Annales Mathematicae et Informaticae, ISSN 1787-5021,
DOI: 10.33039/ami.2021.03.004 - in press

Efficient Taylor expansion computation of multidimensional vector functions on GPU*

Vaclav Skala[a]

[a]University of West Bohemia, Pilsen, Czech Republic
skala@kiv.zcu.cz

Annales Mathematicae et Informaticae
Manuscript
March 13, 2021

Abstract

The Taylor expansion [19] is used in many applications for a value estimation of scalar functions of one or two variables in the neighbour point. Usually, only the first two elements of the Taylor expansion are used, i.e. a value in the given point and derivatives estimation. The Taylor expansion can be also used for vector functions, too. The usual formulae are well known, but if the second element of the expansion, i.e. with the second derivatives are to be used, mathematical formulations are getting too complex for efficient programming, as it leads to the use of multi-dimensional matrices.

This contribution describes a new form of the Taylor expansion for multidimensional vector functions. The proposed approach uses "standard" formalism of linear algebra, i.e. using vectors and matrices, which is simple, easy to implement. It leads to efficient computation on the GPU in the three dimensional case, as the GPU offers fast vector-vector computation and many parts can be done in parallel.

Keywords: Taylor expansion, vector functions, vector-vector operations, approximation, GPU and SSE instructions, parallel computation, radial basis functions.

MSC: 41A58, 65D15, 26B05, 65D05

*This research was supported by the Czech Science Foundation (GACR), project No. GA 17-05534S

Annales Mathematicae et Informaticae, ISSN 1787-5021,
DOI: 10.33039/ami.2021.03.004 - in press

1. Introduction

The Taylor expansion was introduced by the English mathematician Brook Taylor in 1715. However, closely related methods were given by Madhava of Sangamagrama in the 14th century [19]. It is used in many applications and used to approximate evaluation of many functions. In particular, the first two elements of the Taylor expansion are used as a linearization of a function behaviour at the given point and its surroundings [20]. The Taylor expansion is used in solutions of partial differential equations (PDE) [1] [7] [17], ordinary differential equations (ODE) [2] [6] [21] , integral equations (IE) integro-differential equations (IDE) [1] [11], approximation of inverse functions (AIF) [8], control theory [4], fluid flow visualization of 3D flow using radial basis functions [12] [13] [15], computer vision [5][18], in statistical mechanics [10], antenna design [9] [6], operator theory [3], etc.

2. Taylor expansion of Scalar Functions

The Taylor expansion of a scalar function is defined as successive derivatives, generally called tensors. In the one-dimensional case, i.e. scalar functions, the first derivative is actually the gradient $\nabla f(x)$, the second derivative has the form of a Hessian matrix, the third form leads to three-dimensional matrix $\mathbf{H}(x)$, i.e. triples of vectors etc. In the following, the Taylor expansion for scalar and for a vector functions are described.

2.1. One-dimensional Case

The Taylor expansion of a continuous scalar function of a one dimensional variable is given as:

$$f(x) = f(x_0) + \sum_{k=1}^{\infty} \frac{1}{k!} \frac{\partial^k f(x_0)}{\partial x^k} (x - x_0)^k \tag{2.1}$$

or as:

$$f(x) = f(x_0) + \sum_{k=0}^{\infty} \frac{1}{k!} \frac{\partial^k f(x_0)}{\partial x^k} \triangle^k \tag{2.2}$$

where $\triangle = x - x_0$. Generally, the Taylor expansion can be described as:

$$f(x) = T_0 + T_1 + T_2 + T_3 + \dots \tag{2.3}$$

where T_k can be expressed as:

$$T_k = \frac{1}{k!} \frac{\partial^k f(x_0)}{\partial x^k} \triangle^k \tag{2.4}$$

It can be seen, that the Taylor expansion of a scalar function of a one dimensional variable can be described as:

$$f(x) = f(x_0) + \frac{\partial^1 f(x_0)}{\partial x} \triangle + \frac{1}{2} \frac{\partial^2 f(x_0)}{\partial x^2} \triangle^2 + \frac{1}{6} \frac{\partial^3 f(x_0)}{\partial x^3} \triangle^3 + ... \tag{2.5}$$

Annales Mathematicae et Informaticae, ISSN 1787-5021,
DOI: 10.33039/ami.2021.03.004 - in press

However, the Taylor expansion is also used for a scalar function of m-dimensional variables, i.e. $f(\mathbf{x}) = f(x_1, ..., x_m)$. In this case, the expanded version of the Taylor expansion gets a little bit more complicated.

2.2. Multi-dimensional Case

In the case of the scalar function with the multidimensional argument, i.e. $f(\mathbf{x}) = f(x_1, ..., x_m)$, the Taylor expansion is more complicated as:

$$f(\mathbf{x}) = T_0 + T_1 + T_2 + T_3 + \dots \tag{2.6}$$

where T_k can be expressed as:

$$T_k = \frac{1}{k!}[D^k f(\mathbf{x}_0)][\triangle^k] \tag{2.7}$$

where:

$$\begin{gathered} D^k f(\mathbf{x}) = \tfrac{\partial^k f(\mathbf{x})}{\partial x_1^{k_1}, \dots, \partial x_m^{k_m}}, \quad [\triangle^k] = [\triangle_1^{k_1}, \dots, \triangle_m^{k_m}]^T, \\ k = \textstyle\sum_{i=1}^m k_i, \quad k_i \geq 0 \end{gathered} \tag{2.8}$$

Now, the Taylor expansion is defined as:

$$f(\mathbf{x}) = f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T \mathbf{H}(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) + T_3 + ... \tag{2.9}$$

or as:

$$f(\mathbf{x}) = f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0) \left[\triangle_i\right] + \frac{1}{2} \left[\triangle_i\right]^T \mathbf{H}(\mathbf{x}_0) \left[\triangle_i\right] + T_3 + ... \tag{2.10}$$

where: $[\triangle_i] = [\triangle_1, \dots, \triangle_m]^T$, $\nabla f(\mathbf{x}_0)$ is a gradient of the function $f(\mathbf{x})$ at the point $\mathbf{x}_0$, $\mathbf{H}(\mathbf{x}_0)$ is the Hessian matrix of the given function, i.e.:

$$\mathbf{H}(\mathbf{x}_0) = \left[\tfrac{\partial^2 f(\mathbf{x}_0)}{\partial x_i \partial x_j}\right], \ i, j = 1, \dots, m \tag{2.11}$$

In the majority of cases:

$$\frac{\partial^2 f(\mathbf{x}_0)}{\partial x_i \partial x_j} = \frac{\partial^2 f(\mathbf{x}_0)}{\partial x_j \partial x_i}, \ i, j = 1, \dots, m \tag{2.12}$$

The element T_3 of the Taylor expansion for a scalar function of m-dimensional variable is:

$$T_3 = \frac{1}{6} \sum_{i,j,k=1,1,1}^{m,m,m} \frac{\partial^3 f(\mathbf{x}_0)}{\partial x_i \partial x_j \partial x_k} \triangle_i \triangle_j \triangle_k \tag{2.13}$$

This is quite complex form leading to higher computational requirements. Similarly to the case of the Hessian matrix, it can be expected that the order of the function derivations is independent.

Annales Mathematicae et Informaticae, ISSN 1787-5021,
DOI: 10.33039/ami.2021.03.004 - in press

It can be seen that the element T_3 of the Taylor expansion consists of a "three dimensional matrix". It leads to the tensor notation, which is usually not part of the engineering education. If this notation is used directly in a program implementation, it leads to redundant computations due to the symmetry of higher order partial derivatives, see (2.11,2.12). Also handling with indexes might be too complicated.

Furthermore, in the physically oriented applications, it is necessary to use the Taylor expansion also for vector functions, i.e. for n-dimensional functions with m-dimensional arguments, in general.

3. Taylor expansion of Vector Functions

Vector functions are used in many physically oriented computations, e.g. fluid mechanics, electromagnetic field computation etc. The Taylor expansion for vector functions is more complicated.

Let us consider a vector function:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x_1, \ldots, x_m \end{bmatrix} \tag{3.1}$$

The Taylor expansion of a vector function can be expressed as:

$$\mathbf{f}(\mathbf{x}) = \sum_{i=0}^{\infty} \mathbf{T}_i(\mathbf{x}_0) \tag{3.2}$$

where $\mathbf{T}_i(\mathbf{x}_0)$ are vectors, now. Explicitly, it is possible to write:

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}_0) + \mathbf{J}(\mathbf{x}_0)\left[\triangle_i\right] + \frac{1}{2}\begin{bmatrix} \left[\triangle_i\right]^T \mathbf{H}^1(\mathbf{x}_0)\left[\triangle_i\right] \\ \vdots \\ \left[\triangle_i\right]^T \mathbf{H}^n(\mathbf{x}_0)\left[\triangle_i\right] \end{bmatrix} + \mathbf{T}_3 + \ldots \tag{3.3}$$

where $\mathbf{J}(\mathbf{x}_0) = \left[\frac{\partial f_i(\mathbf{x}_0)}{\partial x_j}\right]$ is the Jacobi matrix $(n \times m)$ and $\mathbf{H}^k(\mathbf{x}_0)$ are the Hessian matrices $(m \times m)$ with the second derivatives of the function $f_k(\mathbf{x})$, $k = 1, \ldots, n$, in general.

It can be seen, that the element $\mathbf{T}_2$ of the Taylor expansion is not expressed by standard linear algebra formalism as its result must be a vector, i.e. a "three-dimensional matrix" would have to be used containing elements $\left[\frac{\partial f_i^2(\mathbf{x}_0)}{\partial x_j \partial x_k}\right]$. Also, it is necessary to point out that memory requirements can be estimated as $O(nm^2)$, as the matrix $\mathbf{H}^k$ is of the size $(m \times m)$ and $k = 1, \ldots, n$.

Annales Mathematicae et Informaticae, ISSN 1787-5021,
DOI: 10.33039/ami.2021.03.004 - in press

4. Re-formulation of the Taylor Expansion

A short summaries of the Taylor expansion for scalar and vector functions have been given in sections 2 and 3. If higher degree elements than the linear ones are to be used, e.g. T_2 or T_3, the efficient representation and implementation gets more complex and computationally time consuming.

In the following, a modification of the Taylor expansion for the case $n = m = 3$ is presented. It uses only standard matrix-vector multiplication and also allows simpler symbolic manipulation of it. However, the given approach can be extended for higher dimensions, i.e. $n > 3$ and $m > 3$.

4.1. Scalar Functions

In the case of a scalar function with a multidimensional argument the Taylor expansion is defined as:

$$f(\mathbf{x}) = f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)\left[\triangle_i\right] + \frac{1}{2}\left[\triangle_i\right]^T \mathbf{H}(\mathbf{x}_0)\left[\triangle_i\right] + T_3 + ... \tag{4.1}$$

where: $\left[\triangle_i\right] = \left[\triangle_1, \ldots, \triangle_m\right]^T$, $\left[\triangle_i^2\right] = \left[\triangle_1^2, \ldots, \triangle_m^2\right]^T$ and $\triangle_i = x_i - x_{i_0}$, $i = 1, \ldots, m$.

The T_2 element is formed by a quadratic form and the T_3 element is formed by a three-dimensional matrix, see (2.10). It causes several complications in formal manipulation and implementation as well. However, in the majority of cases:

$$\frac{\partial^2 f(\mathbf{x}_0)}{\partial x_i \partial x_j} = \frac{\partial^2 f(\mathbf{x}_0)}{\partial x_j \partial x_i}, \qquad i, j = 1, \ldots, m \tag{4.2}$$

Therefore only $m(m+1)/2$ values are needed for evaluation of the T_2 element of the Taylor expansion. It means that the T_2 element of the Taylor expansion, i.e. the element with the Hessian matrix, can be split to two parts using the inner product (dot product) as follows:

$$T_2 = \frac{1}{2}\left[\frac{\partial^2 f(\mathbf{x}_0)}{(\partial x_i)^2}\right]\left[\triangle_i^2\right] + \sum_{i,j \& i>j}^{m,m} \frac{\partial^2 f(\mathbf{x}_0)}{\partial x_i \partial x_j}\left[\triangle_i \triangle_j\right] \tag{4.3}$$

where: $\left[\triangle\right] = \left[\triangle_1, \ldots, \triangle_m\right]^T$ and $\left[\triangle^2\right] = \left[\triangle_1^2, \ldots, \triangle_m^2\right]^T$, in general.

It means, that in the three-dimensional case, i.e. $f(\mathbf{x}) = f(x_1, ..., x_3)$, the Taylor expansion gets quite simple as the element T_2 has the form:

$$T_2 = \frac{1}{2}\nabla^2 f(\mathbf{x}_0)\begin{bmatrix}\triangle_1^2\\ \triangle_2^2\\ \triangle_3^2\end{bmatrix} + \begin{bmatrix}\frac{\partial^2 f(\mathbf{x}_0)}{\partial x_1 \partial x_2} & \frac{\partial^2 f(\mathbf{x}_0)}{\partial x_2 \partial x_3} & \frac{\partial^2 f(\mathbf{x}_0)}{\partial x_3 \partial x_1}\end{bmatrix}\begin{bmatrix}\triangle_1\triangle_2\\ \triangle_2\triangle_3\\ \triangle_3\triangle_1\end{bmatrix} \tag{4.4}$$

Now, using the matrix notation, the Taylor expansion can be rewritten as:

$$f(\mathbf{x}) = f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)\left[\triangle_i\right] + \frac{1}{2}\mathbf{D}\left[\triangle_i^2\right] + \mathbf{R}\left[\triangle_i\triangle_j\right] + T_3 + \ldots \tag{4.5}$$

Annales Mathematicae et Informaticae, ISSN 1787-5021,
DOI: 10.33039/ami.2021.03.004 - in press

where:

$$\mathbf{D} = \left[\frac{\partial^2 f(\mathbf{x}_0)}{(\partial x_i)^2}\right]^T \qquad \mathbf{R} = \left[\frac{\partial^2 f(\mathbf{x}_0)}{\partial x_i \partial x_j}\right],\ i \neq j \tag{4.6}$$

and $\mathbf{D}$ is a vector, $\mathbf{R}$ is a matrix.

The above given formulation uses just inner products (dot products) instead of matrix multiplications, which leads to significantly faster computation especially on GPU (requires just only one clock) or if SSE instructions are used.

In some cases, it is useful to use the element T_3 of the Taylor expansion, as it enables to represent "inflections" of a function and increase precision of approximation. It leads to a necessity to replace "three-dimensional matrix" used in the T_3 element, see (2.13) by more simple formulation. Originally, the 3D matrix contains 27 values of partial derivatives. However, using the algebraic operations the T_3 element can be expressed as:

$$\begin{aligned} T_3 = \frac{1}{6}\{\ (&\sum_{i=1}^{3} \frac{\partial^3 f(\mathbf{x}_0)}{\partial x_i^3} \triangle_i^3) + 6\frac{\partial^3 f(\mathbf{x}_0)}{\partial x_1 \partial x_2 \partial x_3} \triangle_1 \triangle_2 \triangle_3 \\ &+3\frac{\partial^3 f(\mathbf{x}_0)}{\partial x_1^2 \partial x_2} \triangle_1^2 \triangle_2 + 3\frac{\partial^3 f(\mathbf{x}_0)}{\partial x_1^2 \partial x_3} \triangle_1^2 \triangle_3 \\ &+3\frac{\partial^3 f(\mathbf{x}_0)}{\partial x_1 \partial x_3^2} \triangle_1 \triangle_3^2 + 3\frac{\partial^3 f(\mathbf{x}_0)}{\partial x_1 \partial x_2^2} \triangle_1 \triangle_2^2 \\ &+3\frac{\partial^3 f(\mathbf{x}_0)}{\partial x_2 \partial x_3^2} \triangle_2 \triangle_3^2 + 3\frac{\partial^3 f(\mathbf{x}_0)}{\partial x_2^2 \partial x_3} \triangle_2^2 \triangle_3\} \end{aligned} \tag{4.7}$$

It means, that in the case of a scalar function of three dimensional variables, the T_3 term can be easily evaluated as only 10 values of partial derivatives are computed instead of 27 in the original formulation.

The T_3 element can be formally expressed as:

$$\begin{aligned} T_3 = &\frac{\partial^3 f(\mathbf{x}_0)}{\partial x_1 \partial x_2 \partial x_3} \triangle_1 \triangle_2 \triangle_3 + \frac{1}{6}\sum_{i=1}^{3} \frac{\partial^3 f(\mathbf{x}_0)}{\partial x_i^3} \triangle_i^3 \\ &+\frac{1}{2}\{\frac{\partial^3 f(\mathbf{x}_0)}{\partial x_1^2 \partial x_2} \triangle_1^2 \triangle_2 + \frac{\partial^3 f(\mathbf{x}_0)}{\partial x_1^2 \partial x_3} \triangle_1^2 \triangle_3 \\ &+\frac{\partial^3 f(\mathbf{x}_0)}{\partial x_1 \partial x_3^2} \triangle_1 \triangle_3^2 + \frac{\partial^3 f(\mathbf{x}_0)}{\partial x_1 \partial x_2^2} \triangle_1 \triangle_2^2 \\ &+\frac{\partial^3 f(\mathbf{x}_0)}{\partial x_2 \partial x_3^2} \triangle_2 \triangle_3^2 + \frac{\partial^3 f(\mathbf{x}_0)}{\partial x_2^2 \partial x_3} \triangle_2^2 \triangle_3\} \end{aligned} \tag{4.8}$$

However, in physically oriented applications, it is necessary to use the Taylor expansion also for vector functions, i.e. n-dimensional functions with m-dimensional arguments.

For the vector-vector operations, i.e. if GPU or SSE instructions are used, the T_3 element can be expressed as:

Annales Mathematicae et Informaticae, ISSN 1787-5021,
DOI: 10.33039/ami.2021.03.004 - in press

$$
\begin{aligned}
T_3 =& \frac{\partial^3 f(\mathbf{x}_0)}{\partial x_1 \partial x_2 \partial x_3} \triangle_i \triangle_2 \triangle_3 \\
&+ \frac{1}{6} \begin{bmatrix} \frac{\partial^3 f(\mathbf{x}_0)}{\partial x_1^3} & \frac{\partial^3 f(\mathbf{x}_0)}{\partial x_2^3} & \frac{\partial^3 f(\mathbf{x}_0)}{\partial x_3^3} \end{bmatrix} \begin{bmatrix} \triangle_1^3 \\ \triangle_2^3 \\ \triangle_3^3 \end{bmatrix} \\
&+ \frac{1}{2} \begin{bmatrix} \frac{\partial^3 f(\mathbf{x}_0)}{\partial x_1^2 \partial x_2} & \frac{\partial^3 f(\mathbf{x}_0)}{\partial x_1^2 \partial x_3} & \frac{\partial^3 f(\mathbf{x}_0)}{\partial x_1 \partial x_3^2} & \frac{\partial^3 f(\mathbf{x}_0)}{\partial x_1 \partial x_2^2} & \frac{\partial^3 f(\mathbf{x}_0)}{\partial x_2 \partial x_3^2} & \frac{\partial^3 f(\mathbf{x}_0)}{\partial x_2^2 \partial x_3} \end{bmatrix} \begin{bmatrix} \triangle_1^2 \triangle_2 \\ \triangle_1^2 \triangle_3 \\ \triangle_1 \triangle_3^2 \\ \triangle_1 \triangle_2^2 \\ \triangle_2 \triangle_3^2 \\ \triangle_2^2 \triangle_3 \end{bmatrix}
\end{aligned} \tag{4.9}
$$

It means, that the T_3 element of the Taylor expansion can be implemented using the inner product (dot product) and therefore, it is possible to extend this approach for the Taylor expansion of vector functions.

4.2. Vector Functions

The Taylor expansion can be easily extended for vector functions, i.e.

$$\mathbf{f}(\mathbf{x}) = [f_1(x_1, \ldots, x_m), \ldots, f_n(x_1, \ldots, x_m)]^T \tag{4.10}$$

However, the formulae get more complex in the general case. As there are many applications using three-dimensional representation, i.e. $n = m = 3$, the re-formulation of the Taylor expansion can be simplified using the analogy of the Taylor expansion for scalar functions as follows:

$$\mathbf{f}(\mathbf{x}) = \sum_{i=0}^{\infty} \mathbf{T}_i(\mathbf{x}_0) \tag{4.11}$$

where $\mathbf{T}_i(\mathbf{x}_0)$ are vectors, now. Using the explicit notation

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}_0) + \mathbf{J}(\mathbf{x}_0) \left[\triangle_i\right] + \frac{1}{2} \mathbf{D} \left[\triangle_i^2\right] + \mathbf{R} \left[\triangle_i \triangle_j\right] + T_3 + \ldots \tag{4.12}$$

where:

$$
\begin{aligned}
\begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_3(\mathbf{x}) \end{bmatrix} =& \begin{bmatrix} f_1(\mathbf{x}_0) \\ \vdots \\ f_3(\mathbf{x}_0) \end{bmatrix} + \begin{bmatrix} \frac{\partial f_1(\mathbf{x}_0)}{\partial x_1} & \cdots & \frac{\partial f_1(\mathbf{x}_0)}{\partial x_3} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_3(\mathbf{x}_0)}{\partial x_1} & \cdots & \frac{\partial f_3(\mathbf{x}_0)}{\partial x_3} \end{bmatrix} \begin{bmatrix} \triangle_1 \\ \vdots \\ \triangle_3 \end{bmatrix} \\
&+ \frac{1}{2} \begin{bmatrix} \frac{\partial^2 f_1(\mathbf{x}_0)}{\partial x_1^2} & \cdots & \frac{\partial^2 f_1(\mathbf{x}_0)}{\partial x_3^2} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f_3(\mathbf{x}_0)}{\partial x_1^2} & \cdots & \frac{\partial^2 f_3(\mathbf{x}_0)}{\partial x_3^2} \end{bmatrix} \begin{bmatrix} \triangle_1^2 \\ \vdots \\ \triangle_3^2 \end{bmatrix} +
\end{aligned} \tag{4.13}
$$

Annales Mathematicae et Informaticae, ISSN 1787-5021,
DOI: 10.33039/ami.2021.03.004 - in press

$$\begin{bmatrix} \frac{\partial^2 f_1(\mathbf{x}_0)}{\partial x_1 \partial x_2} & \frac{\partial^2 f_1(\mathbf{x}_0)}{\partial x_2 \partial x_3} & \frac{\partial^2 f_1(\mathbf{x}_0)}{\partial x_3 \partial x_1} \\ \frac{\partial^2 f_2(\mathbf{x}_0)}{\partial x_1 \partial x_2} & \frac{\partial^2 f_2(\mathbf{x}_0)}{\partial x_2 \partial x_3} & \frac{\partial^2 f_2(\mathbf{x}_0)}{\partial x_3 \partial x_1} \\ \frac{\partial^2 f_3(\mathbf{x}_0)}{\partial x_1 \partial x_2} & \frac{\partial^2 f_3(\mathbf{x}_0)}{\partial x_2 \partial x_3} & \frac{\partial^2 f_3(\mathbf{x}_0)}{\partial x_3 \partial x_1} \end{bmatrix} \begin{bmatrix} \triangle_1 \triangle_2 \\ \triangle_2 \triangle_3 \\ \triangle_3 \triangle_1 \end{bmatrix} + T_3 + \dots$$

where: $\left[\triangle_i\right] = \left[\triangle_1, \dots, \triangle_m\right]^T$ and $\left[\triangle_i^2\right] = \left[\triangle_1^2, \dots, \triangle_m^2\right]^T$.

Now, similar approach can be taken as in the Taylor expansion for scalar functions. It means, that in the three-dimensional case, i.e. $f(\mathbf{x}) = f(x_1, ..., x_3)$, the Taylor expansion gets quite simple as the element $\mathbf{T}_2$, which is a vector, has the form:

$$\mathbf{T}_2 = \frac{1}{2} \begin{bmatrix} \nabla^2 f_1(\mathbf{x}_0) \\ \nabla^2 f_2(\mathbf{x}_0) \\ \nabla^2 f_3(\mathbf{x}_0) \end{bmatrix} \begin{bmatrix} \triangle_1^2 \\ \triangle_2^2 \\ \triangle_3^2 \end{bmatrix} + \begin{bmatrix} \frac{\partial^2 f_1(\mathbf{x}_0)}{\partial x_1 \partial x_2} & \frac{\partial^2 f_1(\mathbf{x}_0)}{\partial x_2 \partial x_3} & \frac{\partial^2 f_1(\mathbf{x}_0)}{\partial x_3 \partial x_1} \\ \frac{\partial^2 f_2(\mathbf{x}_0)}{\partial x_1 \partial x_2} & \frac{\partial^2 f_2(\mathbf{x}_0)}{\partial x_2 \partial x_3} & \frac{\partial^2 f_2(\mathbf{x}_0)}{\partial x_3 \partial x_1} \\ \frac{\partial^2 f_3(\mathbf{x}_0)}{\partial x_1 \partial x_2} & \frac{\partial^2 f_3(\mathbf{x}_0)}{\partial x_2 \partial x_3} & \frac{\partial^2 f_3(\mathbf{x}_0)}{\partial x_3 \partial x_1} \end{bmatrix} \begin{bmatrix} \triangle_1 \triangle_2 \\ \triangle_2 \triangle_3 \\ \triangle_3 \triangle_1 \end{bmatrix} \tag{4.14}$$

The above given formulation uses just three inner products (dot products) instead of matrix multiplications, which leads to significantly faster computation especially on GPU (requires just only one clock) or if SSE instructions are used.

In some cases, it is useful to use the element $\mathbf{T}_3$ of the Taylor expansion, as it enables to represent "inflections" of a function and increase precision of approximation. It leads to a necessity to replace "three-dimensional matrix" used in the $\mathbf{T}_3$ element, see (2.13) by simpler formulation. In the original formulation, the 3D matrix contains 27 values of partial derivatives.

However, using the algebraic operations the $\mathbf{T}_3$, which is a vector, the k^{th} element, $k = 1, \dots, 3$ can be expressed as:

$$\begin{aligned} T_{3_k} = \frac{1}{6} \{ \sum_{i=1}^{3} \frac{\partial^3 f_k(\mathbf{x}_0)}{\partial x_i^3} \triangle_i^3 + 6 \frac{\partial^3 f_k(\mathbf{x}_0)}{\partial x_1 \partial x_2 \partial x_3} \triangle_1 \triangle_2 \triangle_3 \\ +3 \frac{\partial^3 f_k(\mathbf{x}_0)}{\partial x_1^2 \partial x_2} \triangle_1^2 \triangle_2 + 3 \frac{\partial^3 f_k(\mathbf{x}_0)}{\partial x_1^2 \partial x_3} \triangle_1^2 \triangle_3 \\ +3 \frac{\partial^3 f_k(\mathbf{x}_0)}{\partial x_1 \partial x_3^2} \triangle_1 \triangle_3^2 + 3 \frac{\partial^3 f_k(\mathbf{x}_0)}{\partial x_1 \partial x_2^2} \triangle_1 \triangle_2^2 \\ +3 \frac{\partial^3 f_k(\mathbf{x}_0)}{\partial x_2 \partial x_3^2} \triangle_2 \triangle_3^2 + 3 \frac{\partial^3 f_k(\mathbf{x}_0)}{\partial x_2^2 \partial x_3} \triangle_2^2 \triangle_3 \} \end{aligned} \tag{4.15}$$

In the case of a vector function of three dimensional variables, the $\mathbf{T}_3$ term can be easily evaluated as only 3×10 values of partial derivatives are computed instead of 3×27 in the original formulation.

The k^{th} element, $k = 1, \dots, 3$, of the $\mathbf{T}_3$ vector element can be formally expressed as:

Annales Mathematicae et Informaticae, ISSN 1787-5021,
DOI: 10.33039/ami.2021.03.004 - in press

$$
\begin{aligned}
T_{3_k} = & \frac{\partial^3 f_k(\mathbf{x}_0)}{\partial x_1 \partial x_2 \partial x_3} \triangle_i \triangle_2 \triangle_3 + \frac{1}{6} \sum_{i=1}^{3} \frac{\partial^3 f_k(\mathbf{x}_0)}{\partial x_i^3} \triangle_i^3 \\
& \frac{1}{2} \{ \frac{\partial^3 f_k(\mathbf{x}_0)}{\partial x_1^2 \partial x_2} \triangle_1^2 \triangle_2 + \frac{\partial^3 f_k(\mathbf{x}_0)}{\partial x_1^2 \partial x_3} \triangle_1^2 \triangle_3 \\
& + \frac{\partial^3 f_k(\mathbf{x}_0)}{\partial x_1 \partial x_3^2} \triangle_1 \triangle_3^2 + \frac{\partial^3 f_k(\mathbf{x}_0)}{\partial x_1 \partial x_2^2} \triangle_1 \triangle_2^2 \\
& + \frac{\partial^3 f_k(\mathbf{x}_0)}{\partial x_2 \partial x_3^2} \triangle_2 \triangle_3^2 + \frac{\partial^3 f_k(\mathbf{x}_0)}{\partial x_2^2 \partial x_3} \triangle_2^2 \triangle_3 \}
\end{aligned} \tag{4.16}
$$

The vector $\mathbf{T}_3$ of the Taylor expansion can be expressed using standard linear algebra notation, instead of using three dimensional matrix notation, as:

$$
\begin{aligned}
\mathbf{T}_3 = & \begin{bmatrix} \frac{\partial^3 f_1(\mathbf{x}_0)}{\partial x_1 \partial x_2 \partial x_3} & \frac{\partial^3 f_2(\mathbf{x}_0)}{\partial x_1 \partial x_2 \partial x_3} & \frac{\partial^3 f_3(\mathbf{x}_0)}{\partial x_1 \partial x_2 \partial x_3} \end{bmatrix} \triangle_1 \triangle_2 \triangle_3 \\
& + \frac{1}{6} \begin{bmatrix} \frac{\partial^3 f_1(\mathbf{x}_0)}{\partial x_1^3} & \frac{\partial^3 f_1(\mathbf{x}_0)}{\partial x_2^3} & \frac{\partial^3 f_1(\mathbf{x}_0)}{\partial x_3^3} \\ \frac{\partial^3 f_2(\mathbf{x}_0)}{\partial x_1^3} & \frac{\partial^3 f_2(\mathbf{x}_0)}{\partial x_2^3} & \frac{\partial^3 f_2(\mathbf{x}_0)}{\partial x_3^3} \\ \frac{\partial^3 f_3(\mathbf{x}_0)}{\partial x_1^3} & \frac{\partial^3 f_3(\mathbf{x}_0)}{\partial x_2^3} & \frac{\partial^3 f_3(\mathbf{x}_0)}{\partial x_3^3} \end{bmatrix} \begin{bmatrix} \triangle_1^3 \\ \triangle_2^3 \\ \triangle_3^3 \end{bmatrix} \\
& + \frac{1}{2} \begin{bmatrix} \frac{\partial^3 f_1(\mathbf{x}_0)}{\partial x_1^2 \partial x_2} & \frac{\partial^3 f_1(\mathbf{x}_0)}{\partial x_1^2 \partial x_3} & \frac{\partial^3 f_1(\mathbf{x}_0)}{\partial x_1 \partial x_3^2} & \frac{\partial^3 f_1(\mathbf{x}_0)}{\partial x_1 \partial x_2^2} & \frac{\partial^3 f_1(\mathbf{x}_0)}{\partial x_2 \partial x_3^2} & \frac{\partial^3 f_1(\mathbf{x}_0)}{\partial x_2^2 \partial x_3} \\ \frac{\partial^3 f_2(\mathbf{x}_0)}{\partial x_1^2 \partial x_2} & \frac{\partial^3 f_2(\mathbf{x}_0)}{\partial x_1^2 \partial x_3} & \frac{\partial^3 f_2(\mathbf{x}_0)}{\partial x_1 \partial x_3^2} & \frac{\partial^3 f_2(\mathbf{x}_0)}{\partial x_1 \partial x_2^2} & \frac{\partial^3 f_2(\mathbf{x}_0)}{\partial x_2 \partial x_3^2} & \frac{\partial^3 f_2(\mathbf{x}_0)}{\partial x_2^2 \partial x_3} \\ \frac{\partial^3 f_3(\mathbf{x}_0)}{\partial x_1^2 \partial x_2} & \frac{\partial^3 f_3(\mathbf{x}_0)}{\partial x_1^2 \partial x_3} & \frac{\partial^3 f_3(\mathbf{x}_0)}{\partial x_1 \partial x_3^2} & \frac{\partial^3 f_3(\mathbf{x}_0)}{\partial x_1 \partial x_2^2} & \frac{\partial^3 f_3(\mathbf{x}_0)}{\partial x_2 \partial x_3^2} & \frac{\partial^3 f_3(\mathbf{x}_0)}{\partial x_2^2 \partial x_3} \end{bmatrix} \begin{bmatrix} \triangle_1^2 \triangle_2 \\ \triangle_1^2 \triangle_3 \\ \triangle_1 \triangle_3^2 \\ \triangle_1 \triangle_2^2 \\ \triangle_2 \triangle_3^2 \\ \triangle_2^2 \triangle_3 \end{bmatrix}
\end{aligned} \tag{4.17}
$$

If matrix notation is used, the $\mathbf{T}_3$ element can be expressed as:

$$
\mathbf{T}_3 = \mathbf{U} \ (\triangle_1 \triangle_2 \triangle_3) + \mathbf{V} \begin{bmatrix} \triangle_1^3 \\ \triangle_2^3 \\ \triangle_3^3 \end{bmatrix} + \mathbf{W} \begin{bmatrix} \triangle_1^2 \triangle_2 \\ \triangle_1^2 \triangle_3 \\ \triangle_1 \triangle_3^2 \\ \triangle_1 \triangle_2^2 \\ \triangle_2 \triangle_3^2 \\ \triangle_2^2 \triangle_3 \end{bmatrix} \tag{4.18}
$$

where:

$$
\begin{aligned}
\mathbf{U} = & \begin{bmatrix} \frac{\partial^3 f_1(\mathbf{x}_0)}{\partial x_1 \partial x_2 \partial x_3} & \frac{\partial^3 f_2(\mathbf{x}_0)}{\partial x_1 \partial x_2 \partial x_3} & \frac{\partial^3 f_3(\mathbf{x}_0)}{\partial x_1 \partial x_2 \partial x_3} \end{bmatrix} \\
\mathbf{V} = & \begin{bmatrix} \frac{\partial^3 f_1(\mathbf{x}_0)}{\partial x_1^3} & \frac{\partial^3 f_1(\mathbf{x}_0)}{\partial x_2^3} & \frac{\partial^3 f_1(\mathbf{x}_0)}{\partial x_3^3} \\ \frac{\partial^3 f_2(\mathbf{x}_0)}{\partial x_1^3} & \frac{\partial^3 f_2(\mathbf{x}_0)}{\partial x_2^3} & \frac{\partial^3 f_2(\mathbf{x}_0)}{\partial x_3^3} \\ \frac{\partial^3 f_3(\mathbf{x}_0)}{\partial x_1^3} & \frac{\partial^3 f_3(\mathbf{x}_0)}{\partial x_2^3} & \frac{\partial^3 f_3(\mathbf{x}_0)}{\partial x_3^3} \end{bmatrix}
\end{aligned} \tag{4.19}
$$

Annales Mathematicae et Informaticae, ISSN 1787-5021,
DOI: 10.33039/ami.2021.03.004 - in press

$$\mathbf{W} = \begin{bmatrix} \frac{\partial^3 f_1(\mathbf{x}_0)}{\partial x_1^2 \partial x_2} & \frac{\partial^3 f_1(\mathbf{x}_0)}{\partial x_1^2 \partial x_3} & \frac{\partial^3 f_1(\mathbf{x}_0)}{\partial x_1 \partial x_3^2} & \frac{\partial^3 f_1(\mathbf{x}_0)}{\partial x_1 \partial x_2^2} & \frac{\partial^3 f_1(\mathbf{x}_0)}{\partial x_2 \partial x_3^2} & \frac{\partial^3 f_1(\mathbf{x}_0)}{\partial x_2^2 \partial x_3} \\ \frac{\partial^3 f_2(\mathbf{x}_0)}{\partial x_1^2 \partial x_2} & \frac{\partial^3 f_2(\mathbf{x}_0)}{\partial x_1^2 \partial x_3} & \frac{\partial^3 f_2(\mathbf{x}_0)}{\partial x_1 \partial x_3^2} & \frac{\partial^3 f_2(\mathbf{x}_0)}{\partial x_1 \partial x_2^2} & \frac{\partial^3 f_2(\mathbf{x}_0)}{\partial x_2 \partial x_3^2} & \frac{\partial^3 f_2(\mathbf{x}_0)}{\partial x_2^2 \partial x_3} \\ \frac{\partial^3 f_3(\mathbf{x}_0)}{\partial x_1^2 \partial x_2} & \frac{\partial^3 f_3(\mathbf{x}_0)}{\partial x_1^2 \partial x_3} & \frac{\partial^3 f_3(\mathbf{x}_0)}{\partial x_1 \partial x_3^2} & \frac{\partial^3 f_3(\mathbf{x}_0)}{\partial x_1 \partial x_2^2} & \frac{\partial^3 f_3(\mathbf{x}_0)}{\partial x_2 \partial x_3^2} & \frac{\partial^3 f_3(\mathbf{x}_0)}{\partial x_2^2 \partial x_3} \end{bmatrix}$$

It means, that the Taylor expansion can be written in the form containing only vectors and matrices.

$$\begin{aligned} \mathbf{f}(\mathbf{x}) =& \mathbf{f}(\mathbf{x}_0) + \mathbf{J}(\mathbf{x}_0) \left[\triangle_i\right] \qquad \# \textit{ linear case} \\ &+ \frac{1}{2}\mathbf{D} \left[\triangle_i^2\right] + \mathbf{R} \left[\triangle_i \triangle_j\right] \quad \# \textit{ quadratic case} \\ &\mathbf{U} \ (\triangle_1 \triangle_2 \triangle_3) + \mathbf{V} \begin{bmatrix} \triangle_1^3 \\ \triangle_2^3 \\ \triangle_3^3 \end{bmatrix} + \mathbf{W} \begin{bmatrix} \triangle_1^2 \triangle_2 \\ \triangle_1^2 \triangle_3 \\ \triangle_1 \triangle_3^2 \\ \triangle_1 \triangle_2^2 \\ \triangle_2 \triangle_3^2 \\ \triangle_2^2 \triangle_3 \end{bmatrix} \quad \# \textit{ cubic case} + \dots \end{aligned} \tag{4.20}$$

It can be seen, that the above given formulae are simple, easy to implement efficiently, especially if GPU or SSE instructions are used. The presented approach can be applied also for the case, when $n \neq m$, in general. However, it should be noted that size of some vectors and matrices grows quadratic.

5. Application

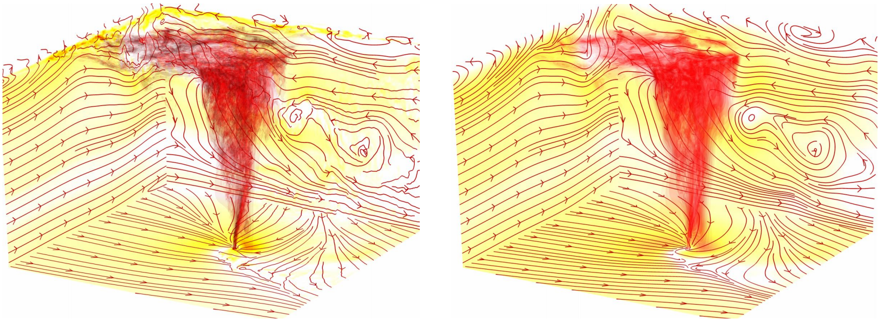
Visualization of 3D vector fields, i.e. fluid flow and electromagnetic fields, uses the Taylor expansion to approximate acquired data (measured or obtained from a simulation). If the data are large, the approximation is also used for data reduction, while keeping the important features of the vector data [14]. If the Taylor expansion is used with Radial Basis Functions (RBF) [12], it is possible to obtain analytical function describing the given vector field data respecting critical points, vector field second derivatives [13] [15].

The Taylor expansion was used for radial basis function (RBF) approximation of the EF5 Tornado data [1] using second derivatives of the Taylor expansion [16]. This led to high compression ratio, see illustrative images in the Tab.1 [16], and analytical form describing the tornado fluid flow in the analytical form for the flow speed as $\mathbf{v} = \mathbf{f}(\mathbf{x})$.

As the second derivatives were used, the proposed new formulation of the Taylor expansion offers simple formal structure, efficient computation and significant speed-up of computation. The formulation is convenient for GPU implementation which offers high speed-up due to parallelism available.

[1] Data set of EF5 tornado courtesy of Leigh Orf from Cooperative Institute for Meteorological Satellite Studies, University of Wisconsin, Madison, WI,USA

Annales Mathematicae et Informaticae, ISSN 1787-5021,
DOI: 10.33039/ami.2021.03.004 - in press



Original data	Approximated data compression ratio $7.10^3 : 1$

Table 1: Tornado data and its approximation using second derivatives (taken from [16])

6. Conclusion

This paper describes a new re-formulation of the Taylor expansion for scalar and vector functions for the multidimensional case and its optimization for the 3D case. This new re-formulation enables representation of the third order of approximation using standard linear algebra formalism, without tensor notation use. The proposed approach leads to significant speed-up of computation, see chapter 4. In the case of the GPU or SSE implementation additional speed up can be expected, especially due to fast vector-vector operations and native parallelism on GPU. Specialized version for the three dimensional case is presented, which is simple to implement as well.

The presented approach can be directly applied to 3D flow or electromagnetic fields computation and simulation. It can be extended to higher dimensions, however, the complexity of formulae grows quadratic. However, the expected speed up will grow with a dimension against "*standard*" implementation.

The influence of the second derivations was explored in [16]. It led to significant improvements for vector fields approximation, i.e. compression ratio and precision. In future, the influence of the *cubic part* of the Taylor expansion is to be studied, as inclusion of points of inflections and curvatures of vector fields should lead to higher compression ratio.

Acknowledgment The author would like to thank to colleagues, especially Michal Smolik for image generation, and to students, especially to Jakub Vasta, Filip Hacha and Martin Cervenka, at the University of West Bohemia (Czech Republic) for hints and suggestions. Thanks belong also to colleagues at Shandong University and Zhejiang University (China) for their critical comments and constructive suggestions, and to anonymous reviewers for their valuable comments and hints provided.

Annales Mathematicae et Informaticae, ISSN 1787-5021,
DOI: 10.33039/ami.2021.03.004 - in press

References

[1] A. Alvandi, M. Paripour: *The combined reproducing kernel method and Taylor series for handling nonlinear Volterra integro-differential equations with derivative type kernel*, Applied Mathematics and Computation 355 (2019), pp. 151–160, issn: 0096-3003, doi: 10.1016/j.amc.2019.02.023.

[2] A. Baeza, S. Boscarino, P. Mulet, G. Russo, D. Zorío: *Approximate Taylor methods for ODEs*, Computers and Fluids 159 (2017), pp. 156–166, issn: 0045-7930, doi: 10.1016/j.compfluid.2017.10.001.

[3] L. Bernal-Gonzalez, H. Cabana-Mendez, G. Munoz-Fernandez, J. Seoane-Sepulveda: *Universality of sequences of operators related to Taylor series*, Journal of Mathematical Analysis and Applications 474.1 (2019), pp. 480–491, issn: 0022-247X, doi: 10.1016/j.jmaa.2019.01.056.

[4] T. Breiten, K. Kunisch, L. Pfeiffer: *Taylor expansions of the value function associated with a bilinear optimal control problem*, Annales de l'Institut Henri Poincare (C) Analyse Non Lineaire 36.5 (2019), pp. 1361–1399, issn: 0294-1449, doi: 10.1016/j.anihpc.2019.01.001.

[5] C.-Z. Dong, O. Celik, F. Catbas, E. Obrien, S. Taylor: *A robust vision-based method for displacement measurement under adverse environmental factors using spatio-temporal context learning and Taylor approximation*, Sensors (Switzerland) 19.14 (2019), issn: 1424-8220, doi: 10.3390/s19143197.

[6] N. Hu, B. Duan, W. Xu, J. Zhou: *A new interval pattern analysis method of array antennas based on Taylor expansion*, IEEE Transactions on Antennas and Propagation 65.11 (2017), pp. 6151–6156, issn: 0018-926X, doi: 10.1109/TAP.2017.2754458.

[7] W. Hu, Y. Gu, C.-M. Fan: *A meshless collocation scheme for inverse heat conduction problem in three-dimensional functionally graded materials*, Engineering Analysis with Boundary Elements 114 (2020), pp. 1–7, issn: 0955-7997, doi: 10.1016/j.enganabound.2020.02.001.

[8] R. Jedynak: *New facts concerning the approximation of the inverse Langevin function*, Journal of Non-Newtonian Fluid Mechanics 249 (2017), pp. 8–25, issn: 0377-0257, doi: 10.1016/j.jnnfm.2017.09.003.

[9] K. Konno, Q. Chen, R. Burkholder: *Efficiency improvement with a recursive Taylor expansion of Bessel functions for layered media Green's function*, 2017 IEEE Antennas and Propagation Society International Symposium, Proceedings 2017-January (2017), pp. 1355–1356, doi: 10.1109/APUSNCURSINRSM.2017.8072720.

[10] Y. Liu, S. Liu, F. Zhan, X. Zhang: *Firing patterns of the modified Hodgkin–Huxley models subject to Taylor's formula*, Physica A: Statistical Mechanics and its Applications 547 (2020), issn: 0378-4371, doi: 10.1016/j.physa.2020.124405.

[11] K. Neamprem, A. Klangrak, H. Kaneko: *Taylor-series expansion methods for multivariate hammerstein integral equations*, IAENG International Journal of Applied Mathematics 47.4 (2017), pp. 437–441, issn: 1992-9978.

[12] V. Skala, M. Smolik: *A new approach to vector field interpolation, classification and robust critical points detection using radial basis functions*, Advances in Intelligent Systems and Computing 765 (2019), pp. 109–115, issn: 2194-5357, doi: 10.1007/978-3-319-91192-2_12.

[13] M. Smolik, V. Skala: *Classification of Critical Points Using a Second Order Derivative*, Procedia Computer Science 108 (2017), pp. 2373–2377, doi: 10.1016/j.procs.2017.05.271.

Annales Mathematicae et Informaticae, ISSN 1787-5021,
DOI: 10.33039/ami.2021.03.004 - in press

[14] M. Smolik, V. Skala: *Radial basis function and multi-level 2D vector field approximation*, Mathematics and Computers in Simulation 181 (2021), pp. 522–538, issn: 0378-4754, doi: 10.1016/j.matcom.2020.10.009.

[15] M. Smolik, V. Skala: *Vector field second order derivative approximation and geometrical characteristics*, LNCS 10404 (2017), pp. 148–158, issn: 0302-9743, doi: 10.1007/978-3-319-62392-4_11.

[16] M. Smolik, V. Skala, Z. Majdisova: *3D vector field approximation and critical points reduction using radial basis functions*, English, International Journal of Mechanics 13 (2019), pp. 100–103, issn: 1998-4448.

[17] X. Wang, Y. Liu, J. Ouyang: *A meshfree collocation method based on moving Taylor polynomial approximation for high order partial differential equations*, Engineering Analysis with Boundary Elements 116 (2020), pp. 77–92, issn: 0955-7997, doi: 10.1016/j.enganabound.2020.04.002.

[18] Q. Wen, F. Xu, J.-H. Yong: *Real-time 3D eye performance reconstruction for RGBD cameras*, IEEE Transactions on Visualization and Computer Graphics 23.12 (2017), pp. 2586–2598, issn: 1077-2626, doi: 10.1109/TVCG.2016.2641442.

[19] Wikipedia: *Taylor series*, Accessed: 2020-05-04, 2020, url: https://en.wikipedia.org/wiki/Taylor_series.

[20] E. Wobbes, M. Möller, V. Galavi, C. Vuik: *Taylor least squares reconstruction technique for material point methods*, Proceedings of the 6th European Conference on Computational Mechanics: Solids, Structures and Coupled Problems, ECCM 2018 and 7th European Conference on Computational Fluid Dynamics, ECFD 2018 (2020), pp. 806–817.

[21] D. Zézé, M. Potier-Ferry, Y. Tampango: *Multi-point Taylor series to solve differential equations*, Discrete and Continuous Dynamical Systems - Series S 12.6 (2019), pp. 1791–1806, issn: 1937-1632, doi: 10.3934/dcdss.2019118.