Matrix Recipes

Javier R. Movellan

December 28, 2006

1 Definitions

Let x, y be matrices of order $m \times n$ and $o \times p$ respectively, i.e..

$$x = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix} \quad y = \begin{pmatrix} y_{11} & \cdots & y_{1p} \\ \vdots & \ddots & \vdots \\ y_{o1} & \cdots & y_{op} \end{pmatrix} \tag{1}$$

We refer to a matrix of order 1×1 as a scalar, and a matrix of order $m \times 1$ where $m > 1$, as a vector. We let f be a generic function of the form $y = f(x)$. If x is a vector and y is a scalar we say that f is a scalar function of a vector. If x is a matrix and y a vector, we say that f is a vector function of a matrix, and so on.

General Definition: Derivative of a matrix function of a matrix.

$$\frac{\partial y}{\partial x} \stackrel{\text{def}}{=} \begin{pmatrix} \frac{\partial y}{\partial x_{11}} & \cdots & \frac{\partial y}{\partial x_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial y}{\partial x_{m1}} & \cdots & \frac{\partial y}{\partial x_{mn}} \end{pmatrix} \tag{2}$$

where

$$\frac{\partial y}{\partial x_{ij}} \stackrel{\text{def}}{=} \begin{pmatrix} \frac{\partial y_{11}}{\partial x_{ij}} & \cdots & \frac{\partial y_{1q}}{\partial x_{ij}} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_{p1}}{\partial x_{ij}} & \cdots & \frac{\partial y_{pq}}{\partial x_{ij}} \end{pmatrix} \tag{3}$$

Thus the derivative of a $p \times q$ matrix y with respect to a $m \times n$ matrix x is a $(mp) \times (nq)$ matrix of the following form

$$\frac{\partial y}{\partial x} \stackrel{\text{def}}{=} \begin{pmatrix} \frac{\partial y_{11}}{\partial x_{11}} & \cdots & \frac{\partial y_{11}}{\partial x_{1n}} & \cdots & \frac{\partial y_{1q}}{\partial x_{11}} & \cdots & \frac{\partial y_{1q}}{\partial x_{1n}} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ \frac{\partial y_{11}}{\partial x_{m1}} & \cdots & \frac{\partial y_{11}}{\partial x_{mn}} & \cdots & \frac{\partial y_{1q}}{\partial x_{m1}} & \cdots & \frac{\partial y_{1q}}{\partial x_{mn}} \\ & \vdots & & \cdots & & \vdots & \\ \frac{\partial y_{p1}}{\partial x_{11}} & \cdots & \frac{\partial y_{p1}}{\partial x_{1n}} & \cdots & \frac{\partial y_{pq}}{\partial x_{11}} & \cdots & \frac{\partial y_{pq}}{\partial x_{1n}} \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ \frac{\partial y_{p1}}{\partial x_{m1}} & \cdots & \frac{\partial y_{p1}}{\partial x_{mn}} & \cdots & \frac{\partial y_{pq}}{\partial x_{m1}} & \cdots & \frac{\partial y_{pq}}{\partial x_{mn}} \end{pmatrix} \tag{4}$$

From this definitions we can derive a variety of important special cases.

Derivative of a scalar function of a vector. We think of a vector as a matrix with a single column and a scalar as a matrix with a single cell. Applying

the general definition we get that

$$\frac{\partial y}{\partial x} \stackrel{\text{def}}{=} \begin{pmatrix} \frac{\partial y}{\partial x_1} \\ \vdots \\ \frac{\partial y}{\partial x_m} \end{pmatrix} \tag{5}$$

Gradient of a scalar function of a vector. We represent gradients using the ∇ sign. For the case of gradients of scalar functions of a vector, we define the gradient to equal the derivative (note this will not be the case for vector functions of vectors)

$$\nabla_x y \stackrel{\text{def}}{=} \frac{\partial y}{\partial x} \tag{6}$$

Derivative of a vector function of a vector. If we think of y and x as $p \times 1$ and $m \times 1$ matrices, then the derivative would be a vector with mp dimensions.

$$\frac{\partial y}{\partial x} \stackrel{\text{def}}{=} (\frac{\partial y_{11}}{\partial x_{11}} \cdots \frac{\partial y_{p1}}{\partial x_{11}} \cdots \frac{\partial y_{11}}{\partial x_{1m}} \cdots \frac{\partial y_{p1}}{\partial x_{1m}})^T \tag{7}$$

Jacobian of a vector function of a vector. A more useful representation can be obtained by working with the derivative of y with respect to the transpose of x. This results on an $p \times m$ matrix which is known as the Jacobian of y with respect to x

$$J_x y \stackrel{\text{def}}{=} \frac{\partial y}{\partial x^T} = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_p}{\partial x_1} & \cdots & \frac{\partial y_p}{\partial x_m} \end{pmatrix} \tag{8}$$

The Jacobian is very useful for linearizing vector functions of vectors

$$f(x) \approx f(x_0) + J_x f(x - x_0) \tag{9}$$

Gradient of a vector function of a vector. Note if y is a scalar then $J_x y = \frac{\partial y}{\partial x^T} = (\nabla_x y)^T$ i.e., the Jacobian of a scalar function is the transpose of the gradient. With this in mind we will define the gradient of a vector function of a vector as the transpose of the Jacobian

$$\nabla_x y \stackrel{\text{def}}{=} (J_x y)^T = \left(\frac{\partial y}{\partial x^T} \right)^T = \frac{\partial y^T}{\partial x} \tag{10}$$

Hessian of a scalar function of a vector. Let $y = f(x)$, $y \in \Re$, $x \in \Re^n$. The matrix

$$H_x y \stackrel{\text{def}}{=} \nabla_x^2 y \stackrel{\text{def}}{=} \nabla_x \nabla_x y \stackrel{\text{def}}{=} \frac{\partial}{\partial x} \left(\frac{\partial y}{\partial x} \right)^T = \begin{pmatrix} \frac{\partial^2 y}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 y}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 y}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2}{\partial x_n \partial x_n} \end{pmatrix} \tag{11}$$

is called the Hessian matrix of y with respect to x.

2 Summary of definitions

For x, y vectors

$$\text{Gradient:} \quad \nabla_x y \stackrel{\text{def}}{=} \frac{\partial y^T}{\partial x} \tag{12}$$

$$\text{Jacobian:} \quad J_x y \stackrel{\text{def}}{=} (\nabla_x y)^T \stackrel{\text{def}}{=} \frac{\partial y}{\partial x^T} \tag{13}$$

$$\text{Hessian:} \quad H_x y \stackrel{\text{def}}{=} \nabla_x^2 y \stackrel{\text{def}}{=} \nabla_x \nabla_x y \tag{14}$$

3 Chain Rules

1. **Vector functions of vectors:** If $z = h(y)$, is a vector function of a vector, and $y = f(x)$ is a vector function of a vector then

$$J_x z = (J_y z)(J_x y) \tag{15}$$

or equivalently

$$\nabla_x z = (\nabla_x y)(\nabla_y z) \tag{16}$$

Example 1: Consider the quadratic function

$$z = y^T a y \tag{17}$$

where y is an n dimensional vector a is an $n \times n$ matrix. It is easy to show that

$$\frac{\partial x^T a x}{\partial x_k} = \sum_j x_j (a_{kj} + a_{jk}) \tag{18}$$

thus

$$J_x x^T a x = x^T (a + a^T) \tag{19}$$

$$\nabla_x x^T a x = (a + a^T) x \tag{20}$$

Moreover if

$$y = bx \tag{21}$$

where x is an m dimensional vector and b is an $n \times m$ matrix then

$$J_x y = b \tag{22}$$

$$\nabla_x y = b^T \tag{23}$$

Thus

$$J_x(bx-c)^T a(bx-c) = \left(J_{bx-c}(bx-c)^T a(bx-c)\right)\ \left(J_x(bx-c)\right) \quad (24)$$
$$= (x-c)^T(a+a^T)b \quad (25)$$

and

$$\nabla_x(x-\mu)^T a(x-\mu) = b^T(a+a^T)(x-c)^T \quad (26)$$

Example 2: Let x, y vectors

$$\nabla_x e^{x^T y}\, y = \nabla_x e^{x^T y}\, \nabla_{e^{x^T y}} e^{x^T y}\, y \quad (27)$$

where

$$\nabla_x e^{x^T y} = \nabla_x x^T y\, \nabla_{x^T y}\, e^{x^T y} = y e^{x^T y} \quad (28)$$
$$\nabla_{e^{x^T y}}\, e^{x^T y} y = \nabla_{e^{x^T y}} y e^{x^T y} = y^T \quad (29)$$

Thus

$$\nabla_x\, e^{x^T y}\, y = y\, e^{x^T y}\, y^T \quad (30)$$

2. **Scalar functions of matrices:** If $z = h(\mathbf{y})$ is a scalar function of a matrix and $\mathbf{y} = f(x)$ is a matrix function of a scalar then

$$\frac{\partial z}{\partial x}\text{trace}\left[\frac{\partial z}{\partial \mathbf{y}}\left(\frac{\partial \mathbf{y}}{\partial x}\right)^T\right] = \text{trace}\left[\left(\frac{\partial z}{\partial \mathbf{y}}\right)^T \frac{\partial \mathbf{y}}{\partial x}\right] \quad (31)$$

4 Useful Formulae (I need to check whether notation is consistent with other parts of the document)

Let a, b, c, d matrices, x, y, z vectors.

4.1 Linear and Quadratic Functions

$$\frac{d}{dx}x^T c = c \tag{32}$$

$$\nabla_x\, cx = \frac{d}{dx}(cx)^T = c^T \tag{33}$$

$$J_x\, cx = (\nabla_x\, cx)^T = c \tag{34}$$

$$\frac{d}{dx}(ax+b)^T c(dx+e) = a^T c(dx+e) + d^T c^T (ax+b), \tag{35}$$

$$\frac{d}{dx}(ax+b)^T c(ax+b) = a^T(c+c^T)(ax+b) \tag{36}$$

$$\frac{d}{da}(ax+b)^T c(ax+b) = (c+c^T)(ax+b)x^T, \tag{37}$$

$$\frac{d}{da}x^T a^T bay = b^T axy^T + bayx^T \tag{38}$$

$$\frac{d}{da}x^T ay = xy^T \tag{39}$$

$$\frac{d}{da}x' e^{ta} y = txe^{ta} y' \tag{40}$$

4.2 Traces

$$\frac{d}{da}\text{trace}\left[a\right] = I \tag{41}$$

$$\frac{d}{db}\text{trace}\left[abc\right] = \frac{d}{db}\text{trace}\left[c^T b^T a^T\right] = a^T c^T \tag{42}$$

$$\frac{d}{db}\text{trace}\left[ab^n\right] = \left(\sum_{i=0}^{n-1} b^i a b^{n-i-1}\right)^T \tag{43}$$

$$\frac{d}{de}\text{trace}\left[adbe^T c\right] = a^T c^T d b^T + caeb \tag{44}$$

$$\frac{d}{db}\text{trace}\left[a^T ba\right] = (b + b^T)a \tag{45}$$

$$\frac{d}{da}\text{trace}\left[a^{-1}b\right] = -a^{-1}b^T a^{-1} \tag{46}$$

4.3 Determinants

$$\frac{d}{da}\det\left[a\right] = \det\left[a\right] a^{-T} \tag{47}$$

4.4 Kronecker and Vecs

$$\frac{\partial \text{vec}[abc]}{\partial \text{vec}[b]} = c \otimes a^T \tag{48}$$

5 Matrix Differential Equations

$$\frac{d(a_t + b_t)}{dt} = \frac{da_t}{dt} + \frac{db_t}{dt} \tag{49}$$

$$\frac{d(a_t b_t)}{dt} = \frac{da_t}{dt} b_t + a_t \frac{db_t}{dt} \tag{50}$$

$$\frac{d(a_t^2)}{dt} = \frac{da_t}{dt} a_t + a_t \frac{da_t}{dt} \tag{51}$$

$$\frac{d(a_t^2)}{dt} = \frac{da_t}{dt} a_t + a_t \frac{da_t}{dt} \tag{52}$$

$$\frac{da_t a_t^{-1}}{dt} = 0 = \frac{da_t}{dt} a_t^{-1} + a_t^{-1} \frac{da_t}{dt} \tag{53}$$

$$\frac{a_t^{-1}}{dt} = - - a_t^{-1} \frac{da_t}{dt} a^{-1} \tag{54}$$

$$\tag{55}$$

It is only possible to expect $de^{at}dt = a_t e^{at}$ under conditions of full commutativity Types of linear matrix equations

$$\frac{da_t}{dt} = ba_t c \tag{56}$$

$$\tag{57}$$

with special cases when b or c are the identity matrix. The following tricks allow moving the coefficients to the left, or right

$$\frac{da_t}{dt} = a_t c \tag{58}$$

then

$$\frac{da_t^T}{dt} = c^T a_t^T \tag{59}$$

$$\tag{60}$$

and if

$$\frac{da_t}{dt} = ba_t \tag{61}$$

then

$$\frac{da_t^{-1}}{dt} = -a^{-1}\frac{da_t}{dt}a^{-1} = -a^{-1}ba_t a_t^{-1} = -a_t^{-1}b \tag{62}$$

$$\tag{63}$$

It can be shown that if a_0 is non-singular, then the solution a_t is non-singular for all t.

6 Determinants

- $|I + xy'| = 1 + x'y,\ x, y \in \Re^n$
- $|e^a| = e^{\text{trace}(a)}$

7 Trace

- $\text{trace}(a + b) = \text{trace}(a) + \text{trace}(b)$
- $\text{trace}(a) = \text{trace}(a^T)$
- $\text{trace}(abc) = \text{trace}(cab) = \text{trace}(bca)$
- $\text{trace}(xy^T a) = y^T ax,\ x \in \Re^m, y \in \Re^n$
- $\text{trace}(a^T rb^T) = a^T rb$ *confirm r does not need rotation*
- If a is $m \times n$ and b is $n \times m$ then $\text{trace}(ab) = \text{trace}(ba) = \text{trace}(a^T b^T)$

8 Matrix Exponentials

- $|e^a| = e^{\text{trace}}(a)$
- if $a = p\Lambda p^{-1}$ then $e^a = pe^{\Lambda}p^{-1}$
- $\frac{de^{ta}}{dt} = ae^{ta}$
- $\frac{d}{da}x'e^{ta}y = te^{ta}xy'$

8.1 Proofs:

$$\begin{aligned}\frac{\partial x'e^{ta}}{\partial a_{ij}} &= \frac{d}{d\delta}x'e^{t(a+\delta 1_i 1_j')}y\Big|_{\delta=0} = x'e^{ta}\left(\frac{d}{d\delta}e^{t\delta 1_i 1_j'}\right)y\Big|_{\delta=0} \\ &= x'e^{ta}t1_i 1_j' e^{\delta 1_i 1_j'}y\Big|_{\delta=0} = t(e^{ta}x)_i y_j \end{aligned} \tag{64}$$

Thus

$$\frac{\partial x'e^{ta}}{\partial a} = te^{ta}xy' \tag{65}$$

9 Matrix Logarithms

Let a a real or complex square matrix of ordeer n with positive eigenvalues. Then there is a unique matrix b such that (1) $a = e^b$ and (2) the imagignary part of the eigenvalues is in $[-\pi, \pi]$. We call b the principle logarithm of a.

- if $a = p\Lambda p^{-1}$ then $log(a) = plog(\Lambda)p^{-1}$

10 Kronecker and Vec

Provide a way to deal with derivatives of matrix functions without having to use cubix or quartix (i.e., matrices with 3 or 4 dimensions). Instead of working with matrix functions we work with vectorized versions of matrix functions. This gives rise to the Kronecker product, or tensor product.

Definition: Kronecker product

$$a \otimes b = \begin{pmatrix} a_{11}b & \cdots & a_{1n}b \\ \vdots & \ddots & \vdots \\ a_{m1}b & \cdots & a_{mn}b \end{pmatrix} \tag{66}$$

Definition: Vec operator

$$\text{vec}[a] = \begin{pmatrix} a_{11} \\ \vdots \\ a_{m1} \\ a_{12} \\ \vdots \\ a_{m2} \\ \vdots \\ a_{1n} \\ \vdots \\ a_{mn} \end{pmatrix} \tag{67}$$

10.1 Properties

1. $$a \otimes b \otimes c = (a \otimes b) \otimes c = a \otimes (b \otimes c) \tag{68}$$
   provided the dimensions of the matrices allows for all the expressions to exist.

2. $$(a + b) \otimes (c + d) = a \otimes c + a \otimes d + b \otimes c + b \otimes d \tag{69}$$

3. $$(a \otimes b)(c \otimes d) = (ac) \otimes (bd) \tag{70}$$

4. $$(a \otimes b)(b \otimes d) = (ac) \otimes (bd) \tag{71}$$

5. $$(a \otimes b)^T = a^T \otimes b^T \tag{72}$$

6.
$$(a \otimes b)^{-1} = a^{-1} \otimes b^{-1} \tag{73}$$

7.
$$\text{vec}\left[ab^T\right] = b \otimes a \tag{74}$$

8.
$$\text{vec}\left[abc\right] = (c^T \otimes a)\,\text{vec}\left[b\right] \tag{75}$$

9. If $\{\lambda_i, u_i\}$ are eigenvalues/eigenvectors of a and $\{\delta_i, v_i\}$ are eigenvalues/eigenvectors of b then $\{\lambda_i\delta_j, u_i \otimes v_j\}$ are eigenvalues/eigenvectors ofr $a \otimes b$

10.
$$\det[a \otimes b] = \det[a]^m \det[b]^n \tag{76}$$
where a, b are of order $n \times n$ and $m \times m$ respectively.

11.
$$\text{trace}(a \otimes b) = \text{trace}[a]\,\text{trace}[b] \tag{77}$$

12.
$$\nabla^2_{\text{vec}x}(axbx^T) = \nabla^2_{\text{vec}x}\text{vec}(x^T)(b^T \otimes a)\text{vec}x = b \otimes a' + b' \otimes a \text{trace}(a \otimes b) = \text{trace}[a]\,\text{trace}[b] \tag{78}$$
where a, b, x are matrices.

11 Optimization of Quadratic Functions

This is arguably the most useful optimization problem in applied mathematics. Its solution is behind a large variety of useful algorithms including Multivariate Linear Regression, the Kalman Filter, Linear Quadratic Controllers, etc. Let

$$\rho(x) = E(bx - C)^T a(bx - C) + x^T d\, x \tag{79}$$

where a and d are symmetric positive definite matrices, b is a matrix, x a vector and C a random vector. Taking the Jacobian with respect to x and applying the chain rule we have

$$J_x\rho = J_{bx-c}(bx - c)^T a(bx - c)\, J_x(bx - c) + J_x x^T d\, x \tag{80}$$
$$= 2(bx - c)^T ab + 2x^T d \tag{81}$$
$$\nabla_x\rho = (J_x)^T = 2b^T a(bx - c) + 2d\, x \tag{82}$$

Setting the gradient to zero we get

$$(b^T ab + d)x = b^T ac \tag{83}$$

This is commonly known as the *Normal Equation*. Thus the value $\hat{x}$ that minimizes ρ is

$$\hat{x} = hc \tag{84}$$

where

$$h = (b^T ab + d)^{-1} b^T a \tag{85}$$

Moreover

$$\rho(\hat{x}) = (bhc - c)^T a(bhc - c) + c^T h^T dhc \tag{86}$$

$$= c^T h^T b^T abhc - 2c^T h^T b^T ac + c^T ac + c^T h^T dhc \tag{87}$$

Now note

$$c^T h^T b^T abhc + c^T h^T dhc = c^T h^T (b^T ab + d)hc \tag{88}$$

$$= c^T a^T b(b^T ab + d)^{-1}(b^T ab + d)(b^T ab + d)^{-1} b^T ac \tag{89}$$

$$= c^T a^T b(b^T ab + d)^{-1} b^T ac \tag{90}$$

$$= c^T h^T b^T ac \tag{91}$$

Thus

$$\rho(\hat{x}) = c^T ac - c^T h^T b^T ac = c^T kc \tag{92}$$

where

$$k = a - h^T b^T a = a - a^T b(b^T ab + d)^{-1} b^T a \tag{93}$$

This is known as the *Riccati Equation* which is found in a variety of stochastic filtering and control problems.

Example Application: Ridge Regression Let $y \in \Re^n$ represents a set of observations on a variable we want to predict, b is an $n \times p$ matrix, where each row is a set of observations on p variables used to predict c, and $x \in \Re^p$ are the weights given to each variable to predict y. A useful measure of error is

$$\rho(x) = (bx - y)^T (bx - y) + \lambda x^T x \tag{94}$$

where $\lambda \geq 0$ is a constant that penalizes for the use of large values of x. Thus the solution to this problem is

$$\hat{x} = (b^T b + \lambda I_p)^{-1} b^T by \tag{95}$$

where I_p is the $p \times p$ identity matrix.

12 Optimization Methods

12.1 Newton-Raphson Method

Let $y = f(x)$, for $y \in \Re$, $x \in \Re^n$. The Newton-Raphson algorithm is an iterative method for optimizing y. We start the process at an arbitrary point $x_0 \in \Re^n$.

Let $x_t \in \Re^n$ represent the state of the algorithm at iteration t. We approximate the function f using the linear and quadratic terms of the Taylor expansion of f around x_t.

$$\hat{f}_t(x) = f(x_t) + \nabla_x f(x_t)(x - x_t)^T + \frac{1}{2}(x - x_t)^T \left(\nabla_x \nabla_x f(x_t)\right)(x - x_t) \quad (96)$$

and then we then find the extremum of $\hat{f}_t$ with respect to x and move directly to that extremum. To do so note that

$$\nabla_x \hat{f}_t(x) = \nabla_x f(x_t) + \left(\nabla_x \nabla_x f(x_t)\right)(x - x_t) \quad (97)$$

We let $x(t+1)$ be the value of x for which $\nabla_x \hat{f}_t(x) = 0$

$$x_{t+1} = x_t + \left(\nabla_x \nabla_x f(x_t)\right)^{-1} \nabla_x f(x_t) \quad (98)$$

It is useful to compare the Newton-Raphson method with the standard method of gradient ascent. The gradient ascent iteration is defined as follows

$$x_{t+1} = x_t + \epsilon I_n \nabla_x f(x_t) \quad (99)$$

where ϵ is a small positive constant. Thus gradient descent can be seen as a Newton-Raphson method in which the Hessian matrix is approximated by $\frac{1}{\epsilon} I_n$.

12.2 Gauss-Newton Method

Let $f(x) = \sum_{i=1}^n r_i(x)^2$ for $r_i : \Re^n \to \Re$. We start the process with an arbitrary point $x_0 \in \Re^n$. Let $x_t \in \Re^n$ represent the state of the algorithm at iteration t. We approximate the functions r_i using the linear term of their Taylor expansion around x_t.

$$\hat{r}_i(x_t) = r_i(x_t) + (\nabla_x r_i(x_t))^T (x - x_t) \quad (100)$$

$$\hat{f}_t(x) = \sum_{i=1}^n (\hat{r}_i(x_t))^2 = \sum_{i=1}^n (r_i(x_t) - (\nabla_x r_i(x_t))^T x_t + (\nabla_x r_i(x_t))^T x)^2 \quad (101)$$

$$(102)$$

Minimizing $\hat{f}_t(x)$ is a linear least squares problem of well known solution. If we let $y_i = (\nabla_x r_i(x_t))^T x_t - r_i(x_t)$ and $u_i = \nabla_x r_i(x_t)$ then

$$x_{t+1} = (\sum_{i=1}^n u_i u_i^T)^{-1} (\sum_{i=1}^n u_i y_i) \quad (103)$$

Note this is equivalent to Newton-Raphson with the Hessian being approximated by $(\sum_{i=1}^n u_i u_i^T)^{-1}$.

History

1. The first version of this document was written by Javier R. Movellan, on May 2004, based on an Appendix from the Multivariate Logistic Regression Primer at the Kolmogorov Project. The original document was 7 pages long.