NTNU
Norwegian University of
Science and Technology

The Heterogeneous Multiscale Method and the Spinning Top

Maren Fredbo

Master of Science in Physics and Mathematics
Submission date: June 2011
Supervisor: Brynjulf Owren, MATH

Norwegian University of Science and Technology
Department of Mathematical Sciences

Problem description

The heterogenous multiscale method is an efficient method for stiff ordinary differential equations. It reduces the computational complexity and is applicable to many types of multiscale problems. We wanted to study the properties of this method by applying it on a physical test problem, the spinning top.

Assignment given: 10. Januar 2011
Supervisor: Brynjulf Owren

Preface

This thesis was carried out at the Department of Mathematical Sciences and Technology at the Norwegian University of Science and Technology (NTNU) during the period Januar 2010 to June 2011.

This is an extension of my semester project on the heterogenous multiscale method. The subject of this thesis was motivated by my supervisor Brynjulf Owren and myself, as we both wished to learn more about this method. The motivation of studying the heterogeneous multiscale method is it's wide applicability in many fields of sciences, and the fact that this is a newly developed framework for multiscale problems.

I would like to thank my supervisor Brynjulf Owren for excellent guidance and for inspiring meetings. I would also like to thank Richard Tsai for his thoughts on the spinning top problem, which led to a great progress in the given problem.

Trondheim, June 2011

Maren Fredbo

Abstract

The heterogeneous multiscale method (HMM) was proposed by E and Engquist in [1] and is considered to be an efficient method for problems with multiple time scales. We give a short introduction to the HMM for multiscale problems in general, before we restrict our work to HMM schemes developed for stiff ODEs, based on results found by Engquist et al. [2, 3]. HMM provides an efficient and systematic way to move between the macroscopic and microscopic model in a problem having multiscale physics. By taking advantage of scale separation in multiscale problems, the HMM approximates the macroscopic variables of the solution without fully resolving the microscopic solution. This introduces computational savings as the total number of evaluations needed for convergence are significantly reduced.

We test the features of the HMM on the spinning top. The governing equations of the top produces a highly oscillatory solution as the top spins fast. Despite this fast oscillating nature, we would intuitively expect some slow behavior of the top, for instance the inclination from the vertical axis or the circulation of the top around the vertical axis. We find a set of slow variables of the spinning top, and show that the HMM provides an accurate solution of the macroscopic variables of the top, with a significant gain in computational cost compared to standard solvers.

We also study the spinning top subjected to a vibrational external force and find a set of slow variables, which can be approximated accurately with HMM. Finally, we find an averaged equation to the spinning top subjected to a vertical vibrating force. This analysis is based on the Modulated Fourier expansion and inspired by the work of Sanz-Serna in [4].

The work of this thesis is an extension of the semester project [5], and we emphasize that the theory part of this thesis is partially from this work.

Contents

1 Introduction

Computer simulations and mathematical modeling have increasingly grown to be an important and efficient tool in science and engineering. While the demand for more accurate solutions is growing, the complexity of problems is increasing simultaneously, and leads to even more complex mathematical modeling. An example is new technologies involving creation of structures at the scales of atoms and molecules and require a detailed insight of physics at small scales. Such new sciences are highly dependent on computational modeling.

However, a detailed model of the small scales, here referred to as the microscales, is seldom enough. Very often it is the physics at the larger scales, or the macroscales, which is of interest. Problems containing multiple scales are considered to have *multiphysics* and require multiscale modeling considering both the behavior at microscales and at macroscales, as the macroscopic behavior is strongly affected by the properties of the microscales.

In physics and chemistry we find that almost all problems have multiple scales. An example is the dynamics of a fluid, which are described by different physical laws at each scale: Quantum mechanics models the behavior at the smallest scale, molecular dynamics and kinetic theory describe the processes of the fluid at nano- and micrometer scales, while continuum theory is applied at the largest scale. The different theories at different scales are a general trend in many areas of sciences and lead to an urgent need for multiscale modeling techniques for development of new technologies.

The traditional approach of solving systems at different scales is to construct analytical or empirical explicit equations providing the information we need at one scale. In this way, the other scales are eliminated. Examples of well known analytical techniques are Fourier analysis, the averaging method and the homogenization method. Other mathematical techniques have been developed in the recent years, but they all seem to be limited when it comes to practical use. Thus, empirical techniques have been important when describing models at small scales. An example of this is the use of the viscosity-parameter in the Navier-Stokes equation, which provides an empirical description of the molecular details of the fluid. Another example is the empirical potentials that model the forces between atoms in molecular dynamics. Such empirical approximations of small-scale behavior are used everywhere in science and allow us to use a simplified theory.

However, these methods lack information on how the microscale structures affect the system at macroscale level. This has motivated the development of several numerical methods in the last decade, most of them providing a detailed solution of fine scale problems. Such traditional multiscale methods are the Multigrid method, domain decomposition and the multiresolution methods. As these methods provide a full representation of the fine scale problem, they give an overwhelming computational cost when it comes to practical use, and more efficient methods are strongly requested.

This has led to a new class of multiscale methods that are developed to reduce the computational complexity. Specifically, the new methods are designed to satisfy a *minimum requirement*:

$$\frac{\text{cost of multiscale method}}{\text{cost of microscale solver}} \ll 1, \tag{1}$$

[6]. A common feature for all these methods is that they take advantage of the separation of scales which occurs in many problems, and in this way become much more efficient than solving the fine scale problem. There are several kinds of multiscale techniques of this class. One example is the Quasi-continuum method [7], which in its simplest version models atomistic systems without explicitly treating every atom in a problem. The basic idea of these multiscale methods is the following: We are given a system which can be described by a microscopic model and a microscopic state variable u. The microscopic model is too costly to solve in full detail, as we are only interested in the behavior at a larger scale, here called the macroscale. The macroscale motions are described by the state variable U and the two variables are connected by the relation $Qu = U$, where Q is a compression operator. The challenging part of these methods is the evaluation of U, which is usually not explicitly defined and must be provided by extracting necessary data from the microscale model.

Methods constructed in this way have grown rapidly in popularity. Since they only consider the microscale model *locally* to extract information for the macroscopic state they fulfill the minimum requirement (1) and are much more efficient than the methods that provide the fine scale model globally. Emerging from all the different approaches of multiscale methods, the need of a systematic technique that can be applied to a wide variety of applications has evolved. A proposal of a rigid method was made in [1], called the heterogeneous multiscale method (abbreviated HMM). This method intends to be a general framework for multiscale methods, relying on existing numerical methods, and providing mathematical theory for stability and accuracy. The method is *heterogeneous*, which essentially means that the problems can exhibit multiphysics and be described by different mathematical models at different scales. This is in contrast to the multigrid method, which uses the same mathematical model at the micro- and macroscales. The heterogeneous multiscale method has turned out to be a useful framework and is used in several applications. We will now introduce the fundamental properties and the structure of this method.

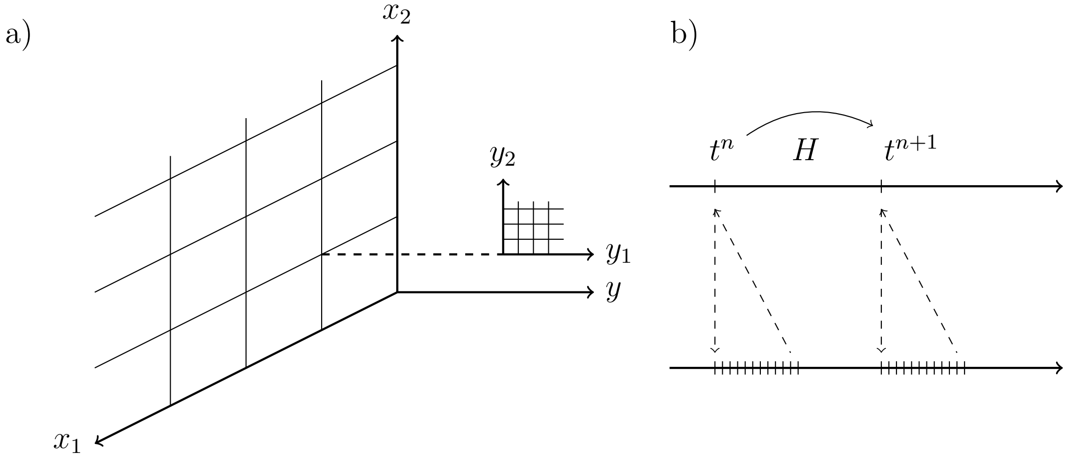


Figure 1: Scheme of the macroscopic and microscopic computational domains. a) Schematics of HMM solver for spatial computational domains, y-space represents the microscopic domain b) Schematics of HMM solver for problems in time, the lower axis represents the microscopic grid with time steps h.

1.1 The HMM Framework

The HMM strategy is based on combining different models at different scales, with the goal of accurately approximating the macroscopic state of the system. This is done by working with a macroscopic grid that resolves the large scale of the problem. The process consists of two main components. The first is a macro-solver solving the scheme for U. The second is a procedure for estimating the missing data from the microscale model. The estimation of the effects from the microscale model has to be restricted such that the estimation is consistent with the macroscale model. This is often the challenging part of the algorithm, and is highly problem dependent. The general algorithm is described in [6] by the following steps at time t^n:

1. Reconstruction: From U^n, construct u^n such that $Qu^n = U^n$. This reconstruction is not unique, and dependent on the problem.

2. Microscale simulation: Evolve the microscale model with initial data $u(x, t^n) = u^n(x)$. The microscale solution should be consistent with the local macroscopic state.

3. Data processing: Extract the macroscale data needed from the microscale data, provided by a microscopic simulation. This usually involves averaging in time/space

Depending on what type of problem we are looking at, we select an appropriate macroscale scheme. When a macroscopic model is not fully known, the missing data is extracted from the microscale model. For example, if the problem is variational, we need to approximate the effective stiffness matrix for the macroscale model. This can be done by solving the microscale problem for a small subdomain of the global computational domain. For dynamic problems we estimate the effective force or the fluxes based on the following procedure: The microscopic state is reconstructed from the macroscale variables at each time step. A microsimulation with microscopic step sizes is done, and provides an estimated force which is used to update the macroscopic state at the next macroscopic time step. Figure 1 illustrates the schematics of these procedures.

As mentioned previously, the key of the HMM is to take advantage of special features in a problem. One such feature is scale separation in time and in this thesis we will study the HMM for stiff ordinary differential equations (ODEs) involving different, well separated, time scales. We will consider dynamical problems where a macroscopic model exist for a set of macroscopic variables. But the macroscale model is not defined explicitly and necessary macroscale data need to be extracted from the microscale model.

In such problems one may only be interested in the slow dynamics of the system, i.e. at the larger time scales of the problem. But stiff ODEs may exhibit an oscillatory nature or fast converging transients, which cannot be neglected because they are affecting the macroscale behavior of the system. HMM applied to these problems has been studied in [2, 3, 8, 4, 9, 10], and is considered to give a significant improvement in terms of computational complexity compared to more traditional methods.

The study of the HMM applied on stiff ODEs will be illustrated by using the example of a simple child's toy: the spinning top. The motions of a spinning top have a highly oscillatory nature, which demands a high computational cost when solving the problem numerically. At a first glance, however, the top seems to have a slowly changing macroscopic behavior. As the top is given a high initial spin, the top's movement are consisting of some fast, and some slow motion. The challenge is to extract the macroscopic behavior from the full model of the spinning top, and still preserve the behavior at the smallest timescales, such as the high spin velocity, which prevents the top from falling down. The HMM applied to such a multiscale problem maintains the interaction between the microscopic and macroscopic motion of the problem and still evaluates the solution of the macroscopic behavior with high computational savings when compared to standard solvers.

The setup of this paper is as follows: In section 2 we will present HMM for stiff ODEs. We apply HMM to the inverted pendulum, which is a common test-problem for HMM [8, 4, 11]. This example will serve to explain the main idea of the method. In section 3 we introduce a multiscale method for highly oscillatory ODEs introduced in [2], which is using the concept of slow variables. In section 3 we introduce the dynamics of the rigid body and the spinning top.

We apply the HMM to the spinning top in section 5 and present the numerical results. At last we consider the spinning top subjected to an external force in section 6 and 7. In section 6 we apply HMM to the spinning top influenced by a large valued, vibrating force. In section 7 we find an averaged equation of the spinning top subjected to a vertical vibrating force by using modulated Fourier expansions, inspired by the work of Sanz-Serna [4]. Conclusions and summary of the thesis are found in section 8.

2 HMM for Stiff Ordinary Differential Equations

The content of this section is a presentation of work of Engquist and Tsai in [3] and [6], where they develop the HMM framework for ordinary differential equations containing different time scales, and is partially from the more extensive summary in [5].

Stiff Ordinary Differential Equations. We define a stiff system as

$$\frac{du}{dt} = f_\omega(u,t), \quad u(0) = u_0, \tag{2}$$

where $u : \mathbb{R}^+ \mapsto \mathbb{R}^d$. Here ω is considered to be a large valued parameter, and characterizes the separation of time scales in the problem. The stiffness of the problem is often related to the eigenvalues $\lambda^{(j)}(t)$, $1 \leq j \leq d$, of the Jacobian of f_ω. Essentially there are two types of stiff problems, both having solutions varying on the ω^{-1} time scale: One is the *dissipative* problems with fast converging transients, having the property $\text{Re}\lambda^{(j)} \ll 0$. The second one is the *oscillatory* problems with rapid oscillations on the ω^{-1} time scale, where $\text{Re}\lambda^{(j)} = 0$.

An example of a stiff system is the nonlinear model system $u(t) = (x(t), y(t))$ described by the equations

$$\begin{aligned}\dot{x} &= \omega f(x,y,t),\\ \dot{y} &= g(x,y,t),\end{aligned}$$

where f and g are smooth functions. In this setting, x is called the fast variable, while y is the slow variable of the system.

As for general multiscale problems, there exist traditional methods for solving stiff systems, both analytical and numerical. An example of analytical techniques is perturbation methods, which approximate an analytical solution to stiff ODEs. These mathematical techniques are often difficult to apply in practice, and require extensive algebraic manipulations for most problems. On the other hand, numerical methods for stiff ODEs suffer from high computational costs when we require accuracy and stability of the numerical approximation. The number of time steps needed for convergence on a larger time scale can be extremely large. Explicit methods for stiff problems require step sizes of order ω^{-1}, which leads to a minimum of $\mathcal{O}(\omega)$ operations. Implicit methods turn out to achieve optimal complexity for the dissipative case, but their shortcomings appear when they are applied to oscillatory problems as they usually need $\mathcal{O}(\omega)$ time steps for convergence.

Stiff systems have motivated the development of several numerical methods aiming to lower the computational cost. A detailed review of numerical methods for stiff problems can be found in [12]. The advantages of the HMM in this

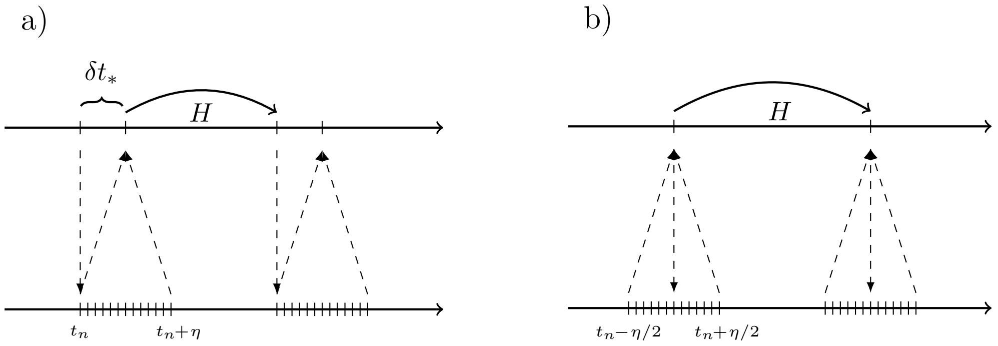


Figure 2: The figure illustrates the time steps taken by the HMM scheme. At the n'th step, (2) is evaluated for a short time η with step size h. This is used to evaluate F in (3), and then take a big step of size H. a) Evaluation of the micro-variables in the interval $[t_n, t_n + \eta]$ and F evaluated at $t_n + \delta t_*$, where $\delta t_* = \eta/2$. b) Microscopic evaluation both backward and forward in time, in $[t_n - \eta/2, t_n + \eta/2]$ and F evaluated at t_n.

setting, are that these methods offer solutions to both oscillatory and dissipative problems. The gain in computational complexity is strongly related to the time duration of the micro evolutions, which usually is much smaller relative to the global computational domain. As long as this time duration is bounded by $C\omega^\alpha$, where $\alpha < 1$, the HMM is guaranteed to gain computational cost [3].

2.1 The HMM Scheme

The general structure of the HMM assumes there exists an "effective" system

$$\frac{dU}{dt} = F(U, t), \tag{3}$$

that is derived from (2) as $\omega \to \infty$ such that the derivatives of F are bounded independently of ω. The right hand side, $F(U, t)$, is not directly used in the algorithm, but calculated via the numerical solution of (2). The algorithm consists of a macro-scheme solving (3) for U, and a micro-scheme solving (2) for evaluating the effective force F. This procedure is illustrated in Figure 2, which shows the relation between the solver at the macroscale grid (upper axis) and the microscale grid (lower axis). The downward arrows represent a reconstruction from the macro-variable U to the micro state $u_n(t_n)$ at the time t_n. Evaluation of $u_n(t)$ is done by solving (2) for a time interval η on the microscale grid. The upward arrows indicate the evaluation of the effective force F from the data processed in the micro-solver. This force is found by averaging the forces $f_\omega(u_n, t)$ with a compactly supported kernel K.

Algorithm 1 HMM solver for stiff ODEs

```
1. Force estimation:
   (a) Reconstruction: at $t_n$, $u_0 = RU^n$.
   (b) Microsimulation: Solve
           $\frac{du_n}{dt} = f_\omega(u_n, t), \quad u_n(t_n) = u_0,$
       for $t \in [t_n, t_n + \eta]$.
   (c) Averaging:
       i. Estimate force:
              $F(t_n + \delta t_*) \sim \bar{f}(t_n + \delta t_*) = K * f_\omega(u_n).$
       ii. Compression:
              $U^* = Q[u_n].$
2. Macrostep: Compute $U^{n+1}$ at $t_{n+1}$ using $\{U^j\}_{j=0}^n$, $\{\bar{f}(t_n)\}_{j=0}^n$ and $U^*$, $F(t_n + \delta t_*)$.
3. Repeat.
```

Two different HMM schemes are depicted in Figure 2, and we notice that F is approximated at the center of the fine scale interval in both schemes. This requires the kernel K to be symmetric. We could also have approximated F at the beginning time of each fine scale calculation by using an asymmetric kernel, which is convenient for dissipative systems. However, a symmetric kernel is preferable as it gives more accurate approximations to the averages [3]. Averaging using kernels will be discussed further in section 2.2.

The complete HMM scheme is described by the steps in Algorithm 1 [3]. Notice that the microsimulation in step 1 (b) is done in the interval $t \in [t_n, t_n + \eta]$, where η denotes the length of the microscopic evaluation. In many problems, such as Hamiltonian systems, one can solve (2) both backward and forward in time in the interval $t \in [t_n - \eta/2, t_n + \eta/2]$, as illustrated in Figure 2 b). In our examples we will only consider Hamiltonian systems and thus approximate the effective force in the interval $[t_n - \eta/2, t_n + \eta/2]$ using a symmetric kernel. The reconstruction and compression operator R and Q in step 1 (a) and 1 (c) are often chosen to be the identity operator, I. In this paper we will also consider the case where $R \neq I$

and $Q \neq I$.

Following the usual notation, we call a method HMM-X-y, where X is the macro-solver in step 2 and y is the micro-solver in step 1 (b). An example is HMM-VE-rk4, which uses a Störmer-Verlet scheme in step 2 and the fourth order Runge-Kutta method as micro-solver. We will now discuss the averaging done in step 1 (c) of Algorithm 1.

2.2 Approximation of Effective Force

We assume that a microsimulation is performed in the interval $t \in [t_n - \eta/2, t_n + \eta/2]$, corresponding to step 1 (b) of the algorithm, and we obtain the micro-variables $u_n(t)$. According to step 1 (c) of the algorithm, these values are used to evaluate the effective force F by averaging over the forces $f_\omega(u_n(t), t)$, for simplicity now referred to as f_ω. This averaging is motivated by analytic averaging techniques [3]: By defining the effective force of the system as

$$F(t) = \lim_{\delta \to 0} \left[\lim_{\omega \to \infty} \frac{1}{\delta} \int_t^{t+\delta} f_\omega(\tau) d\tau \right], \tag{4}$$

we let s be the largest integer such that

$$\left| \frac{d^p}{dt^p} F(t) \right| \leq C \quad \text{for} \quad 0 \leq p \leq s, \tag{5}$$

where C is a constant independent of ω. Then we say that F is slowly varying. The goal is to find an appropriate approximation to the average in (4). Let the kernel K be defined as

$$\int_{-1}^{1} K(t) t^r dt = \begin{cases} 1, & r = 0 \\ 0, & 1 \leq r \leq p, \end{cases} \tag{6}$$

and let K_η denote the scaling of the kernel,

$$K_\eta(t) := \frac{1}{\eta} K\left(\frac{t}{\eta}\right).$$

It can be shown that the convolution $K_\eta * f_\omega$ approximates (4),

$$(K_\eta * f_\omega)(t) = (K_\eta * (F + g_\omega))(t) \longrightarrow F(t) \quad \text{as} \quad \omega \longrightarrow \infty, \tag{7}$$

in the cases where g_ω vanishes exponentially or oscillates. For a detailed discussion and proofs, see [3]. Thus approximation by averaging with kernels applies to stiff problems in (2), and we can estimate the effective force F by the convolution

$$F(t) = (K_\eta * f_\omega)(t) = \int_{t-\eta/2}^{t+\eta/2} K_\eta(s-t) f_\omega(u_n(s), s) ds. \tag{8}$$

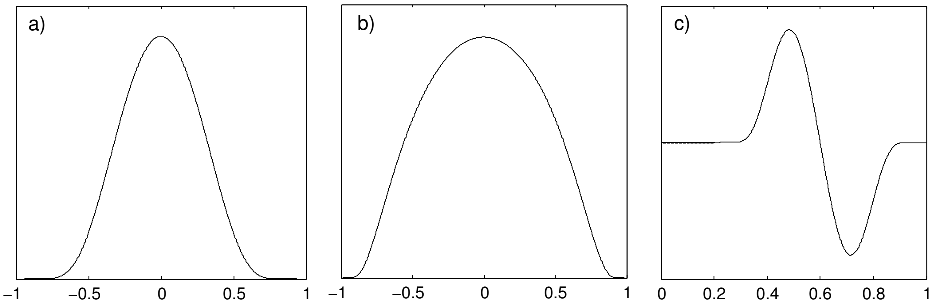


Figure 3: a) The symmetric kernel $K^{\exp(1)}$ with support on $[-1, 1]$. b) The symmetric kernel $K^{\exp(2)}$ with support on $[-1, 1]$. c) An asymmetric kernel with support on $[-1, 0]$ for dissipative systems.

This corresponds to step 1 (c) of Algorithm 1.

For the HMM schemes illustrated in Figure 2, a symmetric kernel should be used to evaluate F in the center of the time interval for each microscale evolution. A typical choice is the exponential kernels

$$K^{\exp(1)}(t) = C_1 \exp\left(\frac{5}{t^2-1}\right) \quad \text{and} \quad K^{\exp(2)}(t) = C_2 \exp\left(\frac{5}{4(t^2-1)}\right), \tag{9}$$

where C_1 and C_2 are the normalization constants. In Figure 3 these kernels are depicted. The figure also illustrates an example of an asymmetric kernel, which is used in the averaging of the force in dissipative systems.

2.3 Accuracy and Stability of HMM for ODEs

The local error of HMM stems from two sources: The error related to the accuracy of the macroscale scheme and the error related to the approximation of the force F. If we denote the error related to approximation of F as ε_{HMM}, we can express the local error as $E_n = \varepsilon_{\text{macro}} + \varepsilon_{\text{HMM}}$. This leads to the following theorem [6]:

Theorem 1. *HMM is stable if the macroscale solver is stable and there exists a constant C such that*

$$|E_n| \leq C(H^{k+1}) + \varepsilon_{HMM} \tag{10}$$

for $n \leq N$, $(N = T/H)$, where k is the order of the macroscale solver.

While the error related to the macro-solver is straightforward to find by using existing error bounds for numerical solvers, the approximation of ε_{HMM} is rather nontrivial. First of all, it consists of both numerical and analytical error. The analytical error is due to the approximation of the force found by (8). The numerical error consists of both the error from the microsimulation in step 1 (b) of Algorithm 1 and the numerical approximation of the force, step 1 (c). In addition, we often have an error related to the reconstruction and compression operations. A detailed review of the error for some basic HMM schemes for both dissipative and oscillatory systems is given in [3].

2.4 Test Example: The Inverted Pendulum

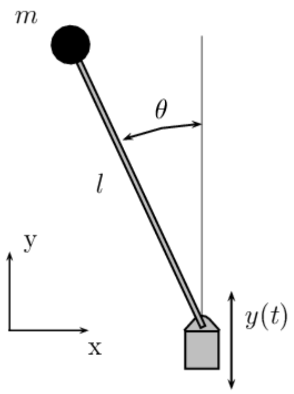


Figure 4: The inverted pendulum attached to a motor vibrating in the y-direction.

The following test problem considers the inverted pendulum depicted in Figure 4. The inverted pendulum is usually unstable as its equilibrium is located at the top of the rod, but the bob can be stabilized in the special case where the pendulum is subject to a vertical force.

For simplicity we consider a light rod of length l with a point mass at the end. The equation of the inverted pendulum subjected to a vertical force is given as

$$l\ddot{q} = (g + a(t)) \sin q, \tag{11}$$

where g, l and q denote the constant of gravity ($g > 0$), the length of the rod and the angle between the upward vertical axis respectively. The force is given by $a(t)$, which is a sinusodial acceleration given by $a(t) = v_{\max}\omega \cos(\omega t)$. Here $a(t) > 0$ when the acceleration points upwards. $v_{\max}$ is the maximum velocity of the pivot, $v(t) = v_{\max} \sin(\omega t)$. The constant $\omega \gg 1$ describes the large valued frequency of scale $\mathcal{O}(\omega)$ and results in a fast vibration of the pivot at microscale, which stabilizes the bob at macroscale. We convert the system (11) into a first order autonomous system of the form

$$\frac{dp}{dt} = l^{-1}(g + a(t)) \sin q, \qquad \frac{dq}{dt} = p. \tag{12}$$

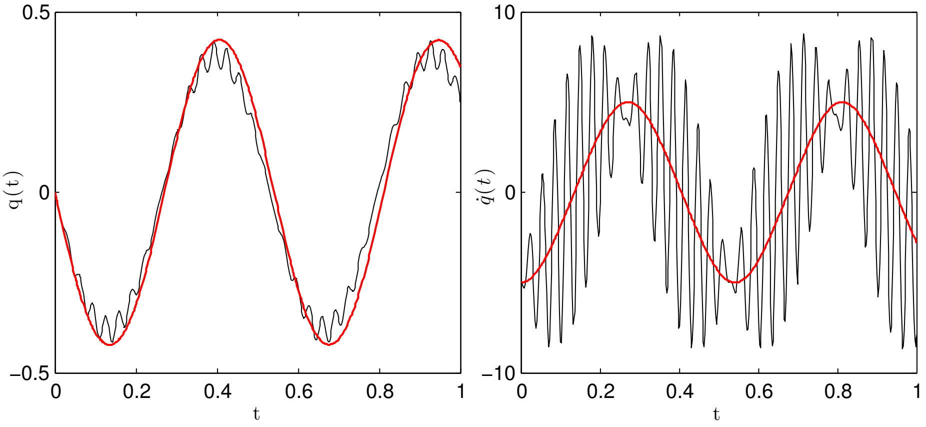


Figure 5: The behavior of q and dq/dt and their corresponding averaged solution Q and P (red line) with $\epsilon^{-1} = 200$.

Figure 5 shows the solution of $p(t)$ and $q(t)$ for the frequency $\omega = 200$. We observe that the fast oscillations with frequency $\mathcal{O}(\omega)$ in the solution of $p(t)$ cause the pendulum to oscillate slowly around the stable equilibrium $q(t) = 0$. Indeed, the angular velocity $q(t)$ exhibits an averaged solution that varies slowly, with small oscillations of angular frequency $\mathcal{O}(\omega)$ and amplitude $\mathcal{O}(\omega^{-1})$.

Averaged Equations. An averaged equation of (11) is found in [4], and is given as

$$\frac{dP}{dt} = \left(\frac{g}{l} - \frac{v_{\max}^2}{2l^2}\cos Q\right)\sin Q, \qquad \frac{dQ}{dt} = P, \tag{13}$$

where Q and P are the averaged solutions of the system in (11). The solution of this system approximates the solution of (12) except from an $\mathcal{O}(\omega^{-1})$ remainder [4]. This system is independent of ω and can be solved accurately with relatively few time steps by any standard numerical solver, in contrast to the original stiff system in (12). This will be used as a reference solution, and we will also use the relation

$$\begin{aligned} p(t) &= P(t) + l^{-1}v_{\max}\sin(\omega t)\sin Q(t) + \mathcal{O}(\omega^{-1}), \\ q(t) &= Q(t) + \mathcal{O}(\omega^{-1}) \end{aligned} \tag{14}$$

which is the relation between the original variables (p, q) and the averaged variables P and Q

The HMM Scheme. The system in (12) will from now be referred to as the

Algorithm 2 HMM Solver for the Inverted Pendulum

```
1. Initial conditions: Given $Q_0 = Q(0)$, $P_0 = \dot{Q}(0)$, $t_0 = 0$, set $n = 0$ and $\hat{P}_0 = P_0$.
2. Force estimation:
    (a) Microevolution:
        i. Reconstruction: Set initial data to
            $$p^n(t_n) = \hat{P}_n + \frac{v_{\max}}{l}\sin(\omega t_n)\sin Q_n, \quad q^n(t_n) = Q_n \qquad (17)$$
        ii. Microintegration: Solve (12) in the interval $t \in [t_n - \eta/2, t_n + \eta/2]$.
    (b) Averaging:
        $$F_n = \frac{2}{\eta}\int_{t_n-\eta/2}^{t_n+\eta/2} K\left(\frac{s-t_n}{\eta/2}\right) f(q^n(s), \omega s; \omega) ds, \qquad (18)$$
3. Macrostep: Evolve the macro-variables $Q_{n+1}$ and $P_{n+1}$.
```

microscale system, in compact form given as

$$\frac{dp}{dt} = f(q, \omega t; \omega), \qquad \frac{dq}{dt} = p. \tag{15}$$

The corresponding macroscale system is given as

$$\frac{dP}{dt} = F(Q), \qquad \frac{dQ}{dt} = P, \tag{16}$$

where Q and P are the solutions of the macroscale system and F is the averaged force from the microscale system acting on the large scale system.

We will now solve this system using the HMM-framework: The macroscale system (16) is solved by numerical integration with step size H, referred to as the *macrointegration*, for $t \in [0, T]$. The effective force F is found numerically using a *microintegrator* with step size h on the original system (15). The macro-solver sampling times are denoted $t_1, t_2, \ldots, t_N$, where $N = T/H$. Numerical solutions of the macro and micro-variables at time t_n are given as P_n, Q_n, p^n and q^n. The complete HMM scheme is described by Algorithm 2 and in what follows we will discuss this algorithm in more detail.

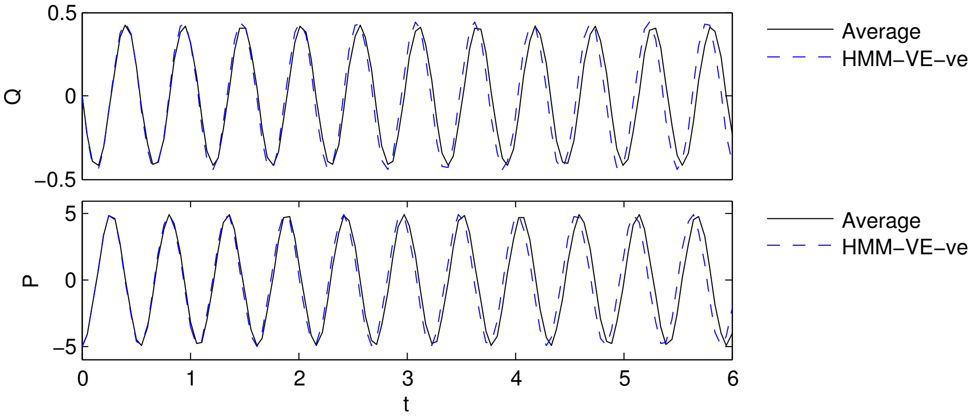


Figure 6: The behavior of Q and P in $t \in [0, 6]$. $H = 1/20$. The solid line shows the solution of the averaged equation (13) and the dotted line shows the HMM approximation of the solution of (11).

The reconstruction in step 2 (a) of the algorithm takes advantage of the modulated Fourier expansions of $p(t)$ and $q(t)$: In order to relate the micro and macro-variables the initial values $(p^n(t_n), q^n(t_n))$ are set equal to the approximation (14), with the predictor $\hat{P}_n = P_{n-1} + HF_{n-1}$. The evaluation of the solution (p^n, q^n) in step 2 (a) is found by using Störmer-Verlet or the fourth order Runge-Kutta method as microintegrator in the interval $[t_n - \eta/2, t_n + \eta/2]$.

In step 2 (b) of the algorithm, F_n is the approximation to $F(Q_n)$. The kernel used in the averaging is the symmetric exponential kernel $K^{\exp(1)}$ defined in (9), where C_1 is chosen such that

$$\int_{-1}^{1} K(t)dt = 1.$$

In practice we can either compute F_n using the trapezoidal rule with step size h, or evaluate the contribution of F_n in each step of the microintegration.

In step 3 of the algorithm we are using Störmer-Verlet as macrointegrator, given as

HMM-VE-*

Given P_n, Q_n, *for* $n = 0,1,2, \ldots$

$$\begin{aligned} P_{n+1/2} &= P_n + \frac{H}{2} F_n \\ Q_{n+1} &= Q_n + HP_{n+1/2} \\ P_{n+1} &= P_{n+1/2} + \frac{H}{2} F_{n+1}. \end{aligned}$$

Numerical Results. We have applied the HMM to the inverted pendulum equations (15) with parameters $l = 0.2\text{m}$, $g = 9.81\text{m/s}^2$, $v_{\max} = 4\text{m/s}$ and $\omega = 10^6$. Figure 6 shows the evolution of the numerical solution (P_n, Q_n) and the solution of the averaged equations (13), $(P(t_n), Q(t_n))$, for $t \in [0, 6]$ using the initial conditions $q(0) = 0$, $p(0) = -5$. Here we used the scheme HMM-VE-ve with $H = 1/20$, $h = 2\pi/(15\omega)$ and $\eta = 40 \cdot 2\pi/\omega$, which corresponds to a microintegration spanned over 40 fast periods.

Complexity. The HMM requires a total of 10^4 operations in the evaluation, compared to a standard solver, which would need a total of 10^6 operations when $\omega = 10^6$. This corresponds to computational savings of 10^2. In fact, we find that the error is almost constant as ω increases (with the same values of H), and leads to computational savings of 10^4 when $\omega = 10^8$.

3 HMM Based on Slow Variables for Highly Oscillatory ODEs

For the inverted pendulum problem we had to relate the micro and macro-variables of the solution by analytical techniques (by the predictor from the averaged equations). Very often the micro-variables, which originates from the original equation (2), do not coincide with the variables that describe the slow motions of the system. This is one of the difficult aspects using HMM for oscillatory ODEs. In [2] they propose a class of HMM algorithms where the need of an analytic relation between the micro and macro-variables is bypassed: The effective slow behavior of highly oscillatory ODEs is found by utilizing a set of slow variables of the system. The next section will summarize some results of [2].

Consider an ODE system of the general form

$$\dot{\mathbf{x}} = \omega f(\mathbf{x}) + g\mathbf{x}, \quad \mathbf{x}(0) = \mathbf{x}_0, \tag{19}$$

where ω is considered to be large and $\mathbf{x} = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d$. The solution for a fixed ϵ and $\mathbf{x}_0$ is $\mathbf{x}(t; \omega, \mathbf{x}_0)$ and for simplicity denoted $\mathbf{x}(t)$. We consider solutions in a bounded domain $\mathcal{D} \subset \mathbb{R}^d$ and in a bounded interval $t \in [0, T]$, where T is independent of ω. As we study the long time properties of (19), by following the slow dynamics of the system, we need to distinguish between fast and slow variables of the system. For the variables in (19), the slow variables are defined as a set of functions whose time derivatives are bounded with a constant independent of ω. In the inverted pendulum problem we could easily identify the slow variables of the system just by looking at the problem and its behavior. The slow variables are in many cases not clearly defined and need to be extracted from a system of fast variables.

3.1 Fast and Slow Dynamics

We say that a real valued smooth function (variable) $\alpha(\mathbf{x})$ is slow with respect to (19) if there exists a nonempty open set $\mathcal{A} \subset \mathbb{R}^d$ such that

$$\max_{\mathbf{x}_0 \in \mathcal{A},\, t \in [0,T]} \left| \frac{d}{dt} \alpha(\mathbf{x}(t; \omega, \mathbf{x}_0)) \right| \leq C_0,$$

where C_0 is a constant independent of ω. Otherwise, $\alpha(\mathbf{x})$ is considered to be fast. An example of slow variables in the action angle variable in a Hamiltonian system. The integrals of the trajectories $\alpha(\mathbf{x})$ are also slow, and are called slow *observables.* This means that if $\alpha(\mathbf{x})$ is an integrable function with respect to time, then

$$\tilde{\alpha}(t) = \int_0^t \alpha(\mathbf{x})(s) ds$$

is slow, since $|d\tilde{\alpha}/dt| \leq C_0$ for some constant C_0. Slow observables can also be obtained using convolution with compactly supported kernels, as we will take advantage of in the following algorithm of this section.

We will now denote the set of slow variables as $\boldsymbol{\xi}(t) := \boldsymbol{\xi}(\mathbf{x}(\mathbf{t}))$. As any function of slow variables is also slow, it is convenient to look for functionally independent slow variables. Consider the set of slow variables $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_r)$. In order to ensure that $\xi_k \in \boldsymbol{\xi}$, $k = 1, \ldots, r$, are functionally independent, we require that $\nabla\xi_1, \nabla\xi_2, \ldots, \nabla\xi_r$ are linearly independent in $\mathcal{A}$. The dimension of the slow variables, r, is bounded by d, and we will look for the maximum number, r, of functionally independent slow variables.

In [2] it is shown that if a sufficient number of slow variables are approximated accurately, they are effectively closed, and *any* other smooth slow variable is approximated accurately. An averaging principle is used to prove that the evolution of $\boldsymbol{\xi}(t)$, with initial condition $\mathbf{x}_0$, is well approximated by the effective equation

$$\dot{\boldsymbol{\xi}} = F(\boldsymbol{\xi}), \quad \boldsymbol{\xi}(0) = \boldsymbol{\xi}_0 = \boldsymbol{\xi}(\mathbf{x}_0), \tag{20}$$

for large ω and $t \in [0, T]$. This corresponds to the effective equation (3) for our stiff system (2). Note that even though $\boldsymbol{\xi}$ is slow, $\dot{\boldsymbol{\xi}}$ is not necessarily slow, but it is bounded independent of ω. As before, we do not assume that the effective equation (20) is explicitly known. Indeed, the HMM algorithm approximates this assumed effective equation by numerical solutions of (19) in short time intervals of the computational domain $[0, T]$.

3.2 The Algorithm

We suppose that $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_r)$ are the slow variables of (19). The macro-solver sample times are denoted $t_1, t_2, \ldots, t_N$, where $N = T/H$, and the corresponding solution at time t_n is $\mathbf{x}(t_n) = \mathbf{x}_n$. The numerical algorithm is almost identical to the algorithm in section 2.1 except for changes related to the introduced slow variables. The macro-solver integrates the effective equation (20) by solving the original system (19) in a short time interval with initial conditions $\mathbf{x}_n$. Instead of averaging over $\dot{\mathbf{x}}$ as in Algorithm 1, an averaging over the time derivatives of $\boldsymbol{\xi}$ is done, found by the usual approximation

$$\dot{\boldsymbol{\xi}}(t) \sim (K_\eta * \dot{\boldsymbol{\xi}})(t) = \int_{t-\eta/2}^{t+\eta/2} K_\eta(s-t)\dot{\boldsymbol{\xi}}(s)ds. \tag{21}$$

Here $K_\eta(\cdot)$ is the compactly supported (scaled) kernel defined in section 2.2. After this approximation, we can evolve the original variables $\mathbf{x}_{n+1}$ using the macro-solver. In order to take a step with the new variables, say $\mathbf{x}(t+H) = \mathbf{x}(t) + \Delta\mathbf{x}$, we have to ensure that the new variables are consistent with the approximated $\dot{\boldsymbol{\xi}}$.

Algorithm 3 HMM Solver for Highly Oscillatory ODEs

```
1. Initial conditions: $\mathbf{x}(0) = \mathbf{x}_0$ and $n = 0$.
2. Force estimation:
    (a) Microsimulation: Solve (19) in $[t_n - \eta/2, t_n + \eta/2]$ with initial conditions $\mathbf{x}(t_n) = \mathbf{x}_n$.
    (b) Averaging: Approximate $\dot{\xi}_k(t_n)$ by $(K_\eta * \dot{\xi}_k)(t_n), \quad k = 1, \dots, r$.
3. Reconstruction: Update $\mathbf{x}_{n+1}$.
    (a) Solve
            $\delta_n \cdot \nabla_k \xi(\mathbf{x}_n) = (K_\eta * \dot{\xi}_k)(t_n), \quad k = 1, \dots, r$
        for $\tilde{F}_n$ with least squares.
    (b) Update $\mathbf{x}_{n+1}$ (forward Euler example):
            $\mathbf{x}_{n+1} = \mathbf{x}_n + H\delta_n.$
4. Repeat step 2-3 until $t_n = T$.
```

This is done by a second order estimate of $\Delta\mathbf{x}$, the least square solution[1] of the linear system

$$\Delta\mathbf{x} \cdot \nabla\boldsymbol{\xi}(\mathbf{x}_n) = H \cdot (K_\eta * \dot{\boldsymbol{\xi}})(t).$$

Here $\nabla\boldsymbol{\xi}(\mathbf{x}_n)$ is a matrix where the k'th row is the partial derivatives $\nabla\xi_k$. The complete HMM scheme is described by the steps in Algorithm 3.

In [2] they use a numerical process in their algorithm to identify the slow variables $\boldsymbol{\xi}$. This step is skipped in our algorithm, as we assume the slow variables are already found analytically for our test problems.

Notice that the micro-solver in step 2 (a) is only invoked whenever the time derivative of a slow variable, ξ_k, needs to be approximated. This means if $\xi_k \in \mathbf{x}$ is slow, then $\dot{\xi}_k$ is already known from the original system (19). An example is the slow variable q from the inverted pendulum example, which is directly solved by the macro-solver. The time derivative $\dot{\xi}_k$ in step 2 (b) is found by using the chain rule, setting $\dot{\xi}_k = \nabla\xi_k \cdot \dot{\mathbf{x}}$. In the following we will introduce an example of this algorithm applied on a highly oscillatory ODE.

[1] This could be solved with higher accuracy and in [9] they extend this algorithm with higher order estimates of $\Delta\mathbf{x}$.

4 The Spinning Top

The spinning top, also called the Lagrangian top, is the model of a simple child's toy. The spinning top has probably been around for centuries and the motion of the top has made both children and adults curious about it's underlying nature. We will give a brief introduction to rigid body dynamics, which finally leads us to the mathematical model of the spinning top.

4.1 Rigid Body Dynamics

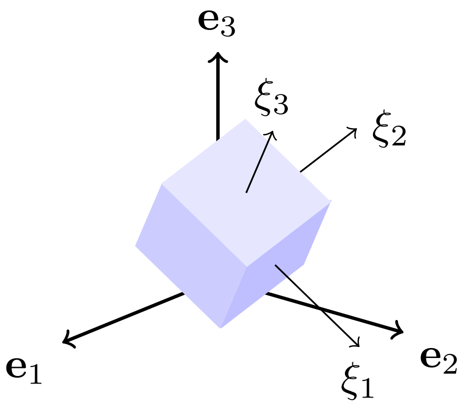


Figure 7: The rigid body with illustration of the spatial coordinate system $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ and the body coordinate system $(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3)$.

A *rigid body* is thought of as a body for which the distances between the points of the body are fixed as the body moves. One simple example is a box, with fixed distances between the corners of the body. We will consider the free rigid body anchored at the origin with no external forces acting.

The dynamics of the rigid body is usually described in *body coordinates*, which is a set of coordinates following the motion of the body. To get a better idea of what the body coordinates actually are describing, we need to go back to the coordinate axes we are usually concerned with. If we let $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) \in \mathbb{R}^3$ be the basis of the fixed space (see Figure 7), then the coordinates of a particle in space can be described by $x = (x_1, x_2, x_3) \in \mathbb{R}^3$. These are called *spatial coordinates*, and describe the motion of the particle *relative* to the fixed coordinate system. We can now define a new time-dependent coordinate system $(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3) \in \mathbb{R}^3$ defined by $\boldsymbol{\xi}_i = \boldsymbol{R(t)e}_i$ for $i = 1, 2, 3$, so that $\boldsymbol{\xi}_i$ is attached to the body. Then the *body coordinates* of a vector in $\mathbb{R}^3$ describe the motion relative to the new reference coordinates, $\boldsymbol{\xi}_i$ [13].

Euler's Rigid Body Equations. The rotation of a rigid body can be described by *Euler's equations.* This equation is found from the Lagrangian of the rigid body, and by defining the two variables, *angular velocity*, $\mathbf{\Omega}$, and *angular momenta*, $\mathbf{\Pi} = I\mathbf{\Omega}$, where I are the moments of inertia. We get the following equations describing the rigid body motions:

$$\dot{\mathbf{\Pi}} = \mathbf{\Pi} \times \mathbf{\Omega}. \tag{22}$$

Here $\mathbf{\Pi} = (\Pi_1, \Pi_2, \Pi_3)$ and $\mathbf{\Omega} = (\Omega_1, \Omega_2, \Omega_3)$ are the angular momenta and angular velocity in body coordinates [13]. We have the relation $\Pi_i = I_i\Omega_i$, $i = 1, 2, 3$, where $I = \text{diag}(I_1, I_2, I_3)$ are the moments of inertia and describe the mass distribution of the body. Note that (22) describes a *free* body, with no external forces acting on the dynamics. In what follows we will consider the rigid body in a gravitational field.

The Heavy Top. The rigid body with a fixed point in a gravitational field is called a *heavy top.* The equations of motion are more complex as we need to consider the effect of the gravity dragging the body downwards. The effect of gravity can be assessed if we introduce a new variable $\mathbf{\Gamma}$ that keeps track of the gravity vector as seen from the body: We let $\mathbf{\Gamma}$ represent the direction of unit vector along the Oz-axis as seen from the body [13] and the equations of motion become

$$\dot{\mathbf{\Pi}} = \mathbf{\Pi} \times \mathbf{\Omega} + Mgl\mathbf{\Gamma} \times \boldsymbol{\chi}$$

and

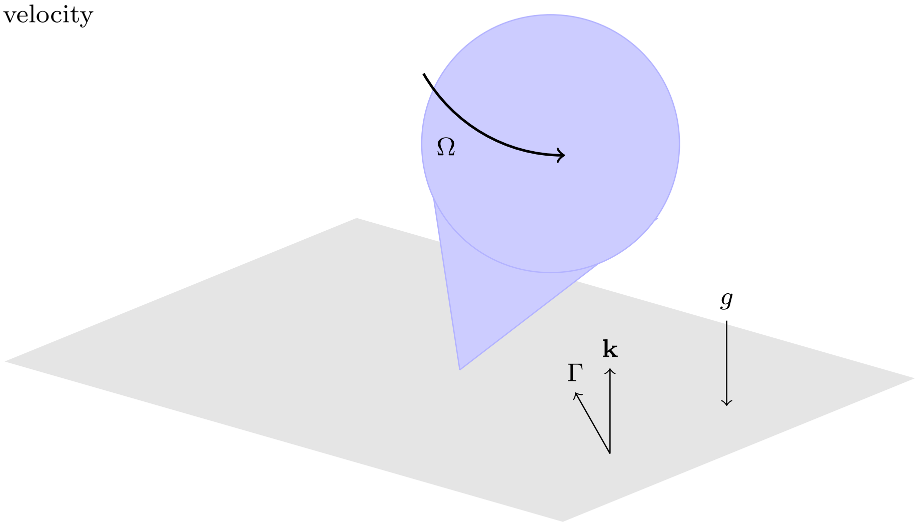
$$\dot{\mathbf{\Gamma}} = \mathbf{\Gamma} \times \mathbf{\Omega},$$

where M is the mass of body and g is the acceleration of gravity. The constant unit vector $\boldsymbol{\chi}$ is a vector along the line which connects the fixed point of the body (the origin) to the center of mass of the body. The line segment along $\boldsymbol{\chi}$ to the center of mass is of length l. Throughout this thesis, we will restrict our attention to the spinning top, which is an axially symmetric rigid body in a uniform gravitational field.

4.2 The Spinning Top

Consider the motion of a spinning top, also called the Langrangian top, which is an axially symmetric rigid body with the center of mass on the axis of symmetry, $\boldsymbol{\chi} = (0, 0, 1)$. This also means that the moment of inertia $I_1 = I_2$. Writing out the

M = mass of the body
g = gravity
l = length to the center of mass from the origin
Ω = angular velocity



equations for the heavy top we get the equations of motion

$$\begin{aligned} \dot{\Pi}_1 &= \frac{I_2 - I_3}{I_2 I_3}\Pi_2\Pi_3 + Mgl\Gamma_2, \\ \dot{\Pi}_2 &= \frac{I_3 - I_1}{I_1 I_3}\Pi_1\Pi_3 - Mgl\Gamma_1, \\ \dot{\Pi}_3 &= 0, \end{aligned} \qquad \text{and} \qquad \begin{aligned} \dot{\Gamma}_1 &= \frac{\Gamma_2\Pi_3}{I_3} - \frac{\Gamma_3\Pi_2}{I_2}, \\ \dot{\Gamma}_2 &= \frac{\Gamma_3\Pi_1}{I_1} - \frac{\Gamma_1\Pi_3}{I_3}, \\ \dot{\Gamma}_3 &= \frac{\Gamma_1\Pi_2}{I_2} - \frac{\Gamma_2\Pi_1}{I_1}. \end{aligned} \tag{23}$$

Note that the symmetry causes the angular momenta around the vertical (body) axis to be constant, since $\dot{\Pi}_3 = 0$. This variable also represents the spin of the top (Ω_3 is the angular velocity around the vertical direction of the top), and thus Π_3 is a conserved quantity. Another conserved variable is the hamiltonian

$$H(\mathbf{\Pi}, \mathbf{\Gamma}) = \frac{1}{2}\mathbf{\Pi} \cdot \mathbf{\Omega} + Mgl\mathbf{\Gamma} \cdot \chi, \tag{24}$$

which is the total energy of the top. In addition the Lagrangian top possesses two conserved quantities. Namely the projection of the angular momenta on the gravity vector, $\mathbf{\Pi} \cdot \mathbf{\Gamma}$, and the norm of the gravity vector, $\|\mathbf{\Gamma}\|^2$. In total this gives us four conserved quantities. We shall focus on the top spinning fast, such that the body is stable and prevented from falling down.

The Fast Top. In [14] they consider the *fast top* as a top where the kinetic energy of the rotation is large compared to the potential energy of the top. From the Hamiltonian in (24) the total potential energy is given by $Mgl\Gamma_3$, where $|\Gamma_3| \leq 1$. Since the kinetic energy from the spin is $I_3\Omega_3^2/2$, this leads to the criteria

$$\Pi_3^2 \gg 2MglI_3 \tag{25}$$

for the initial spin. In Marsden and Ratiu [13] they prove a similar condition for the stable top by the following theorem:

Theorem 2 (Heavy Lagrange Top Stability Theorem)**.** *An upright spinning Lagrange top is stable, provided that the angular velocity is strictly larger than $2\sqrt{MglI_1}/I_3$. It is unstable if the angular velocity is smaller than this value.*

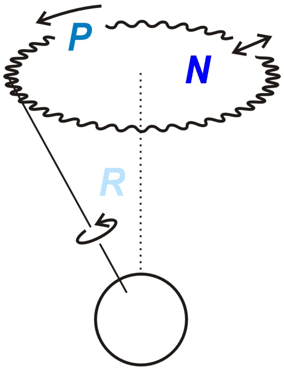


Figure 8: Precession, nutation and rotation of the spinning top.

Precession, Nutation and Rotation. In order to understand the physics of the spinning top we step back from the rigid body equations for a moment, and rather think of the behavior of the top, as we would intuitively expect it to be. Imagine you are placing the top on the ground and giving the top a high initial spin. You observe how the top spins fast, but you can also observe a slow circulation around the vertical axis (in the spatial stationary coordinate system, see Figure 7). This circulation is the *azimuthal* motion of the top and is called *precession.* The precession is described in spherical coordinates by $\dot{\varphi}$, where φ is the *azimuth* angle, which describes the rotation around the vertical axis.

Together with the precession we observe a wobbly motion of the top: We start spinning the top in an initial inclination θ_0 from the vertical upright position, where θ is the angle with respect to the vertical axis in spherical coordinates. But the inclination θ changes as the top precesses around the vertical axis. This periodical motion is called *nutation* and is described by $\dot{\theta}$.

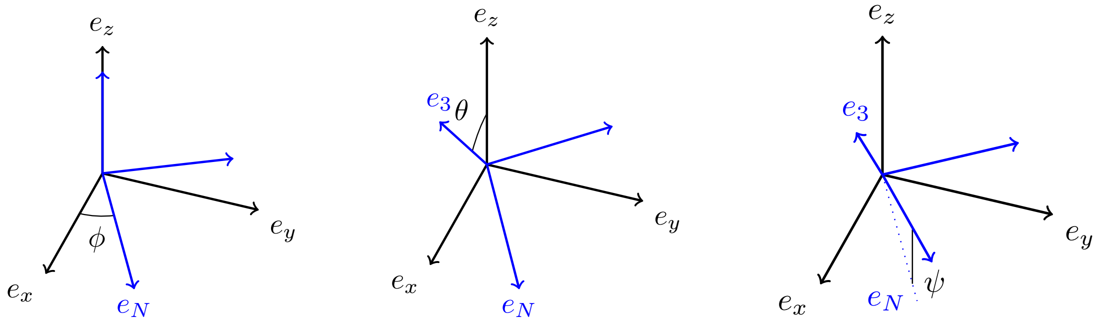


Figure 9: The conversion from the stationary frame $(\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z)$ to the moving frame by performing three rotations: **1**. Through an angle φ around the $\boldsymbol{e}_z$. **2**. Through an angle θ around $\boldsymbol{e}_N$. **3**. Through an angle ψ around the $\boldsymbol{e}_3$ axis.

In fact, the motion of the top can be completely described by the rotation around its own axis, nutation and precession. The three motions of the top are depicted in Figure 8. These motions can be described by the Euler angles, as we now will introduce.

The Euler Angles of the Spinning Top. The Euler angles introduce another way to describe the orientation of a rigid body. The movement of the spinning top can be described as a composition of three rotations with respect to the spatial coordinate system, (φ, θ, ψ), and their derivatives, $(\dot{\varphi}, \dot{\theta}, \dot{\psi})$. The orientation and definition of the three angles can bee seen in Figure 9. We refer the reader to the book by Arnold [14] or Marsden [13] for a more involved description of the Euler angles.

In [14] they derive the Lagrangian function in terms of the Euler angles and their derivatives to be

$$\mathcal{L} = \frac{I_1}{2}(\dot{\theta}^2 + \dot{\varphi}^2 \sin^2\theta) + \frac{I_3}{2}(\dot{\psi} + \dot{\varphi}\cos\theta)^2 - Mgl\cos\theta,$$

which are the *kinetic energy* of the system, minus the *potential energy* of the system. Once the Lagrangian is known, the equations of motions are obtained as the Euler-Lagrange equations. This means that we set

$$\frac{\partial \mathcal{L}}{\partial \varphi} = \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\varphi}}\right), \quad \frac{\partial \mathcal{L}}{\partial \theta} = \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\theta}}\right) \quad \text{and} \quad \frac{\partial \mathcal{L}}{\partial \psi} = \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\psi}}\right), \tag{26}$$

in order to find the differential equations for φ, θ and ψ.

Expression of $\boldsymbol{\Pi}$ and $\boldsymbol{\Gamma}$ in Terms of Euler Angles. In order to find the relation between the angular momenta, $\mathbf{\Pi}$, and the gravity vector, $\mathbf{\Gamma}$, and the Euler angles we would like to express the transformation from the spatial coordinate system to the body coordinate system, i.e. from the basis $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$ to the basis $(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3)$. Assume that $\boldsymbol{x} = (x_1, x_2, x_3)$ and $\boldsymbol{\zeta} = (\zeta_1, \zeta_2, \zeta_3)$ describe a vector in respectively the basis $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$ and $(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3)$. Then the transformation matrix $\boldsymbol{P}$, such that $\boldsymbol{\zeta} = \boldsymbol{P}\boldsymbol{x}$ is given as

$$\boldsymbol{P} = \begin{bmatrix} \cos\psi\cos\varphi - \cos\theta\sin\varphi\sin\psi & \cos\psi\sin\varphi + \cos\theta\cos\varphi\sin\psi & \sin\theta\sin\psi \\ -\sin\psi\cos\varphi - \cos\theta\sin\varphi\cos\psi & -\sin\psi\sin\varphi + \cos\theta\cos\varphi\cos\psi & \sin\theta\cos\psi \\ \sin\theta\sin\varphi & -\sin\theta\cos\varphi & \cos\theta \end{bmatrix}.$$

As mentioned previously, the time-dependent basis $(\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3)$, is defined by $\boldsymbol{\xi}_{\boldsymbol{i}} = \boldsymbol{R}(t)\boldsymbol{e}_i$, $i = 1, 2, 3$, so that $\boldsymbol{\xi}_1$, $\boldsymbol{\xi}_2$ and $\boldsymbol{\xi}_3$ move attached to the body. We can express the angular momenta in terms of (φ, θ, ψ) and $(\dot{\varphi}, \dot{\theta}, \dot{\psi})$ by noticing that $\boldsymbol{R}(t)$ is equal to $\boldsymbol{P}^T$ [13]:

$$\begin{bmatrix} \Pi_1 \\ \Pi_2 \\ \Pi_3 \end{bmatrix} = \begin{bmatrix} I_1 \sin\theta \sin\psi & I_1 \cos\psi & 0 \\ I_2 \sin\theta \cos\psi & -I_2 \sin\psi & 0 \\ I_3 \cos\theta & 0 & I_3 \end{bmatrix} \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{27}$$

We can also express the motion of the vector $\mathbf{\Gamma}$ in terms of Euler angles by observing that the last column of the matrix $\boldsymbol{P}$ transforms the unit vector along the Oz-axis, which gives the relation

$$\begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \Gamma_3 \end{bmatrix} = \begin{bmatrix} \sin\theta \sin\psi \\ \sin\theta \cos\psi \\ \cos\theta \end{bmatrix}. \tag{28}$$

We can now substitute the expression of the gravity vector $\mathbf{\Gamma}$ in terms of the Euler angles into (27), and formulate the angular momenta $\mathbf{\Pi}$ by

$$\begin{bmatrix} \Pi_1 \\ \Pi_2 \\ \Pi_3 \end{bmatrix} = \begin{bmatrix} I_1\Gamma_1 & I_1\Gamma 2/\sin\theta & 0 \\ I_2\Gamma_2 & -I_2\Gamma_1/\sin\theta & 0 \\ I_3\Gamma_3 & 0 & I_3 \end{bmatrix} \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \tag{29}$$

By inversion of this matrix the transformation from $(\dot{\varphi}, \dot{\theta}, \dot{\psi})$ to (Π_1, Π_2, Π_3) is given as

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{\Gamma_1^2 + \Gamma_2^2} \begin{bmatrix} \Gamma_1/I_1 & \Gamma 2/I_2 & 0 \\ -\Gamma_1\Gamma_3/I_1 & -\Gamma_2\Gamma_3/I_2 & (\Gamma_1^2 + \Gamma_2^2)/I_3 \\ \Gamma_2 \sin\theta/I_1 & -\Gamma_1 \sin\theta/I_2 & 0 \end{bmatrix} \begin{bmatrix} \Pi_1 \\ \Pi_2 \\ \Pi_3 \end{bmatrix}. \tag{30}$$

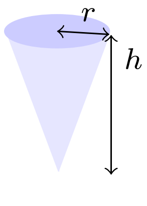
4.3 The Fast Spinning Top

We will consider the fast spinning top by using the criterion in (25). This means that we restrict the angular momentum Π_3 to be much larger than $\sqrt{2MglI_3}$. By defining the variable $\hat{\Pi}_3 = \sqrt{2MglI_3}$ and let $\Pi_3 = \omega\hat{\Pi}_3$, we choose ω sufficiently large to ensure the criterion $\Pi_3 \gg \sqrt{2MglI_3}$ is fulfilled. In this way we can observe the behavior of the top as we increase ω. We rewrite the equations of motion from (23) by replacing Π_3 with $\omega\hat{\Pi}_3$, and get

$$\begin{aligned} \dot{\Pi}_1 &= \omega\frac{I_2 - I_3}{I_2 I_3}\Pi_2\hat{\Pi}_3 + Mgl\Gamma_2, & & & \dot{\Gamma}_1 &= \omega\frac{\Gamma_2\hat{\Pi}_3}{I_3} - \frac{\Gamma_3\Pi_2}{I_2}, \\ \dot{\Pi}_2 &= \omega\frac{I_3 - I_1}{I_1 I_3}\Pi_1\hat{\Pi}_3 - Mgl\Gamma_1, & & \text{and} & \dot{\Gamma}_2 &= \frac{\Gamma_3\Pi_1}{I_1} - \omega\frac{\Gamma_1\hat{\Pi}_3}{I_3}, \\ \dot{\Pi}_3 &= 0, & & & \dot{\Gamma}_3 &= \frac{\Gamma_1\Pi_2}{I_2} - \frac{\Gamma_2\Pi_1}{I_1}. \end{aligned} \tag{31}$$

We solve the system in (31) numerically in the time interval $[0, T]$ by a fourth order Runge-Kutta (from now on referred to as RK4) scheme in order to see how the variables behave. The norm of the gravity vector, $\|\mathbf{\Gamma}\|^2$, was set to be equal to one by using the initial value $\mathbf{\Gamma}^0 = (0, \sin(0.2), \cos(0.2))^T$. The initial angular momenta of the top were $\mathbf{\Pi}^0 = (0, 0, \omega\hat{\Pi}_3)^T$. The geometry of the top were chosen to imitate a circular cone, see Figure 10. The mass was $M = 5$, height equal to one and radius 0.4. The moments of inertia with the chosen parameters are $I = (3.12, 3.12, 0.24)$ and the center of mass of the circular cone is located at the vertical axis, positioned at $3/4\ h$ from the tip (the origin). Thus the length from the fixed point at the origin to the center of mass is $l = 0.75$. The gravity field was $g = 9.81$. In Figure 11 the numerical solution of $\mathbf{\Pi}$ and $\mathbf{\Gamma}$ is plotted for $T = 0.1$, using RK4 with step size 10^{-5} for the modest value $\omega = 100$.

Time Scales of The Spinning Top. The fast oscillating nature of the spinning top is easily seen Figure 11. But different motions are observed at different



$$I_1 = I_2 = \tfrac{3}{5}M(\tfrac{r^2}{4} + h^2)$$
$$I_3 = \tfrac{3}{10}Mr^2$$

Figure 10: Moments of inertia for a circular cone with mass M, radius r and height h.

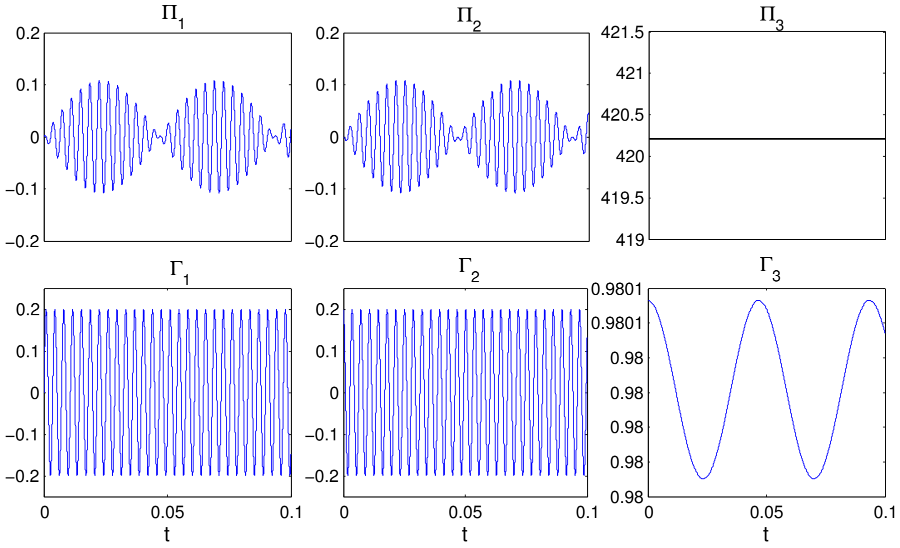


Figure 11: The solution of the spinning top with initial spin $\Pi_3 = \omega\sqrt{2MglI_3}$, $\omega = 100$ solved with fourth order Runge-Kutta with step size 10^{-5}. We observe that the angular momenta Π_1 and Π_2, along with Γ_1 and Γ_2, are highly oscillatory with frequency of order ω. The variable Γ_3 is oscillating slowly with a frequency of order $\omega/10$.

time scales: At the scale ω^{-1} we observe an oscillatory behavior of the variables Π_1, Π_2, Γ_1 and Γ_2. The variable Γ_3 oscillates slowly at the time scale $10 \cdot \omega^{-1}$, while Π_3 remains constant, as expected. As we increase ω, the frequency of Π_1, Π_2, Γ_1, Γ_2 and Γ_3 are increasing proportionally to ω: The angular momenta Π_1 and Π_2 are oscillating with frequencies of order ω and amplitudes of order $10/\omega$. The solutions of Γ_1 and Γ_2 oscillate with frequencies of order ω, while the amplitudes remain constant of order 10^{-1}. The variable Γ_3 is changing slowly and has a frequency of order $\omega/10$.

Numerical solutions of the spinning top require an overwhelming computational cost using standard solvers, as for instance RK4. For $\omega = 100$, an accurate solution can only be obtained by using a step size of order ω^{-2} with RK4. As we increase the size of ω, we would need a minimum of ω^2 operations in order to solve the full system accurately, causing a high computational cost as $\omega \to \infty$. In this setting the HMM could be used to reduce the computational cost substantially by eliminating the need of the full representation of the problem and rather solve the system for some slow constituents of the system in (31).

The Slow Variables of the Fast Spinning Top. Although the top seems to have a fast varying nature, there are still some quantities that we would expect to behave slowly as the spin of the top is increased. Our goal is now to extract the slow behavior of the fast spinning top. The slowly changing variables of the system can then be approximated accurately using the multiscale method considered in section 3.

In order to analyze the fast and slowly changing variables of the top, we turn to the differential equation in (31). From the set of equations, we can easily see that the variables Π_1, Π_2, Γ_1 and Γ_2 are determined to vary fast as ω increases. This is also the behavior we have found numerically (see Figure 11). However, it is possible to extract some slow behavior from the system in (31).

Remember that a slowly changing variable has a derivative which is bounded independent of ω (see section 3.1). Keeping this in mind, we can find some slowly changing quantities from (31). Two slow variables can be found directly from these equations: The derivative of Γ_3 is a function of Π_1, Π_2, Γ_1 and Γ_2, and we have found numerically that these variables are bounded as $\omega \to \infty$ (Π_1 and Π_2 are proportional to $1/\omega$ and Γ_1 and Γ_2 are constant), while the derivative of Π_3 is zero. Another slow variable is identified by observing that $d/dt(\Pi_1^2+\Pi_2^2) = Mgl(\Pi_1\Gamma_2 - \Pi_2\Gamma_1)$, which is bounded independently of ω. We identify the remaining slow variables in the same manner, resulting in the five variables

$$\begin{aligned}
\xi_1 &= \Pi_3 \\
\xi_2 &= \Gamma_3 \\
\xi_3 &= \Pi_1^2 + \Pi_2^2 \\
\xi_4 &= \Gamma_1^2 + \Gamma_2^2 \\
\xi_5 &= \Pi_1\Gamma_1 + \Pi_2\Gamma_2.
\end{aligned} \tag{32}$$

Remarks on The Choice of Slow Variables. There are two things to be aware of when we determine the slow variables: The first thing to notice is that the set of slow variables $\boldsymbol{\xi}$ needs to be *functionally independent*, which also implies that the rows (or columns) of $\nabla\boldsymbol{\xi}$ are linearly independent. This needs to be valid for the chosen set of slow variables (32), and we check if this condition is fullfilled by an inspection of the Jacobian matrix $\nabla\boldsymbol{\xi}$. We find that the rows of the Jacobian are linearly independent except for at the surface $\Pi_1/\Gamma_1 = \Pi_2/\Gamma_2$, or equivalently, $\Pi_1/\Pi_2 = \Gamma_1/\Gamma_2$. This will however not lead to problems when it comes to numerical evaluations.

Another aspect we need to consider when choosing the slow variables is to ensure that the solution of ξ_k gives non-zero values of $F(\xi_k)$. This is not necessarily valid with our choice of slow variables, unless we choose the initial values of the system in (31) carefully: With the starting values $\mathbf{\Pi}_0 = (0, 0, \omega\hat{\Pi}_3)^T$ and $\mathbf{\Gamma}_0 = (0, \sin(0.2), \cos(0.2))^T$ the microsimulation in the first step of the macrosolver (see step 2 a in Algorithm 3) gives symmetric solutions of the variables $\mathbf{\Pi}$ and $\mathbf{\Gamma}$. Thus the slow variables from the microsimulation also become symmetric.[2] This is resulting in anti-symmetric derivatives of the slow variables, and thus leading to the approximate force $\tilde{F}(\boldsymbol{\xi}) = 0$. Hence, we need to choose the initial conditions to give a non-symmetric solution of $\boldsymbol{\xi}$.

[2] If X_1 and X_2 are symmetric, then $X_1 + X_2$ and X_1^2 are also symmetric.

5 The Fast Spinning Top and HMM

The solution of the slow variables in (32) of the fast spinning top is obtained by using Algorithm 3. An outline of the procedure is described more extensive in Algorithm 4. We will now discuss the algorithm in detail.

Algorithm 4 HMM solver for the slow variables of the spinning top

1. Initial conditions: Given $(\mathbf{\Pi}^0, \mathbf{\Gamma}^0) = (\mathbf{\Pi}(0), \mathbf{\Gamma}(0))$, t_n and $n = 0$.
2. Predict the fast variables $\mathbf{\Pi}$, $\mathbf{\Gamma}$ by
$$\begin{bmatrix} \hat{\mathbf{\Pi}} \\ \hat{\mathbf{\Gamma}} \end{bmatrix}^n = \begin{bmatrix} \mathbf{\Pi} \\ \mathbf{\Gamma} \end{bmatrix}^n + H\delta_{n-1}.$$
(At $n = 0$ the prediction is $\hat{\mathbf{\Pi}}^0 = \mathbf{\Pi}^0$ and $\hat{\mathbf{\Gamma}}^0 = \mathbf{\Gamma}^0$).
3. Force estimation:
   (a) Microsimulation: Solve (31) for the variables $\mathbf{\Pi}$ and $\mathbf{\Gamma}$ in $[t_n - \eta/2, t_n + \eta/2]$ with initial conditions $\hat{\mathbf{\Pi}}$ and $\hat{\mathbf{\Gamma}}$.
   (b) Averaging: Approximate the derivatives of ξ_2, ξ_3, ξ_4 and ξ_5:
$$\tilde{F}(\xi_i) = \dot{\xi}_i(t_n) \sim (K_{\eta/2} * \dot{\xi}_i)(t_n), \quad i = 2, 3, 4, 5.$$
4. Reconstruction:
   (a) Solve
$$\begin{bmatrix} \nabla\xi_2(t_n) \\ \nabla\xi_3(t_n) \\ \nabla\xi_4(t_n) \\ \nabla\xi_5(t_n) \end{bmatrix} \cdot \delta_n(\mathbf{\Pi}, \mathbf{\Gamma}) = \begin{bmatrix} \dot{\xi}_2(t_n) \\ \dot{\xi}_3(t_n) \\ \dot{\xi}_4(t_n) \\ \dot{\xi}_5(t_n) \end{bmatrix}$$
   for $\tilde{F}_n$ with least squares.
   (b) Update $\mathbf{\Pi}$ and $\mathbf{\Gamma}$:
$$\begin{bmatrix} \mathbf{\Pi} \\ \mathbf{\Gamma} \end{bmatrix}^{n+1} = \begin{bmatrix} \mathbf{\Pi} \\ \mathbf{\Gamma} \end{bmatrix}^n + H\delta_n(\mathbf{\Pi}, \mathbf{\Gamma}).$$
5. Repeat step 2-4 until $t_n = T$.

The HMM Scheme. The system in (31) will from now on be referred to as the microscale system, in compact form given as

$$\dot{\mathbf{\Pi}} = f(\mathbf{\Pi}, \mathbf{\Gamma}), \quad \dot{\mathbf{\Gamma}} = g(\mathbf{\Pi}, \mathbf{\Gamma}). \tag{33}$$

The macroscale system, containing the slow variables in (32), is given as

$$\dot{\boldsymbol{\xi}} = F(\boldsymbol{\xi}). \tag{34}$$

We remark that we do not solve for $\xi_1 = \Pi_3$ in practice, since we already know that this is a constant, thus the remaining slow variables to solve are ξ_2, ξ_3, ξ_4 and ξ_5. The estimated force $F(\boldsymbol{\xi})$ is found by a microsimulation of the original system in (31) using a numerical solver with step size h in a small interval η (step 3 (a) of the algorithm). For this purpose we use a Störmer-Verlet like scheme, given as

$$\begin{aligned} \mathbf{\Pi}^{n+1/2} &= \mathbf{\Pi}^n + \frac{h}{2} f(\mathbf{\Pi}^n, \mathbf{\Gamma}^n) \\ \mathbf{\Gamma}^{n+1} &= \mathbf{\Gamma}^n + hg(\mathbf{\Pi}^{n+1/2}, \mathbf{\Gamma}^n) \\ \mathbf{\Pi}^{n+1} &= \mathbf{\Pi}^n + 1 + \frac{h}{2} f(\mathbf{\Pi}^{n+1/2}, \mathbf{\Gamma}^{n+1}), \end{aligned} \tag{35}$$

or the fourth order Runge-Kutta scheme (for a detailed description of this integrator see [12] page 28-30). In step 3 (b) of the algorithm we approximate the derivatives of the macroscale system in (34) by a convolution with the exponential kernel $K^{\exp(2)}$ given in (9)[3]. In practice, the convolution is done either by using the trapezoidal rule with step zize h, or evaluate the contribution to $F(\boldsymbol{\xi})$ in each step of the microintegration.

At last, in step 4 of the algorithm, we need to address the force to the original variables in (33). This is done by solving the linear system $\nabla\boldsymbol{\xi} \cdot \delta_n = \dot{\boldsymbol{\xi}}$ with least squares and updating the variables $\mathbf{\Pi}$ and $\mathbf{\Gamma}$ by using the macrointegrator (step 4 b). This integration is performed using Forward Euler or Leapfrog integration with step size H, given as

HMM-FE-*
Given $\mathbf{\Pi}^n, \mathbf{\Gamma}^n$, *for* n = *0,1,2, ...*

$$\begin{bmatrix} \mathbf{\Pi} \\ \mathbf{\Gamma} \end{bmatrix}^{n+1} = \begin{bmatrix} \mathbf{\Pi} \\ \mathbf{\Gamma} \end{bmatrix}^{n} + H\delta_n(\mathbf{\Pi}, \mathbf{\Gamma}).$$

[3] We also used the kernel $K^{\exp(1)}$ given in (9), but the results with $K^{\exp(2)}$ gave better estimates (smaller error).

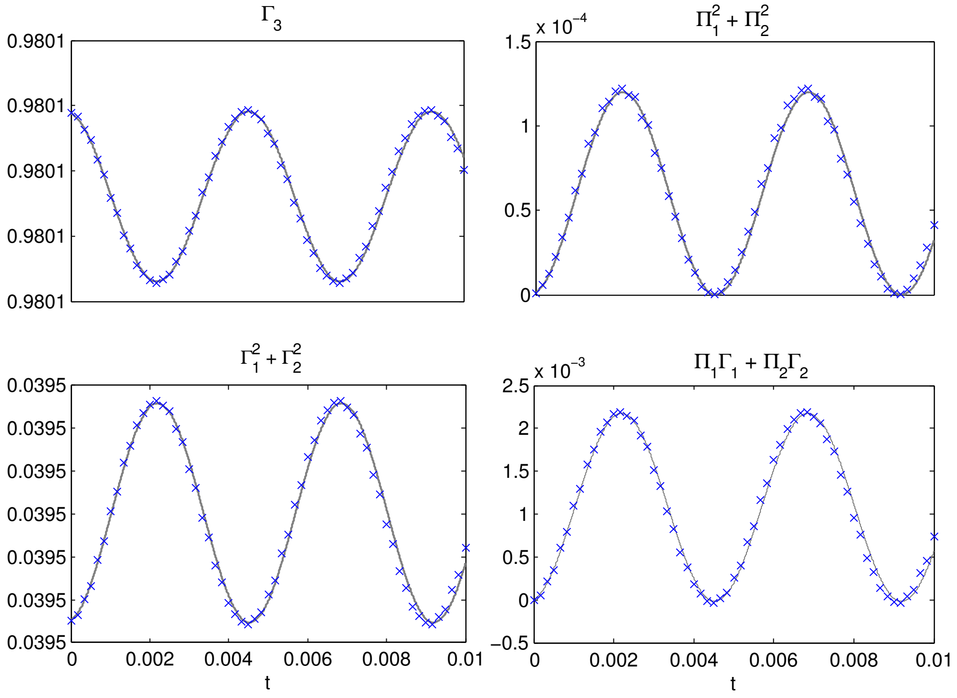


Figure 12: The HMM approximation (×) of the slow variables for $\omega = 10^3$ plotted with the solution of the slow variables obtained by Matlab's ODE45 (gray line).

HMM-LF-*
Given $\mathbf{\Pi}^n, \mathbf{\Gamma}^n$, *for* n *= 0,1,2, . . .*

$$\begin{bmatrix} \mathbf{\Pi} \\ \mathbf{\Gamma} \end{bmatrix}^{n+1} = \begin{bmatrix} \mathbf{\Pi} \\ \mathbf{\Gamma} \end{bmatrix}^{n-1} + 2H\delta_n(\mathbf{\Pi}, \mathbf{\Gamma}).$$

5.1 Numerical Results

The numerical solution of the slow variables of (31) is evaluated using the procedure in Algorithm 4. The parameters used in the numerical evaluations are $M = 5$, $l = 0.75$, $g = 9.81$ and $I = (3.12, 3.12, 0.24)$. As discussed in section 4.3, the solution of $F(\boldsymbol{\xi})$ becomes zero with the initial values $\mathbf{\Gamma} = (0, \sin(0.2), \cos(0.2))^T$ and $\mathbf{\Pi}_0 = (0, 0, \omega\hat{\Pi}_3)^T$. This problem is avoided by letting the angular momentum Π_1 have an initial value a bit larger than zero, for example $\Pi_1 = 1/\omega$. The initial value $\mathbf{\Pi}_0 = (1/\omega, 0, \omega\hat{\Pi}_3)^T$ leads to a nonzero $F(\boldsymbol{\xi})$.

Figure 12 shows the approximated solution using the HMM scheme with $T = 0.01$, $H = T/60 \approx 1.67 \cdot 10^{-4}$, $h = 10^{-6}$ and $\eta = 10h$. Note that the microintegration is done in an interval of length $10h$, which is only 6% of the length of

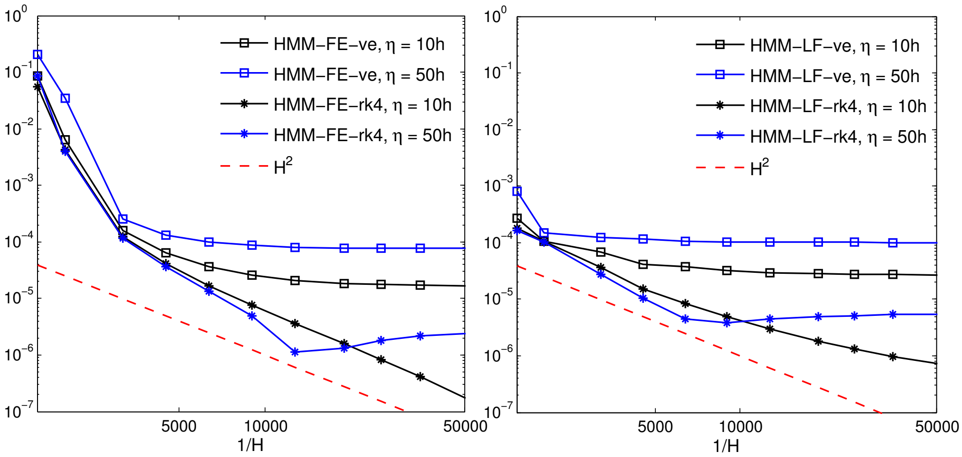


Figure 13: Errors in ξ_3 as a function of $1/H$ for $\omega = 10^3$. The convergence of the first order scheme HMM-FE-rk4 and HMM-FE-ve are shown in the plot to the left. The convergence of the Leapfrog schemes HMM-LF-rk4 and HMM-LF-ve are shown in the plot to the right. The $\mathcal{O}(H)$ behavior of the schemes stagnates as H gets small enough.

the large scale time step H. The HMM requires in total 600 operations in the evaluation with this choice of parameters, compared to RK4 which would require an amount of 10^4 operations in order to give an accurate solution. The solution is compared to the solution obtained with Matlab's solver ODE45, which is almost exact (except for an error 10^{-16}). From now on this solution will be referred to as the exact solution.

Error of the HMM Scheme. The local error is found from the estimate in (10), which gives the local truncation error of $E_n = \mathcal{O}(H^{k+1} + \varepsilon_{\text{HMM}})$, where k is the order of the macrointegrator. The error of ε_{HMM} consists of four main sources [2]: The first source is the analystical error from the force approximation in (8). The second error contribution is the error related to the microintegration, used to find the approximate variables $\mathbf{\Pi}^n$ and $\mathbf{\Gamma}^n$. The two last contributions are the error from using the quadrature rule to estimate the force $F(\boldsymbol{\xi})$, and the error due to the inaccurate slow variables. Altogether ε_{HMM} can be expressed as

$$\varepsilon_{\text{HMM}} = \varepsilon_{\text{force}} + \varepsilon_{\text{micro}} + \varepsilon_{\text{quad}} + \varepsilon_{\text{slow}} \tag{36}$$

In the proceeding, we will consider the error metric of the numerical approximation as the maximum in the interval $t \in [0, 0.01]$ of the error $|\xi_i^n - \xi_i(t_n)|$ [4], where ξ_i^n is

[4] It would have been more convenient to consider at the relative error $|\xi_i^n - \xi_i(t_n)|/|\xi_i(t_n)|$, as the values of the slow variables are small. However, since our intention is to examine the error behavior as H and h are changing, this does not affect the analysis.

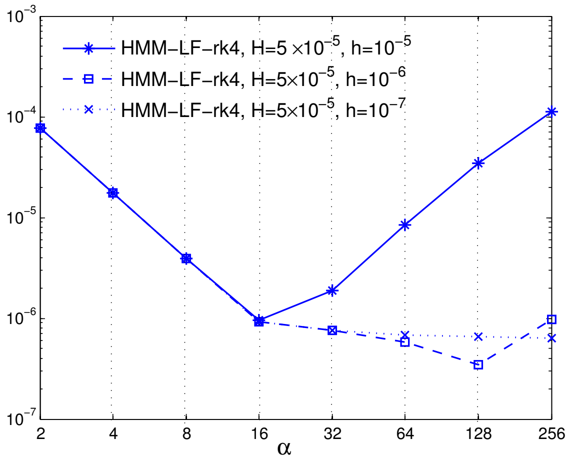


Figure 14: Errors in ξ_3 as a function of $\eta = \alpha h$ for $\omega = 10^3$, $H = T/160$ with $h = 10^{-5}$, 10^{-6} and 10^{-7} using the scheme HMM-LF-rk4.

the numerical solution and $\xi_i(t_n)$ is the exact solution at the time $t_n = nH$.

Figure 13 shows the error of ξ_3 as a function of the macro scale step size H using Forward Euler or Leapfrog integration as macrointegrators. The parameters that were used are $h = 10^{-6}$ and $\eta = 10h$ or $50h$. We compare the error of using both the Störmer-Verlet scheme in (35) and RK4 as microintegrators.

The figure shows that the error is decreasing as we decrease H, but converges to a value as H gets small enough. The error in HMM-LF-* are converging faster to this value, as this method performs better than HMM-FE-* for larger values of H. The $\mathcal{O}(H)$ dependence of HMM-LF-* is not shown here, but is illustrated in Table 1 and will be discussed subsequently.

The flattening shown in Figure 13 is a consequence of the error related to HMM, ε_{HMM}, which becomes dominant as H decreases. The local error of the HMM scheme can thus be formulated as $E_n \leq C \max\{H^{k+1}, \varepsilon_{\text{HMM}}\}$, for some $C > 0$ [2]. The starting point of this the flattened error behavior is strongly dependent on the microintegrator: The flattening of the error with RK4 as microintegrator is delayed, especially by using $\eta = 10h$. However, this could be fixed easily by choosing a smaller step size h in the microintegrator.

Another important observation in Figure 13 is the dependence on the time interval η. The error increases as we increase the interval of the force estimation, which may not be what we would expect intuitively. The dependence on η is shown in Figure 14 for different values of step sizes h. The optimal choice of η is almost independent of H (this has been tested numerically), but as we decrease h, we could get better results by increasing η. According to the figure, the optimal length of the microintegration is $\eta = 16h$.

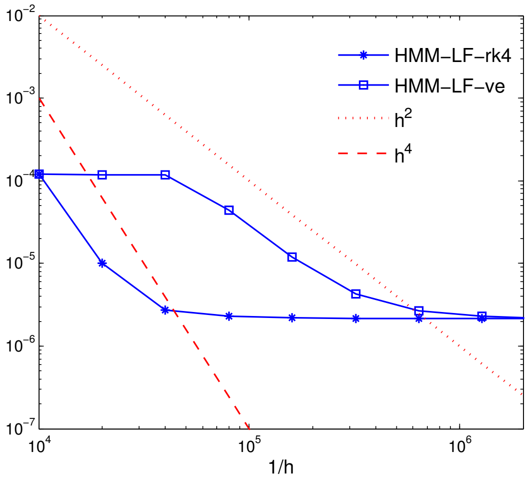


Figure 15: Errors in ξ_3 as a function of $1/h$ with $H = T/160$ for $\omega = 10^3$ for the HMM schemes HMM-LF-rk4 and HMM-LF-ve.

Table 1: Error of the slow variable ξ_3 the interval $t \in [0, 0.01]$ for $\omega = 10^3, 10^4, 10^5$, using HMM-LF-rk4.The micro step size is $h = 10/\omega^2$ and $\eta = 10h$.

$\omega = 10^3$			$\omega = 10^4$			$\omega = 10^5$		
H	Steps	Error	H	Steps	Error	H	Steps	Error
$T/40$	400	2.19(-5)	$T/400$	4000	2.64(-6)	$T/4000$	40 000	3.85(-7)
$T/80$	800	5.90(-6)	$T/800$	8000	9.45(-7)	$T/8000$	80 000	2.65(-7)
$T/160$	1600	2.25(-6)	$T/1600$	16000	3.49(-7)	$T/16000$	160 000	2.29(-7)
$T/320$	3200	1.06(-6)	$T/3200$	32000	1.86(-7)	$T/32000$	320 000	1.59(-7)

Figure 15 shows the error of ξ_3 as a function of the microscale step size h using the Leapfrog integrator as macrointegrator. The plot shows the same behavior as in the previous error plot (13): The error is flattened out as h decreases, and this error is not overcome by decreasing h. Again, this is a result of the combinations of error sources in the HMM scheme, the error related to the macrointegration are dominating the error E_n. We also observe that the accuracy of ξ_3 using RK4 as microintegrator is much better for larger values of h. The same behavior is observed with Forward Euler as macrointegrator.

A scaling of the errors can be done, by balancing the two errors $\varepsilon_{\text{macro}}$ and ε_{HMM}, to a prescribed accuracy Δ. This has been done in [2], and we refer the reader to this article for a detailed description.

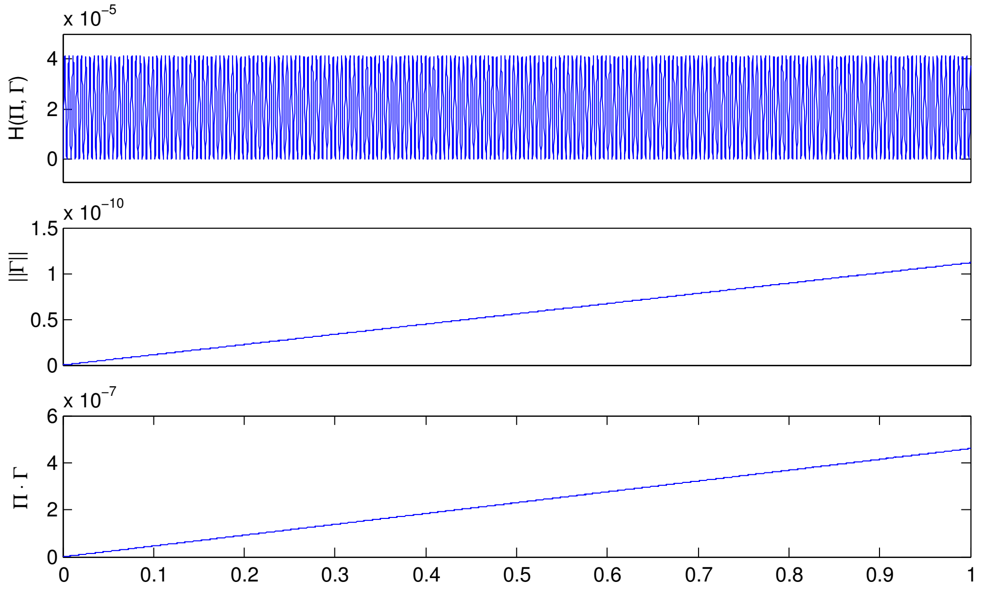


Figure 16: Preservation of first integrals for $\omega = 10^3$ for HMM-LF-rk4 in the interval $t \in [0, 1]$ with $H = T/6000 \approx 1.67 \cdot 10^{-4}$, $h = 1/\omega^2$, $\eta = 10h$.

Computational Complexity. Table 1 shows the error of ξ_3 using the scheme HMM-LF-rk4 with $\omega = 10^3, 10^4$ and 10^5. The error possess an $\mathcal{O}(H)$ behavior, until the error ε_{HMM} is dominating. We need to emphasize that the behavior of the slow variables are not independent of ω: We observed that the slow variables oscillate with a frequency $\omega/10$ (see Figure 12). This means that we need to adjust the step sizes H and h as ω increases in order to maintain an accurate solution. However, this will not lead to much more effort for the microsimulation, as the interval η is adjusted by h: The interval length $\eta = 10h$ is enough to get accurate solutions. Thus the only increasing computational cost originates from the macrosolver.

From the table, it is easy to see the computational gain by using HMM. For $\omega = 10^3$ and $H = T/160$, the HMM produces an error of size 10^{-6} with only 1600 microsteps. A standard solver, for instance RK4, would need a minimum of 10^5 microsteps to evaluate an accurate solution. As we increase ω, the computational savings are more significant. For $\omega = 10^5$ we would need a total of $\mathcal{O}(10^7)$ operations to evaluate the solution using RK4, while the HMM needs a total of $\mathcal{O}(10^4)$ operations to get an accurate solution, leading to computational savings of $\mathcal{O}(10^3)$. The error of the remaining slow variables behave in the same way as for ξ_3, and will not be further discussed here.

Another aspect of the HMM is the good preservation of the conserved quantities. Remember that the spinning top possesses four conserved quantities: the Hamiltonian given in (24), the projection of the angular momenta on the gravity vector, $\mathbf{\Pi} \cdot \mathbf{\Gamma}$, and lastly the norm of the gravity vector, $\|\mathbf{\Gamma}\|^2$. Figure 16 shows the error of the conserved quantities with $\omega = 10^3$ and $T = 1$. Notice that the number of microsteps taken in this interval is only $6 \cdot 10^4$.

5.2 Precession, Nutation and Rotation of the Spinning Top

Another way to describe the dynamics of the fast spinning top is through the Euler angles, described in section 4.2. The orientation of the top is given by the three angles (φ, θ, ψ), and their derivatives ($\dot{\varphi}$, $\dot{\theta}$, $\dot{\psi}$). In section 4.2 we found the relation between the Euler angles and the angular momenta $\mathbf{\Pi}$ and the gravity vector $\mathbf{\Gamma}$, see (29) and (30). This transformation can be written in compact form as $(\Pi_1, \Pi_2, \Pi_3)^T = A(\dot{\varphi}, \dot{\theta}, \dot{\psi})^T$, and visa versa, $(\dot{\varphi}, \dot{\theta}, \dot{\psi})^T = A^{-1}(\Pi_1, \Pi_2, \Pi_3)^T$, where A is the transformation matrix in (29).

In this way, we can evaluate the Euler angles from the solution of $\mathbf{\Pi}$ and $\mathbf{\Gamma}$ of (31). The behavior of $(\dot{\varphi}, \dot{\theta}, \dot{\psi})$ is found by numerical simulation using RK4, and

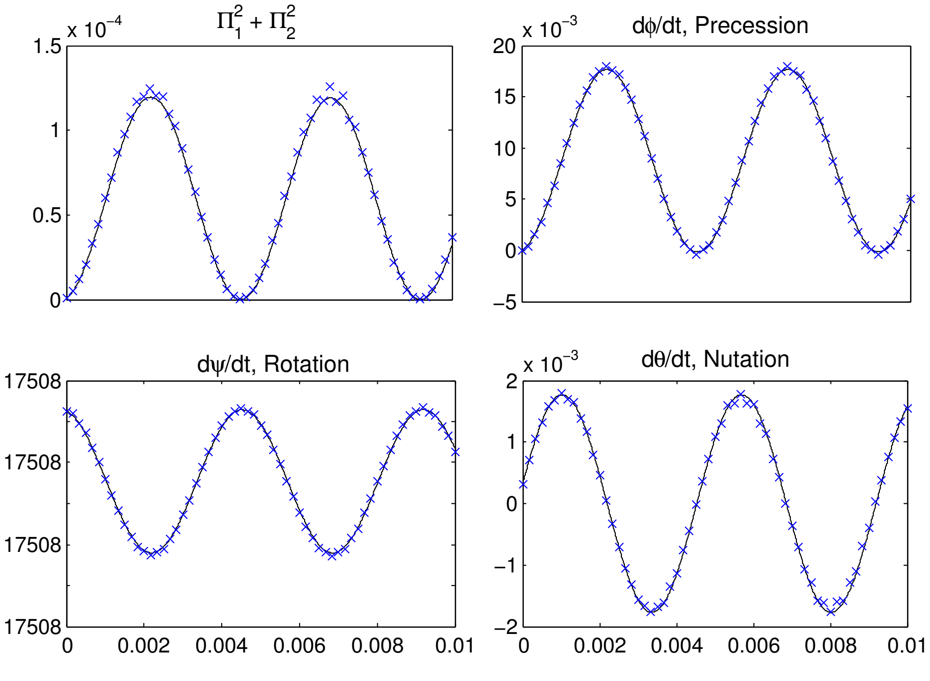


Figure 17: The HMM approximation ($\times$) of the slow variables in (37) for $\omega = 10^3$ plotted with the solution of the slow variables obtained by Matlabs ODE45 (gray line).

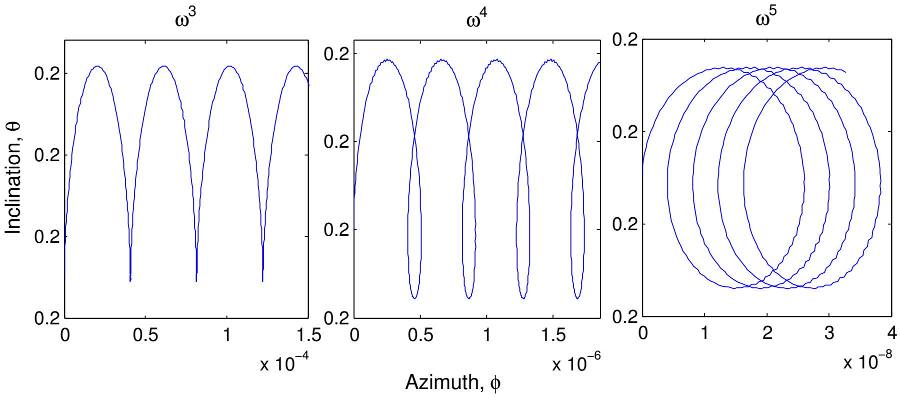


Figure 18: Plot of the solutions for the Euler angles in the (φ, θ)-plane, showing the inclination as a function of the azimuthal motion. The precession of the top gets smaller as the spin of the top, ω, increases.

we observe that $(\dot{\varphi}, \dot{\theta}, \dot{\psi})$ changes slowly compared to the fast oscillations of $\mathbf{\Pi}$ and $\mathbf{\Gamma}$. The choice of slow variables of the previous section is not unique: we can let $(\dot{\varphi}, \dot{\theta}, \dot{\psi})$ be slow variables of the problem in (31), and impose the new set of slow variables:

$$\begin{aligned}
\xi_1 &= \Pi_3 \\
\xi_2 &= \Pi_1^2 + \Pi_2^2 \\
\xi_3 &= \dot{\varphi} \\
\xi_4 &= \dot{\theta} \\
\xi_5 &= \dot{\psi}.
\end{aligned} \tag{37}$$

We use the procedure in Algorithm 4 and find the approximate solution of the new set of slow variables. The parameters M, g, l, I and the initial values $\mathbf{\Pi}^0$ and $\mathbf{\Gamma}^0$ are the same as in the previous section. Figure 17 shows the solution with $T = 0.01$, $H = T/60$, $h = 10^{-6}$ and $\eta = 300h = 3 \cdot 10^{-5}$. Note that the HMM solver needs an microintegration interval of length $300h$ to get an accurate solution when using the new set of slow variables. In addition, the solution is more sensitive to the step size h, which needs to be of order 10^{-7} to achieve good results.

This choice of slow variables is demanding more computational effort in the evaluation than with the previous choice of slow variables (32). But we know that when the maximum set of slow variables ($d-1$ is the maximum set of slow variables, where d is the dimension of our original variables in (31)) are approximated accurately, then *any* other slow variable is automatically approximated. Thus, we rather approximate the solution of $(\dot{\varphi}, \dot{\theta}, \dot{\psi})$, and $(\varphi,\ \theta,\ \psi)$, by using the first choice of slow variables in (32), leading to a reduced number of computational operations [2], [10].

Figure 18 shows the evolution of θ, the inclination form the z-axis, as a function of φ, the azimuth, for $\omega = 10^3$, 10^4 and 10^5. Note the different behavior of the inclination, the nutation is increasing as ω increases. The precession gets smaller as ω increases, causing a slowly changing azimuth.

We remark that this is only an alternative way to approximate the Euler angles. The solution of these variables can be obtained more efficiently by the differential equation originating from the Euler-Lagrange equations in (26).

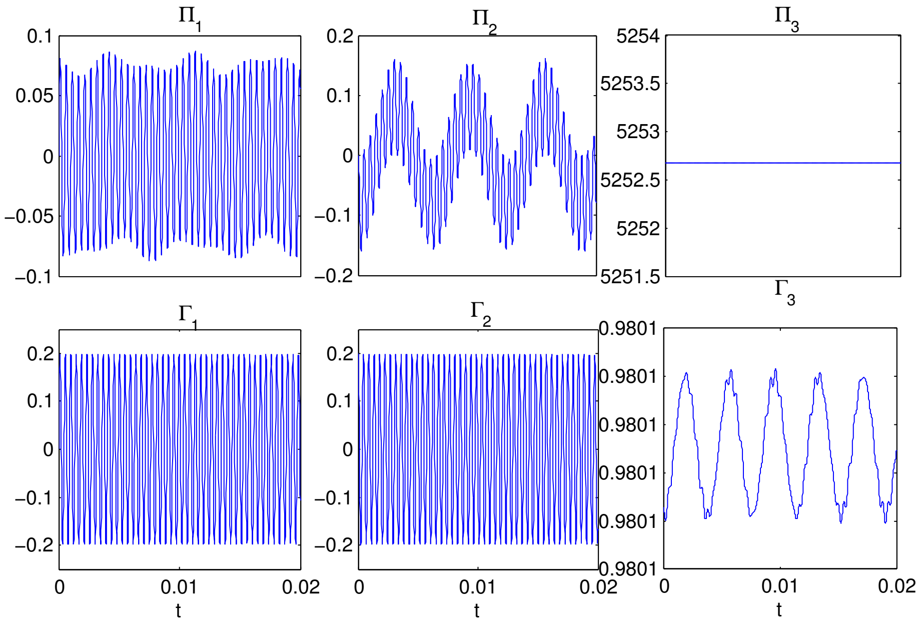


Figure 19: The solution of the spinning top subjected to an external force $a(t) = \omega\cos(\omega t)$ for $\omega = 10^3$.

6 Vibrational Forces on the spinning top

We want to extend the equations in (31) by adding an external force to the top. We let the force be time-dependent and directed in a direction perpendicular to the vertical axis of the top, making the motion unstable as we spin the top. This can be described by changing the differential equations for the angular momenta in (31) to

$$\begin{aligned}\dot{\Pi}_1 &= \frac{I_2 - I_3}{I_2 I_3}\Pi_2\Pi_3 + Mgl\Gamma_2 + a(t),\\ \dot{\Pi}_2 &= \frac{I_3 - I_1}{I_1 I_3}\Pi_1\Pi_3 - Mgl\Gamma_1,\\ \dot{\Pi}_3 &= 0,\end{aligned} \tag{38}$$

where $\dot{\mathbf{\Gamma}}$ remains as in (31). The force is described by $a(t)$ and is only affecting the derivative of Π_1, thus the force will be directed perpendicular to the vertical axis of the top.

The stable solution of the top is not affected by the external force unless the force is sufficiently large. To test this, we let $a(t)$ be a small oscillating force and solve the system in (38) using ODE45 with $\omega = 10^3$ (we use the initial conditions

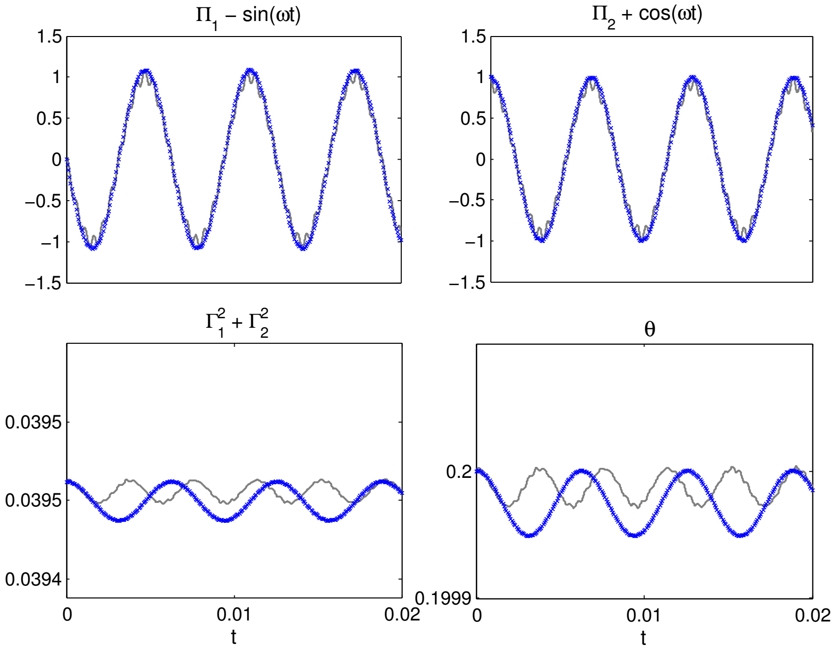


Figure 20: The HMM approximation ($\times$) of the slow variables for $\omega = 10^3$ plotted with the solution of the slow variables obtained by RK4 (gray line) for $T = 0.02$.

from the previous section). This leads to a solution almost identical to the solution in Figure 11. In fact, we find that the vibrational force needs to be very large in order to perturb the stable solution of the top (this has also been tested numerically by increasing $a(t)$). The stability of the top can however be overcome if we let

$$a(t) = \omega cos(\omega t), \tag{39}$$

which leads to leads to a significant change to the solution of (38). Figure 19 shows the approximated solution of $\mathbf{\Pi}$ and $\mathbf{\Gamma}$ for $\omega = 10^3$ with the updated system in (38) using Matlab's ODE45. We observe that the solution of Π_1, Π_2, Γ_1 and Γ_2 oscillate fast, but the solution of Π_2 seems to exhibit slowly varying constituents in addition to the fast oscillating behavior.

Slow Variables. The impact of the force $a(t)$ in (39) leads to a disturbed solution of the spinning top. In Figure 19 we observed an underlying slowly changing nature of Π_2 and we would expect that the system in (38) exhibits some slow behavior.

The choice of slow variables from the previous section ((32) and (37)) are not slow constituents of the new system in (38), and we need to define a new set of slow variables by considering the effect of the external force. A new set of slow

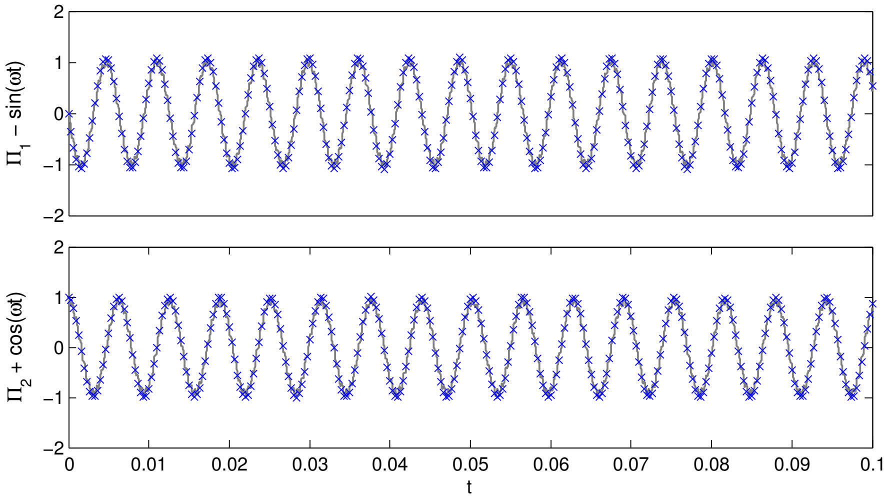


Figure 21: The long time HMM approximation of ξ_2 and ξ_3 for $\omega = 10^3$.

variables can be found by analyzing the equations of motions (38) and finding the slow constituents numerically. We find the following set of slowly changing variables:

$$\begin{aligned}
\xi_1 &= \Pi_3\\
\xi_2 &= \Pi_1 - \sin(\omega t)\\
\xi_3 &= \Pi_2 + \cos(\omega t)\\
\xi_4 &= \Gamma_1^2 + \Gamma_2^2\\
\xi_5 &= \theta = \cos^{-1}(\Gamma_3).
\end{aligned} \tag{40}$$

We applied Algorithm 4 and found the numerical approximation of the slow variables using HMM. Figure 20 shows the solution with $T = 0.02$, $H = T/300$, $h = 10^{-7}$ and $\eta = 5h$. The HMM solution is compared to the numerical solution found by using RK4 with step size 10^{-7} as Matlab's ODE45 failed to give accurate solutions of the variable ξ_4 (even with the error tolerance set to 10^{-17}). The solution of the slow variables ξ_2 and ξ_3 using HMM seems to be very good approximations (this is also illustrated in Figure 21), but the inclination θ and $\xi_4 = \Gamma_1^2 + \Gamma_2^2$ are not approximated accurately. This error is not overcome as we increase η, or decrease h and H. We remark that this error is of order 10^{-4}, and may be a result of the aggregated error from the microsimulation and estimation of the force $F(\boldsymbol{\xi})$.

Despite the error in the solution of ξ_4 and ξ_5, the variables ξ_2 and ξ_3 are approximated very well. Figure 21 shows the long time solution of the HMM approximation, compared to the solution using ODE45 (the evaluation of ξ_2 and

ξ_3 using RK4 takes too long with $\omega = 10^3$). With $T = 0.1$, $H = T/1500$, $h = 10^{-7}$ and $\eta = 5h$, the HMM only requires a total of 10^4 steps to get an accurate solution of ξ_4 and ξ_5. Solving the system with RK4 is computationally demanding as we need a step size of order 10^{-7}, leading to a total of 10^6 operations to evaluate the solution for the time $T = 0.1$.

7 Averaged Equations of the Spinning Top Subjected to Vertical Vibration

We shall consider the slowly spinning top with initial spin $\Pi_3 = 10\hat{\Pi}_3$. This means that the top is no longer prevented from falling down. We learned from section 2.4 that the inverted pendulum is stabilized as we expose the pendulum to a highly oscillatory vertical vibration. Thus, we would intuitively expect the same behavior as we apply a similar kind of force on the spinning top.

The vibrating force results in the following equations of the spinning top:

$$\begin{aligned} \dot{\mathbf{\Pi}} &= \mathbf{\Pi} \times \mathbf{\Omega} + M(g + a(t))l\mathbf{\Gamma} \times \boldsymbol{\chi} \\ \dot{\mathbf{\Gamma}} &= \mathbf{\Gamma} \times \mathbf{\Omega}, \end{aligned} \tag{41}$$

where $a(t) = A\omega\cos(\omega t)$. This system exhibit a solution that is a superposition of a slowly varying solution, also called the *averaged solution*, and rapid oscillations of frequency ω (see Figure 22 and 23). The solution of $\mathbf{\Pi}$ has fast oscillations with amplitudes of order 1, while the solutions of $\mathbf{\Gamma}$ has amplitudes of order ω^{-1}. Inspired by the analysis done by Sanz-Serna in [4], we want to present the spinning top equations influenced by an external force, as an averaged equation.

For further analysis, we consider the spinning top equations (41) written in the format

$$\begin{aligned} \dot{\boldsymbol{\pi}} &= f_1(\boldsymbol{\pi}, \boldsymbol{\gamma}) + h(\boldsymbol{\gamma})\omega e^{i\omega t} + \mathrm{cc} \\ \dot{\boldsymbol{\gamma}} &= f_2(\boldsymbol{\pi}, \boldsymbol{\gamma}) \end{aligned} \tag{42}$$

where $\boldsymbol{\pi}$ and $\boldsymbol{\gamma}$ are the exact solutions. The notion cc means the "complex conjugate of the preceding term".

The corresponding *modulated Fourier expansion* of (42) is given as

$$\begin{aligned} \boldsymbol{\pi}(t) &= \mathbf{\Pi}(t) + \sum_{0<|k|<N} e^{ik\omega t}\boldsymbol{z}^{(k)}(t), \\ \boldsymbol{\gamma}(t) &= \mathbf{\Gamma}(t) + \sum_{0<|k|<N} e^{ik\omega t}\boldsymbol{x}^{(k)}(t), \end{aligned} \tag{43}$$

where $\mathbf{\Pi}$ and $\mathbf{\Gamma}$ are *averaged variables* of order 1. The coefficients $\boldsymbol{z}^{(k)}$ and $\boldsymbol{x}^{(k)}$ are called the *modulation functions* and are, in contrast to standard Fourier expansion, time-dependent. An important property of $\mathbf{\Pi}$, $\mathbf{\Gamma}$ and the modulation functions are the fact that they, along with all their derivatives, are bounded independently of ω. Moreover, the modulation functions are determined to be $\boldsymbol{z}^{(k)} = \mathcal{O}(\omega^{-k+1})$ and $\boldsymbol{x}^{(k)} = \mathcal{O}(\omega^{-k})$, for $|k| = 1, 2, \ldots$. (This can be verified by observing the behavior of the solution in Figure 23 and by examination of the expression in (42)). This

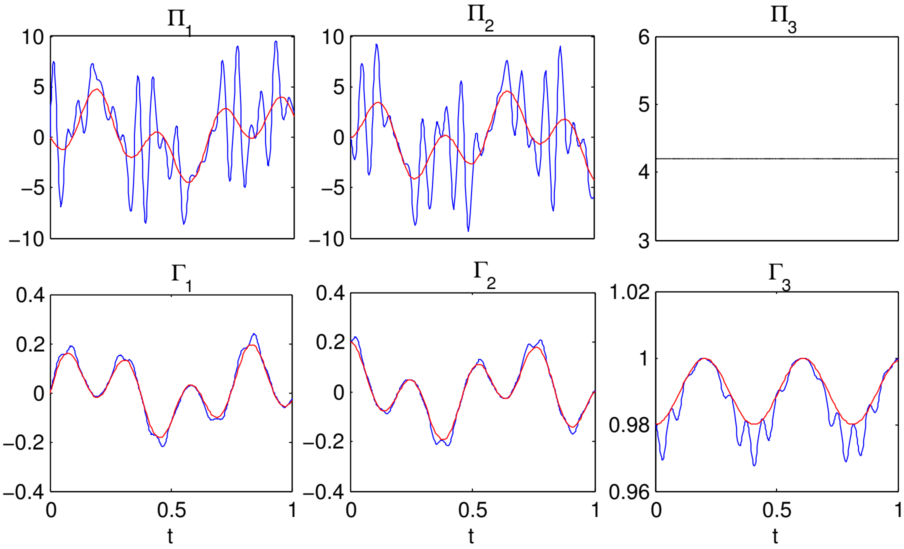


Figure 22: The solution of the averaged system in (48) (red line) plotted with the exact solution (blue line) for $\omega = 10^2$, $A = 10$.

concludes that the only oscillating terms in the Fourier expansions (43) are the exponential functions $e^{ik\omega t}$, $|k| = 1, 2, \ldots$.

The first terms of the modulated Fourier expansion (43) are

$$\begin{aligned} \boldsymbol{\pi}(t) &= \boldsymbol{\Pi}(t) + \boldsymbol{z}^{(1)}(t)e^{ik\omega t} + \mathrm{cc} + \boldsymbol{z}^{(2)}(t)e^{ik\omega t} + \mathrm{cc} + \cdots\,, \\ \boldsymbol{\gamma}(t) &= \boldsymbol{\Gamma}(t) + \boldsymbol{x}^{(1)}(t)e^{ik\omega t} + \mathrm{cc} + \boldsymbol{x}^{(2)}(t)e^{ik\omega t} + \mathrm{cc} + \cdots\,. \end{aligned} \tag{44}$$

In order to find an averaged solution of (44) we need to derive the differential equation for the averaged variables $\boldsymbol{\Pi}$ and $\boldsymbol{\Gamma}$. This can be done by recursively determine the modulation functions of the modulated Fourier expansion (43). As in the analyzis of Sanz-Serna, [4], we do this step-by-step, starting with the coefficients $\boldsymbol{z}^{(1)}$ and $\boldsymbol{x}^{(1)}$, respectively of order 1 and ω^{-1}.

First step. We compare the ansatz in (44) to the first equation in (42) and get

$$i\omega \boldsymbol{z}^{(1)} e^{i\omega t} = h(\boldsymbol{\Gamma})\omega e^{i\omega t} + \mathcal{O}(1).$$

Thus, the first modulation function may be written as

$$\boldsymbol{z}^{(1)} = ia \begin{bmatrix} -\Gamma_2 \\ \Gamma_1 \\ 0 \end{bmatrix} + \mathcal{O}(\omega^{-1}), \tag{45}$$

where a is the constant $MAl/2$. This shows that the components with frequency ω and amplitude $\mathcal{O}(1)$ of the angular momentum $\boldsymbol{\pi}$ is dependent on the variation

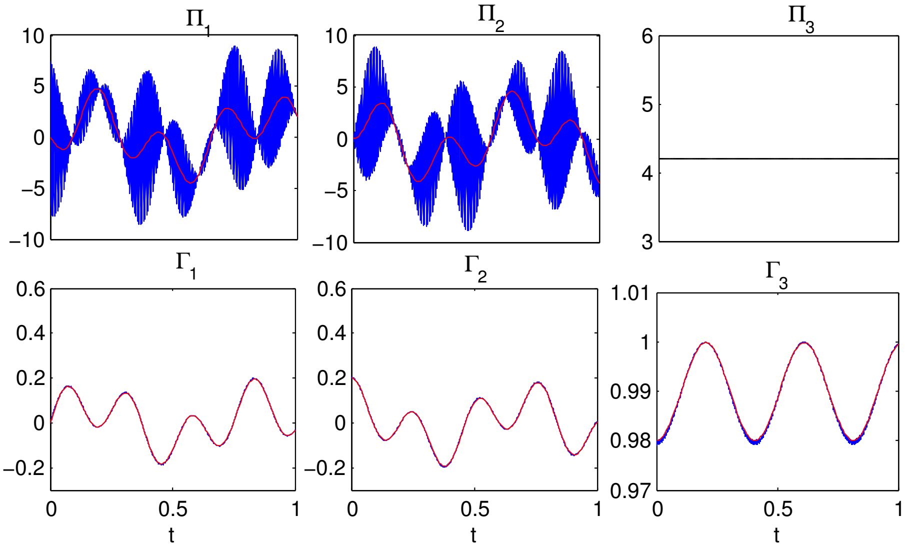


Figure 23: The solution of the averaged system in (48) (red line) plotted with the exact solution (blue line) for $\omega = 10^3$, $A = 10$.

of Γ_1 and Γ_2.

Second step. We now use (45) in the second equation of (42), which gives

$$\begin{aligned}\dot{\mathbf{\Gamma}} + i\omega \boldsymbol{x}^{(1)} e^{i\omega t} &= f_2(\boldsymbol{\pi}, \boldsymbol{\gamma}) + \mathcal{O}(\omega^{-1}) \\ &= f_2(\mathbf{\Pi} + \boldsymbol{z}^{(1)} e^{ik\omega t} + \mathrm{cc}, \mathbf{\Gamma}) + \mathcal{O}(\omega^{-1}).\end{aligned}$$

By equating the terms involving $e^{i\omega t}$, the leading order equations for $\boldsymbol{x}^{(1)}$ reads

$$\boldsymbol{x}^{(1)} = \omega^{-1} a \begin{bmatrix} -\Gamma_1\Gamma_3/I_2 \\ -\Gamma_2\Gamma_3/I_1 \\ \Gamma_1^2/I_2 - \Gamma_2^2/I_1 \end{bmatrix} + \mathcal{O}(\omega^{-2}). \tag{46}$$

Thus the components with frequency ω of $\mathbf{\Gamma}$, are only dependent on $\mathbf{\Gamma}$ itself. We have now obtained an $\mathcal{O}(\omega^{-1})$ estimate of $\boldsymbol{\pi}$, and an $\mathcal{O}(\omega^{-2})$ estimate of $\boldsymbol{\gamma}$, leading us to the new misson; finding the $\mathcal{O}(\omega^{-1})$ components of $\boldsymbol{\pi}$.

Third step. In order to find the modulated Fourier expansion up to order ω^{-2}, we proceed by again considering the first equation in (42). By substitution of the ansatz (44), applying the modulation functions (45) and (46), we get

$$f_1(\boldsymbol{\pi}, \boldsymbol{\gamma}) + h(\boldsymbol{\gamma})\omega e^{i\omega t} + \mathrm{cc} = f_1(\mathbf{\Pi} + \boldsymbol{z}^{(1)} e^{ik\omega t} + \mathrm{cc}, \mathbf{\Gamma} + \boldsymbol{x}^{(1)} e^{ik\omega t} + \mathrm{cc}) + \mathcal{O}(\omega^{-1}).$$

This should now be compared to the ansatz

$$\dot{\mathbf{\Pi}} + i\omega \boldsymbol{z}^{(1)} e^{i\omega t} + \text{cc} + \boldsymbol{z'}^{(1)} e^{i\omega t} + cc + 2i\omega \boldsymbol{z}^{(2)} e^{i2\omega t} + \mathcal{O}(\omega^{-1}),$$

which leads us to an $\mathcal{O}(\omega^{-2})$ estimate of the solutions of (42),

$$\begin{aligned}\boldsymbol{\pi} &= \mathbf{\Pi} + \begin{bmatrix} -ia\Gamma_2 + \omega^{-1}a(\Gamma_1\Pi_3/I_3 - \Gamma_3\Pi_1/I_1) \\ ia\Gamma_1 + \omega^{-1}a(\Gamma_2\Pi_3/I_3 - \Gamma_3\Pi_2/I_2) \\ 0 \end{bmatrix} e^{i\omega t} + \text{cc} \\ &\quad + \begin{bmatrix} \frac{i}{2}\omega^{-1}a^2\Gamma_2\Gamma_3/I_1 \\ \frac{i}{2}\omega^{-1}a^2\Gamma_1\Gamma_3/I_2 \\ 0 \end{bmatrix} e^{2i\omega t} + \text{cc} + \mathcal{O}(\omega^{-2}) \\ \boldsymbol{\gamma} &= \mathbf{\Gamma} + \begin{bmatrix} -\omega^{-1}a\Gamma_1\Gamma_3/I_2 \\ -\omega^{-1}a\Gamma_2\Gamma_3/I_1) \\ \omega^{-1}a(\Gamma_1^2/I_2 - \Gamma_2^2/I_1) \end{bmatrix} e^{i\omega t} + \text{cc} + \mathcal{O}(\omega^{-2}).\end{aligned} \tag{47}$$

This process may be repeated in order to find the modulation functions up to any desired order of ω^{-1}. Further calculations of the modulation functions will not be pursued here, but we will rather use these calculations to find the differential equations for the slowly varying averaged variables $\mathbf{\Pi}$ and $\mathbf{\Gamma}$.

Averaged differential equations. We now have enough information about the modulation functions to derive an averaged equation of the system in (42). This is done by equating the ansatz (47) to the first equation in (44), and then retain only the non-oscillatory terms of the derivatives $d\boldsymbol{\pi}/dt$ and $d\boldsymbol{\gamma}/dt$, i.e:

$$\begin{aligned}\dot{\boldsymbol{\pi}} &= f_1(\mathbf{\Pi}, \mathbf{\Gamma}) + \begin{bmatrix} -2a^2(\Gamma_2\Gamma_3/I_1) \\ 2a^2(\Gamma_1\Gamma_3/I_2) \\ 0 \end{bmatrix} + (\cdots)e^{i\omega t} + \text{cc} + (\cdots)e^{2i\omega t} + \text{cc} + \cdots \\ \dot{\boldsymbol{\gamma}} &= f_2(\mathbf{\Pi}, \mathbf{\Gamma}) + (\cdots)e^{i\omega t} + \text{cc} + (\cdots)e^{2i\omega t} + \text{cc} + \cdots .\end{aligned}$$

This expression of $\dot{\boldsymbol{\pi}}$ is compared to the expansion in (44), which determines the average equations at leading order to be

$$\begin{aligned}\dot{\mathbf{\Pi}} &= f_1(\mathbf{\Pi}, \mathbf{\Gamma}) + \begin{bmatrix} -2a^2(\Gamma_2\Gamma_3/I_1) \\ 2a^2(\Gamma_1\Gamma_3/I_2) \\ 0 \end{bmatrix} + \mathcal{O}(\omega^{-2}) \\ \dot{\mathbf{\Gamma}} &= f_2(\mathbf{\Pi}, \mathbf{\Gamma}) + \mathcal{O}(\omega^{-2}).\end{aligned}$$

By insertion of the constant a from equation (42), we get

$$\begin{aligned}\dot{\mathbf{\Pi}} &= \mathbf{\Gamma} \times I^{-1}\mathbf{\Pi} \ + \ Mgl\mathbf{\Gamma} \times \boldsymbol{\chi} \ + \ \begin{bmatrix} -MA/2(\Gamma_2\Gamma_3/I_1) \\ MA/2(\Gamma_1\Gamma_3/I_2) \\ 0 \end{bmatrix} + \mathcal{O}(\omega^{-2}) \\ \dot{\mathbf{\Gamma}} &= \mathbf{\Gamma} \times I^{-1}\mathbf{\Pi} \ + \ \mathcal{O}(\omega^{-2}).\end{aligned} \tag{48}$$

Figure 22 and 22 shows the solution for the averaged system with $\omega = 10^2$ and $\omega = 10^3$. The initial values were $\mathbf{\Gamma}^0 = (0, \sin(0.2), \cos(0.2))^T$, $\mathbf{\Pi}^0 = (0, 0, 10\hat{\Pi}_3)^T$. The parameters used in the evaluation was $M = 5$, $I = (3.12, 3.12, 0.24)$ and $l = 0.74$.

8 Conclusions

We have studied the properties of the heterogeneous multiscale method for stiff ordinary differential equations. HMM allows us to move between the macroscopic and microscopic models, and to take advantage of scale separation in problems for improving efficiency. The main idea is to integrate an averaged equation for slow constituents of the problem, eliminating the need of a full solution on the fine scale grid. In this way HMM is able to use larger time steps and reduce the computational complexity compared to standard solvers.

We found however, that a relation between the micro and macro-variables needs to be established, for instance in the inverted pendulum example. The algorithm proposed in [2] bypasses the problem of relating the microscopic variables to the macroscopic variables by utilizing a set of slow variables.

HMM was successfully applied to the fast spinning top by extracting the slow constituents of the system. It turned out that the top had a slowly varying nature in addition to its highly oscillatory components. Even though this slow behavior was dependent on the fast spin ω, the HMM approximated the solution efficiently compared to standard solvers. The macroscale behavior was captured well by the HMM and the method showed a significant gain in computational complexity.

For the problems investigated here, the HMM correctly approximates variables that are slow with respect to the systems dynamics. However, the method requires us to define a set of slow variables, and that these are the variables we want to approximate. The fast varying variables are not approximated in the evaluation.

Finally, we found an averaged equation of the spinning top subjected to a vertical force. The averaged equation was found by an analysis using the Modulated Fourier expansion, and approximated the averaged behavior accurately.

References

[1] E. Weinan and B. Engquist, “The heterogeneous multiscale methods,” *Commun. Math. Sci.*, vol. 1, no. 1, pp. 87–132, 2003.

[2] G. Ariel, B. Engquist, and R. Tsai, “A multiscale method for highly oscillatory ordinary differential equations with resonance,” *Math. Comp.*, vol. 78, no. 266, pp. 929–956, 2009.

[3] B. Engquist and Y.-H. Tsai, “Heterogeneous multiscale methods for stiff ordinary differential equations,” *Math. Comp.*, vol. 74, no. 252, pp. 1707–1742, 2005.

[4] J. M. Sanz-Serna, “Modulated Fourier expansions and heterogeneous multiscale methods,” *IMA J. Numer. Anal.*, vol. 29, no. 3, pp. 595–605, 2009.

[5] M. Fredbo, “The heterogeneous multiscale method for stiff ordinary differential equations.” 12 2010.

[6] E. Weinan, B. Engquist, X. Li, W. Ren, and E. Vanden-Eijnden, “Heterogeneous multiscale methods: a review,” *Commun. Comput. Phys.*, vol. 2, no. 3, pp. 367–450, 2007.

[7] E. Tadmor, M. Ortiz, and R. Phillips, “Quasicontinuum analysis of defects in crystals,” *Phil. Mag. A*, vol. 73, no. 6, pp. 1529–1564, 1996.

[8] R. Sharp, Y.-H. Tsai, and B. Engquist, “Multiple time scale numerical methods for the inverted pendulum problem,” in *Multiscale methods in science and engineering*, vol. 44 of *Lect. Notes Comput. Sci. Eng.*, pp. 241–261, Berlin: Springer, 2005.

[9] G. Ariel, B. Engquist, and R. Tsai, “A reversible multiscale integration method,” *Commun. Math. Sci.*, vol. 7, no. 3, pp. 595–610, 2009.

[10] G. Ariel, B. Engquist, and R. Tsai, “Numerical multiscale methods for coupled oscillators,” *Multiscale Model. Simul.*, vol. 7, no. 3, pp. 1387–1404, 2009.

[11] E. Weinan, “Analysis of the heterogeneous multiscale method for ordinary differential equations,” *Commun. Math. Sci.*, vol. 1, no. 3, pp. 423–436, 2003.

[12] E. Hairer, C. Lubich, and G. Wanner, *Geometric numerical integration*, vol. 31 of *Springer Series in Computational Mathematics*. Berlin: Springer-Verlag, second ed., 2006. Structure-preserving algorithms for ordinary differential equations.

[13] J. E. Marsden and T. S. Ratiu, *Introduction to mechanics and symmetry*, vol. 17 of *Texts in Applied Mathematics*. New York: Springer-Verlag, second ed., 1999. A basic exposition of classical mechanical systems.

[14] V. Arnold, *Mathematical methods of classical mechanics*, vol. 60. Springer, 1989.