E11-2018-43

Cs. Török [1,*]

SPEEDUP OF INTERPOLATING SPLINE CONSTRUCTION

[1] P. J. Šafárik University, Košice, the Slovak Republic
[*] E-mail: csaba.torok@upjs.sk

Торок Ч. E11-2018-43
Ускорение конструкции интерполяционных сплайнов

Предложен эффективный последовательный алгоритм для вычисления коэффициентов кубического сплайна с целью уменьшения времени счета. Построена трехдиагональная система линейных уравнений с уменьшенной размерностью, основанная на недавно полученном соотношении для коэффициентов однородных кубических сплайнов класса C^2 в нечетных узлах сетки. Коэффициенты в четных узлах вычисляются по явной формуле. Благодаря уменьшению размерности и простой формуле число делений в алгоритме сокращено в два раза, что приводит к ускорению расчетов. Предложен общий подход к диадическому уменьшению размерности, вследствие чего размер трехдиагональных систем постепенно уменьшается до половины, четверти и т. д.

Работа выполнена в Лаборатории информационных технологий ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна, 2018

Török Cs. E11-2018-43
Speedup of Interpolating Spline Construction

The article introduces an efficient sequential algorithm for computing spline coefficients. The aim is to decrease the computational time. A reduced-size tridiagonal system of linear equations is constructed based on a recently derived relation for the unknown coefficients of uniform cubic splines of class C^2 at the odd grid points. The even coefficients are computed from an explicit formula. Thanks to the half-size system and use of a simple formula, the suggested new sequential algorithm needs less division than the traditional one and it results in a non-negligible computational speedup. Finally, a general approach to dyadic reduction of the dimensionality of tridiagonal linear systems is proposed in consequence of which the size of the systems gradually shrinks to half, quarter, etc.

The investigation has been performed at the Laboratory of Information Technologies, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna, 2018

INTRODUCTION

Interpolation of discrete data given at equispaced points is one of the most common tasks in scientific computing and computer graphics due to its application in modeling of physical processes, CAD systems and video games, therefore we need to perform this task efficiently.

Consider an interpolating uniform cubic clamped spline of class C^2 defined over an equispaced grid $\{x_0,\, x_1, \ldots, x_{N+1}\}$, $x_i = x_0 + hi$, $i = 1, \ldots, N+1$ by values $\{y_0,\, y_1, \ldots, y_{N+1}\}$ and $\{d_0, d_{N+1}\}$.

We show that the unknown first derivatives d_i at odd grid points $x_2, x_4, \ldots x_{N-1}$, needed for the spline construction, can be computed as a solution to a reduced tridiagonal system of linear equations

$$\begin{bmatrix} -14 & 1 & 0 & & & \\ 1 & -14 & 1 & & & \\ 0 & 1 & -14 & & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & & & -14 & 1 \\ & & & & 1 & -14 \end{bmatrix} \begin{bmatrix} d_2 \\ d_4 \\ d_6 \\ \vdots \\ d_{N-3} \\ d_{N-1} \end{bmatrix} =$$

$$= \begin{bmatrix} \frac{3}{h}(y_4 - y_0) - \frac{12}{h}(y_3 - y_1) - d_0 \\ \frac{3}{h}(y_6 - y_2) - \frac{12}{h}(y_5 - y_3) \\ \frac{3}{h}(y_8 - y_4) - \frac{12}{h}(y_7 - y_5) \\ \vdots \\ \frac{3}{h}(y_{N-1} - y_{N-5}) - \frac{12}{h}(y_{N-2} - y_{N-4}) \\ \frac{3}{h}(y_{N+1} - y_{N-3}) - \frac{12}{h}(y_N - y_{N-2}) - d_{N+1} \end{bmatrix}. \quad (1)$$

This system corresponds to odd N. The system for even N differs only in the last equation. The remaining coefficients $d_1, d_3, \ldots, d_N$ are computed by an explicit formula

$$d_i = -\frac{1}{4}\left(\frac{3(y_{i+1} - y_{i-1})}{h} - d_{i+1} - d_{i-1}\right), \qquad i = 1, 3, \ldots, N. \quad (2)$$

We emphasize that the number of equations in (1) is two times less than in the corresponding system of the traditional approach. Hence and thanks to the simple form of (2), we can expect that the spline construction based on (1) and (2) is computationally more effective. Really, we will see that the total number of expensive divisions in the computational algorithm of the new approach is substantially less. Notice further that the information of the original system is preserved due to more complex right-hand sides of the reduced one. However, this fact does not increase the computational costs due to the instruction level parallelism property of modern processors. As a result, we will show that the reduced system based algorithm provides approximately a 1.6x assessed and measured speedup over the standard one in sequential computation.

Fast algorithms for solving tridiagonal systems are critical. The classical way to achieve speedup is parallelization, where the tridiagonal matrix elements are eliminated by tricky patterns, such as partition [2]. We have changed the system's model equation and arrived at 1.6x speedup from sequential parameter estimation. We often study noisy data of complex structure that cannot be smoothed by a polynomial. Preferring models with interpolating properties, allowing for easier interpretation, we omitted B-splines and smoothing splines. To approximate data with piecewise polynomials, smooth transition should be granted between the components. First, we studied two-part models with smooth connection thanks to shared interpolating points [8]. Since derivatives appear passing to the limit in this approach, we turned to Hermite splines that in cubic case, by default, are of class C^1. We have got the reduced system (1) as a generalization and application of a recently gained result within a two-part model [9] and using Hermite splines. During the analysis of this reduced system we derived and rediscovered the full tridiagonal system for uniform cubic clamped splines

$$\begin{bmatrix} 4 & 1 & 0 & & & \\ 1 & 4 & 1 & & & \\ 0 & 1 & 4 & & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & & & 4 & 1 \\ & & & & 1 & 4 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{N-1} \\ d_N \end{bmatrix} = \begin{bmatrix} \frac{3}{h}(y_2 - y_0) - d_0 \\ \frac{3}{h}(y_3 - y_1) \\ \frac{3}{h}(y_4 - y_2) \\ \vdots \\ \frac{3}{h}(y_N - y_{N-2}) \\ \frac{3}{h}(y_{N+1} - y_{N-1}) - d_{N+1} \end{bmatrix}, \qquad (3)$$

also using Hermite splines. It is interesting that the classical works on splines such as [1, 4, 5] or even [7] use a different approach to computation of $\{d_1, \ldots, d_N\}$ at the grid points. We have found the more intuitive and simpler approach to the derivation of system (3) for computation of the coefficients d_i based on Hermite splines after all in [6], however we do not know who has proposed it first.

The main idea that led to the proposed tridiagonal systems comes from [9], where we have proved that a two-component uniform cubic C^1-class Hermite spline will be of class C^2, if its coefficients are computed from a corresponding four-degree polynomial. This geometrical approach using quartic and cubic polynomials could not be applied to further reduction of (1). Nevertheless, we succeeded in dyadic reduction of tridiagonal systems thanks to algebraic abstraction using matrices.

The structure of the paper is as follows. The reduced tridiagonal system for solving the spline coefficients at odd grid points is derived in Section 1, where a new algorithm for computation of all coefficients is described as well. The next section shows the efficiency of the proposed algorithm based on the assessed and measured speedup. Further dyadic reduction of tridiagonal systems is described in Section 3. Formulas for one-component cubic Hermite splines and quartic polynomials with respect to function values and derivatives are given in Appendix.

1. THE REDUCED TRIDIAGONAL SYSTEM

First, we show some interrelations between quartic polynomials and cubic splines deriving three relations, based on which we set up a reduced tridiagonal system and an efficient computational algorithm for solving the coefficients of C^2-class uniform cubic clamped splines. We will consider gradually three, four, five and then any number of grid points.

Consider three values y_0, y_1, y_2 and three first derivatives d_0, d_1, d_2 at points $u_0 < u_1 < u_2$. These values and derivatives define a quartic polynomial

$$f(x) \equiv f(x;\, u_0, u_1, u_2, y_0, y_1, y_2, d_0, d_2)$$

and a two-component Hermite spline $\{H_1, H_2\}$, that is by default of class C^1, with components

$$H_1(x) \equiv H_1(x;\, u_0, u_1, y_0, y_1, d_0, d_1) \text{ and } H_2(x) \equiv H_2(x;\, u_1, u_2, y_1, y_2, d_1, d_2),$$

where the quartic polynomial f and the one-component Hermite spline H are defined in the Appendix. The next result, that sheds light on an interrelation between polynomials of degree three and four, shows under what condition this spline $\{H_1, H_2\}$ will be of class C^2.

Lemma 1. *The left and right bicubic Hermite spline components H_1, H_2 have equal second derivatives at u_1*

$$H_1''(u_1) = H_2''(u_1),$$

if d_1 *is computed from* f, *i.e.,*

$$d_1 = f'(u_1;\, u_0, u_1, u_2, y_0, y_1, y_2, d_0, d_2)$$

and $u_1 = u_{10} \equiv \dfrac{u_0 + u_1}{2}$.

This result was gained in [9]. To be self-contained we give here a different, direct proof. Naturally, this result can be proved without using quartic polynomials. However, just their geometrical perspicuity inspired the upcoming second and third relations that ultimately led to the reduced system proposed here as a base for an efficient way of computing uniform cubic splines.

Proof. Let $Z_1 = (x - u_0)^2(x - u_1)^2$, $Z_2 = (x - u_1)^2(x - u_2)$ and a_4 be the leading coefficient of $f(x)$. If $d_1 = f'(u_1)$, then

$$H_1(x) + Z_1(x)a_4 = H_2(x) + Z_2(x)a_4 = f(x).$$

Therefore, $H_1''(u_1) = H_2''(u_1)$ is equivalent to $Z_1''(u_1) = Z_2''(u_1)$. From this last equation we get $(u_0 - u_2)(u_0 - 2u_1 + u_2) = 0$, which actually ends the proof. □

We mention that at u_{10} the above second derivative equals $\frac{1}{h}(d_0 - d_2) + \frac{3}{h^2}(y_0 - 2y_1 + y_2)$, the first derivative is

$$d_1 = f'(u_1) \equiv \frac{1}{4}\Big(\frac{3}{h}(y_2 - y_0) - d_2 - d_0\Big), \tag{4}$$

from which one can get the model equation for the full system and the auxiliary formula of the reduced system, and for u_{10} the leading coefficient $a_4 = \frac{1}{2h^4}\left(-y_0 + 2y_1 - y_2 + \frac{h}{2}(d_2 - d_0)\right)$.

So, given an equispaced grid $[u_0, u_1, u_2]$, $u_1 = u_0 + h$, $u_2 = u_0 + 2h$, y_0, y_1, y_2 and d_0, d_2 at these points, then a quartic polynomial $f(x)$ is uniquely defined. If d_1 is computed from $f(x)$ as its first derivative at u_{10}, and based on these values we construct H_1, H_2, then the two-component Hermite spline $\{H_1, H_2\}$ will be a uniform cubic clamped spline of class C^2.

In addition to formula (4), we need two further relations as model equations that can be derived on equispaced four- and five-point grids.

Consider the construction task of a three-component uniform cubic spline of class C^2 over an equispaced grid $[u_0, u_1, u_2, u_3]$, $u_i = u_0 + (i-1)h$, $i = 1, 2, 3$, with given y_0, y_1, y_2, y_3, d_0, d_3. Let us derive an equation for the inner d_2 using a quartic polynomial. Based on the given data, we construct first a quartic polynomial $f(x) \equiv f(x;\, u_0, u_1, u_2, y_0, y_1, y_2, d_0, d_2)$ and a one-component Hermite spline $H_3(x) \equiv H_3(x;\, u_2, u_3, y_2, y_3, d_2, d_3)$. The further two components H_1, H_2 are constructed as before using $f'(u_1)$; therefore, based on Lemma 1 it holds $H_1''(u_1) = H_2''(u_1)$. The spline $\{H_1, H_2, H_3\}$ will be a clamped spline of class C^2, if $H_2''(u_2) = H_3''(u_2)$, whence we get an equation for the parameter d_2

$$d_0 - 15d_2 - 4d_3 = \frac{3}{h}(y_2 - y_0) - \frac{12}{h}(y_3 - y_1). \tag{5}$$

By using two quartic polynomials and similar reasoning for the construction of a four-component uniform cubic spline of class C^2 over an equispaced five-point grid $[u_0, \ldots, u_4]$, $u_i = u_0 + (i-1)h$, $i = 1, \ldots, 4$, with given $y_0, \ldots, y_4$ and d_0, d_4, we derived in [9] the following equation for d_2

$$d_0 - 14d_2 + d_4 = \frac{3}{h}(y_4 - y_0) - \frac{12}{h}(y_3 - y_1). \tag{6}$$

The interesting point here is that the three unknown spline parameters d_1, d_2, d_3 are not gained as a solution to a system of three equations as usual, but we get d_2 by solving the equation (6), and the other two are computed from f_1 and f_2 using the simple formula (4) and analogously $d_3 = f'(u_3) \equiv \frac{1}{4}\Big(\frac{3}{h}(y_4 - y_2) - d_4 - d_2\Big)$.

All is prepared for moving on to the general case. It turned out that based on the relations (4), (5) and (6) an efficient solver for the unknown d_i coefficients of uniform clamped splines with any number of components can be proposed. As we show in the next section, the speedup is approximately 1.6.

Consider the task of constructing a C^2-class uniform cubic clamped spline over the grid $[u_0, u_1, \ldots, u_{N+1}]$, $u_i = u_0 + (i-1)h$, $i = 1, \ldots, N+1$, based on the given $y_0, \ldots, y_{N+1}$ and d_0, d_{N+1} at grid points.

Theorem 1. *The tridiagonal system*

$$\begin{bmatrix} -14 & 1 & 0 & & & \\ 1 & -14 & 1 & & & \\ 0 & 1 & -14 & & & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & & & -14 & 1 \\ & & & & 1 & \mu \end{bmatrix} \begin{bmatrix} d_2 \\ d_4 \\ d_6 \\ \vdots \\ d_{\nu-2} \\ d_\nu \end{bmatrix} =$$

$$= \begin{bmatrix} \frac{3}{h}(y_4 - y_0) - \frac{12}{h}(y_3 - y_1) - d_0 \\ \frac{3}{h}(y_6 - y_2) - \frac{12}{h}(y_5 - y_3) \\ \frac{3}{h}(y_8 - y_4) - \frac{12}{h}(y_7 - y_5) \\ \vdots \\ \frac{3}{h}(y_\nu - y_{\nu-4}) - \frac{12}{h}(y_{\nu-1} - y_{\nu-3}) \\ \frac{3}{h}(y_{\nu+\tau} - y_{\nu-2}) - \frac{12}{h}(y_{\nu+1} - y_{\nu-1}) - \eta d_{N+1} \end{bmatrix}, \tag{7}$$

where

$$\begin{array}{ll} \mu=-14,\ \tau=2,\ \eta=1,\ \nu=N-1, & \textit{if } N \textit{ is odd}, \\ \mu=-15,\ \tau=0,\ \eta=-4,\ \nu=N, & \textit{if } N \textit{ is even}, \end{array} \tag{8}$$

and the formula

$$d_i=\frac{3}{4h}(y_{i+1}-y_{i-1})-\frac{1}{4h}(d_{i+1}+d_{i-1}),\ i=1,3,\dots,\nu+\tau-1, \tag{9}$$

grant that the second derivatives of spline components at the inner grid points are equal.

Before the proof we make some remarks.

If in accordance with Theorem 1 the (approximately) first half of the unknown coefficients d_i is computed as a solution to the system (7) and the second half from the explicit formula (9), then the Hermite spline $\{H_1,\ H_2,\dots,H_{N+1}\}$ with the gained first derivatives $\{d_1,\ d_2,\dots,d_N\}$ will be a uniform cubic clamped spline of class C^2.

Notice that the last equation of (7) was constructed based on the model equation (5), the rest of its equations based on (6), and the formula (9) corresponds to (4). For even and odd N, the systems (7) differ only in the last row that takes the form

$$\begin{bmatrix} \dots & 0 & 1 & -15 \end{bmatrix}\begin{bmatrix} d_N \end{bmatrix}= \\ =\left[\frac{3}{h}(y_N-y_{N-2})-\frac{12}{h}(y_{N+1}-y_{N-1})+4d_{N+1}\right]$$

and

$$\begin{bmatrix} \dots & 0 & 1 & -14 \end{bmatrix}\begin{bmatrix} d_{N-1} \end{bmatrix}= \\ =\left[\frac{3}{h}(y_{N+1}-y_{N-3})-\frac{12}{h}(y_N-y_{N-2})\ -\ d_{N+1}\right],$$

respectively. For even N, equal number of unknowns are solved by (7) and (9), but for odd N, (7) solves by one unknown less than (9). This is why we speak shortly about the *approximately* first and second half of coefficients, and odd and even coefficients.

Proof. a) The equations of (7) were set up using the model equations (6) and (5), therefore the solution to (7) grants the equality of the second derivatives of spline components $H_i,\ H_{i+1},\ i=2,4,\dots$, at odd grid points $u_2,u_4,\dots$, and in the case of even N at the last but one (even) point u_N.
b) The use of formula (9) ensures the equality of the second derivatives of spline components $H_i,\ H_{i+1},\ i=1,3,\dots$, at even grid points $u_1,u_3,\dots$. □

The next section shows that Theorem 1 based the sequential reduced algorithm is computationaly more efficient than the standard one that uses the full system (3).

2. REDUCED ALGORITHM AND COMPUTATIONAL SPEEDUP

This section shows the efficiency of the reduced algorithm against the full one based on the assessed and measured speedup, but first of all we have a look at the LU decomposition that both algorithms use.

The standard way of solving tridiagonal linear systems

$$\underbrace{\begin{bmatrix} b & 1 & 0 & & \\ 1 & b & 1 & & \\ 0 & 1 & b & & \\ & \ddots & \ddots & \ddots & \ddots \\ & & & & b \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_K \end{bmatrix}}_{d} = \underbrace{\begin{bmatrix} r_1 - d_0 \\ r_2 \\ r_3 \\ \vdots \\ r_K - d_{K+1} \end{bmatrix}}_{r}, \tag{10}$$

uses the $L\,U$ factorization $A\,d = L\,\underbrace{U\,d}_{y} = r$, where

$$L = \begin{bmatrix} 1 & 0 & & & \\ l_2 & 1 & & & \\ 0 & l_3 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots \\ & & & l_K & 1 \end{bmatrix}, \quad U = \begin{bmatrix} u_1 & 1 & 0 & & \\ 0 & u_1 & 1 & & \\ & & u_2 & & \\ & & \ddots & \ddots & \ddots \\ & & & & u_K \end{bmatrix},$$

the u_i and l_i elements are computed as (see [3])

$$LU: \quad u_1 = b, \left\{ l_i = \frac{1}{u_{i-1}},\, u_i = b - l_i \right\}, i = 2, ..., K, \tag{11}$$

and the forward and backward steps (Fw and Bw) of the solution are

$$\text{Fw: } Ly = r \quad y_1 = r_1,\, \{y_i = r_i - l_i y_{i-1}\}, i = 2, \ldots, K, \tag{12}$$

$$\text{Bw: } Ud = y \quad d_K = \frac{y_K}{u_K},\, \left\{ d_i = \frac{1}{u_i}(y_i - d_{i+1}) \right\}, i = K-1, \ldots, 1. \tag{13}$$

Not only the standard full system (3) is solved by LU decomposition but also our reduced one (7).

The new algorithm for computing $\{d_1, \ldots, d_N\}$ in accordance with Theorem 1 is as follows.

Reduced algorithm

Step 1 Compute the right-hand side of (7) as an array $\boldsymbol{r}$ of length ν, see (8).

Step 2 Solve the reduced system (7) using (11), (12), (13) of the LU factorization. Based on this solution and the given derivatives d_0, d_{N+1} create an array $\boldsymbol{d}$ of length N.

Step 3 Compute the missing values of $\boldsymbol{d}$ using (9).

Table 1. **Count of operations in the algorithms, even** N

Step	Full algorithm	Reduced algorithm
RHS (3, 7)	$N(1^{\pm}+1^{\times})$	$\frac{N}{2}(3^{\pm}+2^{\times})$
LU (11)	$(N-1)(1^{\pm}+1^{\div})$	$(\frac{N}{2}-1)(1^{\pm}+1^{\div})$
Fw (12)	$(N-1)(1^{\pm}+1^{\times})$	$(\frac{N}{2}-1)(1^{\pm}+1^{\times})$
Bw (13)	$(N-1)(1^{\pm}+1^{\div})$	$(\frac{N}{2}-1)1^{\pm}+\frac{N}{2}1^{\div}$
Rest (9)		$(\frac{N}{2}-1)(3^{\pm}+2^{\times})$

The number of algebraic operations n°, $\circ\in\{\pm,\times,\div\}$ in the full and reduced algorithms are in Table 1. They were assessed using (11), (12), (13), the right-hand sides of (3) and (7), and in the case of the reduced system also (9). To assess the cost of operations three assumptions were used, that are in line with the properties of modern microprocessors:

- the operations $\{\pm,\times\}$ have nearly the same computation speed;
- the division is approximately *five* times slower than the other three operations;
- because of instruction level parallelism the expressions of the form $(a-b)c+(d-e)f$ are computed in *three* units of time instead of five.

The assessed speedup is the ratio of the costs of the full and reduced algorithms that are computed from the total number of operations based on Table 1

$$\frac{N(6+2\cdot 5)}{\frac{N}{2}\Big(4+2\cdot 3+2\cdot 5\Big)}=\frac{16}{10}. \tag{14}$$

Since the division is contained in the LU and Bw steps and it is five times slower than the addition, the division counts in (14) are multiplied by this number. The right-hand side of (7) and the rest formula (9) of the reduced algorithm utilize the instruction level parallelism, so it is reflected in the denominator of (14) as well.

One can see from (14) and Table 1 that in computation of the unknown derivatives the reduced system based algorithm should perform in sequential computation better than the traditional one. The expected speedup is approximately 1.6.

Measurements of speedup were conducted on computers with three different CPUs. The function values $y_i, i = 0, 1, \ldots, N$ were generated over the interval $[-1, 1]$ with equistep using $N = 10^5$, $N = 10^7$ and function $\sin(1 + x^2)$.

Table 2. **Computation times in ms and speedups,** $N = 10^5$

Processor	Full algorithm	Reduced algorithm	Speedup
A6 3650M	2.81	1.62	1.7
i3 2350M	1.93	1.13	1.7
i7 6700K	0.78	0.43	1.8

Table 3. **Computation times in ms and speedups,** $N = 10^7$

Processor	Full algorithm	Reduced algorithm	Speedup
A6 3650M	268.9	170.5	1.6
i3 2350M	198.7	121.6	1.6
i7 6700K	82.5	48.34	1.7

As we see from Tables 2 and 3, the speedup is increasing with decreasing the number of grid points N. It means naturally that the real speedup depends not only on the type of processors but on the available memory and caching as well.

3. DYADIC REDUCTION

Since the reduced system for the computation of uniform bicubic Hermite splines of class C^2 with parameters $1, -14, 1$ needs less divisions than the original one with parameters $1, 4, 1$, we were interested what can be said about tridiagonal systems with constants a, b, c, and whether the reduced system can be further reduced.

This section shows how to derive new model equations for equivalent reduced tridiagonal systems with less equations than the base tridiagonal one. The approach would be similar for general tridiagonal systems with a_i, b_i, c_i that correspond to nonuniform cubic splines.

First, we give a general definition of the model equation for the i-th order reduced system and then give examples. Meanwhile, we get the third derivation of (6) and the model equation of the second order reduced system, too.

Notation 1. *Let $D_j^{i,n}$ denote the j-th solution of the i-th order tridiagonal system with n equations.*

Definition 1. *The model equation M^i of the i-th order reduced tridiagonal system E^i, $0 < i$, is defined by the second solution $D_2^{i-1,3}$ of the $(i-1)$-th tridiagonal system $E^{i-1,3}$ with three equations, where E^0 represents the default tridiagonal system.*

Let us see some examples for the process of deriving models M^i for reduced systems E^i.

1) $\boldsymbol{M^1}$ for $\boldsymbol{E^1}$. From the base model equation

$$M^0: \; a\,x_1 + b\,x_2 + c\,x_3 = r_2 \tag{15}$$

we construct the tridiagonal system $E^{0,3}$ with three equations

$$\begin{bmatrix} b & c & 0 \\ a & b & c \\ 0 & a & b \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} r_1 - ax_0 \\ r_2 \\ r_3 - cx_4 \end{bmatrix}, \tag{16}$$

whence from its solution $x_2 \equiv D_2^{0,3}$ we get after algebraic rearrangement the model equation

$$M^1: \quad a^2x_0 + (2ac - b^2)x_2 + c^2x_4 = ar_1 - br_2 + cr_3. \tag{17}$$

If we substitute

$$1, 4, 1 \text{ for } a, b, c, \;\; d_i \text{ for } x_i \text{ and } \tfrac{3}{h}(y_{i+1} - y_{i-1}) \text{ for } r_i \tag{18}$$

in (17), then we get the first reduced model for the uniform clamped spline of class C^2

$$M^1: \quad d_0 - 14d_2 + d_4 = \frac{3}{h}(y_4 - y_0) - \frac{12}{h}(y_3 - y_1). \tag{19}$$

It was the third way of derivation (19) using tridiagonal systems.

Remark 1. Mention must be made that naturally, substituting x_0 and x_4 from the first and third equation of (16) into (17), we get back the second equation of (16).

2) $\boldsymbol{M^2}$ for $\boldsymbol{E^2}$. Based on the model equation M^1 (17) we set up the $E^{1,3}$ system

$$\begin{bmatrix} 2ac - b^2 & c^2 & 0 \\ a^2 & 2ac - b^2 & c^2 \\ 0 & a^2 & 2ac - b^2 \end{bmatrix} \begin{bmatrix} x_2 \\ x_4 \\ x_6 \end{bmatrix} = \begin{bmatrix} r_1a - r_2b + r_3c - a^2x_0 \\ r_3a - r_4b + r_5c \\ r_5a - r_6b + r_7c - c^2x_8 \end{bmatrix}. \tag{20}$$

From it's central solution $x_4 \equiv D_2^{1,3}$ we express the model equation

$$
\begin{aligned}
M^2 :\; & - a^4 x_0 + \left(\left(-b^2 + 2ac\right)^2 - 2c^2a^2 \right) x_4 - c^4 x_8 \\
& = \\
& - a^3 r_1 + a^2 b r_2 - \left(-a^2 c + ab^2\right) r_3 - \left(2cba - b^3\right) r_4 \\
& - \left(-ac^2 + cb^2\right) r_5 + bc^2 r_6 - c^3 r_7 .
\end{aligned}
\tag{21}
$$

If we exequte the substitution (18) in (21), then we get the second reduced model for the uniform clamped splines of class C^2

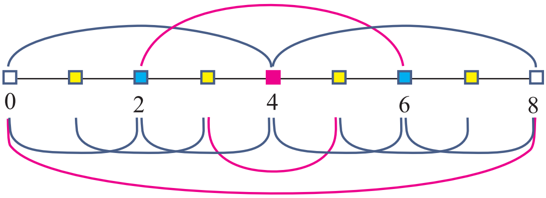
$$
\begin{aligned}
M^2 :\; +d_0 - 194\, d_4 + d_8 = 3\, \frac{y_8 - y_0}{h} - 12\, \frac{y_7 - y_1}{h} + 42\, \frac{y_6 - y_2}{h} - \\
- 156\, \frac{y_5 - y_3}{h} .
\end{aligned}
$$

Remark 2. For the grid points $x_0, \ldots, x_8$ expand the $E^{1,3}$ system (20) by four equations $ax_0 + bx_1 + cx_2 = r_1$, $ax_2 + bx_3 + cx_4 = r_3$, $ax_4 + bx_5 + cx_6 = r_5$ and $ax_6 + bx_7 + cx_8 = r_7$. Based on Remark 1 the system (20) is equivalent to $ax_1 + bx_2 + cx_3 = r_2$, $ax_3 + bx_4 + cx_5 = r_4$, $ax_5 + bx_6 + cx_7 = r_6$, and therefore the expanded $E^{1,3}$ is equivalent to $E^{0,7}$, thus $D_4^{0,7} = D_2^{1,3}$. Really, $D_4^{0,7}$ is the fourth solution of the default tri-diagonal system with seven equations. It is the central one, which in the first reduced system with three equations corresponds to the second solution $D_2^{1,3}$.

So we showed that the M^2 model (21) can be derived even from the M^0 model (15) by setting up the $E^{0,7}$ system with seven equations, because it's central solution $x_4 \equiv D_4^{0,7}$ equals the central solution $x_4 \equiv D_2^{1,3}$ of system $E^{1,3}$ (20).

We mention that in the context of Theorem 1 the system (20) corresponds to the reduced system (7) and the four equations to the rest formula (9).

To better understand the process of dyadic reduction we give a geometrical scheme of the process. The figure depicts the equality of solutions $D_4^{0,7}$ and $D_2^{1,3}$ for the second reduced system E^2 and indicates that the model equation M^2 can be expressed from any of these two equivalent solutions. In the picture the base (full) system's *unknowns* $x_1, \ldots, x_7$ correspond to the *inner* points $1, \ldots, 7$ denoted by squares (these squares together with the two terminal ones represent the nine grid points of a cubic spline with eight components and given *derivatives* $x_0 \equiv d_0$, $x_8 \equiv d_8$ at the terminal grid points). Every arc corresponds to an *equation* and is associated with three *principal* squares, the middle square and the two border ones under the arc, that correspond to the equation's three unknowns. The three upper arcs make up a $E^{1,3}$ system, the seven smallest lower ones a $E^{0,7}$ system (they represent the cubic's spline components). The *model* equations M^0, M^1 and M^2 are always the first, leftmost arcs.



Equivalence of solutions $D_4^{0,7}$ and $D_2^{1,3}$

Both solutions $D_4^{0,7}$ and $D_2^{1,3}$ refer to the middle point denoted by 4 in the picture, because this point corresponds to the solution x_4 of the default system with seven equations and also to the second solution x_2 of the first reduced system with three equations.

The second reduced system's model M^2 itself is denoted by the largest lower arc. This arc indicates that the default system with seven equations is represented in system E^2 by the model equation M^2. Under this largest arc there are three principal squares as well, two of which are the terminal ones that are associated with the given values x_0, x_8. The third one is associated with x_4 that equals either $D_4^{0,7}$ or $D_2^{1,3}$ and the model equation M^2 can be expressed from either of these equations.

3) $\boldsymbol{M^3}$ for $\boldsymbol{E^3}$. By generalization of the two examples and remarks the model M^3 can be equally derived from the central solution of systems $E^{2,3}$, $E^{1,7}$ and $E^{0,15}$.

Table 4. **Models and equivalent defining solutions for i-th reduced systems E^i**

Reduced systems	Models	Defining solutions				
E^1 :	M^1	$D_2^{0,3}$				
E^2 :	M^2	$D_4^{0,7}$	$\Leftrightarrow$	$D_2^{1,3}$		
E^3 :	M^3	$D_8^{0,15}$	$\Leftrightarrow$	$D_4^{1,7}$	$\Leftrightarrow$	$D_2^{2,3}$
. . .						

Table 4 summarizes the considered examples. Hence, noting that $D_{2^{i-j}}^{j,-1+2^{i-j+1}}$ is the central solution of the j-th order system $E^{j,-1+2^{i-j+1}}$, we get the following lemma that says how the i-th model equation as the central solution of the full tridiagonal system can be computed from different dyadic reduced systems.

Lemma 2. *For models M^i, $1 \leqslant i$, the model-defining solutions*

$$D_{2^i}^{0,-1+2^{i+1}} \quad \textit{and} \quad D_{2^{i-j}}^{j,-1+2^{i-j+1}}, \quad 0 \leqslant j < i, \tag{22}$$

are equivalent.

4. SUMMARY

We propose a general approach to dyadic reduction of the dimensionality of tridiagonal linear systems in consequence of which the number of divisions and the size of the systems gradually shrink to half, quarter etc. To preserve the information from the original full system the right-hand sides of the i-th order reduced systems get more complex. We show why modern processors can cope with this effect up to some degree. The considered new approach based on sequential algorithm yields more than 1.6x speedup.

Acknowledgements. The author is thankful to V. Kačala for conducting the measurements in C^{++} and pointing out the role of instruction level parallelism in computation. This work was partially supported by project APVV-15-0091 "Effective algorithms, automata and data structures".

APPENDIX

Below we provide the definition of a one-component Hermite spline on an interval and of a quartic polynomial on an equispaced grid determined by three values and two first derivatives.

Definition 2. *The one-component cubic Hermite spline on the interval* $[t_0,\, t_1]$ *with function values* f_0, f_1 *and first derivatives* d_0, d_1 *at* t_0, t_1 *is given by*

$$H(t) \equiv H(t;\, t_0, t_1, f_0, f_1, d_0, d_1) = \sum_{i=1}^{4} c_i \lambda_i,$$

where $c = [f_0, f_1, d_0, d_1]^T$ *and*

$$\lambda \equiv \lambda(t, t_0, t_1) = \begin{bmatrix} ((t-t_1)/(t_0-t_1))^2\,(1+2(t-t_0)/(t_1-t_0)) \\ ((t-t_0)/(t_1-t_0))^2\,(1+2(t-t_1)/(t_0-t_1)) \\ ((t-t_1)/(t_0-t_1))^2\,(t-t_0) \\ ((t-t_0)/(t_1-t_0))^2\,(t-t_1) \end{bmatrix}$$

Definition 3. *A quartic polynomial on an equispaced grid* $[t_0, t_1, t_2]$, $t_1 = t_0 + h$, $t_2 = t_0 + 2h$, *with function values* f_0, f_1, f_2 *and first derivatives* d_0, d_2 *at* t_0, t_1, t_2 *is given by*

$$f(t) \quad \equiv \quad f(t;\, t_0, t_1, t_2, f_0, f_1, f_2, d_0, d_2) = \sum_{i=1}^{4} \phi_i L_i, \tag{23}$$

where $\phi = [f_0, f_1, f_2, d_0, d_2]^T$ *and*

$$L(t, t_0, h) = \begin{bmatrix} -(1 + 2\dfrac{t - t_0}{h})(t - t_1)(t - t_2)^2/(4h^3) \\ (t - t_0)^2(t - t_2)^2/h^4 \\ (1 - 2\dfrac{t - t_2}{h})(t - t_0)^2(t - t_1)/(4h^3) \\ -(t - t_0)(t - t_1)(t - t_2)^2/(4h^3) \\ (t - t_0)^2(t - t_1)(t - t_2)/(4h^3) \end{bmatrix}.$$

REFERENCES

1. *Ahlberg J. H., Nilson E. N.* The Theory of Splines and Their Applications. Elsevier, 1967.
2. *Austin T. M., Berndt M., Moulton J. D.* A Memory Efficient Parallel Tridiagonal Solver. Preprint LA-VR-03-4149, 2004.
3. *Björck A.* Numerical Methods in Matrix Computations. Springer, 2015.
4. *de Boor C.* A Practical Guide to Splines. Springer, 2001.
5. *Powell M. J. D.* Approximation Theory and Methods. Cambridge: Cambridge Univ. Press, 1981.
6. *Salomon D.* Curves and Surfaces for Computer Graphics. Springer, 2006.
7. *Stoer J., Bulirsch R.* Introduction to Numerical Analysis. 3rd ed. New York: Springer-Verlag, 2002. ISBN-13 978-0387954523.
8. *Török Cs.* Reference Points Based Transformation and Approximation // Kybernetika. 2013. V. 49, No. 4. P. 644–662; http://www.kybernetika.cz/content/2013/4/644/paper.pdf
9. *Török Cs.* On Reduction of Equations' Number for Cubic Splines // Matematicheskoe modelirovanie. 2014. V. 26, No. 11. P.33–36.

Received on August 6, 2018.

Редактор *Е. И. Крупко*

Подписано в печать 13.08.2018.
Формат 60 × 90/16. Бумага офсетная. Печать офсетная.
Усл. печ. л. 1,12. Уч.-изд. л. 1,44. Тираж 215 экз. Заказ № 59461.

Издательский отдел Объединенного института ядерных исследований
141980, г. Дубна, Московская обл., ул. Жолио-Кюри, 6.
E-mail: publish@jinr.ru
www.jinr.ru/publish/