

UMEÅ UNIVERSITY
Department of Physics

January 25, 2008

Rigid Body Simulation of Macro Molecules

Christian Svebilus
chnsws01@student.umu.se



Abstract

A computer model for simulating experiments done on surface organelles (so called pili) on the Escherichia Coli bacteria have been developed and implemented. The objective of the computer simulation was to mimic the results of experiments done with optical tweezers and to display a graphical, three dimensional, representation of these experiments.

The experiments measured the force response to elongation of pili. This force response can be divided into three regions of elongation, region I, II and III, each with different properties. Region I is characterized by a constant increase in force, in region II the Pilus is unfolded under constant force, and in region III the force versus elongation curve assumes a non-trivial shape with increasing force. The pili are also able to retract to its original length giving a similar force response curve. The computer model should be able to handle all these properties. The developed model could handle elongation in region I and II. In region III, the force response given by the simulation differed from the one given by the experiments.

Sammanfattning

En datormodell för att simulera experiment gjorda på ytorganeller (så kallade pili) på *Escherichia Coli* bakterien har utvecklats och implementerats. Syftet med arbetet var att återskapa resultat från experiment utförda med optisk pincett samt ge en grafisk, tre-dimensionell, representation av dessa experiment.

I experimenten mättes kraftresponser vid utdragning av pili. Denna kraftrespons kan delas in i tre regioner, region I, II och III, där varje region har olika egenskaper. Region I karaktäriseras av en konstant ökning av kraften, i region II vecklas pilin ut under konstant kraft och i region III antar kraftresponserna en icke-trivial form men med ökande kraft. Pili kan också återgå till sin ursprungliga längd vilket ger en liknande kraftresponsskurva. Datormodellen skall vara kapabel att hantera dessa egenskaper. Den utvecklade modellen kunde hantera region I och II. I region III gav simuleringen andra resultat än experimentet.

Contents

1	Introduction	4
1.1	The Structure of the Pilus	4
1.2	Optical tweezers	4
1.3	Measurements	5
1.4	Numerical simulation of macro molecules	8
2	Theory	8
2.1	Definitions	9
2.1.1	Coordinate Transformations	10
2.2	Rigid body dynamics	11
2.3	Constrained dynamics	12
2.4	Constraint regularization	13
3	Numerical Simulation	14
3.1	Pairwise interaction	16
3.2	Layer-to-layer interaction	18
3.3	Other forces	19
3.4	Unit normalization	20
3.5	Parameters	21
3.6	Implementation	22
3.6.1	Algorithms	22
3.6.2	Special solutions	25
3.6.3	Computational complexity	25
4	Results	26
4.1	Model validation - Region I	29
4.2	Model validation - Region II	32
4.3	Computational complexity	34
5	Discussion	36
5.1	The third region of elongation	39
5.2	Computational complexity and simplifications	39
5.3	Parameters and further simulations	42
6	Conclusion	42
7	Appendix	43
7.1	Numerical method	43
7.2	Spring coefficient approximation	43

1 Introduction

The Escherichia Coli bacteria is the cause of a majority of all uncomplicated urinary tract infections. The bacterial membrane is equipped with surface organelles, called pili, which is responsible for the adhesive properties of the bacteria. These properties have been studied and measured in [1] and [2].

The pili must withstand the hosts mechanical defenses, mainly urine flows, that exposes it to forces. The pili has evolved into a three-dimensional helix-like structure, able to unfold when subjected to external forces. It is this property of the structure that enables the bacteria to maintain contact with its host. The adhesion must be mediated by several parallel attachments, if the pili where stiff the bonds would break under any significant external force.

1.1 The Structure of the Pilus

The pili of Escherichia Coli bacteria have, in its equilibrium state, the structure of a rod with diameter $\sim 6.8nm$ and a length of $\sim 1\mu m$, and an about $15nm$ long fibrillum tip with diameter $2-3nm$. The pili are mainly composed of PapA subunits arranged in a helical shape with 3.28 units per turn and about 1000 units in total. The helical rod is fastened to the bacterial membrane by PapH and connected to the fibrillum by PapK. The fibrillum tip is in turn made of the PapE, PapF and PapG minor subunits, this is illustrated in figure 1.

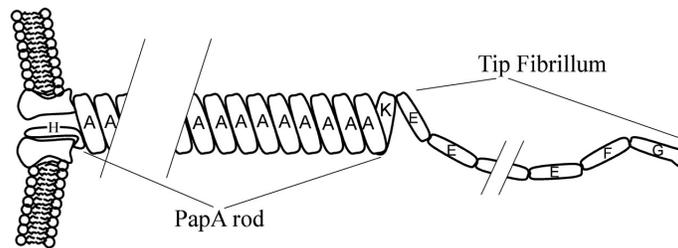


Figure 1: Schematic illustration of a Pilus, taken from [1].

1.2 Optical tweezers

Optical tweezers can be used to measure forces in the pN range. A small object (μm sized) is trapped by a highly focused laser beam. If an external force is acting on the particle it will be displaced a distance, x , from the laser focus, this will in turn result in a restoring force, shown in figure 2. For small displacements the restoring force depends linearly on the displacement distance

$$F_r = kx, \quad (1)$$

for some force constant k . In equilibrium the restoring force will be equal to the displacing external force. Optical tweezers are described in more detail in [4].

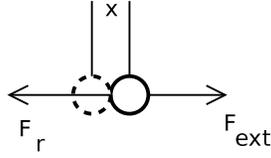


Figure 2: Particle trapped in an optical tweezers.

1.3 Measurements

By use of optical tweezers it is possible to measure the force response to elongation of individual pili on the E. Coli bacteria. This has been done in [1] by capturing a bacterium with an optical tweezers and letting it attach to a large bead. Then, when it is strongly linked, a smaller bead (captured by the optical tweezers) is placed into contact with the bacterium, shown schematically in figure 3. Now the force response of the Pilus upon unfolding can be assessed by

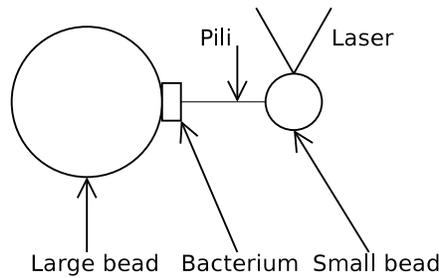


Figure 3: Bacterium attached to a small and a large bead.

measuring the position of the small bead while moving the large bead.

The response can be divided into the three regions shown in figure 4. See also figure 5 for force response of an experiment with several pili. As can be seen, there is a small jump in the response during refolding between regions II and III.

In region I there is a linear relation between the external force and elongation that is assumed to be caused by stretching of the layer-to-layer bonds. When the layer-to-layer bonds start to break the curve enters region II, given that the Pilus doesn't detach from the bead. Here the force will remain constant until all bonds are broken and the Pilus has unfolded. When the Pilus has lost its helical structure it enters the third state of elongation which is characterized by an increase in the head-to-tail angle between PapA units or an elongation of the actual PapA units themselves. The three elongation states are illustrated in figures 6 to 8.

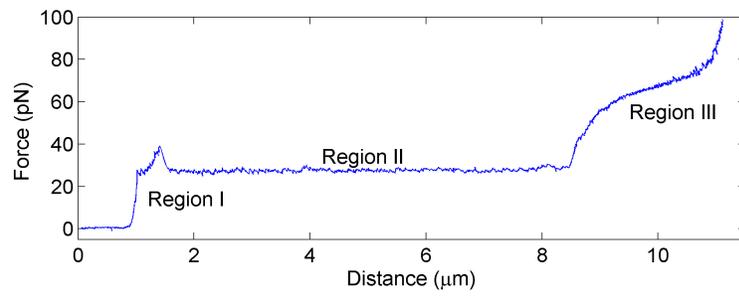


Figure 4: Force versus bead-to-bead distance of a single pili caught in optical tweezers. The figure is taken from [5].

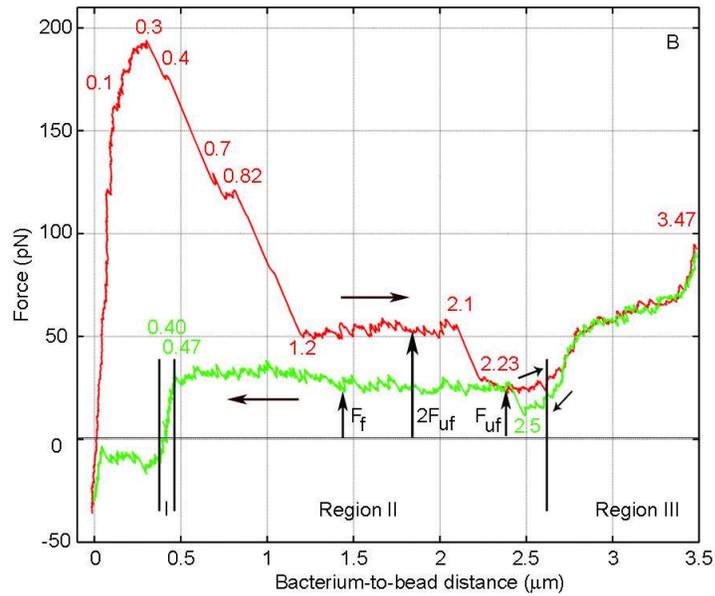


Figure 5: Force versus bead-to-bead distance of two consecutive elongation and contraction cycles displayed by red and green curves respectively. The numbers in the graphs indicate important bacterium-to-bead distances. The experiment was done on several pili and the drop in force response during unfolding is due to pili detaching. The figure was taken from [6].

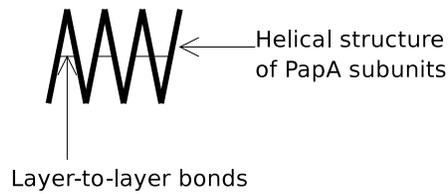


Figure 6: Elongation in region I projected into two dimensions.

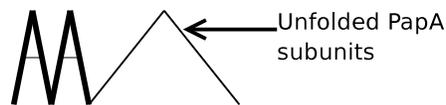


Figure 7: Elongation in region II projected into two dimensions.



Figure 8: Elongation in region III projected into two dimension.

1.4 Numerical simulation of macro molecules

The objective of this thesis is to construct a multi body system dynamic model of the helix like structure of a pili. It should be possible to perform experiments on the model, e.g. measure force response to elongation. It should be possible to adjust some of the properties of the Pilus. The model should be implemented and giving a graphical representation of the experiments on a μm scale.

There are a number of forces that have to be taken into account in the simulation:

1. A layer-to-layer force between each layer in the helix.
2. A force between each PapA block in the Pilus.
3. An external force acting to keep the Pilus attached to the bacteria.
4. An external force between the Pilus and the small bead, acting as a trap unfolding the Pilus.

It should be possible to easily adjust properties of all these forces, e.g. strength and range.

There are two main ways to perform molecular computer simulations, those are molecular dynamics and Monte Carlo simulation. The Monte Carlo method relies on repeated samplings using a random number distribution. The model developed in this thesis uses molecular dynamics which is further described in [8].

The developed model uses rigid bodies to simulate individual PapA subunits in the Pilus, this makes it easy to give a graphical and meaningful representation of the experiments. It was implemented using a method called constraints regularization, explained in sections 2.3 and 2.4. The method has previously been used in simulation and visualization of a rigid body cable, see [7]. Constraints regularization are also explained in more detail in [9].

2 Theory

In the following section, concepts necessary to understand the chosen approach for simulating macro molecules will be presented. These include rigid body dynamics and constraint regularization.

2.1 Definitions

The following entities need to be defined to describe a system of rigid bodies. Capital letters will be used to represent matrices and generalized vectors¹. Lower case letters will be used to denote object specific (non generalized) vectors. It is assumed to be N bodies in the system.

I further let subscripts represent the component and superscripts denotes which object the vector or matrix belongs to. If nothing else is mentioned, all vectors are column vectors. The unit matrix with dimensions a, b will be written $\mathbf{1}_{a,b}$, and the zero matrix with the same dimensions will be written $\mathbf{0}_{a,b}$. If no subscript is given for the zero matrix ($\mathbf{0}$), then it is assumed to be large enough to fill the elements that are not specified. This is used in diagonal and block diagonal matrices:

$$\begin{pmatrix} x_1 & & \mathbf{0} \\ & x_2 & \\ \mathbf{0} & & x_3 \end{pmatrix} \equiv \begin{pmatrix} x_1 & 0 & 0 \\ 0 & x_2 & 0 \\ 0 & 0 & x_3 \end{pmatrix}. \quad (2)$$

Let \mathbf{X} be a generalized position vector defined as

$$\mathbf{X} = [(\mathbf{x}^1)^T \quad (\mathbf{q}^1)^T \quad (\mathbf{x}^2)^T \quad (\mathbf{q}^2)^T \dots (\mathbf{x}^N)^T \quad (\mathbf{q}^N)^T]^T, \quad (3)$$

where \mathbf{x}^i is the position of object i relative some coordinate system, it has three components. The vector \mathbf{q}^i is the quaternion (see [3]) representing the orientation of object i , it has four components. This gives the generalized position vector the dimension

$$\mathbf{X} \in \mathbb{R}^{7N}. \quad (4)$$

The entity X is, at each point in time, enough to describe the state of a system of rigid bodies completely.

Further, let \mathbf{V} be the generalized velocity vector,

$$\mathbf{V} = [(\mathbf{v}^1)^T \quad (\omega^1)^T \quad (\mathbf{v}^2)^T \quad (\omega^2)^T \dots (\mathbf{v}^N)^T \quad (\omega^N)^T]^T, \quad (5)$$

where \mathbf{v}^i is the linear velocity vector and ω^i is the angular velocity of object i . As both \mathbf{v}^i and ω^i are three dimensional vectors; $\mathbf{V} \in \mathbb{R}^{6N}$.

In line with the previous definition, the generalized force is defined as

$$\mathbf{F} = [(\mathbf{f}^1)^T \quad (\tau^1)^T \quad (\mathbf{f}^2)^T \quad (\tau^2)^T \dots (\mathbf{f}^N)^T \quad (\tau^N)^T]^T, \quad (6)$$

where $\mathbf{f}^i \in \mathbb{R}^3$ is the linear force and $\tau^i \in \mathbb{R}^3$ is the torque of object i . The generalized force vector do have the same dimension as the generalized velocity:

$$\mathbf{F} \in \mathbb{R}^{6N}. \quad (7)$$

¹Vectors describing a property of more than one object. Often comprised of several object specific vectors.

Now, define a general mass matrix to contain both the mass and the inertia tensor of all objects in the following block diagonal form

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}^1 & & & \mathbf{0} \\ & \mathbf{M}^2 & & \\ & & \ddots & \\ \mathbf{0} & & & \mathbf{M}^N \end{pmatrix}, \quad (8)$$

where \mathbf{M}^i contains the mass (m^i) and inertia tensor (\mathbf{I}^i) of object i in the form

$$\mathbf{M}^i = \begin{pmatrix} m^i \mathbf{1}_{3,3} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & \mathbf{I}^i \end{pmatrix}. \quad (9)$$

This gives the generalized mass matrix the following dimension

$$\mathbf{M} \in \mathbb{R}^{6N \times 6N}. \quad (10)$$

A way to describe the relation between the time derivative of the position, $\dot{\mathbf{X}}$, and the velocity \mathbf{V} is also needed. For this I define

$$\dot{\mathbf{X}} = \mathbf{T}\mathbf{V}, \quad (11)$$

where \mathbf{T} is the block diagonal matrix

$$\mathbf{T} = \begin{pmatrix} \mathbf{1}_{3,3} & & & \mathbf{0} \\ & \mathbf{Q}^1 & & \\ & & \ddots & \\ \mathbf{0} & & & \mathbf{1}_{3,3} & & \mathbf{Q}^N \end{pmatrix}. \quad (12)$$

This is a logical assumption: because $\dot{\mathbf{x}}^i = \mathbf{v}^i$, while $\dot{\mathbf{q}}^i$ does not even have the same dimension as ω^i . There is a need for a transformation matrix, $\mathbf{Q}^i \in \mathbb{R}^{4 \times 3}$, to transform the former entity to the latter. In [3] it has been shown that this is the required matrix:

$$\mathbf{Q}^i = \frac{1}{2} \begin{pmatrix} -q_2^i & -q_3^i & -q_4^i \\ q_1^i & q_4^i & -q_3^i \\ -q_4^i & q_1^i & q_2^i \\ q_3^i & -q_2^i & q_1^i \end{pmatrix}. \quad (13)$$

2.1.1 Coordinate Transformations

All vectors and matrices are assumed to be given relative a global coordinate system that is independent of the objects. It may be necessary or convenient to define several coordinate systems local to objects in space. Vectors and matrices that is expressed relative the local coordinate system of object i will be given the subscript $l(i)$, e.g. $\mathbf{v}_{l(i)}$ represents the velocity vector relative the local coordinate system of object i .

The inertia tensor relative a local coordinate system, $\mathbf{I}_{l(i)}^i$, will generally be constant because the coordinate system rotates with its object. The same inertia tensor given relative the global coordinate system, \mathbf{I}^i , could on the other hand vary if the object changes orientation in time. It is possible to transform an inertia tensor from the local to the global coordinate system by a coordinate transformation, given that the local inertia tensor is symmetric, with the following transformation:

$$\mathbf{I}^i = \mathbf{R}^i \mathbf{I}_{l(i)}^i (\mathbf{R}^i)^T. \quad (14)$$

Here, \mathbf{R}^i is the transformation matrix from $l(i)$ to the global coordinate system. \mathbf{R}^i can, according to [3], be written with the help of the object quaternion in the following way:

$$\mathbf{R}^i = \begin{pmatrix} 1 - 2((q_3^i)^2 + (q_4^i)^2) & 2q_2^i q_3^i - 2q_1^i q_4^i & 2q_1^i q_3^i + 2q_2^i q_4^i \\ 2q_2^i q_3^i + 2q_1^i q_4^i & 1 - 2((q_2^i)^2 + (q_4^i)^2) & -2q_1^i q_2^i + 2q_3^i q_4^i \\ -2q_1^i q_3^i + 2q_2^i q_4^i & 2q_1^i q_2^i + 2q_3^i q_4^i & 1 - 2((q_2^i)^2 + (q_3^i)^2) \end{pmatrix}. \quad (15)$$

2.2 Rigid body dynamics

A system of N rigid bodies is governed by Newton-Euler's equations of motion

$$\dot{\mathbf{V}} = (\mathbf{M})^{-1} \mathbf{F}, \quad (16)$$

where the force may be separated in three parts

$$\mathbf{F} = \mathbf{F}_{constraint} + \mathbf{F}_{external} + \mathbf{F}_{gyroscopic}. \quad (17)$$

The constraint force is the force originating in the constraints placed on the system. They can be seen as internal forces limiting the movement of the objects. The gyroscopic forces comes from rotation of asymmetric bodies and is calculated as

$$\mathbf{F}_{gyroscopic} = -\dot{\mathbf{M}}\mathbf{V}, \quad (18)$$

where the time derivative of the individual mass matrices can be shown to be

$$\dot{\mathbf{M}}^i = \begin{pmatrix} \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \\ \mathbf{0}_{3,3} & (\omega^i)^x \mathbf{I}^i + \mathbf{I}^i (\omega^i)^x \end{pmatrix}. \quad (19)$$

The cross product matrix², $(\omega^i)^x$, is defined to be

$$(\omega^i)^x = \begin{pmatrix} 0 & -\omega_3^i & \omega_2^i \\ \omega_3^i & 0 & -\omega_1^i \\ -\omega_2^i & \omega_1^i & 0 \end{pmatrix}. \quad (20)$$

All other forces are called external. Equation (11) and (16) together describes the dynamics of a system if the forces are known.

²The cross product matrix is actually given by the relation $\mathbf{a} \times \mathbf{b} = \mathbf{a}^x \mathbf{b}$.

2.3 Constrained dynamics

There is a need to handle internal forces in the system, e.g. connection between rigid bodies and connections between external objects and the bodies of the simulation. In other words, all things that restricts, or constrains, the movement of the simulated objects. If it is possible to determine a potential, U , for these constraints, then it is theoretically possible to determine all resulting forces directly by using

$$\mathbf{F} = \nabla U. \quad (21)$$

This easily leads to numerical instabilities, as the internal forces can become very large even for small changes in position from equilibrium. It would generally require small time-steps and expensive integration methods to get a stable simulation. And even then, it is not certain that it is possible to formulate a potential function that gives the wanted effect.

An alternative way is by constrained dynamics. First, define a constraint function

$$\begin{aligned} \Phi(\mathbf{X}) &\in \mathbb{R}^{N_c}, \\ \mathbf{X} &\in \mathbb{R}^{7N}, \end{aligned}$$

where \mathbf{X} is the generalized state vector,³ and N_c is the number of constraints placed on the system. This function determines a geometric relation between bodies and can be constructed to represent physical properties of the system. In this project, there is a need for constraints locking objects in place, and effectively reducing the total degrees of freedom. These are called holonomic constraints and are of the form

$$\Phi(\mathbf{X}) = \mathbf{0}. \quad (22)$$

This equation specifies a surface in generalized position space on which the system is restricted to and reduces the degrees of freedom of the system by N_c .

What is now needed is a way to calculate the constraint force, $\mathbf{F}_{constraint}$, that enforces the constraints. First take the time derivative of $\Phi(\mathbf{X})$:

$$\dot{\Phi}(\mathbf{X}) = \frac{\partial \Phi}{\partial \mathbf{X}} \dot{\mathbf{X}} = \frac{\partial \Phi}{\partial \mathbf{X}} \mathbf{T} \mathbf{V} = \mathbf{0}, \quad (23)$$

which is true because the constraint function is defined as constant zero. The Jacobian is defined as

$$\mathbf{G} \equiv \frac{\partial \Phi}{\partial \mathbf{X}} \mathbf{T},$$

which gives

$$\dot{\Phi}(\mathbf{X}) = \mathbf{G} \mathbf{V} = \mathbf{0}. \quad (24)$$

³The state vector consist of a position vector and a 4 dimensional quaternion. Therefore, \mathbf{X} must be $7N$ dimensional.

The constraints keeps the objects on a specified surface keeping $\dot{\Phi} = \mathbf{0}$, adding the requirement that the restoring constraint force do no work

$$\mathbf{F}_{constraint} \cdot \mathbf{V} = 0. \quad (25)$$

According to [3], this together with equation (23) leads to

$$\mathbf{F}_{constraint} = \mathbf{G}^T \lambda, \quad (26)$$

for some

$$\lambda \in \mathbb{R}^{N_c}.$$

One way to determine λ is to take the second time derivative of $\Phi(\mathbf{X})$

$$\ddot{\Phi}(\mathbf{X}) = \dot{\mathbf{G}}\mathbf{V} + \mathbf{G}\dot{\mathbf{V}} = \mathbf{0}. \quad (27)$$

Replacing $\dot{\mathbf{V}}$ by $\mathbf{M}^{-1}\mathbf{F}$ gives

$$\dot{\mathbf{G}}\mathbf{V} + \mathbf{G}\mathbf{M}^{-1}(\mathbf{F}_{external} + \mathbf{F}_{gyroscopic} + \mathbf{F}_{constraint}) = \mathbf{0}, \quad (28)$$

so finally

$$\mathbf{G}\mathbf{M}^{-1}\mathbf{G}^T \lambda = -\dot{\mathbf{G}}\mathbf{V} - \mathbf{G}\mathbf{M}^{-1}(\mathbf{F}_{external} + \mathbf{F}_{gyroscopic}). \quad (29)$$

To summarize, the governing equation of a constrained rigid body system can be expressed

$$\dot{\mathbf{X}} = \mathbf{T}\mathbf{V}, \quad (30)$$

$$\dot{\mathbf{V}} = \mathbf{M}^{-1}(\mathbf{F} + \mathbf{G}^T \lambda), \quad (31)$$

$$\mathbf{G}\mathbf{M}^{-1}\mathbf{G}^T \lambda = -\dot{\mathbf{G}}\mathbf{V} - \mathbf{G}\mathbf{M}^{-1}(\mathbf{F}_{external} + \mathbf{F}_{gyroscopic}). \quad (32)$$

Using equations (30) to (32) enforces $\Phi = \mathbf{0}$ strictly. Unfortunately, this could cause numerical drift in a simulation. There are several ways of correcting and stabilizing the method, one is to add correction terms to λ ,

$$\lambda \rightarrow \lambda + \kappa_s \Phi + \kappa_a \dot{\Phi},$$

where κ_a and κ_s are parameters that has to be guessed or determined in some way. The method which I have used is constraint regularization, described in section 2.4.

2.4 Constraint regularization

If the potential energy of a closed system can be written as

$$U(\mathbf{X}) = \frac{1}{2} \Phi^T(\mathbf{X}) \epsilon^{-1} \Phi(\mathbf{X}), \quad (33)$$

for some arbitrary function $\Phi(\mathbf{X})$ and diagonal matrix ϵ . Then the internal forces in the system can be calculated

$$\mathbf{F}_{constraint} = -\mathbf{T}^T \frac{dU}{d\mathbf{X}} = -\mathbf{T}^T \frac{d\Phi^T}{d\mathbf{X}} \epsilon^{-1} \Phi = -\mathbf{G}^T \epsilon^{-1} \Phi, \quad (34)$$

where

$$\mathbf{G} = \frac{d\Phi}{d\mathbf{X}}\mathbf{T}. \quad (35)$$

By defining

$$\lambda \equiv -\epsilon^{-1}\Phi, \quad (36)$$

the force becomes

$$\mathbf{F}_{constraint} = \mathbf{G}^T\lambda. \quad (37)$$

The final equations for regularized constraints will look similar to the constraint equations shown in section 2.3:

$$\dot{\mathbf{X}} = \mathbf{T}\mathbf{V}, \quad (38)$$

$$\dot{\mathbf{V}} = \mathbf{M}^{-1}(\mathbf{F}_{external} + \mathbf{F}_{gyroscopic} + \mathbf{G}^T\lambda), \quad (39)$$

$$\lambda = -\epsilon^{-1}\Phi. \quad (40)$$

This is no longer a system of differential equations but a differential algebraic equation. Equations (38) to (40) can be solved numerically using a semi implicit Euler algorithm. This is shown in section 7.1. The advantage of using this method is the stability; the parameter ϵ can be adjusted to avoid instabilities and numerical drift as it controls the strength of the potential. Also, the resulting forces will correspond to physical forces originating in the potential U in equation (33). It is possible to first define a potential, then find the required constraint function giving that potential.

3 Numerical Simulation

The objective of this thesis is to simulate experiments done on a E. Coli bacterial pili. These experiments are described in more detail in section 1, as well as details about the Pilus structure. The Pilus should be fastened to a cell membrane in one side and to a trap in the other side. Then the trap should be able to move with a constant velocity, while recording the force response of the unfolding Pilus.

The Pilus is modeled as several interconnected rigid bodies representing the PapA units constructing the real-world Pilus. These rigid bodies, or blocks, are geometrically cuboids with side lengths a , b and c . Each block is assigned a local coordinate system with unit perpendicular axes. These axes are, for block i , given relative a chosen global coordinate system, \mathbf{a}^i , \mathbf{b}^i and \mathbf{c}^i , see figure 9. Positions and tensors will from here be denoted by an $l(i)$ subscript if it is given relative to the local coordinate system of block i .

As each block has a uniform density the inertia tensors in the local frames are given by

$$\mathbf{I}_{l(i)}^i = \begin{pmatrix} a^2 + b^2 & 0 & 0 \\ 0 & b^2 + c^2 & 0 \\ 0 & 0 & a^2 + c^2 \end{pmatrix}. \quad (41)$$

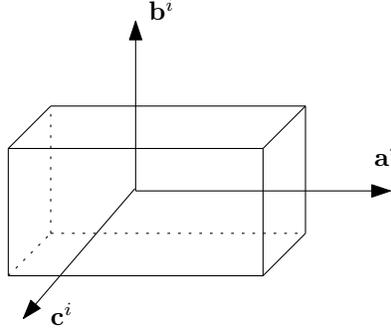


Figure 9: The local coordinate system of block i .

These matrices are symmetric so it is possible to transform them to the global frame using the transformation in (14). This is useful because the local inertia tensor is, as can be seen, constant in time. So, it is only necessary to calculate it once, then use a coordinate transformation to transform it into the global frame at each time step.

Each block is connected to the next in connection points placed at the center of two of its sides, see figure 10. In the equilibrium state, these blocks are oriented in a helical shaped chain with 3.28 units per turn. The orientation is set using the following algorithm:

1. Give block 1 some random orientation.
2. Let $i \rightarrow 2$.
3. Give block i the same orientation as block $i - 1$.
4. Rotate block i $\frac{2\pi}{3.28}$ around axis \mathbf{c}^i .
5. Rotate block i r around axis \mathbf{b}^i , where r is an angle that determines the layer to layer distance. It should be set so that it brings the structure to initial equilibrium.⁴
6. Increment i by one.
7. Repeat steps 3 to 6 until $i = N$.
8. Rearrange block 1 so that axis \mathbf{a}^1 is perpendicular to \mathbf{a}^2 .
9. Rearrange block N so that axis \mathbf{a}^N is perpendicular to \mathbf{a}^{N-1} .

This orientation is partly maintained by a potential depending on the relative angles between adjacent blocks, and partly by a potential depending on the distance between adjacent layers in the helical structure. These two potentials are defined in sections 3.1 and 3.2.

⁴The angle, r , can be found by trial and error, either manually or with the help of a computer program.

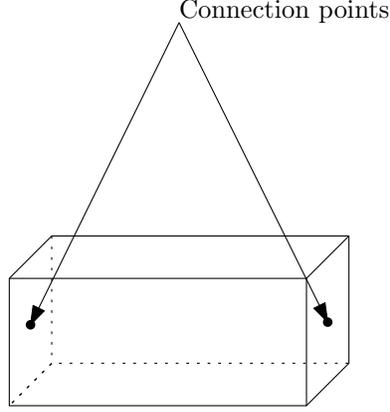


Figure 10: The connection points of a block.

3.1 Pairwise interaction

The pili are divided into blocks of rigid bodies, each with length a , width b and height c . The blocks are connected pairwise by a strong potential, U_0 , dependent on the distance between pairs of blocks. There is also a potential, U_1 to U_3 , acting to keep the equilibrium angle between blocks. The potentials are defined like this

$$U_0 = \frac{2}{\epsilon_0} \sum_i (\mathbf{x}_+^i - \mathbf{x}_-^{i+1})^2, \quad (42)$$

$$U_1 = \frac{2}{\epsilon_1} \sum_i (\theta_a^{i,i+1} - \Theta_a^{i,i+1})^2, \quad (43)$$

$$U_2 = \frac{2}{\epsilon_2} \sum_i (\theta_b^{i,i+1} - \Theta_b^{i,i+1})^2, \quad (44)$$

$$U_3 = \frac{2}{\epsilon_3} \sum_i (\theta_c^{i,i+1} - \Theta_c^{i,i+1})^2, \quad (45)$$

where

$$\mathbf{x}_+^i = \mathbf{x}^i + \frac{1}{2}a\mathbf{a}^i, \quad (46)$$

and

$$\mathbf{x}_-^i = \mathbf{x}^i - \frac{1}{2}a\mathbf{a}^i, \quad (47)$$

are vectors pointing to the edges of the blocks, as shown in figure 11. The entity $\theta_k^{i,i+1}$ is the angle between the coordinate axes k of body i and $i+1$. $\Theta_k^{i,i+1}$ is the equilibrium state of $\theta_k^{i,i+1}$. The ϵ_k is a constant determining the strength of the potential.

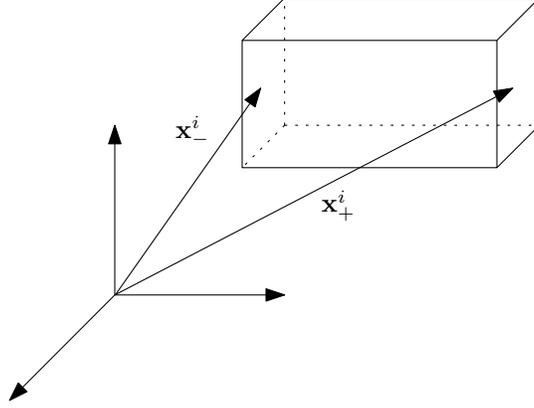


Figure 11: Edge vectors \mathbf{x}_+^i and \mathbf{x}_-^i .

Each of these potentials can be written in the form of constraints, using equation (33):

$$\Phi_{4(i-1)+1} = \mathbf{x}_+^i - \mathbf{x}_-^{i+1}, \quad (48)$$

$$\Phi_{4(i-1)+2} = \theta_a^{i,i+1} - \Theta_a^{i,i+1}, \quad (49)$$

$$\Phi_{4(i-1)+3} = \theta_b^{i,i+1} - \Theta_b^{i,i+1}, \quad (50)$$

$$\Phi_{4(i-1)+4} = \theta_c^{i,i+1} - \Theta_c^{i,i+1}, \quad (51)$$

with the total constraint vector

$$\Phi = [\Phi_1^T \quad \Phi_2 \quad \Phi_3 \quad \Phi_4 \cdots \Phi_{4(N-1)+1}^T \quad \Phi_{4(N-1)+2} \quad \Phi_{4(N-1)+3} \quad \Phi_{4(N-1)+4}]^T. \quad (52)$$

The Jacobian can be calculated from these constraints using the relation

$$\mathbf{G}\mathbf{V} = \dot{\Phi}, \quad (53)$$

which gives the block diagonal Jacobian

$$\mathbf{G} = \begin{pmatrix} \mathbf{G}_{1,1} & \mathbf{G}_{1,2} & & & 0 \\ & \mathbf{G}_{2,2} & \mathbf{G}_{2,3} & & \\ & & \ddots & \ddots & \\ 0 & & & \mathbf{G}_{N_c,N-1} & \mathbf{G}_{N_c,N} \end{pmatrix}, \quad (54)$$

where

$$\mathbf{G}_{i,i} = \begin{pmatrix} \mathbf{1}_{3,3} & -(\mathbf{a}\mathbf{a}^i)^\times \\ \mathbf{0}_{1,3} & \mathbf{A}^T \\ \mathbf{0}_{1,3} & \mathbf{B}^T \\ \mathbf{0}_{1,3} & \mathbf{C}^T \end{pmatrix}, \quad (55)$$

and

$$\mathbf{G}_{i,i+1} = - \begin{pmatrix} \mathbf{1}_{3,3} & -(-a\mathbf{a}^{i+1})^\times \\ \mathbf{0}_{1,3} & \mathbf{A}^T \\ \mathbf{0}_{1,3} & \mathbf{B}^T \\ \mathbf{0}_{1,3} & \mathbf{C}^T \end{pmatrix}. \quad (56)$$

The vector $a\mathbf{a}^i$ is pointing to the forward edge of block i and $-a\mathbf{a}^i$ is pointing to in the opposite direction, this is shown in figure 12. The vectors \mathbf{A} , \mathbf{B} and

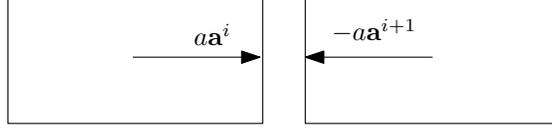


Figure 12: Vectors $a\mathbf{a}^i$ and $-a\mathbf{a}^{i+1}$.

\mathbf{C} can be calculated

$$\mathbf{A} = -\frac{1}{(a_\times)^2 + (a.)^2} \left[a_\times + \frac{(a.)^2}{a_\times} \right] \mathbf{a}_\times, \quad (57)$$

$$\mathbf{B} = -\frac{1}{(b_\times)^2 + (b.)^2} \left[b_\times + \frac{(b.)^2}{b_\times} \right] \mathbf{b}_\times, \quad (58)$$

$$\mathbf{C} = -\frac{1}{(c_\times)^2 + (c.)^2} \left[c_\times + \frac{(c.)^2}{c_\times} \right] \mathbf{c}_\times, \quad (59)$$

where $a. = \mathbf{a}^i \cdot \mathbf{a}^{i+1}$, $\mathbf{a}_\times = \mathbf{a}^i \times \mathbf{a}^{i+1}$ and $a_\times = |\mathbf{a}^i \times \mathbf{a}^{i+1}|$. The rest follows similarly⁵.

3.2 Layer-to-layer interaction

There is also an interaction between each layer of the Pilus. This interaction should be strong for small distances but quickly decrease once the distance gets larger. This is done by assigning each block two points with different ‘colors’ (red and blue), see figure 13. For performance reasons, only the points on every

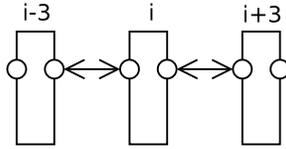


Figure 13: Three PapA units with corresponding layer-to-layer potential points.

third block effect each other⁶, which is possible because of the fact that the

⁵There is a numerical singularity when a_\times equals zero. This can be avoided by adding a small number to the a_\times term.

⁶This means every block will interact with a block one turn forward and one turn backward in the helical structure, as there are 3.28 blocks per turn.

interaction should be short range. The red points are placed in the center of one of the large sides of each block. Then the blue points are placed on the opposite side so that the distance in the beginning state between red point on block $i - 3$ and blue point on block i is minimized (see figure 14).

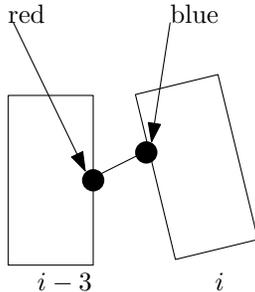


Figure 14: The placement of red and blue points.

The layer-to-layer interaction is modeled by an exponential potential

$$U_{LL} = -U_0 e^{-\frac{r^2}{L_{LL}^2}}, \quad (60)$$

where r is the distance between a red and a blue point, U_0 is the potential depth and L_{LL} is a characteristic length of the potential. An example of this potential with U_0 and L_{LL} set to one is shown in figure 15. It should be noted that this potential is not implemented using constraints. The resulting forces are instead found using

$$\mathbf{F}_{LL} = -\nabla U_{LL}, \quad (61)$$

directly.

3.3 Other forces

The Pilus is exposed to three different external forces in the simulation. The first block in the structure is fastened in a cell membrane, this is modeled by locking it in the starting position.

The last block is effected by a trap potential representing the optical tweezers. This trap force is modeled by a spring potential

$$U_{Trap} = \frac{2}{\epsilon_T} (\mathbf{x}^{Trap} - \mathbf{x}^N)^2, \quad (62)$$

where \mathbf{x}^{Trap} is the trap position and ϵ_T is the inverse potential strength. This gives the following constraint function:

$$\Phi_T = \mathbf{x}^{Trap} - \mathbf{x}^N. \quad (63)$$

The trap is, at the beginning of the simulation, placed at the center of the last block. Then it is moved with a constant velocity away from the Pilus until it

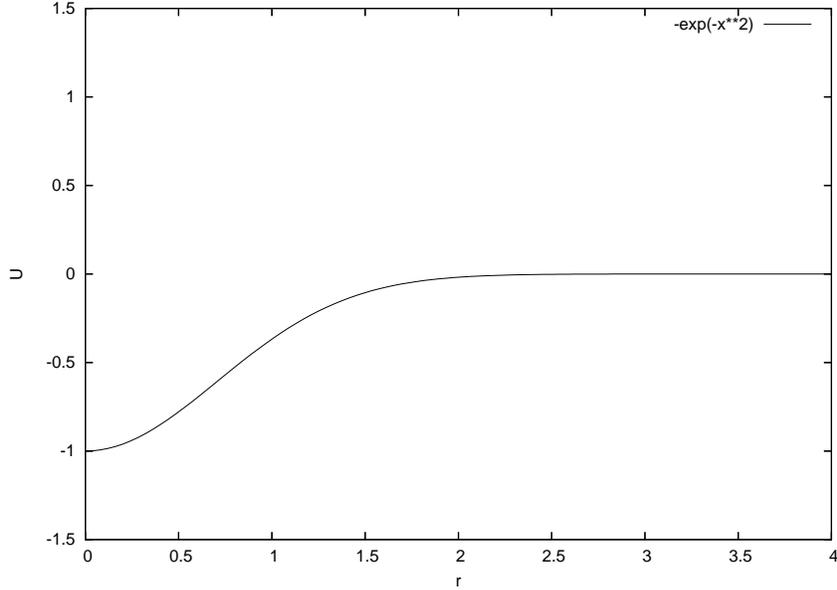


Figure 15: Layer-to-layer potential as a function of point-to-point distance with unit parameters.

reaches a predetermined distance at which the velocity is reversed. The trap then moves back until the Pilus is in its original position.

All blocks are subject to a friction force, causing energy dissipation. This force is defined as an external force

$$\mathbf{F}_v^i = -k_v \mathbf{v}^i, \quad (64)$$

$$\mathbf{F}_\omega^i = -k_\omega \omega^i, \quad (65)$$

where k_v and k_ω are coefficients determining the strength of the dampening force. This gives the following total dampening force

$$\mathbf{F}_f = -[k_v(\mathbf{v}^1)^T \quad k_\omega(\omega^1)^T \dots k_v(\mathbf{v}^N)^T \quad k_\omega(\omega^N)^T]^T. \quad (66)$$

3.4 Unit normalization

While running the simulation, all units used are simulation units. It has to be possible to transform between these simulation units and real world units, such a transformation will be given in this section. The simulation units for length and force are named in the following way

$$\begin{aligned} S_L &= [length], \\ S_F &= [force]. \end{aligned}$$

These units can be determined to fit with the experimental results.

The pili length should be $\sim 1\mu m$ and it consist of about 1000 PapA subunits. Assuming that the rest length scales linearly with the number of blocks, it can be determined from measured values in the simulation that a Pilus with 1000 blocks should have the length $320S_L$. This gives

$$1S_L = \frac{1}{320}\mu m. \quad (67)$$

The force response in region II is increasing with the number of blocks. If this increase is assumed to be linear the height using 1000 blocks would be $1.13 \cdot 10^{-2}S_F$. The force of a single pili in elongation region II should be $27pN$ according to experiments ([5]), this gives

$$1S_F = \frac{27}{1.13 \cdot 10^{-2}}pN. \quad (68)$$

3.5 Parameters

The following parameters, shown in table 1, where chosen to be used in the main simulation. All values are given in simulation units.

Table 1: Parameter values mostly used in the simulation, given in simulation units.

Parameter	Description	Value
U_0	Layer-to-layer potential depth	0.005
L_{LL}	Layer-to-layer potential characteristic length	0.75
m	Mass of PapA blocks	0.0004
k_v	Linear friction force coefficient	$5 \cdot 10^{-5}$
k_ω	Angular friction coefficient	10^{-5}
ϵ_0	Pairwise linear inverse potential strength	0
ϵ_1	Pairwise angular inverse potential strength	125
ϵ_2	Pairwise angular inverse potential strength	260
ϵ_3	Pairwise angular inverse potential strength	260
ϵ_T	Trap inverse potential strength	100
V_t	Speed of trap	0.1
a	Base length of PapA blocks	2
b	Width of PapA blocks	1
c	Height of PapA blocks	1
N	The number of blocks	100
Δt	The size of the time step	0.01

3.6 Implementation

The following software and hardware was used to implement the model and run simulations on it.

- Hardware
 - CPU - AMD Athlon XP 1800 @ 1.5GHz
 - Internal memory - 256 MB
- Software
 - MATLAB 6.5
 - Programming language - C

The rest of this section describes the details of the implementation as well as the algorithms used.

3.6.1 Algorithms

The following general algorithm was used to implement the model. The details of the more complex steps will be explained later in this section.

1. Set all constant parameters, e.g., mass of blocks and their dimensions, potential strengths, length of time step. Also place the blocks in a starting configuration and set all derived parameters like generalized mass matrix.

2. Start the time stepping.
3. Calculate Φ , \mathbf{G} , \mathbf{G}^T and $\mathbf{F}_{external}$.
4. Solve the equation of motion, this will give updated generalized position and velocity vectors.
5. Update the derived object parameters according to the new velocity and position vectors.
6. Move trap forward/backward depending on if the Pilus has reached its maximal elongation or not.
7. Save the force on the Pilus by the trap during this time step together with the length of the Pilus.
8. Repeat step 2 to 8 until a stop condition is reached. This stop condition is given by the user.
9. Save all variables to a file.

In point 1 the constant parameters include all parameters mentioned in table 1, they are set to the values given in the table. The PapA blocks are placed in their equilibrium positions using the algorithm given in the beginning of this section, when that have been done the derived parameters could be determined and stored. These include:

- The body centered inertia tensors and their inverses.
- The generalized mass matrix and its inverse.
- The transformation matrix given in equation (12).
- The equilibrium angles between each pair of blocks, these are the same as the initial angles.
- The position of the two connection points on each block.
- The diagonal matrix ϵ . The components of this matrix are ϵ_0 , ϵ_1 , ϵ_2 , ϵ_3 and ϵ_T given in table 1. These are actually duplicated for each block as there need to be one component for each constraint⁷.

In point 3 the vectors and matrices Φ , \mathbf{G} and $\mathbf{F}_{external}$ are calculated. Φ is a vector determining the constraints⁸. It is mainly defined in section 3.1, with one addition given by equation (63). The Jacobian, \mathbf{G} , is also given in section

⁷Remember that the components in the diagonal of the ϵ matrix determines how stiff the corresponding constraint is.

⁸Which means that the forces derived from the constraints will try to keep $\Phi = 0$.

3. Using that $\mathbf{G}\mathbf{V} = \dot{\Phi}$ and $\Phi_T = \mathbf{x}^N - \mathbf{x}^{Trap}$ the trap component of \mathbf{G} is given by

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} v_1^N \\ v_2^N \\ v_3^N \\ \omega_1^N \\ \omega_2^N \\ \omega_3^N \end{pmatrix} = \frac{d}{dt}(\mathbf{x}^N - \mathbf{x}^{Trap}). \quad (69)$$

The trap position can be considered to be a constant, as the trap itself is not effected by any constraints and is in a way external to the whole physical model. The external forces include the friction forces which are for block i

$$-k_v \mathbf{v}^i,$$

and

$$-k_\omega \omega^i,$$

respectively. The layer to layer forces are also considered external in this simulation, these are defined in section 3.2. Finally the gyroscopic force is added to $\mathbf{F}_{external}$:

$$\mathbf{F}_{gyroscopic} = -\dot{\mathbf{M}}\mathbf{V}. \quad (70)$$

The equations of motion solved in point 4 are the ones given in equations (38) to (40). These are solved by first defining

$$\mathbf{S} = \mathbf{G}\mathbf{M}^{-1}\mathbf{G}^T + \Delta t^{-2}\epsilon, \quad (71)$$

called the Schur complement. Then, one can get $\lambda(t + \Delta t)$ using

$$\lambda(t + \Delta t) = \mathbf{S}^{-1} \left(\frac{-\mathbf{G}^T\mathbf{V} - \Delta t\mathbf{G}\mathbf{M}^{-1}\mathbf{F}_{external} - \Delta t^{-1}\Phi}{\Delta t} \right). \quad (72)$$

$\mathbf{V}(t + \Delta t)$ and $\mathbf{X}(t + \Delta t)$ can then be solved for by using

$$\mathbf{V}(t + \Delta t) = \mathbf{M}^{-1}(\mathbf{G}^T\lambda(t + \Delta t)\Delta t + \mathbf{M}\mathbf{V}(t) + \Delta t\mathbf{F}_{external}), \quad (73)$$

and

$$\mathbf{X}(t + \Delta t) = \mathbf{X}(t) + \Delta t\mathbf{T}\mathbf{V}(t + \Delta t). \quad (74)$$

The updated parameters in point 5 are the same as the ones in point 1. They are now updated using the new generalized velocity and position, $\mathbf{X}(t + \Delta t)$ and $\mathbf{V}(t + \Delta t)$. The movement of the trap in point 6 is straightforward, just check if the Pilus is in its unfolding or retraction phase, and then change the position accordingly.

When the time step is completed the force between the trap and the Pilus needs to be saved together with the length of the Pilus. This force can be extracted from the following

$$\mathbf{F}_{constraint} = \mathbf{G}^T\lambda, \quad (75)$$

which is equation (37). The length of the Pilus is simply calculated by taking the difference between the last and the first block in the chain.

The stop condition is of the form

- IF $L_p \geq N \cdot L_t$: reverse trap velocity,
- IF trap velocity is reversed AND trap is at original position: stop simulation,

where L_p is the length of the Pilus, L_t is a parameter given by the user and N is the number of blocks. There is also an option not to reverse the simulation but to end it if the first condition is true.

3.6.2 Special solutions

Many of the matrices used in the simulation are huge and has a large number of elements set to zero. These were implemented as sparse matrices. These matrices are shown in table 2. This has the benefit of only needing to allocate storage space for the elements which are actually non zero and also gives MATLAB a hint to which the non zero elements are when solving the linear equation systems.

Table 2: The matrices implemented as sparse matrices.

\mathbf{M}	\mathbf{M}^{-1}	\mathbf{M}
\mathbf{I}_l	\mathbf{I}_l^{-1}	\mathbf{T}
\mathbf{G}	\mathbf{G}^T	ϵ

It was also found to be rather inefficient implementing certain calculations using the scripted language of MATLAB. Instead, some parts implementation were written in C and compiled. Both step 3 and 5 in the main algorithm (section 3.6.1) were implemented in this way.

3.6.3 Computational complexity

The time complexity of the simulation can generally be written

$$F(N) = f_1(N) + N_t \cdot f_2(N), \quad (76)$$

where N is the number of blocks, N_t is the number of time steps the simulation runs and $f_1(N)$, $f_2(N)$ are some functions.

The function f_1 corresponds to the initialization of the constant and derived parameters. The time complexity of setting the constants are independent on N , the derived parameters on the other hand need to be initialized for each block. This gives

$$f_1(N) = C_1 + C_2 \cdot N, \quad (77)$$

where C_1 is the time taken to initialize the constants and C_2 is the time taken to initialize the derived parameters for each block.

The function $f_2(N)$ includes the time consumed by all computations done in the loop. The constraints calculation include some constant complexity initializations and some calculations done for each block. These calculations are however independent on the number of blocks so the total time complexity is linear. $f_2(N)$ also includes solving the equations of motion, the solution to those are shown in section 3.6.1. Following are the dimensions of the matrices and vectors used in the calculations:

$$\begin{aligned}
\mathbf{S} &\in \mathbb{R}^{N_C \times N_C}, \\
\mathbf{G} &\in \mathbb{R}^{N_C \times 6N}, \\
\mathbf{M} &\in \mathbb{R}^{6N \times 6N}, \\
\epsilon &\in \mathbb{R}^{N_C \times N_C}, \\
\lambda &\in \mathbb{R}^{N_C}, \\
\mathbf{V} &\in \mathbb{R}^{6N}, \\
\mathbf{F}_{external} &\in \mathbb{R}^{6N}, \\
\Phi &\in \mathbb{R}^{N_C}, \\
\mathbf{X} &\in \mathbb{R}^{7N}, \\
\mathbf{T} &\in \mathbb{R}^{7N \times 6N}.
\end{aligned}$$

There are two especially importing things to say about these dimensions. First, the number of constraints, N_C , depends on the number of blocks, N , linearly. Secondly, all matrices are band diagonal. These two facts together with the dimensions and equations (71) to (74) gives a time complexity depending linearly on the number of blocks. The derived parameters are also updated in each time step, these are the same calculations as those done outside of the loop and does therefore have a linear time complexity. Putting this together gives

$$F(N) = C_1 + C_2 \cdot N + N_t \cdot [(C_3 + C_4 + C_5 + C_6) N + C_7], \quad (78)$$

where C_1 is the time it takes to initialize constants, C_2 to initialize all derived parameters for each block, C_3 to calculate constraints for each block, C_4 to solve equations of motion for each block, C_5 to update derived parameters for each block, C_6 to draw each block and C_7 to save forces, check stop condition and update trap position. The total number of time steps, N_t will also be dependent on N , which can be seen from the stop conditions in section 3.6.1. This gives the whole simulation a time complexity dependent on the square of its size

$$F(N) = O(N^2). \quad (79)$$

The resulting time complexity is also confirmed in section 4.3.

4 Results

The simulation should correspond to the physical experiments where the force on the trap was compared to the elongation distance, see section 1. There should

be three distinct regions called region I, II and III, all due to the change of state of the Pilus. Figure 16 shows the result of a simulation with 320 units of PapA. The force is larger when the Pilus is unfolded than when it is retracted. As can be seen, there is a jump in the response force at about $1\mu m$ during retraction. This is a somewhat expected behavior, see section 5. A sequence of images from the three different regions in the simulation are shown in figure 17, 18 and 19 respectively.

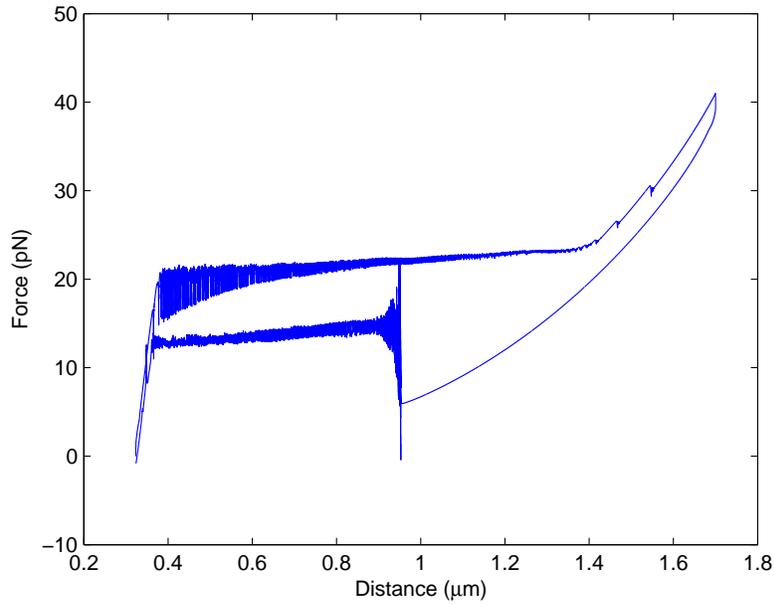


Figure 16: Force on trap as a function of the length of the Pilus.

The remaining of this section is devoted to, region by region, examine how the force/elongation curve depends on simulation parameters and thus how the model compares to the real system under these experiments.

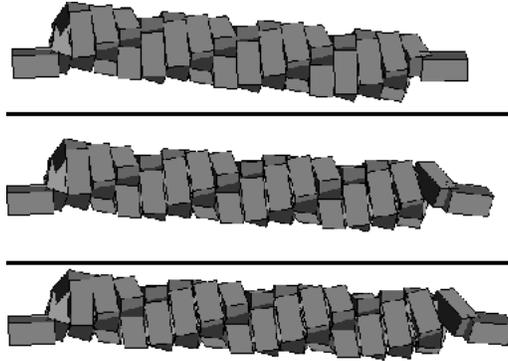


Figure 17: A sequence of images from region I in the simulation. The distance between different layers are small and the dominant force is derived from the layer to layer potential.

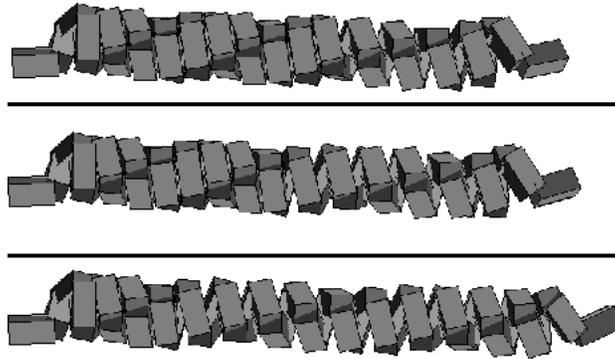


Figure 18: A sequence of images from region II in the simulation. Some of the layer to layer bonds have broken and the Pilus is unfolding. This occurs as a chain reaction from the right to the left.



Figure 19: A sequence of images from region III in the simulation. All the layer to layer bonds have broken. Further elongation now comes from the stretching of the pairwise bindings.

4.1 Model validation - Region I

In region I the dominant interaction is the layer to layer interaction explained in section 3.2. The potential from equation (60) can, for small deformations, be approximated as a spring potential ($U = 0.5k(\Delta L)^2$) using

$$U_{LL} \approx -\frac{U_0}{L_{LL}^2}r^2. \quad (80)$$

Each layer has three bindings and there is a total of $\frac{N}{3.28}$ layers so the effective spring constant for the entire Pilus⁹ should be

$$k_e = \frac{3.28^2 \cdot 2U_0}{NL_{LL}^2}, \quad (81)$$

as can be seen from the calculations in section 7.2. This relation can be tested by plotting the force applied by the Pilus on the trap as a function of the displacement. These results are shown in table 3. The data is also presented in figure 20 with experimental and theoretical spring coefficient as a function of the number of blocks.

The effective spring coefficient should also depend on the potential depth and characteristic length. Measurements have been done to determine if this relation is correct. The result given while varying the potential depth is shown in figure 21. Here the strength is normalized against the mainly used value for U_0 given in section 3.5. The values given in the simulations are compared to theoretical values given by equation (81). As can be seen, there seem to be some constant factor that isn't in equation (81).

In a similar way, results of measurements while varying the characteristic potential length are shown in figure 22. Again, an unknown constant factor appears.

The length of region I should be dependent on the number of blocks in the pili in the following way

$$R_I = f_I N L_{LL}, \quad (82)$$

where R_I is the region length and f_I is an unknown factor. The region I length is shown as a function of N in table 4 and figure 23. This length should also

⁹If the whole Pilus is considered to be a spring, this would be the spring constant.

Table 3: Effective spring constant in region I shown in simulation units.

N	U_0	L_{LL}	k_e (simulation)	k_e (theoretical)
50	0.005	0.75	$2.388 \cdot 10^{-3} \pm 3 \cdot 10^{-6}$	$3.825 \cdot 10^{-3}$
75	0.005	0.75	$1.782 \cdot 10^{-3} \pm 2 \cdot 10^{-6}$	$2.550 \cdot 10^{-3}$
100	0.005	0.75	$1.4196 \cdot 10^{-3} \pm 9 \cdot 10^{-7}$	$1.9126 \cdot 10^{-3}$
150	0.005	0.75	$9.987 \cdot 10^{-4} \pm 5 \cdot 10^{-7}$	$1.275 \cdot 10^{-3}$
200	0.005	0.75	$7.17 \cdot 10^{-4} \pm 1 \cdot 10^{-6}$	$9.56 \cdot 10^{-4}$
320	0.005	0.75	$4.699 \cdot 10^{-4} \pm 4 \cdot 10^{-7}$	$5.467 \cdot 10^{-4}$

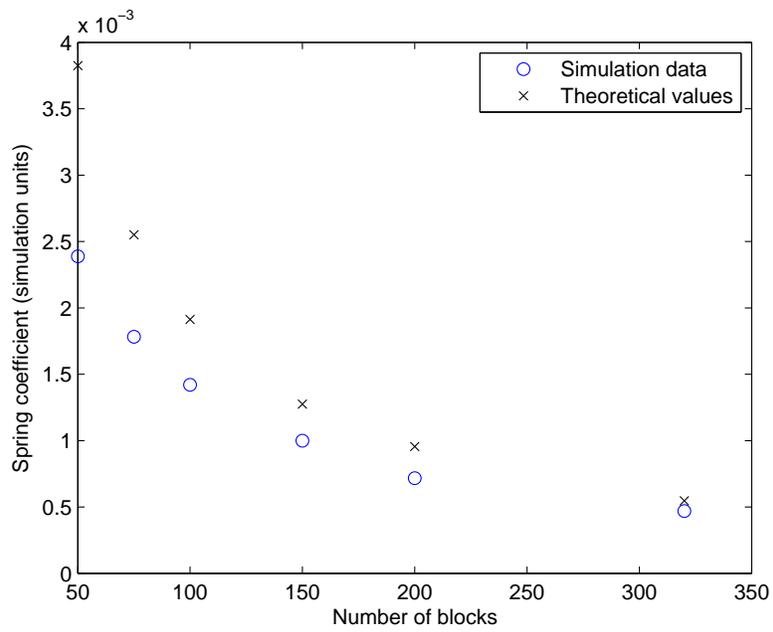


Figure 20: Spring coefficient as a function of number of blocks in region I.

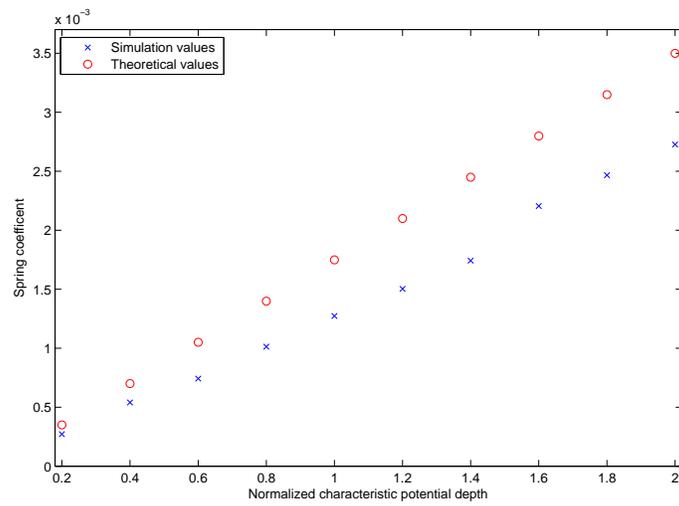


Figure 21: Slope of region I as a function of the normalized characteristic layer to layer potential depth for 100 blocks.

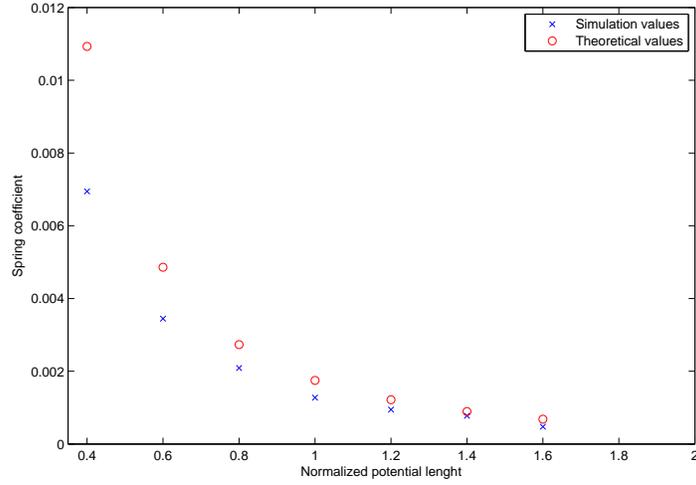


Figure 22: Slope of region I as a function of the normalized characteristic layer to layer potential length.

depend on the layer to layer potential depth and characteristic length, as those parameters determine when the layer to layer bonds begin to break. There is also a dependence on the pairwise angular potentials. If these were made stronger they would effect the simulation even in region I. From these measurements it can be seen that the value of the unknown parameter is

$$f_I = 0.080 \pm 0.007. \quad (83)$$

Table 4: Length of region I as a function of the number of blocks.

N	U_0	L_{LL}	R_I (simulation)
50	0.005	0.75	3 ± 2
75	0.005	0.75	4 ± 2
100	0.005	0.75	7 ± 2
150	0.005	0.75	7 ± 2
200	0.005	0.75	13 ± 2
320	0.005	0.75	19 ± 2

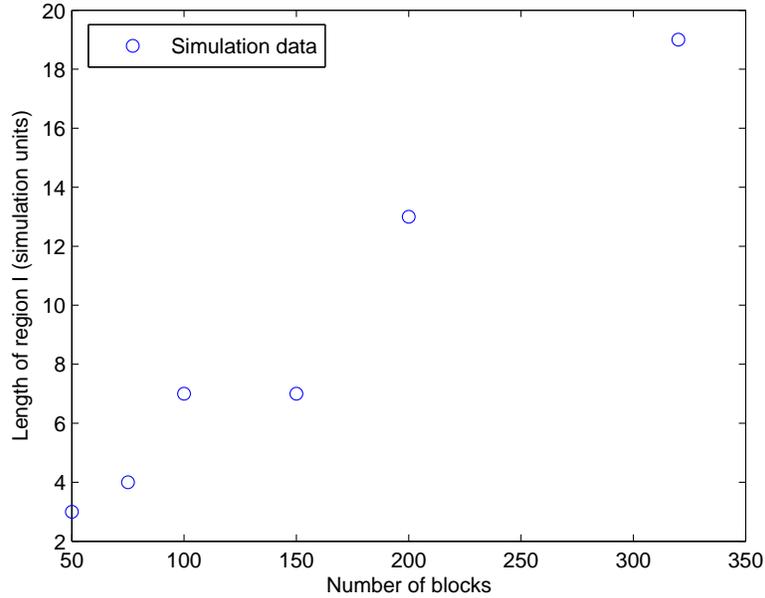


Figure 23: Length of region I as a function of the number of blocks.

4.2 Model validation - Region II

When the Pilus has reached a certain length, which is determined in section 4.1, the layer to layer bonds will begin to break. At this point the force needed to extend the Pilus further should remain constant until region III is reached where all bonds are broken. There is however a small slope even in region II. I have examined its dependence on the number of PapA blocks in the pili and the resulting slope versus number of blocks is shown in table 5 and figure 24.

The end of region II will occur after all layer to layer bonds are broken, that

Table 5: Slope of region II as a function of the number of blocks shown in simulation units.

N	Slope
50	$1.29 \cdot 10^{-5} \pm 3 \cdot 10^{-7}$
75	$7.61 \cdot 10^{-6} \pm 1 \cdot 10^{-7}$
100	$6.17 \cdot 10^{-6} \pm 7 \cdot 10^{-8}$
150	$5.46 \cdot 10^{-6} \pm 3 \cdot 10^{-8}$
200	$5.31 \cdot 10^{-6} \pm 2 \cdot 10^{-8}$
320	$4.593 \cdot 10^{-6} \pm 4 \cdot 10^{-9}$

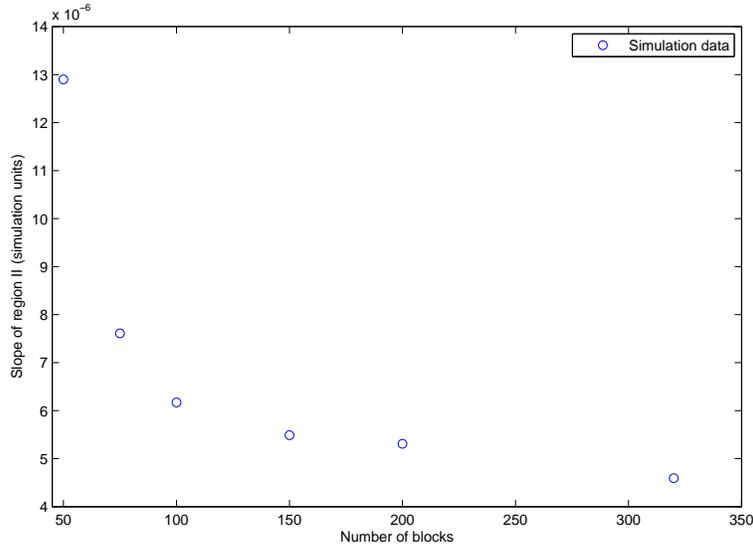


Figure 24: Slope of region II as a function of the number of PapA blocks.

is at length

$$R_{II} = f_{II}NL_{LL}, \quad (84)$$

where R_{II} is the length of region II, N is the number of blocks, L_{LL} is the characteristic length of the layer to layer potential and f_{II} is a constant parameter. The length R_{II} is shown as a function of N in table 6. The measured values gives

$$f_{II} = 1.431 \pm 0.008.$$

See also figure 25 for R_{II} plotted against N .

Table 6: Length of region II as a function of number of blocks shown in simulation units.

N	R_{II}
50	42 ± 5
75	70 ± 5
100	96 ± 5
150	150 ± 5
200	201 ± 5
320	333 ± 5

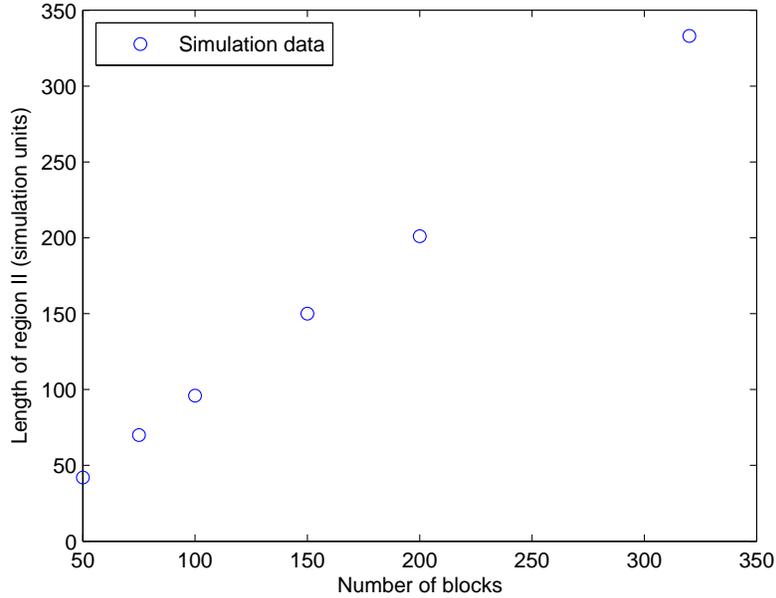


Figure 25: Length of region II as a function of the number of PapA blocks.

4.3 Computational complexity

The time complexity was in section 3.6.3 found to depend on the square of the number of blocks, it can be written

$$F(N) = C_1 N^2 + C_2 N + C_3. \quad (85)$$

To test this, a number of timed simulations have been done, the results are shown in table 7. From these results the constants in equation (85) could be assessed:

$$C_1 = 0.40 \pm 0.04s, \quad (86)$$

$$C_2 = -13 \pm 9s, \quad (87)$$

$$C_3 = 400 \pm 500s, \quad (88)$$

$$(89)$$

As can be seen, only C_1 could be determined with acceptable accuracy using these limited measurements. The measured values are shown together with the plotted function in figure 26.

Table 7: Time to run a simulation as a function of the number of blocks.

N	Time (s)
50	779.5
75	1708
100	3085
125	4838
150	7842
175	10430
200	13880

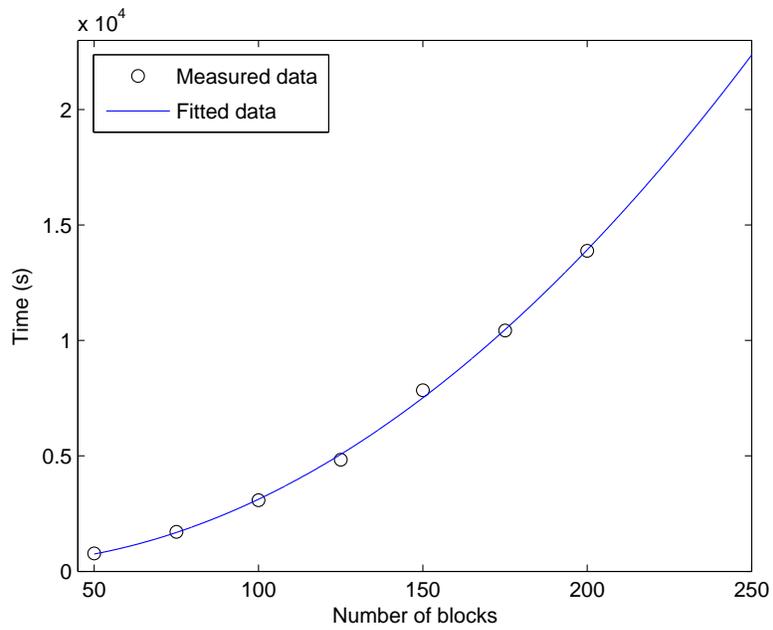


Figure 26: Time to run a simulation as a function of the number of blocks.

5 Discussion

As can be seen in figure 16, when entering region II during refolding of the Pilus at about $1\mu m$, there is a jump in the response force. There is a similar phenomenon occurring in the experiments, but it is much more pronounced in the simulation. In the simulation, the jump occurs at about 50% of the region II length¹⁰, this is in contrast with the experimental values. A possible reason for this anomaly is the absence of random thermal motion in the simulation. The model used is purely mechanical, and because of this a major component of the folding of the Pilus is missing. If a thermal factor was added, there would be an added chance that two layers would get inside the layer-to-layer potential range earlier during the retraction phase. This would start the chain reaction that brings the force response curve into region II. Other than fundamentally changing the computer model, the jump is effected by parameter values. For example, increasing the angular pairwise potential strength will make the jump occur earlier during the retraction. This is because, with a stronger angular potential, the Pilus will not unfold to the same length while in region II. Other parameters that might effect this is, of course, characteristic length and depth of the layer-to-layer potential.

Another problem with the simulation is the disturbances in the force response curve, especially in region II. Each time a layer in the Pilus breaks, there is a sharp drop in the force. It may be possible to get a smoother force response curve by taking the average of the force over time, or sampling the result at an interval. There are also some parameters that effects this; mainly the pairwise potential strengths and the friction coefficients, although adjusting these would give other side effects discussed later in this section. It also seems to be the case that an increased number of blocks decreases the amplitude of the disturbance.

The force response is clearly lower during retraction than during the unfolding. This means that there must be some sort of energy dissipation during the process. This is not very surprising as there is friction in the system, regulated by two friction coefficients. However, it is not certain that the explicit friction forces can account for the whole dissipation. More simulations with varied friction coefficients would be needed to determine how large the share of the energy loss is caused by friction. To make the simulation more in line with the experiments, reducing or removing these coefficients could be a way. But that would also increase the amplitude of various disturbances, and may even cause the simulation to fail because friction is used as a stabilizer. The velocity of the trap could also have an effect on this dissipation, with higher speed giving more energy loss. Further simulations with varied velocities should be done to determine by how much this effects the energy dissipation.

If the Pilus is stretched beyond a certain length the simulation will fail. The blocks will violently realign in a configuration different from the one originally intended. The effects of this is that the Pilus will be unable to retract correctly, which can be seen in figure 27. What is shown is that the force response never

¹⁰Note that the height of the jump is directly linked to its length. By reducing the length, the height is also reduced.

makes the jump back to region II during retraction. Also look at figure 28, which shows the simulated Pilus after a failed simulation. The cause for this failure is, probably, the way the angular constraints are implemented. At the beginning of the simulation the angles are in their rest alignment, shown in figure 29. It should be especially noted that the angles between axis \mathbf{c}^1 and \mathbf{c}^2 are, approximately, zero. When the simulation has gotten a bit into region III, one of the three axes will have drifted from the equilibrium angle. This is shown in figure 30. From this figure it is clear that the axes \mathbf{c}^1 and \mathbf{c}^2 have an angle far from zero between each other, and the restoring force should be close to its maximum. What now happens is that one of the blocks makes a turn around the \mathbf{a} axis and causing a chain reaction in the structure. This puts the Pilus in a local equilibrium state, unable to retract properly. To stop this from happening, changes of the pairwise potentials in region III might be needed. In a real experiment the Pilus would detach if a too large force is applied but this might not be wanted in a simulation because that would, as with the current model, stop it from completing. A better way would be to increase all potentials at a certain elongation making it impossible to go any further, and then reversing the trap. No further simulations have been done to achieve this, but some change in the model is probably needed.

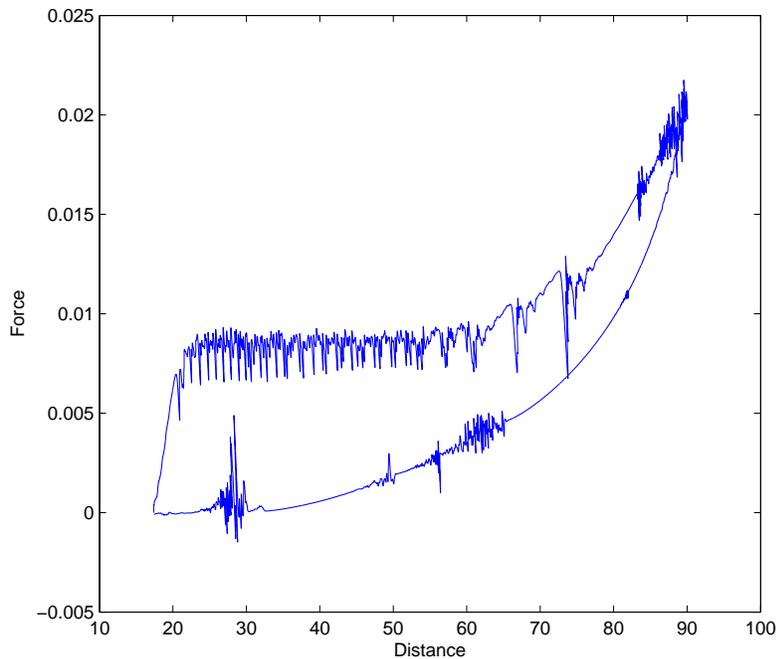


Figure 27: The force response curve of a failed simulation.

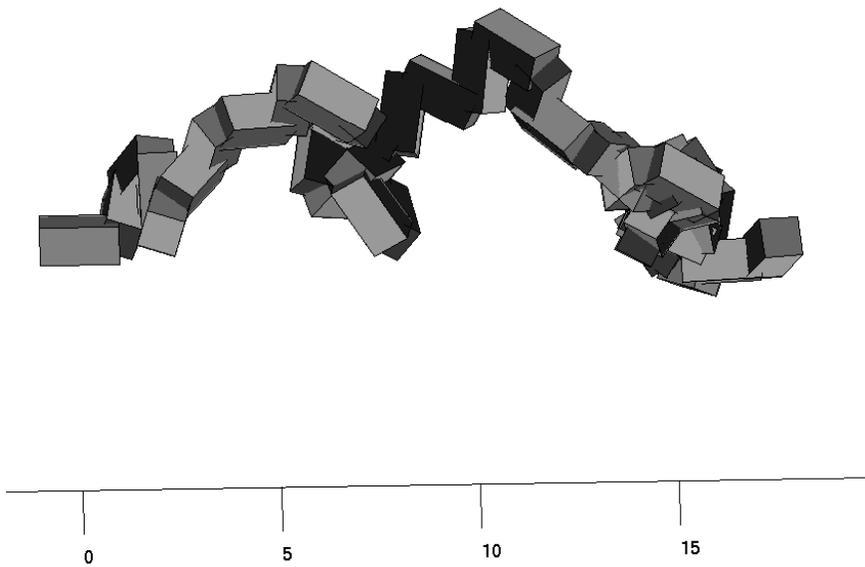


Figure 28: The Pilus refolding erroneous after a failed simulation.

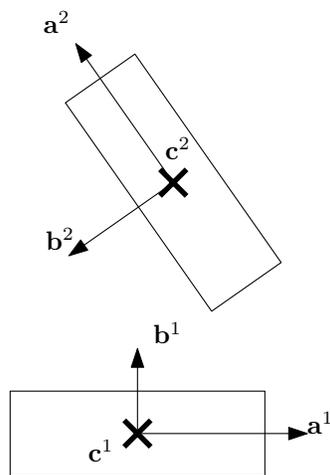


Figure 29: Two PapA blocks in equilibrium state. Seen from the trap looking at the Pilus, the blocks have been separated to easier see the angles between the local axes.

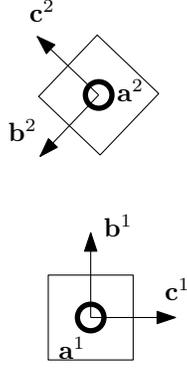


Figure 30: Two PapA blocks in maximal unfolded state. Seen from the trap looking at the Pilus, the blocks have been separated to easier see the angles between the local axes.

5.1 The third region of elongation

Region III is the part of the simulation that corresponds the least with experimental results. The behavior of the force response in region III have not been taken into account when the computer model was designed. This means that any results achieved in this region is an extrapolation using models and parameters designed for the two previous regions.

In the simulation, the force seems to increase exponentially with increasing elongation in region III. This is in contrast to the experimental results, where the force increases, then levels off and then increases again. This could be possible to simulate in a number of different ways. Possibly the best method would be to reformulate the angular constraint functions, e.g., make them depend on the square or cube of the angles in a way that does not modify regions I and II but still gives the wanted result in region III. Another method might be to make the angular potential strengths¹¹ dependent on the corresponding angle. In this way the simulation could be kept identical to the current behavior in region I and II but reflect the experiments in region III. It should be noted that none of these two methods have been tested so no conclusion can be made regarding which method, if any, is the more appropriate. What can be said is that the introduction of such a solution would require a great deal of tuning the parameters to get acceptable result.

5.2 Computational complexity and simplifications

The computational complexity of the simulation have been shown in sections 3.6.3 and 4.3 to be dependent on the square of the number of blocks. Extrapolating using the formula in 4.3, it can be seen that simulating a whole pili with 1000 PapA blocks would take 100 ± 13 hours or four to five days. This would be

¹¹That is ϵ_1 , ϵ_2 and ϵ_3 .

on a low end personal computer by today’s standard and can be cut by a large constant factor using a more powerful computer. It would not be too far fetched to assume the simulation would finish in less than a day if using a modern high end computer.

There is an assumption made in this model to decrease the time complexity: the layer to layer force is only calculated between adjacent layers. The potential chosen looks like this

$$U_{LL} = -U_0 e^{-\frac{r^2}{L_{LL}^2}}, \quad (90)$$

and the absolute value of the force resulting from this potential becomes

$$F_{LL} = |-\nabla U_{LL}| = 2 \frac{U_0}{L_{LL}^2} r e^{-\frac{r^2}{L_{LL}^2}}. \quad (91)$$

A valid question is if there would be any differences in the final simulations if the force wasn’t truncated. That is, if there could be interactions between non adjacent layers if this potential was used. This force is shown in figure 31. As can be seen, the force is at about half of its maximum strength at a distance equal to the width of a PapA block¹². If the truncation of the force were to be fully removed, the helical structure of the Pilus would probably implode at the beginning of a simulation with the current parameters. Of course, the characteristic length of the layer to layer potential could be decreased to give better initial stability. As an example, see figure 32 for the same force with L_{LL} set to 0.35 simulation units¹³. With this modified potential, the jump in response force would however occur even later during retraction. Another thing to consider is the efficiency if the range restriction were removed. The time to calculate layer to layer forces between each block increases with the square of the number of blocks, which means that the total time complexity of a full simulation would increase to $F(N) = O(N^3)$ instead of $F(N) = O(N^2)$. This could be managed by dividing space into a grid of cubes with a suitable side length, and then calculate the force between each block inside the same and every adjacent grid cubes. Using the current model with $L_{LL} = 0.35$, a good side length would be 0.65 at which distance the force is 0.01 times its maximal strength.

In an experiment there are often several pili attached to the small bead at once. If a simulation was done with 100 pili using the current model with minimal modifications, it would take 100 times as long to finish, give or take some overhead. This is assuming there would be no interaction between pili. On a personal computer, this is about 40 to 45 days. If the layer to layer potential was changed, as mentioned above, to allow for such an interaction, the time could increase considerably. Assuming the layer to layer force has to be calculated between each block, the simulation would take 100^2 times as long, or a bit over 100 years. This is not a practical time frame, a grid division as described above would have to be done.

¹²Which is one simulation unit.

¹³Instead of the standard value 0.75 used in the simulation, see section 3.5.

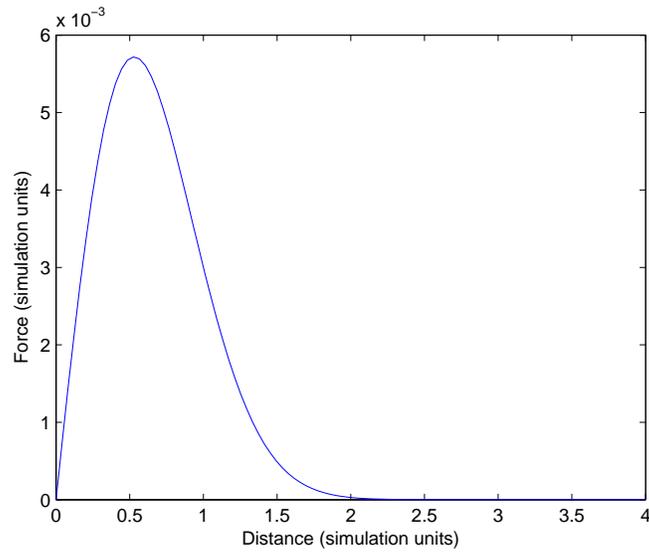


Figure 31: The absolute value of the layer to layer force as a function of the distance between red and blue points.

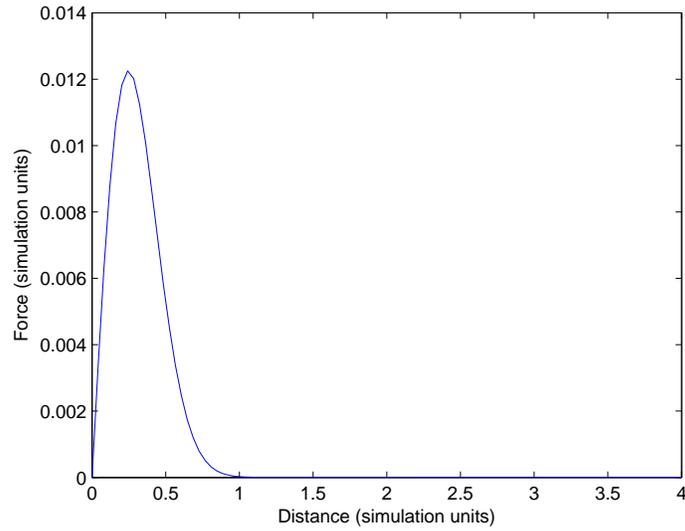


Figure 32: The absolute value of the layer to layer force as a function of the distance between red and blue points. The potential characteristic length has been set to 0.35.

5.3 Parameters and further simulations

Many more simulations could be done while varying the parameters to give results more in line with the experimental ones. There are twelve important parameters given in section 3.5, ignoring the dimensions of the blocks and using $N = 1000$. If ten different values were to be tested for each of these parameters, 10^{12} simulations would have to be done. Using the generous estimate that each simulation takes ten hours to complete, all tests would take a total of 10^{13} hours or 1.2 billion years divided by each computer used. This is not realistic, but perhaps not all combinations of parameters have to be tested. If the parameters were tested one at a time, only $10 \cdot 12$ or 120 simulations would have to be done. This would take approximately 50 days on a single computer.

6 Conclusion

The objective of the thesis was to develop a computer program suitable to simulate experiments done on the pili of E. Coli bacteria. A model usable for the two first regions of elongation were developed and tested. The model parameters were adjusted to get results in line with those given by experiments. The final results were in line with the objectives on the following points

1. The force response curve in the simulation during elongation follow that of the experiment well in region I and region II.
2. The force response during retraction has a jump between region III and II just as in the experiments.
3. It is easy to adjust parameter values.
4. The simulation is somewhat time efficient.
5. The simulation is presented graphically.

There are however some points at which the simulation is lacking, those are given in the following table:

1. The jump in force response during retraction is too large and occurs too late.
2. There are disturbances in the force response in region II.
3. Region III is not modeled at all, only extrapolated using a model designed for region I and region II.
4. If the Pilus is stretched too far, the simulation will fail in a way making retraction impossible.

At least point 2 and possibly also point 1 are repairable without changing the current model. The other points would probably require more fundamental changes. It should also be noted that point 3 is connected with point 4, it might be possible that the solution of the former would lead to the solution of the latter.

7 Appendix

7.1 Numerical method

Equations (38) to (40) is solved using a semi implicit Euler algorithm. In the following derivation, subscript denotes time:

$$A_{n+1} \equiv A(n\Delta t + \Delta t),$$

where Δt is the length of the time-step used.

First, the positions are expanded in time¹⁴

$$\mathbf{X}_{n+1} = \mathbf{X}_n + \Delta t \frac{\partial \mathbf{X}}{\partial t} + O(\Delta t^2) \approx \mathbf{X}_n + \Delta t \mathbf{T}_n \mathbf{V}_{n+1}. \quad (92)$$

Then the same is done to the velocities

$$\mathbf{V}_{n+1} = \mathbf{V}_n + \Delta t \frac{\partial \mathbf{V}}{\partial t} + O(\Delta t^2) \approx \mathbf{V}_n + \Delta t \mathbf{M}_n^{-1} (\mathbf{G}_n^T \lambda_{n+1} + \mathbf{F}_n), \quad (93)$$

where \mathbf{F} is the force from equation (17). Finally, λ is also expanded in time

$$\lambda_{n+1} = \lambda_n + \Delta t \frac{\partial \lambda}{\partial t} + O(\Delta t^2) \approx \lambda_n - \epsilon^{-1} (\Phi_n + \Delta t \mathbf{G}_n \mathbf{V}_{n+1}). \quad (94)$$

7.2 Spring coefficient approximation

The layer to layer potential is given by

$$U_{LL} = -U_0 e^{-\frac{r^2}{L_{LL}^2}}. \quad (95)$$

By Taylor expansion one gets

$$U_{LL} = C - U_0 \frac{r^2}{L_{LL}^2} + O(r^4) \approx C - U_0 \frac{r^2}{L_{LL}^2}, \quad (96)$$

where C is a constant. The constant can be ignored since it will not be used when calculating the resulting force. What is left is a spring potential with spring coefficient $\frac{2U_0}{L_{LL}^2}$, corresponding to a single binding. Each layer in the Pilus has 3.28 bindings, so the spring coefficient for each layer should be 3.28 times as large. Then there are also $\frac{N}{3.28}$ layers in the Pilus, so if the whole structure is elongated by R , each layer is separated by $\frac{3.28}{N}R$. This gives an effective spring coefficient

$$k_e = \frac{3.28^2 \cdot 2U_0}{NL_{LL}^2}. \quad (97)$$

¹⁴This expansion could be done in several ways.

References

- [1] Jass, J., Schedin, S., Fällman, E., Ohlsson, J., Nilsson, U., Uhlin, B., AND Axner, O. 2004 *Physical Properties of Escherichia coli P Pili Measured by Optical Tweezers*. Biophysical Journal 87:4271-4283.
- [2] Andersson, M., Fällman, E., Uhlin, B-E., AND Axner, O. 2006 *Dynamic Force Spectroscopy of E. coli P Pili*. Biophysical Journal 91:2717-2725.
- [3] Erleben, K., Sporring, J., Henriksen, K., AND Dohlmann, H. 2005 *Physics-Based Animation*. Charles River Media.
- [4] Fällman, E., Schedin, S., Jass, J., Andersson, M., Uhlin, B-E., AND Axner, O. 2004 *Optical tweezers based force measurement system for quantitating binding interactions: system design and application for the study of bacterial adhesion*. Biosensors and Bioelectronics 19:1429-1437
- [5] Andersson, M., Fällman, E., Uhlin, B-E., AND Axner, O. 2006 *A Sticky Chain Model of the Elongation and Unfolding of Escherichia coli P Pili under Stress* Biophysical Journal 90:1521-1534
- [6] Fällman, E., Schedin, S., Jass, J., Uhlin, B-E., AND Axner, O. 2004 *The unfolding of the P pili quaternary structure by stretching is reversible, not plastic* EMBO Rep. 6:52-56.
- [7] Servin, M., Lacoursière, C. 2008 *Rigid Body Cable for Virtual Enviroments* Accepted for publication In IEEE Transactions on Visualizations & Computer Graphics
- [8] Allen, M. P., 2004 *Introduction to Molecular Dynamics Simulation* NIC Series, Vol. 23, pp. 1-28
- [9] Lacoursière, C., 2007 *Ghosts and Machines: Regularized Variational Methods for Interactive Simulation of Multibodies with Dry Frictional Contacts* PhD Thesis, Department of Computing Science, Umeå University, Sweden