Interner Bericht

DLR-IB-AS-BS-2024-6

Overview of research activities during the secondment at the US Air Force Academy in Colorado Springs, October 2022 until July 2023

Institutsbericht

Markus Widhalm

Deutsches Zentrum für Luft- und Raumfahrt

Institut für Aerodynamik und Strömungstechnik

Deutsches Zentrum für Luft- und Raumfahrt

Bericht des Instituts für Aerodynamik und Strömungstechnik
Report of the Institute of Aerodynamics and Flow Technology

DLR-IB-AS-BS-2024-6

Overview of research activities during the secondment at the US Air Force Academy in Colorado Springs, USA, October 2022 until July 2023

Markus Widhalm

Herausgeber:
Deutsches Zentrum für Luft- und Raumfahrt e.V.
Institut für Aerodynamik und Strömungstechnik
Lilienthalplatz 7, 38108 Braunschweig

ISSN 1614-7790

Stufe der Zugänglichkeit: 1
Braunschweig, im Januar 2024

Institutsdirektor: Dr.-Ing. O. Brodersen Prof. Dr.Dr. hab. A. Dillmann	Verfasser: Dr.-Ing. Markus Widhalm

Abteilung: $C^2A^2S^2E$
Abteilungsleiter:
Prof. Dr. S. Görtz
Dr.-Ing. C. Grabe

Der Bericht enthält:

50	Seiten
8	Bilder
5	Tabellen
45	Literaturstellen

DLR-IB-AS-BS-2024-6

Overview of research activities during the secondment at the US Air Force Academy in Colorado Springs, USA, October 2022 until July 2023

Markus Widhalm

Verteiler:

Institutsbibliothek AS	1	Exemplar
Verfasser/Co-Autoren	1	Exemplar
Institutsleitung	1	Exemplar
Abteilungsleiter	1	Exemplar
Deutsche Bibliothek in Frankfurt/Main	2	Exemplare
Niedersächsische Landesbibliothek Hannover	1	Exemplar
Techn. Informationsbibliothek Hannover	1	Exemplar
Zentralbibliothek BS	1	Exemplare
Zentralarchiv GÖ	1	Exemplar

Abstract

This report is dedicated to research activities conducted during the secondment at the US-Air Force Academy (USAFA) in Colorado Springs from end of October 2022 until end of July 2024. The content includes a brief description of the three published contributions to the AIAA Aviation Forum in San Diego, which took place in summer 2023. Also listed are the additional research that have arisen outside the common activities with the USAFA, namely bio-inspired Swarm Intelligence optimisation strategies and the treatment and implementation of the geometrically exact beam. Finally, there is a summary about the topics and an evaluation of the stay at the USAFA.

Contents

Acronyms

ACO	Ant Colony Optimisation
AIAA	American Institute of Aeronautics and Astronautics
CFD	Computational Fluid Dynamics
CRM	Common Research Model
DLR	Deutsches Zentrum für Luft- und Raumfahrt
GA	Genetic Algorithm
GEB	Geometrically Exact Beam
GMI	Grassmann Manifold Interpolation
GSO	Glowworm Swarm Optimisation
IGEB	Intrinsic Geometrically Exact Beam
LFD	Linear Frequency Domain
ODE	Ordinary Differential Equation
oLAF	Optimally Load-adaptive Aircraft
PDE	Partial Differential Equation
PSO	Particle Swarm Optimisation
ROM	Reduced Order Model
SCPW	Stability and Control Prediction Workshop
SI	Swarm Intelligence
SMARTy	Surrogate Modeling for Aero-Data Toolbox
TAU	Unstructured Hybrid Flow Solver
URANS	Unsteady Reynolds Averaged Navier-Stokes
USAFA	US Air Force Academy

1 Introduction

Every year, the Deutsches Zentrum für Luft- und Raumfahrt (DLR) offers the opportunity to apply for an Otto Lilienthal research semester which was awarded to me in the year 2021. After a period of discovery, the US Air Force Academy (USAFA) in Colorado Springs, USA, was chosen for the secondment. The connection for a successful cooperation was provided by the NATO AVT-351 group, which deals with numerical methods of reduced order models, and in which the USAFA and the DLR have a leading role. A topic that fits well with current activities embedded in the Optimally Load-adaptive Aircraft (oLAF) project. After administrative preparations for the secondment to the USA, the stay at USAFA started on October 24, 2022 and ended nine months later on July 24, 2023. American Institute of Aeronautics and Astronautics (AIAA) hosts biannual conferences that are held in the U.S., and the time frame of the secondment made it an excellent opportunity to submit papers for the 2023 summer conference in San Diego. This is also a great opportunity to document the joint work with USAFA and all work that has been completed during this secondment.

One of the main reasons for the secondment was to strengthen the long-term collaboration between the USAFA and DLR. The common topic was already narrowed down in advance and shortly after arrival at the USAFA the content was fixed with an abstract for the AIAA Aviation Forum 2023. The abstract included a comparison of both numerical codes used at USAFA (Kestrel) and DLR (TAU), for the DLR-F22 ONERA wind tunnel model, a combat aircraft configuration. Furthermore, different Reduced Order Model (ROM) approaches for the evaluation of stability derivatives were also considered.

Another component during the stay was to complete the final report as a publication for DLRs internal project oLAF. Here, too, an abstract was submitted to AIAA Aviation Forum 2023 with results from the efficient high-accuracy loads package. Both of the mentioned earlier publications were submitted by taking the role as the lead author.

The third publication was a follow-up to the first Stability and Control Prediction Workshop (SCPW) also organized by AIAA and held in early 2021. The purpose was to conduct numerical simulations and investigations on a sub-problem of the original problem of the first SCPW to improve the predictive accuracy of dynamic derivatives and then to integrate this into best practice guidelines. The main-lead for this paper had a DLR colleague from the AS-Institute. My contribution was related to the preparation and classification of the numerical simulations and their results.

In addition, the secondment also provided time to focus on other topics. An overview of the work performed is given in chapter 3, which covers Swarm Intelligence (SI) optimisation and the simulation of a Geometrically Exact Beam.

This report contains the following chapters, the main activities chapter 2, these are the three publications presented at the AIAA Aviation Forum 2023 in San Diego, the additional research in chapter 3 conducted at the USAFA, and finally a conclusion in chapter 4 about the secondment to the USAFA in Colorado Springs, USA.

2 Main Activities

While at USAFA, three publications were submitted to the AIAA 2023 Aviation Forum summer conference. Two publications were authored as lead author and a third by a colleague from another department of DLR-AS. The first paper serves as a documentation for the excellent cooperation in the field of simulations of unsteady flows with the USAFA as well as the application of ROM's for the applicability of high agile aircraft. The second contribution is a paper on the work done in the DLR internal project oLAF, in which I was involved at the time of the secondment. The third paper was an add on to the first SCPW held by the AIAA. Finally, all three papers were presented at the 2023 AIAA Summer Conference in San Diego.

2.1 Comparison of Reduced Order Models for Evaluating Stability Derivatives for the DLR-F22 ONERA model

This paper [44] is about the comparison of reduced order models to predict aerodynamic stability derivatives. The study involved two widely used frameworks for simulating unsteady flows, DoD HPCMP CREATERM-AV/Kestrel and the DLR Unstructured Hybrid Flow Solver (TAU). The DLR TAU contains a linearised version of the discrete unsteady Reynolds-averaged Navier-Stokes equations based on the small perturbation approach which are solved in the frequency domain. This allows a comparison between distinctly different approaches of reduced order aerodynamic modelling. Two reduced order models are applied to unsteady simulations to extract stability derivatives from predicted time histories of the force and moment coefficients. A third approach is based on the linear frequency domain solver which computes stability derivatives directly. In the case of unsteady simulations, a system identification approach is applied to the time history of the force and moment coefficients for different training manoeuvres to extract the stability derivatives. The second reduced order model is based on the calculated indicial responses to unit step changes in the angle of attack and pitch rate. The weaknesses and strengths of the individual approaches are shown and, in particular, the efficiency of the methods is outlined. The use case is the DLR-F22 ONERA model, a generic research wind tunnel model of a triple delta wing fighter type aircraft configuration, at transonic flow conditions for various angles of attack.

The paper [44] includes the following outline, which are Numerical flow solvers, applied ROM approaches, Test Case Description, Comparison of flow simulations, Dynamic response data, and Evaluation of Static and Dynamic Derivatives. The ROM approaches involves the time-linearised approach (Linear Frequency Domain (LFD)), the Stability Derivative Method, Indicial Response Modelling, and System Identification.

This investigation covers various reduced order models applied for predicting stability derivatives with two well established Unsteady Reynolds Averaged Navier-Stokes

(URANS) flow solvers, Kestrel and TAU. In addition, the TAU LFD method has been introduced to show the extent whereto it is possible to use this method for vortex-dominated flow applications. Both URANS methods, Kestrel and TAU, predict stability derivatives in general in good agreement to each other, however certain deviations are found since the prediction of the flow physics shows differences and thus the dynamic response data differs. Regarding the ROM approaches applied on the dynamic response data of the unsteady simulations, the trend at lower angles of attack range is well preserved. At and above the 20 deg case predictions between the URANS methods are difficult to judge because deviations in the global force and moment coefficients become evident and therefore a better insight in predicting flow features like shock location, separation and vortical structure would be necessary. Vortex-dominated flows are mainly unsteady occurrences, and this transonic case surely represents a special case, since steady-state solutions are possible, a prerequisite for the application of the TAU LFD method. However, one LFD simulation in comparison with URANS boosts the time reduction factor above two orders of magnitude. Because of its good agreement for the smaller angle of attack range, the LFD is well suited as a prediction tool. A moderate range of angles of attack up to twenty degrees can be predicted with the LFD in less than a day, providing exactly this advantage of a fast response time. An URANS simulation takes about a week for a single angle of attack and a single motion, and it requires a good amount of experience to get the flow solver parameters right to get a usable result the first time. All of this is eliminated with the LFD method. It has to be stated that the current LFD linearisation can be further improved by the linearisation of the SA turbulence model with rotational correction, actually a must for consistency with the discretisation of the governing equations of the URANS equations using the SA turbulence model with rotational correction. Indicial responses were generated using Computational Fluid Dynamics (CFD) and a grid motion approach. These responses show an initial peak, a transient behaviour before asymptotically reaching steady-state values. These functions could be used to generate a non-linear unsteady aerodynamic model. However, in this study, the indicial functions were used to estimate the stability derivatives from the steady-state values and transient behaviour. These derivatives were compared with LFD and classical methods and a reasonably good agreement was found. Finally, a system identification method was used to estimate stability derivatives of the model from two different training signals. The Schroeder motion was generated using a optimal criterion of the input parameter space, however, the model covers a wide range of angles of attack. The second motion is a piece-wise chirp signal, that is its instantaneous frequency is time-dependent. The aerodynamic model is a polynomial with input parameters of angle of attack, time-rate of change in the angle of attack, and pitch rate. The model was then used to predict static aerodynamic data and stability derivatives. The model based on the piece-wise chirp predicts the static data better than the model based on the Schroeder motion, in particular at high tested angles of attack. For stability derivatives, the model based on Schroeder predicts smooth curves for the stability derivatives though the values are similar to evaluated data from other methods. The model based on piece-wise chirp shows more changes in the prediction plots, however, the values seem to be off for some derivatives at some angles.

The full text of the manuscript is available on the AIAA internet page at the following link doi:10.2514/6.2023-4199.

2.2 Nonlinear Low-Dimensional Model Order Reduction with Subspace Interpolation for Gust Applications with the Linear Frequency Domain Approach

Gust load predictions are time-consuming for aircraft design since these are unsteady events and depend not only on the flow conditions but also on many other parameters such as weight and altitude changes. The linear frequency domain solver allows a rapid calculation of gust loads in contrast to a time-accurate method, although the calculation of several harmonics is necessary to reconstruct the equivalent time signal. A further increase in efficiency can be obtained with the use of a Proper Orthogonal Decomposition-Galerkin method for interpolating certain frequencies for the gust time signal for a single operating point of the flight envelope. If several operating points are already known, an interpolation of gust loads between the operating points can be performed with the Grassmann manifold. Considering the flight conditions, subsonic to transonic, of a whole flight envelope, the Proper Orthogonal Decomposition-Galerkin method achieves a good prediction accuracy. For the Grassmann manifold, however, it turns out that it is only suitable in the immediate vicinity of sampled operating points.

The results of the Hauptarbeitspacket 4 (HAP) entitled 4.3.5 Efffiziente hochgenaue Lasten für MDO (Multi-Disciplinary Optimisation), in the project oLAF is described in detail in the manuscript [43]. It includes the following outline, Method formulation, POD Galerkin-Projection method (LFDROM), Grassmann Manifold Interpolation, Numerical Methodology, Results for the Common Research Model (CRM) wing/tail section, as well as Results for the CRM configuration, and Computational Efficiency.

Two subspace models, the POD-Galerkin projection and the Grassmann manifold interpolation technique were applied in conjunction with POD to obtain reduced order models for solutions of the linearized Navier-Stokes equations. While the first approach is well suited for interpolating individual solutions from a set of LFD solutions to reconstruct the gust loads of a single operating point, the second approach interpolates the POD modes of the total gust loads between operating points. The two approaches yield very low-dimensional systems of Ordinary Differential Equation (ODE) to describe the dynamics of coefficients that modify the POD basis functions. The coefficients were reduced-order variables that were used to reconstruct the entire flow field for a time-accurate solution. For subsonic flows, the time signal of the gust LFD solutions can be interpolated from the second to third frequency without major loss in accuracy by using the POD-Galerkin projection method (LFDROM). For transonic flows, interpolation of LFD solutions may start from the fourth or fifth frequency, which still results in a high efficiency. It should be noted; however, that the interpolation should not be applied to the first two to three reduced frequencies of the frequency spectrum, since these are the dominant frequencies and large errors occur. The reduced frequencies for a time signal also drop exponentially from the third to the fifth reduced frequency onwards which is another indication why it works well at higher frequencies. Applications for the LFDROM can pay off for gust simulations of a complete flight envelope, if only single missing frequencies for a gust signal have to be determined to reconstruct the complete gust signal. It is much faster than performing an additional LFD simulation. For the Grassmann Manifold Interpolation (GMI) it is currently only applicable in the near vicinity of sampling points for a gust evaluation, especially in the transonic flow region. Its merit in saving computational time lies in the reduction to number of snapshot times snapshot POD matrix, and that is an almost negligible time effort to reconstruct all LFD field solutions for a single gust encounter between operating points. Regarding the evaluation of the integral force and moment derivatives to reconstruct the gust time signal to obtain the load factor, currently both subspace ROM models implemented in Surrogate Modeling for Aero-Data Toolbox (SMARTy) are slowed down under the inherent construction of the flux Jacobian matrix in the TAU code. It remains to this day a question between memory space that is storing the Jacobian, or computational effort that is computing the Jacobian on the fly. It has been shown that GMI still needs improvements for the interpolation of gust signals, especially in the transonic range. Since the gust encounter analysis is a multidisciplinary process, the applications of GMI can be readily adopted to other disciplines such as structural deformations.

The full text of the manuscript is available on the AIAA internet page at the following link doi:10.2514/6.2023-3947.

2.3 DLR Results for the 1.5 AIAA Stability and Control Prediction Workshop Investigating a Wing and Wing-Tail Section of the Common Research Model

A summary about the DLR results for the Special Session of the AIAA Stability and Control Prediction Workshop regarding the wing section only and wing/tail section of the Common Research model (CRM) is presented. The first prediction workshop revealed a deficit in the accurate determination of the pitching moment coefficient of the Common Research wind tunnel model. Thus, the idea is to perform investigations on a wing only and wing/tail section to improve the prediction accuracy for the pitching moment, and in consequence pitching moment derivatives, to increase the prediction reliability for future workshops. The two-dimensional sections of the CRM wind tunnel model are used to calculate a transonic Mach number sweep for a given lift coefficient to explore mesh-independent solutions. Using aerodynamic coefficients and their partial and total derivatives, and determining the velocity vectors at a defined location provides insight into the different meshing strategies for both airfoils and the wing wake. In addition, DLR also computed the optional test cases by evaluating the lift coefficient range from zero to the maximum for the wing only and the wing/tail case, and evaluating the derivative of the pitching moment coefficient with respect to angle of attack and tail incidence angle for the wing/tail case. The analysis shows that the derivative prediction on the two-dimensional CRM case proves to be very sensitive regarding the grid quality, wing and tail incidence angle as well as the shape of the tail geometry.

The manuscripts [37] describes the Test Cases performed for the workshop, and includes the outline for the DLR-TAU flow solver, and the summary of the results for the CRM wing and wing/tail section.

This work covers the DLR results for the 2D Airfoil Study of the 1.5th Stability & Control prediction workshop. The test cases are a consequence of results from the first Stability & Control prediction workshop, where the numerical simulation results of the participants diverged for the prediction of stability derivatives for the CRM model, especially for the prediction of the pitching moment. This stripped down approach with the two-dimensional airfoil section of the CRM wind tunnel model of the wing and the tail proved, that the prediction of derivatives is very sensitive to a number of factors.

The grid requires a high quality on the wing and tail surface, on their respective shock locations and in the wake area between wing and tail. The compression shock on wing and tail for higher Mach numbers and the incidence angle of wing and tail need a sufficient resolution, in order to achieve acceptable results. Also, the pitching moment has shown to be very sensitive to changes in the angle of attack, the addition of the tail geometry and the tail incidence angle. Because of its lever arm, the tail is responsible for most of the pitching moment generation, therefore the exact numerical simulation of the aerodynamics of the tail is crucial for the pitching moment prediction. However, for an ideal prediction of the tail aerodynamics, the main wing aerodynamics need to be of the same high accuracy as well, since the tail is affected by the wake of the main wing. The counter pressure of the tail body in the wake of the main wing also influences the aerodynamics on the main wing, decreasing the suction peak at its leading edge and pushing the shock location upstream.

The studies conducted in this special session have shown, that even for a simplified two-dimensional case of the CRM wind tunnel model with only wing and tail, the numerical simulations must be very accurate to achieve a satisfying stability derivative prediction. Next steps will be the comparison of the results of all participants, the analysis in which magnitude and where the predictions agree and the deduction for the next steps leading to the 2nd AIAA Stability & Control prediction workshop.

The full text of the manuscript is available on the AIAA internet page at the following link doi:10.2514/6.2023-4397.

3 Additional Research

This chapter summaries work performed during the secondment which was not included in the main focus. These are topics that have little affinity with the author's usual research topics. However, the first topic in section 3.1 is related to optimisation techniques for finding the global optima based on meta-heuristic approaches, and the second topic dealt with the mathematical treatment for uniqueness in solving the set of equations for the geometrically exact beam in section 3.2.

3.1 Swarm Intelligence (SI) Algorithms

SI is one of the computational intelligence techniques which are used to solve complex problems. Historically, the phrase SI was coined by Beny and Wang in the context of cellular robotics [1]. SI involves collective study of the individuals behaviour of population interact with one another locally and with their environment. Simple rules are followed by agents and no centralized control structure exists in order to predict the behaviour of individual agents. The random iteration of a certain degree between the agents provides an "intelligent" behaviour which is unknown to individual agents. Millonas [29] proposed that SI must satisfy five basic principles: adaptability, diverse response, stability, quality, and proximity.

Table 3.1: SI basic principles by Millonas [29].

Principle	Definition
Adaptability	Swarm should have the potential to change its search behaviour when the computational cost is high.
Diverse response	Swarm should not perform its activities along extremely narrow channels.
Stability	Swarm should not change its search behaviour in response to the environment changes.
Quality	Swarm should have the potential to respond to performance measures in the environment.
Proximity	Swarm should have the potential to smoothly perform time- and space-consuming computations.

Table 3.1 lists their meanings. The following algorithms are evolutionary computational techniques with a meta-heuristic approach. Especially for biological systems nature often acts as an inspiration to mimic the collective and social behaviour of living creatures. The behaviour of a solitary ant, bee, termite and wasp often is too simple, but their combined and social actions are of paramount consequence. Although there is normally no centralized control structure dictating how individual agents must behave, limited

interactions between such agents often lead to the emergence of global behaviour [17]. Many biological creatures such as fish shoaling and schooling [1] and bird flocks clearly display structural order, with the behaviour of the organisms so integrated that even though they may change shape and direction, they appear to move as a single coherent entity [5]. The main properties of the collective behaviour can be pointed out as follows and is summarized in fig. 3.1.

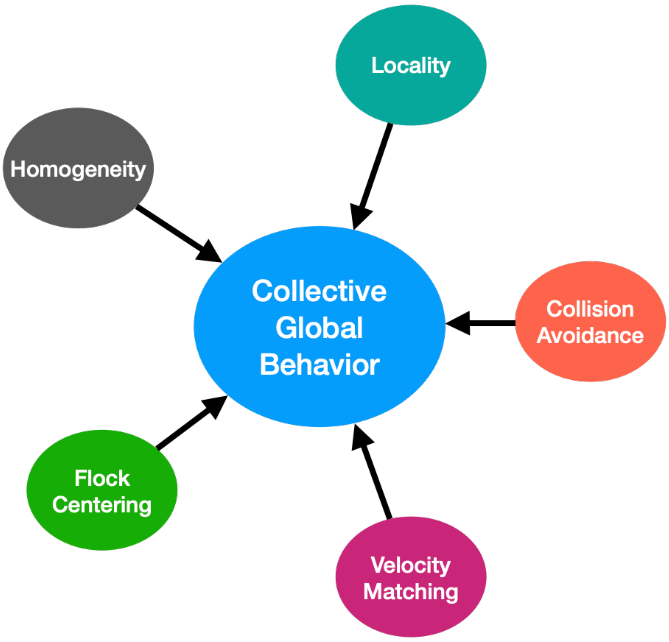


Figure 3.1: Main properties of collective behaviour.

Homogeneity is that every agent in flock has the same behavioural model. The flock moves without a leader, even if leaders emerge at times. Locality means that the nearest flock-mates just influence the motion of each agent. Vision is considered the most important sense for the organization of the flock. Collision avoidance is used to avoid collisions with nearby flock mates. The velocity matching tries to match the velocity of nearby flock mates. The flock centring involves trying to stay close to flock mates. Individuals attempt to maintain a minimum distance between themselves and others at all times. This rule has the highest priority and corresponds to a frequently observed behaviour of animals in nature [24]. If individuals do not perform an avoidance manoeuvre, they tend to attract other individuals (to avoid being isolated) and align themselves with neighbours [32, 31].

Currently, SI algorithms are very popular for optimization because they promise to find global optima, and can also be used efficiently in machine learning approaches. The differences in the algorithms are often only marginal, which explains the multitude of available SI algorithms. The uniqueness of the algorithms is not always evident from the biologically related names. Nevertheless, certain algorithms offer different approaches that are worth mentioning by name. One of the initial SI algorithms was Ant Colony Optimisation (ACO) [6, 3], introduced in 1992 by Marco Dorigo, a probabilistic technique for solving non-deterministic (discretised) problems which creates different

[1] In biology, when any group of fish swim together in a loose cluster for social reasons, this is typically called a shoal (shoaling). It can be a mix of different species. If the group of the same fish species is swimming in the same direction in a coordinated manner, they are schooling [38], thus, a school is a group of fish swimming together in synchrony; turning, twisting and forming sweeping, glinting shapes in the water.

solutions through a sequence of decisions. Another significant different algorithm is the Artificial Bee Colony (ABC) [18, 19] developed by Karaboğa and Akay which has been proposed for solving continuous and unconstrained optimisation problems and gained success on especially high dimensional and multi-modal unconstrained problems. And finally the Particle Swarm Optimisation (PSO) [20] established by Kennedy and Eberhart where the coordinates in the search space that are associated with the best solution is tracked by each particle and most noticeable it is performed randomly.

3.1.1 Ant Colony Optimisation (ACO)

ACO is inspired by the concept of the self-organisation of swarms and derived from swarm intelligence. This technique was first developed by Marco Dorigo [6, 3]. The fundamental idea is that ants organize themselves to travel to the food source and have the ability to follow each other.

The two important properties of ACO that basically simulate the real ant system are as follows [3]:

- **Stigmergy**: This is a property that plays an important role in developing a collective behaviour of the social insects. The stimulatory factor pheromone[2] trail is secreted from an ant, the amount of which decides the preference for the next ant to choose a path. This basically depicts the property called self-organization.
- **Autocatalysis**: According to this property, the shorter the path, the sooner the pheromone is deposited by the ants, and the more ants use the shorter path. This ensures the fact that the algorithm introduces the chance of rapid convergence while heading toward the optimal solution. The important property of this algorithm is the decaying of pheromone, which influences the convergence by governing the amount of accumulation of pheromone in the paths. It ensures that the search process does not get stuck in the local optima.

In essence, ACO mimics the foraging behaviour of social ants in a colony, and pheromone is used for simulating the local interactions and communications among ants. Pheromone is deposited by each ant and it evaporates gradually with time. The exact form of evaporation model may vary, depending on the variant and form of ACO used in implementations. Both incremental deposition and exponential decay of pheromone are widely used. ACO is particularly suitable for discrete optimization problems. For example, for routing problems, a route or path is encoded as a solution. When ants explore different paths, explored routes are marked with deposited pheromone that evaporates over time. The fitness or quality of a path (a solution) is related to the concentration of pheromone on the path. Routes with higher pheromone concentrations will be preferred or be chosen with a higher probability at a junction. Similar to Genetic Algorithm (GA), ACO is a mixed procedure with many variants and applications. ACO can be combined easily with other methods; it shows well performance in resolving complex optimization problems.

So far, ACO has been widely and successfully implemented for solving discrete optimization problems. It has been tried on both static and dynamic combinatorial optimization

[2] A pheromone is a secreted or excreted chemical factor that triggers a social response in members of the same species. Pheromones are chemicals capable of acting like hormones outside the body of the secreting individual, to affect the behaviour of the receiving individuals.

problems (Dorigo et al., 1999). A few examples of static optimization problem are given as follows:

- **Travelling salesman problem**: In this problem, n cities are travelled in such a way that the total travelling cost is minimized. ACO has shown better performance than the GA for a small problem (30-city problem), but not for a larger problem (Dorigo et al., 1997).

- **Quadratic assignment problem**: It is the problem of assigning n facilities in n locations so that the total cost of assignment is minimized. The results obtained for this problem is shown in Dorigo et al. (1999).

- **Job-shop scheduling problem**: A set J of jobs are to be assigned on M machines in such a fashion that the total completion time is minimal, satisfying the constraint that no two jobs can be processed on the same machine at the same time. ACO was able to find 10% of the optimal value of the results for a 15-machine, 15-job problem (Dorigo et al., 1999).

- **Vehicle routing problem**: This problem is about obtaining minimum-cost vehicle routes for a fleet of vehicles starting from a depot or more than one depot. Dorigo et al. (1999) applied ACO on this problem and obtained reasonable results.

- **Truss structures**: ACO in structural engineering was presented by Bland (2001) for design optimization of a 25-bar space truss. Kaveh and Shojaee (2007) employed ACO for optimal design of skeletal structures. A hybridization of ACO and other meta-heuristic techniques, like Particle Swarm Optimisation, have been applied for design optimization of steel frames and truss structures (Kaveh and Talatahari, 2007; Kaveh and Talatahari, 2009).

Some of the dynamic optimization problems where ACO is applied are the connection-oriented network routing problem and the connectionless network routing problem.

The principle of ACO can be explained using the Travelling Salesman Problem as follows using a simple example. Suppose that there are two different reversible paths, $\overleftrightarrow{ECADF}$ and $\overleftrightarrow{ECBDF}$ available to a group of ants. The ants can travel in either direction, with the objective of deriving the shortest path. The $\overleftrightarrow{CAD}$ leg is twice as long as the $\overleftrightarrow{CBD}$ leg. The underlying concept is that ants lay pheromone along the travelled path, which evaporates over time. Thus, the shorter a travelled leg, the longer the pheromone lasts. An ant travelling in the $\overrightarrow{ECDF}$ direction is faced with two options, $\overrightarrow{CBD}$ or $\overrightarrow{CAD}$, at point C. The decision of choosing a path over the other at point C is purely arbitrary and has equal probability. But the probability of choosing the shorter path $\overrightarrow{CBD}$ grows in time for the follower ants, as the pheromone trail left by preceding ants lasts longer on the shorter path. After sufficient time intervals, all ants converge to the shortest path.

Algorithm 1 outlines a high-level description of the ACO to the Travelling Salesman Problem (TSP). This algorithm was taken from the books [3] as well as [40, 41]. Three algorithms are available which are *PyACO.py* [40], *PyACO_1.py* is a high-level ACO-TSP program with various modifications of the original ACO of Dorigo, and there is also a Matlab version *MatLabACO.m*.

Essentially, then, the limitations of ACO are that it does not exploit a continuous search space as GA does. The continuous space must be discretised for ACO application, one of the inherent impediments in ACO applications.

Algorithm 1 Pseudo Code of ACO for Travelling Salesman Problem

```
Set the algorithmic parameters
$X, Y$ vertexes
$\alpha = 1.0$, $\beta = 1.0$, $\rho = 0.02$
antPop = 4, maxIterations = 10                                ▷ Initial ant population
$\tau_0 = 0.01$                                               ▷ Initial value of pheromone
$Q = 1$                                                       ▷ Constant
procedure ACO( $X, Y$, $\alpha$, $\beta$, $\rho$, antPop, maxIterations, $\tau_0$, $Q = 1$))
    for every edge (i,j) do                                   ▷ Initialisation
        $\tau_{ij} = \tau_0$
    end for
    for $k = 1$ to antPop do
        Place ant $k$ on a randomly chosen city
    end for
    Compute visibility $\eta(i,j) = 1/d(i,j)\ \ \forall i, j \leftarrow d(i,j)$ distance between vertexes
    Let $T^+, L^+$ shortest tour and length found from the beginning
    for $t = 1$ to maxIterations do                           ▷ Main loop
        for $k = 1$ to antPop do
            Build tour $T^k(t)$ by applying (n-1) times the steps
            if exists at least one city $j$ then
                Choose next city $j$, $j \in J_i^k$, among cities $cl$
                $j = \begin{cases} \arg\max_{u \in J_i^k}\{[\tau_{iu}(t)]^\alpha \cdot [\eta_{iu}]^\beta\} & \text{if } q \leq q_0, \\ J & \text{if } q > q_0, \end{cases}$
                where $J \in J_i^k$ is chosen according to the probability
                and $q$ is a random variable uniformly distributed over $[0, 1]$
                $p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in J_i^k}[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}$
                where $i$ is the current city
            else
                chose the closest $j \in J_i^k$
            end if
            After each transition ant $k$ applies a local update
            $\tau_{ij} \leftarrow (1-\rho)\tau_{ij}(t) + \rho\tau_0$
        end for
        for $k = 1$ to antPop do
            Compute $L^k(t)$ of the tour $T^k(t)$ maintained by ant $k$
        end for
        if an improved tour is found then
            Update $T^+$ and $L^+$
        end if
        for every edge $(i,j) \in T^+$ do
            Update pheromone trails by applying
            $\tau_{ij} \leftarrow (1-\rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}(t)$ where $\Delta\tau_{ij}(t) = Q/L^+$
        end for
    end for
    return Shortest tour $T^+$ and its length $L^+$
end procedure
```

3.1.2 Particle Swarm Optimisation (PSO)

PSO is inspired and adapted from the behaviour of birds or fish schooling in a swarm. In PSO, a population is a swarm and individuals are particles. A population is composed of many particles. For PSO searching, each particle moves to find an optimal solution with a velocity, which is adapted from the particle by exchanging the information of particles in the swarm. A strong point of PSO is that the algorithm saves the local best positions (the best solution of each particle), and a global best position, which is the best solution of the swarm, in each iteration. This optimization algorithm was originally introduced by Kennedy and Eberhart [20] in 1995. The PSO algorithm has the advantage of simplicity for implementation, insensitive to scaling of design variables, it is gradient free, and when used for non-linear programming problems with multiple constraints it can quickly converge to a reasonably good solution. It has very good global searching ability. One disadvantage is the slow convergence in the refined search stage (weak local search ability). However, the original PSO cannot be used to solve certain problems, for example a scheduling problem directly because of its update formulas. Meanwhile, hybrid algorithms evolved and use the genetic operators to re-define the update strategy of PSO. Thus, hybrid algorithm combine the advantages for example of PSO and GA, where differences between both are well documented by Eberhart and Shi [7], to solve the integrated process planning and scheduling (IPPS) problem [28]. A detailed survey of PSO and its variants is given by Gad [9] for the years 2017 to 2022 and mentioned approaches such as Clonal Selection Approach used for surrogate modelling by Bernardino et al [2].

The particle position and velocity of the i^{th} particle at iteration t are represented by D dimensional vectors. The particle position is represented by

$$x_i = (x_{i1}, x_{i2}, ..., x_{iD}). \tag{3.1}$$

The velocity of each particle is randomized at the initial iteration and is represented by $pbest$ and $gbest$ by:

$$v_i = (v_{i1}, v_{i2}, ..., v_{iD}). \tag{3.2}$$

The best local and global positions are represented by $pbest$ and $gbest$, respectively. At each iteration, the particles are updated for their positions by $pbest$ and $gbest$, formulated as:

$$\begin{aligned} v_i^{(t+1)} &= w \times v_i^t + c_c r_1 \times (pbest_i^t - x_i^t) + c_s r_2 \times (gbest_i^t - x_i^t), && (3.3) \\ x_i^{(t+1)} &= x_i^t + v_i^{(t+1)}. && (3.4) \end{aligned}$$

where w represents the inertia weight (between 0.8 and 1.2), which is the impact of the previous velocity on the current velocity. When w is set to a medium value, that is in between $[0.8, 1.2]$, PSO has the best chance to find the global optimum but also takes a moderate number of iterations [8]. Next, c_c, c_s represent cognitive and social parameters, which are acceleration coefficients, in a range of $[0, 2]$, and for all simulations were taken from the paper [8] ($c_c = c_s = 0.1$). (These hyper-parameter values w, c_c, and c_s need to be chosen carefully, see [8]) The values of r_1, r_2 are uniformly distributed random numbers in the range of $[0, 1]$. The original PSO procedure from Kennedy and Eberhart [20] is shown in algorithm 2.

Algorithm 2 Pseudo Code of PSO

```
Set PSO initial parameter value population size $P$, inertia weight $w$, and acceleration constants $cc, cs$
Set $t = 0$
for $i = 1$ to $P$ do
    Generate initial $x_i^t, v_i^t \in [L, U]$ randomly
    Evaluate fitness function $f(x_i^0)$
end for
Set $pbest_i^t, gbest_i^t$
repeat
    $v_i^{t+1} = w \times v_i^t + c_c r_1 \times (pbest_i^t x_i^t) + c_s r_2 \times (gbest_i^t - x_i^t)$    ▷ Calculate new velocity
    $x_i^{(t+1)} = x_i^t + v_i^{(t+1)}$    ▷ Update positions of particles
    Evaluate the fitness function $f(x_i^{(t+1)})$ of all particles in $P^t$
    if $f(x_i^{(t+1)}) \geq f(pbest_i^t)$ then
        $pbest_i^{(t+1)} = x_i^{(t+1)}$
    else
        $pbest_i^{(t+1)} = pbest_i^t$
    end if
    if $f(x_i^{(t+1)}) \geq f(gbest_i^t)$ then
        $gbest_i^{(t+1)} = x_i^{(t+1)}$
    else
        $gbest_i^{(t+1)} = gbest_i^t$
    end if
    $t \leftarrow (t + 1)$    ▷ (Increase iteration counter)
until termination criterion is met
Get the best solution $gbest^t$
```

An application example is shown to visualise the solution strategy, that is intermediate steps are shown how the algorithm behaves. In the following, the algorithm is applied to the Himmelblau's function eq. (3.5), which has four minima at zero and one local maxima occurs, table 3.2, in the chosen parameter range for x and y between -8 and 8.

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2, \quad -8 \leq x, y \leq 8 \tag{3.5}$$

Table 3.2: Minima and Maxima of Himmelblau's function.

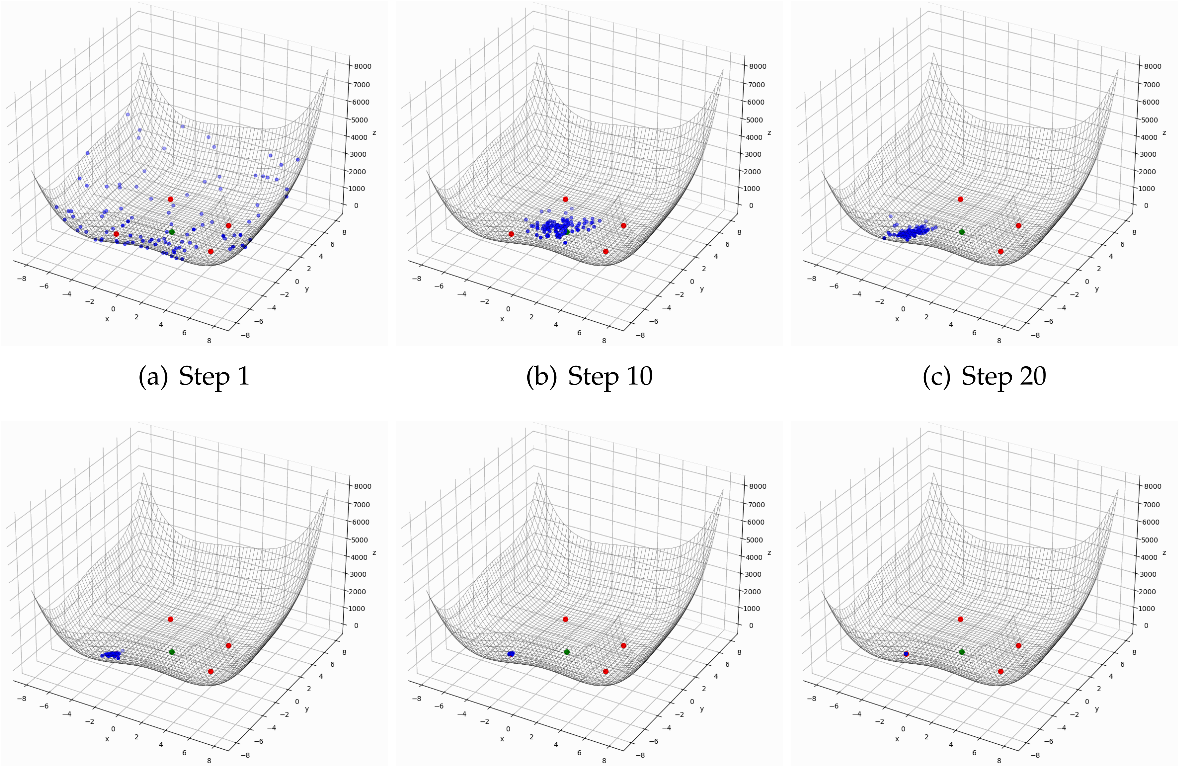
Global minima	
	$f(3, 2) = 0$, $f(-2.805, 3.131) = 0$, $f(-3.779, -3.283) = 0$, $f(3.584, -1.848) = 0$,
Local maxima	
	$f(-0.271, -0.923) = 181.617$

Table 3.3 lists the settings to launch the PSO. Firstly, the population of one hundred is randomly seeded in the parameter space ±8 for the coordinates x and y. Subsequently,

the initialisation of the particles, their best position, velocity and fitness value is performed. In addition, the best global position is calculated based on the initial position of the particles. Then it loops from one generation to another. The algorithm should stop when it reaches the max number of generations, set here to 400, or a success (fitness) criterion (1×10^{-4}). In this case, it is when the average fitness value surpasses a specific value.

Table 3.3: Parameter and solution details for the PSO to perform the simulation for Himmelblau's function.

Parameter	
Dimension - 2D (x,y)	$(\pm 8/\pm 8))$
Population	100
Generation (Iterations)	400
Fitness criterion (abort)	1×10^{-4}
Inertia weight w	0.5
Constant acceleration values cc, cs	0.1
Solution	
Global best solution	$(-3.779, -3.283)$
Best fitness value	2.28×10^{-7}
Average particle best fitness value	$0.000\,99$
Number of generations	60



(a) Step 1 (b) Step 10 (c) Step 20

(d) Step 30 (e) Step 40 (f) Step 51

Figure 3.2: PSO simulation for Himmelblau's function for the first, various intermediate and final step.

Figure 3.2 shows the initial, intermediate, and final steps of the PSO simulation for the Himmelblau function. The initial step in fig. 3.2(a) points out the randomly seeded population in the two-dimensional parameter space. The first steps are very large and already after ten steps, fig. 3.2(b), the population has gathered in the middle of the parameter range. After another ten steps, fig. 3.2(c), the decision is made for approaching an optimum. The last thirty steps in figs. 3.2(d) to 3.2(f) reveal a rather long buzzing around the optimum until the abort criterion takes effect, since all members of the population need to meet the abort criterion. However, an important aspect of using a PSO is that each simulation can find one of the four optima. It is not possible to find all four optima simultaneously with this implementation of the PSO.

3.1.3 Glowworm Swarm Optimisation (GSO)

Glowworm Swarm Optimisation (GSO) is a nature-inspired optimisation algorithm that simulates the behaviour of lighting worms. The GSO is a swarm intelligence algorithm for optimization developed by Krishnanand and Ghose which imitate the flashing behaviour of glowworms [25, 26]. Each glowworm carries a luminescence amount called luciferin[3], which is decided by the function value of glowworm's current location. All through the course of movement, glowworm identifies its neighbours based on local-decision area and selects a neighbour which has a luciferin value higher than its own using a probabilistic mechanism and moves towards it [26, 40, 41]. GSO algorithm is suitable for a concurrent search of several solutions and dissimilar or equal objective function values.

The basic principle of GSO is developed on the behaviour of glowworms by which they can change the intensity of bio-luminescence and appear to glow at different intensities. The quantity of luciferin encodes the fitness of its location in the search space. This makes the agent glowworms glow at an intensity approximately proportional to the function value being optimised. It is assumed that agents glowing brighter attract those that glow with lower intensity. In the algorithm, each glowworm selects, using a probabilistic mechanism, a neighbour that has a luciferin value higher than its own and moves towards it.

A critical aspect of the GSO algorithm is that it incorporates an adaptive neighbourhood range by which the effect of distant glowworms is discarded when a glowworm has a sufficient number of neighbours with brighter glow or when the range to a neighbouring agent goes beyond the maximum range of perception. These movements, based only on local information and selective neighbour interaction, enable the swarm of glowworms to split into disjoint subgroups that converge to high function value points. It is this property of the algorithm that allows it to be used to identify multiple optima of a multi-modal function. It has been shown that GSO can tackle the following class of multi-modal functions: unequal peaks, equal peaks, peaks of concentric circles, peaks surrounded by regions with step-discontinuities (non-differentiable objective function), peaks comprising plateaus of equal heights, peaks located at irregular intervals, change in landscape features with change in scale, and non-separability involving interdependence of objective function variables.

GSO initially distributes a swarm of agents randomly in the search space. The algorithm encapsulates the interplay between the following three mechanisms:

[3] Luciferin is a generic term for the light-emitting compound found in organisms that generate bioluminescence. Luciferins typically undergo an enzyme-catalysed reaction with molecular oxygen. The resulting transformation, which usually involves splitting off a molecular fragment, produces an excited state intermediate that emits light upon decaying to its ground state.

1. **Fitness broadcast**
   Glowworms carry a luminescent pigment called luciferin, whose quantity encodes the fitness of their locations in the objectice space. This allows them to glow at an intensity that is proportional to the function value being optimised. It is assumed that the luciferin level of the glowworm as sensed by its neighbour does not reduce due to distance.

2. **Positive taxis**
   Each glowworm is attracted by, and moves toward, a single neighbour whose glow is brighter than that of itself; when surrounded by multiple such neighbours, it uses a probabilistic mechanism to select one of them.

3. **Adaptive neighbourhood**
   Each glowworm uses an adaptive neighbourhood to identify neighbours; it is defined by a local decision domain that has a variable range $r_{d,i}$ bounded by a hard-limited sensor range ($0 < r_{d,i} < r_s$). A glowworm i considers other glowworm j as its neighbour if j is within the neighbourhood range of i and the luciferin level of j is higher than that of i.

Algorithm 3 outlines a high-level description of the GSO. An application example is shown to visualise the solution strategy, that is intermediate steps are shown how the algorithm behaves. In the following, the algorithm is applied to Matlab's peak function eq. (3.6), which is a function of two variables, obtained by translating and scaling Gaussian distributions. The peak function has in total six local optima, table 3.4, in the chosen parameter range for x and y between -3 and 3.

$$
\begin{aligned}
f(x,y) = {} & 3(1-x)^2\,\mathrm{e}^{\left(-(x^2)-(y+1)^2\right)} \\
& - 10\left(\frac{x}{5} - x^3 - y^5\right)\mathrm{e}^{\left(-x^2-y^2\right)} \\
& - \frac{1}{3}\,\mathrm{e}\left(-(x+1)^2 - y^2\right)
\end{aligned}
\tag{3.6}
$$

Table 3.4: Local optima of Matlab's peak function.

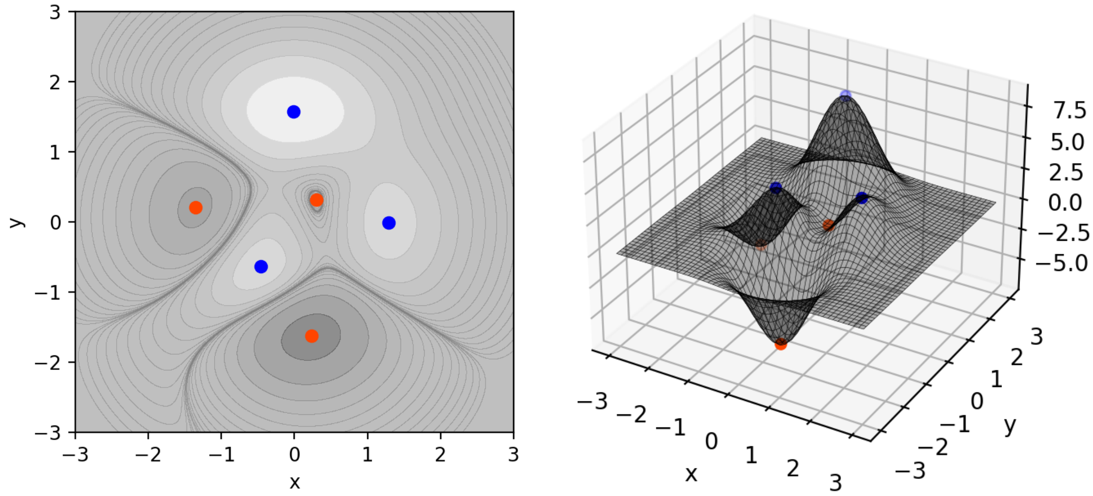
Local maxima	
	$f(-0.0093, 1.5814) = 8.1062,$
	$f(-0.4600, -0.6292) = 3.7766,$
	$f(1.2857, -0.0048) = 3.5925,$
Local minima	
	$f(0.2312, -1.6246) = -6.5510,$
	$f(-1.3483, 0.2072) = -3.0498,$
	$f(0.2973, 0.3213) = -0.0649,$

Figure 3.3 visualises the fitness function (Matlab's peak function) with its six local optima (table 3.4). In both figs. 3.3(a) to 3.3(b) local maxima are marked with blue dots, and local minima are marked as orange dots.

Table 3.5 lists the settings to launch the GSO. A population of sixty is randomly seeded in the parameter space ± 3 for the coordinates x and y. The abort criterion is set to one hundred fifty iterations, since a useful bets fitness value does not apply for the GSO.

Algorithm 3 Pseudo Code of GSO

```
Set number of dimensions $n$
Set number of glowworms $m$
Set step size $s$    ▷ Distance moved by a glowworm at each step
Set luciferin decay constant $\rho$    ▷ $0 < \rho < 1$
Set maximum iteration number = maxIterations
Set decision range gain $\beta$    ▷ $\beta = 0.08$
Set desired number of neighbours $n_t$    ▷ Mostly $n_t = 5$
for $i = 1$ to $n$ do
    $x_i \leftarrow deploy\_glowworm\_randomly()$
    $l_i \leftarrow l_0$    ▷ Initial luciferin
    $r_{d,i} \leftarrow r_0$    ▷ Adaptive neighbourhood sensor range - Local decision range
end for
Set $t = 1$
while $(t \leq maxIterations)$ do
    for $i = 1$ to $n$ do    ▷ Luciferin update phase
        $l_i^{(t)} = (1-\rho)l_i^{(t-1)} + \gamma J(x_i^{(t)})$    ▷ $\gamma$: Luciferin enhancemnet constant
    end for
    for $i = 1$ to $n$ do    ▷ Movement phase
        $N_i^{(t)} = \{j : d_{i,j}^{(t)} < r_{d,i}^{(t)}; l_i^{(t)} < l_j^{(t)}\}$
        for $j \in N_i^{(t)}$ do
            $p_{ij}^{(t)} = \dfrac{l_j^{(t)} - l_i^{(t)}}{\sum_{k \in N_i^{(t)}} (l_k^{(t)} - l_i^{(t)})}$
        end for
        $j \leftarrow select\_glowworm(\vec{p})$
        $x_i^{(t+1)} \leftarrow x_i^{(t)} + s\left(\dfrac{x_j^{(t)} - x_i^{(t)}}{\|x_j^{(t)} - x_i^{(t)}\|}\right)$    ▷ Glowworm movement
        $r_{d,i}^{(t+1)} \leftarrow \min\{r_s, \max\{0, r_{d,i}^{(t)} + \beta(n_t - |N_i^{(t)}|)\}\}$    ▷ Neighbourhood range update phase
    end for
end while
```



(a) 2D

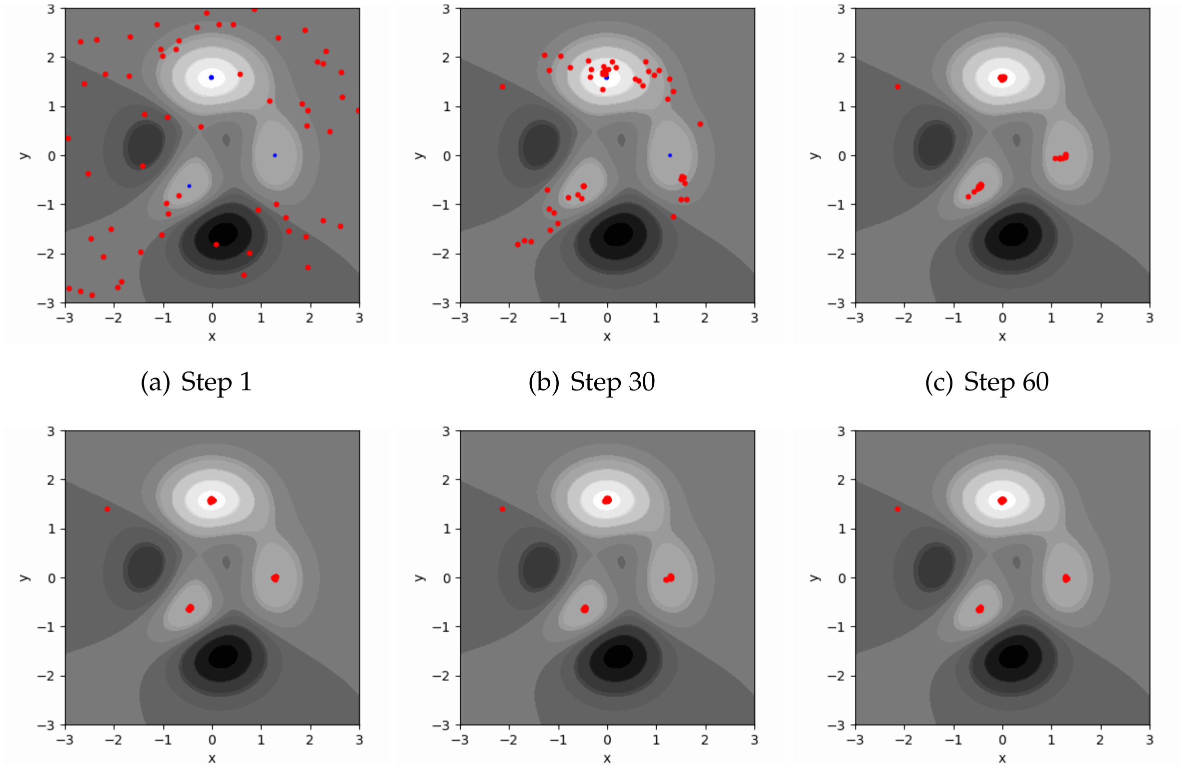
(b) 3D

Figure 3.3: Matlab's Peak function with local optima, blue dots are local maxima, and orange dots are local minima. (a) is the projected view from above, and (b) is the three-dimensional view with the z-axis showing the fitness function.

Table 3.5: Parameter and solution details for the GSO to perform the simulation for Matlabs Peak function.

Parameter	
Dimension - 2D (x,y)	$(\pm 3/\pm 3)$
Population	60
Number of iterations	150
Luciferin decay constant, ρ	0.4
Luciferin enhancement factor, γ	0.6
Neighbourhood enhancement factor, β	0.08
Sensor range, radius r_d	1.8, 2.0, 3.0
Desired number of neighbours, n_t	5
6 Step size, s	0.05
Solution - $r_d = 3$	
Local maximum $(-0.0093, 1.5814)$	8.1062
Solution - $r_d = 2$	
Local maximum $(-0.0093, 1.5814)$	8.1062
Local maximum $(-0.4600, -0.6292)$	3.7766
Solution - $r_d = 1.8$	
Local maximum $(-0.0093, 1.5814)$	8.1062
Local maximum $(-0.4600, -0.6292)$	3.7766
Local maximum $(-1.2857, -0.0048)$	3.5925

Hyper-parameters for the GSO such as ρ, γ, β, n_t, and the step size s are kept constant for each simulations. Only the sensor range r_d was reduced from 3, to 2, and finally 1.8. From the solutions of the simulations it can be seen that as the sensor is reduced, more local maxima are continuously detected. This parameter is one of the most important for successfully finding optima. A prerequisite for the successful application of this algorithm is therefore a good knowledge of the problem. With these techniques in particular, the setting of the hyper-parameters allows a great amount of freedom, but this also often means that the algorithm sometimes behaves contrary to the expected behaviour.



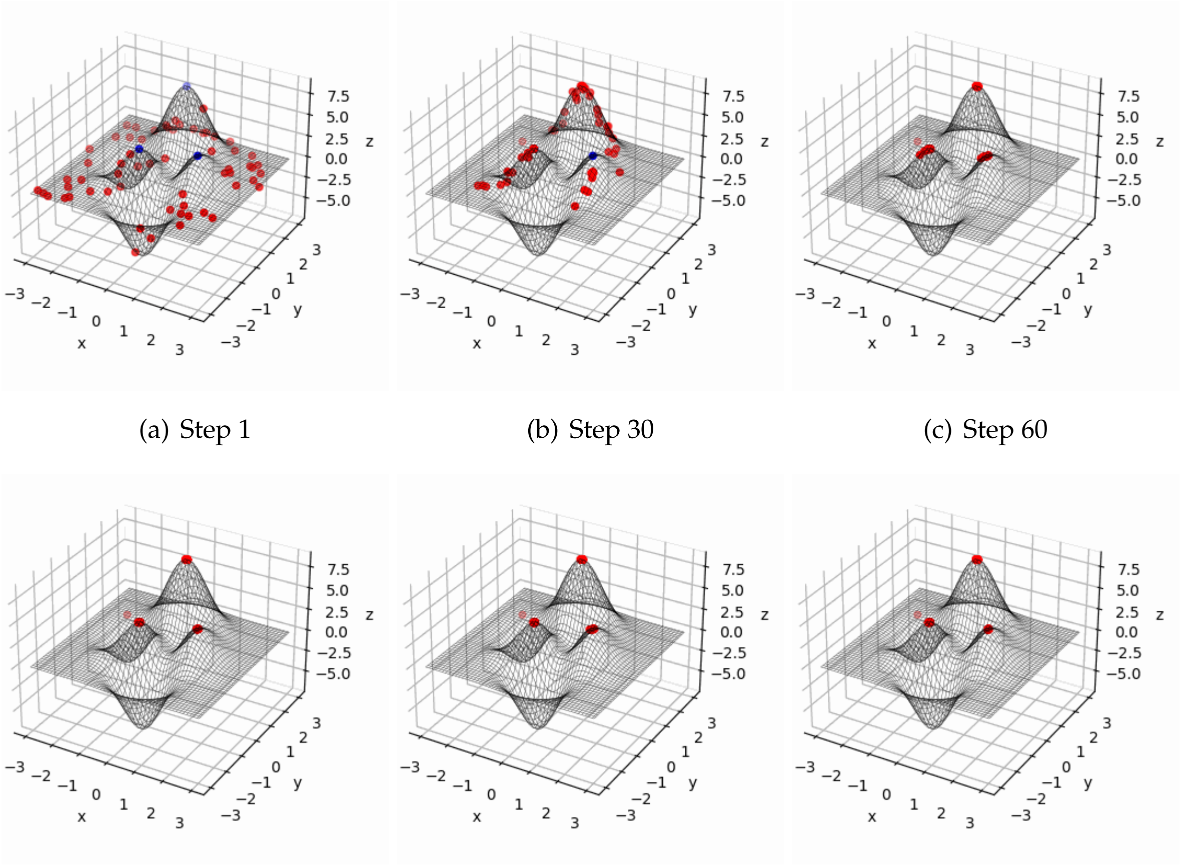
(a) Step 1 (b) Step 30 (c) Step 60

(d) Step 90 (e) Step 120 (f) Step 150

Figure 3.4: GSO simulation for Matlab's Peak function with the sensor range set to 1.8 in a two-dimensional view for the first, various intermediate and final step from (a) to (f).

Figure 3.4 shows intermediate steps of the GSO simulation for Matlabs Peak function for a sensor range of 1.8 in a two-dimensional view. Its corresponding three-dimensional view can be seen in the same order in fig. 3.5. The initial step in fig. 3.4(a) points out the randomly seeded population in the two-dimensional parameter space. The first steps are very large and already after thirty steps, fig. 3.4(b), the population has identified three local maxima and tend to move toward these maxima. Only thirty steps further, fig. 3.4(c), each local population has gathered around the three local maxima. Similar to the PSO, the last sixty to seventy steps, figs. 3.4(d) to 3.4(f), reveal a rather long buzzing around the three optima until the abort criterion takes effect, which is in this case reaching the maximum iteration count of 150. Amazingly, a single particle remains motionless, which is not captured by any local population at around (x/y = -2/1.5). There are several reasons for this: the number of particles per local population is limited to the specified input parameter termed number of members. Once this number is reached, no new particles are captured. Furthermore, it is possible, and especially due to the specification of the sensor range, that a particle is first attracted by one population and then by another in the progress of tracking, but due to the different speeds of movement, the attraction is not sufficient and it falls out of the sensor range. And so it remains motionless in space.

Finally, it should be noted that the algorithm has a weak point in the current implementation, namely the termination criterion. Several authors have taken on this problem, but have never been able to solve it satisfactorily. The reason is the treatment of several



(a) Step 1 (b) Step 30 (c) Step 60

(d) Step 90 (e) Step 120 (f) Step 150

Figure 3.5: GSO simulation for Matlab's Peak function with the sensor range set to 1.8 in a three-dimensional view for the first, various intermediate and final step from (a) to (f).

local optima at once. To the author's knowledge, a solution for an efficient termination criterion is still being worked on.

3.1.4 Advances for Aerodynamic Applications

Since the birth of what is now known as swarm intelligence, bio-inspired solution methods have evolved into algorithms with a higher degree of sophistication that involve mimicking other processes and behaviours observed in nature. Bio-inspired computations are based on the emergence of collective intelligence from large populations of agents with simple behavioural patterns for communication and interaction. A myriad of bio-inspired SI methods have been proposed and the current list is huge and growing almost daily. Almost each SI algorithm usually works well for a specific problem only. Many of the recent developments have set themselves the goal of combining the advantages and disadvantages of individual SI methods as well as non-SI approaches, which are referred to as hybridisation approaches in the literature. Hybridisation can be used to extend the range of applications or the methodology of SI algorithms. An example of this is the development by Ceollo and Lechuga [4] for the extension of PSO for multi-objective optimisation. Their approach uses the concept of Pareto dominance to determine the flight direction of a particle and it maintains previously found nondominated vectors in a global repository that is later used by other particles to guide their own flight. Efficiency measurements indicate that their approach is highly competitive with current evolutionary multi-objective optimisation techniques. Another valuable approach was proposed by Han and Liu [12] for the hybridisation of SI algorithms. Proposed is a diversity-driven hybrid PSO based on gradient search to improve the search ability, since it is easy for the PSO to cause the swarm to lose its diversity and lead to premature convergence. The adaptive PSO is first used to search the solution till the swarm loses its diversity. Then, the search process turns to a new PSO, and the particles update their velocities with their gradient directions as well as repel each other to improve the swarm diversity. Depending on the diversity value of the swarm, the proposed hybrid method switches alternately between two PSOs. The hybrid algorithm adaptively searches local minima with random search realized by adaptive PSO and performs global search with semi-deterministic search realized applying the second PSO, which then improves the search ability. Results show that the proposed hybrid algorithm has better convergence performance with better diversity compared to classical PSOs.

From an application perspective, bio-inspired inspired computation is popular in computer graphics, robotics and collaborative cluster control scenarios such as path planning and task assignment. For example, Gálvez and Iglesias [10] investigate the use of PSO to recover the shape of a surface from clouds of (either organized or scattered) noisy 3D data points, a challenging and well known problem in computer graphics that appears recurrently in a wide range of applications such as CAD design, data visualization, virtual reality, medical imaging and motion picture industry. SI algorithms have also found their way into the field of aerodynamic activities, even if the applications are still limited. A series of developments and improvements for airfoil optimisation were conducted by Khurana and colleagues [23, 22, 21]. In this context, an adaptive mutation-particle swarm optimisation method has been developed as this method tends to converge to a sub-optimal solution for complex aerospace design problems. Khurani has thus found out that for his applications and formulation of the optimisation problem a local optimum is found instead of a global optimum. The reasons for this may be manifold. Thus, a Gaussian-based operator is implemented to optimise the diversity of the particle search by mutation with a given probability. The amount of mutation during the optimisation phase is determined by the collective search patterns of the swarm. The proposed algorithm prevents convergence towards a sub-optimal design and at the same time limits the computing resources required during the optimisation cycle. The aerodynamic optimisation of high-speed trains has also been successfully examined. He and Liu [13] proposed also hybridisations of the particle swarm optimization algorithm for improving the aerodynamic optimization efficiency of a high-speed train head shape. At first, a hybrid particle swarm optimization algorithm, which employs the basic PSO and a backward learning strategy in particle swarm optimization, is introduced for constructing an optimal least squares support vector regression model. The second approach is an elite-evolved multi-objective particle swarm optimiser, which employs a grouping-based stochastic elite competition mechanism and elite gathering behaviour to further improve the efficiency of multi-objective optimisation. In addition, various surrogate models were evaluated to increase the prediction accuracy of the total drag coefficient. The results show that the regression model achieves the lowest prediction errors in comparison to their standard PSO approach. In a concluding remark, it can be stated that due to the large number of hybridisations, the core of the basic SI algorithm fades into the background, or the original algorithm is modified to such an extent that it is difficult to understand whether the application of an SI algorithm still makes sense.

A more promising approach has been undertaken for the simulation of dynamic, uncertain environments and the complex tasks that arise in this context to ensure that the unmanned aerial system (UAV) will evolve towards clustering, autonomy and intelligence. Zhou, Rao and Wang [45] provide an elaborate survey for the problems arising for hierarchical control frameworks. Their view to investigate the research work by classifying UAV swarm intelligence research was split into five layers, namely, decision-making layer, path planning layer, control layer, communication layer, and application layer. The relationship between each level is explicitly illustrated, and the research trends of UAV swarm intelligent technology for each layer are given. Finally, limitations and possible technology trends of UAV swarm intelligence are also covered to enable further research interests.

In recent years, the research community has witnessed an explosion of literature dealing with the mimicking of behavioural patterns and social phenomena observed in nature towards efficiently solving complex computational tasks. This trend has been especially dramatic in what relates to optimization problems, mainly due to the unprecedented complexity of problem instances, arising from a diverse spectrum of domains such as transportation, logistics, energy, climate, social networks, health and industry, among many others. Numerous papers outline the state of the art and identify open challenges concerning the most relevant areas within bio-inspired optimisation. It remains to be seen whether SI algorithms will gain a foothold in aerodynamics, as a single optimum is usually sought rather than multiple optima, highlighting the need to reach consensus and join forces to gain valuable insights into understanding this family of optimisation techniques.

3.2 Geometrically Exact Beam (GEB)

The following introduction to the geometrically exact beam theory was taken from papers of Charlotte Rodriguez [36, 35, 27] and her dissertation manuscript [34].

Beam models, which describe the three-dimensional motion of thin elastic bodies that are subject to large deflections and rotations, have found many applications in civil engineering, mechanical engineering and aerospace engineering. Depending on the assumptions made for the beam (material law, motion magnitude, shearing), there are various Partial Differential Equation (PDE) models for flexible beams, e.g. the Euler-Bernoulli, Rayleigh and Timoshenko beam equations, which apply to small displacements and strains. However, if the deflections and rotations are not small compared to the overall dimensions of the body, which is the case with modern highly flexible lightweight structures, a geometrically non-linear model is required. Examples include robotic arms [11] as well as flexible aircraft wings [30] or wind turbine blades [42] designed to be lighter and slender to improve aerodynamic efficiency. The Geometrically Exact Beam (GEB) model and the Intrinsic Geometrically Exact Beam (IGEB) model discussed herein are such geometrically non-linear models describing the motion of a three-dimensional beam. They take account of shearing without warping: the cross-sections remain plane, i.e. do not change shape, but can rotate independently of the motion of the centre line. The beam can experience large displacements of its centre line and large rotations of its cross-sections. Both systems are one-dimensional. A frequently used categorisation for the definition of a GEB is as follows

- Beams are structures with one dimension that is much larger than the other two.
- The term "geometrically accurate" refers to the fact that there are no approximations in the geometry of the reference line or the reference cross-section.

- The geometrically exact equations of motion in their most fundamental form are first-order PDEs, which may be written with or without displacement or rotational variables.

One important reason why both systems are described is that the solution of the equations is done with the IGEB model and with a non-linear transformation the GEB model can be restored and the deformation of the beam can be calculated.

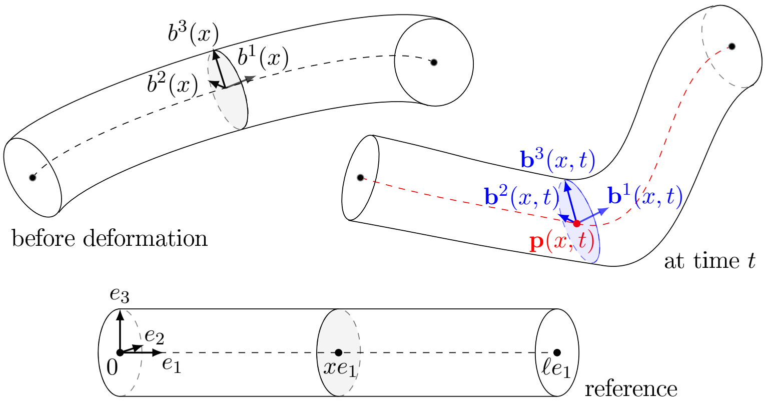


Figure 3.6: The straight reference beam (bottom), the beam before deformation characterized by the curvature change $\Upsilon_c = \text{vec}(R^T\, d/dx\, R)$ where $R(x) = [b^1\, b^2\, b^3]$ (upper left), and the beam at time t described by the state variables $\mathbf{p}(x,t)$ and $\mathbf{R}(x,t) = [\mathbf{b}^1\, \mathbf{b}^2\, \mathbf{b}^3]$ (upper right), Rodriguez [34].

Figure 3.6 explains the three stages of the beam. The beam is idealised as a reference line (bottom), also termed the centre line, and a family of cross-sections with a fixed coordinate system $\{e_j\}_{j=1}^3$. Before freely vibrating and upon the initial boundary condition (upper left), that is at rest with an initial centre line curvature change defined as Υ_c with $R(x) = [b^1 b^2 b^3]$, both the position of the centre line $p(x) \in [0, l]$ as well as the orientation of the cross-sections are known. The cross-sections are given by the columns of the rotation matrix $R(x) \in [0, l]$. At any time $t \in [0, T]$ (upper right), both the position $\mathbf{p}(x,t) \in [0, l] \times [0, T]$ of the centre line as well as the orientation of the cross sections, which is given by the columns of a rotation matrix $\mathbf{R}(x,t) \in [0, l] \times [0, T]$, are unknown. Since shear deformation is permissible, the cross-sections do not necessarily have to be perpendicular to the centre line. Note that for the following notations of cross-products, for example between vectors $p, q \in \mathbb{R}^3$, these can also be written as a matrix-vector multiplication $\hat{p}q$, where $\hat{p} \in \mathbb{R}^{3x3}$ is the skew-symmetric matrix

$$\hat{p} = \begin{bmatrix} 0 & -p_3 & p_2 \\ p_3 & 0 & -p_1 \\ -p_2 & p_1 & 0 \end{bmatrix}, \tag{3.7}$$

and hence for any skew-symmetric $p \in \mathbb{R}^{3\times3}$, the vector $\text{vec}(p) \in \mathbb{R}^3$ can be applied as $p = \text{vec}(\hat{p})$. In other words, these matrix-like cross-products become very useful for further programming, evaluating the inverse, and for the quaternion treatment. The standard basis of $\mathbb{R}^3$ is denoted by $\{e_j\}_{j=1}^3 = \{(1,0,0)^T, (0,1,0)^T, (0,0,1)^T\}$

The mathematical model for geometrically exact beams may be written in terms of the position of the centre line of the beam and the orientation of its cross sections, with respect to a fixed coordinate system $\{e_j\}_{j=1}^3$ (the standard basis of $\mathbb{R}^3$ here). This is the commonly known Geometrically Exact Beam model, or GEB, which originates from the work of Reissner [33] and Simo [39]. The governing system for the GEB model is quasilinear, consisting of six equations and of second order both in space and time. The state is then $(\mathbf{p}, \mathbf{R})$, expressed in $\{e_j\}_{j=1}^3$ and composed of the centre line position $\mathbf{p}(x,t) \in \mathbb{R}^3$ and cross section orientation. For a freely vibrating beam of length l, that is if the external forces and moments, which can represent gravity or aerodynamic forces, for example, are set to zero, the governing equations of the GEB model is given by

$$\begin{bmatrix} \partial_t & \mathbf{0} \\ (\partial_t \hat{\mathbf{p}}) & \partial_t \end{bmatrix} \left[\begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \mathbf{M} v \right] = \begin{bmatrix} \partial_x & \mathbf{0} \\ (\partial_x \hat{\mathbf{p}}) & \partial_x \end{bmatrix} \left[\begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} z \right], \tag{3.8}$$

$$v = \begin{bmatrix} V \\ W \end{bmatrix}, \quad z = \begin{bmatrix} \Phi \\ \Psi \end{bmatrix}, \quad \text{or } \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} z = \begin{bmatrix} \phi \\ \psi \end{bmatrix} \tag{3.9}$$

where the unknown states, for $x \in [0, l]$ and $t \geq 0$, are the position $\mathbf{p}(x,t) \in \mathbb{R}^3$ of the beam's centre line and a rotation matrix $\mathbf{R}(x,t)$ giving the orientation of the cross sections of the beam, both expressed in the body-fixed coordinate system. Rotation matrices are a set of unitary real matrices ($\mathbf{R}^{-1}\mathbf{R} = \mathbf{I}_3$) of size 3 and with a determinant equal to 1. Note that for any $p \in \mathbb{R}^3$, $\hat{p}$ denotes the skew-symmetric matrix equivalent to the vector cross multiplication by p, that arises when treating the cross product as a linear map in the second argument, while vec$(\cdot)$ permits to recover $p = \text{vec}(\hat{p})$. The mass matrix $\mathbf{M}(x) \in \mathbb{R}^{6x6}$ is a positive definite symmetric matrix and is set to

$$\mathbf{M} = \rho \, \text{diag}\,(a\mathbf{I}_3, J), \quad J = \text{diag}\,(I_2 + I_3 k_1, I_2, I_3), \quad k_1 > 0. \tag{3.10}$$

The cross-section area is denoted by a, $I_3 \in \mathbb{R}^{3\times3}$ is a diagonal unitary matrix, ρ is the density, and J is the inertia tensor. The polar moment of area is corrected by the factor k_1. $V(x,t), W(x,t), \Phi(x,t), \Psi(x,t) \in \mathbb{R}^3$ denote the linear velocity, angular velocity, internal forces and internal moments of the beam respectively, all expressed in a moving coordinate system attached to the centre line of the beam, body-fixed axis, and are defined by

$$v := v(x,t) = \begin{bmatrix} V \\ W \end{bmatrix} = \begin{bmatrix} \mathbf{R}^T \partial_t \mathbf{p} \\ \text{vec}(\mathbf{R}^T \partial_t \mathbf{R}) \end{bmatrix}, \tag{3.11}$$

$$z := z(x,t) = \begin{bmatrix} \Phi \\ \Psi \end{bmatrix} = \mathbf{C}^{-1} \begin{bmatrix} \mathbf{R}^T \partial_x \mathbf{p} - e_1 \\ \text{vec}(\mathbf{R}^T \partial_x \mathbf{R} - \Upsilon_c) \end{bmatrix}, \quad \Upsilon_c = R^T \frac{\mathrm{d}}{\mathrm{d}x} R \tag{3.12}$$

where $e_1 = [1,0,0]^T$, and $R(x)$ depends on the initial form of the beam, as it may be pre-curved and twisted before deformation. The flexibility (or compliance) matrix $C(x) \in \mathbb{R}^{6x6}$ is a positive definite symmetric matrix and is defined as

$$\mathbf{C} = \text{diag}\,(S_1, S_2)^{-1}, \quad \begin{cases} S_1 = a \, \text{diag}\,(E, k_2 G, k_3 G) \\ S_2 = J \, \text{diag}\,(G, E, E) \end{cases}. \tag{3.13}$$

G is the shear modulus, E is the Young modulus, and k_2, k_3 are shear correction factors. The curvature before deformation, when at rest, is written as $\Upsilon_c \in \mathbb{R}^3$.

The mathematical model can also be written in terms of so-called intrinsic variables – namely, velocities V, W and internal forces/moments Φ, Ψ, or equivalently velocities and strains – expressed in a moving coordinate system attached to the beam (the body-fixed moving basis). The unknown state is

$$y = \begin{bmatrix} V \\ W \\ \Phi \\ \Psi \end{bmatrix}. \tag{3.14}$$

The unknown state $y(x,t) \in \mathbb{R}^{12}$ (eq. (3.14)) whose dynamics are then given by a system of the form

$$\partial_t y + A(x)\partial_x y + \bar{B}(x)y = \bar{g}(x,y), \tag{3.15}$$

where the coefficient matrices A, B and the source g depend on $\mathbf{M}, \mathbf{C}$ and R. The matrix $A := A(x)$ is defined as

$$A = \begin{bmatrix} \mathbf{0}_6 & \mathbf{M}^{-1} \\ \mathbf{C}^{-1} & \mathbf{0}_6 \end{bmatrix}. \tag{3.16}$$

It is worth mentioning that the matrix $\bar{B}(x)$ is indefinite and cannot be assumed to be arbitrarily small. In particular cases, the norm of this matrix can be calculated explicitly and it turns out that it is different from zero for realistic beam parameters. The matrix $\bar{B} := \bar{B}(x)$ is defined as

$$\bar{B} = \begin{bmatrix} \mathbf{0}_6 & -\mathbf{M}^{-1}\mathbf{E} \\ \mathbf{C}^{-1}\mathbf{E}^T & \mathbf{0}_6 \end{bmatrix}, \quad \text{with } \mathbf{E} = \begin{bmatrix} \hat{\Upsilon}_c & \mathbf{0}_3 \\ \hat{e}_1 & \hat{\Upsilon}_c \end{bmatrix}, \quad \Upsilon_c = R^T \frac{\mathrm{d}}{\mathrm{d}x} R \tag{3.17}$$

The function $\bar{g} : [0,l] \times \mathbb{R}^{12}$ is defined for all $x \in [0,l]$ and $u = [u_1^T, u_2^T, u_3^T, u_4^T]^T \in \mathbb{R}^{12}$ with each $u_j \in \mathbb{R}^3$, where the map G is defined as

$$G(u) = -\begin{bmatrix} \mathbf{M}^{-1} & \mathbf{0}_6 \\ \mathbf{0}_6 & \mathbf{C}^{-1} \end{bmatrix} \begin{bmatrix} \hat{u}_2 & \mathbf{0}_3 & \mathbf{0}_3 & \hat{u}_3 \\ \hat{u}_1 & \hat{u}_2 & \hat{u}_3 & \hat{u}_4 \\ \mathbf{0}_3 & \mathbf{0}_3 & \hat{u}_2 & \hat{u}_1 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \hat{u}_2 \end{bmatrix} \begin{bmatrix} \mathbf{M} & \mathbf{0}_6 \\ \mathbf{0}_6 & \mathbf{C} \end{bmatrix} \tag{3.18}$$

Furthermore, the function $\bar{g}$ is non-linear – quadratic – with respect to the unknown. Equation (3.15) is the IGEB, which originates from the work of Hodges [15, 16]. It is of its own interest in aeroelastic modelling and engineering, see [30] and references therein. Considering the IGEB model raises the number of governing equations from six to twelve, but with the advantage of dealing with a first-order hyperbolic system (as $A(x)$ is a hyperbolic matrix thus all eigenvalues of $A(x)$ are real and one may find 12 associated independent eigenvectors) which is only semi-linear; and a large literature, beyond the context of beam models exists on such models.

Both approaches, that is the GEB versus the IGEB, are related to each other by a non-linear transformation. The transformation reads

$$\tau : (\mathbf{p}, \mathbf{R}) \mapsto \begin{bmatrix} \mathbf{I}_6 & \mathbf{0}_6 \\ \mathbf{0}_6 & \mathbf{C}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{R}^T \partial_t \mathbf{p} \\ \mathrm{vec}(\mathbf{R}^T \partial_t \mathbf{R}) \\ \mathbf{R}^T \partial_x \mathbf{p} - e_1 \\ \mathrm{vec}(\mathbf{R}^T \partial_x \mathbf{R}) - \Upsilon_c \end{bmatrix} = y\,. \tag{3.19}$$

The full set of equations and boundary conditions for the GEB, for instance, for the freely vibrating beam clamped at $x = 0$ and free at the other end is defined as

$$\begin{cases} \begin{bmatrix} \partial_t & \mathbf{0} \\ (\partial_t \hat{\mathbf{p}}) & \partial_t \end{bmatrix} \left[\begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \mathbf{M} \begin{bmatrix} V \\ W \end{bmatrix} \right] = \begin{bmatrix} \partial_x & \mathbf{0} \\ (\partial_x \hat{\mathbf{p}}) & \partial_x \end{bmatrix} \begin{bmatrix} \phi \\ \psi \end{bmatrix} & \in [0,l] \times [0,T], \\ (\mathbf{p}, \mathbf{R})(0,t) = (f^{\mathbf{p}}, f^{\mathbf{R}}) & t \in [0,T], \\ \begin{bmatrix} \phi \\ \psi \end{bmatrix} (l,t) = -\begin{bmatrix} \mathbf{R}(l,t) & \mathbf{0} \\ \mathbf{0} & \mathbf{R}(l,t) \end{bmatrix} K v(l,t) & t \in [0,T] \\ (\mathbf{p}, \mathbf{R}, \partial_t \hat{\mathbf{p}}, \mathbf{R}W)(x,0) = (\mathbf{p}^0, \mathbf{R}^0, \mathbf{p}^1, w^0)(x) & x \in [0,l], \end{cases} \tag{3.20}$$

and in this case the boundary data is written with the feedback control for the internal forces and moments which is denoted by $K \in \mathbb{R}^{6\times 6}$. That matrix K [34] is based on a

Lyapunov stabilisation and was taken as it is. A simple switch in the code then enables $K \neq \mathbf{0}$ or $K = \mathbf{0}$, which means that $[\phi, \psi]^T = \mathbf{0}$ can be set. For the IGEB the set of equations is

$$\begin{cases} \partial_t y + A(x)\partial_x y + \bar{B}(x)y = \bar{g}(x,y), & \in [0,l] \times [0,T], \\ v(0,t) = \mathbf{0} & t \in [0,T], \\ z(l,t) = -Kv(l,t) & t \in [0,T], \\ y(x,0) = y^0(x) & x \in [0,l] \end{cases} \tag{3.21}$$

where for eq. (3.20) $(\mathbf{p}^0, \mathbf{R}^0, \mathbf{p}^1, w^0)$ as well as in eq. (3.21) $y^0(x)$ are the initial conditions on the beam over the length l for the set of equations. All others are boundary data which are applied for the whole simulations over time.

The following simulations are based on the description of Hesse and Palacios [14] for which the mass and flexibility matrix are set to

$$\mathbf{M} = \operatorname{diag}(1,1,1,20,10,10), \quad \mathbf{C} = \operatorname{diag}(10^4, 10^4, 10^4, 500, 500, 500)^{-1} \tag{3.22}$$

A single beam is considered, clamped at $x = 0$ and controlled at $x = l$ which is described by the system in eq. (3.20) or its intrinsic counterpart in eq. (3.21). An initial datum without shear, that is the cross-sections remain perpendicular to the centre line, was chosen. The initial position of the centre line $\mathbf{p}^0$ and the cross-section with $\mathbf{R}^0$ are given by

$$\mathbf{p}^0(x) := \frac{1}{\sqrt{2}} \begin{bmatrix} x \\ 1 - \cos(x) \\ \sin(x) \end{bmatrix}, \quad \mathbf{R}^0(x) := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & -1 \\ \sin(x) & \sqrt{2}\cos(x) & \sin(x) \\ \cos(x) & -\sqrt{2}\sin(x) & \cos(x) \end{bmatrix}. \tag{3.23}$$

Initial internal forces and moments take the form

$$z^0 = \begin{bmatrix} z_1^0 \\ z_2^0 \end{bmatrix}, \quad z_1^0 = \mathbf{0}_3, \quad z_2^0 = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} - \Upsilon_c, \tag{3.24}$$

and for certain cases $\Upsilon_c := \mathbf{0}$ meaning that at rest the beam is not curved. In other words and to sum that up, the initial angular velocities are set to $W := [W_1 \neq 0, W_2 = 0, W_3 \neq 0]^T$ over the beams length $[0,l]$ and the initial internal moments are set to $\psi := [\psi_1 \neq 0, \psi_2 = 0, \psi_3 \neq 0]^T$.

It should be noted with regard to the test cases that this is only a small selection and also an incomplete selection from the numerical simulations performed. Two benchmark test cases from Hesse and Palacios [14] were selected. Unfortunately, there was not enough time to complete the analyses and comparisons with both test cases. Therefore, only the intermediate steps from the first numerical simulations are shown.

Figure 3.7 shows the simulation of the rotating arm (beam) by applying initial angular velocities and internal moments without initial curvature of the beam in fig. 3.7(a) and with initial curvature in fig. 3.7(b). Intermediate time steps of the beams deformation while freely vibrating are indicated with black solid lines. For this simulations, the total time shown is for fig. 3.7(a) at $T = 5\,\mathrm{s}$, and for fig. 3.7(b) it is $T = 9\,\mathrm{s}$. Both final beam settings are marked with a solid red line. The red dashed line in both fig. 3.7 should be compared with the data from [14] but is missing because there was not enough time to complete this endeavour. It is important to note that the red dashed line is not purely circular, as the beam is subject to deformation. Proof of the correctness of the simulation also includes determining the position when the beam is completely stretched, i.e. has

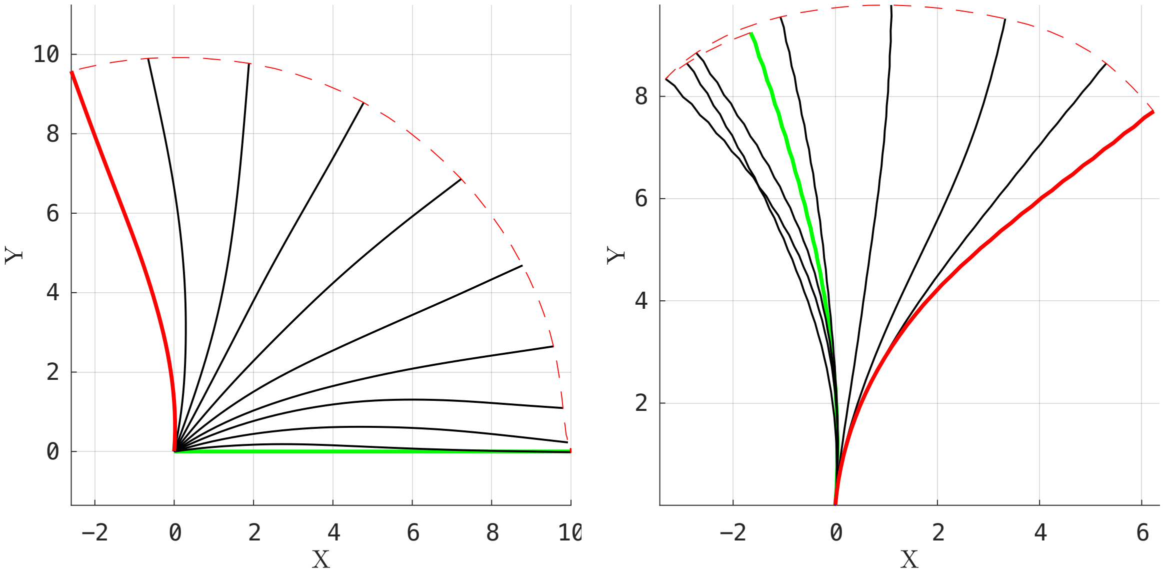


Figure 3.7: (a) Rotating arm clamped at $x = 0$ and free at $x = l$ by setting initial angular velocities and internal moments across the beams length with an initial curvature set to $\Upsilon_c = 0$, and (b) with an initial curvature $\Upsilon_c \neq 0$. In both figures, the initial position of the beam ($t = 0$) is shown with a green solid line, black solid lines are intermediate positions at time t, and the red solid line shows the final position of the beam at time T.

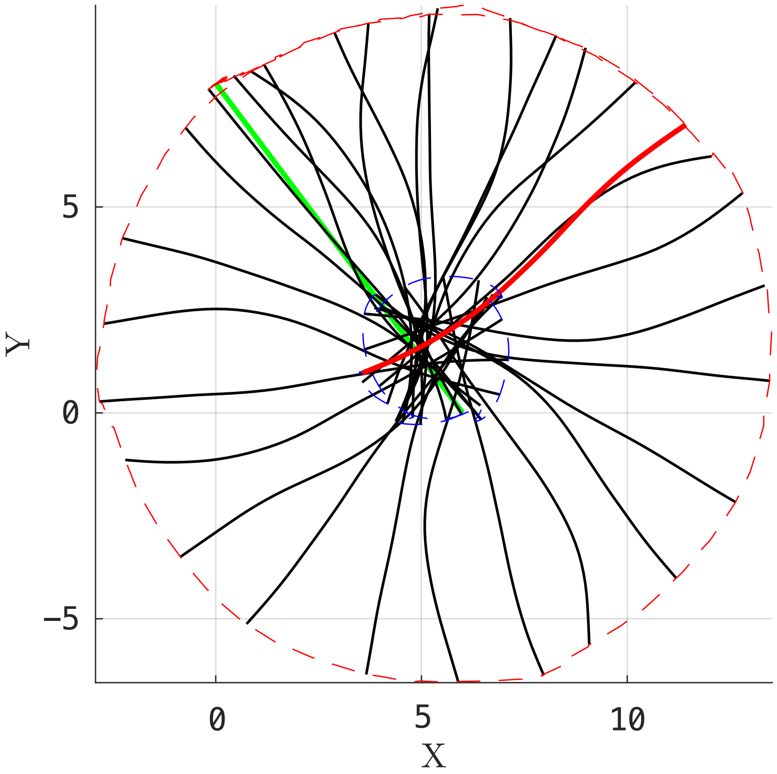


Figure 3.8: Flying spaghetti problem.

no deformation, and determining when the exact circle (midpoint at $x = 0$ with radius l) is touched from the beam at $x = l$.

The second benchmark test case was termed the flying spaghetti, fig. 3.8. Both ends of the beam are free and again an initial angular velocity and internal moment was applied. No initial curvature is introduced. The green solid line shows the initial position ($t = 0$) of the beam, while the black solid lines indicate intermediate deformation states of the beam at time t, and finally the red solid line is the final position ($T = 15\,\mathrm{s}$) of the beam for that numerical simulation. The representation looks unusual because the beam moves in the x-direction, however makes it more convenient to compare the results with the data in [14]. Proof of correctness of the simulation is by comparing the red dashed line for the beams position at $x = l$ as well as for the blue dashed line for the beams position at $x = 0$. An important aspect of this simulation is that the external forces are zero, i.e. there is no gravity and no aerodynamic forces acting on the beam. This means that the y-position always remains the same. In contrast, the frequency of the oscillation decreases because of internal shear stresses (friction) of the beam, and after a certain time, returns to its resting state. Also for this experiment and for these assumed conditions, the beam is deformation-free at a certain time, and the end point of the beam at $x = l$ touches the exact circle with centre at ($x/y = 5/1$) and a radius of $l4/5$.

Sadly, this remains an incomplete work, but the implementation has shown the most important steps to obtain the uniqueness of the solution of this differential equation, in this case inverting the non-linear operator from IGEB to GEB in eq. (3.19) to obtain the deformation of the beam.

4 Conclusion

This report summarises the main activities as well as additional research activities during the secondment to the USAFA from end of October 2022 until end of July 2023. An important part of the secondment was the joint publication at the AIAA Aviation Forum 2023 in San Diego. The joint publication not only served as documentation for the secondment, but was also a common topic that brought the two institutions closer together. It enriched the understanding of vortex-dominated flows around aircraft, and also contributed to the application of surrogate modelling and its limitations for such applications. It further showed development paths that have potential to be investigated in more detail. A continuation of this collaboration between the USAFA and the DLR depends on the expectations of both institutions. The topics in the field of high-speed aircraft most often overlap, in this case the numerical aerodynamic evaluation of industrial applications, and in addition both institutions are integrated in the NATO AVT-351 research group.

The second paper mentioned in this report, which was also presented at the AIAA 2023 in San Diego, is dedicated to the work conducted in DLR's internal oLAF project. It summarises the developments on the efficient prediction of gust interactions on an aircraft using a linearised frequency domain solver that incorporates various interpolation techniques. Since the oLAF project will be completed by the end of 2023, the presentation of the work at AIAA 2023 is also a good opportunity to present the work and evaluate it in a discussion.

The final presentation at the AIAA 2023 in San Diego was contributed to the Stability and Control Workshop held by the AIAA. During the first workshop it turned out that the prediction of the pitching moment coefficient showed a large spread among the participants. This led to an investigation on a smaller test case that would significantly improve the prediction of the pitching moment coefficient. The results were presented at the AIAA conference, however no final conclusion or best practice was established, for example for improving grid generation for moving control surfaces.

The additional developments apart from the main focus also had a personal and scientifically important component. Swarm Intelligence for optimisation and the treatment of the geometrically exact beam were tasks to explore their potential as well as to take a closer look at their theory. Despite the conditions, not all investigations could be completed and some questions remain unanswered.

From a personal point of view, the stay was very successful. Contacts at the USAFA were refreshed. It is worth mentioning that the USAFA is a military institution in the USA. A large part of the activities at USAFA are sensitive and the exchange outside of the common topic is usually difficult or impossible. It remains to be seen whether a lively exchange in the form of secondments of individual persons will become active again on both sides.

List of Figures

List of Tables

List of Algorithms

Bibliography

[1] G. Beni and J. Wang. Swarm Intelligence in Cellular Robotic Systems. In P. Dario, G. Sandini, and P. Aebischer, editors, *Robots and Biological Systems: Towards a New Bionics?*, pages 703–712, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

[2] H. S. Bernardino, H. J. C. Barbosa, and L. G. Fonseca. Surrogate-assisted clonal selection algorithms for expensive optimization problems. *Evol. Intel.*, 4(2):81–97, 2011. doi:10.1007/s12065-011-0056-1.

[3] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swam Intelligence: From Natural to Artificial Systems*. Oxford University Press, 1st edition, 1999.

[4] C. A. Coello Coello and M. Salazar Lechuga. MOPSO: a proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, volume 2, pages 1051–1056, 2002. doi:10.1109/CEC.2002.1004388.

[5] I. D. Couzin, J. Krause3, R. James, G. D. Ruxton, and N. R. Franks. Collective memory and spatial sorting in animal groups. *Journal of Theoretical Biology*, 218(1):1–11, 2002. doi:10.1006/jtbi.2002.3065.

[6] M. Dorigo. *Optimization, Learning and Natural Algorithms*. PhD thesis, Politecnico di Milano, Italy, 1992.

[7] R. C. Eberhart and Y. Shi. Comparison between Genetic Algorithms and Particle Swarm Optimization. In V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, editors, *Evolutionary Programming VII*, pages 611–616. Springer Berlin Heidelberg, 1998. doi:10.1007/BFb0040812.

[8] R. C. Eberhart and Y. Shi. Particle Swarm Optimization: Developments, Applications and Resources. In *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, volume 1, pages 81–86, 2002. doi:10.1109/CEC.2001.934374.

[9] A. G. Gad. Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. *Archives of Computational Methods in Engineering*, 29(5):2531–2561, 2022. doi:10.1007/s11831-021-09694-4.

[10] A. Gálvez and A. Iglesias. Particle swarm optimization for non-uniform rational B-spline surface reconstruction from clouds of 3D data points. *Information Sciences*, 192:174–192, 2012. Swarm Intelligence and Its Applications. doi:10.1016/j.ins.2010.11.007.

[11] S. Grazioso, G. Di Gironimo, and B. Siciliano. A Geometrically Exact Model for Soft Continuum Robots: The Finite Element Deformation Space Formulation. *Soft Robotics*, 6(6):790–811, 2019. doi:10.1089/soro.2018.0047.

[12] F. Han and Q. Liu. A diversity-guided hybrid particle swarm optimization based on gradient search. *Neurocomputing*, 137:234–240, 2014. Advanced Intelligent Computing Theories and Methodologies. doi:10.1016/j.neucom.2013.03.074.

[13] Z. He, T. Liu, and H. Liu. Improved particle swarm optimization algorithms for aerodynamic shape optimization of high-speed train. *Advances in Engineering Software*, 173:103242, 2022. doi:10.1016/j.advengsoft.2022.103242.

[14] H. Hesse and R. Palacios. Consistent structural linearisation in flexible-body dynamics with large rigid-body motion. *Computers & Structures*, 110-111:1–14, 2012. doi:10.1016/j.compstruc.2012.05.011.

[15] D. H. Hodges. A mixed variational formulation based on exact intrinsic equations for dynamics of moving beams. *International Journal of Solids and Structures*, 26(11):1253–1273, 1990. doi:10.1016/0020-76839090060-9.

[16] D. H. Hodges. Geometrically Exact, Intrinsic Theory for Dynamics of Curved and Twisted Anisotropic Beams. *AIAA Journal*, 41(6):1131–1137, 2003. doi:10.2514/2.2054.

[17] T. Kalaiselvi, P. Nagaraja, and Z. A. Basith. A Review on Glowworm Swarm Optimization. *International Journal of Information Technology*, 3(2):49–56, 2017.

[18] D. Karaboğa. An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering, 2005.

[19] D. Karaboğa and B. Akay. A comparative study of Artificial Bee Colony algorithm. *Applied Mathematics and Computation*, 214(1):108–132, 2009. doi:10.1016/j.amc.2009.03.090.

[20] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995. doi:10.1109/ICNN.1995.488968.

[21] M. Khurana and K. Massey. Swarm algorithm with adaptive mutation for airfoil aerodynamic design. *Swarm and Evolutionary Computation*, 20:1–13, 2015. doi:10.1016/j.swevo.2014.10.001.

[22] M. Khurana, H. Winarto, and A. Sinha. Application of Swarm Approach and Artificial Neural Networks for Airfoil Shape Optimization. In *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, volume AIAA 2008-5954, 2008. arXiv:https://arc.aiaa.org/doi/pdf/10.2514/6.2008-5954, doi:10.2514/6.2008-5954.

[23] M. S. Khurana. Application of an Hybrid Optimization Approach in the Design of Long Endurance Airfoils. In *26$^{(}$th) International Congress of the Aeronautical Sciences*, 2008.

[24] J. Krause and G. D. Ruxton. *Living in Groups*. Oxford Series in Ecology and Evolution. OUP Oxford, 2002.

[25] K. N. Krishnanand and D. Ghose. Detection of multiple source locations using a glowworm metaphor with applications to collective robotics. In *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, pages 84–91, 2005. doi:10.1109/SIS.2005.1501606.

[26] K. N. Krishnanand and D. Ghose. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. *Swarm Intelligence*, 3(2):87–124, 2009. doi:10.1007/s11721-008-0021-5.

[27] G. Leugering, C. Rodriguez, and Y. Wang. Nodal profile control for networks of geometrically exact beams. *Journal de Mathématiques Pures et Appliquées*, 155:111–139, 2020. doi:10.1016/j.matpur.2021.07.007.

[28] X. Li and L. Goa. Particle swarm optimization hybridized with genetic algorithm for uncertain integrated process planning and scheduling with interval processing time. *Computers & Industrial Engineering*, 135:1036–1046, 2019. doi:10.1016/j.cie.2019.04.028.

[29] M. M. Millonas. Swarms, Phase Transitions, and Collective Intelligence; and a Nonequilibrium Statistical Field Theory of Swarms and Other Spatially Extended Complex Systems, 1993. doi:10.48550/arXiv.adap-org/9306002.

[30] R. Palacios, J. Murua, and R. Cook. Structural and Aerodynamic Models in Nonlinear Flight Dynamics of Very Flexible Aircraft. *AIAA Journal*, 48(11):2648–2659, 2010. doi:10.2514/1.J050513.

[31] B. L. Partridge. The Structure and Function of Fish Schools. *Scientific American*, 246(6):114–123, 1982.

[32] B. L. Partridge and T. J. Pitcher. The sensory basis of fish schools: Relative roles of lateral line and vision. *Journal of Comparative Physiology*, 135(4):315–325, 1980. doi:10.1007/BF00657647.

[33] E. Reissner. On finite deformations of space-curved beams. *Z. angew. Math. Phys.*, 32:734–744, 1981. doi:10.1007/BF00946983.

[34] C. Rodriguez. *Control and stabilization of geometrically exact beams*. PhD thesis, Friedrich-Alexander-University, Erlangen-Nürnberg, 2021.

[35] C. Rodriguez. Networks of geometrically exact beams: Well-posedness and stabilization. *Mathematical Control and Related Fields*, 12(1):49–80, 2021. doi:10.3934/mcrf.2021002.

[36] C. Rodriguez and G. Leugering. Boundary feedback stabilization for the intrinsic geometrically exact beam model. *SIAM Journal on Control and Optimization*, 58(6):3533–3558, 2020. arXiv:https://doi.org/10.1137/20M1340010, doi:10.1137/20M1340010.

[37] R. B. Seidler, A. R. Huebner, B. Ernst, and M. Widhalm. DLR Results for the 1.5 AIAA Stability and Control Prediction Workshop Investigating a Wing and Wing-Tail Section of the Common Research Model. *AIAA 2023-4397*, 20213. doi:10.2514/6.2023-4397.

[38] E. Shaw. Schooling Fishes: The school, a truly egalitarian form of organization in which all members of the group are alike in influence, offers substantial benefits to its participants. *American Scientist*, 66(2):166–175, 1978.

[39] J. C. Simo, J. E. Marsden, and P. S. Krishnaprasad. The Hamiltonian Structure of Nonlinear Elasticity: The Material and Convective Representations of Solids, Rods, and Plates. *Arch. Rational Mech. Anal.*, 104:125–183, 1988. doi:10.1007/BF00251673.

[40] A. Slowik, editor. *Swarm Intelligence Algorithms: A Tutorial*. CRC Press, 1st edition, 2020. doi:10.1201/9780429422614.

[41] A. Slowik, editor. *Swarm Intelligence Algorithms: Moodifications and Applications*. CRC Press, 1st edition, 2020. doi:10.1201/9781003046882.

[42] L. Wang, X. Liu, N. Renevier, M. Stables, and G. M. Hall. Nonlinear aeroelastic modelling for wind turbine blades based on blade element momentum theory and geometrically exact beam theory. *Energy*, 76:487–501, 2014. doi:10.1016/j.energy.2014.08.046.

[43] M. Widhalm and P. Bekemeyer. Nonlinear Low-Dimensional Model Order Reduction with Subspace Interpolation for Gust Applications with the Linear Frequency Domain Approach. *AIAA 2023-3947*, 2023. doi:10.2514/6.2023-3947.

[44] M. Widhalm, M. Stradtner, A. Schuette, M. Ghoreyshi, A. Jirasek, and J. Seidel. Comparison of Reduced Order Models for Evaluating Stability Derivatives for the DLR-F22 ONERA model. *AIAA 2023-4199*, 2023. doi:10.2514/6.2023-4199.

[45] Y. Zhou, B. Rao, and W. Wang. UAV Swarm Intelligence: Recent Advances and Future Trends. *IEEE Access*, 8:183856–183878, 2020. doi:10.1109/ACCESS.2020.3028865.