsymbol{w}) \big) \boldsymbol{w} \\ &= \boldsymbol{w} \odot \operatorname{grad}^{\mathrm{E}} f_0(\boldsymbol{w}) - \big(\boldsymbol{w}^\top \operatorname{grad}^{\mathrm{E}} f_0(\boldsymbol{w}) \big) \boldsymbol{w} \\ &= \boldsymbol{w} \odot \operatorname{grad}^{\mathrm{E}} f_0(\boldsymbol{w}) + T \boldsymbol{w} \end{aligned} \tag{2}$$

が得られます．

次に，$\operatorname{grad}_{\boldsymbol{\mu}_k} f(\Theta)$ を求めます．これは，$\mathbb{R}^n$ 上の関数 $\boldsymbol{\mu}_k \mapsto f(\Theta)$ のユークリッド勾配と一致するので容易に計算でき，

$$\operatorname{grad}_{\boldsymbol{\mu}_k} f(\Theta) = -\sum_{t=1}^{T} \frac{w_k p_n(\boldsymbol{x}_t; \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{l=1}^{K} w_l p_n(\boldsymbol{x}_t; \boldsymbol{\mu}_l, \Sigma_l)} \Sigma_k^{-1}(\boldsymbol{x}_t - \boldsymbol{\mu}_k)$$

となります．

最後に，$\operatorname{grad}_{\Sigma_k} f(\Theta)$ を求めます．まず，補助的に n 次正則行列全体（一般線形群）$\mathbb{R}_*^{n\times n}$ 上で定義された関数 $g(X) \coloneqq \frac{1}{\sqrt{\det X}} \exp(-\frac{1}{2}\boldsymbol{a}^\top X^{-1}\boldsymbol{a})$（$\boldsymbol{a} \in \mathbb{R}^n$ は定数ベクトル）のユークリッド勾配を計算すると，$X^{-\top} \coloneqq (X^{-1})^\top$ として

$$\operatorname{grad} g(X) = \frac{\exp\left(-\frac{1}{2}\boldsymbol{a}^\top X^{-1}\boldsymbol{a}\right)}{2\sqrt{\det X}} (X^{-\top}\boldsymbol{a}\boldsymbol{a}^\top X^{-\top} - X^{-\top}) \tag{3}$$

となることがわかります（式 (3) の証明は本節の末尾で行います）．このことに注意し，$\operatorname{Sym}_{++}(n)$ 上の関数 $\Sigma_k \mapsto f(\Theta)$ を $f_k\colon \operatorname{Sym}_{++}(n) \to \mathbb{R}$（$k = 1, 2, \ldots, K$）とおくと，$f_k$ のユークリッド勾配は (3) より

$$\operatorname{grad}^{\mathrm{E}} f_k(\Sigma_k) = -\frac{w_k}{2\sqrt{(2\pi)^n \det \Sigma_k}} \sum_{t=1}^{T} \left(\frac{\exp\left(-\frac{1}{2}(\boldsymbol{x}_t - \boldsymbol{\mu}_k)^\top \Sigma_k^{-1}(\boldsymbol{x}_t - \boldsymbol{\mu}_k)\right)}{\sum_{l=1}^{K} w_l p_n(\boldsymbol{x}_t; \boldsymbol{\mu}_l, \Sigma_l)} \times \left(\Sigma_k^{-1}(\boldsymbol{x}_t - \boldsymbol{\mu}_k)(\boldsymbol{x}_t - \boldsymbol{\mu}_k)^\top \Sigma_k^{-1} - \Sigma_k^{-1}\right) \right) \tag{4}$$

となります．ここで，書籍の式 (9.26) のように，$X \in \operatorname{Sym}_{++}(n)$ における $\xi, \eta \in T_X \operatorname{Sym}_{++}(n) = \operatorname{Sym}(n)$ に対して $\langle \xi, \eta \rangle_X = \operatorname{tr}(X^{-1}\xi X^{-1}\eta)$ により定まる $\operatorname{Sym}_{++}(n)$ のリーマン計量を考えます．すると，書籍の 9.4 節と同様の議論により，$\operatorname{Sym}_{++}(n)$ 上のユークリッド勾配 $\operatorname{grad}^{\mathrm{E}} f_k$ とリーマン勾配 $\operatorname{grad} f_k$ の間に $\operatorname{grad} f_k(\Sigma_k) = \Sigma_k \operatorname{grad}^{\mathrm{E}} f_k(\Sigma_k)\Sigma_k$ の関係が成り立ちます．ゆえに，

$$\operatorname{grad} f_k(\Sigma_k) = -\frac{w_k}{2\sqrt{(2\pi)^n \det \Sigma_k}} \sum_{t=1}^{T} \left(\frac{\exp\left(-\frac{1}{2}(\boldsymbol{x}_t - \boldsymbol{\mu}_k)^\top \Sigma_k^{-1}(\boldsymbol{x}_t - \boldsymbol{\mu}_k)\right)}{\sum_{l=1}^{K} w_l p_n(\boldsymbol{x}_t; \boldsymbol{\mu}_l, \Sigma_l)} \times \left((\boldsymbol{x}_t - \boldsymbol{\mu}_k)(\boldsymbol{x}_t - \boldsymbol{\mu}_k)^\top - \Sigma_k\right) \right) \tag{5}$$

が成り立ちます．

なお，Manopt に基づいて目的関数の勾配を実装する際には，`problem.egrad` にユークリッド勾配を定義するだけでもかまいません．ただし，あらかじめリーマン勾配を導出して実装することで，計算効率を高められる場合があります．実際，今回の目的関数については，$\boldsymbol{w}^\top \operatorname{grad}^{\mathrm{E}} f_0(\boldsymbol{w}) = -T$ を用いて得られた (2) を用いると，$\operatorname{grad} f_0(\boldsymbol{w})$ を計算してから $P_{\boldsymbol{w}}$ による射影の計算を行うより，$\mathbf{1}_K^\top \operatorname{grad} f_0(\boldsymbol{w})$（$= -T$）の数値計算を省略できるため効率的です．また，$\operatorname{grad}^{\mathrm{E}} f_k(\Sigma_k)$ を経由せずに式 (5) により直接 $\operatorname{grad} f_k(\Sigma_k)$ を計算すると，式 (4) の右辺に現れる逆行列 Σ_k^{-1} を計算する必要がありません．なお，$\Sigma_k^{-1}(\boldsymbol{x}_t - \boldsymbol{\mu}_k)$ は連立 1 次方程式 $\Sigma_k \boldsymbol{y} = \boldsymbol{x}_t - \boldsymbol{\mu}_k$ を $\boldsymbol{y} \in \mathbb{R}^n$ について解くことで計算できるため，その過程で逆行列 Σ_k^{-1} 自体の計算は不要であることに注意してください．以上を踏まえて，サンプルコード `SymmetricPositiveDefiniteManifold.m` では，`problem.grad` として目的関数のリーマン勾配を定義しています．

本節の最後に式 (3) を証明します．まず，$X \in \mathbb{R}_*^{n\times n}$ に対して $\mathrm{inv}(X) \coloneqq X^{-1}$ と定義し，$Y \in \mathbb{R}^{n\times n}$ について g の方向微分を計算すると，

$$\begin{aligned}
\mathrm{D}g(X)[Y] &= \frac{d}{dt} g(X+tY)\Big|_{t=0} \\
&= \frac{d}{dt} \frac{1}{\sqrt{\det(X+tY)}} \exp\Big(-\frac{1}{2}\boldsymbol{a}^\top \mathrm{inv}(X+tY)\boldsymbol{a}\Big)\Big|_{t=0} \\
&= \frac{\exp\big(-\frac{1}{2}\boldsymbol{a}^\top X^{-1}\boldsymbol{a}\big)\Big(-\frac{1}{2}\boldsymbol{a}^\top\big(\mathrm{D}\,\mathrm{inv}(X)[Y]\big)\boldsymbol{a}\sqrt{\det(X)} - \frac{\mathrm{D}\det(X)[Y]}{2\sqrt{\det(X)}}\Big)}{\det(X)} \\
&= -\frac{\exp\big(-\frac{1}{2}\boldsymbol{a}^\top X^{-1}\boldsymbol{a}\big)}{2(\det X)^{3/2}}\Big(\boldsymbol{a}^\top\big(\mathrm{D}\,\mathrm{inv}(X)[Y]\big)\boldsymbol{a}\cdot\det X + \mathrm{D}\det(X)[Y]\Big) \quad (6)
\end{aligned}$$

となります．ここで，$\mathrm{inv}(X)X = I_n$ の Y についての方向微分を計算することで，$\mathrm{D}\,\mathrm{inv}(X)[Y]\cdot X + \mathrm{inv}(X)\cdot Y = 0$，すなわち

$$\mathrm{D}\,\mathrm{inv}(X)[Y] = -X^{-1}YX^{-1} \tag{7}$$

を得ます．一方，$\det(X+tY) = \det\big(X(I_n + tX^{-1}Y)\big) = \det X \det(I_n + tX^{-1}Y)$ に注意すると，

$$\mathrm{D}\det(X)[Y] = \lim_{t\to 0}\frac{\det(X+tY)-\det X}{t} = \det X\cdot\frac{d}{dt}\det(I_n+tX^{-1}Y)\Big|_{t=0}$$

が成り立ちます．ここで，$X^{-1}Y =: Z = [z_{ij}]$ とおくと，$I_n = [\delta_{ij}]$ と行列式の定義から

$$\det(I_n + tX^{-1}Y) = \det(I_n+tZ) = \sum_{\sigma\in\mathfrak{S}_n}(\delta_{1\sigma(1)}+tz_{1\sigma(1)})\cdots(\delta_{n\sigma(n)}+tz_{n\sigma(n)})$$

であり（$\mathfrak{S}_n$ は書籍の例 A.4.2 の n 次対称群），その t の 1 次の項の係数は

$$\sum_{\sigma\in\mathfrak{S}_n} z_{1\sigma(1)}\delta_{2\sigma(2)}\dots\delta_{n\sigma(n)} + \cdots + \delta_{1\sigma(1)}\delta_{2\sigma(2)}\dots z_{n\sigma(n)} = z_{11}+\cdots+z_{nn} = \mathrm{tr}(Z)$$

なので，$\dfrac{d}{dt}\det(I_n+tX^{-1}Y)\Big|_{t=0} = \mathrm{tr}(Z) = \mathrm{tr}(X^{-1}Y)$ となります．よって，

$$\mathrm{D}\det(X)[Y] = \det X\cdot\mathrm{tr}(X^{-1}Y) \tag{8}$$

を得ます．式 (6)，(7)，(8) より，

$$\begin{aligned}
\mathrm{D}g(X)[Y] &= -\frac{\exp\big(-\frac{1}{2}\boldsymbol{a}^\top X^{-1}\boldsymbol{a}\big)}{2(\det X)^{3/2}}\Big(-(\boldsymbol{a}^\top X^{-1}YX^{-1}\boldsymbol{a})\det X + \det X\,\mathrm{tr}(X^{-1}Y)\Big) \\
&= \frac{\exp\big(-\frac{1}{2}\boldsymbol{a}^\top X^{-1}\boldsymbol{a}\big)}{2\sqrt{\det X}}\,\mathrm{tr}\big((X^{-1}\boldsymbol{a}\boldsymbol{a}^\top X^{-1} - X^{-1})Y\big) \\
&= \mathrm{tr}\left(\frac{\exp\big(-\frac{1}{2}\boldsymbol{a}^\top X^{-1}\boldsymbol{a}\big)}{2\sqrt{\det X}}(X^{-\top}\boldsymbol{a}\boldsymbol{a}^\top X^{-\top} - X^{-\top})^\top Y\right)
\end{aligned}$$

を得ます．上式の 2 番目の等号では $\boldsymbol{a}^\top X^{-1}YX^{-1}\boldsymbol{a}$ が実数であることに注意して，$\boldsymbol{a}^\top X^{-1}YX^{-1}\boldsymbol{a} = \mathrm{tr}\big(\boldsymbol{a}^\top X^{-1}YX^{-1}\boldsymbol{a}\big) = \mathrm{tr}(X^{-1}\boldsymbol{a}\boldsymbol{a}^\top X^{-1}Y)$ となることを用いました．ところで，g の X におけるユークリッド勾配 $\mathrm{grad}\,g(X)$ は任意の $Y\in\mathbb{R}^{n\times n}$ に対して $\mathrm{D}g(X)[Y] = \mathrm{tr}\big(\mathrm{grad}\,g(X)^\top Y\big)$ を満たすので，式 (3) が導かれました．

III.6　球面上の拡張ラグランジュ法（書籍 10.5 節，10.6 節）

本節では，書籍の 10.5 節の拡張ラグランジュ法を 10.6 節の球面上の制約付き最適化問題 (10.25) に適用する例を紹介します．

書籍の (10.25) で定義される問題は等式制約をもたないため，拡張ラグランジュ関数は

$$L_\rho(\boldsymbol{x},\boldsymbol{\mu}) \coloneqq f(\boldsymbol{x}) + \frac{\rho}{2}\left\| \max\left\{\boldsymbol{0}, \frac{\boldsymbol{\mu}}{\rho} + \boldsymbol{g}(\boldsymbol{x})\right\}\right\|_2^2 = -\boldsymbol{x}^\top A\boldsymbol{x} + \frac{\rho}{2}\left\| \max\left\{\boldsymbol{0}, \frac{\boldsymbol{\mu}}{\rho} - \boldsymbol{x}\right\}\right\|_2^2$$

と定義されます．書籍の演習問題 10.5 より，ρ と $\boldsymbol{\mu}$ を固定したときの $\mathbb{R}^n$ 上の関数 $\bar{l}(\boldsymbol{x}) \coloneqq \bar{f}(\boldsymbol{x}) + \frac{\rho}{2}\left\| \max\left\{\boldsymbol{0}, \frac{\boldsymbol{\mu}}{\rho} + \bar{\boldsymbol{g}}(\boldsymbol{x})\right\}\right\|_2^2 = -\boldsymbol{x}^\top A\boldsymbol{x} + \frac{\rho}{2}\left\| \max\left\{\boldsymbol{0}, \frac{\boldsymbol{\mu}}{\rho} - \boldsymbol{x}\right\}\right\|_2^2$（ただし $\bar{f}(\boldsymbol{x}) = -\boldsymbol{x}^\top A\boldsymbol{x}$，$\bar{\boldsymbol{g}}(\boldsymbol{x}) = -\boldsymbol{x}$）のユークリッド勾配は，

$$\operatorname{grad}\bar{l}(\boldsymbol{x}) = \operatorname{grad}\bar{f}(\boldsymbol{x}) + \rho \mathrm{D}\boldsymbol{g}(\boldsymbol{x})^\top \max\left\{\boldsymbol{0}, \frac{\boldsymbol{\mu}}{\rho} + \boldsymbol{g}(\boldsymbol{x})\right\} = -2A\boldsymbol{x} - \rho \max\left\{\boldsymbol{0}, \frac{\boldsymbol{\mu}}{\rho} - \boldsymbol{x}\right\}$$

となります．よって，球面上の関数 $l(\boldsymbol{x}) \coloneqq \bar{l}|_{S^{n-1}}(\boldsymbol{x}) = L_\rho(\boldsymbol{x},\boldsymbol{\mu})$ のリーマン勾配は

$$\operatorname{grad} l(\boldsymbol{x}) = (I_n - \boldsymbol{x}\boldsymbol{x}^\top)\operatorname{grad}\bar{l}(\boldsymbol{x}) = -(I_n - \boldsymbol{x}\boldsymbol{x}^\top)\left(2A\boldsymbol{x} + \rho \max\left\{\boldsymbol{0}, \frac{\boldsymbol{\mu}}{\rho} - \boldsymbol{x}\right\}\right)$$

であることがわかります．また，リーマン多様体 $\mathcal{M}$ 上の拡張ラグランジュ法（書籍のアルゴリズム 10.1）を，等式制約がない特別な場合に改めて書き下すと，次のアルゴリズム III.1 のように記述することができます．

アルゴリズム III.1 リーマン多様体 $\mathcal{M}$ 上の制約付き問題（書籍の (10.1)）に対する等式制約がない場合の拡張ラグランジュ法

Input: 初期点 $(x_0, \boldsymbol{\mu}_0) \in \mathcal{M} \times \mathbb{R}^m$，$\varepsilon_0 > 0$，$\rho_0 > 0$，$\theta_\varepsilon \in (0,1)$，$\theta_\rho > 1$，$\theta_\sigma \in (0,1)$，$\boldsymbol{\mu}_{\max} \in \mathbb{R}^m$（$\boldsymbol{\mu}_{\max} \geq \boldsymbol{0}$）

Output: $\mathcal{M}$ 上の点列 $\{x_k\}$

for $k = 0, 1, 2, \ldots$ **do**
  $l_k(x) \coloneqq L_{\rho_k}(x, \boldsymbol{\mu}_k)$ の近似的な最小点を x_{k+1} とする．ただし，$\|\operatorname{grad} l_k(x_{k+1})\|_{x_{k+1}} < \varepsilon_k$ を満たすようにする
  $\boldsymbol{\mu}_{k+1} \coloneqq \Pi_{[\boldsymbol{0}, \boldsymbol{\mu}_{\max}]}\big(\boldsymbol{\mu}_k + \rho_k \boldsymbol{g}(x_{k+1})\big)$
  $\boldsymbol{\sigma}_{k+1} \coloneqq \max\{\boldsymbol{g}(x_{k+1}), -\boldsymbol{\mu}_k/\rho_k\}$
  $\varepsilon_{k+1} \coloneqq \theta_\varepsilon \varepsilon_k$
  if $k = 0$ **then**
    $\rho_{k+1} \coloneqq \rho_k$
  else if $\|\boldsymbol{\sigma}_{k+1}\|_\infty \leq \theta_\sigma \|\boldsymbol{\sigma}_k\|_\infty$ **then**
    $\rho_{k+1} \coloneqq \rho_k$
  else
    $\rho_{k+1} \coloneqq \theta_\rho \rho_k$
  end if
end for

以上のことに注意して，アルゴリズム III.1 を実装したものがサポートページのディレクトリ `chapter10` の `AugmentedLagrangian_Inequality.m` であり，非負主成分分析に

対応する最適化問題（書籍の 10.6 節の，球面上の非負制約付き問題 (10.25)）に対して拡張ラグランジュ法を実行するコードが `NonnegativeSphere.m` です．

`NonnegativeSphere.m` では，拡張ラグランジュ法のパラメータを設定し，関数 M ファイル `AugmentedLagrangian_Inequality.m` で定義される関数 `AugmentedLagrangian_Inequality` を呼び出して拡張ラグランジュ法を適用しています．なお，拡張ラグランジュ法のパラメータについては，書籍の参考文献 [25] (C. Liu and N. Boumal: Simple algorithms for optimization on Riemannian manifolds with constraints, Applied Mathematics & Optimization, Vol.82, pp.949–981 (2020)) および，同論文の著者である Changshuo Liu 氏によるコード（同氏の GitHub ページ

https://github.com/losangle/

の多様体上の制約付きアルゴリズムに関するリポジトリ

`Optimization-on-manifolds-with-extra-constraints`

で公開されている `solvers/almbddmultiplier.m`）を参考にしました．

一方，`AugmentedLagrangian_Inequality.m` は，アルゴリズム III.1 をその通りに実装したものです．詳細な説明はコード中のコメント文を参照してください．ここでは，各反復において部分問題を解く部分，すなわち関数 $l_k\colon \mathcal{M} \to \mathbb{R}$ の近似的な最小点 x_{k+1} を，$\|\operatorname{grad} l_k(x_{k+1})\|_{x_{k+1}} < \varepsilon_k$ を満たすように求める部分についてのみ説明します．この計算に該当するのは，`for` 文の内部にある次の部分です．

```
problem.cost = @(x) L(x, mu, rho);
problem.egrad = @(x) egradf(x) + rho*Dg(x)'*max(0,mu/rho+g(x));
options.tolgradnorm = epsilon;
[x, ~, ~] = rlbfgs(problem,x,options);
```

これは，拡張ラグランジュ法の反復（外部反復）ごとに解くべき最適化問題が異なることにさえ注意すれば，本資料の II 章で説明した `basicexample.m` と同様に理解することができます．目的関数 `problem.cost` を，拡張ラグランジュ関数 L_ρ において $\boldsymbol{\mu}_k$ と $\rho = \rho_k$ を固定して得られる関数 l_k とし，そのユークリッド勾配を `problem.egrad` に指定しています．そして，外部反復の番号 k に応じて部分問題を解くべき精度 ε_k が変化することに注意して，`options.tolgradnorm` を定義しています．最後に，`rlbfgs` により，準ニュートン法（書籍の 8.4.1 項）を用いて部分問題を解いています．

本節の最後に，`NonnegativeSphere.m` を実行することで，$n = 5$ の場合にランダムに生成した対称行列 $A \in \mathrm{Sym}(n)$ についての最適化問題（書籍の (10.25)）を解いた例を紹介します．$K := 35$ 反復で $\varepsilon_K = 10^{-6}$ となるように拡張ラグランジュ法を実行したところ，

$$\boldsymbol{x}_K = [0.0000 \quad 0.2119 \quad 0.6065 \quad 0.5525 \quad 0.5310]^\top \in S^{n-1}$$

となりました．また，このとき $(I_n - \boldsymbol{x}_K\boldsymbol{x}_K^T)A\boldsymbol{x}_K$ を計算すると，

$$(I_n - \boldsymbol{x}_K\boldsymbol{x}_K^T)A\boldsymbol{x}_K = [-0.2497 \quad 0.0000 \quad -0.0000 \quad 0.00000 \quad 0.0000]^\top$$

となりました．なお，右辺のベクトルの成分のうち最大のものは 1.022×10^{-7}（≈ 0）でした．したがって，得られた解 $\boldsymbol{x}_K$ は，書籍の (10.25) の球面上の最適化問題に対するKKT 条件（書籍の例 10.4.1）から導かれる最適性の必要条件（書籍の 10.6 節）

$$\boldsymbol{x}_K \geq \mathbf{0}, \quad (I_n - \boldsymbol{x}_K \boldsymbol{x}_K^\top) A \boldsymbol{x}_K \leq \mathbf{0}$$

を高い精度で満たすことがわかります.