

ETH Lectures 401-0663-00L Numerical Methods for Computer Science
401-2673-00L Numerical Methods for CSE

Homework Problems and Projects

Prof. R. Hiptmair, SAM, ETH Zurich

Autumn Term 2021, Version of October 2, 2022
(C) Seminar für Angewandte Mathematik, ETH Zürich

[Link](#) to the current version of this homework collection

0.1 General Information

0.1.1 Weekly Homework Assignments

All problems will be published in this single “.pdf” file. Every week, we publish a list of problem numbers on the lecture Moodle page. These are the current assignments for that week. Details are given on the lecture Moodle page.

0.1.2 Importance of Homework

Homework assignments are **not** mandatory. However, it is **very important** that you constantly exercise with the material you learn. “Solving” the homework assignments one week before the main exam by looking at the solutions will likely result in failure.

We provide hints and solutions from the beginning. It is your responsibility to look at the solutions only if you are stuck with a difficult problem and you tried to find a solution for a sufficient amount of time.

Most homework problems come with an estimate for the time required for its solution. These times are what we expect a bright and well-prepared student to require for that problem during an exam.

0.1.3 Corrections and Grading of Assignments

Homework assignments will not be graded.

However, you may submit your handwritten solutions by handing them to your tutor directly or uploading them via the Moodle upload interface, see the lecture Moodle page for instructions. The assistants will

Problem 3-9: Matrix Fitting by Constrained Least Squares

In this problem we look at a particular *constrained* linear least squares problem, as they have been discussed in [Lecture → Section 3.6]. In particular, we will study the augmented normal equation approach to solve such type of least squares problem, see [Lecture → Section 3.6.1]. In this context we refresh our understanding of the Lagrange multiplier technique.

Related to [Lecture → Section 3.6]; involves implementation with EIGEN.

Consider the following problem of finding the minimal-norm matrix fitting a linear system of equations with given solution and right-hand side vector:

$$\text{Given } n \in \mathbb{N}, \mathbf{z} \in \mathbb{R}^n, \mathbf{g} \in \mathbb{R}^n, \text{ find } \mathbf{M}^* = \underset{\mathbf{M} \in \mathbb{R}^{n,n}, \mathbf{M}\mathbf{z}=\mathbf{g}}{\operatorname{argmin}} \|\mathbf{M}\|_F, \quad (3.9.1)$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix as introduced in [Lecture → Def. 3.4.4.17].

Definition [Lecture → Def. 3.4.4.17]. Frobenius norm

The **Frobenius norm** of $\mathbf{A} \in \mathbb{K}^{m,n}$ is defined as

$$\|\mathbf{A}\|_F^2 := \sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2.$$

(3-9.a)  (30 min.) Reformulate the problem as an equivalent standard **linearly constrained least squares problem** [Lecture → Eq. (3.6.0.1)]

$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathbb{R}^N, \mathbf{C}\mathbf{x}=\mathbf{d}}{\operatorname{argmin}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2, \quad (3.9.2)$$

for suitable matrices $\mathbf{A}$, $\mathbf{C}$ and vectors $\mathbf{b}$ and $\mathbf{d}$, see [Lecture → Eq. (3.6.0.1)]. These matrices and vectors have to be specified based on $\mathbf{z}$ and $\mathbf{g}$.

SOLUTION of (3-9.a):

We denote $\mathbf{m}^* := \operatorname{vec}(\mathbf{M}^{*\top}) \in \mathbb{R}^{n^2}$ the vector obtained by the concatenation of the rows of $\mathbf{M}^*$ [Lecture → Rem. 1.2.3.4]. Then, since $\|\mathbf{M}^*\|_F = \|\mathbf{m}^*\|_2$, the functional to be minimised is just the 2-norm of $\mathbf{m}^*$, i.e., $\mathbf{A} = \operatorname{Id}_{n^2}$ and $\mathbf{b} = \mathbf{0}$. The constraint $\mathbf{M}^*\mathbf{z} = \mathbf{g}$ can be written as $\mathbf{C}\mathbf{m}^* = \mathbf{d}$, where $\mathbf{d} = \mathbf{g}$ and the $n \times n^2$ matrix $\mathbf{C} := \operatorname{Id}_n \otimes \mathbf{z}^\top$ is the **Kronecker product** of the $n \times n$ identity matrix and $\mathbf{z}^\top$. More explicitly,

$$\mathbf{C} = \begin{bmatrix} \mathbf{z}^\top & \mathbf{0}^\top & \cdots & \mathbf{0}^\top \\ \mathbf{0}^\top & \mathbf{z}^\top & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0}^\top & \cdots & \mathbf{0}^\top & \mathbf{z}^\top \end{bmatrix} \in \mathbb{R}^{n,n^2}. \quad (3.9.3)$$

▲

(3-9.b)  (15 min.) [depends on Sub-problem (3-9.a)]

State a necessary and sufficient condition for the matrix $\mathbf{C}$ found in Sub-problem (3-9.a) to possess full rank.

SOLUTION of (3-9.b):

$\mathbf{C}$ has full rank, if $\mathbf{z}$ has at least one non-zero component ζ , because in this case we can certainly find n linearly independent columns, because $\mathbf{C}$ contains $\zeta \mathbf{I} \in \mathbb{R}^{n,n}$ as a sub-matrix.

Equivalently, an argument can be made based on the n rows:

$$x_1 \begin{bmatrix} \mathbf{z} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ \mathbf{z} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \dots + x_n \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \mathbf{z} \end{bmatrix} = \mathbf{0}$$

has a unique solution $x_1 = x_2 = \dots = x_n = 0$, if $\mathbf{z} \neq \mathbf{0}$.



(3-9.c) 🕒 (15 min.) [depends on Sub-problem (3-9.a)]

State the **augmented normal equations** [Lecture → Eq. (3.6.1.6)] corresponding to the constrained linear least squares problem found in Sub-problem (3-9.a) and give necessary and sufficient conditions on $\mathbf{g}$ and $\mathbf{z}$ that ensure existence and uniqueness of solutions.

HINT 1 for (3-9.c): Generic **augmented normal equations** associated with (3.9.2) read

$$\mathbf{S} \begin{bmatrix} \mathbf{m}^* \\ \mathbf{p} \end{bmatrix} := \begin{bmatrix} \text{Id}_{n^2} & \mathbf{C}^\top \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{m}^* \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{g} \end{bmatrix} =: \mathbf{h}. \quad (3.9.4)$$

Then carry out a block Gaussian elimination [Lecture → Rem. 2.3.1.14] of the unknown $\mathbf{m}^*$. Which linear system for $\mathbf{p}$ emerges? ┘

SOLUTION of (3-9.c):

By block Gaussian elimination of the unknown $\mathbf{m}^*$ from the augmented normal equations (3.9.4) we obtain

$$-\mathbf{C}\mathbf{C}^\top \mathbf{p} = \mathbf{g}. \quad (3.9.5)$$

By [Lecture → Thm. 3.1.2.9] the matrix $\mathbf{C}^\top \mathbf{C}$ is regular, if and only if $\mathbf{C} \in \mathbb{R}^{p,n}$ has full rank p , see Sub-problem (3-9.b).



(3-9.d) 🕒 (30 min.) [depends on Sub-problem (3-9.c)]

Write a C++ function

```
Eigen::MatrixXd min_frob(const Eigen::VectorXd &z, const
    Eigen::VectorXd &g);
```

that computes the solution $\mathbf{M}^*$ of the minimization problem (3.9.1) given the vectors $\mathbf{z}, \mathbf{g} \in \mathbb{R}^n$. Use the augmented normal equations [Lecture → Eq. (3.6.1.6)] derived in ((3-9.c)).

It is convenient to use EIGEN's built-in Kronecker product class **Eigen::KroneckerProduct** (`#include <unsupported/Eigen/KroneckerProduct>` required).

SOLUTION of (3-9.d):

See code in Equation 3.9.6. As pointed out in the solution of Sub-problem (3-9.a) the matrix **C** is a Kronecker product.

C++ code 3.9.6: Solution of minimization problem

```

2  Eigen::MatrixXd min_frob(const Eigen::VectorXd &z, const Eigen::VectorXd &g) {
3      assert(z.size() == g.size() && "Size mismatch!");
4      const unsigned int n = g.size();
5      Eigen::MatrixXd M_mat = Eigen::MatrixXd::Zero(n, n);
6      // TODO: (3-9.d) solve the augmented normal equations
7      // Hint: Use Map to reshape a vector into a matrix
8      // START
9      // Build temporary matrix C.
10     Eigen::MatrixXd C =
11         kroneckerProduct(Eigen::MatrixXd::Identity(n, n), z.transpose());
12     Eigen::VectorXd minus_p = (C * C.transpose()).ldlt().solve(g);
13     Eigen::VectorXd M = C.transpose() * minus_p;
14     M_mat = Eigen::Map<Eigen::MatrixXd>(M.data(), n, n).transpose();
15     // END
16     return M_mat;
17 }

```

Get it on  GitLab (matrixfit.hpp).

This code is not optimal in terms of efficiency and memory usage: Of course, exploiting the special structure of the matrix **C** it is possible to solve the linear system of equations more efficiently.



(3-9.e)  (15 min.) [depends on Sub-problem (3-9.d)]

Using the test vectors $\mathbf{z} = [1, 2, \dots, n]^T \in \mathbb{R}^n$ and $\mathbf{g} = [1, -1, 1, \dots, 1, -1]^T \in \mathbb{R}^n$, implement a C++ function

```
bool testMformula(unsigned int n);
```

that checks in a numerical experiment whether

$$\mathbf{M} = \frac{\mathbf{g}\mathbf{z}^T}{\|\mathbf{z}\|_2^2} \quad (3.9.7)$$

is a solution to (3.9.1). To that end call the function `min_frob` implemented in Sub-problem (3-9.d), conduct the test and return **true** if it succeeds.

HINT 1 for (3-9.e): Heed the advice given in [Lecture $\rightarrow$ Rem. 1.5.3.15]. 

SOLUTION of (3-9.e):

See code in Code 3.9.8. Note that we test for the relative deviation of the two computed matrices with respect to the Frobeniusnorm, which is evaluated by EIGEN's `norm()` member function. If this test is

passed the norm of the difference between $\mathbf{M}$ and $\mathbf{M}^*$ is zero almost up to precision and we conclude that $\mathbf{M} = \mathbf{M}^*$.

C++ code 3.9.8: Testing code

```

2  bool testMformula(unsigned int n) {
3      constexpr double tol = 1e-10;
4      // TODO: (3-9.e) compare the formula: return true if the relative
5      // error in Frobenius norm is less than tol.
6      // Hint: Define z using LinSpaced and g using replicate()
7      // START
8      Eigen::VectorXd z = Eigen::VectorXd::LinSpaced(n, 1, n);
9      Eigen::Vector2d v = {1, -1};
10     Eigen::VectorXd g = v.replicate(static_cast<int>(n) / 2, 1);
11
12     Eigen::MatrixXd Mstar = min_frob(z, g);
13
14     Eigen::MatrixXd M = g * z.transpose() / z.squaredNorm(); // M
15     // Safe test for realtive smallness of error
16     return ((Mstar - M).norm() < tol * M.norm());
17     // END
18     return false;
19 }

```

Get it on  GitLab (matrixfit.hpp).



(3-9.f) (30 min.) [depends on Sub-problem (3-9.c)]

Now, give a rigorous proof that (3.9.7) gives a solution of (3.9.1), provided that $\mathbf{z} \neq \mathbf{0}$.

SOLUTION of (3-9.f):

We show that $\mathbf{M} = \frac{\mathbf{g}\mathbf{z}^\top}{\|\mathbf{z}\|_2^2}$ in vectorized form solves the augmented normal equations. To begin with note that

$$\mathbf{C}\mathbf{C}^\top = \|\mathbf{z}\|_2^2 \mathbf{I} \in \mathbb{R}^{p,p}.$$

Hence the solution of (3.9.4) is

$$\mathbf{p} = -\|\mathbf{z}\|_2^{-2} \mathbf{g}, \quad \mathbf{m}^* = \mathbf{C}^\top \|\mathbf{z}\|_2^{-2} \mathbf{g} = \|\mathbf{z}\|_2^{-2} \mathbf{g} \otimes \mathbf{z} \in \mathbb{R}^{n^2}. \quad (3.9.9)$$

Then “un-vectorize” $\mathbf{m}^*$!



So far, we have simply applied the formulas from [Lecture → Section 3.6.1] to (3.9.1) and its reformulation as a constrained linear least squares problem. Now we take a closer look at the method of Lagrangian multipliers, which was used to derive these equations in class.

(3-9.g) (20 min.) As in [Lecture → Eq. (3.6.1.2)] and [Lecture → Eq. (3.6.1.3)] from the lecture notes, find a **Lagrange functional** $L : \mathbb{R}^{n,n} \times \mathbb{R}^n \rightarrow \mathbb{R}$ such that:

$$\mathbf{M}^* = \operatorname{argmin}_{\mathbf{M} \in \mathbb{R}^{n,n}} \left\{ \sup_{\mathbf{m} \in \mathbb{R}^n} [L(\mathbf{M}, \mathbf{m})] \right\}. \quad (3.9.10)$$

The matrix $\mathbf{M}^*$ should provide the solution of (3.9.1).

HINT 1 for (3-9.g): Study [Lecture → Section 3.6.1] again and try to grasp the heuristic idea behind enforcing constraints by means of Lagrangian multipliers. ┘

SOLUTION of (3-9.g):

We want to find $L(\mathbf{M}, \mathbf{m})$ which:

1. attains its minimum value at $\mathbf{M}^*$ (which minimizes $\|\mathbf{M}\|_F$ and respects the condition $\mathbf{M}\mathbf{z} = \mathbf{g}$) and
2. is maximized by $\mathbf{m}$.

In general, $\mathbf{m}$ can make the functional L go to ∞ . To counterbalance its effect, it must be multiplied by a vector which is $\mathbf{0}$ when $\mathbf{M}\mathbf{z} = \mathbf{g}$, i.e. $\mathbf{M}\mathbf{z} - \mathbf{g}$. The functional L is therefore:

$$L(\mathbf{M}, \mathbf{m}) = \|\mathbf{M}\|_F^2 + \mathbf{m}^\top (\mathbf{M}\mathbf{z} - \mathbf{g}) .$$



(3-9.h)  (15 min.) Find the derivative $\text{grad } \Phi \in \mathbb{R}^{n,n}$ of the function Φ defined as:

$$\Phi : \mathbb{R}^{n,n} \rightarrow \mathbb{R}, \quad \Phi(\mathbf{X}) = \|\mathbf{X}\|_F^2 \quad (3.9.11)$$

HINT 1 for (3-9.h): You might be confused that you are asked to compute a gradient in the form of a matrix. However, function Φ takes matrices as arguments: in this subproblem matrices play the role of the usual vector arguments. Indeed, matrices can easily be identified with vectors by e.g. stacking their columns one on top of the other.

Hence, you may simply consider matrix argument of Φ as a vector written in a different way, compute the gradient of Φ from this perspective and, at the end, convert the result to the matrix form. ┘

SOLUTION of (3-9.h):

The simplest approach is by component-wise partial derivatives:

$$\frac{\partial}{\partial x_{ij}} \Phi(\mathbf{X}) = 2(\mathbf{X})_{ij} \Rightarrow \text{grad } \Phi(\mathbf{X}) = 2\mathbf{X} . \quad (3.9.12)$$



(3-9.i)  (20 min.) [depends on Sub-problem (3-9.h)]

Use the result of Sub-problem (3-9.h) to derive **saddle point equations** for $\text{grad } L(\mathbf{M}, \mathbf{m}) = \mathbf{0}$, for the L obtained in Sub-problem (3-9.g).

HINT 1 for (3-9.i): What are the necessary and sufficient condition for minimizing (3.9.10)? Remember [Lecture → Eq. (3.6.1.5)] and set

$$\frac{\partial L}{\partial \mathbf{M}} = \mathbf{0} \quad \text{and} \quad \frac{\partial L}{\partial \mathbf{m}} = \mathbf{0} .$$

These are the saddle-point equations. ┘

SOLUTION of (3-9.i):

Using $\text{grad } \Phi(\mathbf{X}) = 2\mathbf{X}$, the derivative of L w.r.t. $\mathbf{M}$ is:

$$\frac{\partial L}{\partial \mathbf{M}} = 2\mathbf{M} + \mathbf{m}\mathbf{z}^\top = 0. \quad (3.9.13)$$

The derivative of L w.r.t. $\mathbf{m}$ is:

$$\frac{\partial L}{\partial \mathbf{m}} = \mathbf{M}\mathbf{z} - \mathbf{g} = 0. \quad (3.9.14)$$



(3-9.j) 🕒 (15 min.) [depends on Sub-problem (3-9.i)]

Solve the saddle point equations obtained in Sub-problem (3-9.i) in symbolic form.

HINT 1 for (3-9.j): What special structure for $\mathbf{M}^*$, solution of (3.9.10), is implied by one of the saddle point equations? ┘

SOLUTION of (3-9.j):

Plugging (3.9.13) into (3.9.14) we obtain:

$$-\frac{1}{2}\mathbf{m}\|\mathbf{z}\|_2^2 - \mathbf{g} = 0 \quad \Rightarrow \quad \mathbf{m} = -2\frac{\mathbf{g}}{\|\mathbf{z}\|_2^2} \quad \xRightarrow{(3.9.13)} \quad \mathbf{M}^* = \frac{\mathbf{g}\mathbf{z}^\top}{\|\mathbf{z}\|_2^2}. \quad (3.9.15)$$

The matrix $\mathbf{M}^*$ is the tensor product of two vectors, see [Lecture → Section 1.3.1].



End Problem 3-9, 205 min.