Problem 1 *We are given that data* $(x_1, y_1), \ldots, (x_n, y_n)$ *where say,* $x_i \in [0, 1]$. *Fit the* best *polynomial* $y = p(x)$ *of degree 2 to fit this data.*

Solution

First, let us choose a basis for the possible set of functions. Since the degree is bounded by 2, a basis is the set $1, x, x^2$. The typical polynomial will look like $a_0 \cdot 1 + a_1 \cdot x + a_1 \cdot x^2$, where a_0, a_1 and a_2 have to be decided.

Next, let us define a metric which will be minimized.We choose

$$||y - p||^2 = \sum_{i=1}^{n} (y_i - p(x_i))^2$$

This is traditionally known as the **least square** problem. Expanding $p(x) = a_0 + a_1 x + a_2 x^2$, we have

$$err(a, y) = \sum_{i=1}^{n} (y_i - (a_0 + a_1 x_i + a_2 x_i^2))$$

Note that the error depends on the input data and the coefficients a. Also note that the function $err(a, y)$ is quadratic in a. If this has to be minimized, we may use that

$$\frac{\partial(err)}{\partial a_r} = 0 \text{ for } i = 0, 1, 2$$

This gives us

$$\sum_i ((y_i - a_0 - a_1 x_i - a_2 x_i^2) \cdot x_i^r) = 0$$

Expanding this we get:

$$\begin{bmatrix} \sum_i 1 & \sum_i x_i & \sum_i x_i^2 \\ \sum_i x_i & \sum_i x_i^2 & \sum_i x_i^3 \\ \sum_i x_i^2 & \sum_i x_i^3 & \sum_i x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum_i y_i \\ \sum_i y_i x_i \\ \sum_i y_i x_i^2 \end{bmatrix}$$

The solution of this linear system gives us the required answer.

If we were to increase the degree of the polynomial to d, the matrix inversion problem would become:

$$M \cdot a = b$$

where $M = (m_{rs}) = \sum_i x_i^{rs}$, $a = [a_0, \ldots, a_d]^T$ and $b = (b_r) = \sum_i y_i x_i^r$. Note that M is $(d+1) \times (d+1)$ and that a, b are column vectors of size $d+1$ corresponding to the variables $a_0, \ldots, a_d$. $\square$

Problem 2 *Again* $D = \{(x_i, y_i) | i = 1, \ldots, n\}$ *is a given set of data. Let* $f_1(x), f_2(x)$ *and* $f_3(x)$ *be three functions (say* $\sin(x), \cos(x)$ *and* $\tan(x)$*). Find the best linear combination*

$$a_1 f_1(x) + a_2 f_2(x) + a_3 f_3(x)$$

which fits the data.

Solution Forming the same error function as before, we see that the equation to be solved is:

$$\sum_i ((y_i - a_1 f_1(x_i) - a_2 f_2(x_i) - a_3 f_3(x_i)) \cdot f_r(x_i)) = 0$$

Expanding this and writing it in matrix notation, we get:

$$\begin{bmatrix} \sum_i f_1(x_i)f_1(x_i) & \sum_i f_1(x_i)f_2(x_i) & \sum_i f_1(x_i)f_3(x_i) \\ \sum_i f_2(x_i)f_1(x_i) & \sum_i f_2(x_i)f_2(x_i) & \sum_i f_2(x_i)f_3(x_i) \\ \sum_i f_3(x_i)f_1(x_i) & \sum_i f_3(x_i)f_2(x_i) & \sum_i f_3(x_i)f_3(x_i) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \sum_i y_i f_1(x_i) \\ \sum_i y_i f_2(x_i) \\ \sum_i y_i f_3(x_i) \end{bmatrix}$$

This may be solved to get the answer. Note that the coefficient matrix is always symmetric. Further, the approach may be generalized to any number of matrices. $\Box$

Problem 3 *Let $p_i = (x_i, y_i)$ be a set of points in the plane, and let z_i be an observation for this point p_i. Fit the best linear combination of the functions $1, xy$ and x^2y to this data.*

Solution: Let $f_1 = 1, f_2 = xy$ and $f_3 = x^2y$. As before, we form the error function:

$$err(a, z) = \sum_i (z_i - a_1 f_1(p_i) - a_2 f_2(p_i) - a_3 f_3(p_i))^2$$

Doing everything analogously, we get the equation:

$$\begin{bmatrix} \sum_i f_1(p_i)f_1(p_i) & \sum_i f_1(p_i)f_2(p_i) & \sum_i f_1(p_i)f_3(p_i) \\ \sum_i f_2(p_i)f_1(p_i) & \sum_i f_2(p_i)f_2(p_i) & \sum_i f_2(p_i)f_3(p_i) \\ \sum_i f_3(p_i)f_1(p_i) & \sum_i f_3(p_i)f_2(p_i) & \sum_i f_3(p_i)f_3(p_i) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \sum_i z_i f_1(p_i) \\ \sum_i z_i f_2(p_i) \\ \sum_i z_i f_3(p_i) \end{bmatrix}$$

Thus multi-dimensional least square approximation is as easy as the 1-dimensional case. $\Box$

Remark 4 *Let $\mathbb{R}^n$ be the usual n-dimensional vector space. An inner product on $\mathbb{R}^n$ is a function:*

$$<>: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$$

such that for any vectors $v_1, v_2, v_3 \in \mathbb{R}^n$ and any reals α, β, we have:

(i) $< v_1, v_2 >=< v_2, v_1 >$.

(ii) $< \alpha v_1 + \beta v_2, v_3 >= \alpha < v_1, v_3 > +\beta < v_2, v_3 >$.

(iii) $< v_1, v_1 >\geq 0$. *Further* $< v_1, v_1 >= 0$ *must imply that* $v_1 = 0$ *is the zero-vector.*

The quantity $< v, v >$ is always non-negative and its positive square-root is denoted by $\|v\|$ and is called the **length** *of the vector v. The above hypotheses implies many things:*

1. *The* **Cauchy-Schwartz** *inequality: for any vectors v, w, we have:*

$$< v, w >\leq \|v\|\|w\|$$

2. *When the elements of $\mathbb{R}^n$ are treated as row vectors, there is an $n \times n$-matrix X such that for all $v, w \in \mathbb{R}^n$, we have:*

$$< v, w >= v \cdot H \cdot w^T$$

Furthermore, this matrix H is symetric and **positive definite**, *i.e., all its eigenvalues are positive.*

Problem 5 *Let w and $v_1, \ldots, v_k$ be vectors in $\mathbb{R}^n$ equipped with an inner product as above. Find $a_1, \ldots, a_k$ such that*

$$\| \sum_i a_i v_i - w \|^2$$

is minimized. In other words, find the closest approximation to w in the linear span of the vectors $v_1, \ldots, v_k$, where the term closest *is given by the inner product.*

Solution: Note that:

$$\| \sum_i a_i v_i - w \|^2 =< \sum_i a_i v_i - w, \sum_i a_i v_i - w >$$

Thus, the objective function is again quadratic in the variables (a_i) with coefficients as various inner-products. This is solved by the following matrix equation:

$$Ma = b$$

where $M_{ij} =< v_i, v_j >$ and $b_r =< w, v_r >$. $\square$

Problem 6 *Let $D = \{(x_i, y_i) | i = 1, \ldots, n\}$ be some observations, with $x_i \in [0, 1]$ for all i. It is required to fit the best quadratic function $p(x) = a_2 x^2 + a_1 x + a_0$ to this data such that $p(x) \geq 0$ for all $x \in [0, 1]$.*

Solution: Note that this is very similar to 1 **except** for the additional positivity requirement. We may again form the error function:

$$err(a, y) = \sum_{i=1}^{n} (y_i - (a_0 + a_1 x_i + a_2 x_i^2))$$

The objective remains the same, i.e., minimize $err(a, y)$. However, now, the variables a_2, a_1 and a_0 may not be arbitrary. For example, the positivity requirement implies that:

$$\begin{array}{rcl} p(0) = & a_0 & \geq 0 \\ p(1) = & a_0 + a_1 + a_2 & \geq 0 \end{array}$$

It is easy to show that this inequalities are not sufficient. However, if one adds $a_2 \leq 0$, then the parabola in concave-downwards and then the requirements $p(0) \geq 0$ and $p(1) \geq 0$ are sufficient to ensure that $p(x) \geq 0$ for $x \in [0, 1]$. However, note that we are restricting the choices for a_i substantially. There are indeed positive choices which we have excluded.

In any case, let us look at the formulation that we have. Upto a constant term (the $\sum_i y_i^2$ term)

$$err(a,y) = \begin{bmatrix} a_0 & a_1 & a_2 \end{bmatrix} \begin{bmatrix} \sum_i 1 & \sum_i x_i & \sum_i x_i^2 \\ \sum_i x_i & \sum_i x_i^2 & \sum_i x_i^3 \\ \sum_i x_i^2 & \sum_i x_i^3 & \sum_i x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} - 2 \cdot \begin{bmatrix} a_0 & a_1 & a_2 \end{bmatrix} \begin{bmatrix} \sum_i y_i \\ \sum_i y_i x_i \\ \sum_i y_i x_i^2 \end{bmatrix}$$

Thus, we may express $err(a,y)$ as

$$\frac{1}{2} a H a^T + a \cdot f$$

where H is a real symmetric matrix, and in fact, the very familiar matrix from 1. The linear part of the cost function if f which is also the familiar RHS in 1. Furthermore, the linear inequalities to be obeyed are:

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

We may succintly express this as:

$$Aa^T \leq b$$

where A, b are the matrices above on either sides of the inequality. Thus we have:

$$\begin{array}{c} \min \frac{1}{2} a H a^T + a \cdot f \\ Aa^T \leq b \end{array}$$

This is the standard **Quadratic Programming** problem. We will use the notation $a = QP(H, f, A, b)$ to denote the solution to this problem. In addition, most solvers will also allow **equality constraints** of the form $Aeq \cdot a^T = beq$. These may arise from requirements such as $p(0.5) = 1$ which translates to

$$a_0 + 0.5a_1 + 0.25a_2 = 1$$

The general form for the QP is then:

$$a = QP(H, f, A, b, Aeq, beq)$$

which is what we will use. Note that there are two parts to the specification:

(i) The cost function, of which (a) the quadratic part is given by H, a real symmetric, positive definite matrix H, (b) and f, the linear part of the cost function.

(ii) The domain of optimization which is given by (a) the inequalties expressed by $Aa^T \leq b$ and (b) the strict equalities, given by $Aeq \cdot a^T = beq$.

Thus the cost function and the domain are separately specified. $\square$

Problem 7 *What is the nature of the domain and is there an effective and general method of specifying it?*

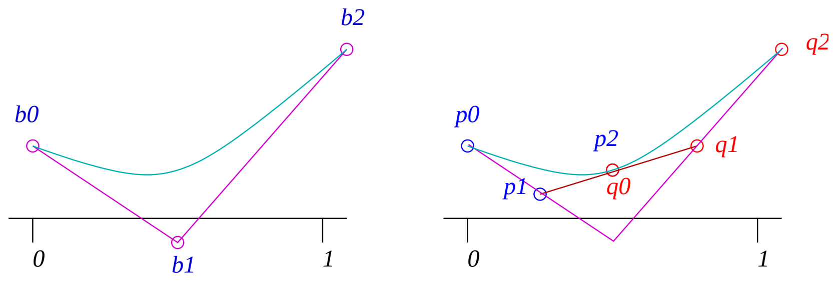

Figure 1: The Positivity Constraint

Solution: Let us tackle two sorts of constraints, (i) say $p(x) \geq c$, where c is a constant, and (ii) $p(\alpha) = d$, where d and α are again some constants.

We say that the vector $a = [a_0, a_1, a_2]$ satisfies these constraints if the polynomial $p_a(x) = a_0 + a_1 x + a_2 x^2$ satisfies the above constraints. Next, suppose that $a = [a_0, a_1, a_2]$ and $a' = [a'_0, a'_1, a'_2]$ satisfy these constraints, then so does the **convex combination** $a'' = \beta a + (1-\beta)a'$. In other words, since

$$\beta p_a(x) + (1-\beta)p_{a'}(x) = p_{a''}(x)$$

it must be that $p_{a''}(x) \geq c$ as well, wherever both p_a and $p_{a'}$ are. A similar analysis holds for the equality constraints. Thus the actual domain of definition is actually a convex domain.

Next, we set about approximating this domain by linear equalities and inequalities. We choose the **Bezier** basis $B_i^n(x) = \binom{n}{i} x^i (1-x)^{n-i}$ and express $p(x)$ as

$$p_b(x) = b_0 B_0^2(x) + b_1 B_1^2(x) + b_2 B_2^2(x)$$

A simple way of specifying $p_b(x) \geq c$ would be to assert $b_i \geq c$ for all i. Clearly, since $p_b(x)$ is a convex combination of the b_i's for $x \in [0,1]$, $b_i \geq c$ for all i implies that $p_b(x) \geq c$ for all $x \in [0,1]$. However, clearly, there may be a choice of control-points b_i's such that some are less than c yet the function $p_b(x) \geq c$ for all $x \in [0,1]$. An example, for $c = 0$ is shown in Fig. 1a below.

In the first part is shown a bezier curve in which not all control points are positive, yet the curve is wholly positive. However, on subdividing the curve at 0.5, we obtain the control points:

$p_0 =$	b_0
$p_1 =$	$0.5b_0 + 0.5b_1$
$p_2 =$	$0.25b_0 + 0.5b_1 + 0.25b_2$
$q_0 =$	$0.25b_0 + 0.5b_1 + 0.25b_2$
$q_1 =$	$0.5b_1 + 0.5b_2$
$q_2 =$	b_2

Thus, to ensure the $p_b(x) \geq 0$, we may say that $p_i \geq 0$ and $q_j \geq 0$ for all i, j. In the original variables, we thus have:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0.5 & 0.5 & 0 \\ 0.25 & 0.5 & 0.25 \\ 0.25 & 0.5 & 0.25 \\ 0 & 0.5 & 0.5 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

It is now clear that this procedure may be repeated *ad nauseum* to get an effective procedure of modelling the postivity constraint more and more accurately. It should also be apparent that such a procedure will work for higher degrees as well. This shows that the Bezier basis is effective in modelling such linear inequality constraints **via its control points**. This relationship between the control points and the curve is called the **variation-diminishing** property of Bezier curves. □

Problem 8 *What constraints $p_b(x)$ can be modelled in this way?*

Solutions: All constraints which (i) are modelled by the behaviour of control points, and (ii) which may be expressed in terms of **linear combinations** of the original cntrol points b.

To explain (i), note that we used the convexity property of a bezier curve with respect to its control points. In other words, a property of the control points ensures the same for the whole curve, **and** if the property is linearly expressible, then its is effectively modelled.

As an example, let us post the constraints that:

(i) **The first derivative $p_b'(x) \geq 0$ for $x \in [0, 0.6]$.** To begin, let $(\mathcal{C}, b)$ denotes the original curve with control points b. Let $(\mathcal{C}[0, 0.6], p)$ denote the restriction of this curve to $[0, 0.6]$ and let its control points be p. We obtain p as:

$$\begin{aligned} p_0 = &\quad b_0 \\ p_1 = &\quad 0.4b_0 + 0.6b_1 \\ p_2 = &\quad 0.16b_0 + 0.48b_1 + 0.36b_2 \end{aligned}$$

Recall that the derivative of $(\mathcal{C}[0, 0.6], p)$ is $(\mathcal{C}'[0, 0.6], q)$, where κ is a constant depending on $[0, 0.6]$ and the degree 2.

$$\begin{aligned} q_0 = &\quad \kappa(p_1 - p_0) &= \kappa(-0.6b_0 + 0.6b_1) \\ q_1 = &\quad \kappa(p_2 - p_1) &= \kappa(-0.24b_0 - 0.12b_1 + 0.36b_2) \end{aligned}$$

Thus, the constraint $p_b'(x) \geq 0$ for $x \in [0, 0.6]$ may be modelled as $q_0, q_1 \geq 0$. This, in the original variables becomes (the constant κ is irrelevant, since the RHS is zero):

$$\begin{aligned} -0.6b_0 + 0.6b_1 &\geq 0 \\ -0.24b_0 - 0.12b_1 + 0.36b_2 &\geq 0 \end{aligned}$$

(ii) **The second derivative** $p_b''(0.5) = 5.1$. If $(\mathcal{C}'', pp)$ were to denote the double derivative of $(\mathcal{C}, b)$, then we have a degree 0 curve $\mathcal{C}''$ with control point pp_0 as below:

$$pp_0 = 2(b_0 - 2b_1 + b_2)$$

Thus the constraint becomes:

$$2b_0 - 4b_1 + 2b_2 = 5.1$$

Note that here, the constant κ is required and evaluated.

Thus the general constraint may be written as an API as follows:

$$[\mathtt{A}, \mathtt{b}, \mathtt{Aeq}, \mathtt{beq}] = \mathtt{Constraint}(\mathcal{C}, \mathtt{k}, \mathtt{Interval}, \mathtt{Range})$$

where:

$\mathcal{C}$	is the curve on which the constraint is to be posted
`k`	is an integer denoting the order of derivative
`Interval`	is a sub-interval of the curve $\mathcal{C}$
`Range`	is a (possibly infinite) interval $[a, b]$
`A,b`	are the inequality constraints
`Aeq,beq`	are the equality constraints

The constraint to be posted is that

$$\mathcal{C}^k[\mathtt{Interval}] \subseteq [\mathtt{Range}]$$

The matrices output by the API may be directly fed to the QP in addition to any other constraints that may already exist. □

Problem 9 *How is one to construct the matrices* `A,b,Aeq,beq` *as above. In other words, how is one to write the API* `Constraint`*?*

Solution: The quantities `k` and `Interval` will decide the constraint matrix, while the quantity `Range` will decide the RHS `b` or `beq`. Clearly, if `Range` is a singleton set of the form $[\alpha, \alpha]$, then `A,b` are empty and `Aeq,beq` need to be computed. In every other case, `Aeq,beq` would be empty and `A,b` to be computed. We will only tackle the case when `Range`$= [-\infty, \alpha]$, the other situations being similar. We construct the *abstract* sequence of curves:

$$\mathcal{C} \longrightarrow \mathcal{C}^k \longrightarrow \mathcal{C}^k[\mathtt{Interval}]$$

In fact, if p are the control points of $\mathcal{C}$, q of $\mathcal{C}^k$ and r of $\mathcal{C}^k[\mathtt{Interval}]$ (all row-vectors), then it is clear that there are matrices X and Y such that:

$$q = pX \quad r = qY$$

Further, these matrices *do not* depend on the actual control points p, q, r but only on the degree, `k` and `Interval`. In fact, we will represent this dependency by the following diagram:

$$(\mathcal{C}, p) \xrightarrow{X} (\mathcal{C}^k, q) \xrightarrow{Y} (\mathcal{C}^k[\mathtt{Interval}], r)$$

We claim that once X and Y are found, the output `A,b` are easily computed. To see this, suppose for the moment that X and Y have been obtained. The constraint that

$$\mathcal{C}^k[\texttt{Interval}] \subseteq [\texttt{Range}]$$

is easily expressed in terms of the control points r. Indeed, if $r = [r_1, \ldots, r_n]$, then we have the constraint (when $\texttt{Range}= [-\infty, \alpha]$):

$$r_i \leq \alpha \text{ for all } i$$

In other words:

$$I_n r^T \leq \alpha_n$$

where I_n is the $n \times n$-identity matrix, α_n is the n-long column vector with all entries α. Now since $r = qY = pXY$, we have:

$$I_n Y^T X^T p^T \leq \alpha_n$$

Thus $\texttt{A}= Y^T X^T$ and $\texttt{b}= \alpha_n$ and the problem is solved. The precise method of computing these matrices X and Y is the subject matter of the next problem.

Remark 10 *Let $\phi : \mathbb{R}^m \to \mathbb{R}^n$ be a linear map. In other words*

$$\phi(\alpha v + \alpha' v') = \alpha\phi(v) + \alpha'\phi(v')$$

for all $v, v' \in \mathbb{R}^m$ and all $\alpha, \alpha' \in \mathbb{R}$. We know that there is a matrix X which represents *ϕ, i.e., if we treat $\mathbb{R}^m$ and $\mathbb{R}^n$ as row-vectors, then for all rows $v \in \mathbb{R}^m$, we have:*

$$\phi(v) = v \cdot X$$

This matrix X may be computed as follows. Let $e_i \in \mathbb{R}^m$ be the i-th unit vector, which is 1 in the i-th place and zero everywhere else. Let $w_i = \phi(e_i)$ and let

$$X = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}$$

be the matrix with the i-th row as the n-vector w_i. Thus X is $m \times n$. It is easy to see that $\phi(v) = v \cdot X$.

Problem 11 *How is one to compute X or Y as above?*

Solution: Let

$$(\mathcal{D}, q) = \texttt{Operation}((\mathcal{C}, p), \texttt{Param})$$

be a typical operation of curves. For example, we have

$$(\mathcal{D}, q) = \texttt{Subdivision}((\mathcal{C}, p), \texttt{Interval})$$

which subdivides the curve $\mathcal{C}$ with control polygon p to obtain curve $\mathcal{D}$ with control polygon q. Suppose further that **for a fixed choice** of `Param`, the `Operation` is a linear map. In other words, if $size(p) = m$ and $size(q) = n$, and `Param` is fixed, then we may define

$$\phi(p) = q$$

where

$$(\mathcal{D}, q) = \texttt{Operation}((\mathcal{C}, p), \texttt{Param})$$

We may now construct the matrix for `Operation` as outlined in the remark above. For each $i = 1, \ldots, m$, we construct the curves $(\mathcal{C}_i, e_i)$, where e_i is the control polygon which takes the value 1 at the i-th position, and zero everywhere else. We observe the output for these curves as input:

$$(\mathcal{D}_i, q_i) = \texttt{Operation}((\mathcal{C}_i, p_i), \texttt{Param})$$

Finally, we define X as:

$$X = \begin{bmatrix} q_1 \\ \vdots \\ q_m \end{bmatrix}$$

Thus, we have the K-version of `Operation` above as:

$$[(\mathcal{D}, q), X] = \texttt{KOperation}((\mathcal{C}, p), \texttt{Param})$$

with the invariant that X depends just on `Param` and that $p \cdot X = q$. $\Box$

Problem 12 *Does this technique work for splines?*

Solution: Indeed it does work for non-rational B-splines. The only facts used were tha typical operations on curves are linear in their control points, and that there is the **variation-diminishing** relationship between the control polygon and the curve itself. Both these properties hold for non-rational B-splines. $\Box$

Remark 13 *This remark is about linear operations on matrices. Let $X = (x_{ij})$ be an $m \times n$-matrix, i.e., with m rows and n columns. Thus, the spaces of all $m \times n$-matrices is an mn-dimensional space isomorphic to $\mathbb{R}^{mn}$. We call this space as $\mathbb{M}_{m,n}$. A basis for this space is the collection*

$$\mathcal{E}_{m.n} = \{E_{ij}^{mn} | i = 1, \ldots, m,\ j = 1, \ldots, n\}$$

where E_{ij}^{mn} is an $m \times n$-matrix with $E_{ij}^{mn}(i,j) = 1$ and zero for every other entry. There are many isomorphisms from $\mathbb{M}_{m,n}$ to $\mathbb{R}^{mn}$. A simple isomorphism is the map ϕ_r which writes a matrix rowwise.Thus if $A = (a_{ij}) = \sum a_{ij} E_{ij}^{mn}$ is the matrix, then

$$\phi_r(A) = [a_{11}, \ldots, a_{1n}, a_{21}, \ldots, a_{2n}, \ldots, a_{m1}, \ldots, a_{mn}]$$

This has its obvious inverse ϕ_r^{-1}.

Let ϕ be a linear map from $\mathbb{M}_{m,n}$ to $\mathbb{M}_{a,b}$. In other words, for every matrix $A \in \mathbb{M}_{m,n}$, $\phi(A)$ is an $a \times b$-matrix. If ϕ is linear, then ϕ can be represented by a $mn \times ab$-matrix. This may be seen by the following composition:

$$\mathbb{R}^{mn} \xrightarrow{\phi_r^{-1}} M_{m,n} \xrightarrow{\phi} M_{a,b} \xrightarrow{\phi_r} \mathbb{R}^{ab}$$

Thus a general linear map from $M_{m,n}$ and $M_{a,b}$ is represented by an $mn \times ab$-matrix. However, there are some special linear mapswhich have a less cumbersome representation. Let A be a **fixed** *$n \times p$-matrix. Consider the map:*

$$\mu_A : \mathbb{M}_{m,n} \rightarrow \mathbb{M}_{m,p}$$

the **right-multiplication** *map, with $\mu_A(X) = XA$. Clearly*

$$\mu_A(X_1 + X_2) = \mu_A(X_1) + \mu_A(X_2) = X_1X + X_2A$$

Other linearity properties may also be easily verified. Thus μ_A may be represented by a $mn \times mp$ matrix. We explain this by an example: Let $m = 3, n = 2$ and $p = 3$, and let

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

The operation of μ_A on a general 3×2-matrix is given by:

$$\mu_A\left(\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \end{bmatrix}\right) = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} x_{11} \cdot 1 + x_{12} \cdot 3 & x_{11} \cdot 2 + x_{12} \cdot 4 \\ x_{21} \cdot 1 + x_{22} \cdot 3 & x_{21} \cdot 2 + x_{22} \cdot 4 \\ x_{31} \cdot 1 + x_{32} \cdot 3 & x_{31} \cdot 2 + x_{32} \cdot 4 \end{bmatrix}$$

Thus, in terms of the row-vector representation $\phi_r(X)$, we have:

$$\mu_A([x_{11}x_{12}x_{21}x_{22}x_{31}x_{32}]) = [x_{11}x_{12}x_{21}x_{22}x_{31}x_{32}] \cdot \begin{bmatrix} 1 & 2 & 0 & 0 & 0 & 0 \\ 3 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 3 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 3 & 4 \end{bmatrix}$$

Thus we see that μ_A has avery special representation as a general linear transformation. A similar phenomenon may be observed for the column-wise representation of a matrix and **left multiplication**. *Thus, in general, these special linear transformation from $\mathbb{M}_{m,n}$ to $\mathbb{M}_{a,b}$ is given by two matrices L and R where*

$$\phi(X) = L \cdot X \cdot R$$

and where L is $a \times m$ and R is $n \times b$. We denote such linear maps as $\lambda_{L,R}$. Note that since $L'(LXR)R' = (L'L)X(RR')$, we have:

$$\lambda_{L',R'}(\lambda_{L,R}(X)) = \lambda_{L'L,RR'}(X)$$

This defines the composition of two such operators.

Next, for a $n \times n$ matrix $X = (x_{ij})$, define the **trace** *of X as:*

$$trace(X) = \sum_i x_{ii}$$

Thus $trace(X)$ is the sum of its diagonal entries. Clearly:

$$trace(A) = trace(A^T)$$

It is easy to verify that when A is $m \times n$ and B is $n \times m$, we have:

$$trace(AB) = trace(BA) = trace(B^T A^T) = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} b_{ji}$$

Thus if both X and A are $m \times n$, then

$$trace(XA^T) = trace(X^T A) = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} x_{ij}$$

This serves as a useful way of storing a linear combination of the elements of a general matrix. For example,

$$\ell(X) = x_{11} + x_{21} + 3x_{22} + 4x_{32} = trace\left(\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 1 & 3 \\ 0 & 4 \end{bmatrix}^T \right)$$

Thus the linear combination $\ell(X)$ may be stored as the matrix A below:

$$A = \begin{bmatrix} 1 & 0 \\ 1 & 3 \\ 0 & 4 \end{bmatrix} \quad \ell(X) = trace(XA^T)$$

A typical situation is when X is a $m \times n$-matrix and L, R are fixed matrices. We wish to post a constraint that the ij-th entry y_{ij} in $Y = LXR$, say $y_{ij} \geq 0$. This may be seen as:

$$y_{ij} = trace(Y(E_{ij}^{ab})^T) = trace(LXR(E_{ij}^{ab})^T) = trace(XR(E_{ij}^{ab})^T L) = trace(X(L^T E_{ij}^{ab} R^T)^T)$$

Thus $y_{ij} \geq 0$ translates to $trace(XA^T) \geq 0$ where $A = L^T E_{ij}^{ab} R^T$. Note that E_{ij}^{ab} is $a \times b$, L is $a \times m$ and R is $n \times b$ and thus A is indeed $m \times n$.

Problem 14 *What constraints on tensor-product surfaces may be represented in an analogous manner.*

Solution: Recall that the general tensor-product surface $\mathcal{S}$ is given by the following data:

$$\mathcal{S} = (\texttt{knotu}, \texttt{knotv}, C)$$

where `knotu`, `knotv` are the knot vectors in the u and v-directions, and C is the control point matrix. We assume that C is $m \times n$, where m is determined by `knotu` and n by `knotv`.

The general constraint on surfaces is as follows:

$$[\texttt{A}, \texttt{b}, \texttt{Aeq}, \texttt{beq}] = \texttt{Constraint}(\mathcal{S}, \texttt{ku}, \texttt{kv}, \texttt{Box}, \texttt{Range})$$

where:

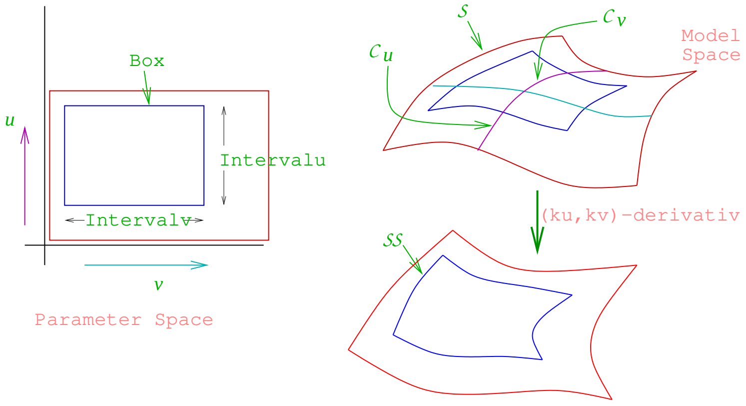

Figure 2: The Surface Constraint

$\mathcal{S}$	is the curve on which the constraint is to be posted
`ku`	is an integer denoting the order of derivative in the u-direction
`kv`	is an integer denoting the order of derivative in the v-direction
`Box`	is `Intervalu`$\times$`Intervalv`, a sub-domain of the surface $\mathcal{S}$
`Range`	is a (possibly infinite) interval $[a, b]$

The structure of `A,b,Aeq,beq` will be described later.

This expresses the constraint that:

$$\mathcal{S}^{\texttt{ku},\texttt{kv}}(\texttt{Box}) \subseteq \texttt{Range}$$

We proceed as before by computing the surface

$$\mathcal{SS} = \mathcal{S}^{\texttt{ku},\texttt{kv}}(\texttt{Box})$$

i.e., the `ku`-th u-derivative and the `kv`-th v-derivative and finally, the restriction to `Box`. Thus if CC is the control-point matrix for $\mathcal{SS}$, then CC is obtained as a linear operation on the matrix C, the control-points of $\mathcal{S}$. We now compute the transformation from C to CC, which depends on `ku,kv` and `Box`. Note that `Box=Intervalu` $\times$ `Intervalv`.

Note that since $\mathcal{S}$ is a tensor product surface, the above constraint factors into a **left** and a **right** operation. Let

$$\begin{aligned} \mathcal{C}_u = &\quad (\mathcal{C}_u, \texttt{knotu}, C_u) \\ \mathcal{C}_v = &\quad (\mathcal{C}_v, \texttt{knotv}, C_v) \end{aligned}$$

where C_u is an m-rowvector and C_v is an n-rowvector, denoting the u-direction (resp. v-direction) curve. $\mathcal{C}_u$ and $\mathcal{C}_v$ represent **generic** u and v curves on the surface $\mathcal{S}$.

Let

$$\begin{aligned} \mathcal{CC}_u = &\quad \mathcal{C}_u^{\texttt{ku}}(\texttt{Intervalu}) \\ \mathcal{CC}_v = &\quad \mathcal{C}_v^{\texttt{kv}}(\texttt{Intervalv}) \end{aligned}$$

If CC_u and CC_v are the control points (row vectors) of $\mathcal{CC}_u$ and $\mathcal{CC}_v$ resp., then there are matrices T_u and T_v such that:

$$CC_u = C_u T_u \quad CC_v = C_v T_v$$

If that is so, then

$$CC = T_u^T \cdot C \cdot T_v$$

Thus the control-points for $\mathcal{SS}$ are obtained from those of $\mathcal{S}$ by a **left-right** transformation on the control-point matrix C of $\mathcal{S}$. Finally the constraint that $\mathcal{S}^{\mathtt{ku,kv}}(\mathtt{Box}) \subseteq \mathtt{Range}$ translates into a constraint on each entry of $CC = T_u^T C T_v$, which was handled in Remark 13. Note now that `A,Aeq` will be a list of $m \times n$-matrices, while `b,beq` continue to be vectors. A typical inequality is to be obtained as:

$$trace(C\mathtt{A}\{i\}^T) \leq \mathtt{b}(i)$$

`Aeq,beq` are similarly interpreted. $\Box$