In: IEEE Int. Conf. on Methods and Models in Automation and Robotics (MMAR-2005), 29. Aug. 1. Sep., Miedzyzdroje, Poland

MANIPULATION OF DEFORMABLE LINEAR OBJECTS: ANALYSIS OF TWO-DIMENSIONAL STATIC APPROXIMATION FUNCTIONS

SERGEJ TIMM, BJÖRN KAHL, DOMINIK HENRICH

Lehrstuhl für Angewandte Informatik III, Universität Bayreuth, D-95445 Bayreuth, [Bjoern.Kahl | Dominik.Henrich] @ uni-bayreuth.de

Abstract. In this paper we analyze different models of deformable linear objects (DLOs) in terms of efficient computation and preciseness. Our approach is to minimize the potential energy (computed as the sum of gravitational and bending energy) of the DLO. To model our DLOs, we approximate the bending of the DLO using a weighted sum of approximation functions. Our objective is to identify the optimal type of approximation functions out of a given set of different types of functions like Fourier, Chebyshev, Haar and other function types.

Key Words. Modelling, deformable linear objects (DLOs), minimizing approximation functions

1. INTRODUCTION

DLOs such as ropes, wires or steel springs are used in a wide range of products. Unfortunately, inherent uncertainties such as bending and compliance make it difficult to handle these objects in a generalized and automated manner. With a fast and reliable simulation of the static (and perhaps dynamic) behavior of a DLO in place, these difficulties can be overcome by using an abstract task model based on contact states [1] to describe a given assembly task. This allows one to avoid the use of exact coordinates and movement instructions in the assembly process. Our research group has developed a library of such sensor-based manipulation skills [7]. The next step is to automatically generate a sequence of contact states (needed for our library) by demonstrating the desired assembly task in virtual reality [3]. Thus, we need a DLO simulation which is real-time capable. Fortunately, the simulation only requires a loose approximation of the DLOs geometry and not the exact shape.

Several researchers have published on the topic of DLO simulation. In the following overview, we give a short summary of prior work: A two-dimensional energy-based model is presented in [2], and an enhancement for a three-dimensional model can be found in [8]. Both approaches utilize the fact that the potential energy of the DLO is minimized in its stable configuration and [9] includes consideration of dynamic energy. While the above mentioned approaches are not real-time capable, [6] presents more efficient approximation methods. All of the above researchers use a linear combination of basic functions to approximate bending or the tangent of the DLO. Fourier functions are used in [2], the so-called "enhanced" Fourier functions (discussed below) are implemented in [8] and [9]. In [6] we find a comparison between Fourier functions and type-1 Chebyshev polynomials. In this paper we re-examine the above-mentioned functions and compare them with additional functions.

All calculations are performed in two-dimensional space. We have only considered the static, unconstrained case, e.g. dynamics and obstacles are neglected. Our model is based on the minimization of the potential energy. The potential energy used is the sum of gravitation and bending. To minimize the energy, we use the established Downhill Simplex Method (DSM). The bending is described by linear superposition of well-known approximation functions. Another approach would be to calculate the tangent using these functions [2].

Our objective is to determine which type of approximation functions, best simulates the bending of a DLO in the static case. In the following, the term "*set of approximation functions*" names all functions of a fixed type, e.g. all Fourier functions of order 1 to N. An "*approximation function*" is one member of this set e.g. the "Fourier function of 2nd order", while the term "*set of approximation function type*s" names all the different types of functions like Fourier, Chebyshev, Legendre etc. An "*approximation function type*" is one such function family.

Our benchmarks are the average time needed to compute a stable shape for our DLO and the geometrical error of the calculated shape compared to the real DLO. We call a shape "stable" if it does not change to much between two approximation steps. Furthermore, we

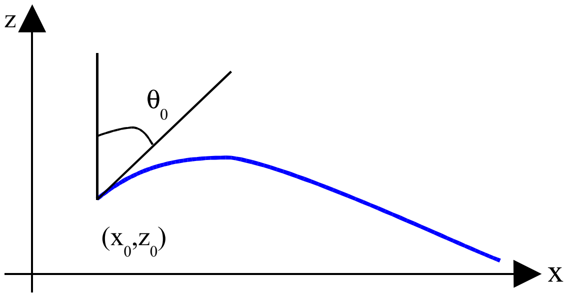

Fig. 1: Model of a DLO

want to determine whether the DSM can be used with a specific set of approximation functions in real-time. The DSM is classified as real-time enabled (with respect to a given function-set) if the calculation could be interrupted at any point in time and the approximation computed up until that point is a realistic representation of a DLO. A methodological problem here is the exact definition of what constitutes a “realistic” representation of a DLO. Note that we test each set of approximation functions against various DLO parameters and initial conditions.

Section 2 describes our approach in detail and introduces the set of approximation functions. Section 3 describes our experimental setup and in Section 4 we present a discussion of our results.

2. MODEL DESCRIPTION

To calculate the shape of our DLO, we use the well-known approach of minimized potential energy. We calculate the internal energy of our DLO as a function of its bending.

2.1.Model

Assuming that one of the DLO's endpoints is fixed, the shape can be computed in parametric representation by

$$x(s)=\int_0^s \sin(\theta_0+\theta(t))\,dt+x_0 \tag{1}$$

$$z(s)=\int_0^s \cos(\theta_0+\theta(t))\,dt+z_0 \tag{2}$$

with (*x(s)*, *z(s)*) Cartesian coordinates of a point with distance *s* (along the DLOs curve), *(x₀, z₀)* the fixed endpoint and θ_0 the two-dimensional orientation of the fixed endpoint. The angle *θ(t)* is measured between the DLO tangent at point (*x(s)* , *z(s)*) and the z-axis of the coordinate system (see Fig 1).

By definition, the bending *k* is equal to the first derivative of the angle between the DLOs tangent and the z-axis of the coordinate system: $k = \theta'$. Thus it is possible to use the stem function *k* instead of *θ* to model the DLO. We model our DLO here using *k:*

$$\theta(t)=\int_0^t k(u)\,du \tag{3}$$

According to [4] the energy of a deformed DLO can be calculated as

$$W_G=\rho A g\int_0^L z(s)\,ds \tag{4}$$

$$W_B=\frac{1}{2}AE\int_0^L \begin{cases} R^2k(s)^2/4 & k(s)\le 2\epsilon_F/R \\ (Rk(s)-\epsilon_F)\epsilon_F & \text{otherwise}\end{cases} ds \tag{5}$$

with the gravitational energy W_G, the bending energy W_B, cross-sectional area *A*, diameter *R*, specific mass *ρ* and bending coefficient *E* of the DLO. We assume an idealized DLO with ε_F equivalent to σ_F/E, whereby σ_F/E is the edge in the force-bending diagram. The sum of W_B and W_G is the total energy *W* to be minimized in respect to *k*.

Successful minimization results in the function *k(s)*, which produces the shape (expressed in the coordinates of points) of the DLO. Unfortunately, it appears impossible to solve the problem analytically. To overcome this drawback, we approximate *k* as a linear superposition of well-known functions (this is also known as “Ritz's method”):

$$k(u)=\sum_{i=1}^{n_c} c_i q_i(u)=\boldsymbol{c}\boldsymbol{q} \tag{6}$$

Thus, we call q_i an *approximation function* and define ***q*** as a given set of such approximation functions and ***c*** the corresponding set of coefficients. Based on these prerequisites, the total energy is a function of n_c unknowns: (most constants ignored):

$$\begin{aligned} W(c_1,\cdots,c_{n_c}) &= \int_0^L\left(\int_0^s \cos\left(\theta_0+\int_0^t\sum_{i=1}^{n_c} c_i q_i(u)\,du\right)dt\right)ds \\ &+\int_0^L\sum_{i=1}^{n_c} c_i q_i(s)^2\,ds \end{aligned} \tag{7}$$

This energy function as shown in eqn. (7) consists of integrals that are difficult or even impossible to solve analytically. To use the Trapezoidal Rule for the required numerical integration, we divide the DLO into n_s segments of equal length. Thus, $O(n_s\, n_c)$ operations are needed to calculate the energy function.

2.2.Minimization

The Downhill Simplex Method (DSM) is an established method to minimize an “objective function” of more than one argument under a set of (linear) constraints; see e.g. [5]. The method requires $n_c + 1$ sets of unknown coefficients that constitutes the $n_c + 1$ vertices of the simplex. We use the following initialization:

$$\boldsymbol{c_0}=\boldsymbol{0},\quad \boldsymbol{c_j}=\lambda\boldsymbol{e_j}\quad j=1,\cdots,n_c,\quad \lambda\in\mathbb{R} \tag{8}$$

with $\boldsymbol{e_j}$ the *j*-th base vector. (I.e. the first set has all c_i initialized to 0 and all following sets have all but c_i set to 0 and one set to *λ*.)

The total computation time depends on *λ* (as *λ* defines the initial size of the simplex) and on the set of approximation functions used (because different types of approximation functions have different levels of complexity and different optimal values for *λ*). To compare the average computation time for different types of approximation functions, we must first find the best *λ* for each type of approximation function, and then perform benchmark tests with each such function type using the optimal *λ* value previously determined. To determine the best *λ* for each type of approximation function, we perform several experiments under varying initial conditions (i.e. R, n_c, n_s, θ_0) for each type of functions. We determine the best λ for each set of initial conditions for a given type of approximation functions. Then we use the arithmetic average of the resulting λ values (rounded to the nearest integer value) as the optimal λ for the given type of approximation functions. It turned out that the optimal λ is 1 in most cases, except for the Chebyshev-1 and Legendre functions.

2.3. Approximation function types

The approximation function types are summarized in Table 1: The first column shows the function's name and the second column specifies the function's domain. The DLO is scaled to fit into this domain. The third column shows whether the function describes an orthogonal system; if it does, the column contains the appropriate normalization function. The fourth column contains the best λ for that function, as discussed above. Finally, the last column summarizes special properties of the function that could be of interest in our study. The set of "Fourier2" functions is the set of "Fourier" functions with an additional function of $y = x$. The additional function covers the linear component of the to-be-approximated function.

Table 1 Table of approximation functions analyzed in this paper.

functions	[a,b]	w(t)	λ	Special properties
Chebyshev 1.	[-1,1]	$1/\sqrt{1-t^2}$	10	
Chebyshev 2.	[-1,1]	$\sqrt{1-t^2}$	1	
Fourier	[0,2π]	1	1	
Fourier2	[0,2π]	-	1	Fourier-Functions + Function *x*
Legendre	[-1,1]	1	10	
Haar	[0,1]	1	1	Step function
Standard	[-1,1]	-	1	Simple polynomials: t^i $i=0,1,\ldots$
Bernstein	[0,1]	-	1	Linear dependent functions ("Partition of One"), symmetrical
Hermite	[-1,1]	-	1	Orthogonal at $(-\infty,\infty)$ wrt. $w(t)=e^{-t^2}$

3. EXPERIMENTS

Our aim is to find out which approximation function type is best suited to approximate the bending of our DLO. Our evaluation criteria are: 1) the time needed to compute the minimized energy; 2) the accuracy of the computed shape. In general, as the computed minimal energy decreases, the bending approximation become more accurate. However, a poor approximation may lead to an overly low calculated energy. We found that the experiment can result in poor approximations, if the following relation is lower than two:

$$r_{cs} = \frac{n_s}{n_c} \tag{9}$$

with n_c the number of coefficients in eqn. 6 and n_s the number of supporting points in the numerical integration. Relation (9) is an experimental finding, we do not know the reason behind this observation, but see discussion below for a possible reason. Unfortunately, the actual value of the energy is unknown and there is no clear distinction between energy values that are reasonable and those that appear to be invalid. Therefore, the entire distribution of energy across different values of n_c is considered when evaluating a given approximation function type.

3.1. Fixed Model Parameters

Our model contains several parameters that can be changed as needed. Some of these are modified during our tests, while others are held constant.

The energy function (eqn. 7) is modelled and minimized in terms of the bending of the DLO. The minimization is performed with the Downhill Simplex Method using $1 \cdot e^{-10}$ as the termination criteria for the difference between minimum and maximum values of the function to be minimized. The integrals involved are calculated using the trapeze method. The DLO is a 1m long copper fiber (specific mass 8960 kg/m^3) with an elastic modulus of $126 \cdot 10^9$ N/m^2. The origin of the DLO (i.e. the fixed end point) is at position (0,1) in the Cartesian coordinate system.

3.2. Variable parameters

The following parameters are modified in our experiments:

- Number of approximation functions n_c: As the number of functions increase, both the accuracy of the approximation and the computation time increase.
- Number of segments of the DLO n_s: Increasing the number of segments improves the accuracy of the calculated numerical integrations, providing more realistic energy values.
- Orientation of the DLO's fixed endpoint θ_0: The shape of the DLO depends on the endpoint orientation (as measured by the angle between the tangent of the DLO at this point and the vertical coordinate axis).
- The DLO radius R: The radius influences the stiffness of the DLO.

3.3. Description of Experiments

Two different types of experiments are performed. First, the energy minimization is examined with respect to different sets of variable parameters as defined above. We record the calculated energy (Fig. 4) and the computation time needed (Fig. 5) depending on n_c for different values of each variable parameter (such as number of segments, endpoint orientation, DLO diameter, etc.). A series of experiments is performed for each function type, in which only one parameter is varied while all other parameters are fixed. All minimization curves resulting from each series, are combined into a single diagram and compared.

Secondly, we determine which set of approximation functions handles interrupted DSM computation best. As before, we compute the minimum energy with respect to different sets of parameters, but this time the minimization is interrupted at specific intervals and the current set of coefficients is recorded. After each extraction of the coefficients, the minimization process is continued. This cycle is repeated several times for each set of parameters. Then, for each such set of coeffi-

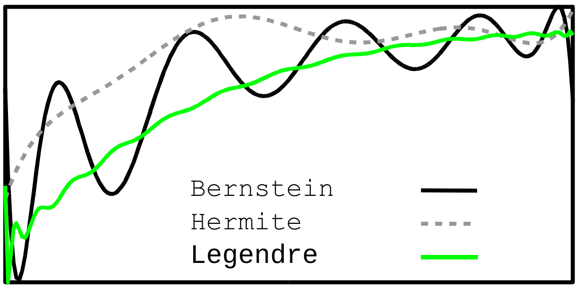

Fig. 2: Calculated bending of a DLO using Bernstein (n_c=12), Hermite (n_c=22) and Legendre (n_c=30); (n_s=200, θ_0 = -90, R=0.0005).

cients, the calculated DLO geometry (as encoded by the set of coefficients) is compared with the expected geometry and the geometry given by the other sets of coefficients. If all computed geometries (beyond a fixed and parameter-independend number of iterations) for a given set of parameters can be considered realistic, the type of approximation functions used is classified as „real-time capable“. However, it is unclear if this type of approximation functions is actually „real-time capable“ with respect to all possible parameters.

Altogether four series of experiments are performed: The first series test the limits of our model and the DSM in respect to the number of coefficients and segments as well as in respect to the resulting energy and shape. The next two series compares the different types of functions with respect to average computation time and preciseness of the computed shape and energy, while the fourth series is used to examine the real-time capabilities of our model. The parameters used in each series can be found in table 2.

Table 2 Summary of the parameter values used in some of the experiments.

Parameter	Values Experiment 1,2	Values Experiment 3	Values Experiment 4
n_c	2,...,40	2,...,20	8,16,24,32
n_s	50,100,200,250, 400	50, 100	100, 200
θ_0 (°)	-10, -90, -160	-10, -90, -160	-10, -90, -160
R (m)	0.0005, 0.0002	0.0005, 0.0002	0.0005, 0.0002

4. CONCLUSIONS

In the first series we vary both, the number of coefficients and the number of segments, using very large and very small numbers. The results of the first series of experiments indicate that not every set of approximation functions can be used to compute the bending of a DLO.

Both the Hermite and Bernstein functions calculate the bending incorrect (Fig. 2) and therefore produce incorrect DLO geometries (Fig. 3). This problem is also visible in Fig. 8, displaying wild and impossible changes in the bending energy. To find out why both sets of functions cannot not be used to properly calculate bending energies, more experiments have to carried out. Causes such as the partition of one and symmetrical effects are excluded by restricting the set to one half of the Bernstein function (B_i^n with i = 0,...,$n/2$). Because of instability in the coefficients of the bending approximation with respect to different initialization and termination criteria of the DSM, we assume that these results are due to numerical problems such as rounding errors. However, it is unclear why the calculation works well for most odd numbers of n.

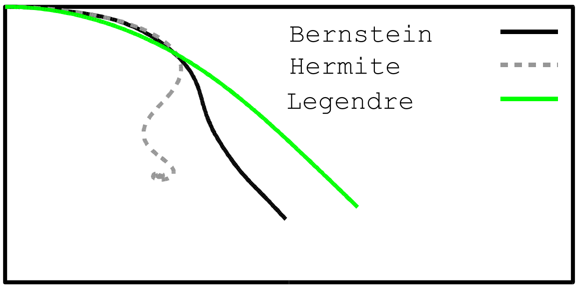

Fig. 3: DLO geometries calculated using Bernstein (n_c=12), Hermite (n_c=22), and Legendre (n_c=30); (n_s = 200, θ_0 =-90, R=0.0005)

Most likely the steepness of the Hermite functions causes their poor approximation of the bending. The Hermite functions are scaled to the length of the DLO in the interval from [-1,1]. The function values increase steeply with each new function in the Hermite sequence, so that increasingly small coefficients are needed to approximate the bending, leading to numerical instabilities due to rounding errors. Thus, Hermite and Bernstein functions are disregarded in the following experiments.

Another important point we found is, that the energy and the resulting shape is wrong if n_c reaches approximately half of n_s. The exact limit depends on the type of functions used. This problem is illustrated in Fig. 9 as a discontinuity in the energy and seems to be related to the sampling theorem as the higher order approximation functions have very high frequencies. If only a small number of segments is used (and therefore providing very few sampling points for the integration), we lose the relation between the original function and the function used to compute our integrals. The same is true for the corresponding function for the tangent angle, which itself corresponds to the stem function of the bending. Consequently, the bending and tangent functions can lose their functional relation under certain circumstances. Apparently the number of segments must be at least twice the number of coefficients.

The next two series of experiments are used to ascertain the precision and computation time of our simulation. The results from the second series are shown in Fig. 4 and Fig. 5 for a fixed value of θ_0 = -90° and averaging time and energy over n_s from 50 to 200. It can be argued that averaging the computation time for different values of n_s is not that useful as the time increases with n_s, but as our objective is to compare function types over a broad range of parameters and not to compare or find optimal parameter sets, we believe averaging the time is useful under these circumstances. Again we can see wrong energy values for high numbers of n_c.

In the final series the DSM is interrupted after 1, 2, 3 and 10 iterations and then after every 50th iteration, as described in section III and the resulting shapes are plotted in one diagramm to visualy compare the shape and to identify a fixed number of iterations after which no significant change in the shape occures.

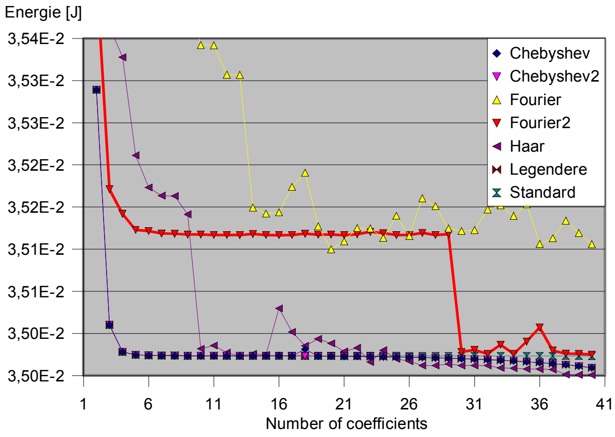

Fig. 4: Average energy after minimization (experiment 2).

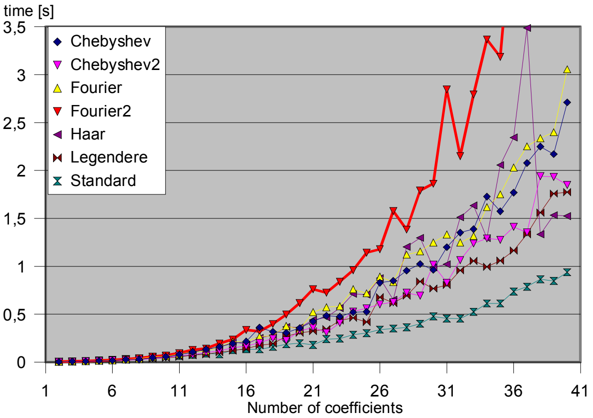

Fig. 5: Average computation time (experiment 2)

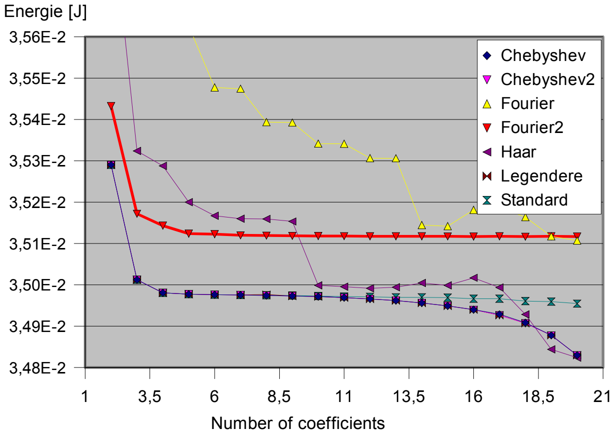

Fig. 6: Average energy after minimization (exp. 3)

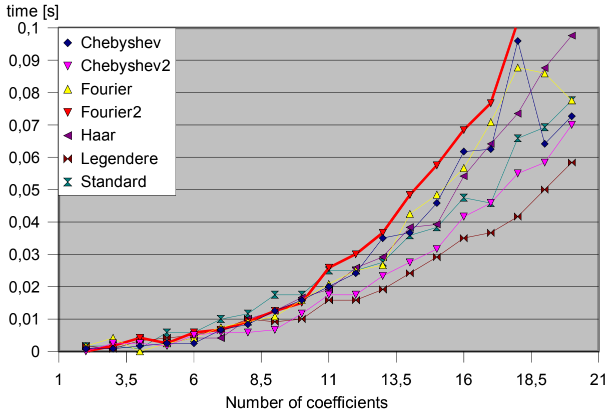

Fig. 7: Average computation time (exp.3)

The results can be summarized as follows:

1 Minimizations using the Fourier, Fourier2 and Haar function sets results in a local minimum for some values of the parameter sets ($\theta_0 = -10$, $R = 0.0002$). This is due to the manner in which the DSM is initialized.

2 The Fourier functions provide by far the worst energy approximation. While the Fourier2 functions produce better results, they still performe worse than other functions (see Fig. 4 and 6) and additionally require long computation times (Fig. 5 and 7). These properties render them unusable in our context.

3 The energy computed with respect to the number of coefficients for Legendre and Chebyshev-1 and -2 functions are almost identical. The Legendre functions require the least computation time, followed by Chebyshev-2 (see Fig. 4 - 7).

4 The Standard functions require short computation times for large n_c and n_s (5) and average computation times for small n_c and n_s (7). Because we mainly use small n_c and n_s, the Legendre functions are more appropriate in our context. Generally, the Standard functions produce good results with respect to both the calculated energy and the necessary computation time.

5 Because the Haar functions are not continuous, they generally provide good minimization results. However, the energy diagrams show a stepwise energy distribution with occasional high energy values for small n_c and n_s. The computation time is a little higher than average (See Fig. 4 and 5 as well as 6 and 7).

In terms of a fast and fairly precise simulation, all of the functions analysed (except the Fourier and Fourier2 and the early excluded Bernstein and Hermite functions) can be used. If a more precise simulation is needed, the Legendre functions are a better choice.

The fourth series of experiments shows that none of the approximation functions can produce a realistic DLO geometry under all circumstances when the computation process is interrupted at a random point. Additionally, one can see what effect varying DSM initializations have on the results when the minimization is interrupted. The DSM may produce a realistic set of coefficients if interrupted randomly during the calculation only if the set used to initialize the DSM is realistic to begin with. However, this raises the question where the realistic initial set of coefficients for the DSM comes from, if it is not first computed using this DSM. Accurate coefficients known beforehand indeed render the entire calculation unnecessary.

After the appropriate type of approximation functions and minimization method has been chosen, additional optimizations can be applied. If only the elastic bending energy is taken into account and if the approximation functions are orthogonal, we can simplify the function for the bending energy to

$$W_B = \frac{1}{8} A E R^2 \sum_{i=1}^{n_c} c_i d_i \qquad (10)$$

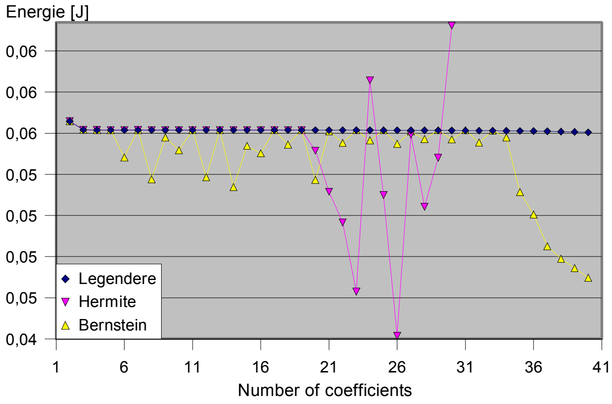

Fig. 8: Energy after minimization $(n_s = 200, \theta_0 = -90, R = 0.0005)$

with the constant d_i derived from the orthogonality of q_i :

$$\int_0^L q_i(s)\, q_j(s)\, ds = d_i \delta_{ij} \tag{11}$$

The limits of the integrals must of course be adjusted accordingly.

5. SUMMARY

We present a 2D-model to simulate a DLO and outline the model's basic characteristics based on several experiments. Our aim is to identify which approximation function from a pre-selected set of functions is best suited to simulate the DLO deformation in the presence of contact forces. We focus on short computation times and a realistic but not necessarily exact physical shape. One special requirement is to make our DSM-driven simulation as "any-time" capable as possible, i.e. it should provide a realistic coefficient set if interrupted at any point during the calculation.

We find the Bernstein functions inappropriate for our purposes. The number of useful Hermite functions are limited to the first 19th order function, as higher-order functions suffer excessively from numerical rounding errors. All other approximation functions analyzed could be used in our system, but the Legendre functions are found to be best suited for our purposes, followed by the Standard functions. This is surprising, as the Fourier functions (not the Legendre functions) are very common in many approximation systems. We cannot prove the method to be real-time or "any-time" capable under all conditions.

6. FUTURE WORK

This paper analyzes the most basic model, involving calculation of a free-hanging DLO's geometry in two-dimensional space without obstacle interaction. In this case the bending function remains nearly constant, but in the presence of obstacles the bending function will change, requiring further analysis. Similarly, consideration of dynamics will require modification of the computation method used to calculate the bending function.

Extending the model from 2D to 3D space requires approximation of three functions rather than just one. This significantly increases the number of combinations that must be taken into account; thus it might be useful to approximate different functions using different sets of approximation functions. Taking obstacles and dynamics into account will naturally result in increased complexity. This can be assumed in the 3D case too.

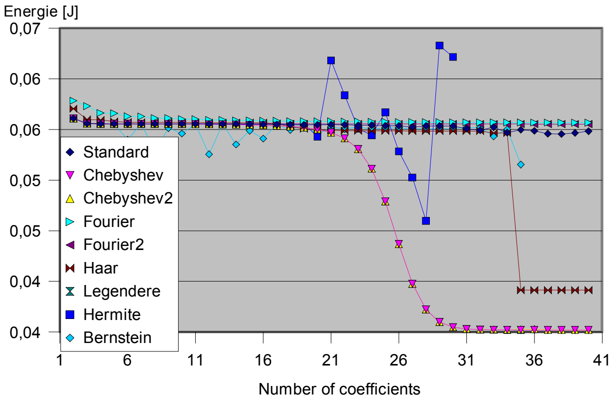

Fig. 9: Energy after minimization $(n_s = 50, \theta_0 = -90, R = 0.0005)$.

7. REFERENCES

1. D. Henrich, H. Wörn (Eds.): "Robot manipulation of deformable objects". Springer-Verlag, London, 2000, ISBN: 1-85233-250-6.
2. S. Hirai, H. Wakamatsu and K. Iwata: "Modeling of Deformable Thin Pats for Their Manipulation". In: Proceedings of the 1994 International Conference on Robotics and Automation, pp. 2955-2960, San Diego, USA, May 1994.
3. B.Kahl, D. Henrich: "Virtual Robot Programming for Deformable Linear Objects: System Concept and Prototype Implementation". In: 12th International Symposium on Measurement and Control in Robotics (ISMCR), Bourges, France, June 20 - 21, 2002.
4. S. Karl: "Physikalische Modellierung des mechanischen Verhaltens von deformierbaren linearen Objekten". Diplomarbeit, Fakultät für Physik, Universität Karlsruhe, 1999.
5. W.H. Press, B.P. Flannery, S.A. Teukolsky und W.T. Vetterling: "Numerical Recipes in Pascal". The Art of Scientific Computing, Cambridge University Press, 1989.
6. A. Remde, D. Henrich, S. Karl und H. Wörn: "Manipulating Deformable Linear Objects-Efficient Simulation of the Workpiece Behaviour". In: Int. Conf. on Robotics and Applications, Santa Barbara, California, USA, Oct. 28-30, 1999.
7. A. Schlechter und D. Henrich: "Manipulating Deformable Linear Objects: Programming using Different Manipulation Skills". In: SPIE's International Technical Group Newsletter, Robotics and Machine Perception, vol.12 no. 1, March 2003.
8. H. Wakamatsu, S. Hirai und K. Iwata: "Modelling of linear objects considering bend, twist and extensional deformations". In: Proc. 1995 Int. Conf. on Robotics and Automation (ICRA'95), vol. 1, pp. 433438, Nagoya, Japan, May 1995.
9. H. Wakamatsu, T. Matsumura, E. Arai und S. Hirai: "Dynamic Analysis of Rodlike Object Deformation towards Their Dynamic Manipulation". In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, Grenoble, France, Sept. 7-11, 19