Data approximation using Lotka-Volterra models and a software minimization function

M. FEČKAN AND J. PAČUTA

Abstract

In recent years, a lot of effort has been put into finding suitable mathematical models that fit historical data set. Such models often include coefficients and the accuracy of data approximation depends on them. So the goal is to choose the unknown coefficients to achieve the best possible approximation of data by the corresponding solution of the model. One of the standard methods for coefficient estimation is the least square method. This can provide us data approximation but it can also serve as a starting method for further minimizations such as Matlab function fminsearch.

Mathematics Subject Classification 2010: 34C60, 65C20, 91B64.
General Terms: Algorithms, Experimentation
Keywords: Systems modelling, linear regression, error minimization, integral method

1. INTRODUCTION

The general goal of modelling is to find a simple model that fits a historical data set of some property and indicates how could this property evolve in future. In this paper using given data set, we estimate parameter values in a classic Lotka-Volterra model. These parameter values describe various interactions between populations, for example, growth, competition, mutualism or predation. The model is fully determined by its parameters so that they need to be determined to apply the model in practice. Therefore the problem is to compute the parameter values of a model as accurately as possible in order to fit the raw data as closely as possible.

To achieve this goal, various methods have been discussed, for example Shatalov et al. in [1], and Fedatov and Shatalov in [2] started an approach of using direct integration of equations in Lotka-Volterra system and apply quadrature rules to obtain the unknown parameters. Michalakelis et al. in [3] use artificial intelligence methods to solve a non-linear system. Our work is connected with results of Kloppers and Greef in [4] who compare the integral method, the log integral method and advanced

M.F. was supported by the Slovak Research and Development Agency under the contract No. APVV-18-0308 and by the Slovak Grant Agency VEGA No. 2/0153/16.
J.P. was supported by the Slovak Research and Development Agency under the contract No. APVV-18-0308 and by the Slovak Grant Agency VEGA No. 1/0347/18.

10.2478/jamsi-2019-0005

method from [3].

In this work, we use Matlab to approximate historical data by solutions of a Lotka-Volterra model

$$\frac{dx_i}{dt} = x_i(p_{i1} + \sum_{j=1}^{m} p_{i,j+1}x_j), \quad i = 1,\dots,m. \tag{1}$$

We use two historical data sets to estimate parameter values in the system (1). First, we compute coefficients p_{ij}, $i = 1,\dots,m$, $j = 1,\dots,m+1$ in the system (1) by using numerical integration along with least squares method. Unlikely in [4], we use these p_{ij} as an initial point for fminsearch - Matlab function for finding minima of functions. We use this Matlab function to find the coefficients P_{ij} such that the corresponding numerical solution of (1) is discreetly closest to historical data in terms of least squares. Similar approach - minimization of some error functional - was previously used in e.g. [3] where a genetic algorithm was applied. Finally, we compare this optimal numerical solution of (1) with P_{ij} with the solution of (1) with initial coefficients p_{ij} (i.e. without using fminsearch).

2. DESCRIPTION OF METHODS

Assume $m = 3$ in the Lotka Volterra model (1). The system is then represented by

$$\begin{aligned} \tfrac{dx}{dt} &= x(p_{11} + p_{12}x + p_{13}y + p_{14}z), \\ \tfrac{dy}{dt} &= y(p_{21} + p_{22}x + p_{23}y + p_{24}z), \\ \tfrac{dy}{dt} &= z(p_{31} + p_{32}x + p_{33}y + p_{34}z). \end{aligned} \tag{2}$$

The unknown functions x, y, z are three species that compete for available resources in a system.

Denote $D = D_{ij}$, $i = 1,\dots,n$, $j = 1,2,3$ the historical data set (Figure 1). In our case, we consider prices of three crypto currencies during $n-$day period.

Now, we will in short explain the integral method for obtaining unknown initial coefficients p_{ij} using historical data D; see [4] for details. We choose the first differential equation in (2) and integrate both sides with respect to t over the interval $[k,k+1]$ for $k = 1,2,\dots,n-1$. We approximate every integral by applying the Trapezium rule. For example, the integral $\int_k^{k+1} x(t)\ \mathrm{d}t$ is approximated by $(x(k+1)-x(k))/2$ what is equal to $(D_{k1}+D_{k+1,1})/2$. The system (2) then turns into

date	eth	dash	xmr
15-Mar-18	614.84	425.95	215.84
16-Mar-18	611.78	418.54	213.21
17-Mar-18	601.68	426.3	215.49
18-Mar-18	551.64	388.18	197.43
19-Mar-18	546.63	392.32	210.33
20-Mar-18	556.72	412.49	217.94
21-Mar-18	559.1	429.15	224.26
22-Mar-18	562.1	434.44	218.56
23-Mar-18	539.86	411.27	212.47
24-Mar-18	542.57	432.17	215.37
25-Mar-18	522.7	420.02	207.32
26-Mar-18	524.29	410.73	211.79
27-Mar-18	489.59	390.24	196.85
28-Mar-18	450.29	354.93	187.47
29-Mar-18	448.08	350.85	199.31
30-Mar-18	385.9	328.65	176
31-Mar-18	395	310.91	172.53
01-Apr-18	397.25	306.21	178.78
02-Apr-18	379.7	292.64	175.87
03-Apr-18	387.31	312.64	177.57

Fig. 1. Price data in dollars

a system of linear equations

$$\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ \vdots & & & \\ a_{n-1,1} & a_{n-1,2} & a_{n-1,2} & a_{n-1,2} \end{bmatrix} . \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \end{bmatrix} \tag{3}$$

or $B = A.p_1$ where $B = b_k = D_{k+1,1} - D_{k1}$, $p_1 = p_{1j}$ $(j = 1,2,3,4)$, $A = a_{kj}$ and $a_{k1} = (D_{k1} + D_{k+1,1})/2$, $a_{kj} = (D_{k1}D_{kj} + D_{k+1,1}D_{k+1,j})/2$, $j = 2,3,4$. The matrix p_1 contains the unknown coefficients. If, in addition, the matrix $A^\intercal A$ is regular ($A^\intercal$ denotes the transpose of A) then p_1 is given by

$$p_1 = (A^\intercal A)^{-1} A^\intercal B.$$

3. EXAMPLES

3.1. Example 1

In this example, we use historical price data of crypto currencies Ethereum, Dash and Monero that are competitors x, y, z respectively. The data are downloaded from https://coinmetrics.io and the date range of the data is from 15th March 2018 till 3rd April 2018; see Figure 1. As can be seen in Figure 2, our data are volatile and not very

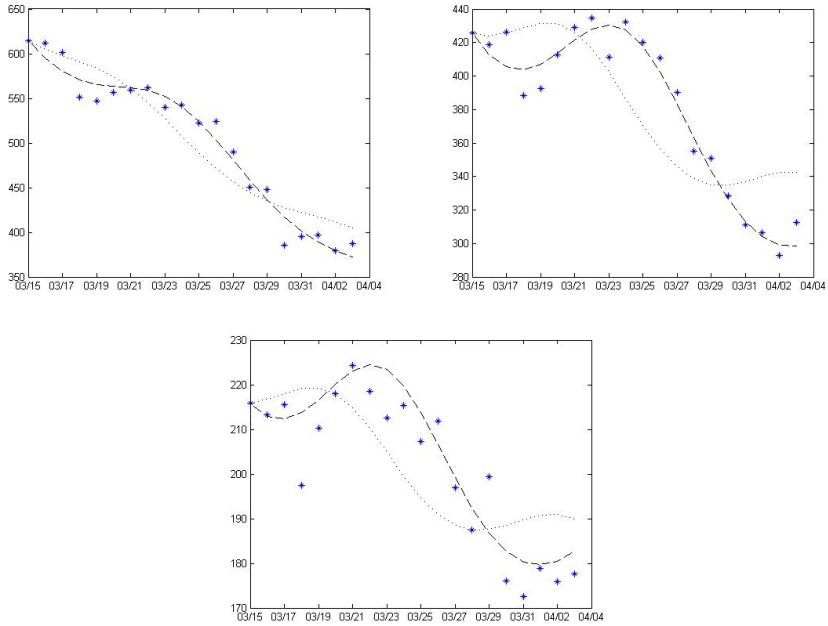

Fig. 2. From left to right: approximation of Ethereum, Dash and Monero currency prices. The asterisks represent real data, dashed lines are solutions of (1) optimized by fminsearch and dotted lines are solutions of (1) computed only by least square method.

suitable for approximation by solutions of model (1). Our choice is due to the problem that suitable data are usually not freely available. Despite of this, it is also seen that the approximation of such bad data using Matlab function fminsearch is better than the approximation by least square method without using fminsearch.

Applying least square method along with fminsearch, we obtained coefficients P_{ij} and the resulting system is given by

$$\begin{aligned}
\frac{dx}{dt} &= x(-0.3844 + 0.0000x - 0.0014y + 0.0044z),\\
\frac{dy}{dt} &= y(-0.6470 - 0.0001x - 0.0025y + 0.0079z),\\
\frac{dy}{dt} &= z(-0.3419 + 0.0002x - 0.0020y + 0.0049z).
\end{aligned} \tag{4}$$

To determine mutual interactions between competitors, we can look at coefficients of mixed terms xy, yz and xz; see [5]. The equal signs of coefficients P_{41}, P_{23} indicate that x is cooperating with z and when the population of x decreases rapidly then also

Period	x	y	z
Jan-98	0.52	0.15	0.33
Jul-98	0.47	0.21	0.32
Jan-99	0.43	0.27	0.3
Jul-99	0.4	0.31	0.29
Jan-00	0.38	0.35	0.27
Jul-00	0.37	0.36	0.27
Jan-01	0.36	0.37	0.27
Jul-01	0.36	0.37	0.27
Jan-02	0.35	0.38	0.27
Jul-02	0.36	0.38	0.26
Jan-03	0.37	0.39	0.24
Jul-03	0.38	0.39	0.23
Jan-04	0.39	0.39	0.22
Jul-04	0.38	0.39	0.23
Jan-05	0.37	0.39	0.24
Jul-05	0.38	0.39	0.23
Jan-06	0.38	0.4	0.22
Jul-06	0.36	0.39	0.25
Jan-07	0.34	0.38	0.28

Fig. 3. Share data

z decreases. Signs of coefficients P_{42} and P_{33} indicate that y is predating on z. The population of y strongly reflects the population of z. Of course in reality, the system of interactions between competitors is not isolated, since the prices of our three crypto currencies coincide with prices of hundreds of others, with economical and political changes etc.

3.2. Example 2

In this example, we use historical data originally used in [3] and also used in [4]. The data represent shares of three telephone service providers (competitors) (see Figure 3) and the sum of these shares in each row is equal to 1. We can look at any of these competitors from two different points of view:

(1) The competitor competes with each of two species and both them compete to each other, i.e. three mutual competitors in one system - the same approach as in Example 1.

(2) The competitor competes with the rest of the species (the two other species form one competitor), i.e. only two competitors in one system - we sum the values of the shares of same date in columns for competitors y, z and then find the coefficients of model (1) with $m = 2$ using fminsearch.

In Figure 4, we see that the second approach still provides us a good approximation of historical data values, especially in case of competitors x and y.

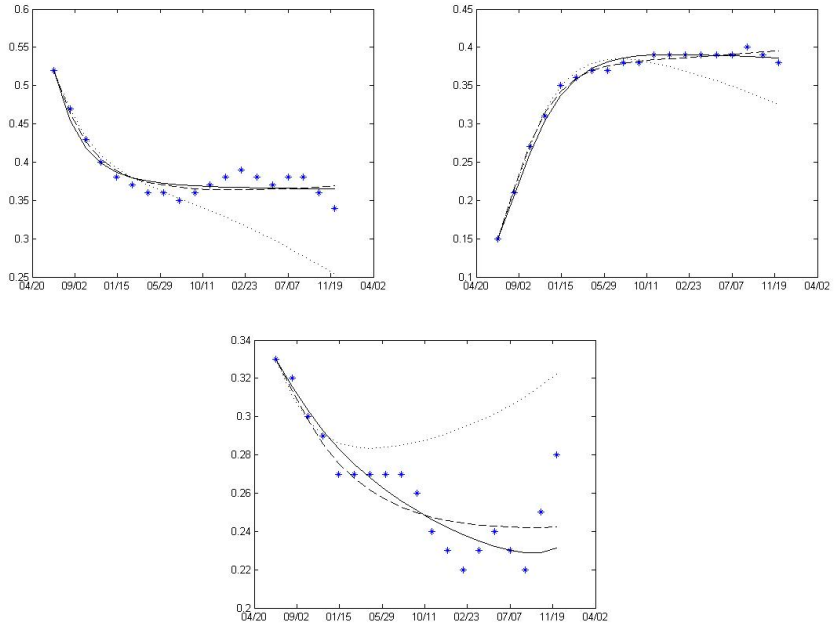

Fig. 4. From left to right: approximation of telephone service providers x, y, z. The asterisks represent real data, dashed lines are solutions of (1) with $m = 3$ (i.e. first point of view), full lines are solutions of (1) with $m = 2$ (i.e. second point of view) and dotted lines are initial solutions of (1) without using fminsearch.

The case z, however, is not so clear. Comparing the values of error function (i.e. sum of squares of differences of data and solutions of system 1 with optimized coefficients P_{ij}) we come to conclusion: second approach did slightly better than first one: In the first approach, the minimum of the error function is 0.0079 and fminsearch found it in 10.806295 seconds, in the second approach, the minimum of the error function is 0.0056 and fminsearch found it in 4.166628 seconds.

4. CONCLUSIONS

(1) As illustrated in Example 1, Matlab function fminsearch provides us better approximations than only the least square method. fminsearch is a robust however relatively slow function and one can consider using a different function for finding minima especially for large data set.

(2) For volatile data set such as prices, it may be better to use this method along stochastic calculus to achieve better approximations.

(3) Example 2 shows us that competition of one species with the rest of species (the second approach in Example 2) may lead to similar results as mutual competition of three species (the first approach). The second one can be even better for small data set, since only minimization of a function of six variables is required instead of twelve. Obviously, this costs less counting time (more than twice less in our case).

APPENDIX

In the Appendix, we provide the Matlab source code we use in our paper. Function "est" is the main function that calculates initial coefficients p_{ij} by using least squares method (function "Param") then calculates optimized P_{ij} with fminsearch and compares corresponding solutions of Lotka - Volterra models with both p_{ij} and P_{ij}.

```
function [  ] = est

data = xlsread('crypto.xls');

T=size(data,1);
m=size(data,2)-1;
D=data(1:T,(2:m+1));
E=data(1:T,1)+693960*ones(T,1);

% The Param function chooses initial p_ij for system
% x' = x ( p_11 + p_21 x + p_31 y + p_41 z ),
% y' = y ( p_12 + p_22 x + p_32 y + p_42 z ),
% z' = z ( p_13 + p_23 x + p_33 y + p_43 z )
% and we calculate initial solution.
p = Param(D);
[~,init]=ode15s(@(t,y) y.*(p(:,1)+p(:,2:m+1)*y),(1:T)',D(1,:));

[P,~]=fminsearch(@(p) err(p,D,T,m),p);

[~,opt1]=ode15s(@(t,y) y.*(P(:,1)+P(:,2:m+1)*y),(1:T)',D(1,:));

i=3;
```

```
opt2=est2(i,D);
plot(E,D(:,i),'*');
hold on
plot(E,init(:,i),':k','LineWidth',0.2);
hold on
plot(E,opt2,'k','LineWidth',0.5);
hold on
plot(E,opt1(:,i),'--k','LineWidth',0.8);
hold off
datetick('x',6,'keepticks','keeplimits');

end
```

Function "Param" uses least squares method to obtain initial values p_{ij} and contains fix for the case of badly conditioned matrix $A^{\intercal}A$.

```
function [ P ] = Param(D)

n=size(D,1);
m=size(D,2);

A = zeros(n-1,m+1);
B = zeros(n-1,1);
P = zeros(m,m+1);

for k = 1:m
    A(:,1) = (D(1:(n-1),k)+D(2:n,k))/2;
    for j = 1:m
        A(:,1+j) = (D(1:(n-1),k).* D(1:(n-1),j)+D(2:n,k).* D(2:n,j))/2;
    end
    B(:,1) = D(2:n,k)-D(1:(n-1),k);
    C = A' * A;
    E = A' * B;
    if rcond(C)<10^(-10)
        l=1;
        C1 = C(setdiff(1:(m+1),1),setdiff(1:(m+1),1));
```

```
        for i = 1:(m+1)
            v=setdiff(1:(m+1),i);
            if rcond(C(v,v))>rcond(C1)
            C1 = C(v,v);
            l=i;
            end
        end
        E = E(setdiff(1:(m+1),l));
        P(k,setdiff(1:(m+1),l)) = C1\E;
        P(k,l) = 0;
    else
        P(k,:) = C\E;
    end
end
```

Function ”est2” is similar to ”est”, however is based on the second approach from Example 2.

```
function [ opt ] = est2(i,data)

T=size(data,1);
m=size(data,2);
D=zeros(T,2);
D(:,1)=sum(data(1:T,setdiff(1:m,i)),2);
D(:,2)=sum(data(1:T,i),2);

p = Param(D);
[P,~]=fminsearch(@(p) err(p,D,T,2),p);
[~,opt0]=ode15s(@(t,y) y.*(P(:,1)+P(:,2:3)*y),(1:T)',D(1,:));
opt=opt0(:,2);
end
```

Function ”err” represents the error functional that is minimized by fminsearch.

```
function [ value ] = err(p,data,T,m)

[~,y] = ode15s(@(t,y) y.*(p(:,1)+p(:,2:m+1)*y),(1:T)',data(1,:));
n=size(y,1);
```

```
value = sum(sum ((y-data(1:n,:)).^2));
end
```

REFERENCES

M. Shatalov, J.C. Greeff, I. Fedotov, S.V. Joubert: Parametric identification of the model with one predator and two prey species. TIME1008 International Conference Proceedings, (2008), 101-110.

M. Shatalov, I. Fedotov: On identification of dynamical system parameters from experimental data. Research Group in Mathematical Inequalities and Applications 10(1) (2007) 1-9.

C. Michalakelis, T.S. Sphicopoulos, D. Varoutas: Modelling competition in the telecommunications market based on the concepts of population biology. Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews 4 (2011) 200-210.

P.H. Kloppers, J.C. Greeff: Lotka-Volterra model parameter estimation using experimental data. Applied Mathematics and Computation 224 (2013) 817-825.

Y. Takeuchi: Global Dynamical Properties of Lotka-Volterra Systems. Singapore: World Scientific Publishing Company, 1996.

Michal Fečkan
Department of Mathematical Analysis and Numerical Mathematics,
Faculty of Mathematics, Physics and Informatics,
Comenius University in Bratislava,
Mlynská dolina, 842 48 Bratislava, Slovakia
Mathematical Institute of Slovak Academy of Sciences,
Štefánikova 49, 814 73 Bratislava, Slovakia
e-mail: Michal.Feckan@fmph.uniba.sk

Július Pačuta
Department of Mathematical Analysis and Numerical Mathematics,
Faculty of Mathematics, Physics and Informatics,
Comenius University in Bratislava,
Mlynská dolina, 842 48 Bratislava, Slovakia
e-mail: julius.pacuta@fmph.uniba.sk

Received February 2019.

Best predictors in logarithmic distance between positive random variables

H. GZYL

Abstract

The metric properties of the set in which random variables take their values lead to relevant probabilistic concepts. For example, the mean of a random variable is a best predictor in that it minimizes the L_2 distance between a point and a random variable. Similarly, the median is the same concept but when the distance is measured by the L_1 norm.
Also, a *geodesic distance* can be defined on the cone of strictly positive vectors in $\mathbb{R}^n$ in such a way that, the minimizer of the distance between a point and a collection of points is their geometric mean.
That geodesic distance induces a distance on the class of strictly positive random variables, which in turn leads to an interesting notions of conditional expectation (or best predictors) and their estimators. It also leads to different versions of the Law of Large Numbers and the Central Limit Theorem. For example, the lognormal variables appear as the analogue of the Gaussian variables for version of the Central Limit Theorem in the logarithmic distance.

Mathematics Subject Classification 2010: 60B99, 60B12, 60A99.
Keywords: Prediction in logarithmic distance, Law of large numbers in logarithmic distance, Central Limit Theorem in logarithmic distance, Logarithmic geometry for positive random variables.

1. INTRODUCTION AND PRELIMINARIES

The study of random variables and processes taking values in spaces with geometries other than Euclidean in not new. Consider the textbooks by Kunita and Watanabe [5] or by Hsu [3] to mention just two. In this line of work, the distance between points in the base manifold is replaced by a geodesic distance derived from a Riemannian metric placed. Such distance is inherited by random variables taking values in the manifold.

It should not then be surprising that the notion of best predictor of a random variable by variables of a given class, should depend on the metric of the manifold. In this note we shall consider the manifold to be $\mathscr{M} = (0,\infty)^n$, which is an open set in $\mathbb{R}^n$, which is also a commutative group with respect to component wise multiplication. We postpone the study of the geometry of this group to the appendix. Here we mention that what we do is the commutative version of a more elaborate geometry in the space of symmetric matrices. The reader can check with Lang [6] in which a relation of this geometry to Bruhat-Tits spaces is explained, or in Lawson and Lim [7] or Mohaker [9]

10.2478/jamsi-2019-0006

and references therein, where the geometric mean property in the class of symmetric matrices is established. More recently Resigny et al. [1] and Schwartzman [10] used the same geometric setting to study the role of such geometry in a large variety of applications. The applications of the geometric ideas in these references concern the non-commutative case, but the simplest commutative case and its potential usefulness for positive random variables seems not to have been explored.

As mentioned in the abstract, it is the purpose of this note to explore the possible usefulness of measuring distances between positive numbers by a logarithmic distance resulting from an interesting group invariant metric.

In the appendix we establish that the distance between any two points $x_i, x_2 \in \mathscr{M}$ is given by

$$d(x_1,x_2)^2 = \sum_{i=1}^{n} (\ln x_1(i) - \ln x_2(i))^2 . \tag{1.1}$$

This makes $\mathscr{M}$ a Tits-Bruhat space in which the distance satisfies a semi-parallelogram law. This is contained in Theorem 8.1. We shall use this property to establish the uniqueness of conditional expectations. And the group structure in $\mathscr{M}$ will be inherited in a curious way by the conditional expectations (or by the best predictors) in the logarithmic distance (8.1).

Starting from the logarithmic distance on $\mathscr{M}$, and from the fact that it satisfies the semi-parallelogram law, we come to the main objective of the paper, which is to consider the notion of best predictor (conditional expectations) in that distance. These matters will be taken up in Sections 2 and 3, where we shall introduce the notion of $\ell-$expected value and $\ell-$conditional expectation, which will denote the best predictors in the logarithmic distance (hence the $\ell-$prefix). We examine there some of the basic properties of these constructs.

In Section 4 we present the two most basic estimators, namely, that of the $\ell-$mean and that of the $\ell-$variance, and explain how the law of large numbers and the central limit theorem for these estimators relates to the standard law of large numbers and the central limit theorems.

In section 5 we examine how the notion of martingale related to the $\ell-$conditional expectation relates to the standard notion of martingale. We shall do it in discrete time, but the extension to continuous time is quite direct. In Section 6 we examine Markowitz portfolio theory when the distance between (gross) returns is the logarithmic distance.

As said, we leave the study of the geometry on $\mathscr{M}$ to the appendix. There we explain how the logarithmic distance between strictly positive vectors is actually a geodesic distance in that manifold. For that we shall present some results from Lang's [6], but in a simpler, commutative setup. The basic idea behind our constructions has been very much studied in geometry. The idea is to put a group action upon $\mathscr{M}$ and construct a scalar product on the tangent bundle TM that is invariant under the group action. That scalar product determines the logarithmic (geodesic) distance on $\mathscr{M}$ that interests us. That appendix can be read independently of the previous sections dealing with probabilistic aspects.

2. BEST PREDICTORS IN LOGARITHMIC DISTANCE

Our set up here consists of a probability space $(\Omega, \mathscr{F}, P)$ and we shall be concerned with the cone $\mathscr{C}$ of $P-$almost everywhere (a.e. for short) finite and strictly positive ($\mathscr{M}$-valued) random variables. As usual, we identify variables that are $P-$a.e. equal. Since the operations among vectors are component wise, to reduce to the case $n = 1$ only takes a simple notational change. To shorten the description of the random variables used in the statements coming up below, let us introduce the following notations. For $p > 1$ (we shall be concerned with $p = 1, 2$ only) define:

$$L_p = \{X \in \mathscr{F} \,|\, E[|X_i|^p] < \infty,\ \ i = 1, ..., n\}$$

$$Ln_p = \{X \in \mathscr{C} \,|\, \ln X \in L_p\}, \qquad LLn_p = L_p \bigcap Ln_p.$$

Let X_1 and X_2 be two strictly positive random variables in Ln_2. The (logarithmic) distance between them is defined to be

$$d_\ell(X_1, X_2)^2 \equiv E\left[\sum_{i=1}^{n} (\ln X_1(i) - \ln X_2(i))^2\right] \tag{2.1}$$

Since we are identifying variables that are a.e equal, $d_\ell(X_1, X_2)$ is a distance on $\mathscr{C}$. Similarly to $m = E[X]$ being the constant that minimizes the Euclidean (squared) distance to X, we have

PROPOSITION 2.1. With the notations introduced above, let $X \in Ln_2$. The vector m_ℓ that minimizes the logarithmic distance to X is given by

$$m_\ell(X) = \exp(E[\ln X]).$$

The proof of the first assertion is computational, and the second results from an

application of Jensen's inequality. When there is no risk of confusion, we shall write $m_\ell(X) = m_\ell$. Keep in mind that the operations are componentwise, and that $m_\ell(X)_j = \exp(E[\ln X_j])$ for $j = 1,...,n$. If $X \in LLn_1$, we also have $m_\ell \leq E[X]$.

COMMENT. To define $m_\ell(X)$ it suffices that $X \in Ln_1$, but if we want it to be a best predictor in the logarithmic distance, we require $X \in Ln_2$.

And the analogue of the notions of covariance and centering are contained in the following definition.

DEFINITION 2.1. Let now $X, Y \in Ln_2$. We define the logarithmic covariance matrix of the non-negative random variables X and Y by

$$Cov_\ell(X,Y) \equiv E[(\ln X - \ln m_\ell(X))(\ln Y - \ln m_\ell(Y))^t] = Cov(\ln X, \ln Y).$$

Let Σ be the matrix with components $E[(\ln X_i - \ln m_\ell(X_i))(\ln Y_j - \ln m_\ell(Y_j))]$. If the matrix Σ is invertible, we define the "centered" (in logarithmic distance) version of X by

$$X^c \equiv \exp\left(\Sigma^{-1/2}(\ln X - \ln m_\ell(X))\right)$$

The need for the exponentiation is clear: First we have to "undo" the taking of the logarithms and second, the argument of the exponential function is a vector in $\mathbb{R}^n$ which yields a positive vector after exponentiation. It takes a simple computation to verify that

$$m_\ell(X^c) = \mathbf{1}, \quad \Sigma_\ell(X^c) = \mathbb{I}.$$

A variation on the previous theme consists of predicting a variable Y by a variable X in logarithmic distance. The extension of the previous result is contained in the following statement.

PROPOSITION 2.2. Let Y and X be in Ln_2. Then the $\sigma(X)-$measurable random variable that minimizes the logarithmic distance (2.1) to Y is given by

$$E_\ell[Y|X] = \exp(E[\ln Y \,|\, X]).$$

And we also have $E_\ell[Y|X] \leq E[Y \,|\, X]$.

The proof of Proposition 2.2 follows the same pattern as the standard proof. Just notice that $\phi(X) = \exp(E[\ln Y \,|\, X])$ is a bounded, $\sigma(X)-$measurable random variable, such that $\ln \phi(X) = E[\ln Y \,|\, X]$ minimizes the Euclidean square distance to $\ln Y$.

COMMENT. As above, we remark that $E_\ell[Y|X]$ can be defined for $Y \in Ln_1$, but if we want it to be a best predictor in logarithmic distance, we need $Y \in Ln_2$.

Note that the last inequality mentioned in the statement does not mean that one of the estimators is better than the other in any sense. They are minimizers in different metrics. Also, since linear combinations in an exponent are transported as scaling and powers, we have the following analogue to linear prediction for positive random variables.

PROPOSITION 2.3. Let Y and X be positive real variables with square integrable logarithms. The values of $a > 0$ and $b \in \mathbb{R}$ that make $Y^{\#} \equiv aX^b$ the best predictor of Y in the logarithmic metric, are given by

$$\begin{cases} a = \exp\left(E[(\ln Y)] - bE[(\ln X)]\right) \\ b = \frac{1}{D}\left(E[\ln X \ln Y] - E[\ln X]E[\ln Y]\right) \\ D = E[(\ln X)^2] - (E[\ln X])^2 = \sigma^2(\ln X). \end{cases}$$

The proof follows the standard computation starting from the definition of $d(Y, aX^b)_\ell$. Certainly the result is natural as the linear structure of $\mathbb{R}$ is transferred multiplicatively onto $(0,\infty)$ by the exponential mapping. Also, the extension to random variables taking values in higher dimensional $\mathscr{M}$ is direct, but notationally more cumbersome.
A simple computation leads to

$$m_\ell(Y^{\#}) = E_\ell[Y^{\#}] = e^{E[lnY]}, \quad \sigma_\ell(Y^{\#}) = b^2\sigma^2(\ln X).$$

3. LOGARITHMIC CONDITIONAL EXPECTATION AND SOME OF ITS PROPERTIES

Here we extend the semi-parallelogram property mentioned in Theorem (8.1) to strictly positive random variables.

LEMMA 3.1. All random variables mentioned are supposed to be in Ln_2. Let X_1 and X_2 be as mentioned. Then there exits $Z \in Ln_2$ such that for any Y we have

$$d(X_1, X_2)_\ell^2 + 4d(Z, Y)_\ell^2 \leq 2d(Y, X_1)_\ell^2 + 2d(Y, X_2)_\ell^2.$$

To prove this, use the second comment after Theorem (8.1) at every $\omega \in \Omega$ to obtain the pointwise version of the semi-parallelogram property, and then integrate with respect to P. Clearly $Z = (X_1X_2)^{1/2} \in LLn_2$. Below we apply this to obtain the uniqueness of the extension of the standard notion of conditional expectation.

THEOREM 3.1. Let $\mathscr{G} \subset \mathscr{F}$ be a $\sigma-$algebra, and let Y be non-negative with square integrable logarithm. Then, the unique -up to a set of P measure 0-, positive $X^* \in \mathscr{G}$ that makes $d(Y,X)_\ell^2$ minimum over $\{X \in \mathscr{G},\, X>0, E[(\ln X)^2] < \infty\}$, is given by $X^* = \exp\left(E[\ln Y \,|\mathscr{G}]\right)$. To be consistent with the notations introduced above, we shall write $X^* = E_\ell[Y\,|\mathscr{G}]$.

PROOF. The existence follows the same pattern of proof as the propositions in the previous section, that is $E[\ln Y\,|\mathscr{G}]$ minimizes the ordinary square distance to $\ln Y$, and it is the unique (up to sets of P measure 0). We shall use the semi-parallelogram property to verify the uniqueness. For that, let X some other possible minimizer of the logarithmic distance. Now set $Z = \sqrt{XX^*}$ (keep in mind the second comment after Theorem (8.1)), and observe that according to the semi-parallelogram property

$$d(X^*,X)_\ell^2 + 4d(Y,Z)_\ell^2 \leq 2d(Y,X^*)_\ell^2 + 2d(Y,X)_\ell^2.$$

Since by definition, $d(Y,Z)_\ell^2$ is larger than any of the two distances in the right hand side of the inequality, it follows that necessarily $d(X^*,X)_\ell^2 = 0$.

Notice that in the group structure on $\mathscr{M}$ described in the Appendix, the analogue to the multiplication by scalars is replaced by the exponentiation. This is used to verify the analogue of the standard definition of conditional expectation in logarithmic distance. For the heuristics see the comment in the Appendix.

THEOREM 3.2. Let Y be a $\mathscr{M}$ valued random variable such that $Y \in Ln_1$. Then $E_\ell[Y|\mathscr{G}] = \exp\left(E[\ln Y|\mathscr{G}]\right)$ is the unique $\mathscr{G}$-measurable, $\mathscr{M}$-valued random variable, such that for any bounded $\mathscr{G}$-measurable real valued H the following holds:

$$E_\ell[Y^H] = E_\ell[E_\ell[Y|\mathscr{G}]^H].$$

PROOF. It follows the standard pattern. Notice that $Y^H \in Ln_1$, therefore

$$E_\ell\big[Y^H\big] = e^{E[H\ln Y]} = e^{E[HE[\ln Y|\mathscr{G}]]} = e^{E\left[\ln\left(\exp(E[\ln Y|\mathscr{G}]\right)^H\right]} = E_\ell\big[E_\ell\big[Y|\mathscr{G}\big]^H\big].$$

The intermediate steps consist of an application of the standard conditional expectation rules.

Let us now verify some standard and non standard properties of the notion of conditional expectation introduced above. Keep in mind that the arithmetic operations with positive vectors are componentwise.

THEOREM 3.3. Let $Y \in LLn_2$ and let $\mathscr{H} \subset \mathscr{G}$ be two sub-σ-algebras of $\mathscr{F}$. Then, up to a set of measure 0, the following hold:

1) $E_\ell[Y\,|\{\emptyset,\Omega\}] = E_\ell[Y]$.
2)$E_\ell[E_\ell[Y|\mathscr{G}]\,|\mathscr{H}] = E_\ell[Y\,|\mathscr{H}]$.
3)Let $Y_1,...,Y_k$ be in LLn_2, and $w_i \in \mathbb{R}$. The analogue of the linearity property of the standard conditional expectation is the following multiplicative property:

$$E_\ell[\prod_{i=1}^{k} Y_i^{w_i}\,|\mathscr{G}] = \prod_{i=1}^{k}\Big(E_\ell[Y|\mathscr{G}]\Big)^{w_i}$$

4) If Y is independent of $\mathscr{G}$ in the standard sense, then $E_\ell[Y\,|\mathscr{G}] = E_\ell[Y]$.

PROOF. The first assertion is simple consequence of the definition . To verify the second we start from the definition and carry on:.

$$E_\ell[E_\ell[Y|\mathscr{G}]\,|\mathscr{H}] = \exp\Big(E\big[\ln\exp E[\ln Y|\mathscr{G}|\mathscr{H}\big]\Big) = \exp\Big(E[E[\ln Y|\mathscr{G}|\mathscr{H}]\Big),$$

and now apply the standard tower property of conditional expectations to finish the proof of the assertion.
It is in the proof of (3) where the logarithmic distance plays a curious role. The proof of the assertion is a simple computation starting from the definition:

$$E_\ell[\prod_{i=1}^{k} Y_i^{w_i}\,|\mathscr{G}] = \exp\Big(E\big[\sum w_i \ln Y_i\,|\mathscr{G}\big]\Big) = \prod_{i=1}^{k}\Big(E_\ell[Y|\mathscr{G}]\Big)^{w_i}.$$

The fourth property is also simple to establish using the definition and the standard notion of independence.

4. ESTIMATORS AND LIMIT THEOREMS

In this section we shall consider the case $n = 1$. The notation is a bit simpler in this case. That is, we shall forget about the symbols in boldface for a while.

Making use of Proposition (8.1) the following definition is clear:

DEFINITION 4.1. Let $X_1,...,X_K$ be positive random variables. We define their empirical logarithmic mean by

$$\hat{m}_{\ell,K}(X) = \left(\prod_{j=1}^{K} X_j\right)^{1/K}.$$

And a the standard law of large numbers becomes:

THEOREM 4.1. Let X_j, $j \geq 1$ be a collection of i.i.d. positive random variables defined on $(\Omega,\mathscr{F},P)$ having finite logarithmic variance σ_ℓ^2 and mean m_ℓ. Then

$\hat{m}_{\ell,K}(X)$ is an unbiased estimator of the logarithmic mean $m_\ell(X)$ and

$$\hat{m}_{\ell,K}(X) = \left(\prod_{j=1}^{K} X_j\right)^{1/K} \to m_\ell(X)$$

almost surely w.r.t. P as $K \to \infty$.

The proof is clear. Since

$$\hat{m}_{\ell(X),K} = \exp\left(\frac{1}{K}\sum_{j=1}^{K} \ln X_j\right),$$

we can invoke the strong law of large numbers, see Borkhar [2] or Jacod and Protter [4], plus the continuity of the exponential function to obtain our assertion. That $\hat{m}_{\ell,K}(X)$ has logarithmic mean $m_\ell(X)$ is clear.

In analogy with the standard notion of empirical variance, we can introduce

DEFINITION 4.2. With the notations introduced above and under the assumptions in Theorem 4.1, the empirical estimator of the logarithmic variance is defined by

$$\hat{\sigma}^2_{\ell,K}(X) = \frac{1}{K-1}\sum_{j=1}^{K} \left(\ln X_j - \ln \hat{m}_\ell(X)\right)^2.$$

And, as in basic statistics, we have

THEOREM 4.2. With the notations introduced above, and under the assumptions of Theorem (4.1), $\hat{\sigma}^2_\ell(X)$ is an unbiased estimator of the logarithmic variance and

$$\hat{\sigma}^2_{\ell,K}(X) \to \sigma^2_\ell(X)$$

almost surely w.r.t. P as $K \to \infty$.

But perhaps more interesting is the following version of the central limit theorem. It brings to the fore the role of lognormal variables as the analogue to the Gaussian random variables in the class of positive variables.

THEOREM 4.3. Suppose that $X_j, j \geq 1$ are a collection of i.i.d. random variables defined on a probability space $(\Omega, \mathscr{F}, P)$ with logarithmic mean $m_\ell = E[\ln X_j]$ and $E[(\ln X_i)^2] < \infty$. Then

$$\left(\prod_{j=1}^{K} \frac{X_j}{m_\ell}\right)^{1/\sqrt{K}} \to e^X$$

in law as $K \to \infty$, where $X \sim N(0, \sigma_\ell^2)$.

PROOF. Observe that

$$\Big(\prod_{j=1}^{K} \frac{X_j}{m_\ell}\Big)^{1/\sqrt{K}} = \exp\Big(\frac{1}{\sqrt{K}} \sum_{j=1}^{K} (\ln X_j - \ln m_\ell)\Big).$$

From the standard proof of the central limit theorem we know that $\frac{1}{\sqrt{K}} \sum_{j=1}^{K} (\ln X_j - \ln m_\ell)$ converges in probability to an $N(0, \sigma_\ell^2)$ random variable and therefore, since the exponential function is continuous, the same convergence holds for $\Big(\prod_{j=1}^{K} \frac{X_j}{m_\ell}\Big)^{1/\sqrt{K}}$. Thus concludes the proof of our assertion.

5. $\ell-$MARTINGALES IN DISCRETE TIME

As there is a notion of $\ell-$conditional expectation, there must be a corresponding notion of $\ell-$martingale. In this section we examine some of its very simple properties. As usual, the basic setup consists of the probability space $(\Omega, \mathscr{F}, P)$ and a filtration $\{\mathscr{F}_n, n \geq 0\}$.

THEOREM 5.1. Let the $\mathscr{M}-$valued process $\{X_n; n \geq 0\}$ be such that $X_n \in \mathscr{F}_n$ and the $\xi_n = \ln X_n$ are integrable. Then X_n is an ℓ-martingale (resp. sub-martingale, super-martingale) if and only if $\{\xi_n\}$is an ordinary martingale (resp. sub-martingale, super-martingale).

Also, if X_n is an $\ell-$martingale, it is an ordinary sub-martingale.

PROOF. For $n \geq 0$ and $k \geq 1$

$$E_\ell[X_{n+k}|\mathscr{F}_n] = e^{E[\xi_{n+k}|\mathscr{F}_n]}$$

from which the assertion of the theorem drops out. For the second assertion note that

$$E_\ell[X_{n+k}|\mathscr{F}_n] = X_n = e^{\xi_n} = e^{E[\xi_{n+k}|\mathscr{F}_n]} \leq E[e^{\xi_{n+k}}|\mathscr{F}_n] = E[X_{n+k}|\mathscr{F}_n]$$

The middle step drops out from Jensen's inequality.

The corresponding version of the Doob decomposition theorem, say for sub-martingales, goes as follows.

THEOREM 5.2. With the notations introduced above, let $\{X_n\}$ be an $\mathscr{M}-$valued $\ell-$sub-martingale. Then there exist an $\mathscr{M}-$valued $\ell-$martingale $\{Y_n\}$ and an increasing $\mathscr{M}-$valued process A_n, such that $X_n = Y_n A_n$.

PROOF. Just apply the Doob decomposition theorem to $\xi_n = \ln X_n$ and use

$X_n = e^{\xi_n}$.

6. LOGARITHMIC GEOMETRY AND PORTFOLIO THEORY

Let us introduce a slight change of notation to conform with the notation in financial modeling. By the generic R we shall denote the (gross) return of any asset of portfolio, which means the quotient of its current value divided by its initial value.

To begin with, we proved in the Appendix, see (8.3), that the curve $R_1^w R_2^{1-w}$ is a geodesic in the logarithmic distance between the points R_1 and R_2. That curve can be thought of as a weighted geometric mean of R_1 and R_2. This remark leads to variation on the theme of "return" of a portfolio. In our setup, a generic portfolio w, characterized by the weights $w_1,...,w_K$, of assets with gross returns $R_1,...,R_k$, has a weighted return given by the geometric mean: $\prod_{i=1}^K R_i^{w_i}$. To push the geodesic interpretation a bit further, that geometric mean can be thought of as a sequence of geodesic walks joining say R_1 to R_K. Anyway, note that the logarithm of the $\ell-$mean m_ℓ given by,

$$\ln m_l = \sum_{i=1}^K w_i E[\ln R_i] \tag{6.1}$$

is clearly the logarithmic rate of growth of the portfolio w. Recall as well that the square of the logarithmic distance of m_ℓ to $\prod_{i=1}^K R_i^{w_i}$ is given by

$$d(\prod_{i=1}^K R_i^{w_i}, m_\ell)^2 = Var(\sum_{i=1}^K w_i \ln R_i) = (w, \Sigma w) \tag{6.2}$$

where Σ is the covariance matrix of the logarithmic returns. Imitating Markowitz's portfolio theory, we assign to any portfolio w its logarithmic mean $m_\ell(w)$ and its logarithmic variance $\sigma_\ell^2(w)$. According to Markowitz's proposal a portfolio is optimal when it minimizes the variance for a given expected value of its (rate of) return. The following result contains the analogue of the classical Markowitz result, but for the case of the logarithmic results.

PROPOSITION 6.1. With the notations introduced above, the weights $w_i^*,....,w_K^*$ that make the logarithmic variance, $\sigma_\ell^2(w) = d(\prod_{i=1}^K R_i^{w_i}, m_\ell)^2$ minimal subject to the constraints $\sum w_i = 1$ and $m_\ell(w) = e^\mu$, are the same as the weights that minimize $Var\Big(\sum_{i=1}^k w_i \ln R_i\Big)$ subject to $E[\sum_{i=1}^k w_i \ln R_i] = \mu$ and $\sum w_i = 1$.

This is a standard quadratic optimization problem, whose solution is simple starting from (6.2). We refer the interested reader to Luenberger [8] or to Shiryaev [11] for

more details about the classical Markowitz portfolio optimization theory.

7. CONCLUDING COMMENTS

In this note we proposed an alternative metric in the set of positive vectors, such that when distance between random variables is measured in this metric, the standard notions of best predictors, their estimation, and the classical convergence results, acquire a different but intuitively related form.

Also, as a simple application to finance, when assets are characterized by their gross returns (which by definition are positive random variables), the concept of return of a portfolio becomes a weighted geometric average, and the standard portfolio choice methodology appears in a slightly different guise. Readers familiar with the basics of the methodology will find it clear that the analogue of the efficient frontier, market portfolio, market line and CAPM have a counterpart within the formalism developed above, but this is not the place to pursue the matters.

8. APPENDIX: THE LOGARITHMIC DISTANCE BETWEEN POSITIVE VECTORS

We shall think of the vectors in $\mathbb{R}^n$ as functions $\xi : \{1,...,n\} \to \mathbb{R}$, and all standard arithmetical operations either as component wise operations among vectors or point wise operations among functions. Let us denote by $\mathscr{M} = \{x \in \mathbb{R}^n \,|x(i) > 0, i = 1,...n\}$ the set of all positive vectors. $\mathscr{M}$ is an open set in $\mathbb{R}^n$ which is trivially a manifold over $\mathbb{R}^n$, having $\mathbb{R}^n$ itself as tangent space at each point. We shall use the standard notation TM_x to stress this point.

COMMENT. As a collateral detail we mention that $\mathscr{M}$ is a vector space, in which the (commutative) group operation is given by the componentwise multiplication, and the standard multiplication by scalars is given by $(a,x) \in (\mathbb{R},\mathscr{M}) \to x^a$. This detail helps to intuitively understand Theorem 3.2.

Here $\mathscr{M}$ plays the role that the positive definite matrices play in the works by Lang, Lawson-Lim and Mohaker mentioned in the Introduction. The role of the group of invertible matrices in those references is to be played here by $G = \{g \in \mathbb{R}^n \,|\, g(i) \neq 0, i = 1,...,n\}$, which clearly is an Abelian group respect to the standard product, in which the identity, denoted by e, is the vector with all components equal to 1. We shall make use the action $G : \mathscr{M} \to \mathscr{M}$ of G on $\mathscr{M}$ defined by $\tau_g(x) = g^{-1}xg^{-1}$. This action is clearly transitive on $\mathscr{M}$, and can be defined in the obvious way as an action on $\mathbb{R}^n$.

The transitivity of the action allows us to transport the scalar product on TM_e to any TM_x as follows. The scalar product between ξ and η at TM_e is defined to be the standard Euclidean product $(\xi,\eta)=\sum\xi_i\eta_i$, where we shall switch between $\xi(i)$ and ξ_i as typographical convenience dictates. Since $x=\tau_g(e)$ with $g=x^{-1/2}$. We define the scalar product transported to TM_x by

$$(\xi,\eta)_x\equiv(x^{-1}\xi,x^{-1}\eta)=(x^{-2}\xi,\eta).$$

This scalar product allows us to define the length of a differentiable curve as follows:

Let $x(t)$ be a differentiable curve in $\mathscr{M}$, its length is given by

$$\int_0^1\sqrt{(\dot{x},\dot{x})_x}dt.$$

With this definition, the distance between $x_1,x_2\in\mathscr{M}$ is defined by the expected

$$d(x_1,x_2)=\inf\{\int_0^1\sqrt{(\dot{x},\dot{x})_x}dt\,|\,x(t)\ \text{differentiable such that}\ x_1=x(0)\ \ x_2=x(1)\}\tag{8.1}$$

It takes an application of the Euler-Lagrange formula to see that the equation of the geodesics in this metric is

$$\ddot{x}(t)=x^{-1}\dot{x}^2,\ \ x(0)=x_1,\ \ x(1)=x_2,\tag{8.2}$$

the solution to which is

$$x(t)=x_1e^{t\ln(x_2/x_1)}=x_2^tx_1^{(1-t)}.\tag{8.3}$$

This allows us to compute the distance between x_1 and x_2 as

$$d(x_1,x_2)^2=\sum_{i=1}^n(\ln x_1(i)-\ln x_2(i))^2.\tag{8.4}$$

Similarly, the solution to (8.2) subject to $x(0)=x$, and $\dot{x}(0)=\xi$ is the (exponential) mapping $xe^{t\xi}$. With this notations we recall some results (in this simpler setup) from Chapter 5 of Lang (1995) under

THEOREM 8.1. With the notations introduced above we have:
1) The exponential mapping is metric preserving through the origin.
2) The derivative of the exponential mapping is measure preserving, that is, $\exp'(\xi)v=ve^{\xi}$ as a mapping $TM_x\to TM_{x\exp\xi}$, satisfies

$$(v,v)=(\exp'(\xi)v,\exp'(\xi)v)_{\exp(\xi)}$$

3) With the metric given by (1.1), $\mathscr{M}$ is a Bruhat-Tits space, that is, it is a complete metric space in which the semi-parallelogram law holds. This means that, given any $x_1, x_2 \in \mathscr{M}$, there exists a unique $z \in \mathscr{M}$ such that for ant $y \in \mathscr{M}$ the following holds

$$d(x_1,x_2)^2 + 4d(z,y)^2 \leq 2d(y,x_1)^2 + 2d(y,x_2)^2.$$

COMMENT.

1) The action τ_g defined a few paragraphs above coincides with parallel transport along geodesics.

2)The proofs take some space but are systematic and computational. In our case, commutativity makes things considerably simpler. The completeness of $\mathscr{M}$ is transferred from $\mathbb{R}^n$ via the exponential mapping.

3) The point z mentioned in item (3) is given by $z = \sqrt{x_1 x_2}$. Actually, a simple calculation provides the proof of the following slightly more general statement.

LEMMA 8.1. Let $x_1, ..., x_K$ be K points in $\mathscr{M}$. The point $\bar{x}_\ell$ that minimizes the sum of logarithmic distances (1.1) to the given points is given by their geometric mean, that is

$$\bar{x}_\ell = \left(\prod_{j=1}^{K} x_j \right)^{1/K}$$

ACKNOWLEDGEMENTS

I wish to thank the reviewers/editors for their comments to improve the presentation.

REFERENCES

[1] Arsigny, V., Fillard, P., Pennec, X. and Ayach, N. (2007). *Geometric Means in a Novel Vector Space Structure on Symmetric positive definite matrices*, SIAM J. Matrix Theory, **29**, 328-347.

[2] Borkhar, V. *Probability Theory*, Springer, New York, (1995).

[3] Hsu, E.P. Stochastic Analysis on Manifolds, Amer. Math. Soc., Providence, (2002).

[4] Jacod, J. and Protter, P. *Probability Essentials*, Springer, New York, (2000).

[5] Kunita, H. and Watanabe, S. *Stochastic Differential Equations and Diffusion Processes*, North Holland Pub. Co, Amsterdam, (1989).

[6] Lang, S. Math talks for undergraduates, Springer, New York, (1999).

[7] Lawson, J.D. and Lim, Y. (2001). *The Geometric mean, matrices, metrics and more*, Amer. Math.,Monthly, **108**. 797-812.

[8] Luenberger, D.G. *Investment Science*, Princeton Univ. Press, Princeton, (1980).

[9] Mohaker, M. *A differential geometric approach to the geometric mean of symmetric positive definite matrices*, SIAM Jour. Matrix Analysis & Applic., **26**, 735-747.

[10] Schwartzman, A. (2015). *Lognormal distribution and geometric averages of positive definite matrices*, Int. Stat. Rev., **84**, 456-486.

[11] Shiryaev, A.N, Essentials of Stochastic Finance. Facts, Models, Theory, World Scient. Pubs. (1999).

Henryk Gzyl
Centro de Finanzas IESA, Caracas, (Venezuela)
email: henryk.gzyl@iesa.edu.ve

Towards the non-stretchable and non-elongating string with stress-strain handling

R. ĎURIKOVIČ AND E. SIEBENSTICH

Abstract

We propose an approach for real-time physically semi-realistic animation of strings which directly manipulates the string positions by position based dynamics. The main advantage of a position based dynamics is its controllability. Instability problems of explicit integration schemes can be avoided. Specifically, we offer the following three contributions. We introduce the non-elongating and non-stretchable mass spring dynamics model based on Position Based Dynamics to simulate 1D string. We introduce a method for propagating the twisting angle along the chain of segments. In addition, we solve collision constraints by regularly distributing the spheres along the chain segments followed by particle projection to validate the positions. Proposed strain limiting constraint can handle the strings fixed in multiple locations contrary to single fixed side as is common for hair models. The use of multiple constraints provides an efficient treatment for stiff twisting and non-stretchable mass spring dynamics model.

Mathematics Subject Classification 2010: 78A05, 68P01, 68U05
Keywords: non-elongating strings, twisting, tearing, flicking, string animation

1. INTRODUCTION

In this paper we present a method for real-time physically semi-realistic animation of strings with emphasis on believable behavior during twisting, tearing and flicking. The last result is the physically-based model that is visually plausible and stable, yet simple to implement.

The contribution of this paper: 1. We propose the non-elongating and non-stretchable properties of mass spring dynamics model based on Position Based Dynamics (PBD) [Müller et al. 2007] to simulate 1D string. 2. We propose a method for propagating the twisting angle increments associated with each segment, which can handle both uniform and nonuniform torsional rigidity. 3. We propose a method for tension calculation resulting from twisting, tearing and flicking.

We describe details of implementation and stumbling blocks we've encountered due to hardware limitation, design choices and other reasons. Finally, we evaluate results of our system compared to real world samples. We analyze system's performance

10.2478/jamsi-2019-0007

both from perspective of visual fidelity and responsiveness and give examples of our system's output.

2. RELATED WORK

The modeling nonlinear dynamics, and simulation of elastic rods is an active field from numerical and topological point of view in mathematical analysis [van der Heijden and Thompson 2000; Goyal et al. 2008]. It is impossible to survey the many works in this area, so for the state of the art in strand and hair simulation, refer to the survey by Ward et al. [Ward et al. 2007].

In graphics, Pai [K. 2002] applied a discretization of the Cosserat rod model to simulate a strand. Bertails et al. [Bertails et al. 2006] used a piecewise helical discretization to produce compelling animations of curly hair using few elements per strand.

In order to simulate a string, several traditional methods have been proposed, such as mass-spring systems [Hair 1991; Selle et al. 2008], rigid multibody serial chains [Hadap 2006], geometric approaches [Rivers and James 2007], and elastic energy-based models [Bertails et al. 2006; Spillmann and Teschner 2007; Bergou et al. 2008]. However, effects such as twisting, tearing, and flicking of a string are not considered all together in a single dynamic system.

In order to handle inextensible objects simulated by deformation models, different methods for stretch resistance have been proposed Provot [Institut and Provot 1996] and Bridson et al. [Bridson et al. 2005] proposed to constrain the length of springs to not stretch or compress beyond a given limit in the context of mass spring simulations. Some alternative ways of stabilizing stiff simulation using a global solver of a regular quad mesh were also proposed [Baraff and Witkin 1998; Müller et al. 2014; Goldenthal et al. 2007], including a constraint method based on impulse [Irving et al. 2007].

In the Finite Element Method (FEM) research area, Picinbono et al. [Picinbono et al. 2003] limit strain by adding an energy term to penalize strain in certain directions. Perez et al. [Perez et al. 2013] used Lagrange multipliers to constrain strain components isotropically. Hernandez at el. [Hernandez et al. 2013] improve this method to support anisotropic material. The linear system resulting from these methods becomes over-constrained and has to be solved as a least squares problem. In contrast, in the PBD framework, vertices are simply fixed by setting their inverse mass to zero.

These methods and many of their sequels are problematic in the case of excessive stretch or when discontinuous events such as rupture should occur.

3. PROPOSED STRING MODEL

Before we describe the methodology of our model, we would like to clarify few terms we often use in reference to out string model that might seem contradictory to how our model actually behaves.

Non-elongating - we regard elongation as permanent deformation of string represented by increased length of segments. What we truly mean when we use a term non-elongating is that this deformation does not happen arbitrarily but only as an exact deformation defined by stress-strain curve. As such it is also possible to create an instance of our string that never elongates.

Non-stretchable - similarly to non-elongation, we use it to define any stretching of our string as systematic and dependent on stress-strain curve. Only exception is created by parameter used in strain limiting that can allow for some stretching to decrease amount of necessary cycles needed when string is fixed in 2 or more particles.

Proposed model based on Position Based Dynamics approach (PBD) in this paper is a modification of [Selle et al. 2008; Rungjiratananon et al. 2012; Rungjiratananon et al. 2010] method used to simulate elastic objects and proven itself to be fast, stable and simple. It was newly modified to take advantage of the fact that string as structure is ultimately 1-dimensional object which allowed for further simplification. Proposed modifications are in collision constraint module that can be handled easily, by resolving the penetrations with regularly spreading the spheres along the segment followed by projection of particles to valid locations. We also propose a modification of strain limiting constraint that can handle the strings fixed in multiple locations contrary to single fixed side as is common for hair models [Rungjiratananon et al. 2012; Rungjiratananon et al. 2010]. We also propose the use of constraints and constraint forces, providing an efficient treatment for stiff twisting and non-stretchable mass spring dynamics model.

3.1. String

Underlying structure of the string model consists of *regions*, *n segments* and $N = (n+1)$ *particles*, where segment consists of two neighbouring particles and manages torsion, twist by an angle θ_i and non-elongating distance between the two particles. Segment is unique and every particle belongs to at most 2 segments. We can think

of it as a very rigid link. Region encompass several consecutive segments with their particles and is used for bending and stiffness simulation, whether in terms of twisting or bending. The bigger the region the faster the torsion propagates along the string and in addition to that the string is harder to bend.

To encapsulate entire functionality of our model we use a string element, see Fig. 1. It can contain multiple instances of itself to represent individual torn of pieces of string from the original. This recursive structure allows for easier handling of multiple instances and free user from micromanaging newly created strings. As such it is possible to efficiently change parameters or check for interactions with environment.

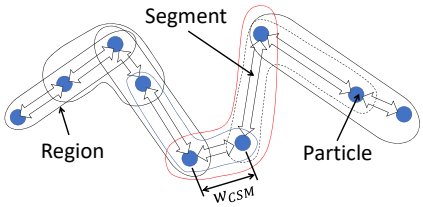

Fig. 1: String structure, Particle, Segment and Region. Region can undergo global transformations translation and rotation depended on w_{CSM} vector.

3.2. Particle

As smallest element we use particles to store information about position, velocity and forces that are applied on string. We made it as only element we allow user to directly interact with, as it is easiest to select through interface and offers way to find unique fit for other elements. User can apply twist and exert force. It is always center for sphere for collision detection.

3.3. Segment

Segments consist of two adjacent particles. They are used to handle twist, stress and strain in given part of string. Length of segment define the distance between particle and can change according to stress-strain curve. During collision detection, depending on pre-set parameter, multiple (even overlapping) spheres can be positions between its particles in regular intervals.

3.4. Region

Region is centered on the particle and several of its neighbours to both sides. It is defined by positive half-width and as such consists at least of 3 particles and 2 segments. It is mainly used to regulate stiffness of string. Wider it is, faster changes in twist spread and string bends less.

4. SIMULATION

We will now derive the new strain-based position constraints such as the traditional distance constraints or positional constraints. Let us assume we have system of N particles with positions $\mathbf{x}_i$, velocities $\mathbf{v}_i$ and inverse masses $\mathbf{w}_i$. For the sake of completeness, we briefly show the basic concepts of our simulation loop based on PBD:

Result: Update of $\mathbf{x}_i, \mathbf{v}_i$
initialize $\mathbf{x}_i, \mathbf{v}_i$;
while *simulation step* **do**
  $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta t \mathbf{f}_i$;
  $\mathbf{p}_i \leftarrow \mathbf{x}_i + \Delta t \mathbf{v}_i$;
  $\mathbf{p}_i \leftarrow solve(C(\mathbf{p}_1, \ldots, \mathbf{p}_N) = 0)$;
  $\mathbf{p}_i \leftarrow solve(CollisionConstraint(\mathbf{p}_1, \ldots, \mathbf{p}_N))$;
  $\mathbf{v}_i \leftarrow (\mathbf{p}_i - \mathbf{x}_i)\mathbf{x}_i + /\Delta t$;
  $\mathbf{v}_i \leftarrow Damping(\mathbf{v}_1, \ldots, \mathbf{v}_N)$;
  $\mathbf{x}_i \leftarrow \mathbf{p}_i$;
end

To achieve plausible behaviour of string we calculate forces, $\mathbf{f}_i$, affecting individual particles. Firstly, external forces are taken into account, such as gravity applied directly on string or any interacting object. Secondly, any direct effect caused by user through interface and lastly, inner forces caused by shape matching, twisting and flicking.

The positions are modified by a solver to meet a set of constraints $C(\mathbf{p}_1, \ldots, \mathbf{p}_N) = 0$ that is zero when the constraint is satisfied. The solver iterate multiple times over all constraints solving the system of non-linear equations. Local linearization of a single constraint function C by Taylor series results in the positional corrections $\Delta\mathbf{p}_i$ for point $\mathbf{p}_i$ computed as

$$\Delta\mathbf{p}_i = -skw_i \frac{\partial}{\partial \mathbf{p}_i} C(\mathbf{p}_1, \ldots, \mathbf{p}_N), \tag{1}$$

where k is the Lagrange multiplier

$$k = \frac{C(\mathbf{p}_1, \ldots, \mathbf{p}_N)}{\sum_j w_j |\frac{\partial}{\partial \mathbf{p}_i} C(\mathbf{p}_1, \ldots, \mathbf{p}_N)|^2} \tag{2}$$

and $k \in [0,1]$ is the stiffness parameter.

4.1. Damping

For damping, we use a general approach that specifically damps the relative velocities with respect to a constraint C as

$$\mathbf{v}_i \leftarrow \mathbf{v}_i - k \left(\sum_{j=1}^{N} \mathbf{v}_j^T \mathbf{n}_j \right) \mathbf{n}_i, \tag{3}$$

where $\mathbf{n}_i = \Delta\mathbf{p}_i / |\Delta\mathbf{p}_i|$. Note that the summation term cancels out when the mode itself does not change.

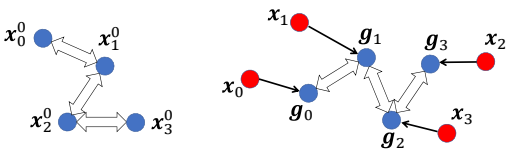

Fig. 2: Region translation and rotation. x_i^0 is the original position, x_i is the position updated by external and constraint forces and g_i is the goal position of particle i

4.2. Positional Constraint within Region

The particles are independently moved by external forces and constraints and then an optimal rigid transformation (i.e., rotation and translation) of each region is computed, refer Fig. 2. The rigid transformation of a particle position is called a goal position. Position of each particle is calculated as multiple goal positions weighted in the overlapping regions by particle per-region mass $\widetilde{m}_i = \frac{m_i}{N_r}$, where m_i is the mass of particle i and N_r is the total number of regions that the particle belongs to.

For the given set of original points ${x_i}^0$ and the set of deformed points x_i, at every step of simulation, for every region, rotational and translational matrix is calculated, which is than used to decide the goal positions, g_i, of particles of that region. Final goal position of particle is a weighted sum of goal positions in regions it belongs to, as

described in Eq. 4. Finally, each particle position is updated toward the goal position.

$$g_i = \sum_{r \in R_i} [R_r({x_i}^0 - {x_{cm,r}}^0) + x_{cm,r}]. \tag{4}$$

Change of region is defined by three variables: ${x_{cm,r}}^0$ is center of mass for region at the beginning of animation step from original points, $x_{cm,r}$ is center of mass for region after particles were moved, R_r is rotational matrix describing by what degree has region rotated.

Translational matrix can be calculated as simply as a difference between centers of mass, which are average of masses of particles. Approach for calculating rotation matrix can be found in [Müller et al. 2005]. Unlike in [Müller et al. 2005] our shape is 1-D string, therefore we can use Singular Value Decomposition (SVD) of weighted covariance matrix

$$A = \left(\sum_i \widetilde{m}_i \mathbf{p}_i \mathbf{q}_i^T \right) \left(\sum_i \widetilde{m}_i \mathbf{q}_i \mathbf{q}_i^T \right)^{-1}, \tag{5}$$

where

$$\begin{aligned} \mathbf{q}_i &= {x_i}^0 - x_{cm,r}^0, \\ \mathbf{p}_i &= x_i - x_{cm,r}, \end{aligned} \tag{6}$$

to obtain the rotation matrix R_r. That computes optimal rigid transformation between two sets of 3D points.

$$[U,S,V] = SVD(A), \tag{7}$$

two of the output matrices are used to calculate the rotation matrix

$$R = VU^T. \tag{8}$$

4.2.1. Goal position calculation. In order to implement the positional changes of particles, regions play the primary role. For every region we compute optimal rigid transformation (i.e. rotation and translation). These are represented by rotation and translation matrix Eq. 8. We keep two sets of particles:

—First set are particles in stable state that are result of last iteration of model and has been most likely displayed to user.

—Second set consists of particles affected by external forces and moved to new position without any constraint.

for all particles i **do**
  initialize original position $x_i^0, x_i \leftarrow x_i^0$
end for
loop
  for all particles i **do**
    $f_{i,ext} \leftarrow ComputeCollisionForce() + f_{gravity}$
  end for
  for all particles i **do**
    $v_i \leftarrow v_i + dt\frac{f_i}{m_i}$
    $x_i \leftarrow x_i + dtv_i$
  end for
  for all chain region R_i **do**
    $x_{cm} \leftarrow$ ComputeOptimalTranslation()
    $R \leftarrow$ ComputeOptimalRotation()
  end for
  for all particles i **do**
    $g_i \leftarrow$ ComputeGoalPosition()
  end for
  for all particles i **do**
    $g_i \leftarrow$ StrainLimiting()
    $x_i \leftarrow g_i$
    $v_i \leftarrow v_i + \frac{g_i - x_i}{dt}$
  end for
end loop

Algorithm 1: Chain shape matching algorithm.

After we compute new goal positions for every region, we calculate goal position for each particle which is an average of per region goal positions of given particle, see Eq. 4. At this point we can calculate force necessary to move particles of second set to goal position:

$$f_i(t+dt) = \frac{g_i(t) - x_i(t)}{dt}. \tag{9}$$

Because of possible collision it is might be necessary to run adjust position several times, so any changes are stored in second set. Only when everything finishes without collision, particles of first set are updated. It is also good practice of not discarding last stable state, in case a new frame of animation would need to be produced before we are finished. Following Alg. 1 should clarify the process and the order of calculations.

4.3. Twisting Constrain

According to the shape of the string, in the initial state, an initial twisting angle, θ_i^0, is assigned to each segment, $\mathbf{e}_i$, see Fig. 3. However, the point of twisting is always a particle and as such, the incurred twist is transferred to one or two adjacent segment without change in magnitude.

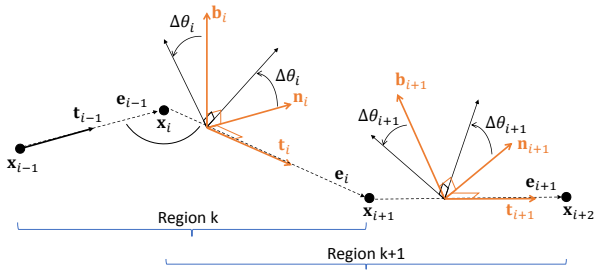

Fig. 3: Material frame the curvature binormal $\mathbf{b}_i$, the heading direction $\mathbf{e}_i$ and the normal vector $\mathbf{n}_i$. Twist propagation along the string depends on the size of region k while the amount of propagated twist depends on twisting stiffness β.

Model allows for a segment to be locked in, to simulate it being fixed and maintaining constant degree of twist. In such case, twisting effect can propagate to its neighbouring segments but the locked segments twist would remain constant until released and twist can't propagate past it.

Segments twisted by an external force, attempt to propagate twist towards the neighbouring segments in effort to even out. Size of neighbourhood influenced in every iteration depends on the size of region while the amount of propagated twist depends on predetermined twisting stiffness β.

The increment of the twisting angle of the segment is propagated to the next segments to minimize the elastic energy in the string. In other words, the string tries to minimize the twisting angles between each connected segment. Similarly, to position calculation, twist of a region, $\Delta\Theta_k^{region}$, is an weighted average of the twisting angle increment [Rungjiratananon et al. 2012], $\Delta\theta_i = \theta_i - \theta_i^0$, of the segments in the region k weighted by mass m_i, see Eq. 10. Where S_k is a set of segments within region k. Finally, the twist of segment, $\theta_i = \Delta\Theta_i^{segment}$, is a weighted sum of regional twists, containing the segment, see Eq. 11. Where R is the set of regions to which segment i belongs to.

$$\Delta\Theta_k^{region} = \frac{\sum_{j\in S_k}[m_j\Delta\theta_j]}{\sum_{j\in S_k}[m_j]}, \tag{10}$$

$$\Delta\Theta_i^{segment} = \sum_{k\in R}[\Delta\Theta_k^{region}]. \tag{11}$$

Relative twist of the segment exerts a force on a particle that is treated as external force and scales with twisting stiffness and relative twist of segment compared to its neighbours derived from elastic energy equation [Bergou et al. 2008] as follows:

$$\begin{aligned} \mathbf{f}_i^{twist} &= \frac{\beta}{L}(\theta_{i+1} - 2\theta_i + \theta_{i-1})\left(\frac{-\kappa\mathbf{b}_{i+1} - \kappa\mathbf{b}_{i-1}}{2l}\right), \\ \kappa\mathbf{b}_i &= 2\frac{\mathbf{e}_{i-1}\times\mathbf{e}_i}{|\mathbf{e}_{i-1}|\,|\mathbf{e}_i| + \mathbf{e}_{i-1}\cdot\mathbf{e}_i}, \end{aligned} \tag{12}$$

where $\kappa\mathbf{b}_i$ is the curvature binormal, $\mathbf{e}_i$ is the segment vector, l is the length of the segment, β is the twisting stiffness of the string, and L is the total length of the string.

Heading of segments is used to find the curvature binormal, which describes the correct heading of twisting force. Its heading does change with direction of twisting. Twisting force, $\mathbf{f}_i^{twist}$, is created when there is a difference in twist between 3 neighbouring segments. Afterwards, the calculated force is distributed among two particles of central segment. As the entire equation is build around the difference of twist and the heading of neighbouring segments, in case of reaching the end of string or fixed point, we create temporary segment with identical heading and twist as the segment with fixed point. User control of constant β allows a user define how much should the string knot.

4.4. Strain Limiting Constrain

To make sure the simulated string remains non-elongating, strain limiting algorithm [Selle et al. 2008] needs to be used. Instead of using the large Young's modulus, which can lead to numerical instability, position based constraints are often imposed so that the length of each segment does not exceed a certain threshold L_{max} [Baraff and Witkin 1998; Ďurikovič 2002; Goldenthal et al. 2007]. There are several heuristics approaches possible: Shrink each segment after shape matching; Adjust near tip particle in the propagation direction.

The basic idea is to change the particle position based on last moved direction to

keep the maximal length of segment, see Fig. 4. Although it works for a single fixed side and as such can finish iteratively in single pass of all particles within the string, this can be extended to 2 fixed ends or multiple fixed ends by subdividing the string and simulating the parts independently. Our strain limiting with 2 fixed ends can reach stable state in several iterations while switching between starting and ending points. Unfortunately, string with 2 fixed points can iterate without stable result, when string is too short to avoid the obstacle. In this case the only option is to allow the string to tear itself at point of deepest conflict.

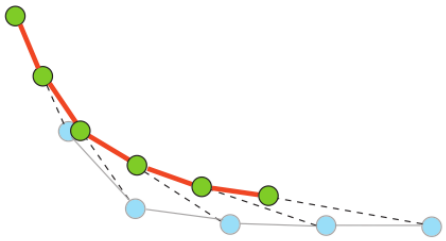

Fig. 4: Vectors of strain limiting. Blue dots are original particle positions and the green dots are new positions after strain limiting.

There can be a special case of string having no fixed particle. It could occur during tearing when free part of string would simply fall off. In such a situation, strain limiting automatically behaves as if the central particle is fixed so in such case strain limiting would be individually enacted on left and right part of string to limit movement of string from its center of mass.

We believe, that by adding collision detection directly into strain limiting we can significantly reduce number of iterations necessary. Important advantage is the cascading approach of strain limiting that moves onto next particle only after the previous was stabilized, see Fig. 5.

4.5. Collision Handling Constrain

To simulate collisions of structure with 1-D string, sphere-sphere collisions are ideal because of simplicity of calculation and easy implementation on GPU. Every particle is a center for collision sphere. Multiple spheres can be regularly spread out over the length of segment, their number and size is adjustable while spheres on the segment can overlap.

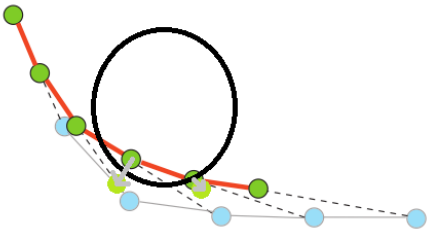

Fig. 5: Vectors of strain limiting with obstacle. Blue dots are original particle positions and the green dots are new positions after collision detection followed by strain limiting .

5. STRESS AND STRAIN HANDLING

In term of managing stress and strain of string we approach it as follows:

(1) After the position of string is calculated, we estimate stress applied on each segment between affected particle and nearest neighbor particles.

(2) Secondly, when every segment has calculated its strain value, we use stress-strain curves to calculate the stretching and elongation that given segment can be exposed to. It is also possible that strain would reach too high value (also defined by stress-strain curve) and it would tear itself into two parts.

(3) When new length of segments is finally calculated, Eq. 13, strain limiting algorithm is used to shift all particles that aren't affixed in such way to keep up the reasonable shape and correct heading of string.

$$L_{segment} = L_0 + L_{elongation} + L_{stretch}. \tag{13}$$

5.1. Stress-Strain curve

To get a proper idea of what stress-strain curves actually are, it is important to understand our perception of terms stress and strain.

—*Stress* is representative of amount of force exerted on given segment. Whether it is stretching force along the main axis of string or torsional forces from change in twist. Regardless of attributes of strings, if the same force is applied or relative twist of neighbouring segments is the same, stress will also be the same.

—*Strain* is the physical representation, or effect, of strain. It describes the stretching of segment and is dependant on specific attributes of string, mainly material.

Stress-strain curve is than our way of transforming abstract value of stress, independent of actual properties of string, into a specific effect. It describes how

much of the stretching is only temporary and will disappear the moment strain lessens and how big stretching will result in permanent elongation. The point between them is referred to as *yield point* and reaching it means segment can no longer stretch and any additional changes are part of permanent elongation.

Perhaps, for us the most important point of stress-strain curve is *rupture point* signalling greatest stress and strain the segment can sustain without breaking. Tensile stress-strain curve is measured by linear elongation and the torsional stress-strain curve is measured by torsion move. Data can be freely downloaded for common materials.

5.2. Stress estimation

Estimation of stress is processed after all forces, both external and internal are calculated but before strain limiting is applied. To avoid accidental tearing a delay is used and the excessive external force is distributed as stress only after it is verified that particle can no longer move in the direction of force. We estimate stress as proportional to distance between segment particles beyond the original length of segment. Because direction of the flicking force is calculated from vector of supplementary particle describing applied force and selected particle of string that is affected. As such we only need to store the magnitude of flicking force.

Strain is transformed into force and applied to particles only if:

—string tears,

—string is released.

In both cases, direction of force is determined by last particle in string (or particle that was held by user) is *head* or *tail*. Function initializing this force iterate over all particles until it reaches a *fixed* particle and always spreads from particle (or 2 in case of tearing).

5.3. Tearing and Flicking

As the model simulates strings that are unstretchable, at most times some of the force that is being applied to model should not be visually present. This excessive force is being distributed among the segments and if at any point exceeds the yield point, it permanently elongates the segments and decreases rupture point. If strain reaches rupture point, segment ceases to exist and strings separates. Because after every iteration the strain is applied as inverted force to particles, if the external force no longer influences the string (or it ruptures), it causes a flicking effect, see Fig. 6.

Tensile stress and strain values can be computed by estimating the tensions in the string. To derive the tensions, we also consider the particle positions computed without strain limiting.

We can handle the behaviour of flicking by using the tension. When an inextensible string is pulled and released or torn apart, the applied stress vanishes, but the tensile strain of the segment from the elongated length remains. The bouncing back force could be computed from an internal tensile stress translated from the tensile strain by referencing the stress-strain curve. However, with our tension estimation technique, we can directly use the tension as the bouncing back force.

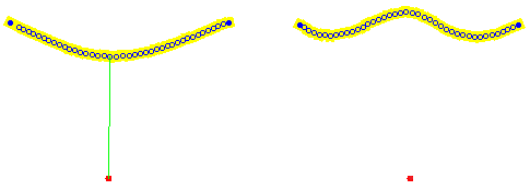

Fig. 6: Flicking of string with both ends fixed.

6. RESULTS

We tested our application on Alienware desktop X51 equipped with Intel Core4 i7-4770 with processor clock speed at 3,40GHz. It is fitted with NVidia GeForce GTX 960Ti OEM graphics card with 32 unified stream-processors (run both vertex and fragment shaders) and runs Windows Vista Service Pack 2 operating system. Any timing measurements performed relate to this setup.

6.1. Speed performance

Table I shows results of speed experiments with different number of particles and region widths. As a control sample, we use frames 20-50, because time of step also depends on size of the portion of string in relative motion. As at the begging sizable part of string is falling without restriction and after 50th frame some parts of string with narrow regions are already static.

As can be seen our from Tab. I, our expectation of region half-width on speed of our algorithm is significantly smaller than the effect of number of particles and could be limited even further with more efficient calls and data storage and retrieval.

Table I: Performance of our implementation.

Half-width of Regions	**Number of Particles**		
	11	***21***	***41***
1	0.68[a]	0.184	0.615
4	0.042[a]	0.219	0.686
8	0.001[a]	0.238	0.872

[a]Measured as [ms] per frame.

From our test we can infer that the only parameter definitely affecting speed of computation is number of particles and segment spheres used in collision detection. From practical perspective, because particles also act as spheres for collision detection, decreasing number of particles and increasing number of segment spheres would positively increase speed as segments spheres are used solely for collision detection. And therefore number of particles is main identifier of algorithmic speed.

6.2. Evaluation

We have created a model that simulates behavior of string with adjustable levels of stretching and elongation. It is capable of propagating twist and computing forces resulting from twisting and can lead to knotting.

When force is applied on particle to which particle can't respond because of constraints, stress-strain curve is used to calculate stretching and elongation. In case stress surpasses given threshold string ruptures. Afterwards flicking comes into effect where stress is converted to force and applied. Flicking effect may be also initiated if affecting force disappears.

Behaviour and movement of string is adjustable trough several parameters. Chief of all are:

—number of particles,

—half-width of regions,

—length of segments,

—stress-strain curve,

—number of segment spheres.

With exception of *number of particles* which influences general precision of model, each of these parameters influences specific behaviour of string.

Half-width of regions is used to define stiffness of string. It should however be always set with respect to *number of particles* as too narrow or wide regions are harmful to stable model.

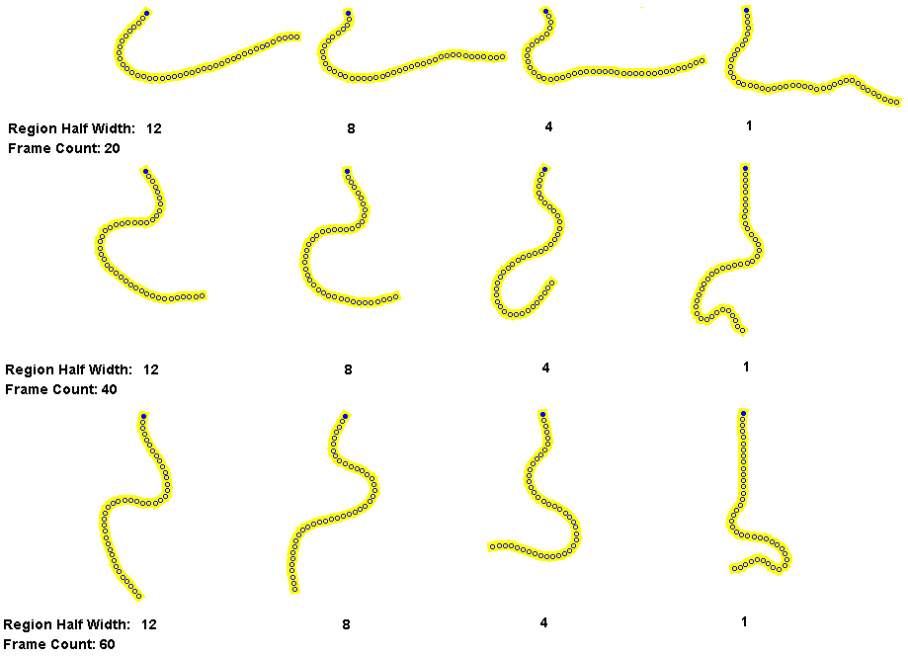

Fig. 7: Flicking of string with one end fixed. String stiffness is determined by half-width of regions as 12, 8, 4, and 1 from left in columns. The columns show the simulation frames at time 20, 40, and 60.

Segment length is fairly self-explanatory and determines total length of string. It is linked with *stress-strain curve* because stretch and elongation are represented as fraction of original *segment length*.

Finally, *number of segment spheres* is used to fine-tune precision of collision detection. Fewer spheres result in bumpier shape used for collision detection but increase speed of calculations.

7. CONCLUSION

This paper described a method for animation of string-like objects, being able to reproduce realistically effects like twisting, tearing and flicking. Ultimately, as our string is semi-realistic model, we will need to tune the attributes and we hope to be as closely with reality as we can. We have chosen several criteria to validate correctness of our model. Correctness of our model is measured by: Fidelity of motion, Reaction to outside forces, Twisting and forces generated by it, Tearing from twisting or straining, Flicking effect cause by release of strain.

REFERENCES

BARAFF, D. AND WITKIN, A. 1998. Large steps in cloth simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '98. ACM, New York, NY, USA, 43–54.

BERGOU, M., WARDETZKY, M., ROBINSON, S., AUDOLY, B., AND GRINSPUN, E. 2008. Discrete elastic rods. *ACM Trans. Graph. 27,* 3 (aug), 63:1–63:12.

BERTAILS, F., AUDOLY, B., CANI, M.-P., QUERLEUX, B., LEROY, F., AND LÉVÊQUE, J.-L. 2006. Super-helices for predicting the dynamics of natural hair. In *ACM SIGGRAPH 2006 Papers*. SIGGRAPH '06. ACM, New York, NY, USA, 1180–1187.

BRIDSON, R., MARINO, S., AND FEDKIW, R. 2005. Simulation of clothing with folds and wrinkles. In *ACM SIGGRAPH 2005 Courses*. SIGGRAPH '05. ACM, New York, NY, USA.

GOLDENTHAL, R., HARMON, D., FATTAL, R., BERCOVIER, M., AND GRINSPUN, E. 2007. Efficient simulation of inextensible cloth. *ACM Trans. Graph. 26,* 3 (jul), 49.

GOYAL, S., PERKINS, N., AND LEE, C. L. 2008. Non-linear dynamic intertwining of rods with self-contact. *International Journal of Non-Linear Mechanics 43,* 1, 65 – 73.

HADAP, S. 2006. Oriented strands: Dynamics of stiff multi-body system. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA '06. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 91–100.

HAIR, D. C. 1991. Legalese: A legal argumentation tool. *SIGCHI Bull. 23,* 1 (Jan.), 71–74.

HERNANDEZ, F., CIRIO, G., PEREZ, A. G., AND OTADUY, M. A. 2013. Anisotropic strain limiting. In *Proc. of Congreso Espanol de Informatica Grafica*.

INSTITUT, X. P. AND PROVOT, X. 1996. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *In Graphics Interface*. 147–154.

IRVING, G., SCHROEDER, C., AND FEDKIW, R. 2007. Volume conserving finite element simulations of deformable models. *ACM Trans. Graph. 26,* 3 (jul).

K., P. D. 2002. Strands: Interactive simulation of thin solids using cosserat models. *Computer Graphics Forum 21,* 3, 347–352.

MÜLLER, M., CHENTANEZ, N., KIM, T.-Y., AND MACKLIN, M. 2014. Strain based dynamics. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA '14. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 149–157.

MÜLLER, M., HEIDELBERGER, B., HENNIX, M., AND RATCLIFF, J. 2007. Position based dynamics. *J. Vis. Comun. Image Represent. 18,* 2 (apr), 109–118.

MÜLLER, M., HEIDELBERGER, B., TESCHNER, M., AND GROSS, M. 2005. Meshless deformations based on shape matching. In *ACM SIGGRAPH 2005 Papers*. SIGGRAPH '05. ACM, New York, NY, USA, 471–478.

PEREZ, A. G., CIRIO, G., HERNANDEZ, F., GARRE, C., AND OTADUY, M. A. 2013. Strain limiting for soft finger contact simulation. In *2013 World Haptics Conference (WHC)*. 79–84.

PICINBONO, G., DELINGETTE, H., AND AYACHE, N. 2003. Non-linear anisotropic elasticity for real-time surgery simulation. *Graph. Models 65,* 5 (sep), 305–321.

RIVERS, A. R. AND JAMES, D. L. 2007. Fastlsm: Fast lattice shape matching for robust real-time deformation. In *ACM SIGGRAPH 2007 Papers*. SIGGRAPH '07. ACM, New York, NY, USA.

RUNGJIRATANANON, W., KANAMORI, Y., METAAPHANON, N., BANDO, Y., CHEN, B.-Y., AND NISHITA, T. 2012. Animating strings with twisting, tearing and flicking effects. *COMPUTER ANIMATION AND VIRTUAL WORLDS 18,* 2 (mar), 113–124.

RUNGJIRATANANON, W., KANAMORI, Y., AND NISHITA, T. 2010. Elastic rod simulation by chain shape matching with twisting effect. In *ACM SIGGRAPH ASIA 2010 Sketches*. SA '10. ACM, New York, NY, USA, 27:1–27:2.

SELLE, A., LENTINE, M., AND FEDKIW, R. 2008. A mass spring model for hair simulation. *ACM Transactions on Graphics 27,* 3 (aug), 64:1–64:11.

SPILLMANN, J. AND TESCHNER, M. 2007. Corde: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA '07. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 63–72.

VAN DER HEIJDEN, G. AND THOMPSON, J. 2000. Helical and localised buckling in twisted rods: A unified analysis of the symmetric case. *Nonlinear Dynamics 21,* 1 (Jan), 71–99.

ĎURIKOVIČ, R. 2002. Animation of soap bubble dynamics, cluster formation and collision. *Computer Graphics Forum 20,* 3, 67–76.

WARD, K., BERTAILS, F., Y. KIM, T., MARSCHNER, S. R., P. CANI, M., AND LIN, M. C. 2007. A survey on hair modeling: Styling, simulation, and rendering. *IEEE Transactions on Visualization and Computer Graphics 13,* 2 (March), 213–234.

Roman Ďurikovič
Faculty of Mathematics, Physics and Informatics,
Comenius University,
842 48 Bratislava, Slovak Republic
email: roman.durikovic@fmph.uniba.sk

Erich Siebenstich
Mathematics, Physics and Informatics,
Comenius University,
842 48 Bratislava, Slovak Republic

Statistical learning for recommending (robust) nonlinear regression methods

J. KALINA AND J. TICHAVSKÝ

Abstract

We are interested in comparing the performance of various nonlinear estimators of parameters of the standard nonlinear regression model. While the standard nonlinear least squares estimator is vulnerable to the presence of outlying measurements in the data, there exist several robust alternatives. However, it is not clear which estimator should be used for a given dataset and this question remains extremely difficult (or perhaps infeasible) to be answered theoretically. Metalearning represents a computationally intensive methodology for optimal selection of algorithms (or methods) and is used here to predict the most suitable nonlinear estimator for a particular dataset. The classification rule is learned over a training database of 24 publicly available datasets. The results of the primary learning give an interesting argument in favor of the nonlinear least weighted squares estimator, which turns out to be the most suitable one for the majority of datasets. The subsequent metalearning reveals that tests of normality and heteroscedasticity play a crucial role in finding the most suitable nonlinear estimator.

Mathematics Subject Classification 2010: 68T05, 62G35, 62J02, 68-04
General Terms: Statistical learning, Nonlinear regression, Robustness, Heteroscedasticity
Keywords: nonlinear least weighted squares, optimal method selection, optimization, computations

1. INTRODUCTION

The aim of regression modeling is to explain a continuous response variable based on one or more independent variables (regressors), where the latter may be continuous and/or discrete. This also allows to predict values of the response for individual values of regressors. The nonlinear regression model can be described as an important category of parametric regression models and will be considered in this paper.

The most traditional estimation tool for nonlinear regression, i.e. the nonlinear least squares estimator, is well known to be too vulnerable to the presence of outlying measurements (outliers) in the data [Seber and Wild 2003]. Therefore, robust estimation techniques for the nonlinear regression have been proposed and have become established e.g.in econometric applications [Riazoshams et al. 2010; Baldauf and Silva 2012]. The concept of breakdown point has become a fundamental

This work is supported by the Czech Research Council projects 17-07384S and 19-05704SS.

10.2478/jamsi-2019-0008

measure of (global) robustness suitable for nonlinear regression estimators [Stromberg and Ruppert 1992]. Nevertheless, it remains unknown to answer the question, which estimator should be used for a given dataset; this is perhaps impossible to be found theoretically across all possible datasets.

Metalearning is a methodology allowing to recommend the most suitable algorithm (or method) for a given dataset, based on information learned over a training database of datasets. It extracts (prior) knowledge from datasets applicable to a new dataset and therefore has become quite popular in recent computer science, optimization or data mining. Metalearning also starts to penetrate to economic applications. To give specific examples of metalearning in computational economics, metalearning was performed to compare economic genetic algorithms (especially with a modified mutation operator suitable for consumption models) [Riechmann 2001] or to compare various optimization tools (mainly hybrid algorithms) for business, economics and finance [Vasant 2012].

Metalearning has been recommended in the seminal paper [Smith-Miles et al. 2014] especially for those domains, in which a theoretical knowledge is too difficult to acquire. Finding the best nonlinear regression estimator is exactly a task, for which it remains very tedious (or perhaps infeasible) to derive any practical solution rigorously [Stromberg and Ruppert 1992]. Thus, metalearning represents a computational (however computationally intensive) approach able to offer some practical advice in this respect. For a discussion of advantages and limitations of metalearning, we refer to [Smith-Miles et al. 2014], but we hold the opinion that aconscientiously critical evaluation of metalearning is still missing. Based on our experience, metalearning in a habitual form is vulnerable to instability and sensitivity to data contamination by noise, outlying values (outliers), or presence of redundant variables.

In this paper, metalearning will be used in the context of nonlinear regression with the aim to predict the best method for particular datasets not contained in the training database. Section 2 recalls various estimators for the nonlinear regression model. Sections 3 presents a numerical example revealing the severe bias of the nonlinear least trimmed squares under heteroscedasticity. Section 4 describes our metalearning study, the results of which are presented in Section 5. Finally, Section 6 concludes the paper.

2. ROBUST ESTIMATION IN NONLINEAR REGRESSION

Let us consider the standard nonlinear regression model

$$Y_i = f(\beta_1 X_{i1}, \ldots, \beta_p X_{ip}) + e_i, \quad i = 1, \ldots, n, \tag{1}$$

where f is a given continuous nonlinear function, $Y_1, \ldots, Y_n$ is a continuous response and $(X_{1j}, \ldots, X_{nj})^T$ is the j-th regressor for $j = 1, \ldots, p$. We use the notation $e_1, \ldots, e_n$ for random errors and X for the matrix with elements X_{ij}, where $i = 1, \ldots, n$ and $j = 1, \ldots, p$. The classical estimator, which is the nonlinear least squares (NLS) estimator of β, is vulnerable to the presence of outliers in the data [Maronna et al. 2006; Fasano et al. 2012]. Therefore, we recall several of its potential robust alternatives in this section. All of them will be also used later in our computations.

2.1. Nonlinear least trimmed squares

The nonlinear least trimmed squares (NLTS) estimator represents one of robust methods with a (possibly) high breakdown point [Stromberg and Ruppert 1992] and also an extension of the popular least trimmed squares from linear regression [Rousseeuw 1983; Hampel et al. 1986; Mount et al. 2014]. We denote by $\mathbb{R}$ the set of real numbers and the residual corresponding to the i-th observation will be denoted as

$$u_i(b) = Y_i - f(b_1 X_{i1}, \ldots, b_p X_{ip}), \quad i = 1, \ldots, n, \tag{2}$$

for any (fixed) $b = (b_1, \ldots, b_p)^T \in \mathbb{R}^p$. Let us arrange squared values in ascending order as

$$u^2_{(1)}(b) \leq \cdots \leq u^2_{(n)}(b). \tag{3}$$

The user must specify a suitable value of the trimming constant h ($n/2 \leq h \leq n$). Then, the NLTS estimator b_{NLTS} of β is obtained as

$$b_{NLTS} = \underset{b \in \mathbb{R}^p}{\arg\min} \sum_{i=1}^{h} u^2_{(i)}(b). \tag{4}$$

While the choice of h should reflect the percentage of contaminated data, one usually takes 25 % of outliers in applications and thus h is commonly chosen as the integer part of $3n/4$. An alternative approach is to select h by repeating the evaluation of the NLTS with an increasing h (starting with $n/2$) up to the moment when the estimates of the model (especially the corresponding estimate of σ^2) abruptly and greatly change.

The computation of the NLTS estimator requires minimization over all $\binom{n}{h}$ h-subsets of $\{1,2,\ldots,n\}$, thus has combinatorial complexity and, depending on the relation between n and h, can become infeasible already for rather small numbers of observations n. Therefore, it is more convenient to use the FAST-LTS algorithm [Rousseeuw and van Driessen 2006], which approximates the LTS estimator in a computationally attractive way; it has become a standard tool for the LTS allowing very large sample sizes n.

2.2. Nonlinear least weighted squares

The nonlinear least weighted squares (NLWS) estimator represents an extension of the least weighted squares estimator from the linear regression and at the same time a weighted analogy of the NLTS estimator [Víšek 2011; Kalina 2014]. Let us assume the magnitudes $w_1,\ldots,w_n$ of nonnegative weights to be given. The NLWS estimator of the parameters in (1) is defined as

$$b_{NLWS} = \arg\min_{b\in\mathbb{R}^p} \sum_{i=1}^{n} w_i u^2_{(i)}(b), \tag{5}$$

where the argument of the minimum is computed over all possible values of $b = (b_1,\ldots,b_p)^T$ and squared residuals are arranged as in (3).

The choice of weights clearly has a determining influence on properties of the estimator; if namely one assigns zero weights to outlying observations, then the estimator is ensured to be highly robust in terms of the breakdown point. The main reason for such robustness of the NLWS estimator is the construction of the estimator itself, just like for the LWS estimator in the linear regression. Various weighting schemes will be described in Section 2.4.

An approximate algorithm for the optimization task (5) can be obtained as a (weighted) adaptation of the NLTS algorithm [Rousseeuw and van Driessen 2006], which is presented here as Algorithm 1. To avoid any confusion, we denote coordinates of some vectors with the upper index there, e.g. the weights are denoted as $w = (w^1,\ldots,w^n)^T$. It also exploits the weighted NLS estimator, which is a well described weighted analogue of the NLS [Seber and Wild 2003]. Concerning the choice of parameters, we use $J = 10000$ and $\varepsilon = 0.0001$ in all computations in this paper.

Algorithm 1 Nonlinear least weighted squares (NLWS) estimator of β in (1)

Input: $X_1,\ldots,X_n$, where $X_i \in \mathbb{R}^p$ for each $i = 1,\ldots,p$
Input: $Y_1,\ldots,Y_n$
Input: Nonlinear function f
Input: $J > 0$
Input: $\varepsilon > 0$
Input: Magnitudes of weights $w^1 \geq \cdots \geq w^n$
Output: Optimal permutation of $w^1,\ldots,w^n$ denoted as $\tilde{w}$

for $j = 1$ to J **do**
$m_{j0} := +\infty$
Select randomly p points
Estimate β by $b_{j1} = (b^1_{j1},\ldots,b^p_{j1})^T$ obtained as a NLS estimator using only the selected p points
$k := 1$
repeat
$u_i(b_{jk}) := Y_i - f(b^1_{ik}X_{i1},\ldots,b^p_{ik}X_{ip})$, $i = 1,\ldots,n$
Assign weights $w_{j,k+1} = (w^1_{j,k+1},\ldots,w^n_{j,k+1})^T$ to individual observations based on (3)
Estimate β by $b_{j,k+1} = (b^1_{i,k+1},\ldots,b^p_{i,k+1})^T$ obtained as a weighted NLS with weights $w_{j,k+1}$
$m_{j,k+1} := \sum_{i=1}^n w^i_{j,k+1}u^2_i(b_{j,k+1})$
$k := k+1$
until $m_{jk} > m_{j,k-1} + \varepsilon$
$\tilde{w}_j = (\tilde{w}^1_j,\ldots,\tilde{w}^n_j)^T := w_{j,k-1}$
$\tilde{b}_j = (\tilde{b}^1_j,\ldots,\tilde{b}^n_j)^T := b_{j,k-1}$
end for

$$j^* := \underset{j=1,\ldots,J}{\arg\min} \sum_{i=1}^n \tilde{w}^i_j u^2_i(\tilde{b}_j) \tag{6}$$

b_{NLWS} := weighted NLS estimate with weights

$$\tilde{w} = (\tilde{w}^1,\ldots,\tilde{w}^n)^T := (\tilde{w}^1_{j^*},\ldots,\tilde{w}^n_{j^*})^T$$

2.3. Nonlinear regression median

Regression quantiles represent a natural generalization of sample quantiles to the linear regression model. The estimator depends on a parameter α in the interval (0,1), which corresponds to dividing the disturbances to $\alpha \cdot 100$ % values below the regression quantile and the remaining $(1-\alpha)\cdot 100$ % values above the regression quantile. In general, regression quantiles represent an important tool of regression methodology, which is popular in economic applications. A natural extension of

Table I. Results of various nonlinear estimators in the example of Section 3.

	Estimates of		
Estimator	β_1	β_2	β_3
1 (NLS)	−4.5	1.1	−1.0
2 (NRM)	−4.7	1.5	−1.0
3 (NLTS)	−4.5	0.9	−1.0
4 (NLWS)	−4.6	1.1	−0.9
5 (NLWS)	−4.5	1.0	−1.0
6 (NLWS)	−4.6	0.9	−1.0
7 (NLWS)	−4.5	1.0	−0.9

regression quantiles to nonlinear regression is also available [Koenker and Park 1996], while the most important special case remains to be the nonlinear regression median (NRM) with $\alpha = 1/2$.

2.4. Estimators used in the computation

In the example of Section 3 as well as in the metalearning study (see Section 4), we use the following seven available estimators, which will be denoted as estimators $1,\dots,7$:

(1) Nonlinear least squares (NLS).

(2) Nonlinear regression median (NRM).

(3) NLTS with h equal to $\lfloor \frac{3n}{4} \rfloor$, where $\lfloor x \rfloor$ denotes the integer part of $x \in \mathbb{R}$.

(4) NLWS with data-dependent adaptive weights of [Čížek 2011].

(5) NLWS with linear weights

$$w_i = \frac{2(n+1-i)}{n(n+1)}, \quad i = 1,\dots,n. \tag{7}$$

(6) NLWS with trimmed linear weights

$$w_i = \frac{h-i+1}{h} I[i \le h], \quad i = 1,\dots,n, \tag{8}$$

where $I[.]$ denotes an indicator function and h equals again to $\lfloor \frac{3n}{4} \rfloor$.

(7) NLWS with weights generated by the (strictly decreasing) logistic function

$$w_i = \left(1 + \exp\left\{\frac{i-n-1}{n}\right\}\right)^{-1}, \quad i = 1,\dots,n. \tag{9}$$

The habitual standardization of weights to $\sum_{i=1}^{n} w_i = 1$ is required in all cases.

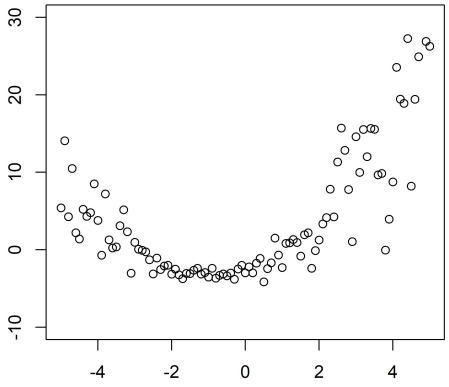

Fig. 1. Heteroscedastic data from the example of Section 3.

3. EXAMPLE: HETEROSCEDASTIC DATA

The effect of heteroscedasticity of errors on classical regression estimation, i.e. deterioration of the estimates or even their complete malfunction, seems to be still underestimated in statistical and econometric literature [Wooldridge 2001; Kalina et al. 2019]. The following example illustrates the performance various nonlinear estimators on artificial (randomly generated) heteroscedastic data shown in Figure 1. We generate the data from the model

$$Y_i = \beta_1 + \beta_2 (X_i - \beta_3)^2 + e_i, \quad i = 1, \ldots, n, \tag{10}$$

with $\beta_1 = -4.6$, $\beta_2 = 1$ and $\beta_3 = -1$. Estimates of β in this nonlinear model, i.e. a nonlinear function of a single regressor $X_1, \ldots, X_n$, are shown in Table I. The NLTS ignores the heteroscedastic character of the data completely and chooses an unsuitable subset of data to play the role of the reliable majority of the data points. The NLS as well as NLWS estimates are able to find a more suitable regression estimate also for observations with a value of the regressors exceeding 2. Their residuals are much more symmetric around 0 compared to NLTS residuals.

It is mainly β_2 which may be affected by heteroscedasticity. The NLTS is a clear loser here as it ignores data generated with smallest (i.e. negative) residuals and considers them to by outlying. On the other hand, the NLWS estimator seems to yield a reasonable result independently on the choice of weights so the particular choice of weights for the NLWS seems to be rather unimportant.

4. DESCRIPTION OF THE METALEARNING STUDY

Metalearning is a computational (machine learning) approach allowing to exploit information from previously observed datasets and to extend them to new datasets [Brazdil et al. 2009; Smith-Miles et al. 2014]. While it has become popular in optimization and to some extent in machine learning, it can be exploited for recommending the optimal method also in the context of statistical estimation. In this paper, we describe our metalearning study with the aim to compare various nonlinear regression estimators. The study allows us also to detect the most relevant criteria for determining the most suitable weights for the NLWS.

The primary learning task is to fit various nonlinear regression estimators for each of the given datasets. The best estimator is found using a specified characteristic of a goodness of fit. The subsequent metalearning part has the aim to learn a classification rule allowing to predict the best regression method for a new dataset not present in the training database. Its input data are only selected features of individual datasets together with the result of the primary learning, which typically has the form of the index of the best method for each of the training datasets. In general, the user of metalearning must specify a list of essential components (parameters). We will now describe our choices for the primary learning as well as the subsequent metalearning part of the task.

4.1. Primary learning

We use 24 real datasets previously used in [Kalina and Peštová 2017]. In each of them, we consider the model

$$Y_i = \beta_0 + \sum_{j=1}^{p} \beta_j X_{ij} + \sum_{j=1}^{p} \beta_{p+j}(X_{ij} - \bar{X}_j)^2 + e_i, \quad i = 1, \ldots, n, \tag{11}$$

which is a nonlinear model with the total number of $2p+1$ regressors (i.e. nonlinear as a function of the original p regressors) and the variables are centered using the mean of the j-th variable $\bar{X}_j$ (for $j = 1, \ldots, p$) for the sake of numerical stability.

We use the most standard choice of the prediction measure, which is the (prediction) mean square error (MSE). We find the best method for each dataset using MSE in a standard leave-one-out cross validation, which represents a popular attempt for an independent validation [Hastie et al. 2008].

4.2. Metalearning

The output of Section 4.1 is used in the form of the factor variable (index, indicator) of the best method for each of the datasets together with a list of features computed for each dataset. In contrary to the linear case of [Kalina and Peštová 2017], we need to consider features for the metalearning more carefully now in the nonlinear model. For example, there is no analogue of the coefficient of determination in the nonlinear model. Other features must be evaluated for the NLS fit. Thus we came to selecting the following set of 9 features.

(1) The number of observations n.

(2) The number of regressors p (excluding the intercept).

(3) The ratio n/p.

(4) Condition number of the matrix $(X^T X)^{-1}$.

(5) Normality of residuals of the NLS, evaluated as the p-value of the Shapiro-Wilk test.

(6) Skewness of residuals of the NLS.

(7) Kurtosis of residuals of the NLS.

(8) Heteroscedasticity of residuals of the NLS evaluated as the p-value of the White's test.

(9) Test of linearity evaluated as the p-value of the test of $H_0: \beta_{p+1} = \cdots = \beta_{2p} = 0$ in (11) based on the NLS estimates.

For the subsequent metalearning task, which is a task of classification to 7 groups (i.e. finding the best among the 7 estimators of Section 2.4), we exploit various classification methods including support vector machines (SVM), k-nearest neighbors, or a classification tree. We use also several other less known methods including a regularized version of linear discriminant analysis (LDA) denoted as SCRDA of [Guo et al. 2007] or a robust version of LDA denoted as linear MWCD classification [Kalina 2012].

5. RESULTS

We used the R software for all the computations including necessary libraries (robustbase, quantreg, rda) for specific tasks (robust estimation, classification). Using MSE as a measure of prediction performance, a leave-one-out cross validation is used as a standard attempt for an independent validation. Table II shows the best

method of the primary learning for each of the datasets. There, the notation $1,\ldots,7$ according to list of methods in Section 2.4 is used. The ranks are presented here, so that method 4 (NLWS with data-dependent weights) is the best method for the Aircraft dataset, method 3 (NLTS) performs as the second best on this dataset etc. On the whole, the NLWS is the best estimator based on Table II, but no weighting scheme seems to be superior to the remaining choices.

Within the subsequent metalearning task, we computed various classifiers, while default settings of parameters was used for those computed using the R software. As the results presented in Table III indicate, the SVM classifier turns out to have the best performance. Because there are as many as 7 classes, the overall prediction ability is not very high (although not much lower compared to e.g. [Kalina and Peštová 2017]. The most useful criteria for the choice of weights for the NLWS turn out to be heteroscedasticity and normality of the NLS residuals, which correspond to our intuition, because their heavy violation requires a robust approach, i.e. a highly robust estimation in terms of the breakdown point.

6. CONCLUSIONS

In this paper, we work with various estimators of parameters in the standard nonlinear regression model. We are especially interested in the NLWS estimator, which is a natural extension of the LWS estimator [Víšek 2011] to the nonlinear regression model. A novel algorithm for the NLWS is proposed in this paper. At the same time, an example with heteroscedastic data is presented in Section 3, which brings an argument in favor of the novel NLWS estimator compared to the previously investigated NLTS estimator. There, the NLWS estimator (basically with any weighting scheme) seems more suitable but at the cost of higher computational demands.

A metalearning study presented in this paper has the aim to construct a classification rule allowing to predict the most suitable nonlinear regression estimator for a particular dataset. For this purpose, we work with 24 real datasets with an economic background and performed a standard metalearning procedure. The NLWS seems to yield the best result for the majority of the datasets, while no weighting scheme is uniformly optimal over all datasets. The NLTS and the nonlinear regression median have a weaker performance due to a low efficiency. Thus, the concept of efficiency (and not only the robustness) seems to play a prominent role in the analysis of real data and it is the newly investigated NLWS estimator, which

Table II. Results of primary learning in a leave-one-out cross-validation study. Ranks corresponding to the mean prediction errors of various nonlinear regression estimators for each of the 24 datasets.

		1	2	3	4	5	6	7
	Dataset	NLS	NRM	NLTS	NLWS	NLWS	NLWS	NLWS
1	Aircraft	7	6	2	1	5	4	3
2	Ammonia	6	4	2	3	5	6	1
3	Auto MPG	3	5	1	4	7	6	2
4	Cirrhosis	2	1	5	6	3	7	4
5	Coleman	1	2	3	7	6	5	4
6	Delivery	7	3	6	1	2	4	5
7	Education	1	2	3	4	7	6	5
8	Electricity	2	1	4	5	3	7	6
9	Employment	4	2	3	7	1	5	6
10	Furniture 1	3	1	5	7	2	4	6
11	Furniture 2	7	4	3	6	1	2	5
12	GDP growth	2	1	7	4	5	6	3
13	Houseprices	6	4	1	3	5	7	2
14	Housing	3	1	2	6	7	4	5
15	Imports	5	2	4	6	7	1	3
16	Kootenay	1	2	5	3	7	6	4
17	Livestock	7	6	2	3	5	1	4
18	Machine	2	7	5	4	3	6	1
19	Murders	3	2	7	4	6	5	1
20	NOx	4	5	1	3	7	6	2
21	Octane	2	1	5	4	6	7	3
22	Pasture	7	2	6	3	5	1	4
23	Pension	7	4	3	1	2	5	6
24	Petrol	2	3	7	1	5	6	4

Table III. Results of metalearning evaluated as the ratio of correctly classified cases in a leave-one-out cross validation study.

Classification method	Prediction accuracy
Classification tree	0.33
k-nearest neighbor ($k = 3$)	0.46
LDA	0.54
SCRDA	0.54
Linear MWCD-classification	0.54
Multilayer perceptron	0.50
Logistic regression	0.50
SVM (linear)	0.54
SVM (Gaussian kernel)	0.58

seems able to combine robustness with efficiency very well.

ACKNOWLEDGEMENTS

The authors would like to thank Barbora Peštová for acquiring the datasets and the reviewer for constructive advice.

REFERENCES

BALDAUF, M. AND SILVA, J. 2012. On the use of robust regression in econometrics. *Economic Letters 114*, 124–127.

BRAZDIL, P., GIRAUD-CARRIER, C., SOARES, C., AND VILALTA, E. 2009. *Metalearning: Applications to data mining*. Springer, Berlin.

ČÍŽEK, P. 2011. Semiparametrically weighted robust estimation of regression models. *Computational Statistics & Data Analysis 55*, 774–788.

FASANO, V., MARONNA, R., SUED, M., AND YOHAI, V. 2012. Continuity and differentiability of regression M-functionals. *Bernoulli 18*, 1284–1309.

GUO, Y., HASTIE, T., AND TIBSHIRANI, R. 2007. Regularized discriminant analysis and its application in microarrays. *Biostatistics 8*, 86–100.

HAMPEL, F. R., RONCHETTI, E. M., ROUSSEEUW, P. J., AND STAHEL, W. A. 1986. *Robust statistics—The approach based on influence functions*. Wiley, New York.

HASTIE, T., TIBSHIRANI, R., AND FRIEDMAN, J. 2008. *The elements of statistical learning*, 2nd ed. Springer, New York.

KALINA, J. 2012. Highly robust statistical methods in medical image analysis. *Biocybernetics and Biomedical Engineering 32*, 3–16.

KALINA, J. 2014. On robust information extraction from high-dimensional data. *Serbian Journal of Management 9*, 131–144.

KALINA, J. AND PEŠTOVÁ, B. 2017. Robust regression estimators: A comparison of prediction performance. In *35th International Conference Mathematical Methods in Economics MME 2017*. University of Hradec Králové, Hradec Králové, 307–312.

KALINA, J., VAŠANIČOVÁ, P., AND LITAVCOVÁ, E. 2019. Regression quantiles under heteroscedasticity and multicollinearity: Analysis of travel and tourism competitiveness. *Ekonomický časopis/Journal of Economics 67*, 69–85.

KOENKER, R. AND PARK, B. 1996. An interior point algorithm for nonlinear quantile regression. *Journal of Econometrics 71*, 265–283.

MARONNA, R., MARTIN, D., AND YOHAI, V. 2006. *Robust statistics: Theory and methods*. Wiley, New York.

MOUNT, D., NETANYAHU, N., PIATKO, C., SILVERMAN, R., AND WU, A. 2014. On the least trimmed squares estimator. *Algorithmica 69*, 148–183.

RIAZOSHAMS, H., MIDI, H., AND SHARIPOV, O. 2010. The performance of robust two-stage estimator in nonlinear regression with autocorrelated error. *Communications in Statistics – Simulation and Computation 39*, 1251–1268.

RIECHMANN, T. 2001. *Learning in economics. Analysis and application of genetic algorithms.* Springer, Berlin.

ROUSSEEUW, P. J. 1983. Multivariate estimation with high breakdown point. In *Proceedings of the 4th Pannonian Symposium on Mathematical Statistics*, W. Grossmann, G. Pflug, I. Vincze, and W. Wertz, Eds. D. Reidel, Dordrecht, 283–297.

ROUSSEEUW, P. J. AND VAN DRIESSEN, K. 2006. Computing lts regression for large datasets. *Data Mining and Knowledge Discovery 12*, 29–45.

SEBER, G. AND WILD, C. 2003. *Nonlinear regression*. Wiley, New York.

SMITH-MILES, K., BAATAR, D., WREFORD, B., AND LEWIS, R. 2014. Towards objective measures of algorithm performance across instance space. *Computers & Operations Research 45*, 12–24.

STROMBERG, A. AND RUPPERT, D. 1992. Breakdown in nonlinear regression. *Journal of the American Statistical Association 87*, 991–997.

VASANT, P. M. 2012. *Meta-heuristics optimization algorithms in engineering, business, economics, and finance*. IGI Global, Hershey.

VÍŠEK, J. Á. 2011. Consistency of the least weighted squares under heteroscedasticity. *Kybernetika 47*, 179–206.

WOOLDRIDGE, J. M. 2001. *Econometric analysis of cross section and panel data*. MIT Press, Cambridge.

Jan Kalina
The Czech Academy of Sciences,
Institute of Computer Science,
Pod Vodárenskou věží 2, CZ-182 07 Praha 8, Czech Republic
&
The Czech Academy of Sciences,
Institute of Information Theory and Automation
Pod Vodárenskou věží 4, CZ-182 00 Praha 8, Czech Republic
e-mail: kalina@cs.cas.cz

Jan Tichavský
The Czech Academy of Sciences, Institute of Computer Science
Pod Vodárenskou věží 2
CZ-182 07 Praha 8
Czech Republic
e-mail: tichavsky@cs.cas.cz

Fractional calculus pertaining to multivariable I-function defined by Prathima

D. KUMAR AND F. Y. AYANT

Abstract

In this paper, we study a pair of unified and extended fractional integral operator involving the multivariable I-functions and general class of multivariable polynomials. Here, we use Mellin transforms to obtain our main results. Certain properties of these operators concerning to their Mellin-transforms have been investigated. On account of the general nature of the functions involved herein, a large number of known (may be new also) fractional integral operators involved simpler functions can be obtained . We will also quote the particular case of the multivariable H-function.

Mathematics Subject Classification 2010: Primary 33C45, 33C60; Secondary 26D20.
Keywords: Multivariable I-function; I-function; Fractional integral operators; General class of multivariable polynomials; Mellin transform; Multivariable H-function.

1. INTRODUCTION AND PRELIMINARIES

Fractional calculus is a field of applied mathematics that deals with derivatives and integrals of arbitrary orders. Recently, it has turned out many phenomena in physics, mechanics, chemistry, biology and other sciences can be described very successfully by models using mathematical tools by models using mathematical tools from fractional calculus. Chaurasia and Srivastava [4], Choi et al. [5], Daiya et al. [6], Kumar and Daiya [10], Kumar et al. [12], and others have studied the fractional calculus pertaining to multivariable H-function defined by Srivastava and Panda [24].

The $\bar{I}$- function, introduced by Rathie [16], however the notation and complete definition is presented here in the following manner in terms of the Mellin-Barnes type integral:

$$\bar{I}(z)=\bar{I}_{p,q}^{m,n}\left[z\left|\begin{array}{l}(a_j,\alpha_i;A_j)_{n,n+1},(a_j,\alpha_i;A_j)_p\\(b_j,\beta_j;B_j)_{m,m+1},(b_j,\beta_j;B_j)_q\end{array}\right.\right]=\frac{1}{2\pi\omega}\int_L\Omega_{p,q}^{m,n}(s)\,z^{-s}\mathrm{d}s, \tag{1}$$

for all $z\neq 0$, and

$$\Omega_{p,q}^{m,n}(s)=\frac{\prod_{j=1}^{m}\Gamma^{B_i}(b_j+\beta_j s)\prod_{j=1}^{n}\Gamma^{A_j}(1-a_j-\alpha_j s)}{\prod_{j=n+1}^{p}\Gamma^{A_i}(a_j+\alpha_j s)\prod_{j=m+1}^{q}\Gamma^{B_j}(1-b_j-\beta_j s)}. \tag{2}$$

10.2478/jamsi-2019-0009

If $\Delta > 0$, the integral (1) converges when $|\arg z| < \frac{1}{2}\Delta$; where

$$\Delta = \sum_{j=1}^{m} B_j\beta_j - \sum_{j=m+1}^{q} B_j\beta_j + \sum_{j=1}^{n} A_j\alpha_j - \sum_{j=n+1}^{p} A_j\alpha_j. \tag{3}$$

The generalized polynomials defined by Srivastava [23], is given in the following manner:

$$S_{N_1,\cdots,N_s}^{M_1,\cdots,M_s}[y_1,\cdots,y_s] = \sum_{K_1=0}^{[N_1/M_1]} \cdots \sum_{K_s=0}^{[N_s/M_s]} \frac{(-N_1)_{M_1K_1}}{K_1!} \cdots \frac{(-N_s)_{M_sK_s}}{K_s!}$$
$$\times A[N_1,K_1;\cdots;N_s,K_s]\ y_1^{K_1}\cdots y_s^{K_s}, \tag{4}$$

where $M_1,\cdots,M_s$ are arbitrary positive integers; and the coefficients $A[N_1,K_1;\cdots;N_s,K_s]$ are arbitrary constants (real or complex).

The multivariable I-function defined by Prathima et al. [14] (see also, [8]) is a extension of the multivariable H-function [5; 6; 22; 24]. It is defined in term of multiple Mellin-Barnes type integral, given by

$$I(z_1,\cdots,z_r) = I_{p,q:p_1,q_1;\cdots;p_r,q_r}^{0,n:m_1,n_1;\cdots;m_r,n_r}\left[\begin{array}{c|l} z_1 & \\ . & \\ . & \\ . & \\ z_r & \end{array}\right.$$
$$\left.\begin{array}{l} \left(a_j;\alpha_j^{(1)},\cdots,\alpha_j^{(r)};A_j\right)_{1,p}: \\ \\ \left(b_j;\beta_j^{(1)},\cdots,\beta_j^{(r)};B_j\right)_{1,q}: \end{array}\right.$$
$$\left.\begin{array}{l} \left(c_j^{(1)},\gamma_j^{(1)};C_j^{(1)}\right)_{1,p_1};\cdots;\left(c_j^{(r)},\gamma_j^{(r)};C_j^{(r)}\right)_{1,p_r} \\ \\ \left(d_j^{(1)},\delta_j^{(1)};D_j^{(1)}\right)_{1,q_1};\cdots;\left(d_j^{(r)},\delta_j^{(r)};D_j^{(r)}\right)_{1,q_r} \end{array}\right] \tag{5}$$

$$= \frac{1}{(2\pi\omega)^r}\int_{L_1}\cdots\int_{L_r}\phi(s_1,\cdots,s_r)\left\{\prod_{i=1}^{r}\theta_i(s_i)\,z_i^{s_i}\right\}\mathrm{d}s_1\cdots\mathrm{d}s_r, \tag{6}$$

where $\omega = \sqrt{-1}$; $\phi(s_1,\cdots,s_r)$ and $\theta_i(s_i)$ for $i = 1,\cdots,r$ are given as follows

$$\phi(s_1,\cdots,s_r) = \frac{\prod_{j=1}^{n}\Gamma^{A_j}\left(1-aj+\sum_{i=1}^{r}\alpha_j^{(i)}s_j\right)}{\prod_{j=n+1}^{p}\Gamma^{A_j}\left(a_j-\sum_{i=1}^{r}\alpha_j^{(i)}s_j\right)\prod_{j=1}^{q}\Gamma^{B_j}\left(1-bj+\sum_{i=1}^{r}\beta_j^{(i)}s_j\right)}, \tag{7}$$

$$\theta_i(s_i) = \frac{\prod_{j=1}^{n_i}\Gamma^{C_j^{(i)}}\left(1-c_j^{(i)}+\gamma_j^{(i)}s_i\right)\prod_{j=1}^{m_i}\Gamma^{D_j^{(i)}}\left(d_j^{(i)}-\delta_j^{(i)}s_i\right)}{\prod_{j=n_i+1}^{p_i}\Gamma^{C_j^{(i)}}\left(c_j^{(i)}-\gamma_j^{(i)}s_i\right)\prod_{j=m_i+1}^{q_i}\Gamma^{D_j^{(i)}}\left(1-d_j^{(i)}+\delta_j^{(i)}s_i\right)}, \tag{8}$$

for more details, reader can refer to recent work given by Prathima et al. [14].
Following the result of Braaksma [3], the I-function of r variables is analytic if

$$U_i = \sum_{j=1}^{p} A_j \alpha_j^{(i)} - \sum_{j=1}^{q} B_j \beta_j^{(i)} + \sum_{j=1}^{p_i} C_j^{(i)} \gamma_j^{(i)} - \sum_{j=1}^{q_i} D_j^{(i)} \delta_j^{(i)} \leqslant 0 \quad (i = 1, \ldots, r). \tag{9}$$

The integral (6) converges absolutely if

$$|\arg(z_k)| < \frac{1}{2}\Delta_k \pi \quad (k = 1, \ldots, r), \text{ where}$$

$$\Delta_k = -\sum_{j=n+1}^{p} A_j \alpha_j^{(k)} - \sum_{j=1}^{q} B_j \beta_j^{(k)} + \sum_{j=1}^{m_k} D_j^{(k)} \delta_j^{(k)} - \sum_{j=m_k+1}^{q_k} D_j^{(k)} \delta_j^{(k)} + \sum_{j=1}^{n_k} C_j^{(k)} \gamma_j^{(k)} - \sum_{j=n_k+1}^{p_k} C_j^{(k)} \gamma_j^{(k)} > 0. \tag{10}$$

The complex numbers z_i are not zero. Throughout this paper, we assume the existence and absolute convergence conditions of the multivariable I-function.
We will note

$$A = \left(a_j; \alpha_j^{(1)}, \ldots, \alpha_j^{(r)}; A_j\right)_{1,p}. \tag{11}$$

$$B = \left(b_j; \beta_j^{(1)}, \ldots, \beta_j^{(r)}; B_j\right)_{1,q}. \tag{12}$$

$$C = \left(c_j^{(1)}, \gamma_j^{(1)}; C_j^{(1)}\right)_{1,p_1}; \ldots; \left(c_j^{(r)}, \gamma_j^{(r)}; C_j^{(r)}\right)_{1,p_r}. \tag{13}$$

$$D = \left(d_j^{(1)}, \delta_j^{(1)}; D_j^{(1)}\right)_{1,q_1}; \ldots; \left(d_j^{(r)}, \delta_j^{(r)}; D_j^{(r)}\right)_{1,q_r}. \tag{14}$$

The Mellin transform of $f(x)$ will be denoted by $M[f(x)]$ or $F(s)$. If p and y are real, we write $s = p^{-1} + iy$. If $p \geqslant 1$, $f(x) \in L_p(0, \infty)$, then for $p = 1$ we have

$$M[f(x)] = F(s) = \int_0^\infty x^{s-1} f(x)\,\mathrm{d}x \text{ and } f(x) = \frac{1}{2i\pi}\int_L F(s) x^{-s}\,\mathrm{d}s. \tag{15}$$

$$\text{For } p > 0,\ M[f(x)] = F(s) = l.i.m. \int_{1/x}^{x} x^{s-1} f(x)\,\mathrm{d}x, \tag{16}$$

where, $l.i.m.$ denotes the usual limit in the mean for L_p-spaces.

2. DEFINITIONS

The pair of new extended fractional integral operators are defined by the following equations:

$$Q_{\gamma_n}^{\alpha,\beta}[f(x)]$$

$$= tx^{-\alpha-t\beta-1}\int_0^x y^{\alpha}\left(x^t-y^t\right)^{\beta} I\left(\begin{array}{c|c}\gamma_1\upsilon_1 & A:C\\ \cdot & \cdot\\ \cdot & \cdot\\ \cdot & \cdot\\ \gamma_n\upsilon_n & B:D\end{array}\right)\prod_{j=1}^{k}\bar{I}_{M_j'',N_j''}^{M_j',N_j'}\left[z_j\left(\frac{y^t}{x^t}\right)^{a_j}\left(1-\frac{y^t}{x^t}\right)^{b_j}\right]$$

$$\times\prod_{j=1}^{r}S_{N_1^{(j)},\ldots,N_s^{(j)}}^{M_1^{(j)},\ldots,M_s^{(j)}}\left(\begin{array}{c}z_1^{(j)}\left(\frac{y^t}{x^t}\right)^{g_1^{(j)}}\left(1-\frac{y^t}{x^t}\right)^{h_1^{(j)}}\\ \cdot\\ \cdot\\ \cdot\\ z_s^{(j)}\left(\frac{y^t}{x^t}\right)^{g_s^{(j)}}\left(1-\frac{y^t}{x^t}\right)^{h_s^{(j)}}\end{array}\right)\psi\left(\frac{y^t}{x^t}\right)f(y)\,\mathrm{d}y. \tag{17}$$

$$R_{\gamma_n}^{\rho,\beta}[f(x)]$$

$$= tx^{\rho}\int_x^{\infty} y^{-\rho-t\beta-1}\left(y^t-x^t\right)^{\beta} I\left(\begin{array}{c|c}\gamma_1\mu_1 & A:C\\ \cdot & \cdot\\ \cdot & \cdot\\ \cdot & \cdot\\ \gamma_n\mu_n & B:D\end{array}\right)\prod_{j=1}^{k}\bar{I}_{M_j'',N_j''}^{M_j',N_j'}\left[z_j\left(\frac{x^t}{y^t}\right)^{a_j}\left(1-\frac{x^t}{y^t}\right)^{b_j}\right]$$

$$\times\prod_{j=1}^{r}S_{N_1^{(j)},\cdots,N_s^{(j)}}^{M_1^{(j)},\cdots,M_s^{(j)}}\left(\begin{array}{c}z_1^{(j)}\left(\frac{x^t}{y^t}\right)^{g_1^{(j)}}\left(1-\frac{x^t}{y^t}\right)^{h_1^{(j)}}\\ \cdot\\ \cdot\\ \cdot\\ z_s^{(j)}\left(\frac{x^t}{y^t}\right)^{g_s^{(j)}}\left(1-\frac{x^t}{y^t}\right)^{h_s^{(j)}}\end{array}\right)\psi\left(\frac{x^t}{y^t}\right)f(y)\,\mathrm{d}y, \tag{18}$$

where, $\upsilon_i=\left(\frac{y^t}{x^t}\right)^{u_i}\left(1-\frac{y^t}{x^t}\right)^{v_i}$, $\mu_i=\left(\frac{x^t}{y^t}\right)^{u_i}\left(1-\frac{x^t}{y^t}\right)^{v_i}$ and $t,u_i,v_i,g_i^{(j)},h_i^{(j)},a_j,\ b_j$ are positive numbers.

The kernels $\psi\left(\frac{y^t}{x^t}\right)$ and $\psi\left(\frac{x^t}{y^t}\right)$ appearing in (17) and (18) respectively, are assumed to be continuous functions such the integrals make sense for wide classes of function

$f(x)$.

The conditions for existence of these operators are as follows:

(a) $f(x) \in L_p(0,\infty)$, (b) $1 \leqslant p,q < \infty$, $p^{-1}+q^{-1}=1$.

(c) $\Re\left(\alpha + ta_j \frac{b_{j'j}}{\beta_{j'j}}\right) + t\sum_{i=1}^{n} u_i \min_{1\leqslant j\leqslant m_i} \Re\left(\frac{d_j^{(i)}}{\delta_j^{(i)}}\right) > -q^{-1}$.

(d) $\Re\left(\beta + tb_j \frac{b_{j'j}}{\beta_{j'j}}\right) + t\sum_{i=1}^{n} v_i \min_{1\leqslant j\leqslant m_i} \Re\left(\frac{d_j^{(i)}}{\delta_j^{(i)}}\right) > -q^{-1}$.

(e) $\Re\left(\rho + ta_j \frac{b_{j'j}}{\beta_{j'j}}\right) + t\sum_{i=1}^{n} u_i \min_{1\leqslant j\leqslant m_i} \Re\left(\frac{d_j^{(i)}}{\delta_j^{(i)}}\right) > -p^{-1}$ where $j = 1\ldots,k;\ j' = 1,\ldots,M'_j$.

Condition (a) ensures that both operators defined in (17) and (18) exist and belong to. These operators are extensions of fractional integral operators defined and studied by several authors like Erdélyi [7], Love [13], Saigo et al. [17], Saxena and Kiryakova [18], Saxena and Kumbhat [20; 21], and etc.

3. MAIN RESULTS

THEOREM 1. If $f(x) \in L_p(0,\infty)$, $1 \leqslant p \leqslant 2$; or $f(x) \in L_p(0,\infty)$, $p > 2$, also following conditions satisfied:

$p^{-1}+q^{-1}=1$,

$$\Re\left(\alpha + ta_j \frac{b_{j'j}}{\beta_{j'j}}\right) + t\sum_{i=1}^{n} u_i \min_{1\leqslant j\leqslant m_i} \Re\left(\frac{d_j^{(i)}}{\delta_j^{(i)}}\right) > -q^{-1},$$

$$\Re\left(\beta + tb_j \frac{b_{j'j}}{\beta_{j'j}}\right) + t\sum_{i=1}^{n} v_i \min_{1\leqslant j\leqslant m_i} \Re\left(\frac{d_j^{(i)}}{\delta_j^{(i)}}\right) > -q^{-1},$$

and the integrals present are absolutely convergent, then

$$M\left\{Q_{\gamma_n}^{\alpha,\beta}[f(x)]\right\} = M\{f(x)\} R_{\gamma_n}^{\alpha-s+1,\beta}[1], \tag{19}$$

where $M_p(0,\infty)$ stands for the class of all functions $f(x)$ of $L_p(0,\infty)$ with $p > 2$, which are inverse Mellin-transforms of the function of $L_p(-\infty,\infty)$.

PROOF. By making Mellin transform of (17), we get

$$M\left\{Q_{\gamma_n}^{\alpha,\beta}[f(x)]\right\}=\int_0^\infty x^{s-1}\left\{tx^{-\alpha-t\beta-1}\int_0^x y^\alpha\left(x^t-y^t\right)^\beta I\left(\begin{array}{c|c}\gamma_1\upsilon_1 & A:C\\ \cdot & \cdot\\ \cdot & \cdot\\ \cdot & \\ \gamma_n\upsilon_n & B:D\end{array}\right)\right.$$

$$\left.\prod_{j=1}^k \bar{I}_{M_j'',N_j''}^{M_j',N_j'}\left[z_j\left(\frac{y^t}{x^t}\right)^{a_j}\left(1-\frac{y^t}{x^t}\right)^{b_j}\right]S_{N_1^{(j)},\ldots,N_s^{(j)}}^{M_1^{(j)},\ldots,M_s^{(j)}}\left(\begin{array}{c}z_1^{(j)}\left(\frac{y^t}{x^t}\right)^{g_1^{(j)}}\left(1-\frac{y^t}{x^t}\right)^{h_1^{(j)}}\\ \cdot\\ \cdot\\ \cdot\\ z_s^{(j)}\left(\frac{y^t}{x^t}\right)^{g_s^{(j)}}\left(1-\frac{y^t}{x^t}\right)^{h_s^{(j)}}\end{array}\right)f(y)\mathrm{d}y\right\}\mathrm{d}x. \tag{20}$$

On interchanging the order of integration, which is permissible under the conditions, the result (19) follows in view of (18).

THEOREM 2. If $f(x)\in L_p(0,\infty)$, $1\leqslant p\leqslant 2$; or $f(x)\in L_p(0,\infty)$, $p>2$, also satisfied following conditions:

$$p^{-1}+q^{-1}=1,$$

$$\Re\left(\beta+tb_j\frac{b_{j'j}}{\beta_{j'j}}\right)+t\sum_{i=1}^n v_i\min_{1\leqslant j\leqslant m_i}\Re\left(\frac{d_j^{(i)}}{\delta_j^{(i)}}\right)>-q^{-1},$$

$$\Re\left(\rho+ta_j\frac{b_{j'j}}{\beta_{j'j}}\right)+t\sum_{i=1}^n u_i\min_{1\leqslant j\leqslant m_i}\Re\left(\frac{d_j^{(i)}}{\delta_j^{(i)}}\right)>-p^{-1},$$

and the integrals present are absolutely convergent, then

$$M\left\{R_{\gamma_n}^{\rho,\beta}[f(x)]\right\}=M\left\{f(x)\right\}Q_{\gamma_n}^{\rho+s-1,\beta}[1], \tag{21}$$

where $M_p(0,\infty)$ stands for the class of all functions $f(x)$ of $L_p(0,\infty)$ with $p>2$, which are inverse Mellin-transforms of the function of $L_p(-\infty,\infty)$.

PROOF. By making Mellin transform of (18), we get

$$M\left\{R_{\gamma_n}^{\rho,\beta}[f(x)]\right\} = \int_0^\infty x^{s-1}\left\{tx^\rho \int_x^\infty y^{-\rho-t\beta-1}\left(y^t-x^t\right)^\beta I\left(\begin{array}{c|c}\gamma_1\mu_1 & A:C\\ \cdot & \cdot\\ \cdot & \cdot\\ \gamma_n\mu_n & B:D\end{array}\right)\right.$$

$$\times \prod_{j=1}^{k} I_{M_j'',N_j''}^{M_j',N_j'}\left[z_j\left(\frac{x^t}{y^t}\right)^{a_j}\left(1-\frac{x^t}{y^t}\right)^{b_j}\right]$$

$$\left.\times \prod_{j=1}^{r} S_{N_1^{(j)},\ldots,N_s^{(j)}}^{M_1^{(j)},\ldots,M_s^{(j)}}\left(\begin{array}{c} z_1^{(j)}\left(\frac{x^t}{y^t}\right)^{g_1^{(j)}}\left(1-\frac{x^t}{y^t}\right)^{h_1^{(j)}}\\ \cdot\\ \cdot\\ \cdot\\ z_s^{(j)}\left(\frac{x^t}{y^t}\right)^{g_s^{(j)}}\left(1-\frac{x^t}{y^t}\right)^{h_s^{(j)}}\end{array}\right)\psi\left(\frac{x^t}{y^t}\right)f(y)\mathrm{d}y\right\}\mathrm{d}x. \tag{22}$$

THEOREM 3. If $f(x) \in L_p(0,\infty)$, $v(x) \in L_p(0,\infty)$, also satisfied following conditions:

$$p^{-1}+q^{-1}=1,$$

$$\Re\left(\alpha+ta_j\frac{b_{j'j}}{\beta_{j'j}}\right)+t\sum_{i=1}^{n}u_i\min_{1\leqslant j\leqslant m_i}\Re\left(\frac{d_j^{(i)}}{\delta_j^{(i)}}\right)>\max\{p^{-1},q^{-1}\},$$

$$\Re\left(\beta+tb_j\frac{b_{j'j}}{\beta_{j'j}}\right)+t\sum_{i=1}^{n}v_i\min_{1\leqslant j\leqslant m_i}\Re\left(\frac{d_j^{(i)}}{\delta_j^{(i)}}\right)>0,$$

and the integrals present are absolutely convergent, then

$$\int_0^\infty v(x)\,Q_{\gamma_n}^{\alpha,\beta}[f(x)]\,\mathrm{d}x=\int_0^\infty f(x)\,R_{\gamma_n}^{\alpha,\beta}[v(x)]\,\mathrm{d}x. \tag{23}$$

PROOF. The result of (23) can be obtained in view of (17) and (18).

4. INVERSION FORMULAS

THEOREM 4. If $f(x) \in L_p(0,\infty)$, $1 \leqslant p \leqslant 2$; or $f(x) \in L_p(0,\infty)$, $p > 2$, also following conditions satisfied:

$p^{-1}+q^{-1}=1$,

$$\Re\left(\alpha+ta_j\frac{b_{j'j}}{\beta_{j'j}}\right)+t\sum_{i=1}^{n}u_i\min_{1\leqslant j\leqslant m_i}\Re\left(\frac{d_j^{(i)}}{\delta_j^{(i)}}\right)>-q^{-1},$$

$$\Re\left(\beta+tb_j\frac{b_{j'j}}{\beta_{j'j}}\right)+t\sum_{i=1}^{n}v_i\min_{1\leqslant j\leqslant m_i}\Re\left(\frac{d_j^{(i)}}{\delta_j^{(i)}}\right)>-q^{-1},$$

and the integrals present are absolutely convergent, also we have

$$Q_{\gamma_n}^{\alpha,\beta}[f(x)]=v(x), \tag{24}$$

then

$$f(x)=\int_0^\infty y^{-1}[v(y)]\left[h\left(\frac{x}{y}\right)\right]\mathrm{d}y, \tag{25}$$

where

$$h(x)=\frac{1}{2i\pi}\int_{c-i\infty}^{c+i\infty}y^{-1}\frac{x^{-s}}{R(s)}\,\mathrm{d}s, \tag{26}$$

$$R(s)=R_{\gamma_n}^{\alpha-s+1,\beta}[1]. \tag{27}$$

PROOF. On taking Mellin transform of (24) and then applying Theorem 1, we get

$$M\{f(x)\}=\frac{M\{v(x)\}}{R(s)},$$

which on inverting leads to

$$f(x)=\frac{1}{2i\pi}\int_{c-i\infty}^{c+i\infty}x^{-s}\frac{M\{v(x)\}}{R(s)}\,\mathrm{d}s=\frac{1}{2i\pi}\int_{c-i\infty}^{c+i\infty}\frac{x^{-s}}{R(s)}\left\{\int_0^\infty[v(y)]\,\mathrm{d}y\right\}\mathrm{d}s.$$

Interchanging the order of integration, we obtain

$$f(x)=\int_0^\infty\frac{v(y)}{y}\left\{\frac{1}{2i\pi}\int_{c-i\infty}^{c+i\infty}\left(\frac{x}{y}\right)^s\frac{1}{R(s)}\,\mathrm{d}s\right\}\mathrm{d}y.$$

Now in view of (26), we obtain the desired result (25).

THEOREM 5. If $f(x) \in L_p(0,\infty)$, $1 \leqslant p \leqslant 2$; or $f(x) \in L_p(0,\infty)$, $p > 2$, also satisfied following conditions:

$$p^{-1}+q^{-1}=1,$$

$$\Re\left(\beta+tb_j\frac{b_{j'j}}{\beta_{j'j}}\right)+t\sum_{i=1}^{n}v_i\min_{1\leqslant j\leqslant m_i}\Re\left(\frac{d_j^{(i)}}{\delta_j^{(i)}}\right)>-q^{-1},$$

$$\Re\left(\rho+ta_j\frac{b_{j'j}}{\beta_{j'j}}\right)+t\sum_{i=1}^{n}u_i\min_{1\leqslant j\leqslant m_i}\Re\left(\frac{d_j^{(i)}}{\delta_j^{(i)}}\right)>-p^{-1},$$

and the integrals present are absolutely convergent, also we have

$$R_{\gamma_n}^{\rho,\beta}[f(x)]=w(x), \tag{28}$$

then

$$f(x)=\int_0^\infty y^{-1}[w(y)]\left[g\left(\frac{x}{y}\right)\right]\mathrm{d}y, \tag{29}$$

where

$$g(x)=\frac{1}{2i\pi}\int_{c-i\infty}^{c+i\infty}y^{-1}\frac{x^{-s}}{T(s)}\,\mathrm{d}s, \tag{30}$$

$$T(s)=Q_{\gamma_n}^{\rho+s-1,\beta}[1]. \tag{31}$$

PROOF. On taking Mellin transform of (28) and then applying Theorem 2, we get

$$M\{f(x)\}=\frac{M\{w(x)\}}{T(s)},$$

which on inverting leads to

$$f(x)=\frac{1}{2i\pi}\int_{c-i\infty}^{c+i\infty}x^{-s}\frac{M\{w(x)\}}{T(s)}\,\mathrm{d}s=\frac{1}{2i\pi}\int_{c-i\infty}^{c+i\infty}\frac{x^{-s}}{T(s)}\left\{\int_0^\infty[w(y)]\,\mathrm{d}y\right\}\mathrm{d}s.$$

Interchanging the order of integration, we obtain

$$f(x)=\int_0^\infty\frac{w(y)}{y}\left\{\frac{1}{2i\pi}\int_{c-i\infty}^{c+i\infty}\left(\frac{x}{y}\right)^s\frac{1}{T(s)}\,\mathrm{d}s\right\}\mathrm{d}y.$$

Now in view of (30), we obtain the desired result (29).

5. GENERAL PROPERTIES

The properties given below are consequences of the definitions (17) and (18).

$$x^{-1} Q_{\gamma_n}^{\alpha,\beta}\left[\frac{1}{x} f\left(\frac{1}{x}\right)\right] = R_{\gamma_n}^{\alpha,\beta}\left[f(x)\right], \tag{32}$$

$$x^{-1} R_{\gamma_n}^{\rho,\beta}\left[\frac{1}{x} f\left(\frac{1}{x}\right)\right] = Q_{\gamma_n}^{\rho,\beta}\left[f(x)\right], \tag{33}$$

$$x^{\mu} Q_{\gamma_n}^{\alpha,\beta}\left[f(x)\right] = Q_{\gamma_n}^{\alpha-\mu,\beta}\left[x^{\mu} f(x)\right], \tag{34}$$

$$x^{\mu} R_{\gamma_n}^{\rho,\beta}\left[f(x)\right] = R_{\gamma_n}^{\rho+\mu,\beta}\left[x^{\mu} f(x)\right]. \tag{35}$$

The properties given below express the homogeneity of operator Q and R respectively.

$$\text{If } Q_{\gamma_n}^{\alpha,\beta}\left[f(x)\right] = v(x) \text{ then } Q_{\gamma_n}^{\alpha,\beta}\left[f(cx)\right] = v(cx), \tag{36}$$

$$\text{If } R_{\gamma_n}^{\rho,\beta}\left[f(x)\right] = w(x) \text{ then } R_{\gamma_n}^{\rho,\beta}\left[f(cx)\right] = w(cx). \tag{37}$$

6. MULTIVARIABLE H-FUNCTION

If $A_j = B_j = C_j^{(i)} = D_j^{(i)} = 1$, the multivariable I-function defined by Prathima et al. [14] reduces to multivariable H-function. We obtain the two following operators:

$$\begin{aligned} &Q_{\gamma_n}^{\alpha,\beta}\left[f(x)\right] \\ &= t x^{-\alpha-t\beta-1} \int_0^x y^{\alpha}\left(x^t - y^t\right)^{\beta} H\left(\begin{array}{c|c} \gamma_1 \upsilon_1 & A:C \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \\ \gamma_n \upsilon_n & B:D \end{array}\right) \prod_{j=1}^{k} I_{M_j'', N_j''}^{M_j', N_j'}\left[z_j \left(\frac{y^t}{x^t}\right)^{a_j} \left(1 - \frac{y^t}{x^t}\right)^{b_j}\right] \\ &\quad \times \prod_{j=1}^{r} S_{N_1^{(j)},\ldots,N_s^{(j)}}^{M_1^{(j)},\ldots,M_s^{(j)}}\left(\begin{array}{c} z_1^{(j)}\left(\frac{y^t}{x^t}\right)^{g_1^{(j)}}\left(1-\frac{y^t}{x^t}\right)^{h_1^{(j)}} \\ \cdot \\ \cdot \\ \cdot \\ z_s^{(j)}\left(\frac{y^t}{x^t}\right)^{g_s^{(j)}}\left(1-\frac{y^t}{x^t}\right)^{h_s^{(j)}} \end{array}\right) \psi\left(\frac{y^t}{x^t}\right) f(y)\,\mathrm{d}y, \end{aligned} \tag{38}$$

under the same notations and conditions that (17) with $A_j = B_j = C_j^{(i)} = D_j^{(i)} = 1$.

$$R_{\gamma_n}^{\rho,\beta}[f(x)]$$

$$= tx^{\rho}\int_x^{\infty} y^{-\rho-t\beta-1}\left(y^t-x^t\right)^{\beta} H\left(\begin{array}{c|c}\gamma_1\mu_1 & A:C\\ \cdot & \cdot\\ \cdot & \cdot\\ \cdot & \\ \gamma_n\mu_n & B:D\end{array}\right)\prod_{j=1}^{k}\bar{I}_{M_j'',N_j''}^{M_j',N_j'}\left[z_j\left(\frac{x^t}{y^t}\right)^{a_j}\left(1-\frac{x^t}{y^t}\right)^{b_j}\right]$$

$$\times\prod_{j=1}^{r}S_{N_1^{(j)},\ldots,N_s^{(j)}}^{M_1^{(j)},\ldots,M_s^{(j)}}\left(\begin{array}{c}z_1^{(j)}\left(\frac{x^t}{y^t}\right)^{g_1^{(j)}}\left(1-\frac{x^t}{y^t}\right)^{h_1^{(j)}}\\ \cdot\\ \cdot\\ \cdot\\ z_s^{(j)}\left(\frac{x^t}{y^t}\right)^{g_s^{(j)}}\left(1-\frac{x^t}{y^t}\right)^{h_s^{(j)}}\end{array}\right)\psi\left(\frac{x^t}{y^t}\right)f(y)\,\mathrm{d}y, \qquad (39)$$

under the same notations and conditions that (18) with $A_j = B_j = C_j^{(i)} = D_j^{(i)} = 1$. We obtain the same theorems and properties concerning these operators.

7. CONCLUSION

The functions involved in the results established in this paper are unified and general nature, hence a large number of known results lying in the literature follows as special cases. Further, on suitable specifications of the parameters involved, numerous new results involving simpler functions may also be obtained.

8. DISCLOSURE STATEMENT

No potential conflict of interest was reported by the authors.

REFERENCES

F. Y. Ayant, An integral associated with the Aleph-functions of several variables, *Int. J. Math. Trends Tech.* 31(3) (2016), 142–154.

D. Baleanu, D. Kumar and S. D. Purohit, Generalized fractional integrals of product of two H-functions and a general class of polynomials, *Int. J. Comput. Math.* 93(8) (2016), 1320–1329.

B.L.J. Braaksma, Asymptotic expansions and analytic continuations for a class of Barnes integrals, *Compos. Math.* 15 (1964), 239–341.

V.B.L. Chaurasia and A. Srivastava, A unified approach to fractional calculus pertaining to H-functions, *Soochow J. Math.* 33(2) (2007), 211–221.

J. Choi, J. Daiya, D. Kumar and R. K. Saxena, Fractional differentiation of the product of Appell function F_3 and multivariable H-function, *Commun. Korean Math. Soc.*, 31(1) (2016), 115–129.

J. Daiya, J. Ram and D. Kumar, The multivariable H-function and the general class of Srivastava polynomials involving the generalized Mellin-Barnes contour integrals, *FILOMAT* 30(6) (2016), 1457–1464.

A. Erdélyi, On some functional transformations, *Univ. Politec. Torino, Rend. Sem. Mat.* 10 (1950-51), 217-234.

D. Kumar and F.Y. Ayant, Some double integrals involving multivariable I-function, *Acta Universitatis Apulensis* 58(2) (2019), 35–43.

D. Kumar, F.Y. Ayant and J. Choi, Application of product of the multivariable A-function and the multivariable Srivastava polynomials, *East Asian Math. J.* 34(3) (2018), 295–303.

D. Kumar and J. Daiya, Fractional calculus pertaining to generalized H-functions, *Glob. J. Sci. Front. Res. (F)* 14(3) (2014), 25–35.

D. Kumar, R.K. Gupta, B.S. Shaktawat and J. Choi, Generalized fractional calculus formulas involving the product of Aleph-function and Srivastava polynomials, *Proc. Jangjeon Math. Soc.* 20(4) (2017), 701–717.

D. Kumar, S. D. Purohit and J. Choi, Generalized fractional integrals involving product of multivariable H-function and a general class of polynomials, *J. Nonlinear Sci. Appl.* 9 (2016), 8–21.

E.R. Love, Some integral equations involving hypergeometric functions, *Proc. Edinb. Math. Soc.* 15 (3) (1967), 169–198.

J. Prathima, V. Nambisan and S. K. Kurumujji, A study of I-function of several complex variables, *Int. J. Engrg. Math.* 2014 (2014), 1–12.

J. Ram and D. Kumar, Generalized fractional integration involving Appell hypergeometric function of the product of two H-functions, *Vijanana Parishad Anusandhan Patrika* 54(3) (2011), 33–43.

A.K. Rathie, A new generalization of generalized hypergeometric function, *Le Matematiche* 52(2) (1997), 297–310.

M. Saigo, R.K. Saxena and J. Ram, On the fractional calculus operator associated with the H-function, *Ganita Sandesh* 6(1) (1992), 36–47.

R.K. Saxena and V.S. Kiryakova, On relation between the two-dimensional H-transforms in terms of Erdélyi-Kober operators, *Math. Balkanica* 6 (1992), 133–140.

R.K. Saxena and D. Kumar, Generalized fractional calculus of the Aleph-function involving a general class of polynomials, *Acta Math. Sci. Ser. B Engl. Ed.* 35(5) (2015), 1095–1110.

R.K. Saxena and R.K. Kumbhat, Fractional integration operators of two variables, *Proc. Indian Acad. Sci. Math. Sci.* 78 (1973), 177–186.

R.K. Saxena and R.K. Kumbhat, Some properties of generalized Kober operators, *Vijnana Parishad Anusandhan Patrika* 18 (1975), 139–150.

R.K. Saxena, J. Ram and D.L. Suthar, Unified fractional derivative formulas for the multivariable H-function, *Vijnana Parishad Anusandhan Patrika* 49(2) (2006), 159–175.

H.M. Srivastava, A multilinear generating function for the Konhauser set of biorthogonal polynomials suggested by Laguerre polynomial, *Pacific J. Math.* 177 (1985), 183–191.

H.M. Srivastava and R. Panda, Some bilateral generating functions for a class of generalized hypergeometric polynomials, *J. Reine Angew. Math.* 283/284 (1976), 265–274.

Dinesh Kumar
Department of Applied Sciences,
College of Agriculture, Sumerpur-Pali,
Agriculture University of Jodhpur, Jodhpur 342304, India
email: dinesh_dino03@yahoo.com

Frédéric Ayant
Collége Jean L'herminier,
Allée des Nymphéas, 83500 La Seyne-sur-Mer, France
Six-Fours-les-Plages 83140, Department of Var, France
email: fredericayant@gmail.com