Rendiconti di Matematica, Serie VII
Volume 26, Roma (2006), 221-230

A spline optimization problem from robotics

Dedicated to Laura Gori on the occasion of her 70th birthday

WALTER HOFFMANN – TOMAS SAUER

Abstract: *We consider the problem of a robot manipulator moving along a prescribed trajectory in as short as possible time while maintaining certain physical constraints concerning acceleration and velocity. Modelling the motion by cubic spline curves, this corresponds to varying the knots of the spline curve for which we describe a simple but efficient algorithm.*

1 – Introduction

A classical problem in robot motion planning is to determine a *motion profile* that allows a robot to follow a given trajectory within a certain accuracy in shortest possible time. That the determination of the *time optimal* solution is a complicated process results from the presence of physical side conditions which limit, for example, the acceleration and the maximal speed of the system. In this paper, we consider a particular case of motion planning for an experimental hydraulic two–joint robot at the department of Control Engineering at the University of Erlangen.

A simplified geometry of the robot is shown in fig. 1 below. The two angles φ_1 and φ_2 are controlled by varying the hydraulic drives and by doing so the working point *WP* can reach each point in the so–called *working area*. This part of the plane is determined by the length and the movability of the robot arms.

Key Words and Phrases: *Spline curve – Robotics – Optimization*
A.M.S. Classification: 41A15 – 65D07

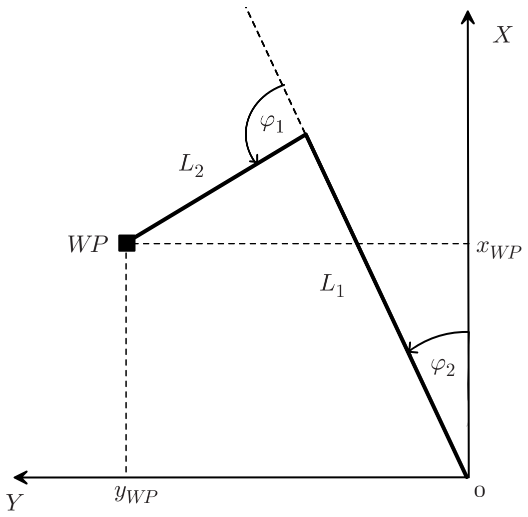

Fig. 1: Geometric configuration of the robot.

The *kinematic transformation* for a robot geometry relates the Cartesian coordinates of the working point to the *robot coordinates*, i.e., the controllable quantities of the robot, in our case the angles φ_1 and φ_2 at the joints. In the case of fig. 1, those equations simply look as follows:

$$\begin{aligned} x_{WP} &= L_1 \cos(\varphi_1) + L_2 \cos(\varphi_1 + \varphi_2) \\ y_{WP} &= L_1 \sin(\varphi_1) + L_2 \sin(\varphi_1 + \varphi_2). \end{aligned}$$

It is easy to see that in general a given position for the work point in the plane can be reached using two different robot coordinates, just imagine an "elbow up" and an "elbow down" position. Consequently, without further restrictions the kinematic transformation cannot be bijective. However, the technical constraints on the hydraulic drives restrict the angles to be chosen within the bounds

$$\begin{aligned} 25^\circ &\le \varphi_1 \le 106^\circ, \\ 48^\circ &\le \varphi_2 \le 150^\circ, \end{aligned}$$

for which the kinematic transform is one–to–one. In addition, this restriction also guarantees that the velocity bound for the maximal feed rate of the robot can never be reached, not even by full acceleration from one end of the working area to the other. For a detailed description of the robot and its control techniques see [4].

2 – Splines in motion

Due to the two–dimensional nature of the robot, a trajectory is now a curve $\boldsymbol{x} : \mathbb{R} \to \mathbb{R}^2$, where $\boldsymbol{x}(t)$ denotes the position of the hand point at the time t. Assuming, for convenience, that the robot starts at time $t = 0$, we can consider the trajectory on the interval $I = [0, t^*]$ where it is assumed that at the time t^* the robot has reached its end position. Since, in addition, we want to start from and end in a resting position, we get the side conditions

$$\dot{\boldsymbol{x}}(0) = \dot{\boldsymbol{x}}(t^*) = \ddot{\boldsymbol{x}}(0) = \ddot{\boldsymbol{x}}(t^*) = 0 \tag{1}$$

requiring zero velocity and acceleration at the beginning and end point. Due to the nature of the hydraulic valves, any curve that can be realized by the robot must be at least twice continuously differentiable. On the other hand, the third derivative, usually called the *jerk* in Control Engineering, may very well have discontinuities. This makes it natural to model the curve $\boldsymbol{x}$ by means of *cubic B–splines*, cf. [1], [2], [3], writing it as

$$\boldsymbol{x}(t) = \sum_{j=0}^{N} \boldsymbol{x}_j \, N_j^3(\cdot \,|\, T) \tag{2}$$

with respect to the *knot sequence*

$$T = \{t_0, \dots, t_{N+4}\}.$$

Here we employ the notation from [7], where the knots have to satisfy

$$t_j \le t_{j+1}, \quad j = 0, \dots, N+3, \qquad t_j < t_{j+1}, \quad j = 3, \dots, N,$$

which permits *multiple* knots, but limits the multiplicity to the local polynomial degree of the spline. Moreover, recall that the B–splines are normalized such that they form a partition of unity, that is

$$\sum_{j=0}^{N} N_j^3(x \,|\, T) = 1, \qquad x \in [t_3, t_{N+1}].$$

Like in [7] we also make the assumptions that

$$t_0 = t_1 = t_2 = t_3 = 0, \qquad t_{N+1} = t_{N+2} = t_{N+3} = t_{N+4} = t^*, \tag{3}$$

i.e., that at both endpoints we have quadruple knots, as this automatically guarantees that (1) is satisfied.

Given a prescribed contour $\boldsymbol{y} : \mathbb{R} \to \mathbb{R}^2$, usually a C^2 curve, the approximating spline curve is generated by sampling $\boldsymbol{y}$ at points s_j, $j = 0, \dots, N$, giving

$$\boldsymbol{x}_j = \boldsymbol{y}(s_j), \qquad j = 0, \dots, N.$$

These points can be simply equidistributed or be chosen in an adaptive way, for example according to curvature information on the curve $\boldsymbol{y}$. The spline curve

$$\boldsymbol{x}(t) = \sum_{j=0}^{N} \boldsymbol{x}_j \, N_j^3(t \,|\, T) = \sum_{j=0}^{N} \boldsymbol{y}(s_j) \, N_j^3(t \,|\, T)$$

can then be expected to be a good approximation to $\boldsymbol{y}$. Indeed, when the s_j are chosen as the so–called *Greville abscissae*

$$s_j = \frac{1}{3}(t_j + t_{j+1} + t_{j+2}), \qquad j = 0, \dots, N,$$

then a classical result for spline “quasi interpolants”, cf. [3], [6], says that

$$\|\boldsymbol{x} - \boldsymbol{y}\| \le C \, \|\ddot{\boldsymbol{y}}\| \, h^2, \qquad h = \max_{j=0,\dots,N} t_{j+1} - t_j, \tag{4}$$

where

$$\|\boldsymbol{x} - \boldsymbol{y}\| := \max_{x \in [t_3, t_{N+1}]} |\boldsymbol{x}(t) - \boldsymbol{y}(t)|,$$

$|\cdot|$ being any norm on $\mathbb{R}^2$, denotes the *parametric* distance between the curves, i.e., the distance based on parameterizing the two curves identically. Consequently, the “true” distance between these curves, usually measured in a Hausdorff norm, will be even smaller.

The quality of approximation of the initial approximating curve $\boldsymbol{x}$ depends only of how fine we sample $\boldsymbol{y}$: suppose that $\boldsymbol{y}$ is defined on $[0, 1]$, then we choose the knots as

$$t_0 = \cdots = t_3 = 0, \quad t_{N-1} = \cdots = t_{N+2} = 1, \qquad t_j = \frac{j-3}{N-4}, \quad j = 4, \dots, N-2,$$

yielding an accuracy of $O(N^{-2})$ by (4), while the multiplicity of the knots guarantees that the initial and terminal conditions (1) are satisfied at $t = 0$ and $t = t^* = 1$, respectively. This initial approximant $\boldsymbol{x}$, however, may fail to satisfy the velocity and acceleration constraints $\|\dot{\boldsymbol{x}}\| \le V$ and $\|\ddot{\boldsymbol{x}}\| \le A$, in which case we enlarge the distance between the knots by a simple scaling $t_j = t^* \, t_j$, $j = 0, \dots, N$, enlarging $t^* > 1$ until the constraints are satisfied eventually. The resulting curve $\boldsymbol{x}^* = \boldsymbol{x}(\cdot / t^*)$, now mapping $[0, t^*] \to \mathbb{R}^d$, is then a good approximant to $\boldsymbol{y}^* = \boldsymbol{y}(\cdot / t^*) \ : \ [0, t^*] \to \mathbb{R}^d$ since

$$\max_{0 \le t \le t^*} |\boldsymbol{x}^*(t) - \boldsymbol{y}^*(t)| = \max_{0 \le t \le 1} |\boldsymbol{x}^*(t^* \, t) - \boldsymbol{y}^*(t^* t)| = \|\boldsymbol{x} - \boldsymbol{y}\|.$$

The determination of the "sampling rate" N, on the other hand, depends essentially on the curvature of $\boldsymbol{y}$ the precise representation of which depends on the application context. Typically, it can be a spline curve itself, though possibly of much higher order, or it could be composed of lines and arc segments as for example in [4].

Writing N_j^k for the B–spline of order k, the well–known derivative formulae for spline curves, see again [3], can be applied to our cubic spline curve

$$\boldsymbol{x} = \sum_{j=0}^{N} \boldsymbol{x}_j \, N_j^3(\cdot \,|\, T).$$

Taking into account the multiplicity of the end knots, we then get

$$\dot{\boldsymbol{x}} = 3 \sum_{j=1}^{N} \boldsymbol{x}_j^{(1)} \, N_j^2(\cdot \,|\, T), \qquad \boldsymbol{x}_j^{(1)} = \frac{\boldsymbol{x}_j - \boldsymbol{x}_{j-1}}{t_{j+3} - t_j}, \quad j = 1, \dots, N, \tag{5}$$

as well as

$$\ddot{\boldsymbol{x}} = 6 \sum_{j=2}^{N} \boldsymbol{x}_j^{(2)} \, N_j^1(\cdot \,|\, T), \qquad \boldsymbol{x}_j^{(2)} = \frac{\boldsymbol{x}_j^{(1)} - \boldsymbol{x}_{j-1}^{(1)}}{t_{j+2} - t_j}, \quad j = 2, \dots, N. \tag{6}$$

Note that since the spline curve $\ddot{\boldsymbol{x}}$ is a linear one, we immediately have that

$$\|\ddot{\boldsymbol{x}}\| := \max_{t \in [t_3, t_{N+1}]} |\ddot{\boldsymbol{x}}(t)| = \max_{j=2,\dots,N} |\boldsymbol{x}_j^{(2)}|. \tag{7}$$

Thus, the time optimal motion planning problem can be formulated as follows:

> Given a *velocity bound* V and an *acceleration bound* A, determine knots $t_0 = \cdots = t_3 < \cdots < t_{N+1} = \cdots = t_{N+4}$ such that $t_{N+1} - t_3$ is minimized under the side conditions
>
> $$\|\dot{\boldsymbol{x}}\| \le V, \qquad \|\ddot{\boldsymbol{x}}\| \le A.$$

Equations (5) and (6) make it clear that this is a highly nonlinear problem in the free parameters which are the knots $t_4 < \cdots < t_{N+1}$. Taking into account that also $\|\dot{\boldsymbol{x}}\| \le 3 \cdot \max_j |\boldsymbol{x}_j^{(1)}|$, we can focus on the following, slightly simpler problem:

$$\min t_N - t_3, \qquad \max_{j=1,\dots,N} |\boldsymbol{x}_j^{(1)}| \le \frac{1}{3} V, \quad \max_{j=2,\dots,N} |\boldsymbol{x}^{(2)}| \le \frac{1}{6} A, \quad t_j < t_{j+1}. \tag{8}$$

As simple consequence of the above observations is then as follows.

THEOREM 1. *A solution of* (8) *is a time optimal spline curve if*

$$|\boldsymbol{x}_j^{(1)}| < \frac{1}{3}V, \qquad j = 1, \dots, N. \tag{9}$$

In this case, we have that

$$\prod_{k=0}^{3} \left(\frac{1}{6}A - |\boldsymbol{x}_{j+k}^{(2)}| \right) = 0, \qquad j = 2, \dots, N, \tag{10}$$

that is, equality must be assumed at least one of any four consecutive acceleration constraints.

PROOF. Note that $\boldsymbol{x}_j^{(1)}$ depends on the knots t_j and t_{j+3} while $\boldsymbol{x}_j^{(2)}$ depends on $t_{j-1}, t_j, t_{j+2}, t_{j+3}$, more precisely, on the differences $t_{j+2} - t_j$, $t_{j+3} - t_j$ and $t_{j+2} - t_{j-1}$. Suppose that (9) holds true. By continuity of (5) with respect to the knots, (9) remains valid for a sufficiently small perturbation of the knots. Now assume in addition that (10) were not true and let j be the smallest index such that $|\boldsymbol{x}_{j+k}^{(2)}| < \frac{1}{6}A$, $k = 0, \dots, 3$. Then there exists some $\varepsilon > 0$ such that after replacement of T by T^*, defined as

$$t_k^* = \begin{cases} t_k, & k = 0, \dots, j+2, \\ t_k - \varepsilon, & k = j+3, \dots, N+2, \end{cases}$$

we still have that

$$|\boldsymbol{x}_{j+k}^{(2)}| \leq \frac{1}{6}A, \qquad k = 0, \dots, 3. \tag{11}$$

In fact, ε can be chosen optimally in such a way that equality is assumed in at least one of the inequalities in (11). Since the constraints $|\boldsymbol{x}_k^{(2)}| \leq \frac{1}{6}A$, $k < j$, depend only on the knots $\{t_0^*, \dots, t_{j+2}^*\} = \{t_0, \dots, t_{j+2}\}$, they still remain satisfied with T being replaced by T^*. On the other hand, the constraints $|\boldsymbol{x}_k^{(2)}| \leq \frac{1}{6}A$, $k > j+3$, involve differences between knots of the form

$$t_{k+3}^* - t_k^* = t_{k+3} - \varepsilon - t_k + \varepsilon = t_{k+3} - t_k, \qquad k \geq j+3,$$

and

$$t_{k+2}^* - t_k^* = t_{k+2} - t_k, \qquad k \geq j+4,$$

and thus also remain valid. Thus, passing from T to T^* we obtain an improved solution of the optimization problem (8) and thus T was no optimal solution. ∎

The proof of Theorem 1 already indicates the main idea for our algorithm to obtain a time–optimal profile: we begin with an arbitrary distribution of knots and then try to get any two successive knots as close as possible while still maintaining the velocity and acceleration constraints.

To illustrate this idea, we briefly look at the simple case of trajectory planning by means of quadratic splines, bounded only by the first order velocity constraints

$$\max_{t\in[t_2,t_N]} |\dot{\boldsymbol{x}}(t)| = 3 \max_{j=1,\dots,N} |\boldsymbol{x}_j^{(1)}| \le V.$$

By means of the respective variant of (5) for quadratic splines, the side conditions can even be met at any of these equations if we choose

$$t_{j+2} = \max\Big\{t_{j+1}, t_j + \frac{|\boldsymbol{x}_j - \boldsymbol{x}_{j-1}|}{V}\Big\}, \qquad j = 1,\dots,N, \tag{12}$$

which leads to the time optimal solution and results from directly solving $|\boldsymbol{x}_j^{(1)}| = V$. Note that this does not exclude the generation of multiple knots; we will return to this issue later.

3 – A knot-shifting algorithm

Unfortunately, the simple approach for velocity–constrained quadratic spline trajectories does no more apply in the case of cubic splines with acceleration constraints. The reason is that the equations $|\boldsymbol{x}_j^{(2)}| = V$ are now highly nonlinear with respect to the knots and that, as seen in the proof of Theorem 1, there are always four knots involved in such an identity. Due to these difficulties, the motion profile will be determined by an iterative algorithm that we describe next.

One step of the algorithm will consist of modifying the knot vector $T^{(k)}$ in such a way that the constraints are still satisfied, but that the new knot vector $T^{(k+1)}$ improves upon $T^{(k)}$ in the sense that $t_{N+1}^{(k+1)} < t_{N+1}^{(k)}$. As an initial knot vector $T^{(0)}$ one can choose, for example, an equidistant distribution that is uniformly stretched such that $\|\ddot{\boldsymbol{x}}\| \le A$. It is easy to see that replacing T by λT, $\lambda > 0$, yields that $\|\ddot{\boldsymbol{x}}\|$ is scaled by a factor of λ^{-2}.

The core part of such an iteration step is to modify a knot, say t_j, $j = 4,\dots,N$, such that $6|\boldsymbol{x}_{j-2}^{(2)}| \in [A-\varepsilon, A]$ for some prescribed tolerance $\varepsilon > 0$. This is done by considering $|\boldsymbol{x}_{j-2}^{(2)}|$ as a function in t_j and numerically finding an approximate solution to the nonlinear equation

$$|\boldsymbol{x}_{j-2}^{(2)}(\cdot)| - \frac{A}{6} = 0, \tag{13}$$

to determine the value of t_j. Though the Newton method would provide faster convergence, we have to rely on bisection here since we need a *one sided* approximation to the solution in order not to exceed the constraints, and this feature is automatically offered by bisection. Since the B–splines which do not vanish at t_j are precisely $N_{j-3}, \dots, N_{j-1}$, the variation of t_j does not affect the curve of the left of t_j – this is the well–known *local control* property of B–splines. Therefore, we shift t_j *to the left* until either $6|\boldsymbol{x}_{j-2}^{(2)}| \in [A-\epsilon, A]$ or $t_j - t_{j-1} \le \eta$ for some small constant $\eta > 0$ that describes how fast the machine control can react; usually the order of this quantity is about 5ms. However, the fact that the knot t_j can be shifted arbitrarily close to t_{j-1} means that the control point $\boldsymbol{x}_j$ lies on the line connecting $\boldsymbol{x}_{j-1}$ and $\boldsymbol{x}_{j+1}$. This, on the other hand, allows us to *remove* the knot t_j together with the control point $\boldsymbol{x}_j$ without significantly changing the trajectory since in this region the curve is a straight line.

Once the knot t_j is either moved such that the constraint is assumed with equality or has been removed, we increase j by one and thus pass to the next knot. Since the process only moves knots to the left, it is clear that it only improves the knot sequence. However, due to the overlap of knots in the eq. (13), a single adaption step of this sort cannot give an optimal solution, so that we have to iterate the knot adaption process. Since every such iteration step reduces the last knot t_{N+1}, this iteration converges and thus leads to a limit knot profile.

Since this limit profile is obtained by a local modification process, we cannot expect it to be a global solution of the optimization problem (8), and indeed this can be verified with the examples provided in [5]. But, as paradox as it may seem, the above algorithm often leads to solutions that are significantly better that those that are obtained by the standard nonlinear optimization routines in `Matlab`. The reason is simple: many prescribed trajectories contain segments that are approximately linear and thus offer a potential for *knot removement*. It even turns out that knot removal smoothes the trajectory in the sense that it keeps the oscillations of the second derivatives (which usually cause mechanical problems) lower than what is obtained by standard optimization methods.

A simple example and its solution is shown in fig. 2. A reference path in shape of a triangle was programmed with 30 control points in Cartesian coordinates. After the kinematic transformation the reference positions of the path in machine coordinates is shown in the upper right figure. The second derivative of the angle is a measure for the forces that affect to the axe of the robot. The computed solution shows the adapted velocity along the path of the programmed contour. At about 0.3 and 0.8 seconds the edges of the triangle can be identified. This Algorithm was designed for an offline–look–ahead functionality. But an online–solution is feasible due to the fact of the discrete computations.

Further details of this comparison and plots of the velocity and acceleration profiles for standard example curves can be found in [5].

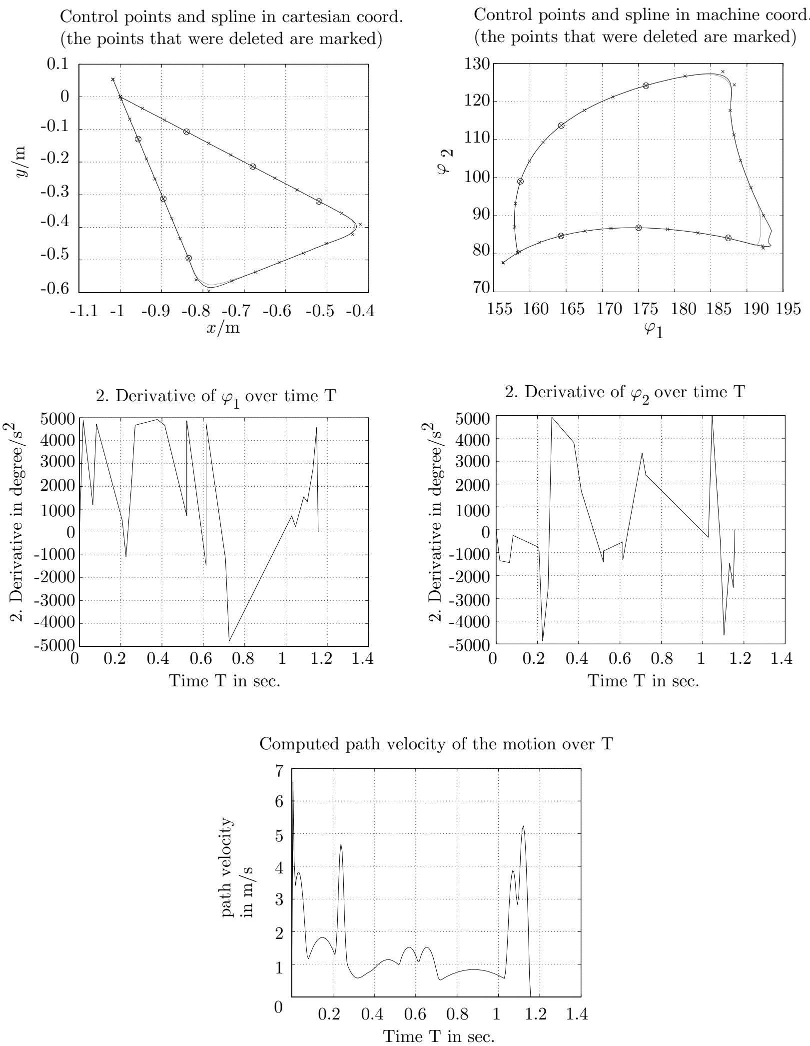

Fig. 2: Solution for a simple example. Upper left figure shows the programmed positions in Cartesian coordinates; upper right figure shows them in machine coordinates; the last picture shows the adapted velocity along the path.

4 – Conclusions

Motion profiles using variable speed provide a productivity superior to those with constant speed by adapting the velocity of the work point to the geometry of the curve. We have presented a method to compute such a profile under the assumption that only the acceleration constraints matter for the profile planning. The method is based on adapting the knot distribution of a cubic spline to the curve and to remove knots where the trajectory is practically linear and often leads to results that are better than what can be obtained by standard optimization methods.

REFERENCES

[1] C. de Boor: *On calculating with B–splines*, J. Approx. Theory, **6** (1972), 50-62.

[2] C. de Boor: *A practical guide to splines*, Springer–Verlag, New York, 1978.

[3] C. de Boor: *Splinefunktionen*, Lectures in Mathematics, ETH Zürich, Birkhäuser, 1990.

[4] K. Graf: *Bahnführung von Handhabungsgeräten mit unterlagerter Beschleunigungsregelung am Beispiel hydraulisch angetriebener Geräte*, Shaker Verlag, Aachen, 2002.

[5] W. Hoffmann: *B–Splinekurven zur zeitoptimalen Robotersteuerung*, Diploma thesis, Erlangen, 2001.

[6] G. Nürnberger: *Approximation by spline functions*, Springer–Verlag, 1989.

[7] H. P. Seidel: *A new multiaffine approach to B–splines*, Comp. Aided Geom. Design, **6** (1989), 23-32.

[8] M. W. Spong: *Motion Control of Robot Manipulators*, in “Handbook of Control”, CRC Press, 1996.

[9] K. G. Shin – N. D. McKay: *Minimum time control of robot manipulators with geometric path constraints*, in “IEEE Transactions on Automatic Control”, Vol. AC-30, No. 6, June 1985.

[10] R. Johanni: *Optimale Bahnplanung bei Industrierobotern*, in “Fortschrittberichte VDI”, Reihe 18 Nr. 51, VDI-Verlag, 1988.

[11] R. P. Paul: *Robot manipulators: mathematics, programming, and control*, MIT Press, 1981.

Lavoro pervenuto alla redazione l' 11 ottobre 2004
ed accettato per la pubblicazione il 4 maggio 2005.
Bozze licenziate il 10 maggio 2006

INDIRIZZO DEGLI AUTORI:

Walter Hoffmann – Siemens AG, A&D MC RD 39 – Frauenauracherstr. 80, D-91056 Erlangen

Tomas Sauer – Lehrstuhl für Numerische Mathematik – Justus–Liebig–Universität Giessen – Heinrich–Buff–Ring 44, D–35392 Giessen– E-mail: Tomas.Sauer@math.uni-giessen.de