Optimising Satellite Geometries to Minimise Drag in Very Low Earth Orbits

A thesis submitted to the University of Manchester for the degree of
Doctor of Philosophy
in the Faculty of Science and Engineering

2022

Luciana A. Sinpetru
Department of Mechanical, Aerospace and Civil Engineering

Contents

Word Count: 43446

List of figures

List of tables

List of publications

Papers intended for publication in peer-reviewed journals, where the author of this thesis is the lead author and main contributor:

1. Luciana Sinpetru, N. H. Crisp and P. C. Roberts, ”Towards a clear definition of very low Earth orbits”, TBD (ready for submission), 2022. (**Paper I**)
2. Luciana Sinpetru, N. H. Crisp, D. Mostaza-Prieto, S. Livadiotti, and P. C. Roberts, “ADBsat: Methodology of a novel panel method tool for aerodynamic analysis of satellites,” *Computer Physics Communications*, vol. 275, p. 108326, 2022, `https://doi.org/10.1016/j.cpc.2022.108326`, ISSN: 0010-4655. (**Paper II**)
3. Luciana Sinpetru, N. H. Crisp, P. C. Roberts, V. Sulliotti-Linner, V. Hanessian, G. H. Herdrich, F. Romano, D. Garcia-Almiñana, S. Rodríguez-Donaire, and S. Seminari, “ADBsat: Verification and validation of a novel panel method for quick aerodynamic analysis of satellites,” *Computer Physics Communications*, vol. 275, p. 108327, 2022, `https://doi.org/10.1016/j.cpc.2022.108327`, ISSN: 0010-4655. (**Paper III**)

Compliance with the DISCOVERER project agreement on publishing and dissemination requires the author list of any documentation produced within the research group to comprise all the members of the project at the time of writing, where possible. For the peer-reviewed and conference papers displayed in the following, the contribution of the author of this manuscript should not be considered of any significance:

1. P. C. E. Roberts, N. H. Crisp, V. A. Oiko, et al., “Early Results from the DISCOVERER Project,” in *72nd International Astronautical Congress (IAC), Dubai, United Arab Emirates, 25-29 October 2021.*, Oct. 2021.
2. N. H. Crisp, A. Macario-Rojas, P. C. E. Roberts et al., ”Launch, Operations, and First Experimental Results of the Satellite for Orbital Aerodynamics Research (SOAR)” in *72nd International Astronautical Congress (IAC), Dubai, United Arab Emirates, 25-29 October 2021.*, Oct. 2021.
3. F. Romano, J. Espinosa-Orozco, M.Pfeiffer et al., ”Intake design for an Atmosphere-Breathing Electric Propulsion System (ABEP)”, *Acta Astronautica*, vol. 187, p. 225-235, 2021, `https://doi.org/10.1016/j.actaastro.2021.06.033`.

4. N. H. Crisp, P. C. E. Roberts, S. Livadiotti et al., ”In-orbit aerodynamic coefficient measurements using SOAR (Satellite for Orbital Aerodynamics Research)”, *Acta Astronautica*, vol. 180, p. 85-99, 2021, `https://doi.org/10.1016/j.actaastro.2020.12.024`.

5. F. Romano, Y.-A. Chan, G. Herdrich et al., ”Design, Set-Up, and First Ignition of the RF Helicon-based Plasma Thruster”, in *SPACE PROPULSION 2020+1, 17-19 March 2021*, Mar. 2021.

6. S. Livadiotti, N. H. Crisp, P. C. E. Roberts et al., ”A review of gas-surface interaction models for orbital aerodynamics applications”, *Progress in Aerospace Sciences*, vol. 119, 2020, `https://doi.org/10.1016/j.paerosci.2020.100675`, ISSN 0376-0421.

7. F. Romano, Y.-A. Chan, G. Herdrich et al., ”RF Helicon-based Plasma Thruster (IPT): Design, Set-up, and First Ignition.”, in *71st International Astronautical Congress (IAC), The CyberSpace Edition, 12-14 October 2020*, Oct. 2020.

8. P. C. E. Roberts, N. H. Crisp, S. Edmonson et al., ”DISCOVERER: Developing Technologies to Enable Commercial Satellite Operation in Very Low Earth Orbit”, in *71st International Astronautical Congress (IAC), The CyberSpace Edition, 12-14 October 2020*, Oct. 2020.

9. N. H. Crisp, A. Macario-Rojas, P. C. E. Roberts et al., ”Investigation of Novel Drag-Reducing and Atomic Oxygen Resistant Materials in Very Low Earth Orbit using SOAR (Satellite for Orbital Aerodynamics Research)”, in *71st International Astronautical Congress (IAC), The CyberSpace Edition, 12-14 October 2020*, Oct. 2020.

10. V. T. A. Oiko, P. C. E. Roberts, A. Macario-Rojas et al., ”GROUND-BASED EXPERIMENTAL FACILITY FOR ORBITAL AERODYNAMICS RESEARCH: DESIGN, CONSTRUCTION AND CHARACTERISATION”, in *71st International Astronautical Congress (IAC), The CyberSpace Edition, 12-14 October 2020*, Oct. 2020.

11. N. H. Crisp, P. C. E. Roberts, S. Livadiotti et al., ”The benefits of very low earth orbit for earth observation missions”, *Progress in Aerospace Sciences*, vol. 117, p. 100619, 2020, `https://doi.org/10.1016/j.paerosci.2020.100619`.

12. Rodriguez-Donaire, S. et al., ”Earth Observation Technologies: Low-End-Market Disruptive Innovation”, in V. Demyanov, J. Becedas (eds.), *Satellites Missions and Technologies for Geosciences*, IntechOpen, London, `https://doi.org/10.5772/intechopen.90923`.

13. F. Romano, Y.-A. Chan, G. Herdrich et al., ”RF Helicon-based Inductive Plasma Thruster (IPT) Design for an Atmosphere-Breathing Electric Propulsion system

(ABEP)”, *Progress In Aerospace Science*, vol. 176, p. 476-483, 2020, `https://doi.org/10.1016/j.actaastro.2020.07.008`

14. V. Cañas Muñoz, D. Gonzáles, J. Becedas et al., ”Attitude control for satellites flying in VLEO using aerodynamic surfaces”, *JBIS - Journal of the British Interplanetary Society*, vol. 73, no. 3, p. 103-112, 2020.

15. D. Gonzáles, V. Cañas Muñoz, J. Becedas et al., ”Modelling and Simulation of Very Low Earth Orbits”, in *8th European Conference for Aeronautics and Space Sciences (EUCASS)*, 2019.

16. F. Romano, G. Herdrich, P. C. E. Roberts et al., ”Inductive Plasma Thruster (IPT) for an Atmosphere Breathing Electric Propulsion System: Design and Set in Operation”, in *36th International Electric Propulsion Conference, Vienna, Austria, September 15-20, 2019*, Sep. 2019

17. S. Livadiotti, N. H. Crisp, P. C. E. Roberts et al.,”Concepts and Applications of Aerodynamic Attitude and Orbital Control for Spacecraft in Very Low Earth Orbit.” in *70th International Astronautical Congress (IAC), Washington D.C., United States, 21-25 October 2019*, Oct. 2019.

18. N. H. Crisp, S. Livadiotti, P. C. E. Roberts et al., ”Demonstration of Aerodynamic Control Manoeuvres in Very Low Earth Orbit using SOAR (Satellite for Orbital Aerodynamics Research)” in *70th International Astronautical Congress (IAC), Washington D.C., United States, 21-25 October 2019*, Oct. 2019.

19. V. A. Oiko, P. C. E. Roberts, S. Worrall et al., ”ROAR – A Ground-Based Experimental Facility for Orbital Aerodynamics Research”, in *70th International Astronautical Congress (IAC), Washington D.C., United States, 21-25 October 2019*, Oct. 2019.

20. P. C. E. Roberts, N. H. Crisp, F. Romano et al., ”KEYNOTE: DISCOVERER – Making Commercial Satellite Operations in Very Low Earth Orbit a Reality”, in *70th International Astronautical Congress (IAC), Washington D.C., United States, 21-25 October 2019*, Oct. 2019.

21. F. Romano, G. Herdrich, P. C. E. Roberts et al., ”Inductive Plasma Thruster (IPT) design for an Atmosphere-Breathing Electric Propulsion System (ABEP)”, in *70th International Astronautical Congress (IAC), Washington D.C., United States, 21-25 October 2019*, Oct. 2019.

22. V. A. Oiko, P. C. E. Roberts, S. Edmondson et al., ”Design and Development of a Hyper-Thermal Atomic Oxygen Wind Tunnel Facility”, in *Proceedings of the 14th ISMSE & 12th ICPMSE, Biarritz, France, 1 to 5 October 2018*, Oct. 2018.

23. D. Gonzáles, V. Cañas Muñoz, J. Becedas et al., ”Scilab and Xcos for VLEO satellites modelling”, in *Scilab Conference, Germany, 14 October 2019*, Oct 2019.

Acronyms and Symbols

Acronyms

AOA	Angle of attack
AOS	Angle of sideslip
ASCII	American standard code for information Interchange
CAD	Computer-aided design
CHAMP	Challenging Minisatellite Payload
CLL	Cercignani-Lampis-Lord gas-surface interaction model
CoG	Centre of gravity
CoP	Centre of pressure
DSMC	Direct simulation Monte Carlo
DTM	Drag temperature atmospheric model
EO	Earth observation
F&T	Front and tail optimisation
FF	Frontal face optimisation
FMF	Free-molecular flow
GA	Genetic algorithm
GEO	Geostationary orbit
GOCE	Gravity Field and Steady-State Ocean Circulation Explorer
GPLv3	General public license version 3
GPS	Global positioning system
GRACE	Gravity Recovery and Climate Experiment
GRACE-FO	Gravity Recovery and Climate Experiment Follow-On
GSIM	Gas-surface interaction model
HEO	High Earth orbit
INMS	Ion and neutral mass spectrometer
IRI	International Reference Ionosphere
JB	Jacchia-Bowman atmospheric model
LEO	Low Earth orbit
MATLAB	Matrix Laboratory program
MEO	Medium Earth orbit
MO	Multi-objective
MRC	Moment reference centre
NaN	Not a number

NRLMSISE-00	United States Naval Research Laboratory’s mass spectrometer and incoherent scatter radar atmospheric model
NSGA-II	Non-dominated sorting genetic algorithm II
SLATS	Super Low Altitude Test Satellite
SLEO	Super low Earth orbit
SO	Single objective
SOAR	Satellite for Orbital Aerodynamics Research
SRP	Solar radiation pressure
TF	Tail face optimisation
TPMC	Test particle Monte Carlo
ULEO	Ultra low Earth orbit
VLEO	Very low Earth orbit
WB	Whole body optimisation

Vectors

$\vec{C}_f$	vector of force coefficients
$\vec{C}_M$	vector of moment coefficients
$\vec{C}_M^{CoG}$	vector of moment coefficients translated to CoG
$\vec{e}_r$	unit vector pointing to satellite from Earth
$\vec{e}_s$	unit vector pointing to perturbing body from Earth
$\vec{F}$	force
$\vec{n}_i$	surface normal of mesh element
$\vec{n}_v$	velocity vector
$\vec{r}$	position vector
$\ddot{\vec{r}}$	acceleration
$\vec{r}_i$	vector from geometric moment reference point to barycentre of panel
$\vec{V}$	velocity vector
$\vec{x}$	candidate solution (vector input to objective function)
$\vec{\tau}_i$	unit vector in direction of shear stress

Roman symbols

A_p	Atmospheric magnetic index
a_p	Surface area of triangular plate
A_{proj}	Projected area
A_{ref}	Reference area
b_p	Barycentre of triangular plate
c	Speed of light
$c_1,\ c_2$	Convexity measures

C_d	Drag coefficient
C_{frac}	Crossover fraction
c_i^{eq}	i equality constraints
c_j^{ineq}	j inequality constraints
C_l	Lift coefficient
C_m	Pitch moment coefficient
C_n	Yaw moment coefficient
C_p	Pressure coefficient
C_{pi}	Pressure coefficient of triangular panel i
C_R	Radiation coefficient
C_r	Solar reflectivity
C_{tol}	Constraint tolerance
c_v	Maximum percentage reduction in volume allowed
C_τ	Shear stress coefficient
c_ξ	Allowed percentage margin on aspect ratio
d_i	Distance from voxel to axis i
d_u	Distance of candidate solution objectives to utopia point
E_i	Energy of incoming particles
E_r	Energy of reflected particles
E_w	Energy that reflected particles would have had, had they been completely accommodated
$F_{10.7}$	Solar radio flux at 10.7 cm in units of $1 \times 10^{-22} W m^{-2} Hz^{-1}$
F_d	Atmospheric drag force
F_{SRP}	Solar radiation pressure force
F_{tol}	Function tolerance
G	Gravitational constant
g	Acceleration due to gravity
h	Step size of central difference formula
I_j	Moment of inertia about axis j
I_{sp}	Specific impulse
k	Rank of GA solutions
K_e	Total linear kinetic energy of simulation
K_n	Knudsen number
L	Characteristic length scale of the object
L_{bw}	Transformation matrix from wind frame to body frame
L_{gb}	Transformation matrix from body frame to geometric frame
L_{gw}	Transformation matrix from wind frame to geometric frame
L_{ref}	Reference length
M_P	Mass of perturbing body P
M_{avg}	Average mass of gas species in mixture

M_f	Initial propellant mass fraction
M_{gen}	Maximum number of generations
m_i	Mass of a voxel
m_j	Mass of gas species
m_s	Mass of satellite
N	Total number of simulation particles
n_0	Number density of the atmosphere
n_{CH}	Number of voxels in convex hull
n_p	Outward surface normal
n_s	Number of voxels in shape
n_v	Number of voxels in shape
P_{frac}	Pareto fraction
p_1, p_2	Parents used by the genetic algorithm crossover
$p(x)$	Pressure as a function of x co-ordinate
r	Magnitude of satellite position vector (geocentric coordinates)
R	Semi-major axis of orbit
r_{bs}	Ratio of maximum body width to maximum skirt width
R_{CoG}	Position of CoG
r_i	Position of a voxel
R_p	Radius of the photosphere
s_P	Magnitude of position vector of perturbing body P (geocentric coordinates)
s	Speed ratio
s_{min}	Minimum distance between satellite and sun
t_c	Convergence time
T_i	Incident temperature
T_{inf}	Ambient temperature
T	Temperature of the sun
T_w	Wall temperature
v	Speed of satellite relative to atmosphere
V	Incident velocity
V_{inf}	Free-stream velocity
V_w	Average normal velocity of diffusely reflected molecules
v’	Most probable speed of particle in Maxwellian distribution
$\overline{v}$	Average volume of initial population
v^s	Volume of candidate solution s
x_i	X co-ordinate of barycentre of triangular panel i

Greek and mathematical symbols

α	angle of attack

α_E	energy accommodation coefficient
α_N	normal energy accommodation coefficient
α_w	weight of location on convexity measure
β	angle of sideslip
$\beta_j,\ \gamma_j,\ \zeta_j, \delta_j$	parameters of gas species j for CLL model
β_w	fraction of shape considered for convexity measure, towards front
γ_w	multiplicative factor on CoM for convexity measure
δ	angle between flat plate and flow
Δt	change in time
Δv	change in velocity
Δx	Mesh cell side length
Δ_{PG}	relative x position of CoP and CoG
Δ_{PG_y}	relative y position of CoP and CoG
Δ_{PG_z}	relative z position of CoP and CoG
θ	angle between $\vec{e}_s$ and $\vec{e}_r$
λ	mean free path
Λ	number of candidate solutions needed for tournament selection
μ	mean of Gaussian
μ_E	gravitational parameter of the Earth
ξ_{ij}	aspect ratio of axes i and j
$\overline{\xi_{ij}}$	average aspect ratio in axes i and j of initial set
ξ_{ij}^s	aspect ratio of axes i and j of candidate solution s
ρ	density of atmosphere
ρ_n	DSMC simulation particle density
σ	standard deviation
σ_N	normal momentum accommodation coefficient
σ_{SF}	Stefan-Boltzmann constant
σ_T	tangential momentum accommodation coefficient
τ	collision timescale
ϕ	number of triangular plates in a mesh
χ_j	gas species mole fraction
ψ	number of points selected for optimisation
Ψ	convexity
ω	angular velocity of a circular orbit
ω_w	weight employed in weighted sum
Ω_{m_α}	Controllability parameter based on pitch moment
Ω_{n_β}	Controllability parameter based on yaw moment
∇_x	gradient vector

Abstract

Very low Earth orbit (VLEO) satellite flight, at less than around 450 km altitude, is becoming increasingly popular. It affords numerous benefits such as increased payload resolution and assured end-of-life de-orbiting due to the presence of atmospheric drag. However, if unaccounted for in mission design, drag will also prematurely de-orbit the satellite. Thus, it is advantageous to reduce the drag force on a satellite as much as possible. One of the factors significantly affecting drag is the outer spacecraft geometry. While previous attempts at drag-oriented design have been made, they considerably simplify the case either by looking at a reduced set of parameters or assuming axisymmetry in reducing a 3D case to a 2D representation. The aim of this thesis is to use computational optimisation in 3D to improve the current understanding of the influence of satellite body geometry on the quality of flight in VLEO. This is primarily achieved through drag reduction, but also through the analysis of other realistic mission requirements such as payload carrying ability and controllability.

Firstly, recent satellite launches and important missions to VLEO are examined. A clear year-on-year increase in satellite flight to this orbital range is seen, notwithstanding the influence of the Coronavirus pandemic which delayed most launches. This increase is analysed in the context of the current industry trends toward standardisation and miniaturisation. While these trends have some observable benefits, most recent missions to VLEO have been individually engineered in order to maximise their science potential. Thus, bespoke engineering is identified to be appropriate for the objective at hand. An overview of the literature regarding VLEO is presented, both current and historical, and a summary of the atmospheric models which can recreate the conditions therein.

Computational drag analysis in VLEO is itself challenging. A panel method program with shadowing analysis called ADBSat, developed and refined in previous studies at Manchester, is employed to this end. The methodology and physical models used therein are comprehensively detailed. A thorough validation is presented, proving good accuracy for its intended purpose. While not universally accurate, particularly for concave geometries, it is fast enough to remove the need for interpolation of drag data, thereby reducing the number of assumptions and simplifications necessary. Another advantage is its implementation in MATLAB, which offers a number of beneficial options for performing optimisation. The limitations of ADBSat can be mitigated through appropriate constraints on the optimisation framework.

Having characterised ADBSat, it is integrated into an optimisation framework which

uses the genetic algorithm (GA) metaheuristic to improve on satellite body designs. A thorough overview of the many choices for computational optimisation that exist is provided. The specific characteristics of the GA which make it particularly suitable for this problem are identified: its well-tested implementation in MATLAB, its ability to handle problems with a high number of objectives, its suitability for parallelisation, and its population-based approach. The optimisation objectives are drag coefficient multiplied by frontal area, volume, the relative position of the centres of gravity and pressure, and two controllability characteristics. Constraints on the problem include aspect ratio, volume, plate intersection, and shape convexity, with the aim of maintaining feasibility of results both in terms of real-life applications and in terms of analysis by ADBSat.

The novel method of employing the GA in conjunction with ADBSat has identified a wide variety of suitable shapes which exhibit trade-offs across the multiple objectives. A general improvement in all five optimisation objectives is seen across the population. Considerable advantages are seen when comparing the multi-objective optimisation to the single-objective drag case: improved controllability characteristics and the choice of more payload capacity. Optimising the drag independent of realistic requirements, while mathematically advantageous, can counter-intuitively be detrimental to mission requirements. Even with sophisticated simulations, engineering expertise is still invaluable for aerodynamic design. In the future, it is hoped that such simulations will be a useful tool in satellite design for flight to VLEO.

Declaration of originality

I hereby confirm that no portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright statement

Acknowledgements

Little acts of kindness,
Nothing do they cost;
Yet, when they are wanting,
Life’s best charm is lost.
Little acts of kindness,
Richest gems of earth,
Though they seem but trifles,
Priceless is their worth.

- *Anonymous*

In no particular order...

To Mum & Dad:

Thank you for unwavering support, unconditional love, and all the uncountable things you do for me. Thank you for phone calls when I needed them most. *Vă iubesc mult!*

To Nick & Peter:

Thank you for guidance, leadership, and encouragement. Thank you for understanding when I needed gentleness and when I needed tough love. Thank you for pushing me!

To Sahil:

Thank you for love, kindness, seeing me and understanding me. Thank you for delicious dinners, innumerable loads of laundry, and everything in between.

To Katie:

Thank you for standing beside me for over ten years through thick and thin, and for never losing compassion in the face of adversity. Thank you for being a true inspiration. *Go raibh maith agat, a chara!*

To Katrina, Max & Elvis:

Thank you for being a source of positivity, kindness, spare rooms to sleep in, people hugs and cat cuddles.

To Kate, Sabrina, Vitor, Claire & the whole DISCOVERER group:

Thank you for lunches, drinks, meetings, and for making my time on this team unforgettable.

To Brandon, Alex, Lisa & the cavers:

Thank you for support, laughs, workouts, and for getting muddy with me, you bunch of weirdos!

To Margarita, Sarah, Simon, James, James, Ben, Kirsten & Niamh:

Thank you for board games, cups of tea, listening to me rant, accepting me as I am into your lives, and always picking up where we left off!

To Coca, Silvia & the extended family:

Thank you for knowing me since the day I was born. *La mulţi ani şi la mulţi bani!*

To all others who extended me acts of kindness, both great and small:

I am truly lucky to have you all in my life. I appreciate you more than you know!

Chapter 1

Introduction

For over six decades, spaceflight has fascinated earthborn explorers. Since the launch of Sputnik 1 in 1957, years of research and billions of pounds have been invested in advancing satellite flight. The reasons for this level of investment of both man-hours and capital are varied. Satellites can provide useful services, in the form of GPS or communications relay. Other times, they serve to gather data which expands our understanding of the Earth, or of space. Regardless of purpose, most satellites have at minimum a cost measured in tens of thousands of man hours spent on mission design or of pounds spent on technology. Even with the advent of disruptive technologies such as miniaturisation of electronic components, it is imperative that each mission is run in the most efficient way possible. Very recently, one proposed way of achieving this has been flying satellites in a lower orbit, closer to the Earth.

These orbits are called very low Earth orbits (VLEO) and commonly defined as 100 km to 450 km in altitude. They are a subdivision of the low Earth orbit range, one of the four generally accepted orbital regimes:

- Low Earth Orbit (LEO): 100 km to 2000 km
- Medium Earth Orbit (MEO): 2000 km to 35 786 km
- Geostationary Orbit (GEO): 35 786 km
- High Earth Orbit (HEO): Altitudes higher than 35 786 km

VLEO missions can have extensive benefits, in particular for the purposes of Earth observation. Examples of uses including shipping lane management, fishing surveys, forestry management, weather forecasting, and natural disaster support. The primary benefit is that the same payload orbiting at a lower altitude will yield a better data resolution, with potentially more accurate positioning. Alternatively, a reduction in payload power will yield data of the same detail as a larger apparatus orbiting at a higher altitude. With this decrease in power requirements comes a desirable decrease in characteristics such as payload mass, size, and cost [1]. All these factors together reduce the cost of manufacturing, launch and operation [2].

An area which has seen increased interest in the past decade, and shows promising growth for the future, is that of constellation missions. VLEO has also shown some potential benefits here, as the decreased altitude can lead to increased launch mass capabilities. Coupled with the decreased mass of each individual payload, this points to a reduction in costs of launch and the capability to inject the swarm into orbit more efficiently than at a higher altitude [2]. End-of-life disposal is also assured in VLEO, in line with the requirement that all modern satellites must de-orbit 25 years after the end of their usable lifetime. Due to the increased atmospheric drag, once thruster systems are shut off, the satellite will begin to fall towards the surface of the Earth. Eventually, it will burn up in the atmosphere. With the number of defunct satellites currently outnumbering operational ones in the higher reaches of LEO, debris mitigation and avoidance is an important issue in higher orbits [3], [4]. This is not a concern in the lower orbital altitudes that make up VLEO [2].

However, these benefits come with some added challenges. Chief among these is the increased influence of atmospheric drag. While useful in debris mitigation, it also significantly reduces the mission lifetime of a satellite. Accounting for drag is arguably one of the most important aspects of designing a mission in VLEO to fulfil its objectives, and survive to its optimum lifetime with minimum fuel expenditure. Mitigating drag forces will lead to important benefits such as longer mission lifetime, smaller fuel requirements, and potentially better control over satellite qualities such as aerostability and agility. The proposed method towards this end is purposefully designing the outer spacecraft geometry to lower the drag force experienced due to orbital motion. Geometry design and optimisation has, of course, been commonplace in many fields dealing with continuum flow. For example, in aircraft design, the addition and subsequent optimisation of winglets to aircraft wing tips has resulted in a desirable increase in lift-to-drag ratio, while keeping wing bending moments within acceptable limits [5]. Computational optimisation of airfoil shapes has been established for many decades, also with the principal goal of maximising the lift-to-drag ratio [6]–[10]. Autonomous underwater vehicles also benefit from the computational optimisation of their body shape to increase lift-to-drag ratio [11]. A complementary topic to vehicle optimisation is that of wind turbine optimisation, which also relies on airfoils, but adapts them with goals such as minimising cost, maximising annual energy production, and minimising turbine blade mass [12]–[16]. Optimisation in continuum flow has proven useful time and again, with many more examples than could be concisely listed here.

In continuum flow, the interactions between the molecules and the resultant boundary effects are described by the well-known Navier-Stokes-Fourier equations. A high-magnitude drag force is produced on any moving object in this relatively dense flow. This does not happen in the upper end of LEO, where the atmosphere is virtually nonexistent and thus drag forces are negligible. However, the environment in VLEO is different to both of these regimes. There is a drag force due to the rarefied gas, which is non-negligible, but

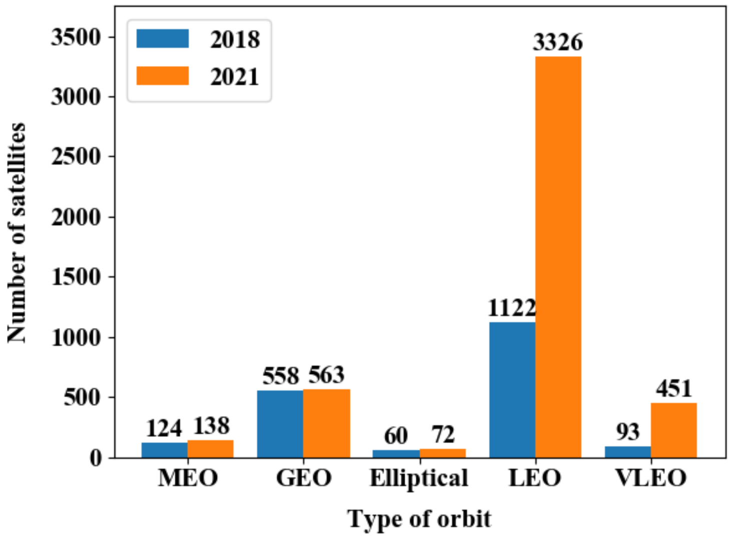


Figure 1.1. Trends in the number of operational satellites over the period December 2018 - September 2021. Data from the Union of Concerned Scientists [17].

gas-surface interactions dominate rather than the gas-gas interactions experienced in continuum flow. These particular conditions mean that specialised analysis is required, with results from continuum conditions not applicable to VLEO conditions. As more and more satellites are launched and operated in VLEO, it is imperative that this analysis is performed sooner rather than later.

1.1 Current Trends in VLEO Satellite Missions

Expert economic analysis of space market trends points to the growth year-on-year of investment into this sector [18]. This is also reflected in the growth of the number of operational satellites over the previous three years. The distribution of operational satellites in December 2018 and September 2021 is shown as a bar graph in fig. 1.1 [17]. Of note is the large increase in the total number of operational satellites in orbit, from 1957 to 4550. This growth of 228% reflects the increasing investment into the space sector and the increase in confidence of satellite data accuracy and utility. Another important factor is the noted increase in satellite operation in VLEO, which, for the purposes of examining the database, is defined as satellites which are in a roughly circular orbit ($e < 0.03$) with a perigee altitude under 450 km. The Starlink constellation satellites, which are known to be launched to VLEO then raised to their operational altitudes of 550 km and above, are not considered to be in VLEO for the purposes of this analysis. The percentage of operational satellites inserted to VLEO has more than doubled, from 4.75% in late 2018 to 9.9% in late 2021. The number of satellites in these orbits at the time of creating the database has almost quintupled, from 92 to 450. Additionally, it should be noted that VLEO represents under 1% of the orbital altitudes from the boundary of space at 100 km to geostationary orbits at an altitude of 35 785 km. The fact that almost one in every ten operational satellites are inserted to this regime indicates that there is a high demand for VLEO missions at the present time. In line with the expected continued growth in the space industry until at least 2030 [19], the number of satellites in VLEO is also likely to continue to grow over coming years.

With this growth of the number of satellites in VLEO, extending the mission lifetime or decreasing the cost of operation by lowering fuel requirements is proving increasingly attractive. This is normally done in one of two ways: computationally or experimentally. Both methods require a high number of assumptions with regards to the atmospheric density and gas-surface interaction model (GSIM) parameters. However, the information gathered can still prove useful. Computationally, it is possible to use simulation to reproduce the mission scenario and calculate the drag coefficient. This requires detailed knowledge of the geometric properties of the satellite, but can be performed relatively quickly for a range of different mission scenarios and satellite geometries. Experimentally, studying the rate of orbital decay of a satellite can lead to conclusions about its drag properties. Crucially, the satellite does not need to be explicitly engineered for drag analysis. Any satellite which exhibits a measurable decay can have its drag coefficient calculated. On the one hand, it is possible to look at the drag coefficients of many more satellites than have been engineered specifically for this purpose. On the other, these values often show a high degree of uncertainty, particularly when the attitude of the spacecraft is evolving in some unknown manner. The specific influence of different materials and shapes may also be difficult to characterize accurately when such characteristics were not studied ahead of time.

Some recent missions to VLEO which all involve a degree of drag analysis are listed below. The dates given are the launch and decay dates, and the orbital altitudes given refer to perigee and apogee.

- **Challenging Minisatellite Payload (CHAMP):** 15/07/2000 - 19/09/2010, 418 km to 474 km. One of the first modern satellites in VLEO, this mission was originally proposed in 1994. A small satellite of 522 kg weight, its main aims were to collect geophysical data in order to study the Earth's gravitational geopotential, magnetic geopotential, and atmospheric parameters [20]. Its operating orbit dropped to 300km for the latter half of its life. The drag coefficient of this satellite was needed in order to study the atmospheric density from on-orbit data. Some sources choose to use the standard value of 2.2 [21], while others attempt to determine a more accurate value [22].

- **Gravity Recovery and Climate Experiment (GRACE):** 17/03/2002 - 27/10/2017, 483 km to 508 km. This mission employed twin minisatellites of 487 kg each, flown approximately 200 km apart along the same orbit. Its main aim was to study the Earth's gravitational field, and its mass fluctuations [23]. The physical characteristics of the spacecraft are known to a high degree of precision, thus facilitating extremely accurate geometry modelling. Because of this, its drag coefficient has been extensively studied and used in model validation, as well as being used to derive atmospheric densities from accelerometer data [24].

- **Gravity Field and Steady-State Ocean Circulation Explorer (GOCE):** 17/03/2009 - 21/10/2013, 255 km to 255 km. This large satellite of 1077 kg was also designed to study anomalies in the gravitational field, with a view to gaining knowledge about the interior workings of the Earth [25]. This required very fine control of its speed and orbital trajectory, achieved through the use of an electric ion propulsion engine. Being in a particularly low orbit, its drag characteristics were studied mostly in the context of analysing the thrust required to maintain orbit [26].
- **Super Low Altitude Test Satellite (SLATS),** AKA **Tsubame:** 23/12/2017 - 01/10/2019, 180 km to 272 km. Increased interest in Earth observation at much lower orbits led to the design of this satellite, engineered to investigate the conditions of flight in VLEO [27]. It used a chemical thruster to achieve its initial operating altitude of 272 km after insertion at 628 km, and electric ion propulsion to counteract the drag force at this altitude and maintain orbit. It performed a controlled descent involving data acquisition at intermediate altitudes until its target altitude of 180km [28] was reached. While data analysis is from this mission is still ongoing, it will undoubtedly yield a wealth of information about this subject.
- **Gravity Recovery and Climate Experiment Follow-On (GRACE-FO):** 22/05/2018 - ongoing, 491 km to 511 km. GRACE-FO is the follow-on mission to GRACE, mentioned above, with similar mission parameters and objectives. The wide success of GRACE and many applications of its data led to a desire to extend the data set of the original mission beyond the original lifetime. GRACE-FO will undoubtedly lead to further useful data about flight in VLEO, in a similar manner to its predecessor [29], [30].
- **Satellite for Orbital Aerodynamics Research (SOAR):** 14/06/2021 - 23/03/2022, 421 km and below. This 3U CubeSat, flown by the University of Manchester, was designed to test the atmospheric conditions and satellite behaviour in VLEO. It had four fins coated with different materials, and a spectrometer on board. By exposing one material at a time to the flow, this mission aimed to gain knowledge about how the materials interact with the atmosphere. The spectrometer measured atmospheric characteristics such as density, flow composition, and velocity. It continued collecting data as its orbit decayed, until it burned up in the upper atmosphere. It is hoped that valuable information about the real conditions in VLEO will be extracted from the data collected on this mission [2], [31], [32].

It can be seen that many of these missions are attempting to establish the characteristics of the Earth. This is because the behaviour of the satellite (through the drag force it experiences) and atmospheric density are closely related. Missions to VLEO such as CHAMP, GRACE and GRACE-FO have the mission goal of experimentally measuring atmospheric density, and thus their drag coefficient was examined as a means to this end. CHAMP, GRACE, GOCE and GRACE-FO have also had the aim of examining the Earth's gravitational field, both on a local and global scale. VLEO is particularly suited to these applications because of its proximity to the Earth.

However, while the data collected from these missions has been invaluable to the scientific community, attempts to engineer spacecraft specifically for the drag conditions in VLEO have not been abundant. SLATS is one of the first satellites looking at the particular challenges to flying in VLEO. Unfortunately, it has proven difficult to find information about the mission outside of press releases. If the data from this satellite is made available to the global scientific community, it will most likely lead to fast advances in understanding the challenges to VLEO flight. SOAR's aim is also specifically to characterise the behaviour of satellites in the drag environment of VLEO, by investigating different materials and their influence on controllability. This will also prove invaluable to future missions and the development of materials particularly suited to flight in VLEO, research into which is currently ongoing at the University of Manchester.

Despite the identifiable tendency towards bespoke engineering in VLEO, current trends indicate movement towards standardisation across the industry rather than effectiveness for individual missions. The main emphasis has been cost reduction by large-scale manufacturing, with the trade-off of a possible lower efficiency for any one particular mission. CubeSats are perhaps the best, though not the only, example of this. Recent years have seen an explosion in the popularity of these cuboid nanosatellites, which are particularly attractive to short missions with limited objectives, such as those run by student clubs in universities and other educational institutions. Their small size also makes them ideal candidates for constellation flight, with the swarm performing the function of one or many larger traditional satellites, with the added benefit of easily being able to replace any malfunctioning swarm member. They are much less expensive to purchase, launch and operate than custom-made satellites. Standard payloads are readily available off-the-shelf for various missions [33], [34]. Figure 1.2 shows the number of nanosatellites launched per year until 2021, with the vast majority being CubeSats. The number launched quadrupled in 2013, and has remained high ever since.

Also of note is the influence of the Coronavirus pandemic on the launch numbers. 2020 shows only 163 launches while 2021 shows 750, with a prediction for this number to drop slightly in future years. This can no doubt be attributed to the halting of most day-to-day activities in 2020 due to the pandemic, as examining a similar figure from early 2019 shows a confirmed 367 planned launches for 2020 at the time. Conversely, the number of nanosatellites launched in 2021 far exceeded the prediction made in 2018. It can be inferred that the Coronavirus pandemic disrupted the number of nanosatellites launched

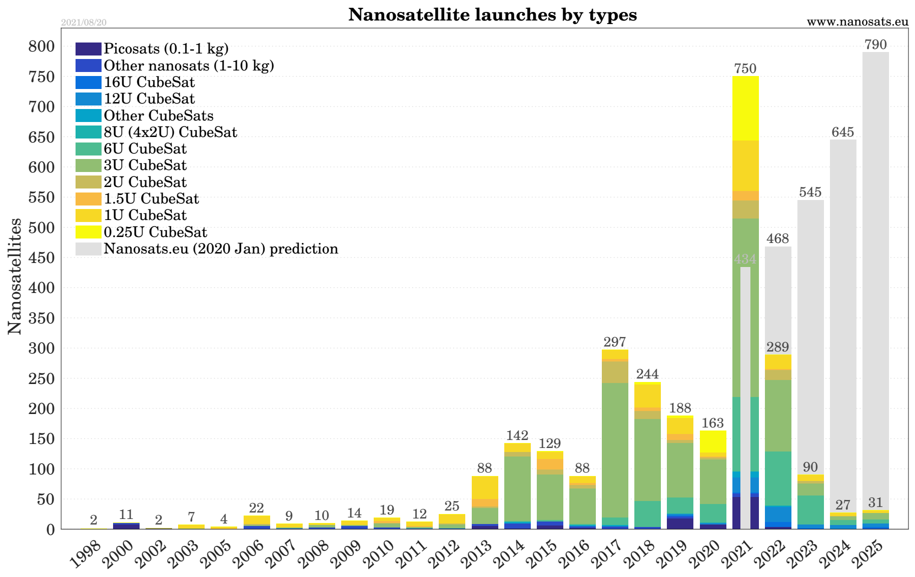


Figure 1.2. The number of nanosatellites launched per year [35]. Real data is represented up to 2021; the data for 2022 onwards are planned launches (in colour) and industry predictions (in grey).

in much the same way it disrupted many other aspects of society, pushing launches planned for 2020 into 2021.

The popularity of the standard CubeSat has fostered a need for miniaturisation of satellite parts without loss of performance. Some attempts have been undertaken to also standardise the design of the inner satellite systems, both for CubeSats and other satellite designs. For example, a system of picosatellites dubbed ”satlets” has been proposed which aggregate on-orbit into one system, to provide the full functionality of a larger satellite [36], [37]. Research is ongoing into this method of deploying a space system, with some companies already having completed on-orbit tests and some proof of concept tests [38].

More recently, a fully modular satellite design named the intelligent Computer Aided Satellite Designer (iCASD) has been proposed [39]. This method uses computer-aided decision-making to guide the design of a satellite from building blocks. While similar to the satlet method above, the addition of computerised decision-making in order to find an optimal building block configuration has the potential to find more efficient designs for the system as a whole. While this technology is still emerging, it may prove to be a viable alternative to CubeSats in the future.

However, one major drawback of these methods is the difficulty - or, in the case of iCASD, inability - to incorporate custom parts for specialised mission objectives. While these standardised satellites may be particularly suited to some missions, CubeSats in particular, being small and having a weight limit imposed by their launchers, are unable to incorporate payloads past a certain size. They also suffer from short mission lifetimes, particularly in VLEO where the atmospheric drag requires strong mitigation. Their limited battery and thruster capabilities, imposed by the same conditions as the payloads mentioned above, lead to very little drag mitigation possibility. Although some Cube-

Sats launched in 2003 were still operational in 2021 [35], this is not typical behaviour in VLEO. Without drag mitigation, some have been known to de-orbit in as little as nine months [40]. Thus, the current leading standardised satellite format is unsuitable for extended missions in VLEO altitudes.

Notably, the important VLEO missions summarised above all required larger, specially-designed spacecraft in order to house thrusters that could mitigate atmospheric drag and specialist equipment to collect a data series accurate enough for scientific research purposes. These spacecraft were far more expensive to manufacture and launch than any CubeSat, but unlike them, were able to operate in VLEO for an extended period of time. Ultimately, the trade-off in performance in some areas which accompanies standardisation and miniaturisation rendered existing standardised designs unsuitable for the purposes needed.

Thus, despite the industry-wide trend towards standardisation, it appears that the particular requirements of missions in VLEO render modern standard designs less suited for extended operation at these altitudes than for higher LEO. Currently, the approach is to individually engineer each satellite for the mission at hand, extending the opportunity to design these spacecraft more specifically for flight conditions in VLEO. With the increased interest particularly in Earth observation (EO) platforms [2], there is also an opportunity for a new standard EO platform designed to maximise lifetime and minimise costs of operation in VLEO. The current standards for LEO operation were simply not engineered with the specific requirements of VLEO altitudes in mind.

1.2 A Plan for Drag-oriented Engineering of VLEO Spacecraft

In order to mitigate atmospheric drag in VLEO through bespoke engineering, first the drag must be accurately characterised. This, in itself, is not a trivial task. Different models exist which attempt to characterise the atmospheric conditions, the most widely adopted among them being NRLMSISE-00 [41], Jacchia-Bowman [42], [43] and NASA's drag temperature model [44]–[46]. These models all involve some degree of interpolation and assumption, and aim to output the atmospheric composition for a certain set of conditions. At the beginning of this work, it was as yet unclear exactly which atmospheric conditions belong to the VLEO regime, and which model most accurately represents them, based on an extensive literature review. Considerable effort is put towards elucidating this in later chapters, as described by Research Question 1 below.

Assuming that likely atmospheric conditions in VLEO have been identified and better understood, their impact on a satellite must be determined. As with atmospheric models above, there are many ways of attempting this. Popular among them are:

1. Applying closed-form equations based on GSIMs which directly determine the

drag of a complete shape.

2. Using numerical simulations to directly simulate the atmosphere and the shape embedded therein.
3. Employing a panel method, which decomposes a shape into flat constituent panels and separately applies closed-form equations for drag to each panel.

There is a balance to be achieved between low time requirement and low accuracy, exhibited by closed-form equations which only apply to certain smooth 3-dimensional shapes, and the high time requirement and high accuracy of simulating the atmosphere directly. It is hoped that a novel panel method for drag analysis of satellites strikes a balance between the two, with accuracy close to that of numerical simulations but a time requirement close to that of closed-form equations. In order to be used for drag-oriented engineering, it was important to gain a much better understanding of its accuracy and limitations than previously existed, as well as try to mitigate the drawbacks. Only then could its results be considered reliable. This is extensively covered in multiple chapters of this work, under Research Question 2 below.

Finally, only once the atmospheric conditions and drag characteristics of satellite bodies are elucidated, can bespoke drag-oriented engineering be undertaken. Applying the aforementioned computational optimisation methods to such a problem involves integrating the inputs and outputs of the panel method with a suitable computational optimisation framework. The choice, requirements and implementation of an optimisation framework capable of tackling this problem form a large part of the latter half of this work. The considerations are concisely explained in Research Question 3.

1.2.1 Research Questions

This PhD thesis tackles the following research questions (RQs) :

RQ1: What are the current trends and industry standards for satellite flight at VLEO?

- How do the different atmospheric models compare when it comes to representing the VLEO atmosphere? Which is the most accurate?
- How is VLEO defined? Is there any way of consistently determining what range of altitudes is considered part of the VLEO regime?

RQ2: What methods can be used to accurately represent the physics involved in simulating the drag on a body in orbit in VLEO for the purposes of optimisation?

- What features of panel methods make them particularly suited to conceptual-level satellite design?

- What advantages and disadvantages does a novel panel method program for fast satellite drag analysis, have? How are these similar to, or different from, previous panel methods?
- How can we best mitigate the disadvantages so that it provides as good a representation of the real drag on the satellite as possible?

RQ3: Can geometric satellite analysis be used to design spacecraft better suited to operation in VLEO?

- How can computational shape optimisation methods be applied to facilitate drag reduction through geometry design?
- Can a panel method be integrated into the optimisation framework for the purposes of drag analysis?
- What other design parameters should be considered in the optimisation?
- Can optimisation methods be used to generate spacecraft geometries that show a measurable and significant improvement in spacecraft design for VLEO?
- How can optimisation of satellite geometries for VLEO be effectively integrated into the holistic mission design process? What benefits would this present?

1.3 Overview of Thesis and Novel Contributions to Science

This thesis is presented in journal format. Four chapters are reproductions of full scientific journal papers written during the course of this research. The papers are interspersed with linking chapters that are intended to bridge the gaps between them and facilitate the presentation of this PhD thesis as one coherent body of work. Although each paper deals with a different aspect of a research question, inevitably there will be some material which is repeated between them, particularly in the introductory sections. While an attempt has been made to repeat the same information as little as possible, at times this proved unavoidable. Where a chapter is comprised of a paper, the paper has been re-formatted to match the rest of the thesis. An introductory paragraph is also provided which states the content and contribution of the author to that paper.

Chapter 1 addresses the first part of RQ1 by providing an overview of the current and predicted market trends for satellite flight in VLEO. Data analysis regarding the number of operational satellites is presented, with previous years compared to the current situation to examine the trend in the number of operational satellites in VLEO. A number of recent important missions to VLEO are also summarised in the context of characterisation of aerodynamic drag. The current industry trend for standardisation and the popularity of small, off-the-shelf nanosatellites are discussed with a view to determining their suitability for extended future missions to VLEO.

Chapter 2 deals with the remainder of RQ1, presenting a scientific paper titled “Towards a clear definition of very low Earth orbits” which addresses the inconsistency in the definition of VLEO in the wider scientific community. A literature review is presented which highlights this inconsistency and the need for a better definition. A review of three atmospheric models is presented and the model most suited to VLEO analysis is identified as the US Naval Research Laboratory’s mass spectrometer and incoherent scatter radar model (NRLMSISE-00). This model is then employed to develop a consistent definition of VLEO by analysing the magnitude of the perturbing forces on a satellite. An explicit general definition for VLEO is suggested to resolve the conflicting nomenclature observed in literature. It is emphasised that this definition, while a useful label for the most likely range where the influence of the atmosphere will play a significant role on spacecraft aerodynamics, does not preclude the need for specific aerodynamics considerations where necessary.

Having provided an overview of the perturbing forces, chapter 3 moves on to the characterisation of the force in question, atmospheric drag. This chapter deals with RQ2. A paper is presented titled “ADBSat: Methodology of a novel panel method tool for aerodynamic analysis of satellites”, which describes modelling the physics behind atmospheric drag in VLEO. It details the workings of panel methods in general, and in particular of the novel panel method employed throughout this work, ADBSat. The differences and similarities between this program and other similar panel method software are highlighted. Finally, an example spacecraft analysis using ADBSat is presented.

Chapter 4 further addresses RQ2 by presenting a journal paper titled “ADBSat: Verification and validation of a novel panel method for quick aerodynamic analysis of satellites”. The validation of ADBSat against other methods of simulating atmospheric drag is presented therein. The details of the Direct Simulation Monte Carlo (DSMC) method are presented, and it is used as an industry standard to ascertain the accuracy of the outputs of ADBSat. Detailed cases are presented which thoroughly assess the performance of different aspects of ADBSat, thus leading to an in-depth knowledge of its advantages and disadvantages which was previously unknown. ADBSat is also compared to published literature sources of drag coefficients for various satellites, further showing its accuracy.

RQ3 is tackled in chapter 5, which outlines existing spacecraft optimisation literature targeting the VLEO orbital regime. General trends and commonly used methods are examined in the context of a realistic multi-objection problem. Metaheuristics are established as the most promising way of tackling the problem at hand due to their ability to handle a highly complex, non-smooth search space. The qualities of genetic algorithms (GAs) which make them particularly suited to the problem at hand are explained, as well as the particular features of the GA in MATLAB which makes it advantageous for use in this scenario.

Chapter 6 lays out how the optimisation problem was tackled. The integration of ADB-

Sat into the optimisation framework is also detailed, and justifications are presented for the choices made. Important features such as geometry parametrisation, design parameters, objectives and constraints are explained. Four different ways of approaching the optimisation of a satellite body are laid out: optimisation of the frontal face, tail face, front and tail simultaneously, and of the whole body. The procedure to generate the initial population of the GA is elucidated.

Chapter 7 concludes RQ3 with a detailed discussion of the quality of the algorithm results. A quantifiable improvement in design is clear across the multi-objective case when compared to both the initial population and single-objective results. A more holistic picture of the characteristics of the spacecraft body is obtained than through single-objective optimisation. Additionally, a wider range of satellite designs is found by the algorithm than those of the initial population. Specific aspects of the physical satellite shapes which resulted from the optimisation are discussed. The benefits of implementing a successful geometry optimisation framework such as this for the design of future missions to VLEO are outlined.

A summary of the thesis, its the novel contributions to science, and recommendations for future work are presented in chapter 8.

Chapter 2

Towards a clear definition of very low Earth orbits

This chapter includes the latest draft of the following paper:

Luciana Sinpetru, Nicholas H. Crisp, Peter Roberts, ”Towards a clear definition of very low Earth orbits”, TBD (2022).

Paper Content and Author Contribution

This paper presents a critical study of the literature regarding VLEO, with a focus on the inconsistency of the definition. Various sources are examined, from the advent of VLEO as a usable orbital regime to today, including alternate nomenclature such as super low Earth orbit (SLEO) and ultra low Earth orbit (ULEO). Many different conflicting definitions are found among the literature. The paper then endeavours to find a better definition for the range of VLEO, if indeed such a definition can be devised.

VLEO is often generally defined as the region of the atmosphere in which drag forces exhibit a considerable effect on spacecraft. At higher altitudes, this effect does not exist. This is an consistent distinguishing feature of VLEO when compared to other orbital regimes across literature. Atmospheric models attempt to measure the variation of the atmosphere with height, in terms of factors such as composition and density. They serve as a proxy for drag, and can be used to ascertain its influence on spacecraft. Thus, using the current existing atmospheric models, this effect is examined in more detail, and is used as the main criterion to attempt a more definite characterisation of VLEO.

To this end, a review of the most common atmospheric models that cover VLEO and their relative popularity is presented, taking in the US Naval Research Laboratory’s mass spectrometer and incoherent scatter radar model (NRLMSISE-00), the Jacchia-Bowman model (JB), NASA’s drag temperature model (DTM), and the International Reference Ionosphere (IRI). The IRI model is found to be lacking some necessary parameter outputs to fully model the VLEO atmosphere based on its data alone, and is therefore deemed to be complementary to the other three models rather than being an alternative. The use of the other three models in the scientific community is examined through the number of citations for each (including subsequent improvements to the model) according to Google Scholar. The performance of the models in reproducing the conditions of VLEO is also compared through a literature summary of the many comparison studies that exist between them. The NRLMSISE-00 is the most cited model, and also shows good performance at VLEO altitudes when compared to the other three models. Therefore, it is selected for the VLEO boundary definition.

The magnitude of atmospheric drag on various spacecraft is calculated and compared to the magnitudes of lunar perturbing force, solar perturbing force, and solar radiation pressure, at a range of solar activity levels spanning from low to high. Theoretical calculations are also performed for a number of test satellites to determine the point where the fuel necessary for a test mission lasting five years exceeds a reasonable mass budget, identified as 10% of the total satellite mass. For most test shapes, the drag force is found to exceed all three other forces in the vicinity of 450 km, a similar altitude to that where the fuel mass exceeds acceptable levels. Therefore, it is recommended that a reasonable upper boundary for VLEO is 450 km. This resolves the ambiguity seen in literature and clarifies the meaning of the nomenclature. As the amount of literature on VLEO missions has exploded in recent years, the adoption of this definition would ensure future consistency across different research groups. However, an emphasis is placed on the fact that this definition is based on some assumptions, and it does not preclude the need for the specific aerodynamic analysis of missions, where necessary.

The author's contributions are as follows:

1. Examined relevant literature concerning VLEO
2. Identified the problem of conflicting definitions and gathered data to support this
3. Conceived the solution of using atmospheric models to examine the influence of drag in VLEO
4. Gathered data on, and performed analysis of, the different atmospheric models
5. Executed the calculations
6. Wrote the paper

The second and third author contributed to the paper through ideas which strengthened the analysis of and solution to the problem, as well as substantial editorial comments which significantly improved the manuscript.

Abstract

This work proposes a general definition of 100 km to 450 km for the recently popular very low Earth orbit (VLEO) regime. A literature review shows conflicting definitions and nomenclature of the orbital range commonly defined as VLEO, in both recent and historical sources. Defining a clearer demarcation is beneficial in the face of inconsistency and promotes clarity across further research. VLEO is broadly defined as the region of the atmosphere where atmospheric drag has a significant effect on spacecraft orbit and design. This imprecise definition is clarified by analysing the effects of drag in comparison to other perturbation forces in the low atmosphere. A review of currently available atmospheric models is provided, based on which NRLMSISE-00 is used to analyse the conditions therein. The orbit-keeping requirements of a number of test spacecraft, both real and hypothetical, are also considered in arriving at the above definition. While this work does not preclude the need for precise aerodynamic analysis as part of a mission design framework, it serves as a useful benchmark for future researchers when discussing VLEO.

Keywords: very low Earth orbits, rarefied flow, satellite flight, review

2.1 Introduction

Very low Earth orbits (VLEO), those at the bottom end of the low Earth orbit range, can have extensive benefits for satellite flight. These include better data resolution, potentially more accurate positioning, and a potential decrease in payload characteristics such as mass, size and cost. Because of this, recent years have seen an increase in the number of satellites orbiting closer to the Earth. Correspondingly, the number of publications describing the flight of satellites in such atmospheric conditions has also increased. These publications show some disagreement in the definition of VLEO.

Traditionally, the generally accepted orbital regimes have been:

- Low Earth Orbit (LEO): 100 km to 2000 km
- Medium Earth Orbit (MEO): 2000 km to 35 786 km
- Geostationary Orbit (GEO): 35 786 km
- High Earth Orbit (HEO): Altitudes higher than 35 786 km

VLEO has not been explicitly included in these traditional definitions of orbital regimes, but instead considered a subset of LEO. However, the phrase “very low Earth orbit” has been in use since at least the 1980s [47]–[50]. It is likely that these low orbits, and the associated nomenclature, were used even before this - some of the earliest orbital spacecraft of the 1960s, the Russian Vostok design, were intended to operate at an altitude of 250 km, well within most modern definitions of the VLEO regime [51]. However, these sources fail to identify exactly what classifies as a VLEO orbit, instead using it as an umbrella term for an unspecified lower part of the LEO regime.

McMillen [48] is concerned with satellite designs engineered for different altitude flights, and uses the term VLEO to refer to the altitudes available to their lowest-altitude designs: 148 km to 334 km. Their use of the term VLEO has the purpose of distinguishing their lowest-altitude designs from those that fly higher in LEO, as opposed to describing a particular range of orbital altitudes. Similarly, Fogel, Fogel, and Atmar [49] also uses the term VLEO to describe their lowest-altitude satellites without defining it outright - however, their satellites are at unspecified altitudes up to 1000 km. Inconsistency is also apparent in other publications, among which Lichten, Thornton, Young, *et al.* [50] defines a VLEO orbiter as being at 500 km. Following the observed trend, it can be assumed that it is to set it apart from their other orbiter, higher in LEO at an altitude of 1300 km.

Of these early papers, Bainum [47] comes closest to defining the term, explaining VLEO as the region within which aerodynamic drag forces dominate over the effects of solar radiation pressure. They examine the effects of solar radiation pressure on LEO orbits, but provide no definition of the relative magnitude of this force when compared to atmospheric drag. The limited information available at the time on the magnitude of atmospheric drag forces is the most likely cause [52]. At the time, while analysis of the uncertainties pertaining to drag force accuracy was well understood, a numerical value for drag force on an orbiting spacecraft proved elusive [53].

More recent papers are oftentimes also in direct conflict, making it difficult to define VLEO more concretely from literature sources. Additionally, alternative nomenclature is also encountered, such as super low Earth orbit (SLEO) and ultra low Earth orbit (ULEO). An industry-wide unification of these synonymous terms would undoubtedly lead to more clarity overall in the reporting of scientific advancements. The following is a non-exhaustive list of literature sources and the definitions employed therein.

- Park, Myong, Kim, *et al.* [54]: 200 km to 300 km
- Hsieh, Pan, and Lo [55], Romano, Espinosa-Orozco, Pfeiffer, *et al.* [56], Virgili-Llop, Roberts, Hao, *et al.* [57] and Virgili-Llop, Polat, and Romano [58], [59] and the DISCOVERER project [1], [2], [60] : less than 450 km
- Walsh and Berthoud [61], Walsh, Berthoud, and Allen [62], Goto, Umeda, Yukumatsu, *et al.* [63]: less than 300 km to 450 km
- Leomanni, Garulli, Giannitrapani, *et al.* [64]: 250 km to 500 km

- Schönherr, Komurasaki, Romano, *et al.* [65] and Romano, Massuti-Ballester, Binder, *et al.* [66]: 100 km to 160 km
- Minton, Schwartzentruber, and Xu [67]: 180 km to 350 km
- Fujita and Noda [27], [68] (SLEO): 160 km to 300 km
- Doroshkin, Zadorozhny, Kus, *et al.* [69] (ULEO): 200 km
- Grasso [70] and Golikov and Filatyev [71] (ULEO): less than 250 km
- Yu and Fan [72] (ULEO): 120 km to 300 km

Of these more recent papers, Virgili-Llop, Roberts, Hao, *et al.* [57] make the most thorough attempt to define this orbital regime, by reproducing a figure from Fortescue and Stark [73]. They assert that:

> “...the effect of atmospheric drag starts to be significant at an altitude around 500 km...”

The figure supports this statement, with the lack of detail available captured in the wording. However, a more thorough analysis of the data behind the graph is necessary for a better quantification of the strength at which drag changes from non-significant to significant.

In order to reduce confusion and remove doubt about VLEO, it is necessary to examine some pertinent aspects of the atmosphere. Based on the references examined thus far, it is unclear what the scientific distinction is between LEO and VLEO - or indeed whether there is one at all. It is also possible that the phrase has no clear scientific definition, but is instead used more informally to cover low-flying missions. While in this context a hard boundary of VLEO is not explicitly required, it is still useful in a mission design context to examine the influence that drag would have on satellites across a range of altitudes.

2.2 Atmospheric Models

To examine the atmospheric composition in the lower end of LEO and determine a boundary for VLEO, it is necessary to choose one of the many available atmospheric models. These models differ in the equations through which they express the atmosphere and its different aspects. Each one claims to most accurately represent the environment in which a satellite travels. The choice of model provides estimates of key atmospheric parameters such as particle number densities, temperatures, and speeds in different conditions. These are necessary in order to examine the perturbation forces on objects in the atmosphere.

Model name	Source	Citations
NRLMSISE-00	Picone, Hedin, Drob, *et al.* [41]	3359
JB2008	Bowman, Tobiska, Marcos, *et al.* [42] Bowman, Tobiska, Marcos, *et al.* [43]	489
DTM	Bruinsma, Thuillier, and Barlier [44] Bruinsma, Sánchez-Ortiz, Olmedo, *et al.* [45] Bruinsma [46]	324

Table 2.1. The most popular atmospheric models, based on number of citations.

There is no one-size-fits-all solution for the most appropriate atmospheric model. The best option for a particular altitude, time, year or solar activity level may not be the most accurate for other combinations [74]. Assessments of the different factors involved in developing and using an atmospheric model have previously been conducted [74], [75]. An important conclusion of these works is that despite their firm theoretical basis, all models involve some degree of assumption and interpolation, leading to a difference in results. A list of these assumptions and interpolations is provided in Vallado and Finkleman [75] with a thorough explanation of each.

The most common models are detailed in table 2.1: the US Naval Research Laboratory's mass spectrometer and incoherent scatter radar model (NRLMSISE-00), the Jacchia-Bowman model (JB) and NASA's drag temperature model (DTM). NRLMSISE-00 is an MSIS model following the methodology of its predecessors, MSIS-86 and MSISE-90, basing its predictions on neutral mass spectrometer measurements and incoherent scatter radar data. It relies on the $F_{10.7}$ proxy for solar activity as well as modelling atmospheric variation with respect to geography and time [41], [76], [77]. The JB model uses its own new exospheric temperature and density equations as well as an 81-day average solar index to calculate atmospheric variation [42]. DTM uses various proxies throughout its many incarnations, including $F_{10.7}$ for DTM2009 and F_{30} for DTM2013, and uses similar equations to NRLMSISE-00, based on the Bates exponential profile [77]. The number of citing sources indicates the relative popularity of each model[1].

Complementary to these atmospheric models is the International Reference Ionosphere (IRI). However, the IRI model does not provide values for some necessary parameters, such as atmospheric density and number density. Instead, ionospheric quantities such as electron and ion temperature, electron density, ion composition, and atmospheric indices are among its output values [78]. This makes it unsuitable for use in the analysis of VLEO or in drag modelling, which specifically requires a value of density. Because of this, it has been excluded from this analysis.

A significant difference in results between the three models under scrutiny could lead to clashing definitions of the VLEO regime. In particular, large reported differences in density and solar flux could translate to large differences in atmospheric drag values be-

[1] Number of citations obtained from Google Scholar, accessed 14/02/2022. For any model with multiple sources, the citations are the total number for the original source and any additions to the model since then.

tween models. Therefore, it is important to understand the differences between the models before a thorough analysis can be undertaken.

Some analysis of these models has already been performed. NRLMSISE-00, DTM-2012 and JB2008 are directly compared by Zhaborovskyi [79], where a linear least-squares method is used to solve for the relationship between point coordinates in two different models. Thus, they remove mutual systemic errors, and can then calculate the dispersion of the remaining random errors to provide a measure of accuracy for each model. The final conclusion of this work is that the models are similar enough as to be arguably identical.

Bruinsma, Sánchez-Ortiz, Olmedo, *et al.* [45] and Bruinsma [46] focus on the inclusion of new experimental data sets from satellite flights into the DTM model, and the subsequent comparison of this model to JB2008 and NRLMSISE-00. Bruinsma, Sánchez-Ortiz, Olmedo, *et al.* [45] performs a statistical analysis to examine the models under conditions of varying altitude and solar activity level. These measures rely on a comparison between observed and calculated data. They conclude that while DTM2009 is most accurate for the newly incorporated data sets, the JB2008 model performs better at altitudes under 400km especially for high solar activity, and the NRLMSISE-00 model performs better above 500km particularly for low and medium levels of solar activity.

Although the DTM model has since been updated, both the JB2008 and NRLMSISE-00 models are still in use, and therefore the comparison above is still partly valid. Bruinsma [46] extends some of this comparison to the most recent version of the DTM model, DTM2013, but, interestingly, does not include NRLMSISE-00 in the comparison. A similar comparison criteria to that described for Bruinsma, Sánchez-Ortiz, Olmedo, *et al.* [45] was employed. They conclude that "DTM2013 is the least biased model", and assert, with supporting evidence, that it is better than DTM2009. They also conclude that DTM2013 is more accurate than JB2008 for all altitudes across the variables they examined. However, the missing comparison to NRLMSISE-00 renders the analysis somewhat incomplete.

Finally, Doornbos [80] presents an extremely detailed analysis of different atmospheric models when compared to real accelerometer data from the Challenging Minisatellite Payload (CHAMP) and the Gravity Recovery and Climate Experiment (GRACE). Particularly useful is the comparison between satellite data and JB2008 and NRLMSISE-00 models, in terms of density across the satellite tracks. A pictorial representation is provided in Fig. 5.5 and Fig 5.6 of this work [80]. While both models are lacking in detail compared to the observational measurements, NRLMSISE-00 performs better at showing the midnight density maximum and at accounting for the influence of the geomagnetic field on the thermosphere. However, the drawback of using either of these models is their tendency to overestimate density at very low solar activity, when $F_{10.7} < 75$. Here, $F_{10.7}$ is the solar radio flux at 10.7 cm, in units of $1 \times 10^{-22}\,\mathrm{W\,m^{-2}\,Hz^{-1}}$, commonly considered a reliable indicator of solar activity.

In light of this analysis, the NRLMSISE-00 model was chosen as the most reliable in capturing density fluctuations overall. Taking as an input a date and time in the past, geographic co-ordinates, as well as solar indices if desired, this model can output atmospheric conditions across a range of heights. Throughout its lifetime, it has been constantly improved by incorporating real recorded data from flights through the thermosphere, such as that from the CHAMP satellite [22], [81]. Its data[2] and source code[3] are freely available online.

2.3 Methodology

Having arrived at a choice of suitable atmospheric model, it is necessary to employ it to examine atmospheric density at different altitudes. Variation in atmospheric density is particularly strong with solar activity levels, but also apparent due to other factors such as time of day and position over the Earth. Translating this variation to atmospheric drag will lead in a fluctuation of the real altitude of the VLEO - LEO boundary. Thus, such a boundary is not an absolute measure in all conditions. It is more beneficial to determine a methodology by which the upper boundary of the significant influence of drag can be determined for any mission which may be affected. As ignoring atmospheric effects is currently the default treatment for the upper part of LEO, a means by which to calculate a boundary - albeit an imprecise one - at which to begin to consider the effects of drag on a mission would provide useful insight for the mission design process.

The four main perturbing forces not due to Earth's gravity are the lunar gravitational perturbing force, solar gravitational perturbing force, solar radiation, and atmospheric drag. These forces can be calculated from well-established formulae [73], [82]–[85]. The boundary of VLEO will be examined by considering the relative magnitudes of these forces, and the point at which atmospheric drag becomes the dominating force. Furthermore, the atmospheric drag will also be examined in the context of the orbit-keeping capabilities of satellites, in order to determine when its influence becomes significant.

2.3.1 Lunar and solar perturbing forces

The acceleration caused by the lunar and solar perturbing forces is a function of the dot product between the vector positions of the influencing body and the satellite in a geo-

[2] Available at: `https://ccmc.gsfc.nasa.gov/modelweb/models/nrlmsise00.php`, accessed 23/09/2021

[3] Available at: `https://www.brodo.de/space/nrlmsise/`, accessed 23/09/2021

centric reference frame [82], as shown in eq. (2.1).

$$\ddot{\vec{r}} \approx \frac{GM_P r}{s_P^3}(-\vec{e}_r + 3\vec{e}_s(\vec{e}_s \cdot \vec{e}_r)) \tag{2.1}$$

Here, $vec\ddot{r}$ is the resultant acceleration vector caused by the perturbation, G is the gravitational constant, M_P is the mass of perturbing body P, r is the magnitude of the satellite position vector in geocentric coordinates, s_P is the magnitude of the perturbing body P position vector in geocentric coordinates, $\vec{e}_r$ is a unit vector pointing to the satellite from the Earth, and $\vec{e}_s$ is a unit vector pointing to the perturbing body from the Earth. Using the well-known relation $\vec{F} = m_s\ddot{\vec{r}}$, where m_s is the mass of the satellite, it is possible to relate this acceleration to the resulting force. The magnitude of the force vector can then be obtained by using eq. (2.2). The dot product has been simplified by taking advantage of the fact that the vectors involved are unit vectors.

$$F = \frac{GM_P m_s r}{s_P^3}|3\cos\theta\vec{e}_s - \vec{e}_r| \tag{2.2}$$

The angle between the two unit vectors is given by θ. Furthermore, one can substitute the equation for the magnitude of the subtraction of two vectors into eq. (2.2). Simplifying further and changing from $\cos^2\theta$ to $\sin^2\theta$ results in eq. (2.3), which gives the instantaneous perturbing force.

$$F = \frac{2GM_P m_s r}{s_P^3}\sqrt{1 - \frac{3}{4}\sin^2\theta} \tag{2.3}$$

A simplifying assumption can be made that the perturbing body is approximately stationary for one satellite orbit. Given the fast orbital speed in VLEO orbits, this assumption is reasonable. Due to the satellite and perturbing body vectors being in geocentric co-ordinates for the purposes of this calculation, one can then average over all possible relative positions in one orbit and find the mean perturbing force. This is shown in in eq. (2.4).

$$F = \frac{GM_P m_s r}{s_P^3}\frac{1}{\pi}\int_0^{2\pi}\sqrt{1 - \frac{3}{4}\sin^2\theta d\theta} \tag{2.4}$$

The integral is an incomplete elliptic integral of the second kind, $E(\phi|m)$, where $\phi = 2\pi$ and $m = 0.75$. It is not reliant on any problem parameters and can be integrated numerically. Thus, the average lunar and solar perturbing forces can be calculated from eq. (2.5) and eq. (2.6), respectively.

$$F_{\rightmoon} = \frac{GM_{\rightmoon} m_s r}{\pi s_{\rightmoon}^3} E(\phi|m) \tag{2.5}$$

$$F_{\odot} = \frac{GM_{\odot} m_s r}{\pi s_{\odot}^3} E(\phi|m) \tag{2.6}$$

Here, $M_{\rightmoon}$ is the mass of the moon, $s_{\rightmoon}$ is the distance from the Earth to the moon, $M_{\odot}$ is the mass of the sun, and $s_{\odot}$ is the distance from the Earth to the sun.

2.3.2 Solar radiation pressure force

The force due to solar radiation pressure relies on knowing the radiation coefficient and reference surface area of the satellite. The radiation coefficient for many common satellite materials can be found in literature [82]. For simplification, it is normally assumed that the surface normal of the satellite points in the direction of the sun. The reference area is normally taken to be the cross-sectional surface area, however if solar panels are present then care must be taken to use an appropriate value. In this case, a worst-case scenario is assumed for solar radiation pressure, where the satellite is closest to the sun. Thus, it is not necessary to account for eclipse conditions. The equation for the magnitude of the solar radiation force is given in eq. (2.7).

$$F_{SRP} = \frac{\sigma_{SF} T^4 R_p^2 C_R A_{ref}}{c s_{min}^2} \tag{2.7}$$

Here, σ_{SF} is the Stefan-Boltzmann constant, T is the temperature of the sun, R_p is the radius of the photosphere, C_R is the radiation coefficient, A_{ref} is the reference cross-sectional area, c is the speed of light, s_{min} is the minimum distance from the satellite to the Sun. C_R is assumed to be 1.5 here, as is common in literature.

2.3.3 Atmospheric drag force

The force due to the atmospheric drag has been well-documented, and can be calculated from eq. (2.8).

$$F_d = \frac{1}{2} \rho v^2 C_d A_{ref} \tag{2.8}$$

Here, ρ is the density of the atmosphere, v is the speed of the satellite relative to the atmosphere, C_d is its drag coefficient, and A_{ref} its reference area. A common simplifying assumption is that the atmosphere co-rotates with the Earth, which has been assumed here. Given this assumption, the orbital altitude of the satellite can be used to determine the values of ρ and v. Pre-existing knowledge of the spacecraft aerodynamics is desirable, in the form of C_d. If the precise drag coefficient is unknown, 2.2 is often used as a standard value. The reference area is the same as that used in eq. (2.7) for solar radiation pressure.

2.3.4 Effect of atmospheric drag

Furthermore, it is not just the magnitude of the drag force that is important, but the effect that this induces on the satellite. The decrease in velocity that this causes can be quantified by approximating the derivative in Newton's second law as a change in velocity Δv over a certain time period Δt. Thus, the change in velocity due to atmospheric drag, which must be compensated for, can be calculated by using eq. (2.9).

$$\Delta v = \frac{F_d}{m}\Delta t \tag{2.9}$$

Using the Tsiolkovsky rocket equation, the initial propellant mass fraction, M_f, can then be calculated using eq. (2.10)

$$M_f = 1 - e^{\frac{-\Delta v}{g I_{sp}}} \tag{2.10}$$

Where g is the acceleration due to gravity and I_{sp} is the specific impulse of the thruster, in units of seconds. M_f does not include the mass of the propulsion system itself, but purely the mass of the working substance. The acceptable mass fraction will vary depending on the satellite and mission requirements. I_{sp} will depend on the choice of thrusters available for the mission. As a generalization, 10% is taken as a limiting mass fraction available for a satellite mission. By examining the value of M_f at a range of altitudes, it is possible to determine where it exceeds 10%, and thus to determine the altitude below which the atmospheric influence becomes significant. While it would be beneficial to perform this calculation for each mission individually depending on mission parameters, it is also useful to provide a general guideline where the boundary of VLEO is by performing this calculation for a range of test spacecraft.

2.3.5 Solar activity conditions

In light of the NRLMSISE-00 model's known inaccuracy at periods of $F_{10.7} < 75$, and its particularly good performance at low and medium solar activity, the primary condition to examine atmospheric factors was chosen as medium solar activity. Criteria for low and high solar activity are $F_{10.7} < 80$, and $F_{10.7} > 180$, respectively [86]. Intermediary values of $100 < F_{10.7} < 170$ were chosen to be representative of medium solar activity levels. To avoid as much as possible the influence of seasonal variations in atmospheric properties, a longitude and latitude of (0,0) were chosen for this analysis. As an exploratory analysis in order gain an understanding of the possible fluctuation of atmospheric drag, high and low solar activity conditions as defined above is also be briefly considered.

Activity level	Start date	End date	Min $F_{10.7}$	Max $F_{10.7}$
Low	01/12/2007	31/01/2008	68.0	79.8
Medium	01/01/2015	30/04/2015	105.5	166.7
High	1/12/2001	31/01/2002	182.4	265.6

Table 2.2. A summary of the dates and $F_{10.7}$ solar index ranges for drag force analysis.

Graphs of $F_{10.7}$ solar activity were examined to determine recent periods of suitable solar activity levels [87]. Figure 2.1 shows the overall trend in $F_{10.7}$ solar flux between 1996 and 2019. Suitable periods of low, medium, and high solar activity were identified as specified in table 2.2. The black line on the graph, representing the real, un-smoothed data, was used to determine the examined time period.

One data set per day was downloaded from NRLMSISE-00, at the same time each day. The data set contained values of total mass density and neutral temperature at values of altitude of 100 km to 1000 km, in steps of 25 km. The data set for 14/01/2015 was found to be corrupted, with undefined values for some of the points, and was discarded. Data sets for the period 09/12/2007 to 16/12/2007 were also discarded from analysis of the low solar activity conditions, as they had $F_{10.7} > 80$ and thus did not qualify as low

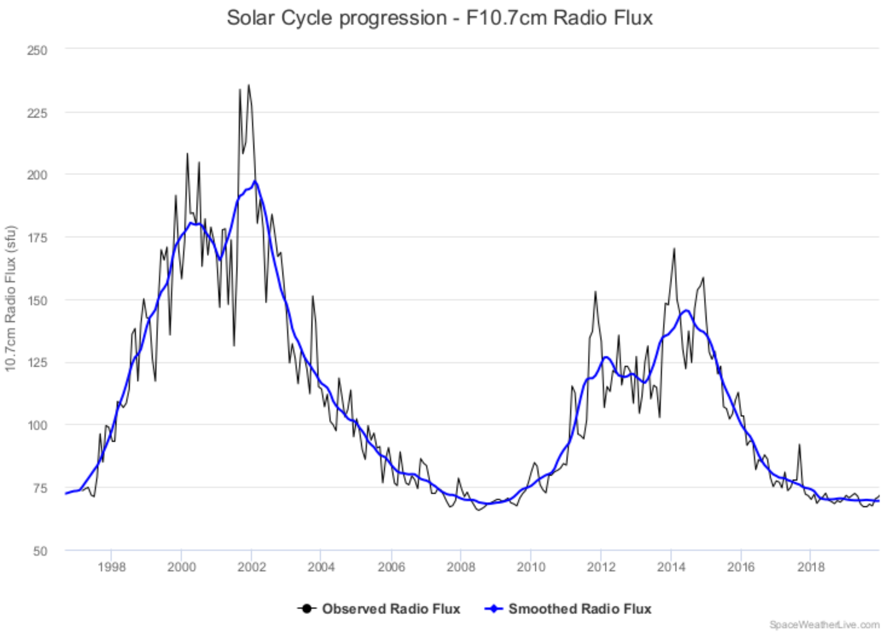


Figure 2.1. Solar activity levels as measured by the $F_{10.7}$ index between September 1996 and December 2019. Reproduced from Parsec vzw [87].

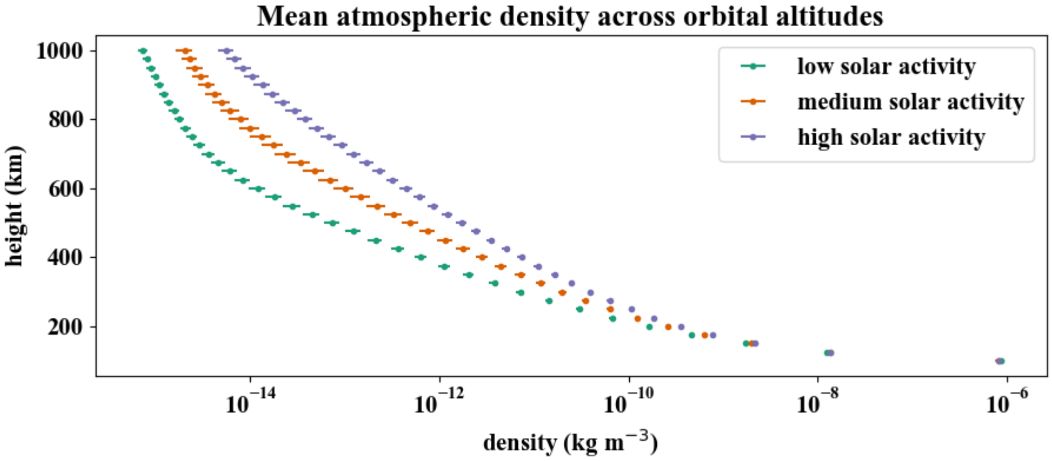


Figure 2.2. A graph of mean atmospheric density at a certain height for medium solar activity, using the NRLMSISE-00 model. The error bars represent 1σ, one standard deviation. Note the logarithmic scale of the x axis.

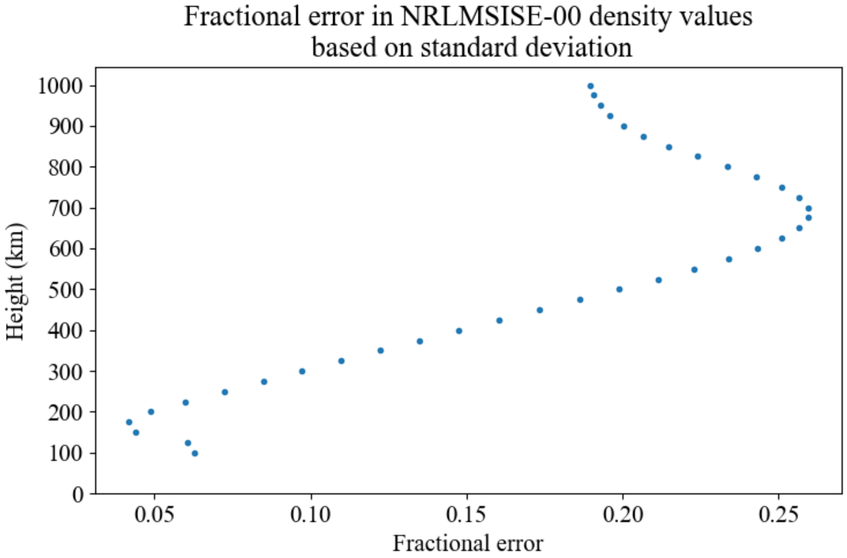


Figure 2.3. The scatter at each height as a fraction of mean density. Scatter σ can also be interpreted as the error on the mean.

solar activity.

Figure 2.2 shows a graph of the mean density of the atmosphere across the whole period of medium solar activity, at each height. A sharp decrease of the mean density with increasing altitude can be observed, as expected. The error bars represent the standard deviation σ, which is the scatter of the density values at that height. This scatter is independently analysed in fig. 2.3, and shows an increase with height up to around 700 km, and a slight decrease after. This may be a real atmospheric effect in that the density may fluctuate more at higher altitudes, or it may be an artefact of the model as less data is available at some altitudes, leading to higher uncertainty. However if this was the case, it would be expected for the trend to show an overall increase. The existence of the turning point at which the scatter begins to get smaller implies that this is a real effect.

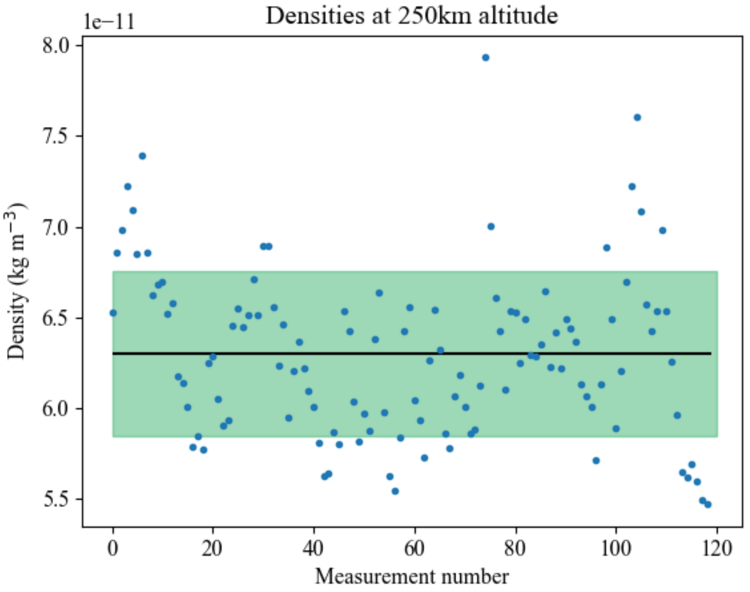


Figure 2.4. A plot of the density at 250km across the data range. The black line represents the mean value of density, and the green box represents the standard deviation.

Graphs of the values of density at a constant altitude were also produced separately for each altitude in order to verify the data, with an example of such a graph shown in figure fig. 2.4. In this graph, the black line is at the mean value, with the standard deviation range represented by the green box. The measurements are numbered by date, from earliest to latest. This graph is representative of the graphs produced for every altitude, except for 125 km; a clear sinusoidal variation in the density values was seen at this altitude, differing from the seemingly random scatter seen across the other altitudes. A reason for this could not be determined.

2.4 Example calculations

Six test satellites were chosen to perform the above calculations in an effort to discern the upper limit of VLEO. The choice was intended to capture a broad picture of the characteristics of satellites that could fly in VLEO in terms of the relevant parameters. In other words, we attempted to choose both large and small, light and heavy satellites of varying body aspect ratios. The choice of propulsion system and associated I_{sp} was restricted to realistic values. An in-depth summary of each satellite is provided below, with a summary of their relevant characteristics in table 2.3. CAD models of the satellites can be seen in fig. 2.5

- **1U CubeSat:** Due to the relative popularity of the CubeSat design when compared to any other standard satellite design, a 1U CubeSat was chosen as one of the test spacecraft. Its side length is 0.1 m. The other parameters are chosen to be representative of the typical configuration of such a satellite. Its drag coefficient C_d and solar reflectivity C_r were taken as the standard commonly-used values for spacecraft in orbit, 2.2 and 1.5 respectively. There are a range of propulsion options for 1U cubeSats. Assuming a cold gas propellant system, the specific impulse is assumed to be in the region of 100 s [88]–[90]. Although thrusters with higher specific impulse are available on the market [89], cold gas thrusters are common, inexpensive compared to other propulsion methods, and have a notably positive track record on 1U cubeSats [88]. Thus, they are considered representative of a typical mission.

- **Test Sphere:** A plain sphere of 0.5 m radius was chosen as the most universally scalable shape. C_d, m, and I_{sp} were chosen to be realistic for conditions in VLEO. This shape is not based on any currently or previously orbiting satellite, but provides valuable insight into a limit of VLEO for the average small satellite.
- **The Gravity Recovery and Climate Experiment (GRACE):** This mission involved two identical satellites flown at 483 km to 508 km altitude, decaying to around 300 km by end of life. The mass and dimensions are reported by Agency [91]. The drag coefficient is taken to be the worst case scenario of the data collated by Sinpetru, Crisp, Roberts, *et al.* [92], as this captures the most extreme case as well as all others in determining the limit of VLEO. The thruster was assumed to be the same as that of its follow-on mission [29], as the missions are extremely similar.
- **The Gravity Recovery and Climate Experiment Follow-On (GRACE-FO):** This direct follow-on from GRACE maintains most of the original mission's parameters, such as altitude, dimensions and thruster [29]. Its science payload increases in mass compared to the original. As it was designed to maintain the same ballistic coefficient C_B as GRACE [29], C_d could be calculated using eq. (2.11).

$$C_B = \frac{m}{C_d A_{ref}} \tag{2.11}$$

- **Challenging Minisatellite Payload (CHAMP):** Orbiting at 418 km to 474 km and decaying to below 300 km by end of life, CHAMP was one of the first modern satellites to fly in VLEO. Its drag coefficient is again taken to be the worst case scenario reported by Sinpetru, Crisp, Roberts, *et al.* [92]. Its thruster is assumed to be the same as that flown on the GRACE missions [93]. Its mass and dimensions are reported by Reigber, Lühr, and Schwintzer [20].
- **Gravity Field and Steady-State Ocean Circulation Explorer (GOCE):** Drag-oriented design can be observed in the geometry of GOCE, with its elongated shape and small frontal surface area. It orbited much lower than the other real satellites in this analysis, at around 260 km. Its mass and thruster capabilities are reported by Wallace, Jameson, Saunders, *et al.* [94], with a worst case scenario for the thruster assumed. The frontal area and drag coefficients are reported by Gini [95], with the highest value of drag coefficient used.

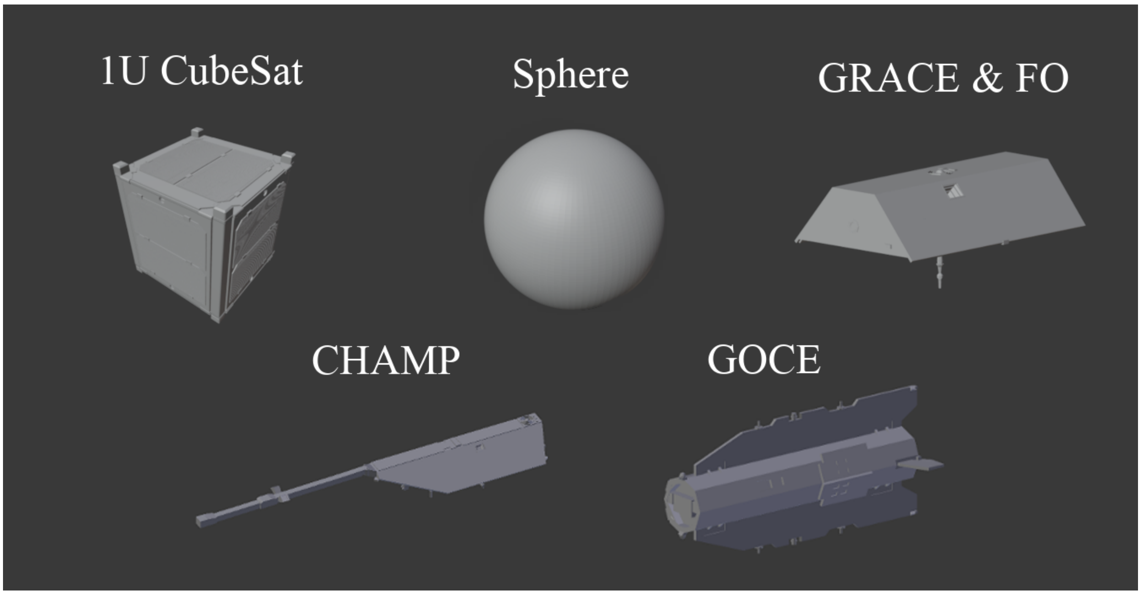
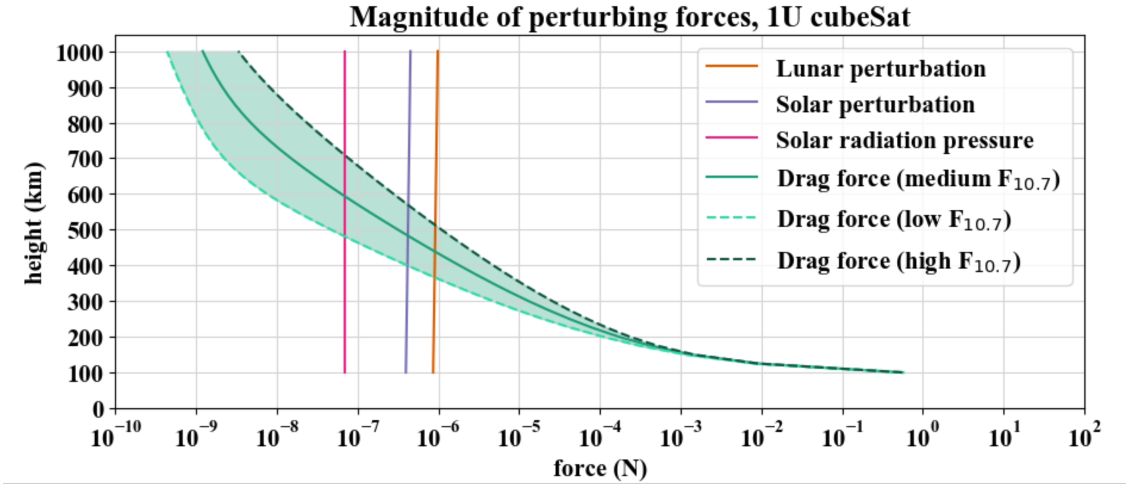
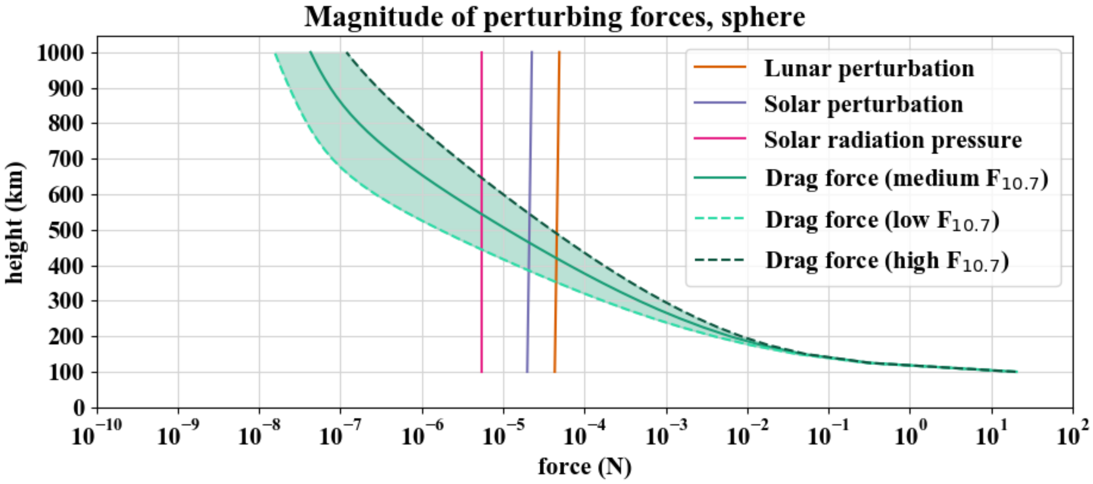


Figure 2.5. CAD models of the test satellites used for the calculations.



(a) Perturbing forces on a 1U CubeSat.



(b) Perturbing forces on a test sphere.

Figure 2.6. The magnitude of perturbing forces not caused by the Earth's gravitational pull on two test objects. The highlighted region around the drag force indicates the drag force at low and high solar activity levels.

Satellite	C_d	$A_{ref}\,(m^2)$	$m\,(kg)$	M_f	$I_{sp}\,(s)$	VLEO height (km)
1U CubeSat	2.2	0.01	1	-	100	458
Test Sphere	1.0	0.785	50	-	100	439
GRACE	3.9	1.0269	487	0.07	70	429
GRACE-FO	4.8	1.0269	600	0.052	70	422
CHAMP	3.6	0.8	522	0.057	70	400
GOCE	3.853	1.1	1050	0.095	500	288

Table 2.3. Relevant parameters of the different satellites used to perform the calculations in section 2.3. The upper limit of VLEO calculated for each is also listed.

The four perturbing forces on the 1U CubeSat and the test sphere which are not due to the Earth's gravity are shown in fig. 2.6. This is similar to the graph presented in Fortescue and Stark [73]. For a 1U CubeSat, the drag force at medium solar activity equals the solar radiation pressure by 600 km altitude, and exceeds all the other perturbing forces at 450 km. The range at which drag force dominates between low and high solar activity is 367 km to 513 km. For a sphere, the drag force exceeds all others at 450 km, in the range 352 km to 489 km. Thus, it would be reasonable for the most general boundary of VLEO to be defined somewhere in the region of 420 km to 450 km, where drag forces begin to dominate for both generic test satellites. It is assumed that if the drag force dominates, it will have a significant effect on spacecraft design, requiring drag minimisation or other orbit-keeping solutions for extended mission lifetimes.

The mass of the thruster unit itself must be taken into account. For CubeSats, this can be up to 50% of the total allowed mass. Therefore it is realistic to assume that a generous estimate of the mass of fuel required for a typical 5-year mission should be no more than 10% of the initial mass of the satellite. This is validated by the real missions, with reported fuel mass fractions ranging from 5.2% to 9.5%, as seen in table 2.3.

Figure 2.7 shows the calculated propellant mass fraction necessary in medium solar conditions for each satellite to maintain a 5-year mission. The highlighted regions indicate high and low solar activity levels. $M_f = 0.1$ is shown on the graph as a vertical line, and the height at this value for each satellite is explicitly shown in table 2.3. A value of 1.0 indicates that the propellant needed to maintain the orbit over the 5-year lifetime would equal or exceed the mass of the satellite itself.

With the exception of GOCE, all shapes show VLEO upper altitudes in the range 400 km to 460 km at medium solar activity. This is similar to where the drag force was found to exceed all three examined perturbing forces in fig. 2.6. Of the literature definitions mentioned above, both analyses support the adoption of 450 km as a representative border on VLEO, which would encompass most shapes shown without excessively overshooting them. The drag-oriented design of GOCE, being long and slender, is the driving factor in lowering its orbital height for $M_f = 0.1$. It is better suited to operation in dense atmosphere than the other satellites, which mostly operated at the top of VLEO. This highlights the importance of drag-oriented design for lower orbital altitudes where the drag

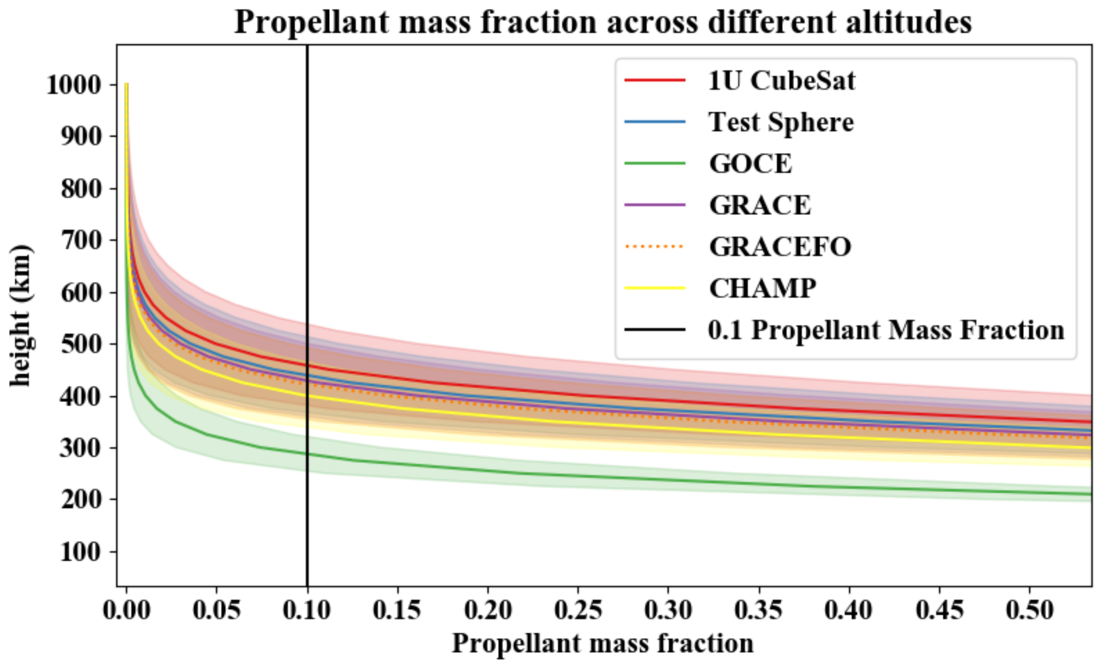


Figure 2.7. Mass fractions required for orbit-keeping manoeuvres over a 5-year mission for different test satellites in intermediate solar conditions. The highlighted regions capture the variance in mass fraction for high and low solar activity conditions.

force dominates, through significant changes in satellite geometry. However, because the upper limit of VLEO is intended to be general and encompass most satellites while reconciling the conflicting literature, this is still in support of 450 km as an upper limit. This is a generous upper limit for satellites exhibiting drag-oriented design, while also encompassing those that do not. Therefore, it is the most general definition.

It is important to note that this is not an absolute value. It is a general guide for the orbital altitude at which drag forces begin to dominate, and where spacecraft geometry design will need to take this into account. The definition is intended to reconcile differences in nomenclature across past literature. It serves as a benchmark to future researchers and promotes consistency when using the term “VLEO”. As with most benchmarks, it is based on a set of assumptions and conditions that are generally, but not universally, appropriate. It provides a useful point of reference when discussing VLEO. It certainly does not preclude the need to include thorough aerodynamic analysis as part of the mission design process.

2.5 Conclusions

Analysis of the existing literature pertaining to VLEO shows that there is some disagreement between authors as to what range of altitudes VLEO refers to. In light of the increasing interest in missions to VLEO in recent years, it is beneficial to define such a boundary as a guideline of where to start considering the influence of atmospheric drag on satellite missions. Reconciling the nomenclature will also serve to provide a reference point for future research into VLEO. A methodology is presented which covers analysis of the four most important perturbing forces outside of the Earth's gravity in VLEO: the lunar gravitational force, solar gravitational force, solar radiation pressure, and atmospheric drag.

The upper limit of VLEO depends on satellite parameters and atmospheric conditions, and thus its absolute value be different for each mission. However, general insight into the influence of atmospheric drag can be gained by examining test satellites, both theoretical and based on real missions. Rather than choosing any one mission with a particular satellite for which to examine the perturbing forces, the geometry of which may never be reproduced, examining a range of shapes means that there is a high likelihood of this analysis applying to future missions. Intermediate solar activity conditions were chosen since they are more prevalent than the extremes, with low and high solar activity being considered for general interest. The most suitable upper limit for VLEO from existing literature was determined to be 450 km. This limit encompasses most test satellites without over-shooting, and is a generous upper limit on satellites with drag-oriented design such as a slender, elongated shape.

In terms of solar activity, when planning a mission which may run two or three years in the future, the actual daily levels of solar activity cannot be predicted. While the 11-year solar cycle is useful in predicting a general expected level, this analysis can serve as a general guide for what to be expected at the different solar activity levels. The work presented herein highlights 450 km as a guideline for future missions to VLEO, in the face of past inconsistency and disagreement between academic sources. It would be useful for future research to use this definition when referring to VLEO, for consistency and clarity. However, it is not intended to replace the need for complete aerodynamic analysis as part of mission design.

Funding Sources

The DISCOVERER project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 737183. Disclaimer: This publication reflects only the views of the authors. The European Commission is not liable for any use that may be made of the information contained therein.

Acknowledgements

The author would like to acknowledge the support of her supervisors P. Roberts and N. Crisp in undertaking this research. She would also like to thank her co-workers for use-

ful comments and discussion.

Chapter 3

ADBSat: Methodology of a novel panel method tool for aerodynamic analysis of satellites

This chapter includes the author accepted manuscript version of the following paper:

L. A. Sinpetru, N. H. Crisp, D. Mostaza-Prieto, S. Livadiotti, and P. C. Roberts, “ADB-sat: Methodology of a novel panel method tool for aerodynamic analysis of satellites,” Computer Physics Communications, p. 108326, 2022, `https://doi.org/10.1016/j.cpc.2022.108326`, ISSN: 0010-4655.

Paper Content and Author Contribution

The methodology behind panel methods, and ADBSat in particular, is described in this paper. Differences and similarities to other panel method programs are presented, with the aim of establishing the novel contributions of ADBSat to the field of aerodynamic analysis of satellites in VLEO. While ADBSat follows similar techniques to previous programs, it boasts a new shadow analysis algorithm, a new interplay with external computer-aided design (CAD) modelling tools, and a new modular design capable of easily integrating additional GSIMs.

An overview of drag models in VLEO is presented, concentrating on the physics underpinning them. This begins with the description of free-molecular flow (FMF), and an overview of GSIMs. Important parameters for drag modelling such as energy and momentum accommodation coefficients are explained. The relationship between the coefficients of drag, lift, pressure and shear stress is explained. Each individual GSIM implemented in ADBSat at the time of writing (the Sentman, Schaaf and Chambre, Cercignani-Lampis-Lord, Storch, Cook, and Newton models) is described. The parameters required for each GSIM, their equations to calculate lift and drag, and recommendations for their use are given.

Specific to ADBsat is the method of importing the CAD model to calculate its drag, which is thoroughly described for future users. An analysis of the mesh resolution required to obtain an accurate result is presented. The specific matrix transformations, reference frames, and unique shading algorithm are also characterized, including a computational flowchart representation of the novel algorithm. An aerodynamic database for a model satellite obtained using ADBSat is then presented, at intermediate solar activity conditions and 200 km orbital altitude. The analysis is performed for two cases: firstly assuming the satellite is made of a single material, and secondly for a satellite which has different materials covering different surfaces of its body. It is mentioned that an accompanying paper, included in chapter 4 of this work, will thoroughly validate the program and its results.

The creation of ADBSat is not attributed to the author. Therefore, some of this work, particularly in sections 3.3 and 3.4 where the equations used by the program are detailed, are closely based on the work of its creator Mostaza-Prieto [96]. These particular sections of [96] have not been previously published in a scientific journal. Their inclusion was crucial to thoroughly explaining the workings of ADBSat. However, the writing, figures and formatting in these sections were completed by the author.

The author's contributions are as follows:

1. Performed de-bugging and made minor changes to ADBSat
2. Examined literature on previous panel methods to identify their similarities and differences to ADBSat
3. Examined literature on, and wrote descriptions of, GSIMs
4. Considered mesh quality checks and mesh resolution
5. Performed the example aerodynamic database analysis
6. Created all figures
7. Wrote the paper

The third author is responsible for the creation of ADBSat, with sections 3.3 and 3.4 being based on his previous work as described above. The second and fourth authors are responsible for useful additions to the program. The second and fifth authors are responsible for editorial comments, assistance with writing, and considerable improvements to the paper. This paper has been submitted to Computer Physics Communications and accepted for publication. A Program Summary section is included following the abstract, in accordance with Computer Physics Communications requirements for papers describing new programs. Any small errata on the published work are enclosed in square brackets, for example [erratum].

Abstract

ADBSat is a novel software that determines the aerodynamic properties of any body in free-molecular flow. Its main advantage is the fast approximation of the aerodynamics of spacecraft in the lower end of the low-Earth orbit altitude range. It is a novel implementation of a panel method, where the body is represented as a set of fundamental elements and the sum of their individual aerodynamic properties makes up the properties of the whole. ADBSat's approach treats the shape as a set of flat triangular plates. These are read from a CAD geometry file in the Wavefront format, which can be created with most common CAD programs. A choice of gas-surface interaction models is available to represent the physics of free-molecular flow under different conditions. Its modular design means that other models can be easily and quickly implemented. It also benefits from a new shading algorithm for fast determination of elemental flow exposure. An example case is presented to show the capability and functionality of the program.

Keywords: Panel method, free molecular flow, orbital aerodynamics, satellite drag, drag analysis

Program Summary

Program Title: ADBSat
CPC Library link to program files: (to be added by Technical Editor)
Developer's repository link: https://github.com/nhcrisp/ADBSat
Licensing provisions: GPLv3
Programming language: MATLAB
Nature of problem: Quickly and accurately determining the aerodynamics of satellites in free-molecular flow.
Solution method: A new implementation of the panel method has been devised. The satellite shape is passed to the program as a CAD model, comprised of a set of flat triangular plates. ADBSat then calculates the aerodynamic characteristics of each element using an appropriate mathematical model, and sums the contributions for the overall properties of the body. A novel shading algorithm identifies and removes the panels which do not contribute to the calculations due to being protected from the flow by other body features. ADBSat also has the capability to account for different materials within the shape.
Additional comments including restrictions and unusual features: MATLAB's Aerospace Toolbox is required for the determination of environmental parameters, unless otherwise provided by the user. As the program takes as an input a completed model of the spacecraft, the user is responsible for all mesh quality checks. The methods employed therein are only valid for strict free-molecular flow, which the user must ensure. The ac-

curacy of the method decreases for surfaces with high concavity or multiple particle impingement.

3.1 Introduction

The panel method technique is an established method of calculating the aerodynamic properties of satellite geometries in rarefied flows, such as those in the low-earth orbit (LEO) regime [97]–[100]. In essence, it involves reducing a model of the spacecraft into a number of simple segments, and calculating the individual aerodynamic contribution of each segment. The contributions are then summed to give the aggregate aerodynamic properties of the body.

Panel methods are just one of the options available for aerodynamic analysis. They are an analytical method, applying closed-form equations to determine the drag contribution of each building block of the body. While relatively simple to implement, they have historically been disadvantaged by the difficulty in determining panels that are shielded from the flow [97] and handling concave shapes that promote multiple particle reflections. Options for aerodynamic analysis that can handle these shortcomings are numerical methods, such as Direct Simulation Monte Carlo (DSMC) and Test Particle Monte Carlo (TPMC) [101], which rely on modelling the movements of particles in a simulation domain. The trade-off for any such method is the high computational complexity. They have become increasingly popular in recent years with the improvement in power of computational facilities and the advent of assistive schemes, such as interpolation and response surface modelling, that decrease the number of simulations necessary [102]. However, even with these improvements to numerical methods, panel methods are still the quickest way to gain an understanding of the aerodynamics of a realistic satellite body.

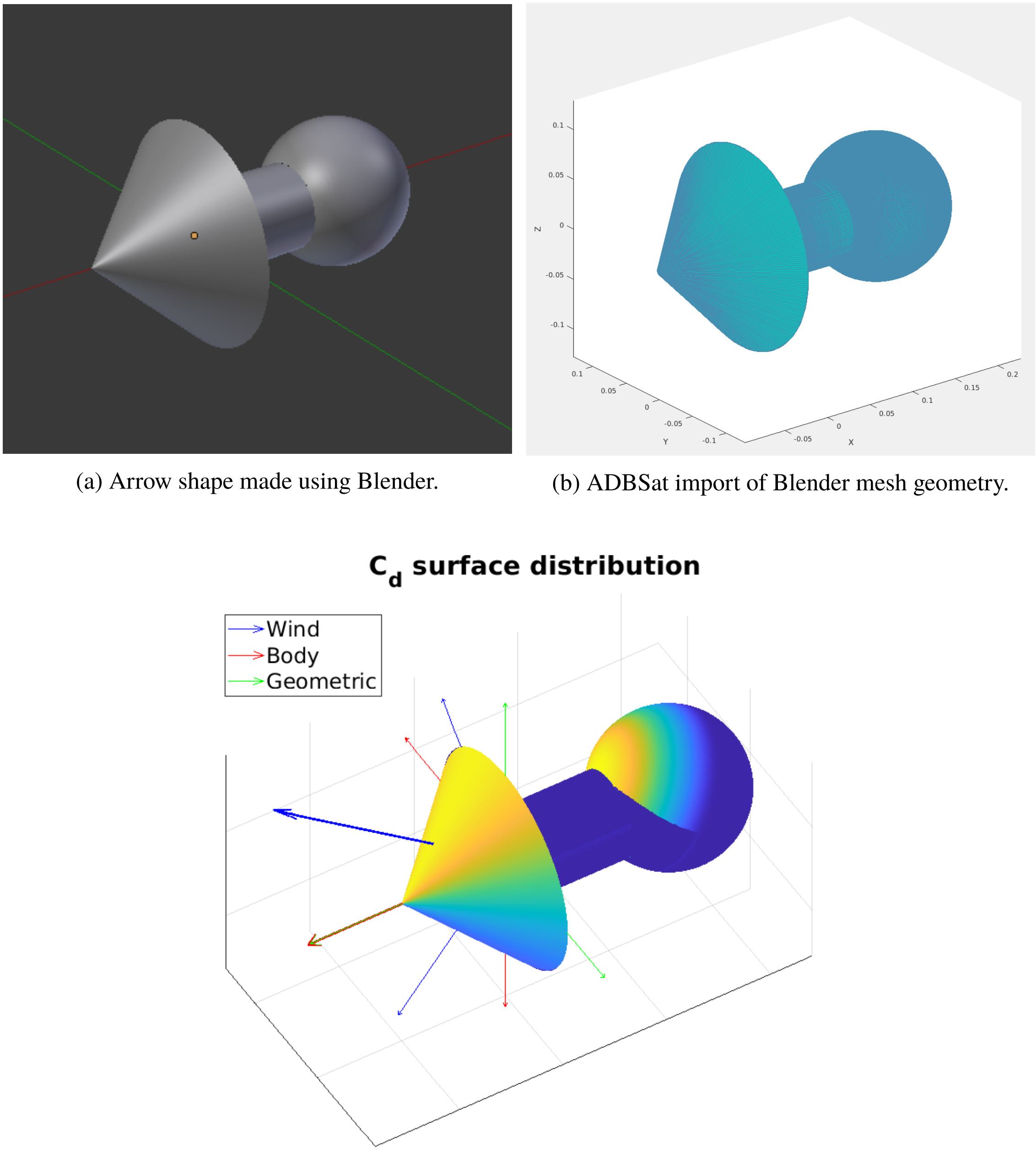
ADBSat is a novel implementation of the established panel method which aims to overcome some of the aforementioned shortcomings. Its aim is to output a useful approximation of the aerodynamic properties of complex spacecraft within the limitations of panel methods. Additionally, it is simple and easy to implement, while further reducing computational time compared to previous applications. One common feature between this and previous applications [97] is that it is primarily built within the MATLAB® environment, making use of the programming language's matrix-based methods to reduce computational time and load. However, it is distinct in that it does not limit the user to a single gas-surface interaction model (GSIM) to describe the physics of spacecraft flight in the rarefied atmosphere. Instead, six of the most widely used GSIMs for space applications have currently been implemented, allowing the user to choose among these to best represent their specific scenario. The modular structure of the program supports the straightforward addition of others. Thus, the program has been purposefully built to be adaptable to further developments in our knowledge of the physics of free-molecular flow (FMF).

Additionally, unlike previous implementations, ADBSat does not have an in-built method for building or otherwise describing satellite shapes. Instead, it takes as an input a model created using any computer-aided design (CAD) program, output in the common Wavefront file format, which is processed into a set of triangular plates for analysis. This makes the geometry easily transferable across different modelling and simulation programs, and thus renders any obtained results easily verifiable. This will undoubtedly prove useful in the context of a wider mission design framework, in which multiple aspects of the design must be integrated in a concordant way.

The long-standing problem of efficiently calculating panel exposure to the flow has also been tackled in a novel way. Due to the hyperthermal flow assumption, which assumes that the bulk gas velocity is much greater than the individual thermal velocities of the particles, the panels that are shielded from the flow by upstream components are assumed to have zero aerodynamic pressure. In other words, they do not contribute to the aerodynamic properties of the spacecraft. This is in line with previous similar implementations [97], [103]. However, a new shading algorithm is used to identify these panels [96], based on a 2-dimensional projection of the triangular elements that the program uses to represent the geometry. There is no need for bounding boxes or subdividing the geometry multiple times, as has been previously done. While still being the most computationally expensive algorithm employed by the program, the novel algorithm significantly speeds up processing time. Thorough validation of the shading algorithm, including a discussion of its limitations, is presented in an accompanying paper [92].

Figure 3.1 shows the steps of importing an example arrow geometry and analysing its aerodynamic properties using the shading determination algorithm. This shape was first designed using the Blender CAD software [104] as shown in fig. 3.1a. It was then imported into ADBSat through the included import function, the results of which can be seen in fig. 3.1b. Furthermore, on examining the aerodynamics of the shape at an angled flow, the effects of the shading algorithm can be seen in fig. 3.1c. In this image, the indigo panels are omitted from the final aerodynamic calculation.

Furthermore, an optional solar coefficient calculation for the body is available to the user. This addition to ADBSat has not been present in previous implementations. The same model geometry is utilised as for the aerodynamic analysis. From a systems engineering perspective, this consistency between geometric models across multiple analyses is desirable for accuracy, rather than using different representations in different software. The solar coefficient calculation can be used to determine the solar radiation pressure accelerations generated on the centre of mass of the spacecraft, which is invaluable for orbit determination. This is currently an optional extra which has yet to be fully tested and validated.



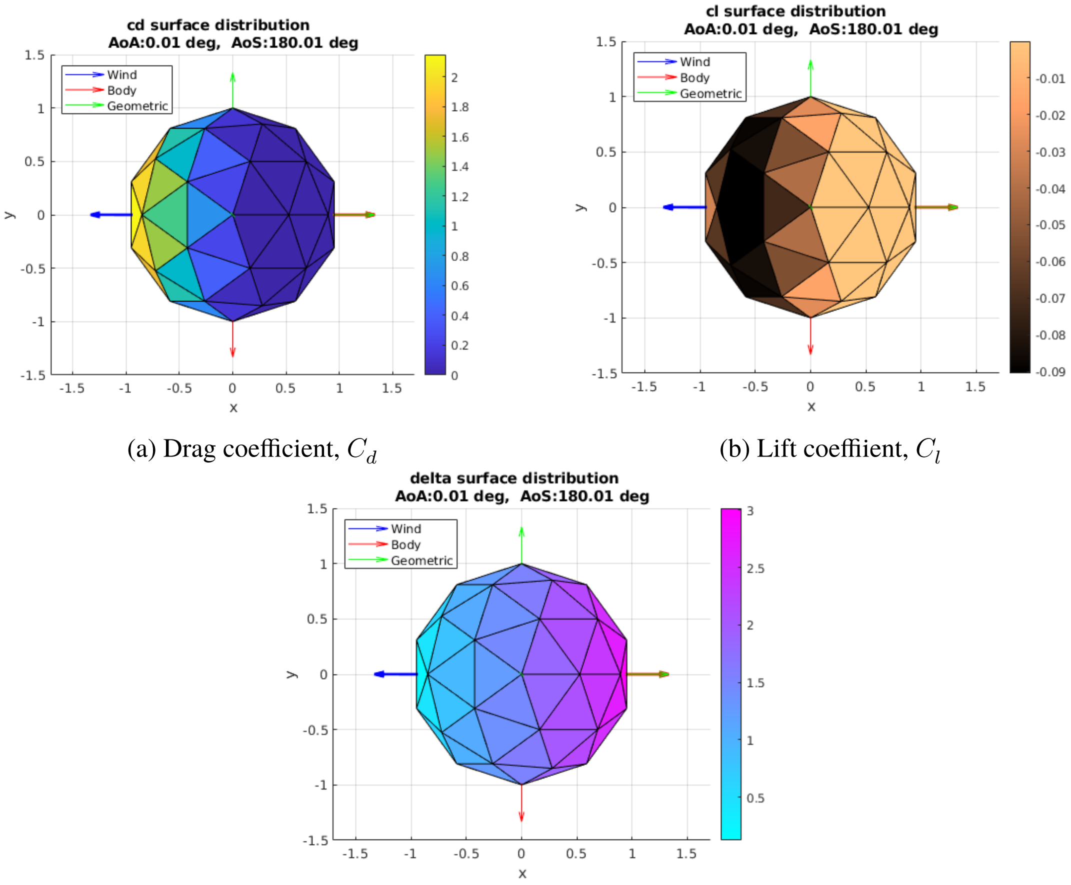
(a) Arrow shape made using Blender.

(b) ADBSat import of Blender mesh geometry.

(c) Analysis at an angled flow demonstrates the shading algorithm. The indigo panels do not contribute to the aerodynamic coefficients.

Figure 3.1. Analysis of an example geometry showing the progression from CAD modelling software, to ADBSat import, to aerodynamic analysis involving the shading algorithm.

The outputs, which can include solar and aerodynamic coefficients, can be either for a single orientation or a database of values at various incidence angles. They are saved in a MATLAB workspace variable file (with a ”.mat” extension). This file also contains relevant parameters that have been used in the calculations, such as the angle of attack (α), angle of sideslip (β), projected area (A_{proj}), reference area (A_{ref}) and length (L_{ref}). These files can either be loaded into MATLAB and viewed, or accessed directly through a MATLAB program to use the values. There is also an optional graphical output that can display the distribution of plate angle, or any of the coefficients of drag, lift, pressure, and shear stress, across the object. An example of this graphical output is shown in



(a) Drag coefficient, C_d

(b) Lift coeffiient, C_l

(c) Angle between the plate and the flow, δ (rad)

Figure 3.2. Example graphical outputs of ADBSat for a quasi-spherical polyhedron.

fig. 3.2, for a quasi-spherical polyhedron.

This work details the implementation and theory behind ADBSat, with a focus on novel contributions. The methodology is thoroughly explained. Potential uses alongside an example case are also discussed. Extensive testing and validation are addressed in an accompanying paper [92].

3.2 Background

The bottom end of the LEO regime, also known as very low Earth orbit (VLEO), has become an increasingly attractive orbital regime for extended satellite missions in recent years. Usually described as 100 km to 450 km, this range of orbital altitudes provides varied benefits for space platforms, in particular those with a focus on Earth observation [2]. However, it also suffers from a significant drawback: aerodynamic drag caused by interaction of the spacecraft with the residual atmosphere. The characterisation of drag at these altitudes has been attempted by various authors throughout the years [101], [105]–[114]. Yet, it is only recently that we have begun to more thoroughly understand this unique challenge on low-altitude spacecraft flight.

Satellite flight in VLEO occurs mostly in FMF, as characterised by the Knudsen number, K_n, given in eq. (3.1). In this equation, λ is the mean free path of the particles in the residual atmosphere, and L is the characteristic length scale of the object.

$$K_n = \frac{\lambda}{L} \tag{3.1}$$

The Knudsen number defines the flow regime as follows [101]:

- $K_n \leq 0.1$: continuum flow
- $0.1 < K_n < 10$: transitional flow
- $K_n \geq 10$: FMF

Another criterion for FMF can be defined through the number density of the atmosphere, n_0. Although less commonly used than the Knudsen number, $n_0 \leq 10^{16}$ can signify FMF [115]. While the exact value of K_n and n_0 are dependent on the instantaneous atmospheric conditions, the criteria for FMF are generally satisfied in VLEO [101], [115].

Due to the small number of intermolecular collisions, the aerodynamic interaction is instead dominated by the collision of the gas particles with the surface. The physical theory behind these collisions is captured in the GSIMs. A range of GSIMs have been developed that make fundamentally different assumptions about the particle-surface interactions and the re-emission characteristics. The accuracy of these assumptions is related to characteristics such as the environmental conditions and the materials present [76], [101]. Most models rely on an energy accommodation coefficient α_E, though momentum accommodation coefficients are also used, and some models use a mixture of both for shear and normal contributions. These relate the GSIM to the material by quantifying the extent of energy or momentum transfer from a gas molecule to the surface [101]. α_E is given by eq. (3.2), and is always in the range $0 \leq \alpha_E \leq 1$. In this equation, E_i is the energy of the incoming particles, E_r is the energy of the reflected particles, and E_w is the energy that the reflected particles would have had, had they been completely accommodated to the temperature of the surface. The momentum accommodation coefficient, σ, is calculated in a similar manner.

$$\alpha_E = \frac{E_i - E_r}{E_i - E_w} \tag{3.2}$$

Due to the well-documented adsorption of atomic oxygen onto the materials of satellites orbiting in VLEO [111], accommodation is highest where atomic oxygen concentration is highest. As the atmosphere becomes more rarefied with height, α_E decreases. Thus, α_E for current materials used in spacecraft production will depend mostly on altitude.

Atmospheric conditions such as solar activity also influence its value. It is left to the user to select suitable values of α_E to provide as inputs to the program. The estimation of these coefficients often presents problems in terms of accuracy. Estimation methods include extrapolating from values obtained from laboratory measurements in different conditions than VLEO [116]–[118], or from empirical experiments with limited observability [100], [110], [119]–[121]. Pilinski, Argrow, and Palo [122] have also devised a Langmuir isotherm model that can be used to calculate the accommodation coefficient for a particular set of atmospheric parameters. This model has not been validated for all altitudes and values, but it is useful in cases where its output is $\alpha_E \geq 0.85$. Widely used and accepted values for LEO range from 0.6 to 1.0 [110], [111], [123]. Research is ongoing into materials that are resistant to adsorption of atomic oxygen and promote specular reflection. However, at the time of writing, there is no experimental data available for these novel materials.

A GSIM provides equations that can be used to analytically calculate the drag coefficient of a body, C_d. The drag force, F_d, can then be calculated using eq. (3.3) by using the atmospheric mass density ρ and the speed of the object relative to the surrounding gas u.

$$F_d = \frac{1}{2}\rho u^2 A_{ref} C_d \tag{3.3}$$

However, an accurate estimate of C_d can often be difficult to obtain, due to the significant uncertainty surrounding atmospheric conditions. The flow parameters, ρ and u, may vary unpredictably across orbits. In particular, u is normally assumed to be the speed of the satellite's orbit around the Earth, with the assumption of a co-rotating atmosphere. The unpredictable nature of thermospheric winds can decrease the accuracy of this assumption [124]. There is also a significant uncertainty associated with atmospheric density modelling that must be considered [75]. Furthermore, the somewhat arbitrary choice of A_{ref} can further impact accuracy. Historically, this has been defined as the cross-sectional area of the object perpendicular to the flow direction (i.e. the projected area) [125]. For complex spacecraft shapes in a variable flow, this may not be accurately determinable at all times. As the selection of A_{ref} and C_d are fundamentally linked, this directly translates to uncertainty in C_d. Thus, it is important to carefully consider the choice of these parameters when invoking ADBSat for such a calculation.

Currently, there are six possible options for the GSIM. Each outputs the drag, lift, pressure, and shear stress coefficients for a flat plate (C_d, C_l, C_p and C_τ respectively). Only one pair of $\{C_d, C_l\}$ and $\{C_p, C_\tau\}$ is calculated directly, with matrix eqs. (3.4) and (3.5) being used to convert between the pairs using the angle between the oncoming flow and surface normal vector δ.

$$\begin{pmatrix} C_d & C_l \end{pmatrix} = \begin{pmatrix} C_p & C_\tau \end{pmatrix} \begin{pmatrix} \cos(\delta) & \sin(\delta) \\ \sin(\delta) & \cos(\delta) \end{pmatrix} \tag{3.4}$$

$$\begin{pmatrix} C_p & C_\tau \end{pmatrix} = \begin{pmatrix} C_d & C_l \end{pmatrix} \begin{pmatrix} \cos(\delta) & \sin(\delta) \\ \sin(\delta) & -\cos(\delta) \end{pmatrix} \tag{3.5}$$

The six GSIMs available are:

- **Sentman:** A single accommodation coefficient is used to quantify energy transfer to the surface. Diffuse re-emission is assumed. A more realistic velocity distribution for incoming particles is used than that of more simplistic models, such as the Maxwell model [101]. It also accounts for the relative motion of the surface and atmosphere. It is most accurate when the assumption of complete diffuse re-emission is correct, which is true for VLEO. Many literature sources exist that employ this model in analysing satellite flight, giving probable values for α_E at different atmospheric conditions [105], [110], [111], [122]. Thus, this model is widely used for the examination of satellite flight in VLEO. Equations (3.6) and (3.7) detail the calculation of C_p and C_τ respectively, requiring α_E, δ, the speed ratio s, the wall temperature T_w, and the incident temperature T_i. The error function erf(x) is also used, and is described in eq. (3.8).

$$\begin{aligned} C_p = &\left(\cos^2(\delta) + \frac{1}{2s^2}\right)\left(1 + \text{erf}(s\cos(\delta))\right) + \frac{\cos(\delta)}{s\sqrt{\pi}} e^{-s^2\cos^2(\delta)} \\ &+ \frac{1}{2}\sqrt{\frac{2}{3}(1 + \frac{\alpha_E T_w}{T_i - 1})}\left[\sqrt{\pi}\cos(\delta)\left(1 + \text{erf}(s\cos(\delta))\right) + \frac{1}{s}e^{-s^2\cos^2(\delta)}\right] \end{aligned} \tag{3.6}$$

$$C_\tau = \sin(\delta)\cos(\delta)\left(1 + \text{erf}(s\cos(\delta))\right) + \frac{\sin(\delta)}{s\sqrt{\pi}} e^{-s^2\cos^2(\delta)} \tag{3.7}$$

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \tag{3.8}$$

- **Schaaf and Chambre:** Unlike the Sentman model which uses one accommodation coefficient, this model uses two: one each for tangential and normal momentum transfer. A more thorough description of the forces on the surface is thus possible [101], [106]. However, it is often more difficult to obtain realistic values for the two accommodation coefficients than for Sentman's one. Therefore, if suitable accommodation coefficients are available, it is recommended to use this model. If not, Sentman should be employed instead. Finding C_p and C_τ requires the normal and tangential momentum accommodation coefficients σ_N and σ_T, the ambient temperature T_{inf}, s, T_w, and δ. Equations (3.9) and (3.10) detail their calculation.

$$C_p = \frac{1}{s^2}\Bigg[\left(\frac{2-\sigma_N}{\sqrt{\pi}} s\cos(\delta) + \frac{\sigma_N}{2}\sqrt{\frac{T_w}{T_{inf}}} \right) e^{-s^2\cos^2(\delta)} + \\ \left([2-\sigma_N]\left[s^2\cos^2(\delta)+\frac{1}{2}\right] + \frac{\sigma_N}{2}\sqrt{\frac{\pi T_w}{T_{inf}}} s\cos(\delta) \right) \Big(1+\text{erf}(s\cos(\delta))\Big)\Bigg] \quad (3.9)$$

$$C_\tau = \frac{\sigma_T \sin(\delta)}{s\sqrt{\pi}}\left[e^{-s^2\cos^2(\delta)} + s\sqrt{\pi}\cos(\delta)\Big(1+\text{erf}(s\cos(\delta))\Big)\right] \quad (3.10)$$

- **Cercignani-Lampis-Lord (CLL):** This model also requires two accommodation coefficients, one for tangential momentum accommodation and one for normal energy accommodation [114]. It is primarily intended for DSMC applications, for which its mathematical formulation renders it particularly well suited. Its basis lies in a complex scattering kernel, for which closed-form solutions are not directly known. Instead, the closed-form solutions are based on modified expressions of the Schaaf and Chambre model that approximate the C_d output of the model when it is applied in DSMC [76]. Therefore, as the ADBSat implementation relies on these approximated closed-form solutions, it is not recommended to use this model independently. Its use should be restricted to cases for which comparable DSMC simulations, also employing the CLL model, are available.

  In this model, C_p and C_τ are calculated for each individual species in the atmosphere, denoted by the subscript j. The species parameters β_j, γ_j, δ_j and ζ_j are are available in literature for each species [126]. As well as the aforementioned accommodation coefficients σ_T and α_N, T_w and T_{inf} are also required.

$$\Gamma_1 = \frac{1}{\sqrt{\pi}}\Bigg[s\cos(\delta)e^{-s^2\cos^2(\delta)} + \\ \frac{\sqrt{\pi}}{2}\Big(1+2s^2\cos^2(\delta)\Big)\Big(1+\text{erf}(s\cos(\delta))\Big)\Bigg] \quad (3.11)$$

$$\Gamma_2 = \frac{1}{\sqrt{\pi}}\left[e^{-s^2\cos^2(\delta)} + s\sqrt{\pi}\cos(\delta)\Big(1+\text{erf}(s\cos(\delta))\Big)\right] \quad (3.12)$$

If $\alpha_N < 1$:

$$C_{p,j} = \frac{1}{s^2}\left[\left(1+\sqrt{1-\alpha_N}\right)\Gamma_1 + \frac{1}{2}\left(e^{-\beta_j(1-\alpha_N)^{\gamma_j}}\left(\frac{T_w}{T_{inf}}\right)^{\delta_j}\frac{\zeta_j}{s}\right)\left(\sqrt{\frac{T_w}{T_{inf}}}\sqrt{\pi}\Gamma_2\right)\right] \tag{3.13}$$

$$C_{\tau,j} = \frac{\sigma_T \sin(\delta)}{s}\Gamma_2 \tag{3.14}$$

In other conditions (i.e. $\alpha_N = 1$), the equation for $C_{p,i}$ is the same as eq. (3.9), with α_N substituted for σ_N. Equation (3.14) remains unchanged. Finally, the total aerodynamic coefficients are computed as the weighted sum of the coefficients for each molecular species. An example of this for C_p is shown in eq. (3.15). M_{avg} is the average mass of the mixture, χ_j is the species mole fraction, m_j is the species mass, and $C_{p,j}$ is the species pressure coefficient.

$$C_p = \frac{1}{M_{avg}}\sum_{j=1}^{N}\chi_j m_j C_{p,j} \tag{3.15}$$

- **Storch:** A key assumption of this model is hyperthermal flow, which exists when the ratio of flow velocity to molecular thermal velocity is small. Tangential and normal momentum accommodation are treated through separate accommodation coefficients, facilitating a more thorough description of the aerodynamic forces [108]. However, while in some cases the hyperthermal assumption is valid in VLEO, care must be taken to ensure that the error introduced by neglecting particle thermal velocities is not significant [76]. Furthermore, as with the Schaaf and Chambre model, the two accommodation coefficients must be carefully chosen. Therefore, this model is not recommended for use unless the parameters of the case are well defined, and the user is confident in both the choice of accommodation coefficients and the hyperthermal conditions. C_p and C_τ for this model are detailed in equations eqs. (3.16) and (3.17), requiring δ, σ_N, σ_T, the incident velocity V, and the average normal velocity of diffusely reflected molecules V_w. For all backward facing panels (i.e. $\delta > \frac{\pi}{2}$), both coefficients are set to zero.

$$C_p = 2\cos(\delta)\left(\sigma_N\frac{V_w}{V} + [2-\sigma_N]\cos(\delta)\right) \tag{3.16}$$

$$C_\tau = 2\sigma_T \sin(\delta)\cos(\delta) \tag{3.17}$$

- **Cook:** A basis in the Storch GSIM means that this model also deals only with hyperthermal flow. However, it introduces the simplification of only one accommodation coefficient. [96], [107]. As with the Storch model, it is important to ensure hyperthermal flow conditions are applicable before selecting this model for analysis. Furthermore, common errors must be avoided, such as the confusion of kinetic and atmospheric temperatures [101], [127]. As the Sentman model converges to this model for hyperthermal conditions, the Sentman model exhibits a broader range of applications, and is recommended instead. The parameters δ, α_E, T_w and T_{inf} are required to calculate C_d and C_l, as shown in eqs. (3.18) and (3.19).

$$C_d = 2\cos(\delta)\left(1 + \frac{2}{3}\cos(\delta)\sqrt{1 + \frac{\alpha_E T_w}{T_{inf} - 1}}\right) \tag{3.18}$$

$$C_l = \frac{4}{3}\sin(\delta)\cos(\delta)\sqrt{1 + \frac{\alpha_E T_w}{T_{inf} - 1}} \tag{3.19}$$

- **Newton:** The gas-surface interaction is determined by Newton's laws of motion. The particles are assumed to be hard spheres, losing all normal momentum on collision with the surface. This is a valid approximation in hypersonic flow at large Mach numbers and moderately small deflection angles. However, these conditions do not normally occur in VLEO, and it is not recommended to use this model for the usual satellite flight conditions. The equation for C_p of a hard sphere, based only on δ, can bee seen in eq. (3.20). C_τ is always zero.

$$C_p = 2\cos^2(\delta) \tag{3.20}$$

$$C_\tau = 0 \tag{3.21}$$

Thorough explanations and comparisons of the different GSIMs for different uses are widely available in literature [53], [76], [96], [101], [111], [115], [120], [128], [129]. As the choice of GSIM is key to the accuracy of the analysis, it is recommended that users of ADBSat are confident in and can justify the model chosen for their analysis.

3.3 Methodology

An important part of computational data estimation is the validity of the input model. In other words, it is important to ensure that the geometric model of the spacecraft input into ADBSat is accurate. Previous programs have attempted to define a body through some form of in-house geometry definition, for example providing a set of shapes that can be arranged to define a satellite [97]. ADBSat differs in that it takes as an input an existing file format common in CAD modelling, the Wavefront file format (with a ".obj" extension).

A Wavefront file represents the surface mesh by means of vertices and face elements. Each vertex is defined by a set of (x,y,z) coordinates, and each face element by a combination of three vertices. By definition this results in triangular faces, which ADBSat requires. Each line of the Wavefront file defines either a vertex (where the line begins with “v”) or a face (where the line begins with “f”). By default, vertices are stored in a counter-clockwise order; therefore, the surface normals are implicitly defined. A particularly useful feature of this file format is the ability to specify multiple materials (where the line begins with “usemtl”). The user can specify a different accommodation coefficient for each material, by providing as an input a list of accommodation coefficients in the same order as the materials specified in the Wavefront file, rather than a single value. ADBSat processes the model into a MATLAB formatted data file (with a “.mat” extension) and explicitly calculates the outward surface normal, n_p, surface area, a_p, and barycentre b_p for each element of the triangular mesh.

The Wavefront file format is standard for the representation of polygonal data in ASCII form. As a result, most current CAD software programs can export directly into it. It is thus accessible to the wider engineering community, among which general CAD knowledge is widespread. Additionally, the same model can be used for different analyses or as an input into different programs, without needing to be approximated or converted. However, it should be noted that as the meshing is independent of ADBSat, the user is responsible for the quality of the mesh. At present, ADBSat has no built-in mesh quality controls. Checks such as the removal of free-floating features, duplicate vertices, and non-manifold faces are left to the user. In particular, ADBSat cannot handle zero-area faces, and will output NaN values when these are encountered, leading to meaningless results.

One important aspect to consider in creating a computational mesh is how accurately this mesh represents the desired shape, particularly for any rounded surfaces. The number of triangular plates it would take to represent such a surface exactly tends to infinity. However, the more plates used, the higher the computational time taken. Thus, there is a trade-off between computational time and accuracy of the model. Such a case can be seen in fig. 3.3 where a sphere of radius 0.1 m using the Sentman model at 200 km altitude is represented by 40 through 79600 triangular plates. It can clearly be seen that the result converges for an increasing number of plates to match that of the closed-form Sentman solution, while a lower number of plates introduces a high fluctuation to the results. It should be noted that even with 79600 plates, the runtime for each individual case is still quick, needing only a few seconds on a single core machine. Thus, as can be seen on the right-hand axis of fig. 3.3, it is recommended for accuracy that rounded features have an average element area to total surface area ratio of 1.5×10^{-4} or lower. Below this point, the result is not particularly sensitive to fluctuations in the plate size. The consequences of the choice of plate size on the efficient of the shading algorithm is discussed in a complementary work [92].

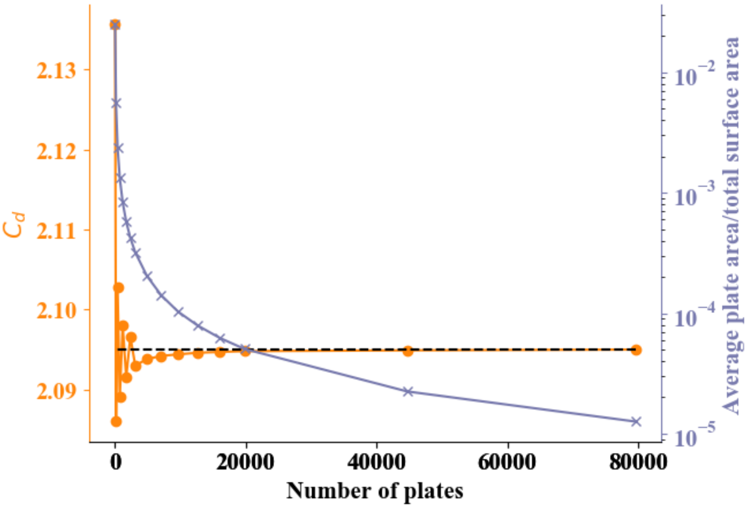


Figure 3.3. Analysis of the change in drag coefficient of a sphere as the number of plates used to represent it is increased. The closed-form Sentman solution is shown as a black dotted line, at $C_d = 2.095$.

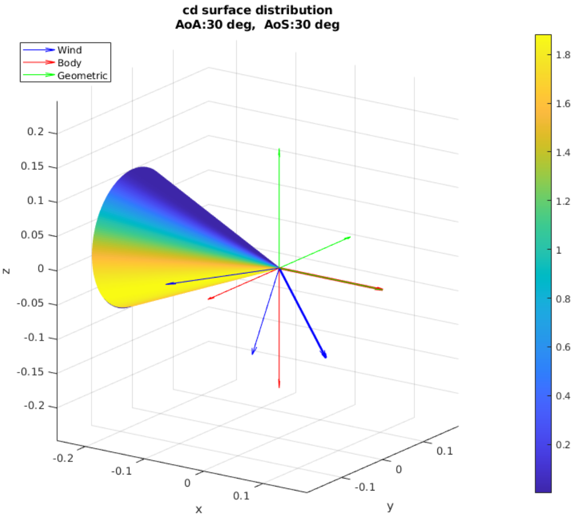


Figure 3.4. Geometric, body and wind frames for a cone at α, $\beta = 30°$. The x, y and z axes are aligned to the geometric reference frame.

In converting the model to its internal data file, ADBSat makes use of three reference frames: the geometric, body, and wind frames. Henceforth, the subscripts g, b and w respectively shall be used to refer to these frames. Figure 3.4 shows the three reference frames alongside a conical object pointing into the flow, with α and β both equal to 30°. Both these parameters are taken as inputs from the user. Inputting a range for either or both of these will result in the calculation of an aerodynamic database.

The geometric reference frame is that in which the Cartesian coordinates of the vertices are defined in the Wavefront file. By definition, the flow is aligned with the negative x-axis direction in the geometric frame when α and β are both zero. These angles define the orientation of the body with respect to the flow.

As is typical for flying bodies, the body frame is a right-handed co-ordinate frame with the positive direction of the z axis pointing towards the Earth. The transformation matrix used by ADBSat to define the body frame is L_{gb}, which transforms a vector expressed in the body frame into the geometric frame. It can be seen in eq. (3.22).

$$L_{gb} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \tag{3.22}$$

For a given α and β, the matrix L_{bw} transforms a vector from the wind frame into the body frame, as shown in eq. (3.23).

$$L_{bw} = \begin{pmatrix} \cos\alpha\cos\beta & -\cos\alpha\sin\beta & -\sin\alpha \\ \sin\beta & \cos\beta & 0 \\ \sin\alpha\cos\beta & -\sin\alpha\sin\beta & \cos\alpha \end{pmatrix} \tag{3.23}$$

Thus, the transformation from the wind to the geometric frame will be given by L_{gw} = $L_{gb}L_{bw}$. By using the definition of the wind frame, which is that the flow in this frame is always in the negative x-axis direction regardless of α and β, the velocity direction in the geometric frame, $\vec{n}_v$, can be calculated from eq. (3.24). Equation (3.25) shows how ADBSat calculates the angle δ_i between the velocity vector $\vec{n}_v$ and the surface normal of any element of the mesh $\vec{n}_i$.

$$\vec{n}_v = L_{gw} \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \tag{3.24}$$

$$\delta_i = \arccos(-\vec{n}_v \cdot \vec{n}_i) \tag{3.25}$$

By means of the chosen GSIM, this angle can be transformed into local pressure and shear stress coefficients for each panel, $c_{p,i}$ and $c_{\tau,i}$ respectively. These coefficients depend not only on the incidence angle, but also on a range of further parameters that characterise the gas-surface interaction and vary for different models. Finally, global force and moment coefficients for the body in geometric axes are obtained through eqs. (3.26) and (3.27).

$$\vec{C}^g_{fw} = \begin{bmatrix} C_{fwx} \\ C_{fwy} \\ C_{fwz} \end{bmatrix} = \frac{1}{A_{ref}} \sum_{i=1}^{n} (c_{\tau,i}\vec{\tau}_i - c_{p,i}\vec{n}_i)A_i \tag{3.26}$$

$$\vec{C}^g_{mw} = \begin{bmatrix} C_{mwx} \\ C_{mwy} \\ C_{mwz} \end{bmatrix} = \frac{1}{A_{ref}L_{ref}} \sum_{i=1}^{n} \vec{r}_i \times (c_{\tau,i}\vec{\tau}_i - c_{p,i}\vec{n}_i)A_i \tag{3.27}$$

In this equation, A_{ref} is the reference surface area, the default being half of the mesh surface area. L_{ref} is the reference length, defined as half the distance between the maximum and minimum x coordinates. A_i is the area of each plate. Vector $\vec{r}_i$ points from the geometric moment reference point to the barycentre of triangular plate i. Vector $\vec{\tau}_i$ is a unit vector in the direction of shear stress, calculated from eq. (3.28).

$$\vec{\tau}_i = \vec{n}_i \times (\vec{n}_v \times \vec{n}_i) \tag{3.28}$$

It is left to the user to reference $\vec{C}^g_{fw}$ and $\vec{C}^g_{mw}$ to the relevant area and length for the problem, according to eqs. (3.29) and (3.30).

$$\vec{C}^g_{fw2} = \frac{A_{ref}}{A_{ref2}} \vec{C}^g_{fw} \tag{3.29}$$

$$\vec{C}^g_{mw2} = \frac{A_{ref}L_{ref}}{A_{ref2}L_{ref2}} \vec{C}^g_{mw} \tag{3.30}$$

The moment and force coefficients can be translated into body or wind frames using the transformation matrices in eqs. (3.22) and (3.23). In addition, moment coefficients are obtained using the origin of the the geometric frame as the moment reference centre (MRC), which does not generally coincide with the centre of gravity (CoG). Equation (3.31) translates these coefficients to the CoG.

$$\vec{C}^{CoG}_{mw} = -\vec{r}_{CoG} \times \vec{C}_{fw} + \vec{C}^{MRC}_{mw} \tag{3.31}$$

Finally, the vehicle moment and force coefficients in the body axes, C^b_M and C^b_F respectively, can be obtained using the transformation matrix defined in eq. (3.22). This is explicitly shown in eqs. (3.32) and (3.33).

$$C^b_M = L^{-1}_{gb} C^g_{mw} \tag{3.32}$$

$$C^b_F = L^{-1}_{gb} C^g_{fw} \tag{3.33}$$

For clarity, these coefficients are defined in eqs. (3.34) and (3.35). Here, C_l, C_m and C_n

are the roll, pitch and yaw aerodynamic moment coefficients respectively and C_A, C_Y and C_N are the axial, lateral and normal force coefficients respectively.

$$C_M^b = \{C_l, C_m, C_n\} \tag{3.34}$$

$$C_F^b = \{C_A, C_Y, C_N\} \tag{3.35}$$

Choosing the GSIM to be applied, in essence, determines which pressure and shear stress coefficient calculations to apply. Should the user wish to define a new model, the template of the existing model scripts can be followed: code should be written which calculates any parameters needed to ultimately calculate the force coefficients for the body according to the new model. Thus, the intended versatility of ADBSat is apparent, and the easy implementation of future breakthroughs in FMF modelling is assured.

3.4 Shading Algorithm

Unassisted, GSIMs cannot account for panels which are shielded from the flow by upstream features. This is often a source of error when calculating aerodynamic coefficients [97]. ADBSat addresses this shortcoming by implementing a simple shadow analysis algorithm [96]. A flowchart of the algorithm is shown in fig. 3.5. Fundamentally, it works as follows:

1. Panels are split into two sets based on the angle between their normal and the oncoming flow vector:

   - set A contains forward-facing panels and those parallel to the flow ($\delta \leq \frac{\pi}{2}$).
   - set B contains backward-facing panels ($\delta > \frac{\pi}{2}$).

   Only set A can be shadowed, and only set B can shadow other panels.

2. The most downwind panel in set A is determined (panel Y). Similarly, the most upwind panel in set B is determined (panel Z).

3. Only panels in set A that are downwind of Z can be shadowed. Similarly, only panels in set B upwind of Y can shadow other panels. Selecting only these panels reduces set A to set A’, and set B to set B’. A pictorial representation of this is shown in fig. 3.6.

4. For each panel in set A’, a sub-set of B’ that are upwind of it is identified. The barycentre of the considered panel is then checked against a 2D projection of each panel

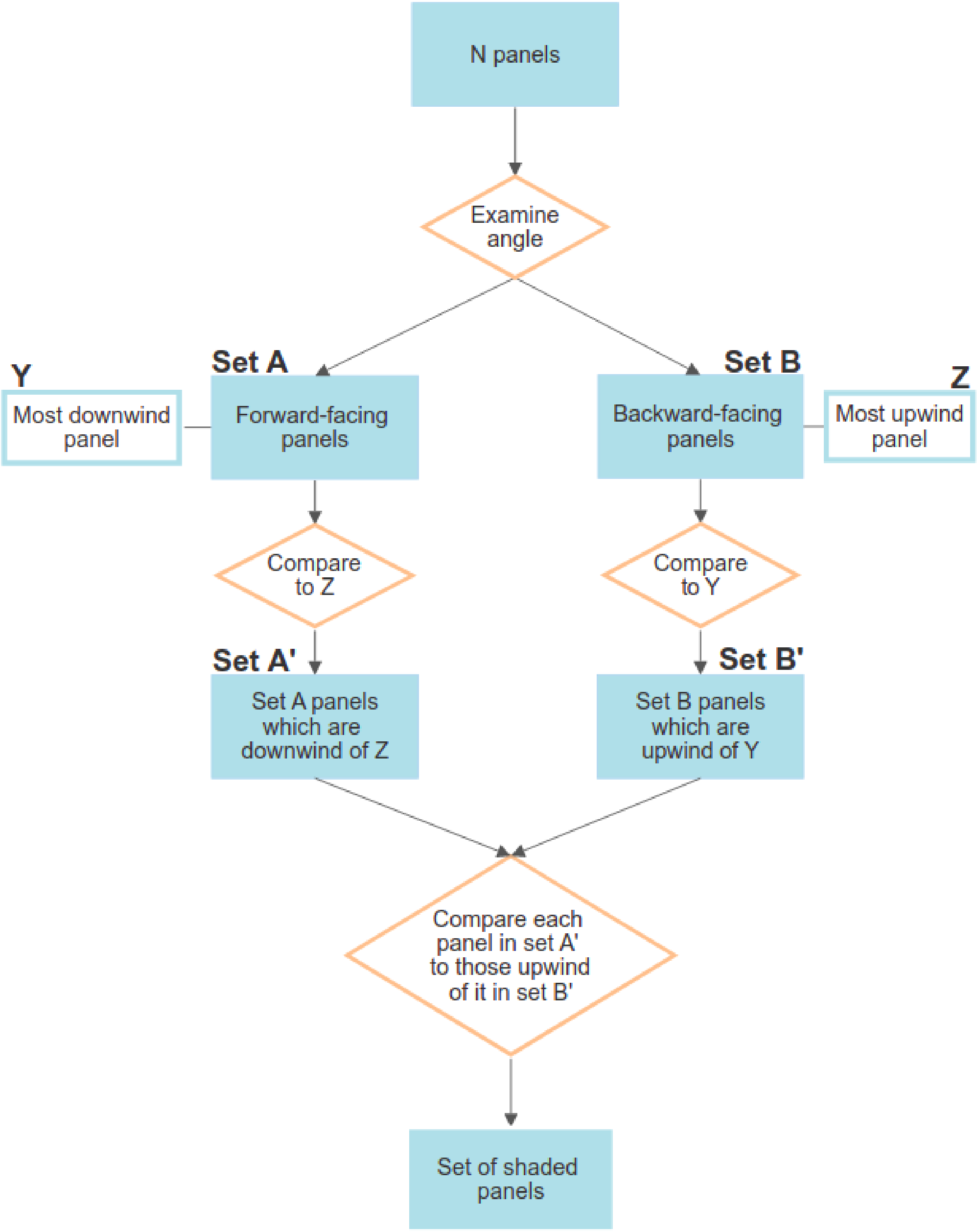


Figure 3.5. A flowchart of the shading analysis algorithm.

in the sub-set. If its barycentre falls inside any of the projections, then the panel is marked as shadowed and its contributions to any aerodynamic coefficients are assigned a value of zero.

This algorithm is an algebraic approximation based on geometric projections and is intended as a fast pseudo ray-tracer. In other words, it is not intended as a foolproof method of shading determination, but rather, a fast one. As panels cannot be part-shaded, a coarse discretisation of the shape may lead to incorrect results. Shapes with large flat sides, for example solar arrays, are particularly susceptible to this. The extent of this effect will be examined and thoroughly discussed in future work.

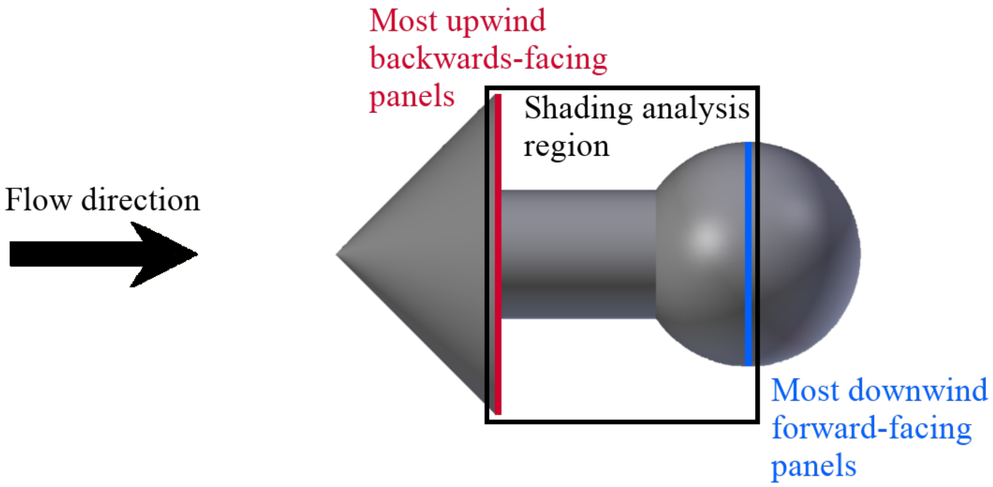
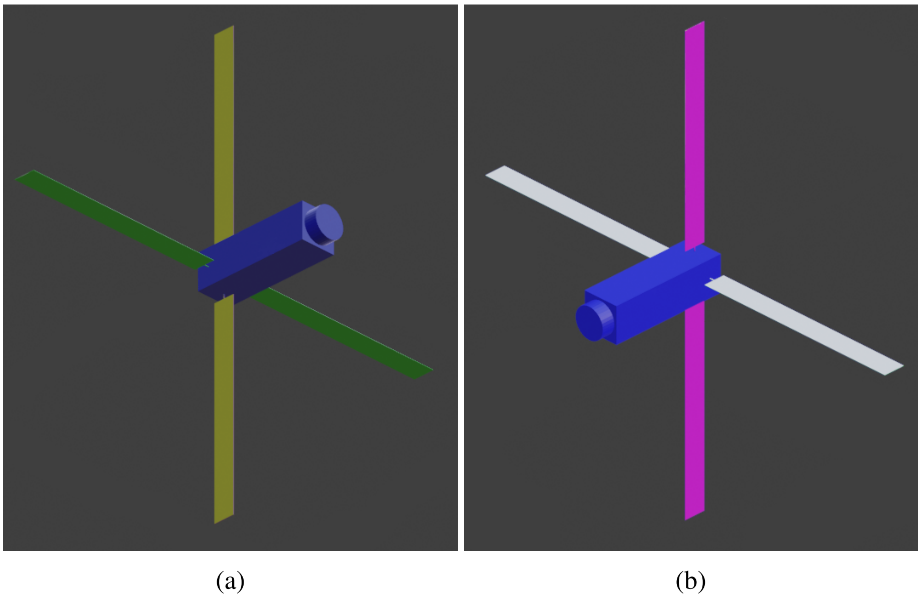

Most upwind backwards-facing panels
Shading analysis region
Flow direction
Most downwind forward-facing panels

Figure 3.6. An example of the shading analysis region of a test case.

3.5 Example case: SOAR

The Satellite for Orbital Aerodynamic Research (SOAR) is a 3U CubeSat mission proposed to study the effects of different materials in VLEO [32]. It aims to do this by using four steerable fins that can expose four different materials to the flow in order to discern their aerodynamic properties. It also carries an ion and neutral mass spectrometer (INMS) that can determine the composition of the VLEO atmosphere in-situ. A simplified version of its proposed configuration, with the fins parallel to the body of the spacecraft, was modelled using the SolidWorks and Blender CAD programs. It is comprised of 3233 flat triangular plates, with the different materials - four test materials and one body material - labelled from 1 to 5. Because different CAD programs can be used to easily generate the geometries, the process of model generation should be relatively simple for most users. Importing this model into ADBSat takes less than one second. It can be seen in fig. 3.7, from two different angles to show all five different-coloured materials. fig. 3.8 shows a side-on close-up of the body and part of the steerable fins after import into MATLAB. The individual triangular plates that make up the model can be seen therein.

The example case was examined at an altitude of 200 km, using Sentman's model. Intermediate solar activity conditions were chosen, using a reference date of 19 January 2015 at 00:00:00 and latitude and longitude (0,0). Representative solar indices corresponding to these conditions were chosen as 81-day average $F_{10.7} = 138.1$ and daily $F_{10.7} = 121.7$. A_p magnetic indices were in the range 2.9 to 9.0. A full aerodynamic and solar database was obtained for the object, with incidence angles in the range $-90 < \alpha < 90$ and $-180 < \beta < 180$, with a step size of one degree, resulting in 65341 individual combinations. Shading analysis was enabled. The total time needed to produce this database without graphical output was around 1.25 hours on an Intel® Core™ i7 vPro® quad-core machine. The mean time to run each calculation loop was approximately 0.07 s.



(a) (b)

Figure 3.7. The SOAR geometry viewed from two different angles. The different colours represent the five materials.

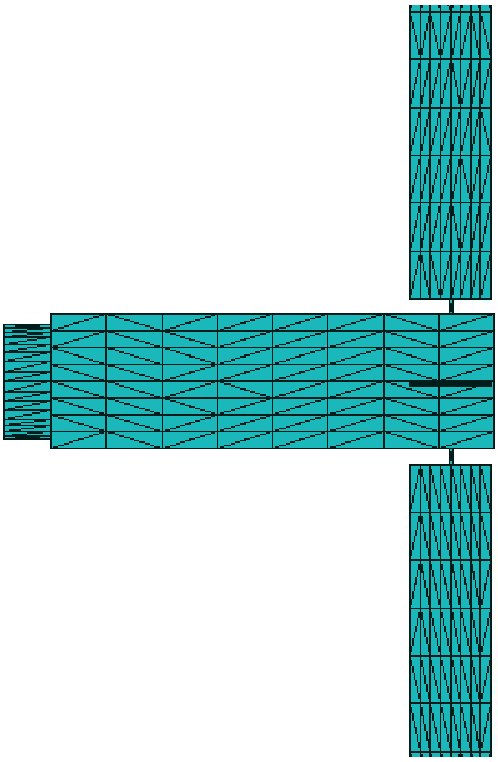


Figure 3.8. A side-on close-up of the main body geometry, showing the individual triangular panels which make up the geometry. Only part of the steerable fins is visible.

The aerodynamic database is stored in the aforementioned MATLAB workspace file. Here, the aerodynamic and solar force and moment coefficients for each combination of α and β can be accessed. As previously mentioned, for all calculations the reference area is half of the total surface area, and the reference length is half of the object size along the X axis. The projected area at each combination of angles is also stored in the output

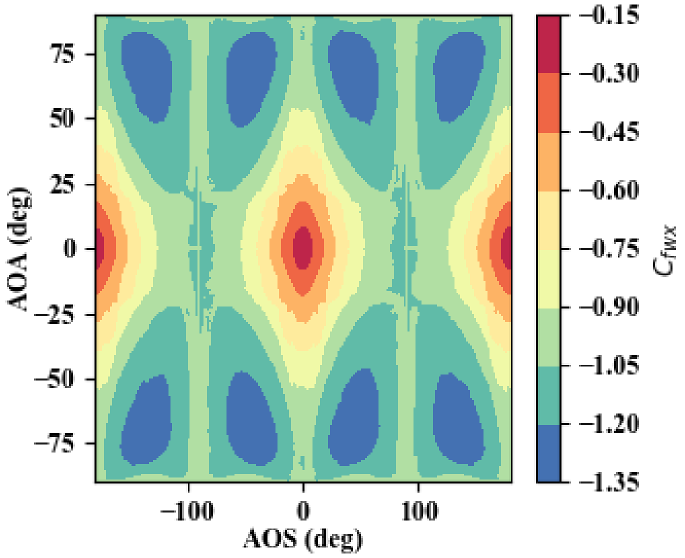


Figure 3.9. Aerodynamic coefficient along the x-axis for the example case where all surfaces have $\alpha_E = 1$. The reference area is half of the total surface area of the body.

file. The value of drag coefficient can then be obtained by referencing the aerodynamic force coefficient along the X axis in the wind co-ordinate system to the projected area, using eq. (3.29).

For the example case, two aerodynamic databases have been calculated. The first uses $\alpha_E = 1$ for all five materials. As the value of α_E is the only material characteristic taken into account, functionally this simplifies the model to having the same material across the entire surface area. A contour plot of the resulting force coefficient in the X axis direction of the wind frame can be seen in fig. 3.9. The values used to produce the plot are unchanged from the ADBSat outputs. It can be seen that as the value of α_E does not vary across the different materials, the plot is symmetric.

The second case is used to demonstrate variation in the gas-surface interaction properties, representing the lack of knowledge of the materials present on the flight model of SOAR. A value of $\alpha_E = 1$ is used for the body of the CubeSat and the four materials on the fins are given values of $\alpha_E \in [0.95, 0.9, 0.85, 0.8]$. As can be seen in fig. 3.10, this results in a more negative aerodynamic force coefficient, which can also be interpreted as higher drag. An asymmetric distribution with angle of attack and sideslip is also observed due to the difference in α_E between the different fins. As aerodynamic force coefficients are particularly susceptible to changes in α_E, this is expected.

Evaluating the accuracy of this output, particularly with regard to the drag characteristics of the object, is an important part of establishing the usability of ADBSat. Verification and validation of the software was undertaken through comparison with established methods of drag analysis and published aerodynamic coefficient values. The final set of test shapes is large, comprising of 14 test shapes analysed across a range of atmospheric

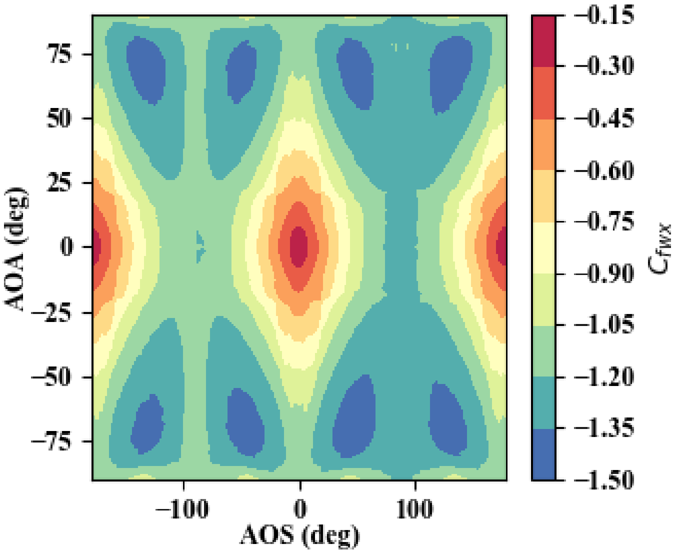


Figure 3.10. Aerodynamic coefficient along the x-axis for the example case where surfaces have varying values of α_E. The reference area is half of the total surface area of the body.

conditions and mesh configurations. The performance of the novel shading algorithm is thoroughly tested, as well as the influence of multiple reflections on the results. Five past satellite missions with publicly available aerodynamic data are also used to verify the program outputs. This extensive validation is thoroughly detailed in a separate paper which focuses on an explanation of our cases, methods, and results [92].

3.6 Conclusions

Much like other similar programs, ADBSat is designed to provide a fast, accurate approximation of the aerodynamic properties of spacecraft in free-molecular flow. It improves on previous implementations by exploiting existing knowledge of CAD modelling in the aerodynamic engineering community, taking as an input a model of the spacecraft which can be made in most common CAD software suites. This reduces the learning necessary to use the software, as well as the time needed for model design. It also implements a novel shading algorithm based on 2-dimensional projections of the triangular plates that make up the model. Additionally, the flat-plate representation of the spacecraft has been harnessed to calculate the solar coefficients, if required. Once validated, this will extend the utility of the software beyond the orbital regime in which aerodynamic forces dominate and into that in which solar radiation pressure dominates. Its modular design also means that new GSIMs and solar coefficient models can be easily implemented in the future.

The validation of the SRP model, and addition of further GSIM and SRP models, is undoubtedly the most significant possible future improvement to the program. This will extend the range of orbital cases for which the program can be applied, both in terms of the aerodynamics and the SRP conditions, and allow it to remain up-to-date with further developments in the field. The addition of mesh compatibility and quality checks will also be explored in the future, to aid users in obtaining useful outputs from the software. Furthermore, auxiliary functionality can be added to aid the user in modelling their case - for example, the program currently does not have the capability to estimate accommodation coefficients, which need to be input by the user. Integrating the Langmuir isotherm model [122] would provide an estimate based on the input atmospheric conditions. While this model is not universally applicable, some users may find it useful, and those for whom it is not could simply bypass it and input their own value(s).

The main advantage of ADBSat is the speed with which it outputs results, with a full aerodynamic and solar database for a model comprising of over 3000 elements taking around 1.25 hours to output. Although the runtime increases with the number of panels, this is still far more time-efficient than other existing methods. Thus, it can complement current existing methods such as direct simulation Monte Carlo (DSMC), particularly for preliminary design cases where there is a need for large aerodynamic databases for which this method is time-consuming. It can also be employed in cases in which there are many feasible satellite geometries being considered, such as at the early mission design stage. Additionally, it can be used for post-mission analysis and data interpretation. To summarise, this program provides a fast, practical solution for the aerodynamic analysis of satellite bodies, with a flexible approach which is conducive to the easy implementation of future advancements in atmospheric physics models.

Acknowledgements

L. Sinpetru would like to thank her supervisors and colleagues for their valuable discussion. This work was supported by doctoral funding provided by the University of Manchester.

Chapter 4

ADBSat: Verification and validation of a novel panel method for quick aerodynamic analysis of satellites

This chapter includes the author accepted manuscript version of the following paper:

Luciana A. Sinpetru, Nicholas H. Crisp, Peter C. E. Roberts, Valeria Sulliotti-Linner, Virginia Hanessian, Georg H. Herdrich, Francesco Romano, Daniel Garcia-Almiñana, Sílvia Rodríguez-Donaire, Simon Seminari, ”ADBSat: Verification and validation of a novel panel method for quick aerodynamic analysis of satellites”, Computer Physics Communications, vol. 275, p. 108327, 2022, https://doi.org/10.1016/j.cpc.2022.108327, ISSN: 0010-4655.

Paper Content and Author Contribution

Following on from the description of the methodology of ADBSat in the previous chapter, this paper describes in detail ADBSat testing and validation. It begins with a brief overview of VLEO and the challenge of aerodynamic drag, touching briefly on material covered in more detail in chapter 3. The theory of DSMC is described, with a particular focus on how numerical simulations differ from analytical methods such as closed-form solutions and ADBSat. The speed with which ADBSat can perform calculations is touted as a significant benefit, with the downside of its hitherto unknown behaviour.

ADBSat is tested by directly examining its results to those of comparable DSMC simulations, closed-form equations, and reported values in literature. As DSMC is the main method of testing, details of how to ensure a successful simulation, with a particular focus on an accurate computational mesh, are provided. Criteria are established for cell traverse time, cell size, the collision timescale, and the simulation particle density. DSMC is implemented through the OpenFOAM package as dsmcFoam, which can only apply the Maxwell GSIM. It is established that in DSMC and for a high accommodation coefficient, this GSIM reproduces results of more complex models well. However, because the physics of non-equilibrium events is not well captured in this GSIM, it is not suitable for closed-form implementation in ADBSat. Due to the basis of the Sentman model lying in a Maxwellian velocity distribution function, but with added details making it more realistic, it can be compared to dsmcFoam.

Details are given of the test altitudes and relevant parameters, such as solar conditions, accommodation coefficient, satellite wall temperature, and magnetic indices. Convergence time for DSMC simulations is found by examining the total number of simulation particles, N and total linear kinetic energy K_e for each time-step. Running the simulation well past the point at which N and K_e stabilize and comparing the values at each time-step to the final simulation value allows for a determination of the stable time-steps, i.e. those from which the aerodynamic data can be examined. The values of drag coefficient output by dsmcFoam at each suitable time-step are assumed to have a Gaussian distribution due to the stochastic nature of DSMC, with the mean μ and standard deviation σ of the Gaussian taken to be the value of drag coefficient and its error margin.

Three categories of test shapes are compared to DSMC simulations. Category A is comprised of basic shapes which also have closed-form solutions in the Sentman GSIM. In strict free-molecular flow with a Knudsen number $K_n \geq 10$, ADBSat is found to be within the error margins of DSMC and consistent with the closed-form solutions. The shading algorithm is specifically tested through Category B shapes, all designed to have some form of self-shading. It is found that the size of the flat plates used to represent a shape are important for accurate shading analysis, due to the algorithm being unable to partly shade flat plates - a plate is considered either fully shaded or not at all. Therefore, using large panels hinders an accurate portrayal of the shading. It is also found that when a lot of plates are at right angles to the flow and to each other, the inherent floating point error in MATLAB causes the shading algorithm to act erratically. A recommendation is made to examine shapes at a small angle to the flow, where the drag is better representative of that for head-on flow. Category C shapes test the influence of multiple particle reflections, and show that ADBSat can accurately predict the influence of most features, exhibiting inaccuracy only with deep concavities.

Finally, literature is examined to ascertain the drag coefficient of real satellites that have been flown in VLEO: Starshine 1-3, the Orion re-entry capsule, GRACE, and CHAMP. ADBSat was found to agree with most sources within 1-3%, which was determined to be the recommended error on its results.

The author's contributions are as follows:

1. Performed all research on testing and validation including methodology
2. Examined literature on DSMC practices and requirements

3. Implemented DSMC through OpenFOAM including mesh generation, determination of simulation parameters, and analysis of convergence and results
4. Designed test shapes and determined suitable aspects of the program to test
5. Researched other satellites in VLEO and their reported drag coefficients, and compared them to the results of ADBSat
6. Created all figures
7. Wrote the paper

The second and third authors were involved with editorial comments, assistance to writing, and considerable improvements to the paper. The remainder of the authors of this paper comprise the institutional leaders of the DISCOVERER project, in accordance with the stipulations of the DISCOVERER project agreement on publishing and dissemination.

Please note that this journal paper is written in the first person, unlike the rest of this thesis which is written in the third person. This is the format in which the paper was accepted for publication, and is reproduced faithfully here. While this may not fit with the third person grammar used in the rest of this thesis, adapting it to be written in third person would have required extensive re-writes and would have rendered the paper unfaithful to the accepted manuscript. Any small errata on the published work are enclosed in square brackets, for example [erratum].

Abstract

We present the validation of ADBSat, a novel implementation of the panel method including a fast pseudo-shading algorithm, that can quickly and accurately determine the forces and torques on satellites in free-molecular flow. Our main method of validation is comparing test cases between ADBSat, the current de facto standard of direct simulation Monte Carlo (DSMC), and published literature. ADBSat exhibits a significantly shorter runtime than DSMC and performs well, except where deep concavities are present in the satellite models. The shading algorithm also experiences problems when a large proportion of the satellite surface area is oriented parallel to the flow, but this can be mitigated by examining the body at small angles to this configuration ($\pm$ 0.1°). We recommend that an error interval on ADBSat outputs of up to 3% is adopted. Therefore, ADBSat is a suitable tool for quickly determining the aerodynamic characteristics of a wide range of satellite geometries in different environmental conditions in VLEO. It can also be used in a complementary manner to identify cases that warrant further investigation using other numerical-based methods.

Keywords: Panel method, free molecular flow, orbital aerodynamics, satellite drag, software validation, Direct Simulation Monte Carlo

4.1 Introduction

Recent years have seen significant interest in the sustained operation of spacecraft at the bottom end of the low-earth orbit (LEO) range. Work has been focused on flight at orbital altitudes of 100 km to 450 km, known as very low Earth orbits (VLEO) [1], [2], [20], [23], [25], [27], [30], [57], [59], [130], [131]. Operating a satellite in a low-altitude orbit can offer extensive benefits, in particular for Earth observation missions such as naval intelligence, fishing surveys, forestry management, and natural disaster support. Principally, the same payload orbiting closer to the Earth's surface will yield a better data resolution, with potentially more accurate positioning. Alternatively, a reduction in payload power will yield data of the same detail as a larger apparatus which orbits at a higher altitude. With this decrease in power requirements comes a desirable decrease in characteristics such as payload mass, size, and cost. All these factors together reduce the cost of manufacturing, launch and operation [1], [2], [57].

However, an important disadvantage in VLEO is the existence of atmospheric drag, which has a significant negative effect on spacecraft orbits. It leads to premature de-orbiting and a significant shortening of mission lifetime. The drag force can be quantified through the drag coefficient, C_d, which is invaluable to determining the drag response of a body in a fluid environment. ADBSat is a new program which determines the C_d of any body, both quickly and accurately [96], [132]. It is a novel implementation of a panel method, in which the equations describing the interaction between the satellite surfaces and the atmospheric particles, known as the gas-surface interaction model (GSIM) equations, are used to calculate the desired outputs [76]. This program overcomes the difficulty of prohibitively complex GSIM formulae by treating the spacecraft as a set of flat triangular panels. Hence, it reduces the detailed spacecraft geometry to a set of simple shapes, to which the GSIM equations are easily applied. A summation of the plates' contributions provides the results for the body as a whole. A shading algorithm based on 2-dimensional projections of the spacecraft panels is also employed, with the aim of increasing accuracy for concave geometries [132]. Past applications of the panel method such as DACFREE [98], [99], FreeMat [97] and FreeMac [103] and have suffered from a lack of reproducible and verifiable validation, and thus a limited knowledge of their accuracy.

A full, detailed description of the workings and implementation of ADBSat is available in an accompanying paper [132]. This paper aims to verify and validate ADBSat, by comparing its results to those of two other common methods of determining C_d, direct simulation Monte Carlo (DSMC), and closed-form equations. Here, we shall only outline those simulation features which are critical to the verification and validation process.

DSMC is a numerical technique that simulates the particles of the atmosphere by approximating many molecules as one simulation particle. First implemented by Bird [133], in recent times it has become " ...the de facto method for modeling rarefied flow in the transition regime" [101]. It relies on embedding a computer-aided design (CAD) model of the spacecraft geometry in a computational domain, into which simulated particles are inserted. The drag force, which results from particles impinging on the shape, can then be measured. It accurately recreates the physics in this regime [134]–[136]. However, a high computational effort and time expense is required, due to the large number of particles in each simulation. As a result, its principal use to date for orbital aerodynamics has been in mission support [54], [137] as well as analysis of the aerodynamic characteristics of the finalized design of a spacecraft [115], [138], [139].

In contrast to DSMC, analytical approaches such as closed-form equations take advantage of the physics of flight in VLEO to avoid particulate modeling. In VLEO, the atmosphere is rarefied, and the gas molecules have a large mean free path. Therefore, the number of inter-molecular collisions is small, and the atmospheric drag is dictated primarily by the interaction between the molecules and the surfaces of a spacecraft. The aforementioned GSIMs are mathematical descriptions of this effect. These models include equations that can be solved to calculate a body's drag coefficient, C_d. However, due to the complex mathematics involved, they quickly become prohibitively difficult for anything beyond basic geometric shapes such as spheres, flat plates, and cones. [111].

The panel method involves breaking down complex geometries in such a way that the simple formulae of the GSIMs can be applied to obtain the total body drag and lift coefficients. They do not require the drastic simplification of complex shapes, as closed-form equations do. Thus, it can provide a much more accurate estimation of C_d than closed-form equations for complex geometries. When compared to DSMC, the execution time for such programs is faster, and the computational load lower, due to the absence of simulated particles. Thus, they are more suited to deliver aerodynamic insight at the mission design stage. With an increasing number of satellites operating in VLEO, this will undoubtedly prove invaluable in the near future. Analysis of the geometries of recent significant missions to VLEO such as CHAMP [20], GRACE [23], GOCE [25], and SLATS [27] reveals a trend towards simple spacecraft shapes. Further aerodynamic considerations would require a thorough investigation of a wide array of design options, for which DSMC is limited in suitability due to its long runtime. Closed-form equations are also unsuitable due to their inability to capture the fine details of satellite bodies. Thus, ADBSat is better suited to this application than either of the other two methods available.

While faster than DSMC, the accuracy of panel methods in VLEO has been hitherto largely unknown. ADBSat suffers from known limitations with regard to modeling phenomena such as multiple particle reflections between different components of the spacecraft body. Through validation, we have investigated a range of cases, both simplistic and realistic, and report here our findings with regards to scenarios for which it is particularly suited for analysis. A discussion is also presented of cases which display a significant inaccuracy. Comparison to DSMC constitutes our main method of validation, implemented through the open-source software suite OpenFOAM [140] as dsmcFoam [134], [139]. For simple geometries, cases are also compared to closed-form GSIM equations. Finally, we report the performance of ADBSat as compared to published literature results for real spacecraft.

The comparison is complicated by the fact that ADBSat does not give an error margin on its outputs. Some uncertainty is expected due to factors such as the decomposition of the body into flat plates and the use of an atmospheric model rather than on-orbit data. These effects are difficult to quantify on a case-by-case basis. Therefore, we use the combination of ADBSat, dsmcFoam, and available literature to propose a fixed error margin, calculated as a percentage of C_d, that reflects the accuracy of ADBSat when compared to other methods of drag analysis. This captures some of the effects and prevents false confidence in what erroneously seems like an absolute result.

4.2 Verification and validation methodology

4.2.1 Case equivalency between dsmcFoam and ADBSat

We regarded dsmcFoam as the benchmark against which the results of ADBSat were tested. It is well-validated and known to be reliable for both transitional and rarefied gas flows [134]–[136]. It is also frequently maintained, with freely available documentation.

The DSMC algorithm relies on splitting the computational domain into $[i]$ cells of side length $\Delta x_{[i]}$ and evolving particle motions by one time-step Δt at a time. [Cells may vary in size across the simulation domain, but the same time-step will be used for all.] A key assumption is that each simulation particle, represented by a position vector $\vec{r}$ and velocity vector $\vec{V}$, can represent many real particles [135]. The motion of particles is treated as being decoupled from collisions, such that the collisionless Boltzmann equation can be solved for each time-step. Once the motion has been propagated, the gas-surface and inter-molecular interactions are implemented.

The computational mesh is of utmost importance in DSMC. We incorporated CAD satellite geometries into the mesh using the *blockMesh* and *snappyHexMesh* utilities of the OpenFOAM package. Simulation parameters were chosen to satisfy a number of criteria:

1. **Cell traverse time:** DSMC particles must not cross an entire simulation cell during one time step. Violation of this criterion could lead to artificial viscosity [141].

$$v'\Delta t < \Delta x_{[i]} \tag{4.1}$$

Where v' is calculated using eq. (4.2).

$$v' = \sqrt{\frac{2kT}{\bar{m}}} \tag{4.2}$$

2. **Cell size:** $\Delta x_{[i]}$ must be on the order of, or smaller than λ [for the largest cells]. If this criterion is not fulfilled, there are many collisions per cell, resulting in the system approaching the continuum limit [139], [141].

$$\Delta x_{[i]} \leq \lambda \tag{4.3}$$

λ can be computed using eq. (4.4).

$$\lambda = \frac{1}{n\pi d^2} \tag{4.4}$$

3. **Collision timescale:** Similarly, Δt must be smaller τ, to maintain an appropriate number of collisions per cell.

$$\Delta t \leq \tau \tag{4.5}$$

This is relatable to the previous criterion through eq. (4.6).

$$\lambda = \tau v' \tag{4.6}$$

4. **Simulation particle density, ρ_n:** The number of real molecules represented by a single simulation particle must be purposefully chosen. The program loses accuracy if ρ_n is too low, due to an inflated number of collisions. Effectively, the result is an artificially lowered K_n [142]. As computational time scales with ρ_n^2, values which are too high needlessly increase simulation time [133]. A suitable range is $7 \lesssim \rho_n \lesssim 20$.

[The mesh was comprised of hexahedrons of quadrilateral faces, generally in the shape of rectangular prisms. Cell size was chosen to fulfil the criteria outlined above. Trial simulations were run, the outputs of which were used to calculate the relevant parameters, which were then examined to ensure consistency with the above conditions. In general, the domain mesh is at a size chosen to fulfil the conditions above, which depends on the scenario being simulated. Refinements are applied to the front face of the object

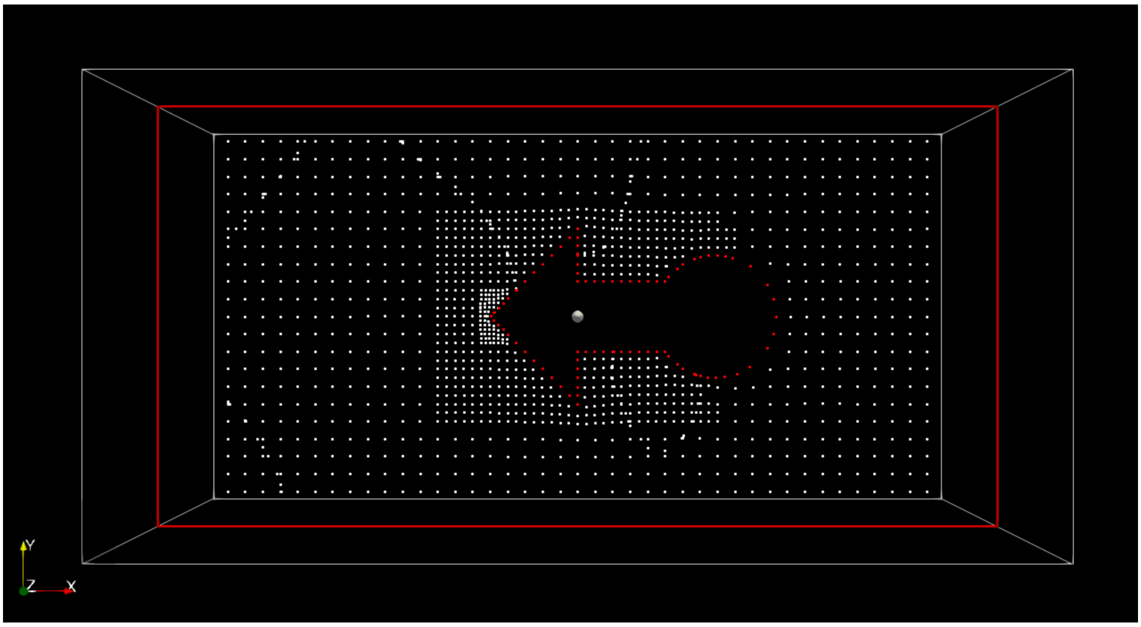


Figure 4.1. [An example mesh used to perform DSMC on an arrow-shaped object. Each white point represents the centre of one simulation cell, with the red points highlighting the shape. The refinement towards the front of the object can clearly be seen.]

and any other faces where particle density is observed to be high. The area behind the object is not refined, as few particles reach it. Conditions 1, 2 and 3 were satisfied for all simulations. The lower bound of condition 4 was always satisfied, while the upper bound was sometimes locally exceeded for practicality - it would have taken more effort to refine the mesh further than the computational time increase due to the locally increased simulation particle density. An example mesh showing the smaller cells at the shape's surface and the larger cells behind it is shown in fig. 4.1.]

DsmcFoam applies the Maxwell GSIM, through the use of a Maxwellian thermal velocity distribution for the particles [134]. This allows the user to specify the fraction of the molecules that are diffusely re-emitted from the surface, with the remainder assumed to be specularly reflected. Reflection in VLEO with current typical spacecraft materials has been shown to be effectively diffuse [111]. Given a high degree of accommodation, which is observed in VLEO [111], this GSIM produces results very close to those of a more complex model which more accurately reproduces molecular dynamics [128], [129]. Therefore, the Maxwell model is appropriate for DSMC applications. However, this GSIM is not appropriate for panel methods, because of the fundamentally unrealistic way in which it describes the physics of non-equilibrium scattering events (those which differ from the average) [143]. These scattering events are integrated across the surface to obtain the mathematical expressions for C_d utilized by ADBSat, which magnify this inaccuracy.

The GSIMs currently available in ADBSat are the Newton, Sentman [105], Schaaf and Chambre [106], Cook [107], Cercignani-Lampis-Lord [114], [144] and Storch [108] models. The Newton model is also fundamentally inaccurate, and only included for estimation purposes. The Cook and Storch models would not yield the most general comparison to DSMC, as both are only applicable to the more limited case of hyperthermal flow. Of the remaining models, the CLL and Schaaf and Chambre models are dissimilar to the Maxwell model in their treatment of momentum accommodation, with both requiring two accommodation coefficients to the Maxwell model's one. This important distinction precludes the possibility of direct comparison. However, there are similarities between the Sentman and Maxwell models, with both using a Maxwellian velocity distribution function and assuming a diffuse re-emission profile. However, the Sentman equations include more physically realistic conditions through consideration of the random thermal motion of the molecules [76], [105]. Thus, we selected the Sentman model to compare with dsmcFoam.

This model requires the accommodation coefficient, α, to be specified. For the current available satellite materials, surfaces in VLEO are known to be contaminated with adsorbed oxygen. As a result, energy accommodation is assumed to be complete (α = 1) at altitudes up to 200 km. This value decreases at higher altitudes, as surfaces become less contaminated [110]–[112]. For these cases, we calculated α by employing the model described by Pilinski, Argrow, and Palo [122]. The assumptions in this model align well with our simulations.It is most accurate at altitudes under 500 km, making it particularly suitable for use in VLEO. While ADBSat has an input parameter to specify α, dsmcFoam cannot account for partial accommodation. We used the work-around devised by Pilinski, Argrow, and Palo [138] and implemented by Mehta, McLaughlin, and Sutton [145] to simulate incomplete accommodation in dsmcFoam. To paraphrase, we set the wall temperature of the satellite to be the temperature of the velocity distribution that the molecules would have, were they partly accommodated to the surface.

The accommodation coefficients we calculated are detailed in table 4.1 and broadly agree with those reported in other sources for similar altitudes [109]–[112], [122]. The satellite wall temperature at each orbital altitude considered is also presented in table 4.1.

Table 4.1. Calculated parameters for the orbital altitudes, using the model described by Pilinski, Argrow, and Palo [122].

Altitude, km	α	$T_{k,r}$, K
100	1	300
200	1	300
300	0.97	1546.2
400	0.81	7346.8

In order to calculate these accommodation coefficients, we needed to determine the corresponding flight conditions. Verification and validation of ADBSat was intended to be as general as possible, hence, we chose intermediate solar activity conditions for the test cases, based on the $F_{10.7}$ solar index [86]. We chose a reference date and location of 19 January 2015 at midnight, latitude and longitude (0,0). Explicitly, 81-day average $F_{10.7} = 138.1$ and daily $F_{10.7} = 121.7$. A_p magnetic indices were in the range 2.9

to 9.0. We took the orbital velocity of the body as a function of altitude to be the free-stream velocity, V_∞.

ADBSat requires these inputs from the user in order to translate them to atmospheric parameters, by applying the US Naval Research Laboratory's mass spectrometer and incoherent scatter radar model (NRLMSISE-00) [41]. While other atmospheric models exist, such as the Jacchia-Bowman model (JB2008) [42], [43] and NASA's drag temperature model (DTM) [44]–[46], this model was chosen for three reasons:

1. It has been compared to the other models [41], [45], [79] and to real satellite data [80], [146]. It was shown to be at least as good as the other models at reproducing realistic atmospheric conditions [80].
2. There is an ongoing effort to maintain and improve the model. Corrections include employing new experimental data to correct the outputs [22], [147]–[149].
3. It is available as a MATLAB package, and is therefore easy to integrate into ADBSat.

The data from the NRLMSISE-00 model can also be found online[1] alongside its source code[2]. By utilizing both sources, we maintained consistency between the manually specified DSMC atmospheric conditions, downloaded through the online tool, and the ones accessed by ADBSat through MATLAB. Thus, a direct comparison between the two methods of drag analysis was facilitated.

4.2.2 Analysis of dsmcFoam outputs

To calculate the aerodynamic forces, dsmcFoam integrates the pressure and skin-friction forces over a specified boundary, which in our case is the CAD geometry. This calculation is performed at each output time-step. The user must process the resulting set of C_d values.

Steady-state convergence is necessary for accuracy [139] i.e., the total number of simulation particles N and the total linear kinetic energy K_e must plateau. We ran the simulations until plots of N and K_e showed no significant change over the preceding 10,000 time-steps. To determine the convergence time t_c, we then used the final value of N as a reference and compared every other value in the set to this. We determined the t_c as the time-step at which the value reached within $\pm 0.5\%$ of the reference. A similar analysis was conducted for K_e. We used the later time-step of the two as the point of convergence. All values of C_d output before this point were discarded. Figure 4.2 shows a

[1] Available at: `https://ccmc.gsfc.nasa.gov/modelweb/models/nrlmsise00.php`, accessed 16/07/2021

[2] Available at: `https://www.brodo.de/space/nrlmsise/`, accessed 16/07/2021

graph of values of N and K_e for the simulation of a sphere at 200 km, scaled to the reference. The convergence time of 0.002 02 s is shown using a vertical gray line, which is equivalent to 4040 time-steps. The details of convergence for the entire sample of test cases is shown in table 4.2, with ”rejected” indicating the percentage of values after t_c which did not lie within ±0.5% of the final value.

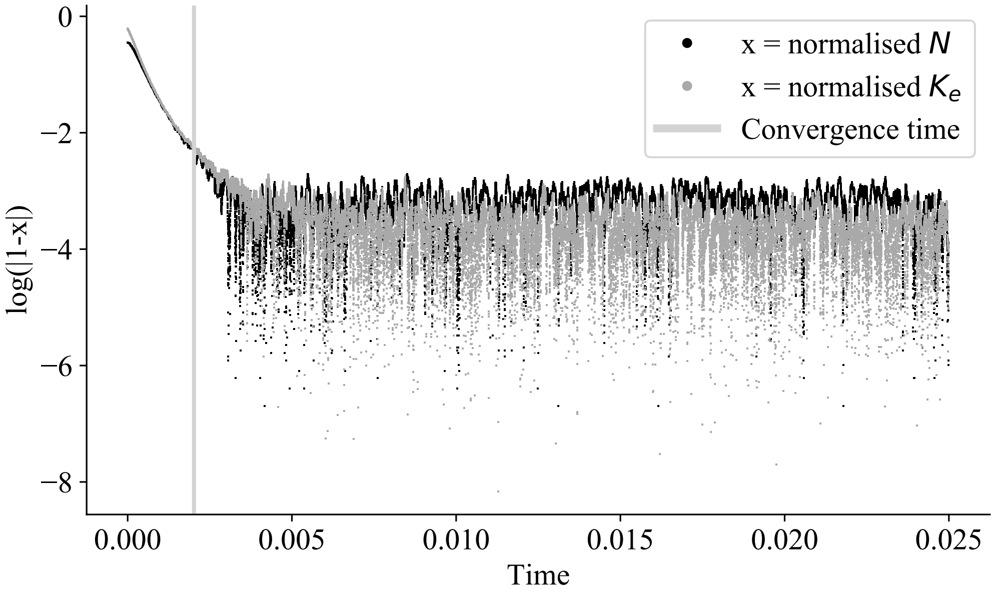


Figure 4.2. Analysis of dsmcFoam convergence for a sphere at 200 km altitude.

Table 4.2. Details of the convergence of the entire test sample.

	t_c	Δt_c	**rejected (%)**
Minimum	0.00019	380	0
Maximum	0.00508	10160	1.69
Mean	0.00208	4168	0.1054
Median	0.00193	3860	0.0299

As DSMC is a stochastic method [133], each measure of C_d is instantaneous, directly related to the individual particle positions and velocities at the time step at which it is calculated. Thus, the values exhibit scatter, due to the fluctuation of instantaneous force with time. This random scatter forms an approximately Gaussian distribution, as shown in fig. 4.3, where the Gaussian function based on μ and σ in the top left corner is shown as a dashed black line. We used μ as the final value of C_d. We interpreted σ, shown in fig. 4.3 in light gray, as the error on the mean. In later figures, where σ is large enough to be shown graphically, it is represented by error bars.

4.2.3 Description of Test Cases

We devised a number of initial shapes to test specific aspects of ADBSat, as shown in fig. 4.4. They range from 0.2 m to 0.9 m in length, with height and width being of the

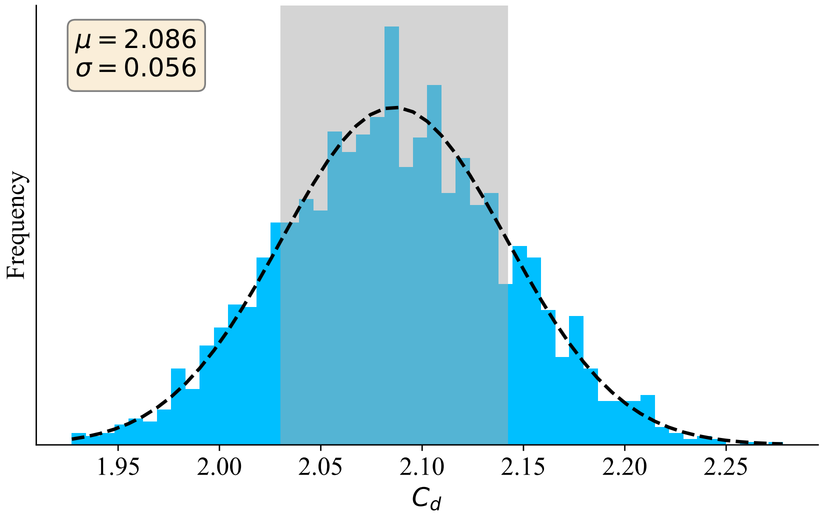


Figure 4.3. A histogram of C_d values for a sphere at 200 km, alongside the corresponding Gaussian function. The 1σ range is highlighted in gray.

same order of magnitude. We categorized them as follows:

1. **Category A**: basic shapes, (a) to (d) in fig. 4.4. The Sentman GSIM provides closed-form solutions not only for a flat plate with one side exposed to the flow, but for four additional basic geometries [105]. We chose these shapes so that results from ADBSat could also be compared to these solutions, as a secondary check of basic functionality.

2. **Category B**: shapes with self-shading, (e) to (i) in fig. 4.4. These shapes test ADBSat's shading algorithm by having some panels shielded from the flow by upwind features of the body.

3. **Category C**: shapes which promote multiple particle reflections, (j) to (n) on fig. 4.4. These shapes employ details such as angled panels or concavities on the forward-facing surfaces to promote reflection of the particles between faces.

We examined shapes in category A at orbital altitudes from 100 km to 400 km, in intervals of 100 km. All other shapes were examined at an altitude of 200 km only. We set AOA and AOS to zero for all shapes. Additionally, we also examined category B shapes a second time at a random selection of small AOA and AOS, ranging from 2.8 to 12.1 degrees, as a secondary test of the shading algorithm. Furthermore, these shapes were also discretised into smaller panels than the default CAD geometry, by manually selecting a target face on each shape and subdividing it. Our aim was to ascertain the influence of panel size on the output value of C_d.

Finally, we identified literature sources that offered drag coefficient data for real satellite missions. The methods employed varied across sources, comprising of analytical equa-

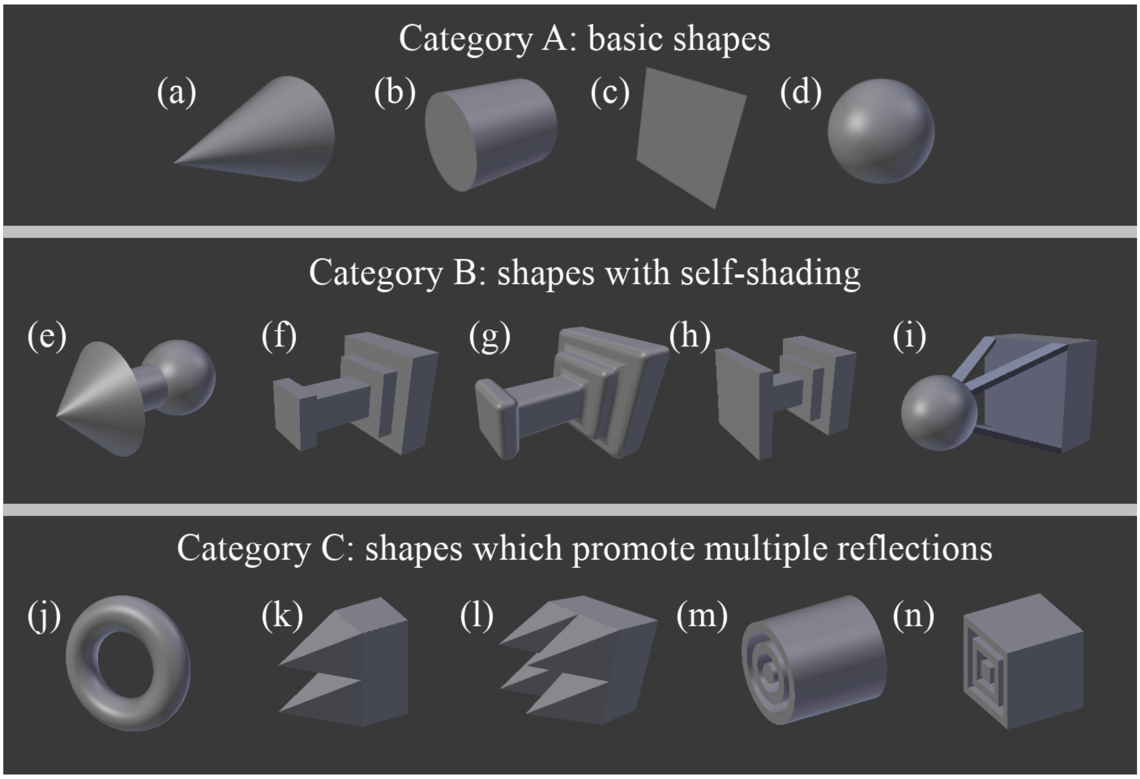

Category A: basic shapes
(a)
(b)
(c)
(d)
Category B: shapes with self-shading
(e)
(f)
(g)
(h)
(i)
Category C: shapes which promote multiple reflections
(j)
(k)
(l)
(m)
(n)

Figure 4.4. Selected shapes used for testing.

tions, DSMC, and free-molecular code. We reproduced as best as possible the simulation conditions reported, in order to directly compare our results to the data. The five objects examined, each at a range of conditions, can be seen in fig. 4.5. We built the CAD models of the Orion capsule [98] and simplified GRACE satellite [102] from technical drawings. The CHAMP geometry is the high-fidelity model of the spacecraft produced by March, Doornbos, and Visser [150]. Both Starshine satellite geometries are those used by Pilinski, Argrow, and Palo [138], and were obtained from the author via personal communication.

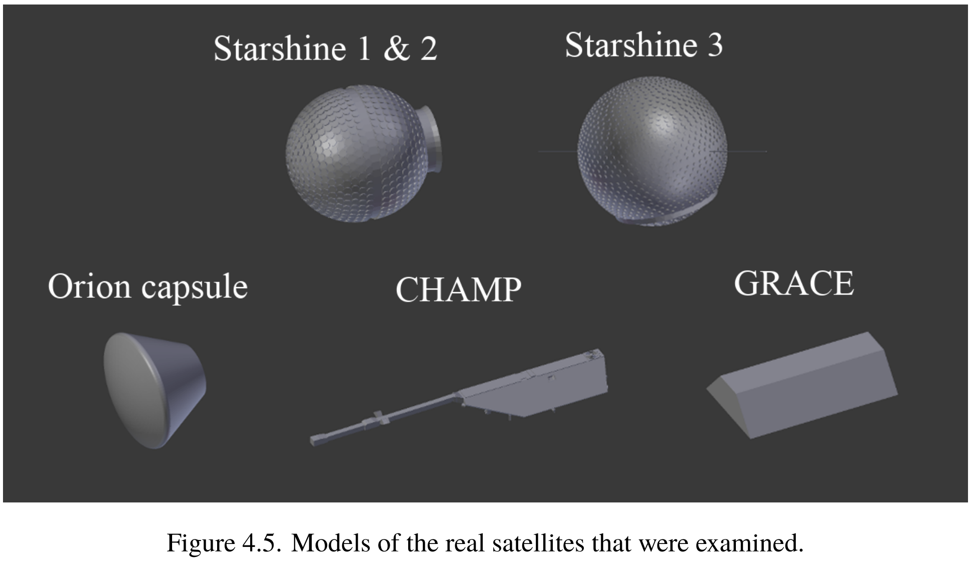


Figure 4.5. Models of the real satellites that were examined.

In actuality, we considered many other sources on the subject of aerodynamic analysis of real satellites, but many did not provide the data required to reproduce their results. Key simulation parameters were omitted which made comparison impossible. As a guide, table 4.3 outlines parameters which are necessary, and those which are advantageous, for reproducing published results.

Table 4.3. Simulation parameters which are necessary or advantageous for reproducibility of results.

Necessary	**Advantageous**
Choice of GSIM	Atmospheric particle density
GSIM parameters, such as α	Atmospheric temperature
Choice of atmospheric model	Mean molecular mass
Altitude	Diagrams of the objects
Solar activity levels	Free-stream velocity
Reference cross-sectional area	Knudsen number

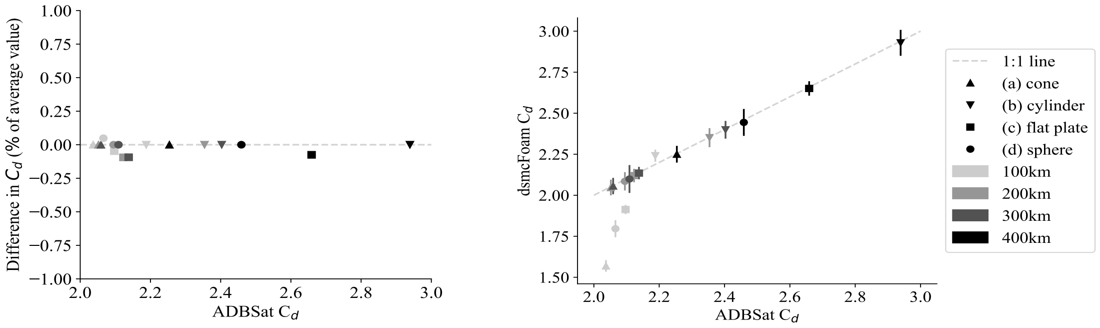
4.3 Results

4.3.1 Category A (basic shapes)

We first compared the results from ADBSat with those from the closed-form Sentman GSIM equations. It was expected that they would agree closely, as they are essentially two different methods of applying the same theory. Indeed, no case shows a difference in C_d higher than 0.1% between the two methods, well within the expected error limits of ADBSat. The percentage difference across the samples can be seen in fig. 4.6a, which shows a good agreement between the two methods. The slightly higher errors on the flat plate cases are due to the Sentman model assuming a zero thickness plate, which ADBSat cannot model. A small non-zero thickness was modelled instead.

Secondly, we verified results from ADBSat against those obtained from dsmcFoam. Runtime analysis revealed that that, for basic shapes, ADBSat is approximately five orders of magnitude faster than dsmcFoam: where dsmcFoam needs about 10^5 s to run a single simulation, ADBSat completes the same analysis in less than 10 s. Running multiple simulations, such as those at varying AOA and AOS that are required to obtain the aerodynamic database of a satellite, will compound this difference further. Thus, there is a clear time advantage to using ADBSat over dsmcFoam. Figure 4.6b shows a graph of the values of C_d obtained from these two methods. Results are consistent for all shapes, at all orbital altitudes except 100 km.

For the atmospheric parameters chosen, at an altitude of 100 km, $n_0 \sim 10^{17}$ and $K_n \sim 0.5$. Thus, the flow has become transitional, and the equations applied by ADBSat do not reflect the physics involved. However, they perform well at higher altitudes, where strict FMF conditions exist. This is consistent with existing literature which addresses the comparison of closed-form equations to DSMC [115]. As a result of the natural fluctuation of atmospheric conditions, attempting to define a lower limit of accuracy for



(a) ADBSat vs Sentman equations

(b) ADBSat vs dsmcFoam

Figure 4.6. Comparison between ADBSat, the Sentman closed-form equations, and dsmcFoam. Note the small scale of the y-axis in fig. 4.6a. [The error bars on the dsmcFoam values of drag coefficient represent 1σ, one standard deviation on the Gaussian distribution of drag coefficient values recorded after simulation convergence, as explained in 4.2.2.]

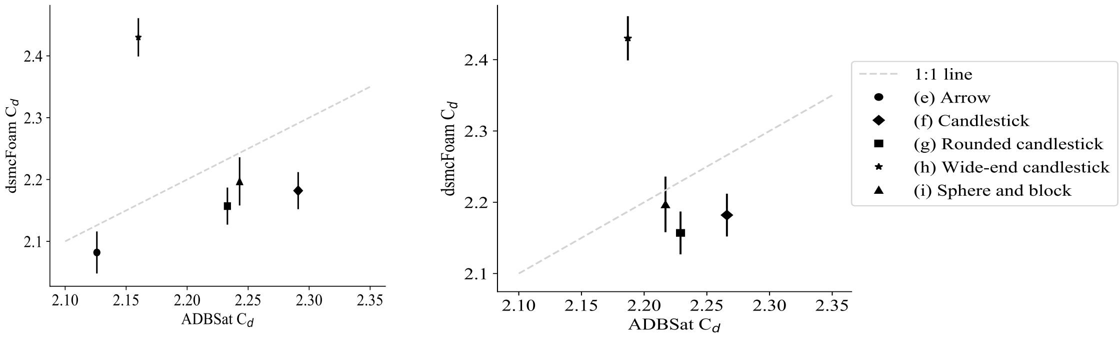
ADBSat in terms of height would be misguided. In conclusion, we have shown that the panel method applied in ADBSat is consistent with dsmcFoam for basic convex geometries across all altitudes where $K_n \geq 10$.

4.3.2 Category B (shading algorithm)

Having established that ADBSat performs well for simple shapes, we examined Category B shapes to ascertain the accuracy of the shading algorithm. The results of this analysis for head-on flow can be seen in fig. 4.7.

Initially, we represented the shapes using the fewest possible panels, in order to keep computational time to a minimum. However, as seen in fig. 4.7a, ADBSat yielded no results that agree with dsmcFoam. While two shapes show results within 2σ, with others showing as much as an 8.7σ difference, it is clear that the two methods are not consistent.

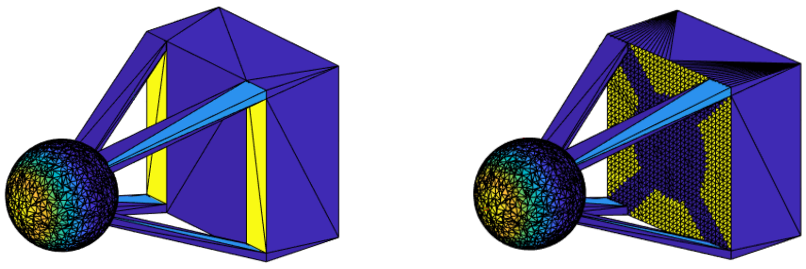
We also manually subdivided some of the large flat sides of shapes (f)-(i) into smaller



(a) shading, no subdivision

(b) shading with subdivision

Figure 4.7. ADBSat vs dsmcFoam results for category B, for head-on flow.



(a) Pre-subdivision

(b) Post-subdivision

Figure 4.8. Shape (i) before and after manual subdivision, with the flow head-on to the spherical feature. Blue regions contribute much less to the total C_d than yellow.

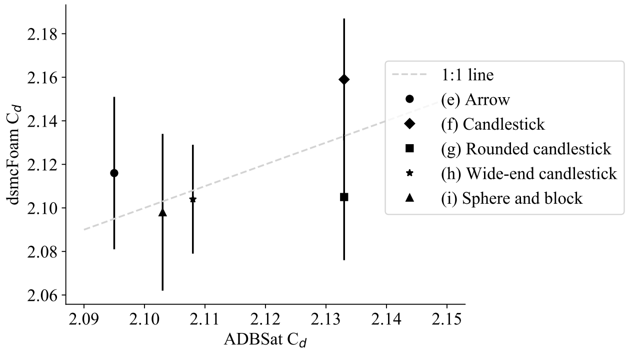
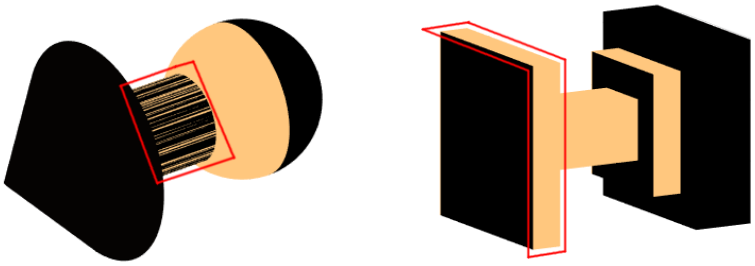


Figure 4.9. Analysis of the shading algorithm performance, for category B shapes at non-zero AOA and AOS.

panels during the CAD design process, in the hope that this would capture more realistic shading effects. This was unnecessary for shape (e), as the curved nature of the body was discretised into small panels by default. We saw some improvement in all the results, with no significant difference in runtime. However, only one was within 1σ of the value from dsmcFoam: shape (i). It exhibited unrealistic results for the shading prior to subdivision, which improved after, as seen in fig. 4.8. For other shapes, the difference before and after subdivision is less apparent. Therefore, while this effect does not fully account for the observed discrepancy, it can have a significant influence on the analysis of complex shapes.

In contrast, the same set of shapes analyzed at an angle yields more consistent results between the two methods, as seen in fig. 4.9. All five geometries now agree within 1σ with dsmcFoam. As each shape was examined at a different combination of random AOA and AOS, it is clear that the previous issues are seen only for head-on flow, when many panels are at an angle of $\delta \sim 90°$ to the flow and to each other. The cause of this is the inherent computational error of MATLAB, caused by the representation of the geometry using floating point values. When panels are at an angle of 90° to each other and to the flow, the 2D projection employed by the shading algorithm will project the barycenter of the downwind shaded panel exactly on the edge of the upwind shading panel. Thus, the barycenter is neither inside nor outside the shading panel. However, the shading algo-



(a) Shading for shape (e)

(b) Shading for shape (h)

Figure 4.10. Erroneously categorized panels with regard to shading, highlighted in red.

rithm must still decide which side of the edge it falls on. The small computational error which MATLAB carries will influence such a precise calculation heavily, resulting in an almost random classification of shaded/non-shaded panels. A visualization of this effect is seen in fig. 4.10. Panels classified by the program as shaded are in orange, with those not shaded in black. The edges of each panel are not outlined, for clarity. The area containing incorrectly classified panels is outlined in red. Figure 4.10a shows shape (e), for which all panels of the central cylindrical section should be shaded. Figure 4.10b shows shape (h), for which the edge panels highlighted should not be shaded. As the shaded panels are effectively removed from the final summation, too few shaded panels lead to an overestimation of C_d, while too many shaded panels will lead to underestimation.

However, when the flow is at an angle of $\delta \geq 0.1°$, the projection of the shaded panel's barycenter is no longer on the edge of the shading panel. Thus, MATLAB's small error is now negligible, and ADBSat yields much more accurate results. In summary, ADBSat can handle the majority of satellite flight scenarios with accurate shading analysis, except those where many large panels are at an angle of $\delta \approx 90°$ to the flow and to each other. As the CAD model of any geometry must be made independently of ADBSat, it should be apparent to the user whether or not shading analysis is required, and if yes, whether any AOA and AOS will pose an problem. For such cases, we recommend interpolating between multiple values of C_d that span across the problem case.

4.3.3 Category C (multiple reflections)

The comparison of category C shapes can be seen in fig. 4.11. The aim was primarily to ascertain the influence of multiple particle reflections. For the three simpler shapes, (j), (k) and (l), the results of ADBSat and dsmcFoam are in agreement. However, the two more detailed shapes, (m) and (n), show discrepancy between the two methods.

Under the assumption of diffuse re-emission, multiple particle reflections should not affect the overall result considerably, due to the shape of the reflected distribution. Unlike specular reflection, it has an element of randomness in the particle velocities [101]. We

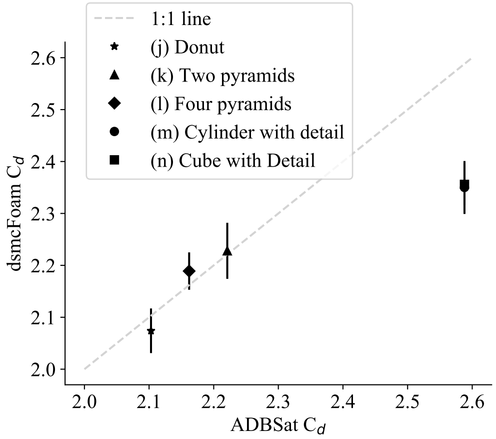


Figure 4.11. ADBSat vs. dsmcFoam results for category C shapes.

have determined that for simpler shapes, the effect is small enough that the results output by ADBSat are within the error limits of dsmcFoam.

However for shapes (m) and (n), the deep grooves of the forward face lead to particles being trapped in the indentations. This means that any panels inside the grooves contribute to C_d in a fundamentally different manner than the single particle reflections which ADBSat assumes. ADBSat essentially treats shape (m) as a cylinder with extra surface area perpendicular to the flow, caused by the sides of the indentations. Thus, it shows a higher C_d for this shape than for the plain cylinder. In contrast, dsmcFoam can capture the effect of the trapped particles and their reflections more accurately. It can take into account the effects of the localized increase in particle number - and thus, pressure and K_n - inside the grooves, which ADBSat cannot. dsmcFoam therefore finds a value of C_d for shape (m) closer to that of the unchanged cylinder. A similar analysis also applies for shape (n).

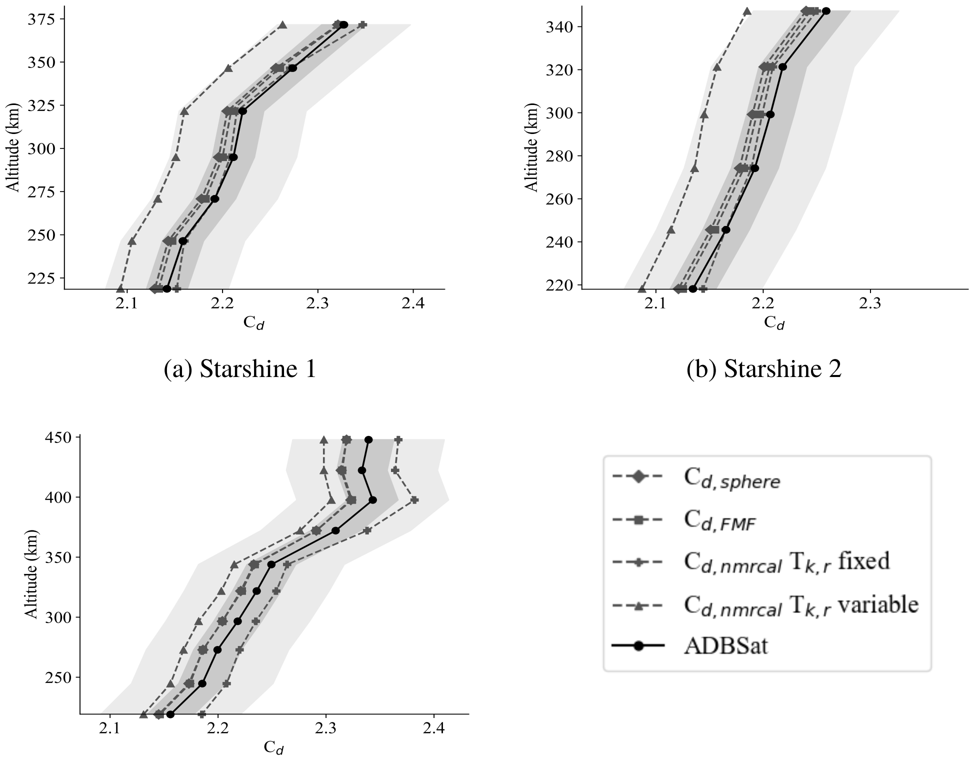
In summary, multiple reflections of particles between the surfaces of the body do not lead to a large inaccuracy, for relatively shallow features. However, ADBSat is unsuitable for satellite shapes that include deep features where particle trapping could occur, such as intakes.

4.3.4 Starshine satellites

Pilinski, Argrow, and Palo [138] used the *DS3V* DSMC code for their drag analysis of the Starshine satellites. As we were able to obtain the CAD geometry files they used, we are confident of the validity of this comparison.

Their results for the variation of C_d with altitude, for all three satellites, are shown in fig. 4.12 alongside our own. They employed multiple methods of calculating C_d:

1. $C_{d,sphere}$: closed-form equations for the drag on a perfect sphere



(a) Starshine 1

(b) Starshine 2

(c) Starshine 3

Figure 4.12. Comparison between C_d output by ADBSat and that presented by Pilinski, Argrow, and Palo [138], for the three Starshine satellites across a range of altitudes. Error ranges of 1% and 3% on the ADBSat results are highlighted.

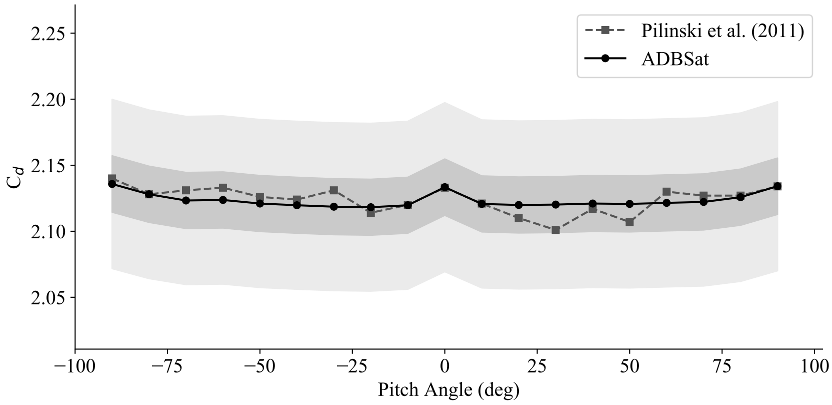


Figure 4.13. C_d at each pitch angle for Starshines 1 and 2, as calculated by Pilinski, Argrow, and Palo [138] and ADBSat. Note the fine scale of the y-axis. The gray bands indicate error ranges of 1% and 3% on ADBSat.

2. $C_{d,FMF}$: a panel method which computes the drag of each mesh element without considerations of shadowing or multiple reflection

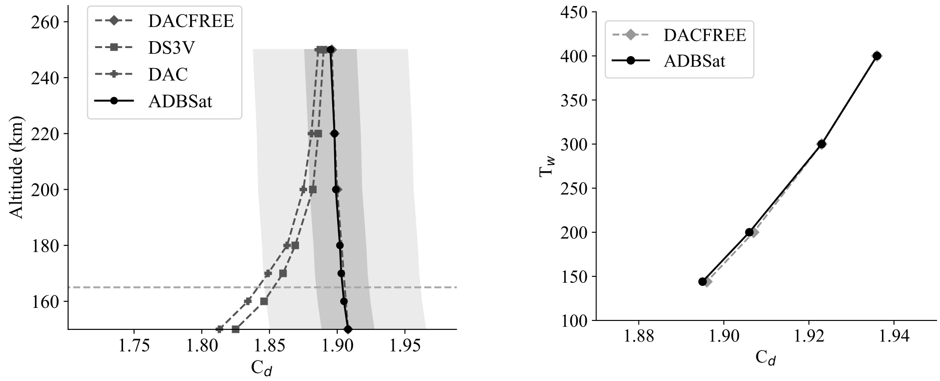
3. $C_{d,nmrcal}$ $T_{k,r}$ $fixed$: a test-particle Monte Carlo (TPMC) method with single-impact accommodation

4. $C_{d,nmrcal}$ $T_{k,r}$ $variable$: the same TPMC with multiple reflections

ADBSat agrees well with the reported values of $C_{d,sphere}$, $C_{d,FMF}$ and $C_{d,nmrcal}$ $T_{k,r}$ $fixed$. An error of 1% (highlighted in dark gray) comfortably encompasses their data. Setting the error to 3% (highlighted in light gray) also covers the more realistic case of multiple reflections.

Furthermore, we also verified our results against their aerodynamic analysis of Starshines 1/2 as a function of pitch angle, shown in fig. 4.13. ADBSat shows a similar trend, with slightly less fluctuation, than the DSMC calculations. All values are within 1%, indicating good agreement between the two methods of aerodynamic analysis.

4.3.5 Orion capsule



(a) Comparison across altitudes

(b) Comparison across T_w

Figure 4.14. Comparison between ADBSat and Moss, Boyles, and Greene [98] of drag analysis for the Orion capsule. The 1% and 3% error intervals on the ADBSat outputs are highlighted in fig. 4.14a.

The detailed description of atmospheric and geometric parameters provided by Moss, Boyles, and Greene [98] allowed us to accurately reproduce their simulations. The authors present results obtained with three different codes, two DSMC codes (*DAC* and *DS3V*) and a free-molecular code whose algorithm is not known (*DACFREE*). Their values of C_d at relevant altitudes are reproduced in fig. 4.14, alongside results from ADBSat. The dark and light grey highlighted areas are the 1% and 3% intervals on ADBSat values, respectively.

ADBSat agrees closely with *DACFREE* for all simulations, even those not in FMF. At approximately 170 km and above, where FMF occurs according to the K_n reported by Moss, Boyles, and Greene [98], it also agrees to within 1-3% with both DSMC codes. A better agreement is noted at higher altitudes, where the flow is strictly FMF. To summarize, our results show a good agreement to the C_d presented, where the assumption of FMF is valid.

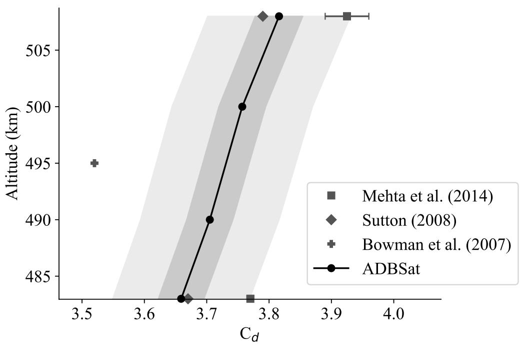


Figure 4.15. ADBSat vs. literature sources for the drag coefficient of the GRACE satellite. 1% and 3% regions on ADBSat outputs are highlighted.

4.3.6 GRACE

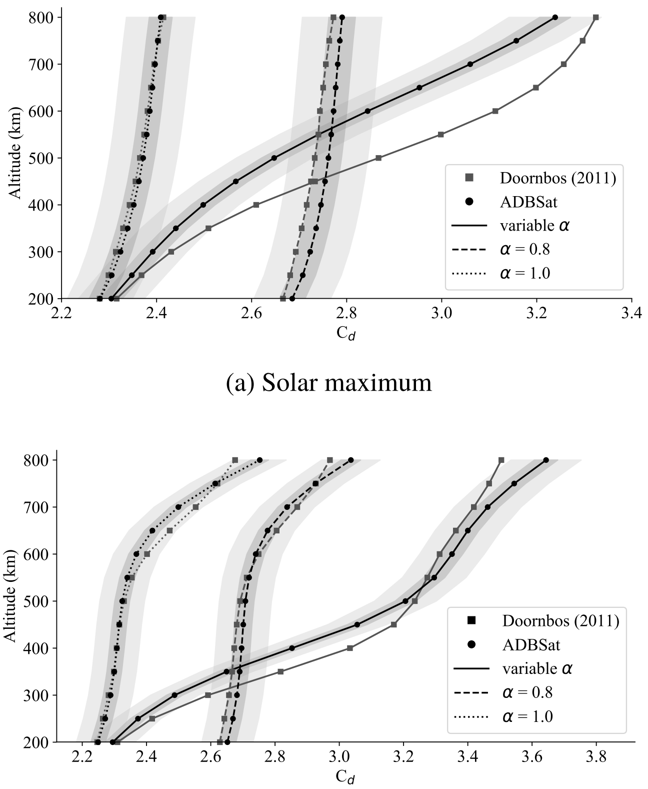
Mehta, Walker, Lawrence, *et al.* [102] report the variation in drag coefficient of GRACE over 24 h, both from their own data and from previous sources. The satellite's orbit over this time will vary from 483 km to 508 km continually. For comparison purposes, we assumed that the maximum and minimum drag coefficients of Mehta, Walker, Lawrence, *et al.* [102] and Sutton [151] occur at the apogee and perigee respectively. The average ballistic coefficient found by Bowman, Marcos, Moe, *et al.* [152] was assumed to be at the midpoint of the altitude range, 495 km.

ADBSat calculated values of C_d closest to those reported by Sutton [151], as shown in fig. 4.15. The agreement is comfortably within 1%. The values reported by Mehta, Walker, Lawrence, *et al.* [102] are within a wider margin of 3% of our results. The average reported by Bowman, Marcos, Moe, *et al.* [152] is lower than our value, most likely due to similar reasons as those reported in Mehta, Walker, Lawrence, *et al.* [102], such as their use of a constant accommodation coefficient as opposed to to our altitude-dependent analysis.

4.3.7 CHAMP

The analysis of CHAMP was complicated by the geometry available being slightly different to that which was used by Doornbos [80]. Comparison of the projected areas showed that the models were much more closely matched when viewed side-on, while the head-on projections did not align as closely. As the value of C_d output by ADBSat relies heavily on the shape and its projected area, we could not directly compare our aerodynamic analysis in head-on flow conditions with that reported.

Doornbos [80] presents C_d for a number of scenarios where the satellite is side-on to the flow. We tested the performance of ADBSat with a varying accommodation coefficient calculated from the Langmuir isotherm model [122], and fixed values of $\alpha = 1$ and



(a) Solar maximum

(b) Solar minimum

Figure 4.16. Drag coefficients of CHAMP at a side-on view. 1% and 3% error ranges on ADBSat are highlighted. The variable α is calculated using the Langmuir isotherm model.

$\alpha = 0.8$. We examined both maximum and minimum solar activity levels, presented in figs. 4.16a and 4.16b respectively.

For the two cases of constant α, some results agree as closely as 1% to the published values, with all being consistently within 3%. When we use a height-dependent α, with all other parameters the same, a larger discrepancy is seen. However, it is of note that we calculated α independently, in an effort to reproduce the values used by Doornbos [80] - we do not know the values of α used therein. As C_d is particularly sensitive to variations in α, our conclusion is that the discrepancy is most likely due to this factor.

4.4 Conclusions

By comparison to both DSMC and established literature sources, we have analysed the accuracy and limitations of ADBSat, a novel program that calculates the aerodynamics of a satellite body. While more simplistic than DSMC, the reduced computational and time cost of ADBSat is advantageous for some aspects of mission design. As an increasing number of satellite missions are developed to operate in VLEO, its efficiency will allow aerodynamic considerations to be employed earlier in the mission design process, and to explore a wider variety of designs. It can be used to quickly find promising satellite geometries that can then be analysed more thoroughly. It allows engineers lacking the extensive expertise required for DSMC simulations to obtain an approximate description of the satellite aerodynamics. Additionally, it can be used in a complementary manner to other aerodynamic analysis methods to obtain a thorough description of the aerodynamic characteristics of the test shape.

The comparison to DSMC, implemented through dsmcFoam, involved examination of specially designed test objects at a range of atmospheric conditions across VLEO altitudes. These objects included basic shapes, shapes that tested the new shading algorithm, and shapes that promoted multiple reflections of the atmospheric particles between the body panels. Analysis of the basic shapes showed a good agreement to both DSMC and closed-form models for strict FMF, where $K_n > 10$. The shading algorithm shows the desired performance at most incidence angles with respect to the flow. It breaks down if a large number of body panels are parallel to each other and to the flow, due to MATLAB's inherent floating point precision. Despite disregarding multiple particle reflections, ADBSat is shown to be accurate for some detailed shapes, but not for those where deep concavities (such as atmospheric intakes) are present on forward-facing sides of the body.

Comparison to existing literature detailing the drag analysis of real satellite shapes reveals that ADBSat differs by up to 3% from the reported values. Therefore, it is recommended that an error interval of 3% is adopted in the future for all ADBSat results. However, larger errors are seen when examining geometries that are unsuited to panel methods such as those with deep concavities in which particle trapping and multiple particle reflections can occur. It is recommended that alternative methods are used for the analysis of such cases.

Funding Sources

The DISCOVERER project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 737183. Disclaimer: This publication reflects only the views of the authors. The European Commission is not liable for any use that may be made of the information contained therein.

Acknowledgments

The authors would like to thank Marcin Pilinski and Eelco Doornbos for providing CAD satellite geometries to facilitate comparison to their work. We would also like to thank all members of the DISCOVERER project for their input. L. Sinpetru would like to thank her supervisors and colleagues for their useful comments.

Chapter 5

Optimisation Methods

We have thus established that ADBSat can obtain a quick, accurate approximation of the C_d of a body. A higher C_d means more drag force, which leads to shorter satellite lifetimes and the need for orbit-keeping solutions such as on-board thrusters. As these solutions can be expensive both in terms of monetary cost and mass budget, it is advantageous to reduce the drag force on a particular satellite configuration as much as possible. One of the factors significantly affecting the magnitude of drag experienced is the external spacecraft geometry. The spacecraft shape and size is key to the aerodynamic qualities of the design [101]. Thus, an efficient way to minimise drag is to construct the spacecraft hull with this specific purpose in mind, in other words to optimise the hull geometry for reduced drag.

Although the optimisation of other satellite mission design problems is common [153]–[162], literature regarding optimisation of the outer satellite geometry is rare due to the difficulty of accurate aerodynamic analysis in VLEO. Where it is seen, it is often not in the context of the aerodynamic characteristics, but of other concerns such as structural integrity, mass, or monetary cost [163], [164]. Nonetheless, some work has investigated spacecraft drag reduction.

Park, Myong, Kim, *et al.* [54] wrote and validated an in-house DSMC code, which was applied to analyse the aerodynamic characteristics of a test spacecraft with the objective of drag reduction. Their spacecraft body, chosen for its similarity to other spacecraft in VLEO, was in the shape of a rectangular parallelipiped with a cylindrical front. A wedge was added to the front of the spacecraft, pointed in the direction of flight, with its length being their design variable. For specular reflection, they noted a significant reduction in drag with an increase in wedge length. For diffuse reflection, the drag was unaffected by the wedge length.

Walsh, Berthoud, and Allen [62], building on their previous work [61], used a surrogate model based on DSMC in conjunction with computational optimisation to both minimise drag and maximise the internal volume of a two-dimensional model representing a three-dimensional axisymmetric spacecraft. They considered the influence of tapering the nose and tail on smooth, cuboid spacecraft bodies. A significant reduction in drag was noted for their primarily diffuse reflection model.

Yu and Fan [165] analysed the drag of slender satellites using the panel method, in order to identify shapes with the lowest drag and best volume to drag ratio. They used four different slender shapes for the spacecraft, and also separately investigated four configurations of the face pointing into the flow (i.e. the head). They reported an increase in drag coefficient with angle of attack and molecular speed ratio. An optimum length to width ratio was found for minimum drag, as a function of the windward and side-on drag coefficients. They also identified the head configuration with the best aerodynamic performance of their four choices.

Hild, Traub, Pfeiffer, *et al.* [166] apply numerical optimisation to a 2D representation of a spacecraft profile, with the aim of increasing operational lifetime. Starting from a simplified cylindrical profile similar to the GOCE spacecraft [25], they split the 2D profile into constituent parts and apply the Sentman model. Constraining length, mass, volume and wall temperature maintains a realistic payload capability. Considering only convex shapes preserves the accuracy of the model. The front surface of the shape is optimised, while the tail is unaffected, but can be a greater or smaller part of the body. Following the optimisation, the shape is converted to a 3D model through one of three methods, the choice of which is also an optimisation parameter. For diffuse reflection, currently the most realistic case with available satellite materials, they show a maximum increase of 46.3% in mission lifetime. Specular reflection leads to an increase of 3300% in mission lifetime.

One characteristic of much the previous work is a lack of continuity between satellite configurations. Most often, a set of discrete shapes are investigated in succession. This approach can be summarised as thorough characterisation of the aerodynamics of a small set of design options. While a detailed knowledge of the performance of each option is obtained, only a limited part of the large design space spanned by the problem is investigated. In the early stages of aerodynamic spacecraft design, it would be beneficial to consider a larger part of the design space, preferably in a continuous manner. It would also be constructive to be able to tailor the designs to other requirements of the mission in terms of factors such as spacecraft volume, control requirements, and constraints on the body shape. Such constraints could arise from conditions imposed by the launch vehicle, for example.

Previous work has also often approached the problem by simplifying 3D satellite shapes to 2D. While reducing dimensions is an established method in simplifying optimisation problems, it also leads to major restrictions on the resultant body, in particular that of rotational symmetry around the direction of motion. It would be beneficial to have more control over the spacecraft shape, in the case where symmetry along one of the three principal axes is not the desirable condition. The results of a full 3D optimisation may exhibit characteristics which are impossible to pinpoint in a 2D simplification.

One way to explore a larger part of the available design space and exert more control over the optimisation shape is by applying computational optimisation. Effectively, this means translating the problem into an optimisation framework made of a number of design parameters, constraints, and objectives. A computer then attempts to find a solution to the problem by applying an optimisation algorithm, with the best design(s) then passed to the user for further examination and refinement. Such an optimisation framework would require a large number of calls to the aerodynamic analysis program. As previously described, panel methods are a good way of quickly determining the aerodynamics of a body, and their fast runtime compared to numerical methods means that no interpolation is necessary.

In this and the following chapters, we examine the suitability of computational optimisation for the problem of aerodynamic analysis in VLEO. We then devise an optimisation workflow which aims to improve on satellite body characteristics using computational optimisation in conjunction with ADBSat. We integrate constraints and additional objectives into the workflow in order to ensure that the outcome of the optimisation is not only low in drag, but also feasible in a real mission design scenario. The speed and accuracy exhibited by ADBSat make it a prime candidate for integration into an optimisation framework. Its shortcomings are well-known and can be mitigated through constraints. Some challenges of integration are immediately obvious. Firstly, the shape is represented by three 2-dimensional matrices which fully describe the triangular panels of the model. This is incompatible with the single, one-dimensional vector required for optimisation. Secondly, to maintain a short computational time, it is crucial to be able to optimise subsections of a shape without affecting other parts of the body. ADBSat has no capability of selecting or otherwise sorting vertices. Finally, ADBSat does not incorporate mesh checks or other suitability tests, but assumes it has been provided with a watertight, good quality mesh. Therefore, checks will need to be implemented to maintain mesh quality throughout the optimisation routine. However, these challenges are all surmountable, and its good qualities lend themselves well to its integration into an optimisation framework within MATLAB.

5.1 General Overview of Optimisation Problems

An optimisation problem normally involves finding the best solution to one or more objective functions, subject to equality and inequality constraints. Such objective problems can be represented as follows:

$$\begin{array}{rl} \text{minimise} & f(\vec{x}) \\ \text{subject to} & c_i^{eq}(\vec{x}) = A_i \\ & c_j^{ineq}(\vec{x}) \geq B_j \\ & x_{k,LB} \leq x_k \leq x_{k,UB} \; (k = 1, \ldots, n) \end{array}$$

Here, $f(\vec{x})$ is the objective function. $\vec{x}$ represents an input to the objective function, in other words the candidate solution. This may take a number of forms, but in realistic problems is normally a vector, $\vec{x} = (x_1, x_2, \dots, x_n)$. The limits on each element are dictated by the lower and upper bounds, $x_{k,LB}$ and $x_{k,UB}$ respectively. Candidate solutions are distinct and exist in a domain called the search space. The topology of the search space is normally highly complex, with each candidate being assigned a fitness value, a measure of how well it satisfies the objectives. Each candidate is normally also assigned a set of neighbours, the extent of which is defined as its neighbourhood. An algorithm attempts to locate the optimum solution(s) of the problem in the entire search space.

The fitness of the candidate solutions is evaluated by the objective function. In the most general case, there are no inherent requirements on the objective function, such as continuity, differentiability or twice-differentiability. This function is problem-dependent and defined by the user. When a problem is described by a high number of dimensions, it may be difficult to elucidate, and particularly to visualise, the shape of the objective function. Any complicating features such as discontinuities or undefined regions make it particularly difficult to optimise the problem, while a smooth objective function generally lends itself well to successful optimisation. Objective functions based on realistic problems are, in general, computationally expensive, non-smooth, high-dimensional, and incapable of being solved analytically. Additionally, most realistic problems require the use of multiple objectives in order to fully capture the behaviour of the system in the context of its limitations. Considering multiple objectives simultaneously affords many benefits, chief among them the ability to examine the trade-off in objectives among different solutions, which cannot be captured through the use of constraints. However, it introduces a high level of complexity into the system.

By convention, objective functions are normally minimised. It is trivial to convert a maximisation problem to minimisation: maximising any objective function $f(\vec{x})$ is equivalent to minimising its negative, $-f(\vec{x})$. However, differentiating between local optima, or candidate solutions which are the best in their neighbourhoods, and global optima is non-trivial. Convex optimisation problems have a convex objective function and a convex set of solutions, and by definition all local solutions are also global solutions. However, the complexity of most real-world objective functions means that they will be non-convex and usually have multiple local optima which do not coincide with global optima. Such solutions pose problems for some optimisation algorithms, which can become trapped in local optima and erroneously report them to be the global optimum solution.

The constraints are categorised as i equality constraints, c_i^{eq}, and j inequality constraints, c_j^{ineq}. Equality constraints, requiring one satisfactory value, are stricter and generally more difficult to satisfy than inequality constraints, which accept a range of solutions. Such constraints can be linear or nonlinear. Nonlinear constraints, as are found in most real-life optimisation problems, significantly increase the difficulty of the problem compared to linear constraints. Maintaining the feasibility of solutions is especially difficult when dealing with non-linear constraints.

Broadly, there are two main methods of exploration: gradient-based methods and meta-heuristics.

5.2 Gradient-based methods

Gradient-based methods compute the search direction by obtaining the gradient of the objective function at the candidate solution, and searching in a downhill direction, a function of the negative of the gradient. The gradient vector is defined by eq. (5.1). By following the gradient in this manner, the algorithm hopes to reach a point at which movements in any direction which would not violate the constraints would worsen the fitness function.

$$\nabla_x f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \dots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \tag{5.1}$$

The step size, also called learning rate, is specified by the user. It primarily determines the efficiency of the algorithm. Too large a learning rate, and it may overshoot the target and bounce across the ”sides” of the valley where the optimum is located, thus failing to converge. Too small and the algorithm will take a long time to perform the large number of necessary steps to reach the optimum. Mitigation strategies are available to overcome this difficulty, including changing the learning rate as the optimisation progresses or setting individual learning rates for different optimisation parameters [167].

There are many variations of gradient descent, some of which show benefits such as computational efficiency or stable convergence, particularly for convex search spaces where only one optimum exists. Their strengths make them particularly popular in the machine learning community, and they have been labelled “...by far the most common way to optimise neural networks.” [167]. However, in a non-convex search space when multiple local minima are present, the algorithm will tend to become trapped in a local minimum rather than finding the global optimum. It has also been argued that such problems are not caused by local minima but, in fact, saddle points, which can appear to be local minima [168]. Regardless of the cause, this susceptibility to entrapment in local minima means that for realistic problems, where the aforementioned objective function is most likely non-smooth, non-convex, discontinuous and highly complex, gradient methods are at a disadvantage. They are inherently difficult to apply when discontinuities are present and functions are non-differentiable. Gradient descent can also be slow when dealing with highly complex problems as realistic ones often are, due to the need to compute the gradient, which is computationally expensive. Although mitigation strategies exist to overcome some of the challenges mentioned, these properties render gradient descent unsuitable as an optimisation method for the realistic problem of aerodynamic optimisation.

5.3 Metaheuristics

In contrast, metaheuristics do not require any gradient information. They employ a search strategy, often inspired by natural processes [169], to efficiently find an optimal or near-optimal solution to a problem. They are also often stochastic, using randomness to facilitate more efficient examination of the search space [170]. Metaheuristics do not guarantee convergence to optimality, or even that they will find a suitable solution. Instead, they attempt to improve the solution to a problem within the bounds of the optimisation parameters, and it is assumed that if those parameters are appropriate to the problem, some of the solutions provided will be near-optimal [171].

Flexibility and freedom in the formulation of the problem are two known strengths of metaheuristics. Unlike gradient-based algorithms, the search space need not be well-behaved, making them particularly suited to tackling complex real-life problems. However, this means that there is no one universally applicable method which is sure to succeed. Wolpert and Macready [172] postulate the No Free Lunch Theorem, stating that no one search method is superior over any other on average - including basic random search. They highlight the necessity of tailoring a solution scheme to the problem in question. This tuning of optimisation parameters can be troublesome [173]–[175], however it is necessary in order to validate the assumption that an algorithm will reach near-optimal solutions.

While a universal classification scheme for metaheuristics has thus far not been established, one common way to separate them is to split them into individual-based algorithms and population-based algorithms [176]–[178]. Individual-based algorithms such as simulated annealing begin with one possible solution, and use an optimisation methodology to iteratively improve that individual. Conversely, a large number of individuals make up the starting point of population-based algorithms, and a solution is found by considering the characteristics of the population as a whole. Due to the nature of aerodynamic satellite design, it is assumed that in the early stages of the design process, there will be many exploratory designs, each of which presents some benefits and drawbacks. Balancing these aspects will be key in arriving at a final optimised design. Thus, population-based algorithms are better suited to the formulation of aerodynamic analysis of satellites in VLEO. There are many options for such optimisation algorithms, with some controversy over the recent explosion in the number of optimisation strategies based on natural metaphors [169], [173], [179]–[183]. A summary of the most widely-used population-based metaheuristics is given below.

5.3.1 Ant Colony Optimisation

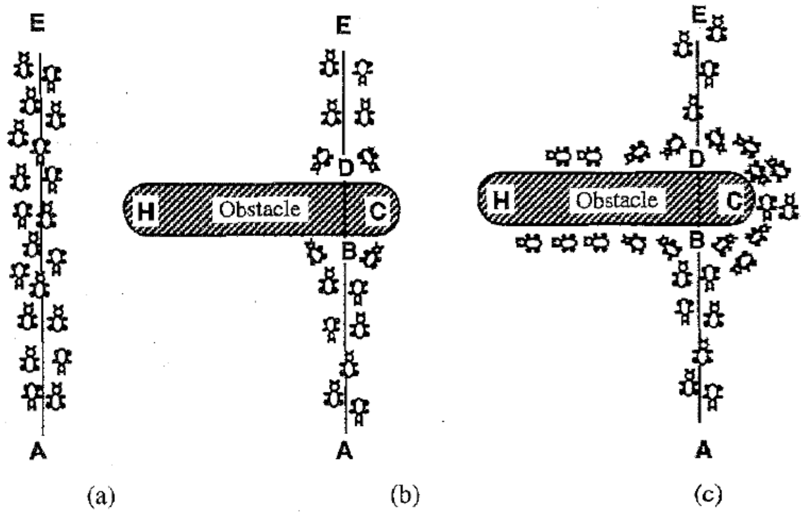


Figure 5.1. An example of the behaviour of an ant colony on which ant colony optimisation is based. When faced with an obstacle with two paths around it, ants which have travelled a path use pheromones to tell other colony members about it. Thus, more colony members will follow the better path. Image reproduced from Dorigo, Maniezzo, and Colorni [184].

First postulated by Dorigo [185], this nature-inspired metaheuristic seeks to harness the behaviour of an ant colony finding the most suitable path between their home and a source of food. It is based on the ant colony's use of pheromones to communicate the path to and quality of solutions between members, with ants being more likely to follow paths that have higher pheromone concentrations. Each ant, or candidate solution, has a memory of its previous states and their order. Determining the next state is dependent on existing pheromone trails from other ants, and also takes other factors into account such as the relative distance of other states from the one it currently inhabits [184]. The pheromone characteristics, such as when to deposit them, how strongly, and how quickly they evaporate, depend on problem-specific parameters [186], [187]. The pheromone evaporation rate is a particularly important parameter of the computational model, determining the amount of time for which a pheromone trail will influence other candidates. It promotes a fair evaluation by the ants of all pheromone trails, as they all exist for an equal amount of time. Thus, it prevents preferential premature convergence to the initial pheromone trails laid down [188].

Ant colony optimisation presents numerous benefits. It is inherently suited to parallelisation and can pinpoint the best solutions in a set relatively quickly [189]. The sequential manner in which it builds solutions is also advantageous for some industries [190]. However, is particularly suited to combinatorial problems, in particular those which require path-finding such as vehicle routing or the well-known travelling salesman problem. Although modifications and improvements have allowed for the successful application of this algorithm to a variety of problems [186], it still suffers from problems such a fluctuating convergence speed and algorithm stagnation [191]. As the aerodynamic optimisation problem is not sequential and does not belong to the group of problems for which it is particularly suited, it is more likely to suffer from these drawbacks.

5.3.2 Harmony Search

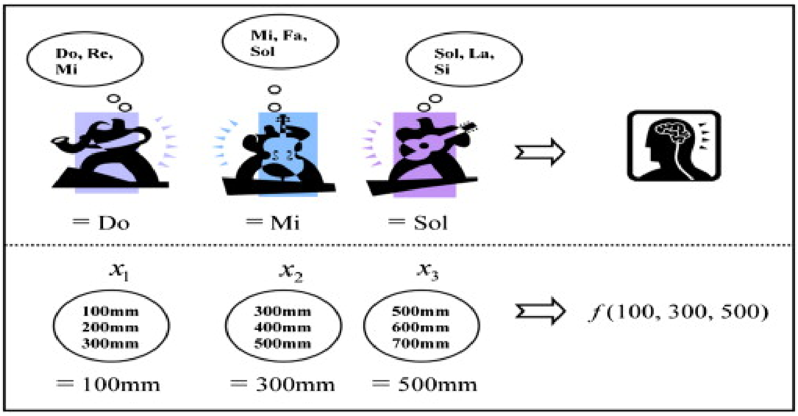


Figure 5.2. A depiction of the equivalence between musicians playing music and the harmony search algorithm finding a solution. Reproduced from Kulluk, Ozbakir, and Baykasoglu [192].

The harmony search metaheuristic attempts to mimic the process through which musicians improvise music together. As musicians rely on practice to tune the notes they play and ultimately achieve an aurally harmonious melody, so does the algorithm attempt to tune the decision variables over many iterations to achieve the optimal solution to the problem [193]. Invented by Geem, Kim, and Loganathan [194] with the purpose of achieving better solutions in fewer iterations than other metaheuristics of the time, it relies on a ”harmony memory” to construct new candidate solutions. The harmony memory is essentially a group of candidate solutions, which are combined to obtain the new solution. If the created solution is better than that of minimum fitness in the existing set, it replaces the minimum fitness solution in the set. There are some available features which attempt to mitigate the problems of stagnation, loss of diversity and trapping in local minima. The algorithm employs a harmony memory considering rate to dictate the balance between choosing values of decision variables already in the solution set, and creating new values from the entire possible range. In this manner, the algorithm aims to explore decision variables outside the starting set. A pitch adjustment rate can introduce a local fluctuation, in changing the value of a decision variable to a near neighbour with a low probability, in a similar way to that of musicians moving a melody up and down a semitone to better suit the group [194]–[196].

Harmony search has many benefits including the relatively low number of control parameters [195], quick convergence [197], and a good balance between exploration and exploitation [198]. However, it does suffer from premature convergence, and is mainly aimed at solving problems which involve discrete variables [199]. This decreases its suitability for problems of a realistic nature, which often involve continuous variables. Another of the key benefits of harmony search, that it does not require the user to specify a starting population of solutions [195], becomes a disadvantage in high-dimensional aerodynamic optimisation. Where there exist a high number of control points on the shape which must be altered to achieve a more advantageous aerodynamic configuration, not specifying an initial population means that the algorithm would randomly try to create the satellite shapes. This would lead to chaotic and unrealistic shapes. Th type of optimisation problem at hand requires engineering expertise in specifying the initial plausible designs of the satellite. Thus, harmony search would be unsuitable for aerodynamic drag optimisation.

5.3.3 Artificial Bee Colony

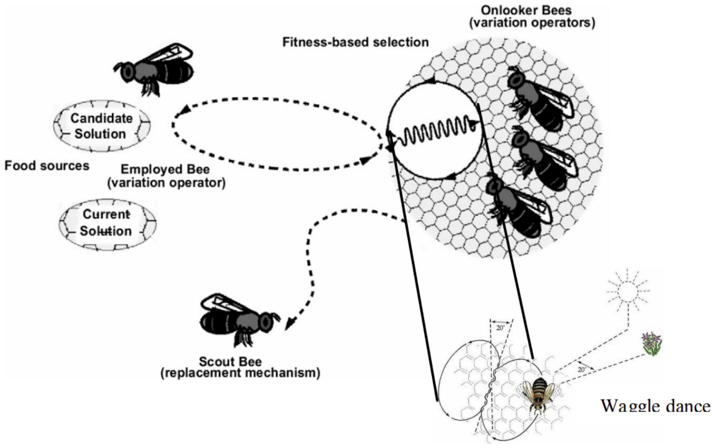


Figure 5.3. An overview of the artificial bee colony algorithm, including the bio-inspired ”waggle dance” performed by bees to relate the information gathered back to the hive. Reproduced from Sharma, Pant, and Singh [200].

The artificial bee colony algorithm, as proposed by Karaboga [201], is based on the structure of a honey bee hive and their foraging activities. Exhibiting some similarity to ant colonies, honey bee hives employ pheromones as well as a ”waggle dance” to tell other colony members about the location and quality of flowers [202]. The algorithm aims to explore the most possible solutions by using three types of exploratory agents: the employed bee, the onlooker bee, and the scout. The employed bee is essentially an assigned solution with a fitness value, using its memory to share the information about that particular location in the search space with the other two types of bees. The onlooker bee collects information from many employed bees, and chooses the best suitable solution of its dataset. The employed bee, having exhausted the information it can gather from the candidate solution, turns into a scout bee whose responsibility is sourcing new candidate solutions. Together, all bees work to move towards the most profitable solution in the search space [203]–[205].

The artificial bee colony is particularly good for discrete problems such as routing, where it has been found to efficiently search routes above a suitable minimum threshold fitness value [206]. Like harmony search, it has few control parameters and can converge quickly [207], [208]. However, it has been observed to be better at exploration rather than exploitation - in other words, it can easily find new candidate solutions, but fails to capitalise on all the information available from the existing solutions [204]. This is important in the aerodynamic optimisation problem, as the fitness function will be complex, and exploitation of fewer fitness function calls is desirable over many calls to evaluate new solutions. Additionally, similarly to harmony search, it generates its initial population randomly [207] which would be disadvantageous for aerodynamic optimisation. Therefore, this algorithm is also unsuitable for the problem at hand.

5.3.4 Particle Swarm Optimisation

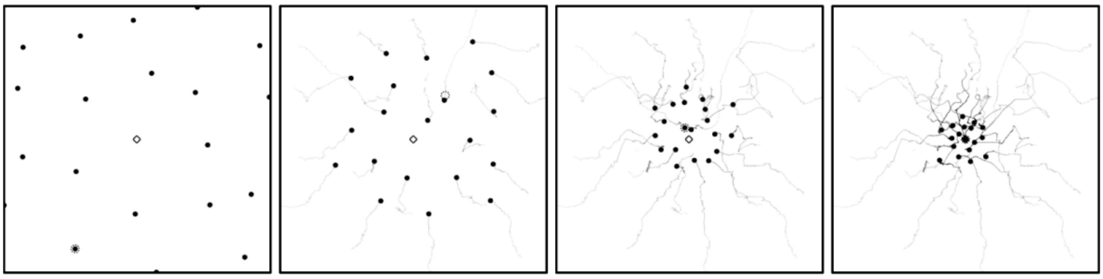


Figure 5.4. A visualisation of the progression of the particle swarm optimisation algorithm in time, viewed from left to right as time passes. Filled circles are candidate solutions, the circle surrounded by dots is the best solution so far, and the diamond is the optimum solution. Reproduced from Kim, Lee, and Yoon [209].

Particle swarm optimisation is modelled on the social behaviour of flocks of birds, as developed by Kennedy and Eberhart [210] and Shi and Eberhart [211]. Each individual in the population, or particle in the swarm, is represented by its position, velocity, and personal best position up to that point in the simulation. The vectors representing a particle are updated at each generation, until a stopping condition is reached. Surrounding particles contribute to the velocity vector of an individual through their personal best positions. There is also an element of randomness in the velocities. This behaviour mirrors the manner in which birds flying together will change their direction and velocity according to their neighbours and the behaviour of the flock as a whole, while still maintaining personal individuality and variety. The strategy for choosing the neighbouring particles which contribute to the velocity vector is referred to as the topology of the swarm, and many different options have been suggested over the years. Three parameters called the inertia weight, cognitive weight, and social weight determine how much influence each contributing factor has on the updated velocity [210]–[212].

The combination between randomness and neighbour cooperation aims to balance exploration of the search space and exploitation of the available local information. The relatively simple concept and implementation also facilitates straightforward parameter control, a notable advantage given the aforementioned importance of parameter tuning [213]. It also exhibits good computational efficiency when contrasted with other comparable optimisation methods [214]. However, the inertia weight must be chosen carefully to avoid a phenomenon known as swarm explosion, where the velocities of all the particles in the swarm grow to infinity. Particle stagnation can be a problem, resulting in non-optimum convergence. It has also been shown that simple particle swarm optimisation is sensitive to transformations of the search space such as rotation, which is a significant problem as the recommended solutions to this issue are inefficient for large-scale problems [212]. This decreases its suitability in applications involving high-dimensional problems such as aerodynamic optimisation.

5.3.5 Genetic Algorithm (GA)

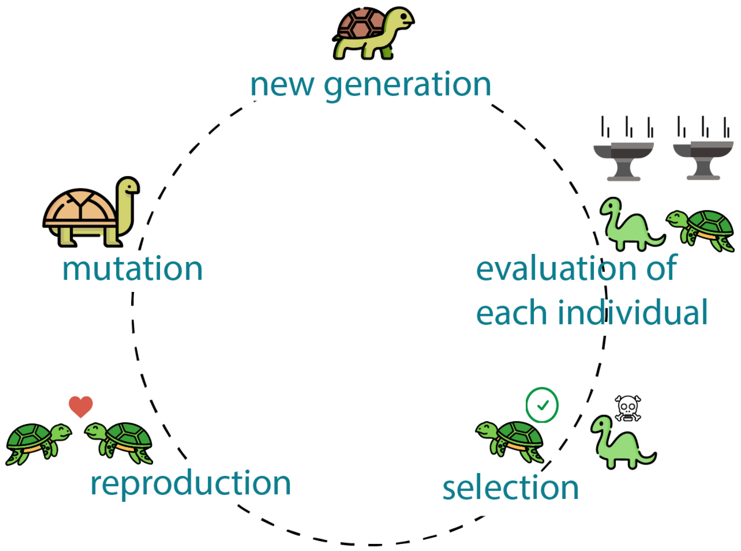


Figure 5.5. A simple overview of the steps of the genetic algorithm, in the context of biological evolution. Image reproduced from Autodesk [215].

Natural selection is the inspiration for this optimisation algorithm, aiming to harness the mechanism of gene transformation in biological reproduction. Arguably first postulated by Alan Turing in the 1950s in his quest to design an intelligent, evolving machine [216], some attempts were made in the 1960s to more closely simulate the biological mechanisms behind genetics [217]. The first instance of a modern GA was developed in 1975 by Holland [218] and De Jong [219]. Each individual in the population, called a phenotype, is made of a set of decision variables called a genotype. Each decision variable is a gene. There are three main methods of altering the genotype to attempt to find new, and hopefully better, solutions: mutation, crossover, and elitism [220], [221]. Much of the efficiency of genetic algorithms is dictated by the crossover method and parameters, and their fine-tuning is essential for good algorithm performance [220]. Complementary to the crossover scheme is the selection scheme, which chooses the two parent phenotypes to be combined. As this directly feeds the crossover step, its choice is also significant [222], [223].

Due to the formulation of the GA, it is conducive to parallelisation, thus reducing runtime over other metaheuristics. This is particularly important for the aerodynamic analysis problem which is highly complex and will therefore carry a heavy computational load. The GA is also highly adaptable to multi-objective problems [224], and particularly unlikely to be trapped in local optima [225]. However, if the parameters are not finely tuned, it is susceptible to loss of diversity in the population and premature convergence. This is especially problematic if the starting population is too small. The choice of starting population is a problem in itself, as too large a population is also harmful to algorithm convergence. It can extend computation time significantly, as the fitness of every member of the population must be evaluated at each iteration [225], [226]. However, considering all factors both negative and positive, it seems that GAs are well suited to solving the aerodynamic optimisation problem.

Additionally, a significant factor is that ADBSat is written in MATLAB. The GA has a readily available MATLAB implementation, and therefore can easily be integrated with ADBSat. The algorithm's implementation is well characterised, with abundant information available as to its use and requirements[1]. The MATLAB implementation also offers a GA tailored to solving multi-objective problems[2]. This is important, as for a realistic problem formulation multiple objectives are required to mirror the real-life situation. The GA methodology is well-documented and has been shown to be effective for a range of similar applications [227]. Since the aforementioned No Free Lunch theorem states that no metaheuristic is generally better than any other [172], for the above reasons the GA was chosen as the most suitable algorithm with which to approach the problem.

While MATLAB offers some creation functions that can be used to initialise the population, it it also capable of receiving a starting population as an input from the user. For the aerodynamic optimisation problem, harnessing the knowledge of the engineers responsible for spacecraft design is advantageous. Therefore, the initial population is provided for the optimisation. The aforementioned crossover, selection, mutation and elitism are the four main characteristics of GA which dictate its efficiency.

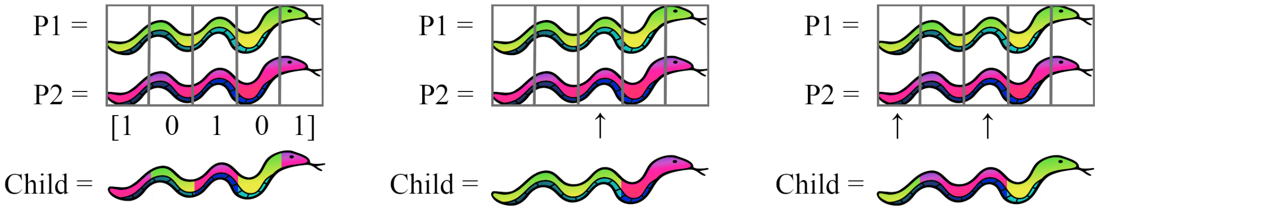
Crossover chooses and combines two parents into an ”offspring”, with many combination methods available [228]. In this way, new solutions, hopefully of a better fitness value, are created from the exchange of information among population members. Without crossover, the fitness of the population would increase only until it reaches that of

[1] Available at `https://uk.mathworks.com/help/gads/ga.html`. Date accessed: 1/03/2022.

[2] Available at `https://uk.mathworks.com/help/gads/gamultiobj.html`. Date accessed: 1/03/2021

the fittest existing candidate solution, but would be unable to improve beyond this. It facilitates both exploration of the search space, and exploitation of the existing solutions [220]. The options for crossover available in MATLAB are:

- **Scattered (Uniform):** A crossover vector is created of the same length as the genotype, whose elements are randomly assigned a value of either 0 or 1. An offspring solution is then created, using elements from the first parent where a 0 is seen in the crossover vector, and elements from the second parent where the crossover vector has a value of 1.
- **Single-point:** A random position in the offspring vector is chosen. The offspring solution is composed of elements from the first parent up to and including that position, and elements from the second parent for any unfilled positions after that point.
- **Two-point:** Similar to single-point crossover, but two positions in the child vector are chosen instead of one. The offspring takes elements from the first parent up to and including the first position, then from the second parent up to and including the second parent, and fills the remainder with the equivalent elements of the first parent.



(a) Scattered crossover (b) Single-point crossover (c) Two-point crossover

Figure 5.6. A visualisation of three different kinds of possible crossover mechanisms.

- **Intermediate:** The offspring is created by a weighted average of the two parents with some random fluctuation. The weights can be controlled either for the genotype as a whole, or for each individual gene, through a ratio vector. Equation (5.2) determines the structure of the child from parents p_1 and p_2, using a random fluctuation $rand$ and the ratio vector [229].

$$\text{child} = p_1 + (\text{rand})(\text{ratio})(p_2 - p_1) \tag{5.2}$$

- **Heuristic:** Here, a line is virtually drawn between the two selected parents. The offspring lies along this line, a small distance from from the better quality parent, in the direction away from the worse quality parent. The ratio vector is used to specify the distance of the child from the parent. Described by eq. (5.3), parent p_1 has the better fitness value compared to parent p_2 [229].

$$\text{child} = p_2 + (\text{ratio})(p_1 - p_2) \tag{5.3}$$

- **Arithmetic:** The two parents are combined through a weighted arithmetic mean to create the offspring. If the weight is dictated by ω_w, then the operation is performed as in eq. (5.4).

$$\text{child} = \omega_w p_1 + (1 - \omega_w) p_2 \tag{5.4}$$

The parent candidate solutions are chosen via selection. In a similar way to natural selection and survival of the fittest, the aim of the selection mechanism is to increase the average fitness of the next generation by promoting good candidates for reproduction and maintaining genetic diversity. Simple selection, such as exclusively selecting the top 50% of solutions to undergo crossover, is detrimental to the performance of the algorithm and quickly leads to stagnation and premature convergence. More sophisticated selection algorithms solve this problem by proportionally selecting candidates for crossover with regard to their fitness value. In this way, poor solutions are still selected with a low probability, maintaining variety among the population [220]. The strength with which the selection mechanism favours better individuals is referred to as the selection pressure.

The only selection mechanism available for the multi-objective GA in MATLAB is tournament selection [229]. This mechanism creates a subset of the population by selecting Λ candidate solutions at random. The solution with the highest fitness value from the subset is chosen to be a parent. As the number of individuals in the subset increases, so does the selection pressure: the worst-performing $\Lambda - 1$ individuals in the population will never be chosen, as there will always be a better solution in the subset. Selection pressure is key to the performance speed and quality of the algorithm [230].

Since selection is based on probability and involves picking candidates randomly, it does not guarantee that the best individuals in the population will be a part of subsequent generations. Therefore, while selection increases the average fitness of the population, there may be a loss of peak fitness across subsequent generations. Elitism aims to mitigate this by including a percentage of the previous population with the highest fitness values in the following generation, unchanged via mutation or crossover, regardless of whether or not they were selected to be a parent. In this way, the best solutions can be guaranteed to be a part of the next population, ensuring that the next population is at least as good as the previous one.

While elitism is based on maintaining individuals across generations, mutation artificially introduces genetic variation by randomly changing genes via a stochastic process.

It helps to prevent stagnation, stave off premature convergence, and escape local optima [231]. There is one mutation function in MATLAB which can handle the type of bounds that a realistic problem requires in order to maintain the physical feasibility of the solution: adaptive mutation. Unlike simpler mutation operators which have a constant probability of altering a gene, the probability of applying adaptive mutation is directly related to the fitness of the candidate solution. Genotypes of low fitness are more likely to be mutated, thus increasing the exploration capability while maintaining those of high fitness. In MATLAB, the mutation is performed by generating directions that take into account the fitness of the previous generation, and randomly choosing a direction and step length along that direction [229], [232].

A detailed flowchart of the GA algorithm including all the components discussed above can be seen in fig. 5.7.

5.4 Multi-objective concerns

As previously mentioned, optimisation problems based on real-life issues will likely consist of a number of objectives, which must all be considered together when solving the problem. In the most general such case, the objective function can be expressed as a function of each of the w objectives: $f(\vec{x}) = (f_1(\vec{x}), \dots, f_w(\vec{x}))$. An optimum solution is defined as one which maximises all objectives simultaneously. However, most realistic optimisation problems will have contrasting objectives, meaning that an optimum solution does not exist. In this case, the goal of the program is to find nondominated solutions, alternatively called Pareto solutions, where no objective can be improved without a decrease in the fitness of another objective. Pareto solutions are equal to or greater than every other solution in fitness [233]. The three main approaches to finding nondominated solutions are scalarising the problem, goal programming, or handling different objectives simultaneously.

Scalarising involves using a priori knowledge of the problem to combine the objective functions into one composite objective function. There are many different ways of performing scalarisation, such as optimising a weighted sum of the objective functions or using utility functions. However, these generally assume linearity of constraints and objective functions, or are particularly suited to linear problems [234]. Furthermore, choosing the combination parameters such as the weights or the utility functions is nontrivial, with solutions fluctuating wildly for small differences in weights [235]. As each such scalarised problem will output a single optimum solution, obtaining a comprehensive set of Pareto solutions is difficult and time-consuming - indeed, a full exploration of the weight space may not reveal all Pareto optimal solutions. This is particularly problematic for a concave Pareto front, as would be found in most complex optimisation problems [236]. Therefore, scalarising the problem at hand is not a viable option.

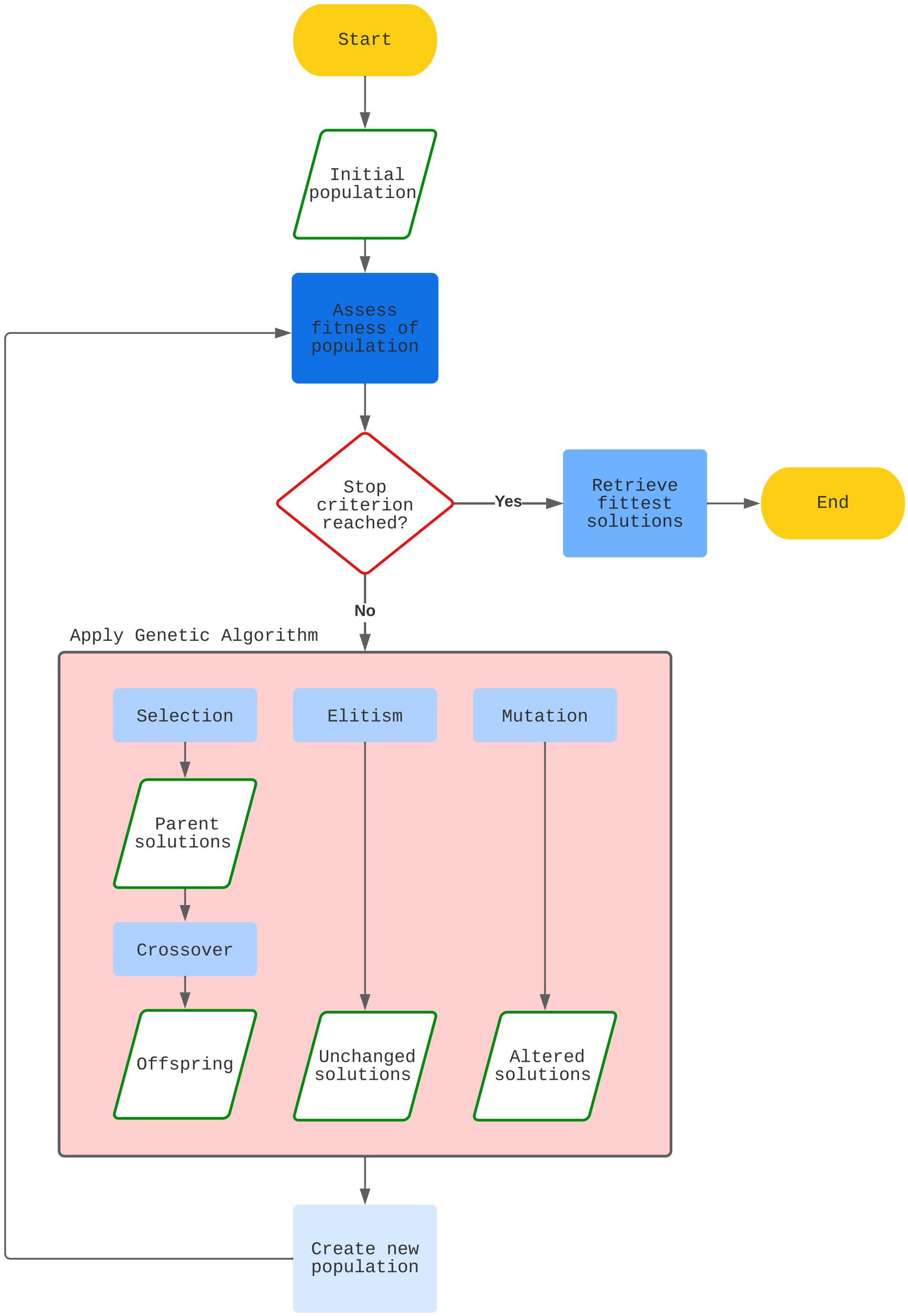


Figure 5.7. A detailed flowchart of the GA algorithm.

Goal programming involves transforming the objectives to target ranges very similar to constraints, and poses some advantages. It is generally easier to obtain information about acceptable ranges than about appropriate weights or utility functions. It is also generally easier to formulate and implement them. However, they provide a more limited control over the problem solutions than objectives. Where objectives will be minimised until the best value has been reached, goals or constraints will be accepted as long as they are within the range specified, without taking into account the relative quality of solutions. In short, the algorithm may not find nondominated solutions [237], [238]. Additionally, algorithms do not attempt to improve on a constraint in the way that they attempt to improve the objective. Instead, they treat the solutions in absolute terms as acceptable or unacceptable, which may lead to a wholly infeasible population and may cause the algorithm to stall. This is particularly problematic when the target range is a single value, much like an equality constraint, which is more stringent than an inequality constraint and therefore harder to satisfy. Consequently, while goals or constraints are an important tool in the realistic formulation of the aerodynamic optimisation problems, they cannot replace objectives completely.

Pareto optimality seeks to consider all objectives and identify all nondominated solutions simultaneously. Multiple Pareto solutions are output with a trade-off of objectives being seen from one solution to the next. This method, while more complex, is particularly suited to real-life optimisation problems as it allows the user to apply their own expertise in considering the different trade-offs independently of the algorithm. One disadvantage is the size of the Pareto set normally scales with the number of objectives, and quickly becomes large enough that it is impossible to consider all the solutions that comprise it. Therefore, most programs will attempt to distribute the chosen output solutions in such a way that as large an area of the Pareto set is considered as possible [220], [235].

GAs are particularly suited to finding the Pareto set of multi-objective problems, not only due to the fact that they are population-based and thus can easily output multiple solutions, but also because of their ability to efficiently search large portions of a discontinuous solution space. They also do not require the user to have a priori knowledge of the relative importance of different objective functions [235]. Extensive literature exists outlining the suitability and previous uses of genetic algorithms in solving multi-objective problems [226], [239]–[242]. For all the reasons outlined above, approaching the realistic multi-objective aerodynamic problem through a multi-objective GA has a high probability of providing good quality solutions via a Pareto set.

The multi-objective GA in MATLAB is based on a controlled, elitist variant of the non-dominated sorted genetic algorithm NSGA-II [243]. First devised by Deb, Pratap, Agarwal, *et al.* [244], this algorithm sorts candidate solutions into successive fronts of increasing rank. Solutions of the same rank are equal in fitness and comprise a trade-off in

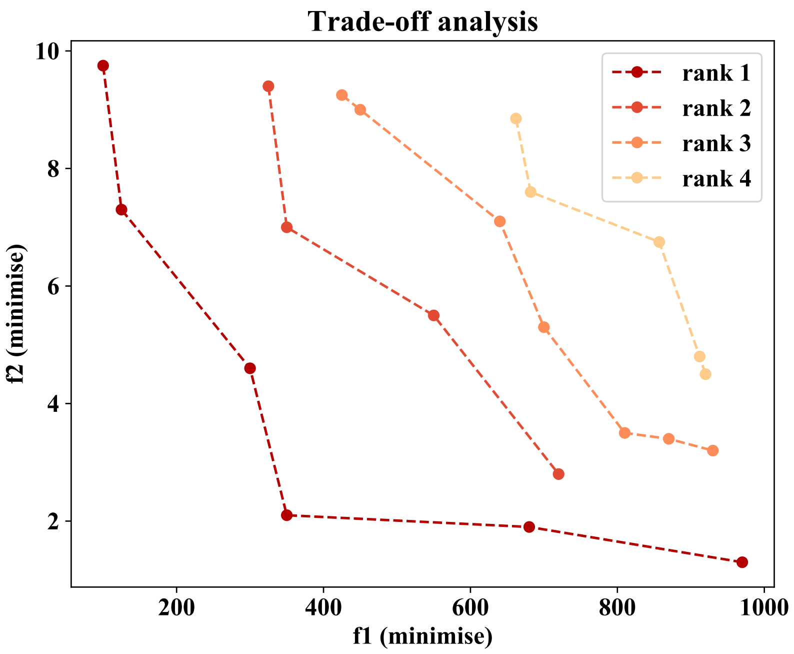


Figure 5.8. A simple dummy data set for the optimisation of two functions, which must both be minimised. An example of the rankings of different fronts by the NSGA-II algorithm can be seen.

various fitness parameters. For $k > 1$, rank k solutions are only dominated by solutions of rank $k - 1$. For $k = 1$, rank 1 solutions are those on the front of highest fitness, and are not dominated by any other solutions in the set. For a converged algorithm, rank 1 is the Pareto front. An example of the ranking system for a simple dummy data set where two functions must be minimised is shown in fig. 5.8.

NSGA-II also calculates the crowding distance of each solution, a measure of the distance of the solution from its nearest neighbours. In order to boost exploration, NSGA-II preferentially selects solutions with a larger crowding distance, in other words those that exist in areas where few solutions are present, for crossover and elitism [245].

The controlled nature of the program refers to the restriction of the number of individuals of rank 1 that can exist in the population. A non-controlled algorithm may stagnate due to the early selection of a large number of rank 1 individuals before convergence and therefore a loss of diversity. By controlling the algorithm and imposing that only N individuals may lie on a front of rank k, genetic diversity is maintained and stagnation prevented [245].

5.5 Summary

The aim of this chapter is to examine the options available for computational optimisation and identify the one most suited to the problem of aerodynamic analysis of satellites in VLEO. Such a problem aims to assume as little as possible about the system and model its behaviour as accurately as possible. Thus, like most realistic optimisation problems, such a problem will be computationally intensive, non-smooth, non-convex, high-dimensional, most likely not differentiable and incapable of being solved analytically. A multi-objective approach will afford the benefit of efficiently examining the trade-off of conflicting factors and their influence on satellite body aerodynamics.

Popular gradient-based optimisation approaches, which base their search direction on the computed gradient of the objective function at each solution, are best suited for convex, continuous, differentiable functions. They are also susceptible to being trapped in local minima and missing the global minimum solution. They can also be slow when dealing with highly complex problems. Thus, they are not well suited to realistic problems such as the one at hand.

Metaheuristics use a search strategy to efficiently search the solution space, which does not rely on gradient. They are more flexible and better suited to problems which exhibit complex features such as non-differentiability and discontinuity, such as the one at hand. They do not guarantee a solution, but aim to search the space in an efficient way, such that the solution found is near-optimal. Many types of metaheuristic exist, many based on nature-inspired algorithms, and they are often divided into those which work on a population, and those which work on a single solution. In order to better examine a collection of possible satellite bodies considered for a particular mission, a population-based metaheuristic is desired

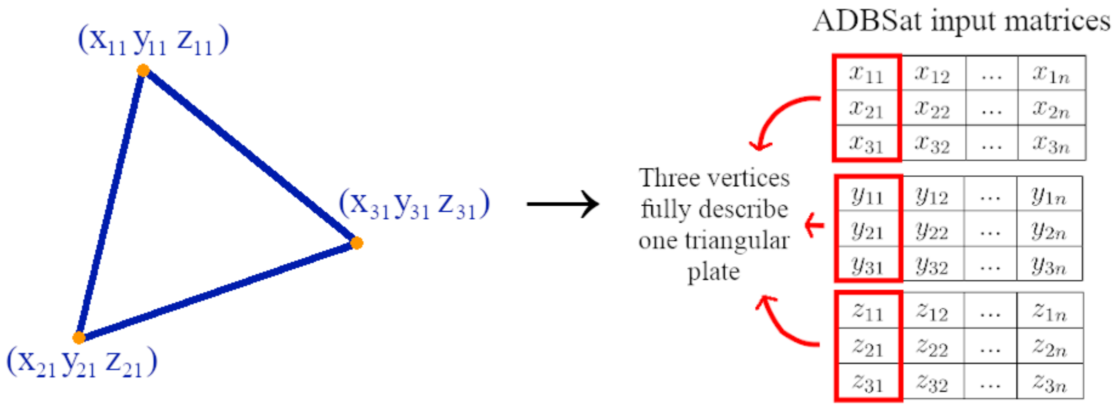
Of the population-based metaheuristics, the GA affords a number of benefits: it is conducive to parallelisation and highly adaptable to multi-objective problems, exhibiting a good exploration of the search space and an adequately low trapping in local minima when parameters are tuned appropriately. It has been shown to perform well for similar problems to that of aerodynamic analysis. It is also implemented in MATLAB, reducing the difficulty of integration with the aerodynamic analysis program, also written in MATLAB. The following chapter describes the process of integration of the two programs, as well as the selected objective functions, constraints, and optimisation parameters.

Chapter 6

Optimisation Framework

Having established the suitability of the GA for computational optimisation of satellite aerodynamics, this chapter aims to depict the optimisation framework that will be employed to find solutions to the problem. First, the geometry parametrisation which allows the shape to transition smoothly between the optimiser and the aerodynamic analysis program is detailed. Five objective functions are fully described including any additions to the program necessary in order to calculate them: drag, volume, relative position of centres of gravity and pressure, and two pointing stability objectives. Furthermore, the chosen constraints are presented: volume, aspect ratio, convexity, and mesh self-intersection. The tuning of optimisation parameters and generation of the initial population of the optimiser is then described. Thus, the full optimisation problem is presented in a thorough, reproducible manner.

6.1 Geometry Parametrisation



$(x_{11}\, y_{11}\, z_{11})$
$(x_{31}\, y_{31}\, z_{31})$
$(x_{21}\, y_{21}\, z_{21})$
ADBSat input matrices
Three vertices fully describe one triangular plate

$$\begin{array}{|c|c|c|c|}\hline x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ x_{31} & x_{32} & \cdots & x_{3n} \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|c|}\hline y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ y_{31} & y_{32} & \cdots & y_{3n} \\ \hline \end{array}$$

$$\begin{array}{|c|c|c|c|}\hline z_{11} & z_{12} & \cdots & z_{1n} \\ z_{21} & z_{22} & \cdots & z_{2n} \\ z_{31} & z_{32} & \cdots & z_{3n} \\ \hline \end{array}$$

Figure 6.1. A pictorial description of how each triangular plate is represented in ADBSat.

With the aim of aerodynamic optimisation in mind, it was necessary to devise a geometry parametrisation which is consistent with both the ADBSat drag analysis program and the GA function. ADBSat requires a geometry made of a set of ϕ triangular flat plates. Each plate is represented by three vertices, each described by an x, y and z co-ordinate. The triangles comprising the entire shape are fully defined by three 2-dimensional data structures, one each for the x, y and z dimensions. Each data structure comprises three rows and ϕ columns. Each column describes the position of a triangular plate by listing the three co-ordinates of its points in a particular axis, coincident with the definitions used in ADBSat. A pictorial representation of this can be seen in fig. 6.1. While advantageous for the drag analysis program, this representation offered a number of challenges in being adapted for optimisation.

The GA function in MATLAB requires the optimisation decision variables to be formatted as a single, one-dimensional vector. Widely used methods of geometric parametrisation normally involve the representation of the shape through mathematical functions such as polynomials, Bezier curves, or non-uniform rational B-splines. The parameters of the mathematical functions are then controlled by the GA. This approach is often used for cases where the optimisation geometry has a well-defined basic shape, such as airfoils [246]–[250]. However, in seeking to afford the designer the most control over any satellite body, such a well-defined basic shape may not exist. Additionally, the desire to optimise the geometry in three dimensions means that the functions required would themselves be extremely complicated - much more complicated than when optimising a two-dimensional cross-section of an object. Finding a function or set of functions that would accurately represent the shape would be non-trivial. At each time-step, the geometry would have to be converted from the plate representation required by ADBSat, to the mathematical function(s), and back to a set of physical three-dimensional triangular plates. The quality of the resulting mesh would be difficult to elucidate, and information loss would occur across the many conversions. Consequently, it was decided to use the three-dimensional mesh points of the input shapes as direct inputs to the optimisation function.

The three-matrix representation of ADBSat itself was not compatible with the GA. Because it represents the triangular plates, and not the points themselves, it would be necessary to find and change all instances of a point everywhere in the shape. There is no restriction on the number of plates that a certain point can be a part of - for example, when representing a cone using flat triangular plates, the point at its tip is a part of all plates that make up the curved surface. Figure 6.2 illustrates this phenomenon, with all 201 triangular plates that make up the curved side of the shape having one point in common. Thus, it was necessary to devise a pipeline which outputs one optimisation vector that contains the points which the optimiser will alter. It is also important that the optimisation vector does not change in length during the simulation, as the GA in MATLAB exits with an error when this occurs.

The capabilities of Blender to work through a command script written in Python were harnessed. A set of shapes were made in Blender which comprised the same number of points. Two cases were considered: one in which only ψ selected points are eligible for optimisation, and one where the entire shape is optimised. Where only part of the

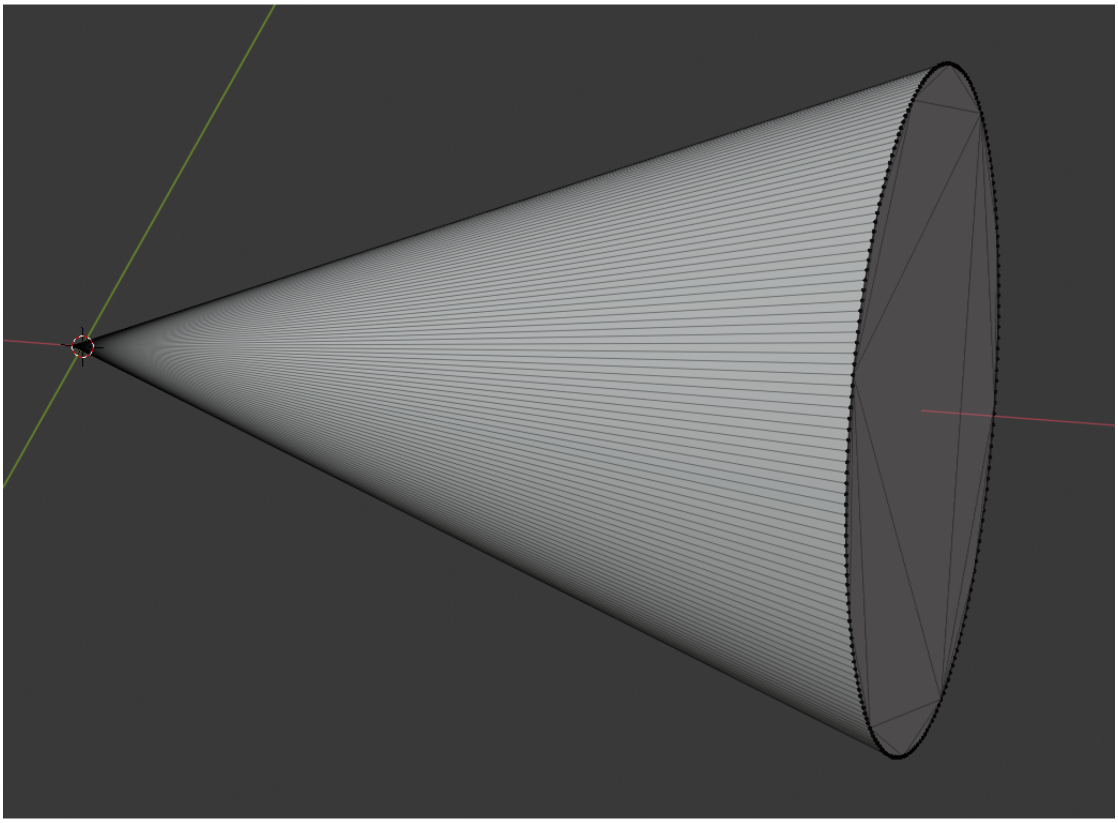


Figure 6.2. A model of a cone in Blender. The individual plates making up the curved surface of the cone can be seen, which all share one common point.

shape is being optimised, it is assumed that the rest of the shape is identical across the whole population, in other words that examining the non-optimisation points of any one shape is representative of the entire set. Once a set of shapes is designed, one shape at a time is imported into Blender, and the optimisation vertices selected. The Python script writes the unique selected vertices to a simple text file with the ".optVerts" extension. The same script then ensures that all plates in the shape are triangular and that the axes match those of ADBSat, before saving the file in ".obj" format.

Once the shape and the related .optVerts file is output by Blender, a modified version of the ADBSat function which converts the Wavefront file to a MATLAB database is used to import the file and its optimisation vertices. This function is identical to the original, with the addition of scanning for a .optVerts file with the same name as the .obj file, reading it, and saving its information as a new field in the resulting database. When all the shapes have been imported via ADBSat, the resulting database files make up the initial population of the GA.

Thus, there exist a number of MATLAB databases which represent the initial shapes, with each having a field listing the vertices to be optimised. An optimisation vector is created from each shape as follows:

1. for any one shape:

   (a) load database

   (b) load 2D x, y and z ADBSat matrices and 2D matrix of ψ optimisation vertices

   (c) reshape x, y and z matrices into one list of all triangle vertices, with each line of the format (x y z), of length g.

   (d) find the unique vertices in the list and the translation back to the previous list.

   (e) find location of optimisation vertices in unique vertices list.

   (f) re-arrange ψ optimisation vertices into separate 1D vectors of x, y and z co-ordinates, each of ψ length

   (g) concatenate the three vectors into one vector of length 3ψ where the elements from 1 to ψ are the x co-ordinates, $\psi + 1$ to 2ψ are the y co-ordinates, and $2\psi + 1$ to 3ψ are the z co-ordinates. This list is the optimisation vector.

2. for all other shapes repeat only 1(a), 1(b), 1(f) and 1(g) above.

A pictorial representation of this can be seen in fig. 6.3. As it is assumed that any non-optimisation vertices are the same across all shapes, it is only necessary to perform steps 1(c), 1(d), and 1(e) for any one shape. The outputs of these steps can then be used for all shapes to re-create an ADBSat-compatible format, which can be analysed for aerodynamic characteristics. Thus, it is possible to extract a 1D vector of optimisation vertices from any CAD geometry and its corresponding .optVerts file, which can easily be converted back to a full shape for analysis by ADBSat. The 1D vector is suitable for optimisation by the GA.

6.2 Objective functions

6.2.1 Drag

Minimising the aforementioned aerodynamic drag is a key objective. ADBSat can be incorporated into the optimisation routine as outlined above, yielding the drag coefficient C_d with a reference area of half of the total surface area of the body. While an adequate measure of drag, optimising C_d alone lacks some necessary information about the body characteristics. In particular, C_d is a function of the shape of the object, but does not contain information about its size. This information is incorporated into the drag force equation by multiplication with the reference area, A_{ref}, as shown in eq. (3.3). Due to this, the assumption that a shape of lower C_d will experience a drag force lower than a shape of higher C_d does not always hold. The reference area must be taken into account for the optimisation to capture more comprehensive information about the shape at hand.

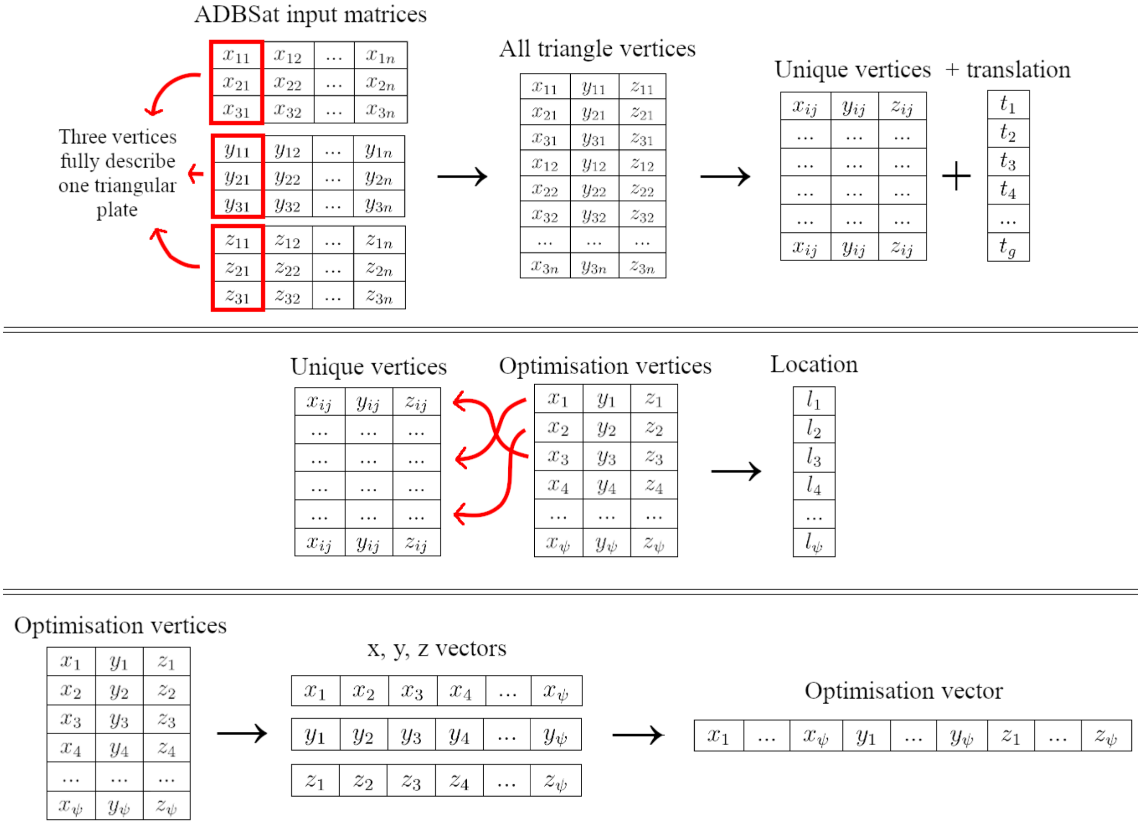


Figure 6.3. A pictorial representation of the conversion from the three ADBSat input matrices to one optimisation vector. The top panel shows steps 1(a) to 1(d), the middle panel shows step 1(e), and the bottom panels shows steps 1(f) and 1(g).

As an example, consider the aerodynamic analysis of two cones of height to diameter ratio 1.0, with all environmental parameters held constant. One cone has a height of 0.5 m while the other has a height of 5 m. The shapes are identical in configuration with only their size differing between the two cases. Although both shapes exhibit the same drag coefficient, they differ by two orders of magnitude in the value of $C_d A_{ref}$, where A_{ref} is the reference area used by ADBSat for the calculations. This can be seen in fig. 6.4, which shows the two different geometries and their associated C_d and $C_d A_{ref}$ values. The combination of $C_d A_{ref}$ gives a more accurate picture of the magnitude of the drag force that the object will experience, and its behaviour with increasing object size. Thus, $C_d A_{ref}$ was selected as the optimisation parameter, such that the dimension of the satellite is also taken into account.

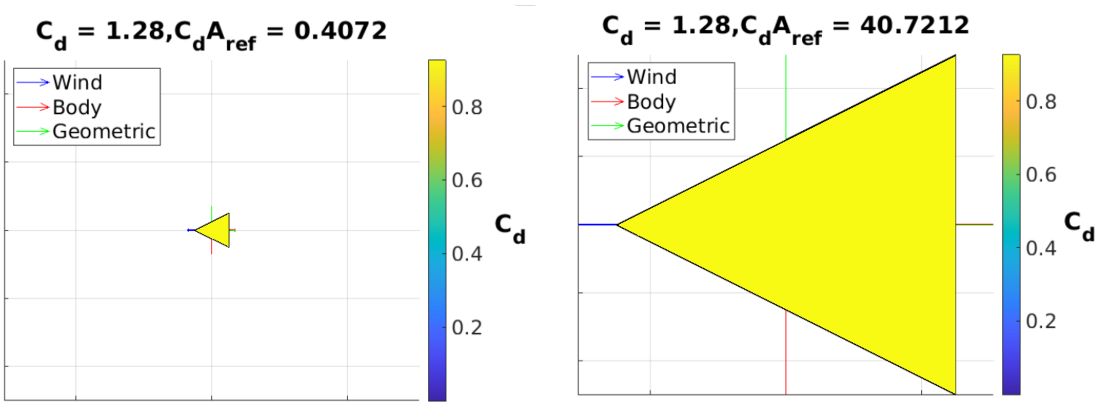
6.2.2 Volume

While reducing drag, it is important to maintain the functionality of the satellite body. If a satellite is low in drag but is unable to carry a useful payload or is uncontrollable and cannot fulfil pointing requirements, it will fail its mission. Thus, another essential objective is to maximise the volume v of the body, or minimise $-v$, to allow space for the payload and internal spacecraft subsystems.

ADBSat does not have the capability to calculate the volume of the shape it analyses. It was therefore necessary to integrate a new routine to perform this calculation. Voxelisation is one common way of representing a 3D shape. It is the 3D equivalent to pixelation, in which a 2D image is approximated by discrete uniform squares each of a single colour, called pixels. Voxelisation approximates a 3D shape by discrete uniform cubes called voxels. Knowing the volume of a voxel, it is a simple matter to count the number of voxels in the shape and obtain an approximate shape volume. Examples of pixelation and voxelisation can be seen in fig. 6.5.

The voxelisation of solid objects is non-trivial, involving complex computational procedures such as ray-tracing and interpolation. Here, the algorithm used is that of Adam A. [251]. It was chosen due to its ability to take as an input a mesh made of triangular facets in the same representation as that used by ADBSat, which are already available as part of the optimisation routine. Thus, it was readily compatible with the existing framework. It has extensive positive community reviews and clear documentation stating its uses, inputs, outputs, and changes. It has undergone changes to promote memory efficiency and decrease runtime. It employs a peer-reviewed ray-tracing algorithm [252]. Thus, it was a logical choice with which to voxelise the candidate solutions. The program outputs are a cube-shaped binary grid with 0 meaning no voxel at that position and 1 meaning a voxel exists. The co-ordinates of each voxel are also output. The difference between the two central co-ordinates in each of the three axes are multiplied together to give the volume of one voxel. By summing the binary grid it is possible to count the voxels quickly. Multiplication by the volume of a single voxel yields the total shape volume.

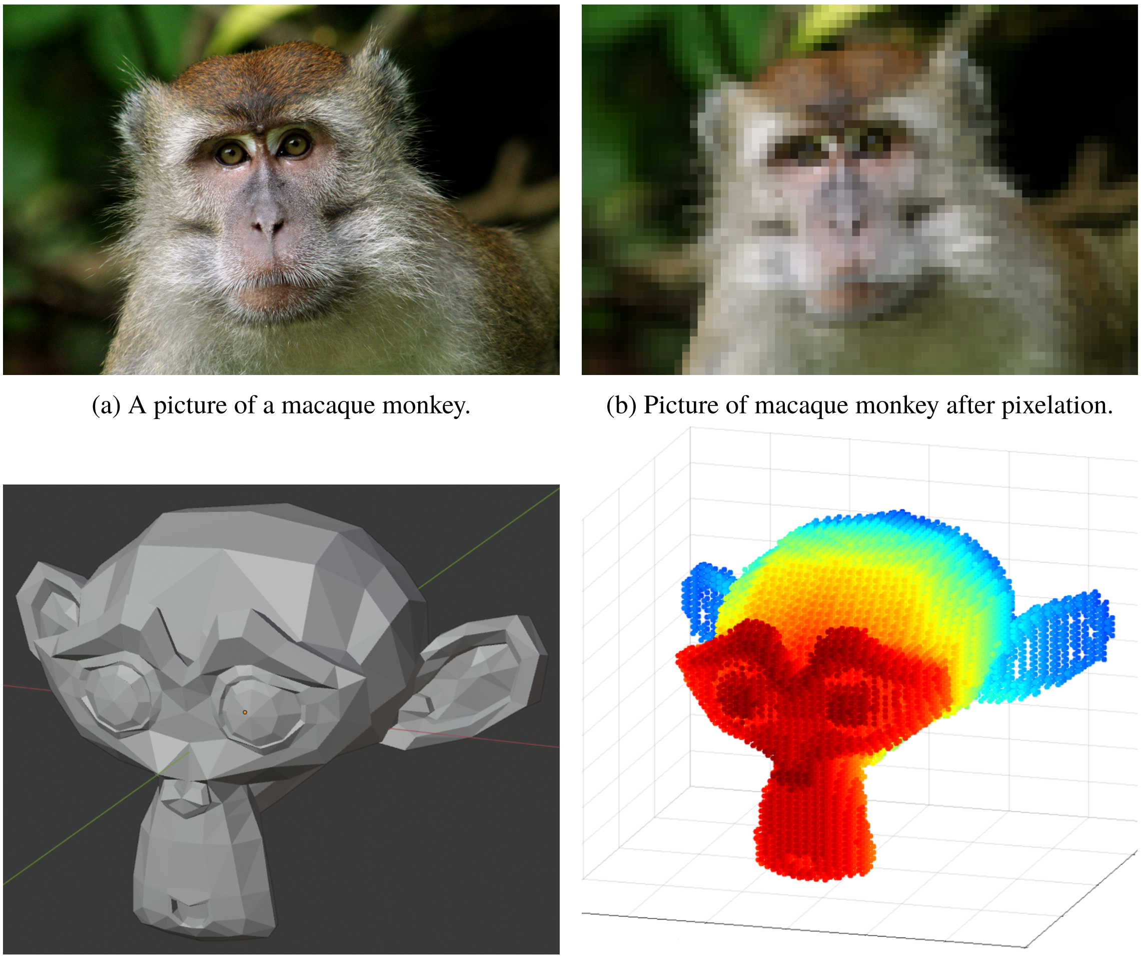
Tests were conducted to ascertain the influence of number of voxels, shape size, and the number of triangular panels making up the shape on runtime and accuracy. A set of 22 test shapes were created in Blender, and the volumes of the triangular meshes in question as output by Blender was recorded. These volumes are assumed to be correct. Grids of



(a) Cone, height and base diameter 0.5 m.

(b) Cone, height and base diameter 5 m.

Figure 6.4. Aerodynamic analysis of the same conical shape, scaled to different sizes. The difference in C_dA_{ref} for shapes of same C_d can be seen.



(a) A picture of a macaque monkey.

(b) Picture of macaque monkey after pixelation.

(c) A 3D model of a monkey head.

(d) Monkey head after voxelisation.

Figure 6.5. A picture of a macaque monkey simplified by pixelation, and the equivalent voxelisation procedure on a 3D model of a monkey head. The location of each voxel is represented by one filled circle.

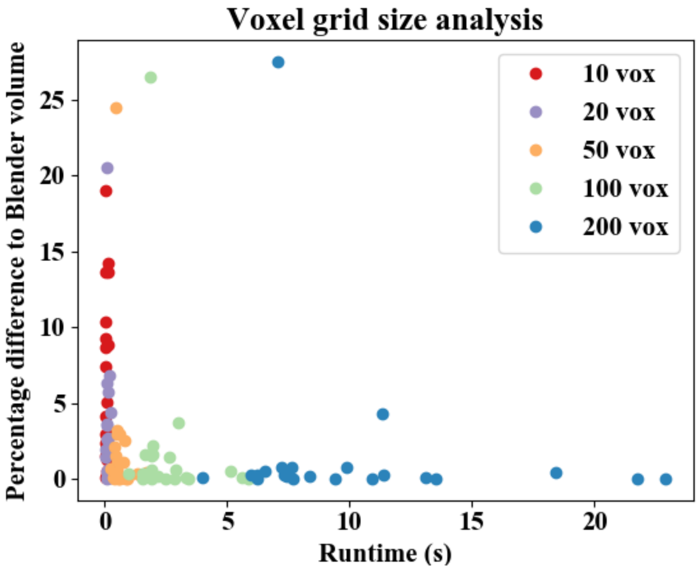


Figure 6.6. An analysis of runtime versus percentage error for different sized voxel grids.

10x10, 20x20, 50x50, 100x100, and 200x200 voxels were used to voxelise all shapes, and their volumes compared to those obtained from Blender. The runtime of each was also examined. Results can be seen in fig. 6.6, and are summarised in table 6.1.

Voxel grid size	Runtime <1 s	Volume within 3%
10x10	100%	50%
20x20	100%	86%
50x50	86%	91%
100x100	5%	91%
200x200	0%	91%

Table 6.1. A summary of the investigation into voxel grid size. The middle column shows the percentage of shapes examined in less than 1 s to run, and the rightmost volume the percentage of shapes where volume was within 3% of that output by Blender.

As ADBSat is accurate within 3%, this limit was also chosen to test voxel accuracy. As expected, using a larger number of voxels increases the accuracy. However, this does not increase uniformly but peaks at a 50x50 voxel grid, beyond which no increase is seen. Furthermore, beyond 50x50 the proportion of shapes examined in less than 1 s decreases dramatically, with a drop from 86% at 50x50 to 5% at 100x100. Therefore, a 50x50 grid seems the logical choice to use in optimisation, balancing accuracy and runtime.

The behaviour of the algorithm for a 50x50 voxel grid was further analysed in order to ensure its suitability for optimisation. Two sets of spherical test models were created, one which varied only in the number of triangular plates used to represent the sphere, and one which varied only in the sphere radius. The runtime of the voxelisation routine was tested across both sets, with the results seen in fig. 6.7. The runtime was seen to fluctuate as the radius of the sphere was increased, but no significant changes were observed. While the runtime generally increased with the number of triangular plates in the object, it was still easily under 1 s even for the sphere made of 175000 triangular plates, far more than necessary to accurately represent the sphere. In conclusion, the runtime remains suitable across a range of model conditions.

Additionally, the outputs of the volume calculation of Blender, the voxelisation routine, and the mathematical formula for volume of a sphere were compared as a function of the number of triangular plates in the spherical model. This is shown in fig. 6.8. Blender and the voxelisation routine agree closely for all models. As the number of triangular plates decreases, the volume deviates from that calculated mathematically. This is seen in both the Blender values and those output by the voxelisation, indicating a problem with the mesh as opposed to the volume measurement. Indeed, as the spherical model at its core is a quasi-spherical polyhedron, using too few triangular plates means it no longer accurately represents a sphere. Instead, the model is the largest quasi-spherical polyhedron with clear flat triangular sides that could be contained in a sphere of equivalent radius. This explains the reduction in volume seen for a low number of faces. Therefore, the volume measurement algorithm is deemed to be accurate, as it closely matches both Blender, and the mathematically calculated value when the CAD model of the shape

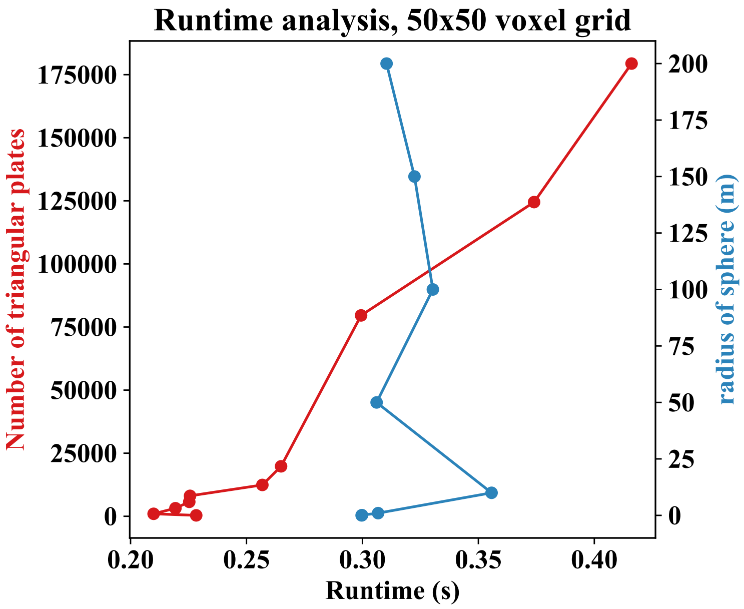


Figure 6.7. An analysis of the runtime for voxelisation of a sphere of varying radius, and separately for a varying number of triangular plates in the mesh.

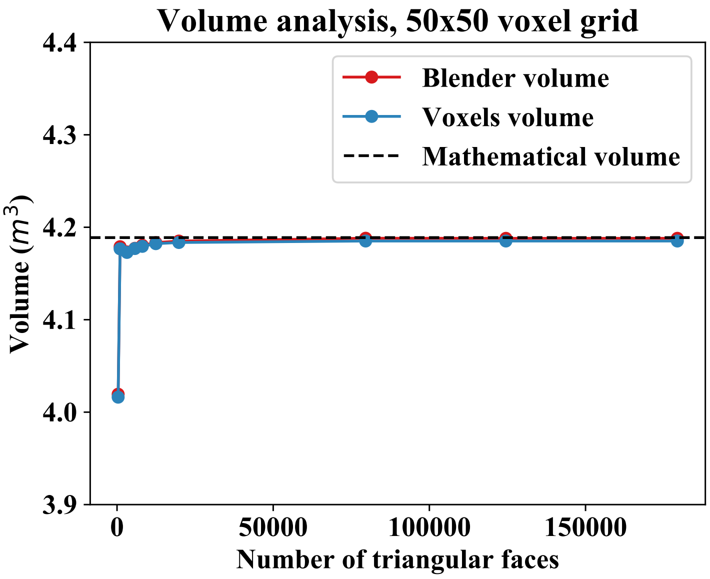


Figure 6.8. An analysis of the volume output by Blender and by the voxelisation algorithm for spheres of constant size, but made of different numbers of triangular plates. The mathematically calculated volume for the sphere in question is also shown.

is itself accurate. As the GA in MATLAB requires objectives to be minimised, the optimisation objective is $-v$.

6.2.3 Relative position of centres of gravity and pressure

The stability characteristics of the satellite are important to a successful mission. Whilst some missions may not differentiate between tumbling and non-tumbling attitude, most will require some degree of flow pointing. Pointing the payload may also be important for obtaining good quality data. One criterion for stability is the relative position of the centre of pressure (CoP) and centre of gravity (CoG). Moving the CoG towards the frontal face of the body increases stability, with aerostable configurations having the CoG in front of the CoP.

An alternative is neutral stability, resulting in an agile spacecraft which does not return to a flow-pointing attitude but is able to maintain a required attitude within a specific range of angles of attack and sideslip. This is particularly useful for missions which require payload pointing, such as some Earth observation missions. In this configuration, the CoG and CoP are located at the same position. This configuration is more difficult to determine by a human, as it requires more accuracy in knowing the positions of the CoG and CoP. Therefore, optimisation will play a significant part in identifying suitable geometries.

Determining the CoG and CoP are non-trivial tasks. As the voxelisation of the shape must be performed for the volume calculation, voxel positions are readily available and advantageous in calculating CoG. However, unlike the calculation of volume where it can be assumed that each voxel is identical, most realistic satellite shapes do not have a uniform mass distribution, and thus the weight of each voxel is not consistent across the shape. Designing the layout of the internal components specifically to have a high mass concentration towards the front of the spacecraft is common practice for missions requiring aerostability. Incorporating this feature into the model is essential to re-creating realistic mission conditions.

To simulate this effect, mathematical schemes were devised which dictate individual voxel weights in the x-axis direction. Using a satellite body mass specified by the user, the uniform mass per voxel can be determined. Each scheme then determines a multiplicative factor for each voxel weight as a function of x position, which renders the voxels at the front of the shape heavier than those at the back. By multiplying with the uniform mass per voxel, a modified weight is assigned to each voxel. As well as uniform mass weighting, six weighting functions were tested which acted on the voxel indices of the 50x50 grid in the x direction. The schemes were devised to span as large a range of weight distributions as possible. These are as follows, with x_{ind} being the x index:

1. **Linear:**

$$w = 0.098x_{ind} + c \tag{6.1}$$

2. **Sinusoidal:**

$$w = 5.6\sin(\frac{x_{ind}}{24}) + 0.1 \tag{6.2}$$

3. **Exponential 1:**

$$w = \exp(\frac{x_{ind}}{20} - 0.85) - 0.3 \tag{6.3}$$

4. **Exponential 2:**

$$w = \exp(\frac{x_{ind}}{6} - 6.75) + 0.1 \tag{6.4}$$

5. **Fractional 1:**

$$w = 5.6\frac{x_{ind}}{x_{ind} + 8} + 0.1 \tag{6.5}$$

6. **Fractional 2:**

$$w = 6.2\frac{(x_{ind} + 4)^2 - 11}{(x_{ind} + 4)^2 + 9} - 1.1 \tag{6.6}$$

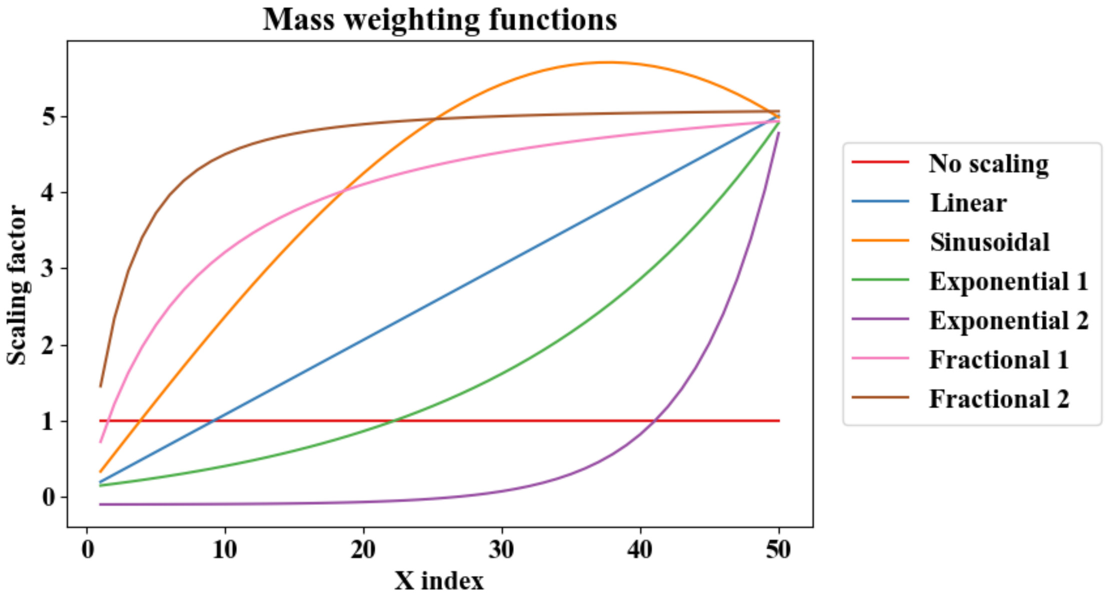


Figure 6.9. The different mass weighting schemes devised for the centre of gravity calculation.

The effect of each weighting scheme can be seen in fig. 6.9, with voxel one being the tail of the satellite. Different satellite body shapes with varying weight distributions were examined. When considering a realistic satellite weight distribution, it is important to factor in the proportions of real satellites, and where different components could be laid out. The exponential mass weighting schemes, while particularly advantageous due to their large concentration of mass at the front, render the tail to be almost empty - an unrealistic requirement in the context of satellite flight, where every cubic centimetre of space is at a premium. Empty space is expensive. Thus, they were ruled out. The linear scheme does not reflect a realistic weight distribution of components, which can seldom be lined up in such a neat ascending order of weight. The advantage of the sinusoidal scheme over the two fractional schemes lies in its peak being before the frontal face. This reflects a possible heavy internal component, placed towards the front of the satellite, which most likely has its weight centred within its own internal dimensions. Thus, such a distribution could be realistic. The two fractional distributions flatten sooner, and may be representative of internal components of a more even weight. While these may also be realistic, the advantages offered by the sinusoidal distribution render it the most useful for the analysis at hand. Thus, the sinusoidal weight distribution was selected for all cases, to reflect more likely conditions.

Having established the weight and position of each voxel, the centre of mass can be determined using the well-known eq. (6.7). Here, R_{CoG} is the position of the CoG, M is the body mass, m_i is the mass of voxel i, r_i is its position, and n_v is the number of voxels. The y and z directions are assumed to have uniform mass weighting, although a similar mass weighting to that used along the x axis could easily be applied. The three directions are treated independently in order to find the x, y and z co-ordinates of the CoG.

$$R_{CoG} = \frac{1}{M}\sum_{i=1}^{n_v} m_i r_i \tag{6.7}$$

The CoP is the point on which the average pressure force acts. For a continuous shape, the x co-ordinate of the CoP, x_{CoP}, can be calculated from eq. (6.8). Here, $p(x)$ is the pressure as a function of the x co-ordinate x. However, the input CAD model is not continuous, but comprised of discretised panels. Discretising eq. (6.8) and assuming the pressure coefficient C_p is a proxy for pressure results in eq. (6.9), where x_i is the x co-ordinate of the barycentre of each panel and C_{pi} is the pressure coefficient of that panel. The same analysis can be applied in the y and z directions to obtain the co-ordinates of the CoP.

$$x_{CoP} = \frac{\int xp(x)dx}{\int p(x)dx} \tag{6.8}$$

$$x_{CoP} = \frac{\sum x_i C_{pi}}{\sum C_{pi}} \tag{6.9}$$

The relative difference in x co-ordinates between CoP and CoG can now be used as an optimisation objective. Henceforth, the relative position of the CoP and CoG shall be referred to as Δ_{PG}. The problem has been formulated such that $\Delta_{PG} < 0$ indicates that the CoG is forward of the CoP. A more negative Δ_{PG} is desirable for aerostability, where the spacecraft will always return to optimum flow-pointing attitude when a disturbance is experienced.

To maintain the validity of increased stability when the CoG is in front of CoP, it is also necessary for the two to line up along the x axis, in other words for them to have the same co-ordinates in y and z. The characteristics of the reference body shape, as described in section 6.5, should play a large part in ensuring that this is the case. To more strongly impose this condition, some cases employed two additional objectives, one each to minimise the difference in y and z co-ordinates between CoP and CoG. These additional objectives are highlighted when used, and are referred to as $|\Delta_{PG_y}|$ and $|\Delta_{PG_z}|$ respectively.

Furthermore, for a neutrally stable aircraft as described above, the problem formulation requires $\Delta_{PG} = 0$. This is achieved by minimising $|\Delta_{PG}|$. The result will be an agile spacecraft that is easily able to change to and maintain a new attitude.

6.2.4 Pointing stability

Further controllability characteristics are discussed by Mostaza-Prieto and Roberts [253], which determine the pointing stability of the satellite body by examining the derivatives of various aerodynamic coefficients. They are advantageous as optimisation objectives in better considering the behaviour of a geometry. Aerostability conditions based on the derivatives of moment coefficients are detailed in eqs. (6.10) and (6.11), with some parameters clarified in eqs. (6.12) to (6.14).

$$1 + \frac{Q}{3\omega^2} \frac{A_{ref} L_{ref}}{I_y} C_{m_\alpha} < 0 \tag{6.10}$$

$$1 - \frac{Q}{\omega^2} \frac{A_{ref} L_{ref}}{I_z} C_{n_\beta} < 0 \tag{6.11}$$

$$C_{m_\alpha} = \frac{\partial C_m}{\partial \alpha} \tag{6.12}$$

$$C_{m_\beta} = \frac{\partial C_n}{\partial \beta} \tag{6.13}$$

$$Q = \frac{1}{2} \rho u^2 \tag{6.14}$$

Here, L_{ref} is the reference length of the shape, which ADBSat defines as difference between the maximum and minimum co-ordinates along the x axis. I_j is the moment of inertia about axis j and ω the angular orbital velocity of a circular orbit. The relevant moment coefficients are C_m, the pitch moment coefficient, and C_n, the yaw moment coefficient, both translated to the CoG. This method is easily compatible with ADBSat, which requires or calculates six of the above parameters as part of its drag coefficient algorithm: ρ, u, A_{ref}, L_{ref}, C_m and C_n. Since ADBSat also requires as an input the orbital altitude, ω^2 can be calculated following the well-known eq. (6.15). In this equation, μ_E is the gravitational parameter of the Earth and R is the semi-major axis of the orbit.

$$\omega^2 = \frac{\mu_E}{R^3} \tag{6.15}$$

C_m and C_n can be calculated by ADBSat at any value of α and β respectively. Therefore, the derivatives with respect to the appropriate angles can be numerically estimated. The method chosen to do this was the central difference formula, as shown in eq. (6.16) [254], [255]. Here, $f(\gamma)$ is some function f of some variable γ, and h is the step size for the central difference formula estimation, assumed to be small. Based on a truncated Taylor series, this formula can approximate the derivative quickly and only requires two additional function calls to ADBSat per parameter, keeping runtime low. With a truncation error of order of magnitude $O(h^2)$, it is comparable to the error exhibited by ADBSat itself for small values of h.

$$\frac{\partial f}{\partial \gamma} = \frac{f(\gamma + h) - f(\gamma - h)}{2h} \tag{6.16}$$

Translating the vector of moment coefficients to the previously determined CoG is done through eq. (6.17) [253]. Here, $\vec{C}_f$ is the vector of force coefficients, $\vec{C}_M$ is the vector of moment coefficients, and $\vec{C}_M^{CoG}$ is the vector of moment coefficients translated to the CoG. It is assumed that $\vec{C}_f$ and $\vec{C}_M$ are both in the correct frame, here being the body frame, and both refer to the same point prior to translation.

$$\vec{C}_M^{CoG} = -(\vec{R}_{CoG} \times \vec{C}_f) + \vec{C}_M \tag{6.17}$$

The voxelisation sub-routine also proves helpful in calculating the moments of inertia, I_y and I_z. The set of voxels representing the body is functionally a set of point masses. Their collective moment of inertia is assumed to be a good approximation of the moment of inertia of the body. For moments of inertia of point masses calculated around the same axis, the body moment of inertia can be found using eq. (6.18). Here, d_i is the distance of a voxel from the relevant axis. It is assumed that the reference axes are parallel to the x, y and z body axes respectively, and their origin is located at the CoG. This is consistent with the translating of $\vec{C}_M$ to the CoG. It is then trivial to find the distance from each point to the axis and perform the sum in each of the three axes.

$$I_j = \sum_{i=1}^{n} m_i d_i^2 \tag{6.18}$$

Thus, the two aerostability characteristics can be used as objectives alongside $C_d A_{ref}$, $-v$, and Δ_{PG}. Henceforth, the two parameters described in this section shall be referred to as Ω_{m_α} and Ω_{n_β}. The problem has been formulated such that aerostability is indicated by $\Omega_{m_\alpha}, \Omega_{n_\beta} < 0$.

Where neutral stability is required, the magnitude of the moment coefficient derivative induced by a change in attitude should be zero, in other words $\Omega_{m_\alpha}, \Omega_{n_\beta} = 0$. The equivalent objectives are a minimisation of $|\Omega_{m_\alpha}|$ and $|\Omega_{n_\beta}|$, in a similar manner to $|\Delta_{PG}|$ above.

6.3 Constraints

Six constraints were designed to maintain the feasibility and realistic nature of the output shapes: a volumetric constraint, two aspect ratio constraints, two convexity constraints to maintain the validity of ADBSat's drag analysis, and a constraint on plate intersection for practicality. MATLAB imposes a standard construct for equality and inequality constraints, shown in equations eqs. (6.19) and (6.20).

$$c^{eq} = 0 \tag{6.19}$$

$$c^{ineq} \leq 0 \tag{6.20}$$

6.3.1 Volume

A constraint was placed on the volume such that it could not be reduced by any more than 20% when compared to the average volume of the initial population. No upper constraint on volume was devised. It may seem that this constraint is unnecessary due to the maximisation of the volume by the optimisation routine, which should ensure that shapes are always at least as big as the initial population. However, the multi-objective nature of the problem requires such a specification. In particular, as a smaller shape will have a smaller value of $C_d A_{ref}$ which is also an objective, the multi-objective algorithm may shrink the shape excessively in order to facilitate a reduction in $C_d A_{ref}$. To prevent this, it is assumed that the initial shapes are representative of the payload volume required, and the algorithm is explicitly prevented from making the shape significantly smaller. The threshold volume can easily be changed for future cases. The volume constraint is seen in eq. (6.21), where $\overline{v}$ is the average volume of the initial population, v^s is the volume of the candidate solution, and c_v is the maximum reduction in volume allowed, as a percentage.

$$\frac{100(\overline{v}-v^s)}{\overline{v}}-c_v\leq 0 \tag{6.21}$$

6.3.2 Aspect Ratio

Two aspect ratio constraints also serve to maintain a degree of similarity between the initial population and the output shapes, again with the assumption that the initial population is representative of the types of shapes that the mission requires. It is also assumed that the launch vehicle will impose some constraints over the dimensions and aspect ratios of the satellite body. Therefore, again assuming that the initial set is representative of the types of shapes that are feasible in this respect, a deviation in aspect ratio that is too extreme would be undesirable.

The reference lengths of the shape in the y and z axes are calculated in the same way as L_{ref} in the x axis, by finding the difference between the two extremes of the relevant coordinates. Two aspect ratios are examined, x:y and y:z. The third aspect ratio, x:z, is not an independent variable. Thus, placing a constraint on it would have resulted in an increase in program runtime without an increase in information gained. The aspect ratios were constrained to vary by no more than 25% in either direction when compared to the average aspect ratio of the initial set. Again this threshold is easily adjustable. The form of the two constraints can be seen in eq. (6.22), where ξ_{ij} denotes the aspect ratio of the axes i and j, $\overline{\xi_{ij}}$ is the average aspect ratio of the initial set, ξ_{ij}^s is the aspect ratio of the candidate solution, and c_ξ is the allowed margin on the aspect ratio as a percentage. Note the use of the absolute value, applying the constraint to both upper and lower limits, unlike the volume constraint in eq. (6.21).

$$\frac{100|\overline{\xi_{ij}}-\xi_{ij}^s|}{\overline{\xi_{ij}}}-c_\xi\leq 0 \tag{6.22}$$

6.3.3 Convexity

Convexity constraints are imposed to maintain the feasibility of shapes for ADBSat, which cannot handle deep concavities on the frontal face of an object. Calculating the convexity of an object is a non-trivial task for which many options exist. The method of Shi, Li, and Sheng [256] is particularly advantageous for the following reasons:

1. It requires only a 3D shape and its convex hull, which can easily be obtained from the list of unique points of the object via a built-in MATLAB function.

2. Voxelisation of the convex hull and the object using the same grid must be performed. The existing method for voxelisation can be harnessed by initially voxelising the convex hull, then using its output grid as the basis for the shape voxelisation.

3. The CoM of both voxelisations assuming uniform mass is needed. As a method for calculating the CoM has already been described, this is trivial.

4. It can emphasise the contribution of concave features at certain locations using a weight $0 \leq \alpha_w \leq 1$, where zero means the location has no influence. This is advantageous as ADBSat is particularly susceptible to errors due to concavities on the frontal face when compared to those from concavities on the rest of the body.

Upon voxelising the shape and its convex hull, the convexity measure can be calculated using eq. (6.23), where Ψ is the convexity (1 for a convex shape, reducing towards 0 as concavity increases), n_s and n_{CH} are the number of voxels in the shape and convex hull respectively, r is the distance of a voxel from the CoM of the convex hull with subscripts i and k denoting the shape and convex hull voxels respectively, and r_{max} is the maximum r.

$$\Psi = \frac{n_s - \alpha_w \frac{\sum_{i=1}^{n_s} r_i}{r_{max}}}{n_{CH} - \alpha_w \frac{\sum_{k=1}^{n_{CH}} r_k}{r_{max}}} \tag{6.23}$$

To further constrain the examination of convexity to the area of the shape that is relevant for ADBSat, a co-ordinate cut-off was introduced which controls how much of the shape is considered for the convexity measurement. A fraction of the total shape length along the x axis, β_w is specified. Starting from the front of the shape, any voxels which lie further back than the specified fraction of the shape are disregarded, resulting in the convexity measure of the front part of the shape. γ_w, a multiplicative factor for the CoM, is also imposed for the same reason, in order to facilitate moving the CoM forwards such that it is within the region of the shape considered for convexity analysis.

Thanks to the extensive validation of ADBSat [92], there exist a set of shapes for which ADBSat's performance is well-characterised. The convexity of the set of shapes used therein was calculated as a function of the three control parameters. The result can be seen in fig. 6.10. This was then used to devise two convexity constraints, which together remove the shapes on which which ADBSat shows higher error from the examined set:

1. $(\alpha_w, \beta_w, \gamma_w) = (0.2, 1.0, 0.2)$, c_1: The convexity of the front of the object is considered independently, with a low influence of concavity position. Shapes must show $c_1 \geq 0.4$ to be accepted.

2. $(\alpha_w,\ \beta_w,\ \gamma_w)\ =\ (1.0,\ 0.8,\ 1.0)$, c_2: The convexity of almost the whole shape is considered, with a heavy influence of concavity position. If $c_2\ >\ c_1$, the shape has more concavity towards the front, and is undesirable. A limit of 0.2 on the criterion is observed to exclude the shapes which show significant error, leading to a constraint of $c_2 - c_1 \leq 0.2$.

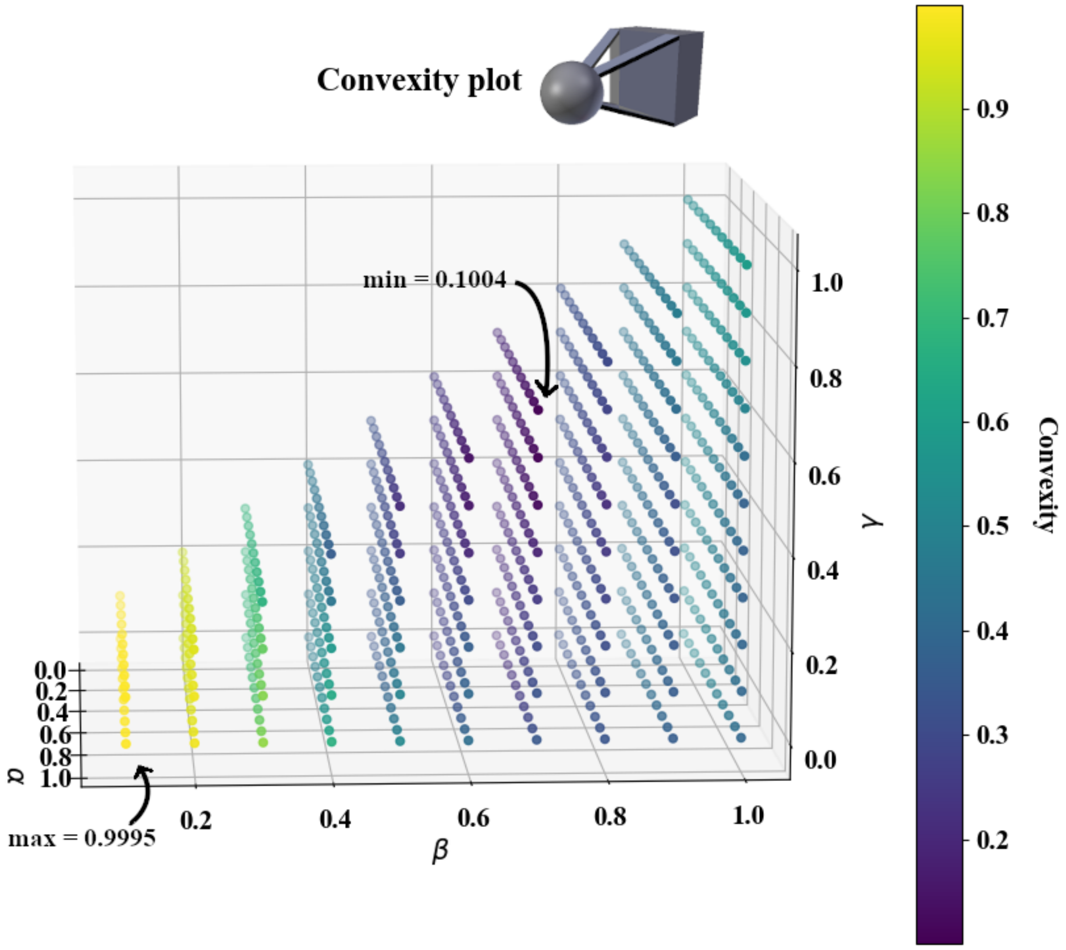


Figure 6.10. Convexity analysis for a test shape as a function of α, β, and γ.

6.3.4 Mesh self-intersection

A constraint on the intersection of the triangular plates comprising the model is also applied. As the plates represent the outside hull of the spacecraft, they are ultimately intended for manufacturing, and thus the constituent parts cannot intersect. The triangular shape of the panels is advantageous here, as triangle intersection algorithms are common in computer modelling [257]–[261]. In general they are non-trivial, computationally intensive, and often difficult to implement. Additionally, many attempt to directly compute the intersection, instead of merely determining if an intersection exists [262]. Since any one intersection of any type rules out a particular candidate solution, the detailed computation increases the function runtime over a simple determination of whether an intersection exists, without adding functionality for the optimisation framework. As the optimisation framework is itself computationally intensive, a short runtime and simplistic implementation was sought. A modified version of the algorithm of Johnson [263] fulfils these conditions well: it is quick, relatively simple and does not compute the intersection of the triangles but only determines if they intersect. Additionally, it is written as a MATLAB function which can easily be inserted into the optimisation framework. It only requires the co-ordinates of the two triangles in question, which can be re-shaped into the correct format from those required by ADBSat and the voxelisation algorithm.

Comprehensive testing of the algorithm was undertaken to ascertain its performance. Due to the parametrisation of the mesh, co-planar triangles cannot intersect without at least one additional triangle intersection off the plane. Therefore, only cross-intersection is of relevance. 49 test cases were devised from seven triangles, to test all possible intersections. It was discovered that the algorithm classifies triangles with one or two common points as intersecting. As the mesh is comprised of interconnected triangles, this is undesirable. Thus, two simple modifications were made to handle these cases:

1. If the two triangles $T1$ and $T2$ have one point in common labelled C, a unit vector $\vec{\hat{u}}$ is found in the direction from the barycentre of $T1$ ($B1$) to C. The common point in $T2$ is then moved along this vector away from $B1$ by a small amount, to create a new point C' and a new triangle $T2'$. A pictorial representation of this can be seen in fig. 6.11, with the movement of C to C' to create $T2$ greatly exaggerated. Then, the triangles are examined for intersection as before. The small movement is enough to differentiate the triangles and return no intersection if the common point was the only thing causing an intersection result, but this will still be a good approximation of the original configuration allowing existing intersections to be identified.

2. If T1 and T2 have two common points, it is assumed that they do not intersect. The only way they could intersect would be if they were co-planar, which is impossible due to the mesh parametrisation.

The algorithm returns zero if two triangles do not intersect, and one if they do. All triangles in the shape are examined for intersection against all other triangles, with the sum of total intersections constrained to be zero, in other words $\Sigma I = 0$. While this is a time-consuming operation and is at present a bottleneck for the code, it is important to maintain the feasibility of results. Without this constraint, the candidate solutions quickly become unrealistic.

6.4 GA parameters

As described in chapter 5, the problem of parameter tuning for the GA is non-trivial, with no obvious best solution. Due to the algorithm's implementation in MATLAB,

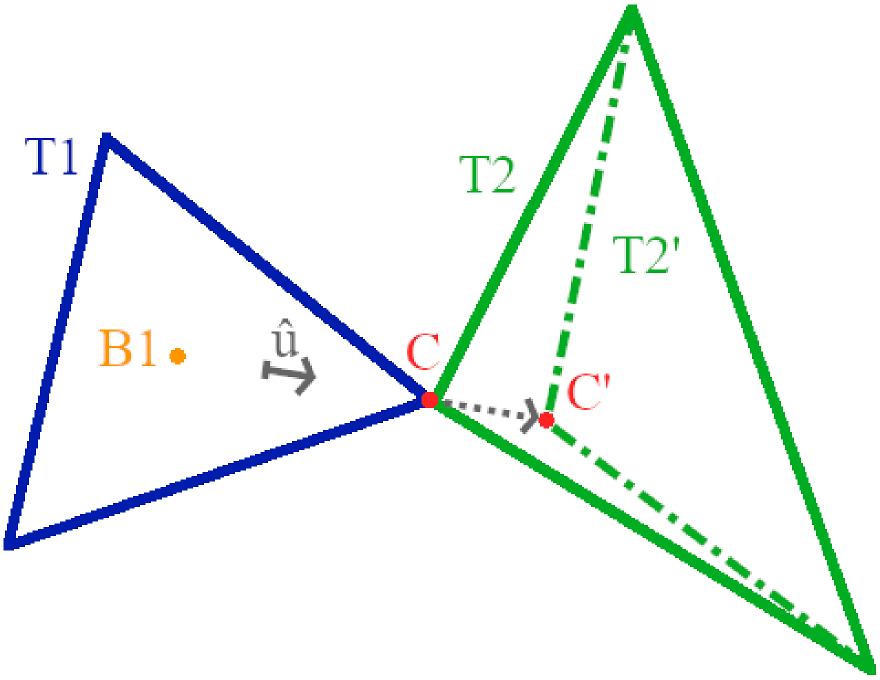


Figure 6.11. A pictorial representation of the method to remove the intersection between two triangles when they have one point in common. The movement of the point is greatly exaggerated for clarity.

some parameters could not be controlled. These include the distance measurement method which determines the distance between two candidate solutions, the mutation function and fraction, and the selection function. GA parameters which must be tuned are the crossover mechanism, crossover fraction, and the Pareto fraction which controls elitism. Stopping criteria were established through function tolerance, constraint tolerance, and maximum number of generations.

Extensive testing of algorithm behaviour was undertaken across a range of different parameter values. Literature which described optimisation by the GA was examined in order to find initial estimates for parameter options. Following this, single-objective and multi-objective optimisation cases were performed. The outcomes were analysed for convergence time, feasibility, stability of results across multiple runs, spread of the Pareto set under multi-objective optimisation, and quality of results. The five parameter combinations which gave the best results are summarised in table 6.2. All optimisation results in chapter 7 are based on combining the Pareto set of each of the five parameter combinations into one final results set, and analysing the combined set as one.

Crossover	C_{frac}	P_{frac}	F_{tol}	C_{tol}	M_{gen}
Scattered	0.4	0.15	10^{-15}	10^{-15}	50000
Intermediate	0.4				
Scattered	0.6				
Intermediate	0.6				
Arithmetic	0.8				

Table 6.2. Genetic Algorithm optimisation parameters. C_{frac} is the crossover fraction, P_{frac} is the pareto fraction, F_{tol} is the function tolerance, C_{tol} is the constraint tolerance, and M_{gen} is the maximum number of generations.

Crossover refers to the crossover mechanism employed by the algorithm, with crossover fraction C_{frac} being the fraction of the next generation created by crossover. While literature values normally span the range 0.6 to 0.8, it was observed that lower values also returned realistic results for the problem at hand. Pareto fraction P_{frac} refers to the number of individuals to select on the Pareto front, and is within the commonly seen range 0.1 to 0.3. Function tolerance F_{tol} refers to the change in value of the spread of solutions, and is used as a stopping criterion to signal convergence. C_{tol}, the constraint tolerance, controls the feasibility with respect to the constraints. Both F_{tol} and C_{tol} were set to values much lower than the default to encourage stringent enforcement of constraints and rigid convergence. The maximum number of generations M_{gen} was set to a high value, such that F_{tol} is normally the stopping criterion and the simulation converges, but was also intended to keep computational time practical.

The utopia point method [264]–[266] is used to examine the quality of results. The minimum C_dA_{ref}, Ω_{m_α}, Ω_{n_β} and Δ_{PG} along with the maximum v were identified across the entire Pareto set output by the optimisation. These best values across the set were combined into one one imaginary utopia point. Of course, this point is impossible to attain due to the trade-off in clashing parameters. However, the distance from each solution to the utopia point is useful in providing a measure of the quality of the results. Solutions which are closer to the utopia point are considered to be superior to the rest of the Pareto set.

There are two methods of obtaining a utopia point: separate single-objective optimisation of each objective, and taking the best result of each objective across the Pareto set in a multi-objective case. While it would seem like single-objective optimisation should always find the best possible solution to a parameter, this is not the case for metaheuristics. Due to their random nature, they do not guarantee finding true optimality on every run. Thus, it is possible that for any one parameter, multi-objective optimisation could find a better solution than its single-objective counterpart. Indeed, performing both single-objective and multi-objective optimisation reveals that multi-objective improves on some single-objective results. Thus, the best values of the Pareto set were chosen as the utopia point.

The distance between each candidate solution and the utopia point was calculated, and normalised by the distance between the utopia point and nadir point. The nadir point is obtained in the same way as the utopia point, except taking the worst instead of best solutions for each objective across the Pareto set. Equation (6.24) shows the formula applied to each candidate solution, with d_u being the distance to the utopia point, z_c being the candidate solution, z_u the utopia point, and z_n the nadir point. In this way, a meaningful measure can be obtained of the proximity of each candidate solution to the utopia point, in the context of the spread of the Pareto set.

$$d_u = \frac{|z_c - z_u|}{|z_u - z_n|} \tag{6.24}$$

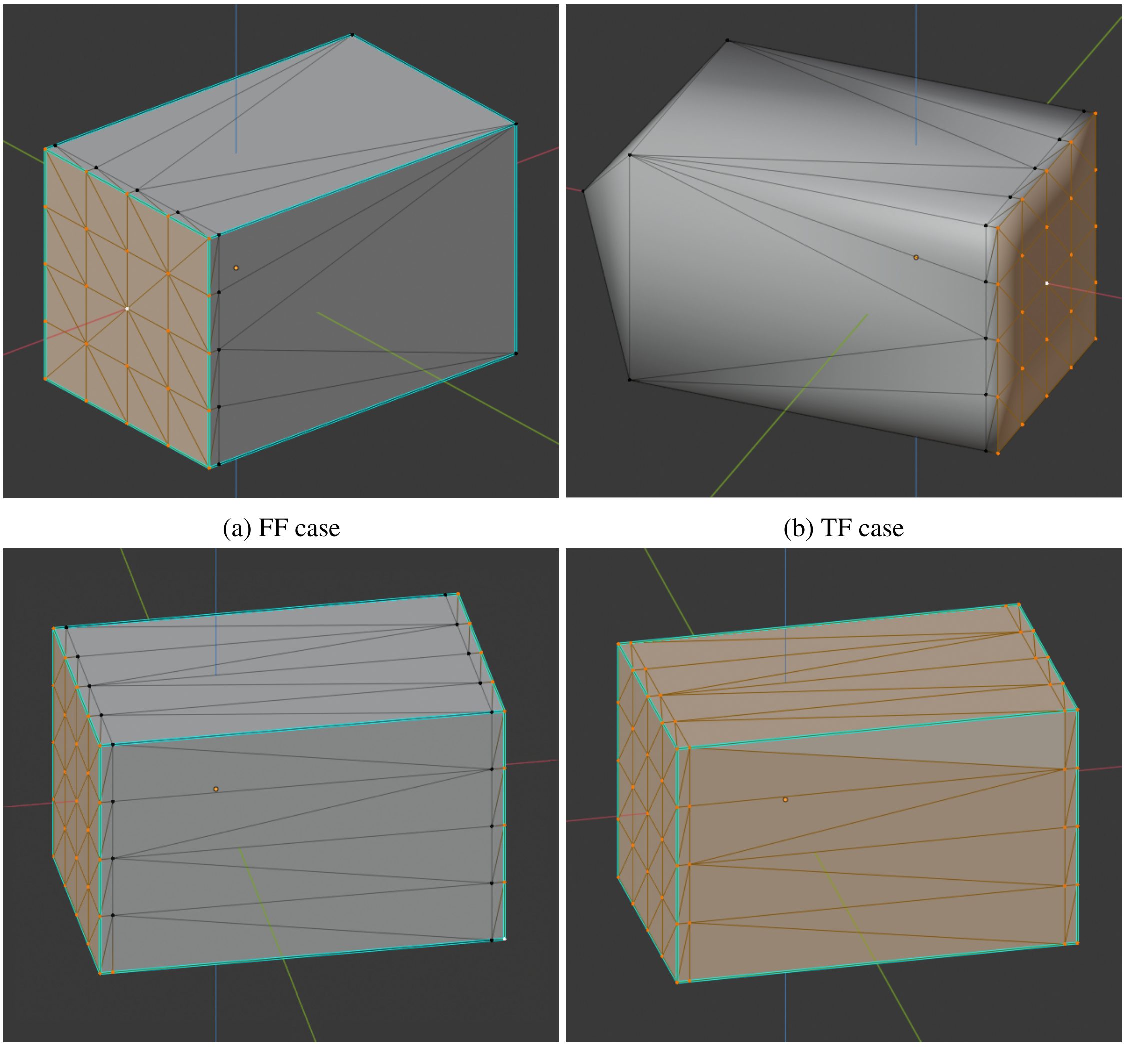
6.5 Initial Population Generation

Determining the size and composition of the initial population is a difficult task that is heavily problem-dependent, with no general guidelines. Having too small a starting population could lead the algorithm to inferior solutions, while increasing the population size excessively leads to infeasible computation times. A balance must be achieved in order to ensure that the algorithm has enough genetic diversity to be effective and prevent premature convergence, within a reasonable timeframe [267].

Initial populations for GAs are normally generated randomly, in order to attempt to capture as large a portion of the solution space as possible [268], [269]. However, it has been shown that the quality of the initial population affects the quality of the final solutions. A good starting population helps the metaheuristic arrive at good solutions [267], [270]. As a result of the geometric parametrisation and the constraints imposed, particularly the convexity constraints and the requirement that no triangular plates may intersect within the model, it was found that a randomly generated initial population was invariably wholly infeasible and of poor quality. Thus, it was decided to create the population by hand in order to ensure initial feasibility.

Four test cases cases were devised for optimisation: frontal face (FF), tail face (TF), front and tail faces (F&T), and the whole body (WB). All were based on extending a rectangular cuboid of dimensions 3 m×2 m×2 m, the shape and proportions of which are representative of many real satellites. TF shapes have a pointed feature on the front face, extruded 0.8 m from the centre point. The FF and TF case are characteristic of the previous attempts at optimisation in literature, summarised in chapter 5. The program is quite limited in its control of the body, with few control points and the majority of the body being fixed. The F&T case was designed to give the algorithm more control by doubling the number of control points. It also facilitates an investigation into the difference between optimising subsections of a geometry separately and putting them together (in other words, combining the results of the FF and TF cases) compared to optimising them at the same time. Finally, the WB case gives the algorithm full control over the body shape, with the highest number of control points and freedom. However, the general shape of the F&T case is maintained across much of the initial population for comparison purposes, and to represent constraints on manufacturing and launch vehicle requirements.

The basic shapes with their optimisation vertices highlighted can be seen in fig. 6.12.



(a) FF case

(b) TF case

(c) F&T case

(d) WB case

Figure 6.12. The basic geometric shape for the four optimisation cases. Optimisation vertices are highlighted in orange.

The (x,y,z) location of the 25 optimisation vertices was varied both systematically and randomly to create the initial population for the FF case. Variation in degree of symmetry, axis of symmetry, length and aspect ratio of features across the candidate solutions was introduced. The same variation was also introduced in both TF and F&T cases, with many of the same features seen across all three sets. A subset of the initial population of the WB case can be seen in fig. 6.13.

While normally a good population size is considered to be up to 20 times the number of decision variables [271], the population size employed here will be smaller than this for a number of reasons. Primarily, the geometric parametrisation and requirement of no self-intersection were designed in such a way as to limit the feasible position of each point to a space much smaller than the full 3D co-ordinate space. In other words, much of the space will be inaccessible to each point. With the solution space being smaller, so can the initial set be reduced in size. Additionally, creating the candidate solutions by hand ensures they are of particularly good quality, compensating for the effects of a

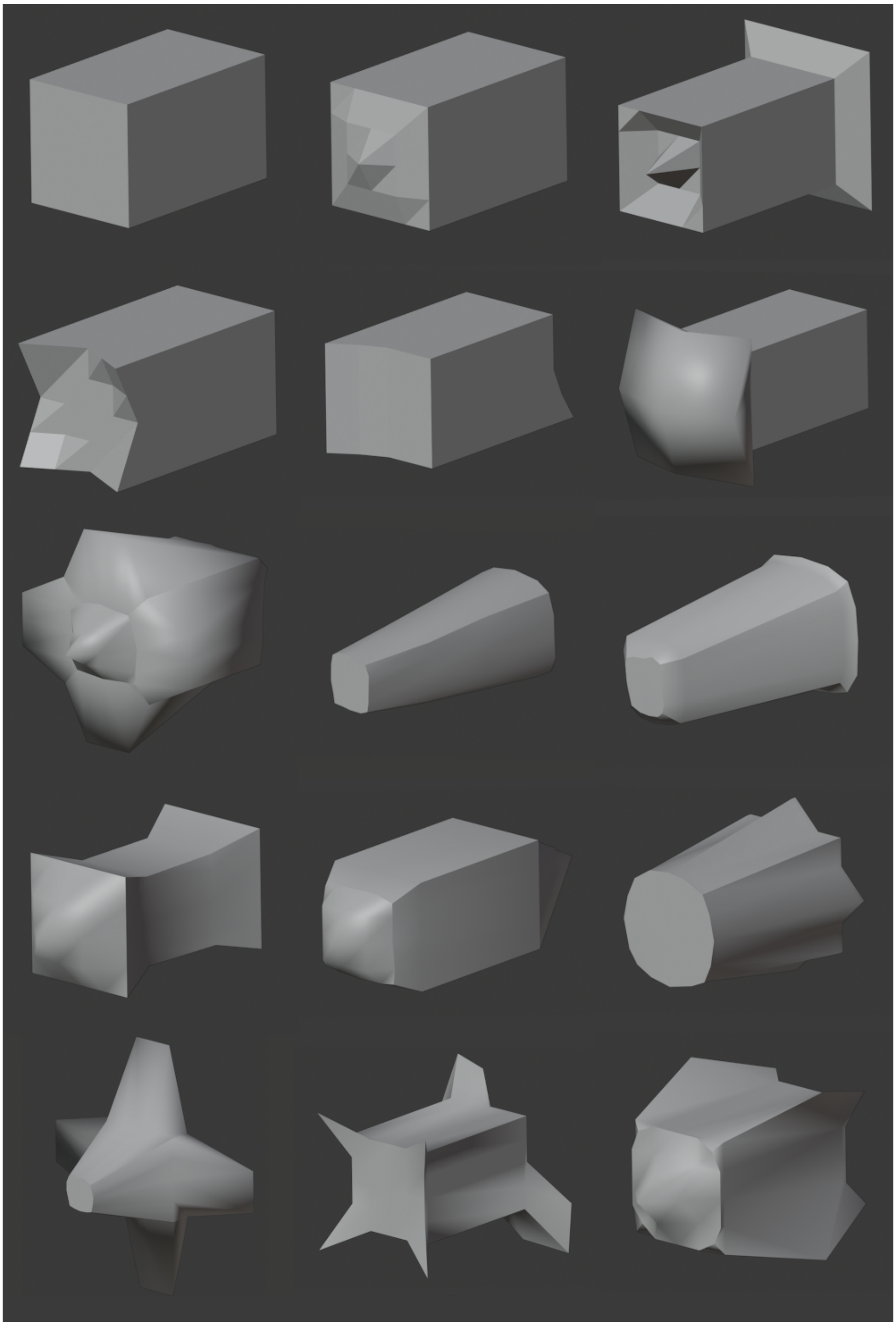


Figure 6.13. A collage of 15 shapes, representative of the initial population of the WB case.

small population on the algorithm performance.

A population size of 30 was chosen for the FF, TF, and F&T single-objective optimisation. The WB case involved some similar features as the other three cases, but the lack of a fixed cuboid body increased the genotype variation and allowed more different shapes to be created. Because of this, the population size was increased to 40 for this case. Optimisation converged well and appeared to produce good quality solutions for these cases. For the multi-objective optimisation, the initial population as above is supplemented by the addition of the single-objective results. For each case, each of the five objectives is individually optimised for the five parameter combinations in table 6.2, leading to 25 additional shapes in the initial population.

6.6 Summary

In summary, this chapter describes a proprietary a multi-objective computational optimisation framework to minimise aerodynamic drag while also considering other factors important for satellite flight. ADBSat has been integrated with the multi-objective GA algorithm available in MATLAB through a shape parametrisation, that successfully translates candidate solutions from the ADBSat mesh format into the single vector required for optimisation and vice versa. This is employed across all optimisation cases.

Appropriate primary objectives for aerostability for have been identified for the optimisation as follows:

- $C_d A_{ref}$: The product of reference area and drag coefficient, signifying drag scaled to the size of the spacecraft
- V: The volume of the body
- Δ_{PG}: The relative position of the centres of gravity and pressure in the X axis, signifying controllability
- Ω_{m_α}, Ω_{n_β}: Two further controllability parameters, as devised by Mostaza-Prieto and Roberts [253]

These objectives are intended to achieve aerostability of the target shape. Additionally, two more objectives were employed in some special cases described in chapter 7: $|\Delta_{PG_y}|$ and $|\Delta_{PG_z}|$, the relative position of the centres of gravity and pressure in the Y and Z axis respectively. Optimising (in other words, minimising) the absolute value signifies pushing the value of the parameters as close as possible to zero. These two parameters serve to keep the centres of gravity and pressure aligned along the X direction.

Some cases also consider optimisation aiming for neutral stability, where instead of minimising Δ_{PG}, Ω_{m_α} and Ω_{n_β}, the equivalent objectives will be $|\Delta_{PG}|$, $|\Omega_{m_\alpha}|$ and $|\Omega_{n_\beta}|$. This follows the same theory that rather than outright minimisation, the target is to reduce the parameters as close as possible to zero, and therefore the absolute value is minimised.

Furthermore, constraints have been identified on the problem, which primarily serve to maintain the realistic feasibility of the optimisation results. These are as follows:

- **Volume**: The volume cannot be reduced by the algorithm to less than 80% of the average volume of the initial population. It is assumed that the shapes in the initial population are representative of the payload volume necessary for the mission, and this constraint is intended to maintain the payload carrying capacity.
- **X-Y aspect ratio, Y-Z aspect ratio**: The aspect ratio can be altered by no more than 25% in either direction when compared to the average of the initial population. Again it is assumed that the initial population is representative of factors such as the shape constraints imposed by the launcher, and this constraint is designed to maintain that information.
- **Two convexity constraints**: The convexity of the shape is calculated as described by Shi, Li, and Sheng [256]. The two constraints on this serve to maintain the feasibility of ADBSat, which fails when shape is very concave, particularly on the frontal face.
- **Mesh self-intersection**: No mesh self-intersection is allowed. This constraint ensures that the shape can actually be realistically manufactured, by making sure it cannot be folded in on itself in strange ways.

Finally, the initial population of the optimisation algorithm has been carefully considered. The size and genetic diversity of the population is extremely significant to the success of the GA, but having too large a population will increase the runtime of the algorithm past reasonable limits. Due to the large number of optimisation variables (in other words, control points on the shape) it was decided not to generate the initial population randomly, as is normally done for the GA, but instead create it by hand. This also serves to employ the expertise of the design engineer in creating good quality satellite shapes. As we are assuming a good quality initial population, the population size was a lot smaller than that normally recommended (20 times the number of optimisation variables). The stringent constraints employed also justify the decision of a small starting population, particularly that of no shape self-intersection, which heavily limits where each control point can be placed in the simulation domain. Additionally, the outputs of single-objective optimisation in each objective will be added to the initial population for the multi-objective optimisation, in order to improve the quality of the initial population.

The optimised shapes found by the algorithm are thoroughly discussed in the following chapter.

Chapter 7

Optimisation Results

7.1 Atmospheric conditions and GSIM parameters

Combinations of atmospheric parameters were chosen to perform the simulations which best replicated the conditions found in VLEO. An overview of all the cases considered can be seen in table 7.1. All atmospheric modelling was performed using the extensively documented and tested NRLMSISE-00 model in MATLAB [41]. Medium solar activity levels as defined by the $F_{10.7}$ solar index were chosen for all simulations, as they are the most general. In this case, 81-day average $F_{10.7} = 138.1$ and daily $F_{10.7} = 121.7$. A_p magnetic indices span across the range 2.9 to 9.0.

Case	GSIM	Altitude (km)	α_E	ρ_{atm} (g cm^{-3})	T_{atm} (K)
1	Sentman	200	1.0	2.52×10^{-10}	869
2	Sentman	300	0.976	1.99×10^{-11}	942
3	Sentman	400	0.871	2.76×10^{-12}	948
4	Sentman	500	0.544	4.73×10^{-13}	949

Table 7.1. An overview of the atmospheric conditions and GSIM parameters considered.

Furthermore, the objectives listed in the previous chapter will be grouped into three optimisation lists for analysis. These reflect an objective list for for aerostability, an extended objective list for aerostability that attempts to counteract some shortfalls of the program outlined in 7.2, and an objective list for neutral stability. These will be referred to throughout the rest of this chapter by the names in bold, and are defined as follows:

- **Aerostability:** $C_d A_{ref}$, $-v$, Δ_{PG}, Ω_{m_α}, Ω_{n_β}
- **Aerostability-extended:** $C_d A_{ref}$, $-v$, Δ_{PG}, Ω_{m_α}, Ω_{n_β}, $|\Delta_{PG_y}|$, $|\Delta_{PG_z}|$
- **Neutral stability:** $C_d A_{ref}$, $-v$, $|\Delta_{PG}|$, $|\Omega_{m_\alpha}|$, $|\Omega_{n_\beta}|$

All constraints were used as described in the previous chapter for all cases.

7.2 Frontal face

The FF problem optimised using the aerostability and aerostability-extended objective lists, for case 1 of table 7.1. For the aerostability case, the spread in objective values of the initial population (InitPop), single-objective optimisation (SO) and multi-objective optimisation (MO) can be seen in fig. 7.1. A point indicates the mean of the population, a solid bar on a point indicates the standard deviation on the mean σ, and the dotted line the minimum and maximum range over the whole set. SO results for all objectives were grouped together and treated as one set for the purposes of this analysis.

For aerostability, is clear that both SO and MO improve on the search space spanned by the initial population, showing larger σ and a larger minimum-maximum range. The initial set has a small variance in objective values relative to the SO and MO sets. Thus, even with a carefully designed set of initial shapes, it is clear that employing a search algorithm greatly improves exploration over what an engineer is able to achieve through manual design. This trend is identified across all the cases devised in this work, and can be seen in figures in later sections.

The two controllability parameters, Ω_{m_α}, Ω_{n_β}, show a significant improvement. Where the initial set was centred near zero and showed variation both above and below, the SO and MO case are able to push both the mean and minimum value well below zero. While some shapes still exhibit positive values of Ω_{m_α} and Ω_{n_β} and thus poor controllability, the ability of the algorithm to find better, more controllable configurations than just using manual design is apparent.

There is also a clear improvement in volume. Both MO and SO cases are able to push the mean volume of the set above the maximum of the initial population, with the maximum increasing significantly. However, this comes at the cost of $C_d A_{ref}$, which also increases significantly, although retaining a range that encompasses the initial population. This suggests that $C_d A_{ref}$ is in direct competition not only with volume, but also with Ω_{m_α} and Ω_{n_β}. Because there are three objectives which can be improved at the cost of one, the algorithm sacrifices $C_d A_{ref}$ for the increase in the other three. This suggests that while having a low-drag configuration is important, neglecting to consider other realistic mission needs can result in a false confidence in the suitability of certain geometries for certain purposes.

A clear improvement in Δ_{PG} is observed, with results from both MO and SO cases spanning into the desirable negative values. It is clear that the algorithm can identify cases in which the CoG is in front of the CoP. However, analysis of results shows that this result is deceptive. For most objects, the front-facing plates contribute the majority of drag to the calculation of C_d, having the highest drag force exerted on them by the atmosphere. In particular, if the frontal face is larger than the fixed cuboid body, the shading algorithm considers the body shaded and its contribution to drag is nullified. Consequently,

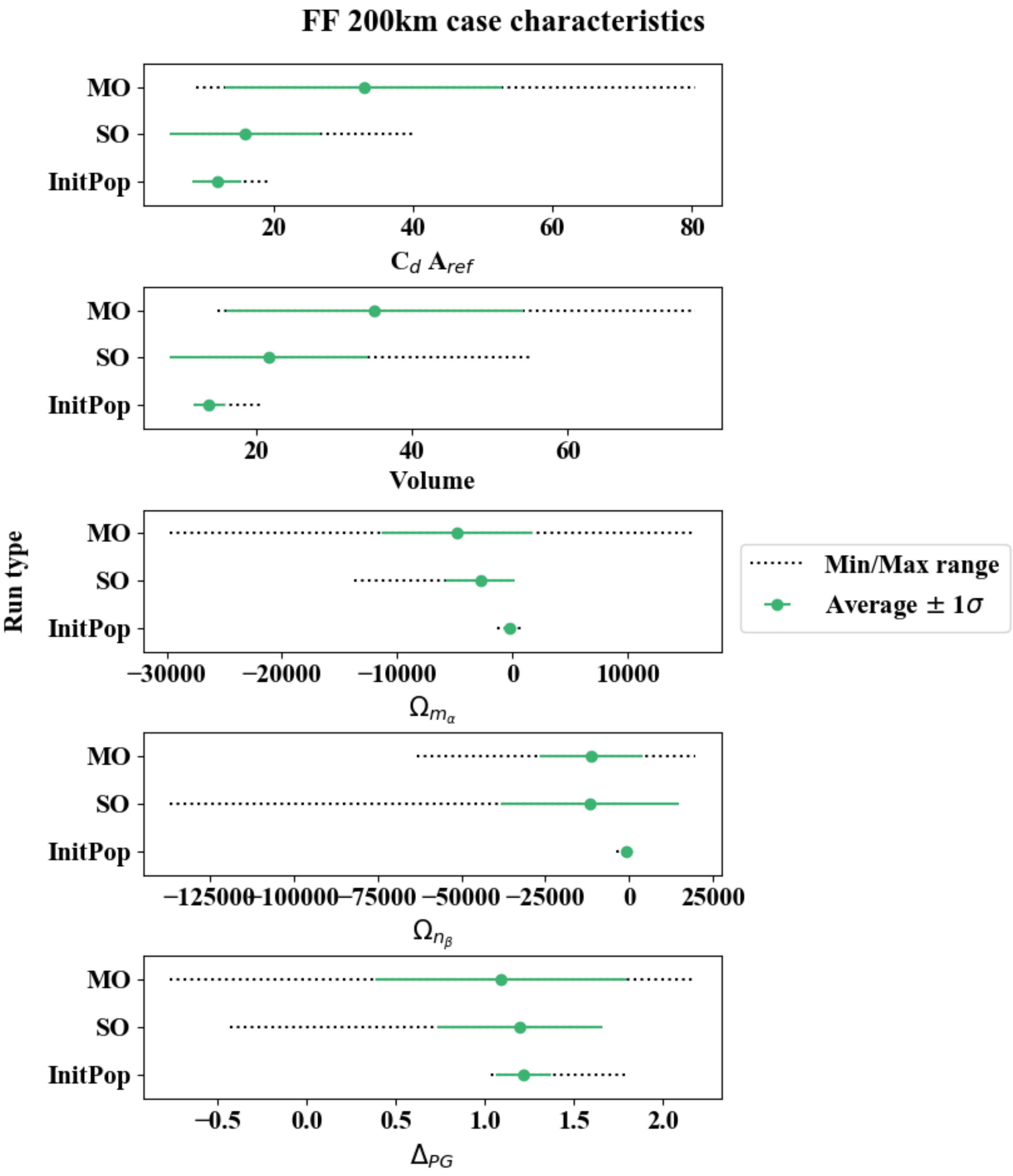
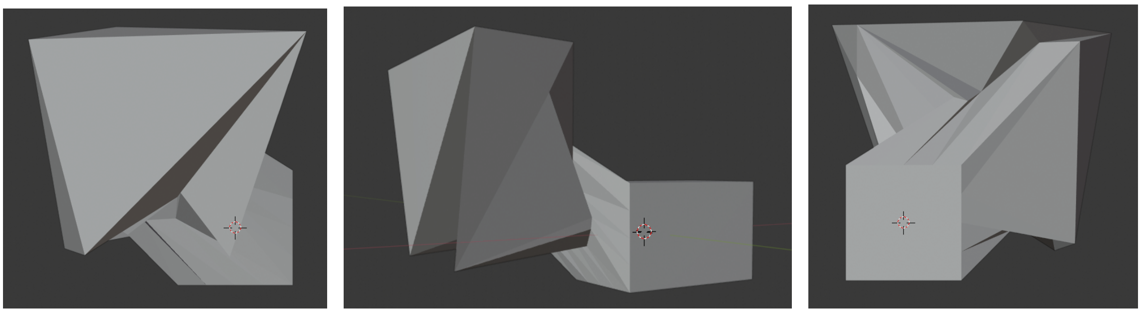


Figure 7.1. Objective values of the initial population, SO, and MO, for the FF case at 200 km.

the CoP is coincident with the frontal face in almost all cases. However, by sacrificing Ω_{m_α} and Ω_{n_β}, the algorithm is able to push the frontal section of the object to one side. This reveals the cuboid body, and a front-facing surface connecting the cuboid body to the optimisation vertices. As this lies further back along the x axis compared to the frontal face, the CoP is pushed backwards, behind the CoG. Thus, Δ_{PG} becomes negative, as desired. However, the resulting shape is infeasible for any real satellite mission, as seen in fig. 7.2.

The aerostability-extended case solves this deceptive Δ_{PG} problem, as the two additional objectives effectively restrict the CoG and CoP to being along the same line, parallel to the x axis. This serves to prevent global asymmetry and preclude the possibil-



(a) Front view (b) Side-on view (c) Back view

Figure 7.2. A shape representative of those for which $\Delta_{PG} < 0$ for the FF case. The asymmetry which pushes the CoP backwards can be seen.

ity of moving the frontal section of the object to the side in the manner seen in fig. 7.2. Thus, it was worthwhile to also run the FF case for the aerostability-extended objective list. The same initial population was used for the MO case as for the aerostability case. The results are shown in fig. 7.3. For C_dA_{ref}, v, Ω_{m_α} and Ω_{n_β}, similar trends are seen as for the initial aerostability MO simulation. However, aerostability-extended MO optimisation cannot decrease Δ_{PG} below zero. Δ_{PG_y} and Δ_{PG_z} can be reduced to near zero, aligning the CoP and CoG. However, when they are at their smallest, Δ_{PG} is large and positive, indicating that the CoP is far forward of the CoG and the shape exhibits poor stability.

Equation (6.24) was used to calculate d_u for the initial population, SO, and MO results. The utopia point is always that of the MO simulation. The distribution of this distance for $\min(d_u) < d_u < 1$, in bins of 0.05, is shown in fig. 7.4. Any points for which $d_u > 1$ are not considered in this analysis, and the total rejected points from each population is also specified in fig. 7.4.

The efficiency of MO optimisation in finding solutions closer to the utopia point than the initial population or SO can clearly be seen. It is the only case which populates the regions of the histogram with the smallest distances, of $0 < d_u < 0.25$. While SO optimisation is also a clear improvement over the initial population, its inability to consider balanced solutions means that it cannot reach the utopia point in the manner of the MO simulation. The balance between objectives captured by the MO clearly adds important additional information about the quality of the overall satellite body in comparison to the SO cases. Of note is that the MO case used the outputs of the SO case as part of its initial population, thus starting with a higher quality population case than the SO, leading to better results for some objectives than SO optimisation. Additionally, because of the inherent randomness present in all metaheuristics, it is not guaranteed that the SO case finds the absolute optimum, and thus it is possible that the MO optimisation improves on an objective when compared to the SO case, especially given its better starting population.

Examining the individual cases reveals that the algorithm favours flat faces. Four rep-

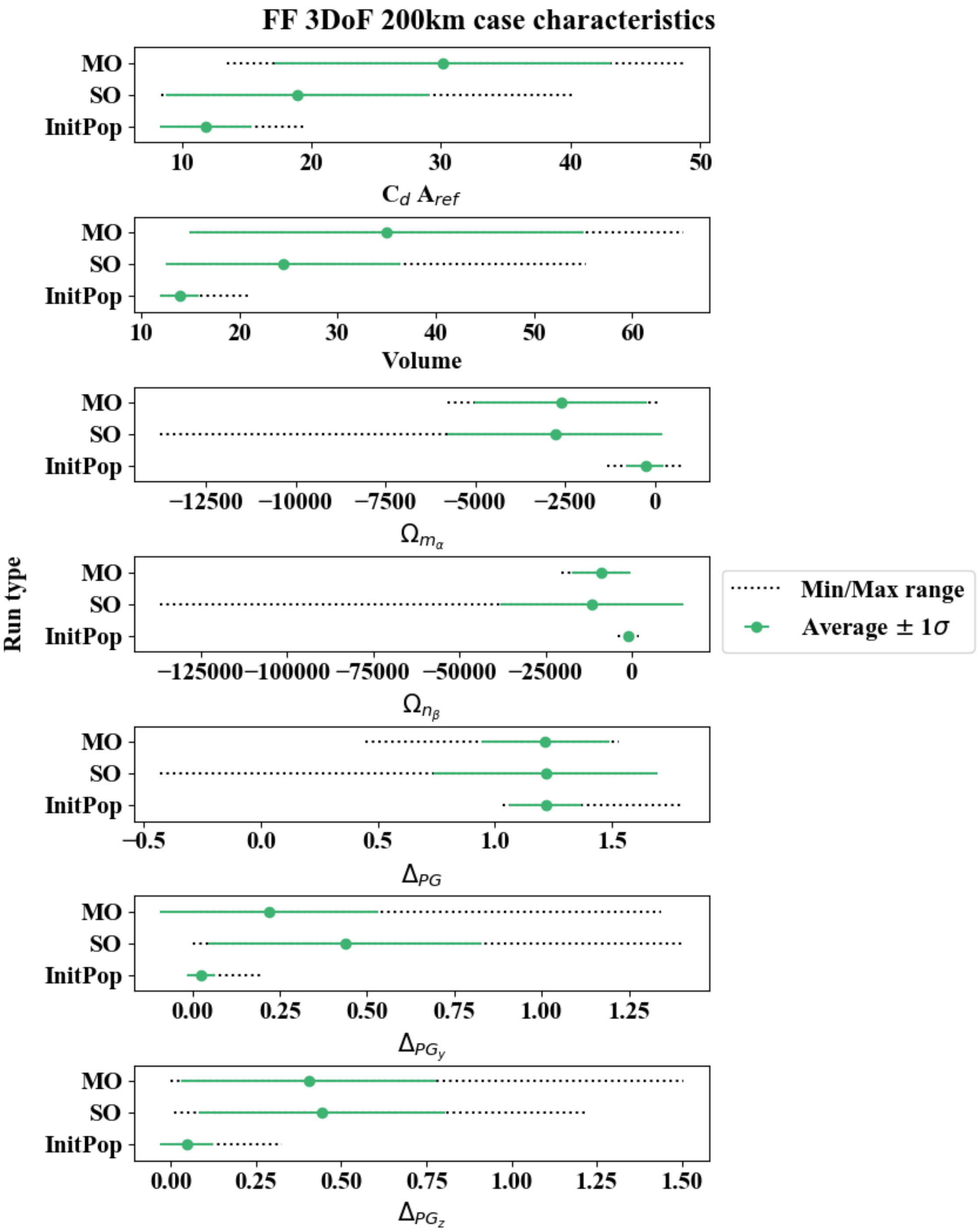


Figure 7.3. Objective values for the seven-objective FF case at 200 km orbital altitude.

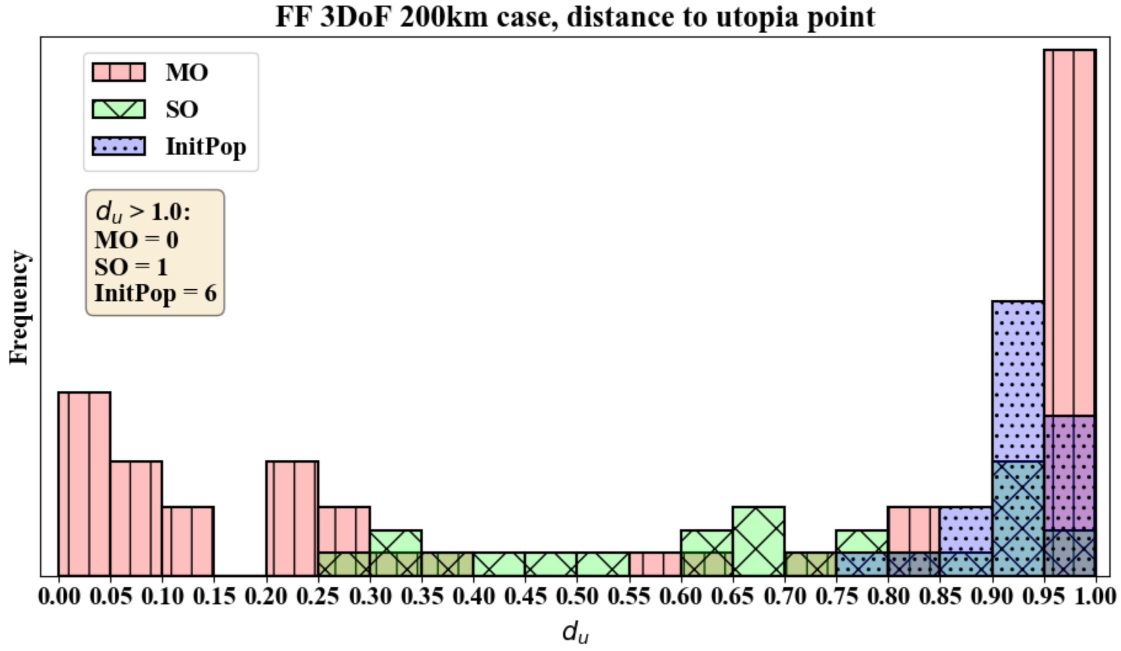
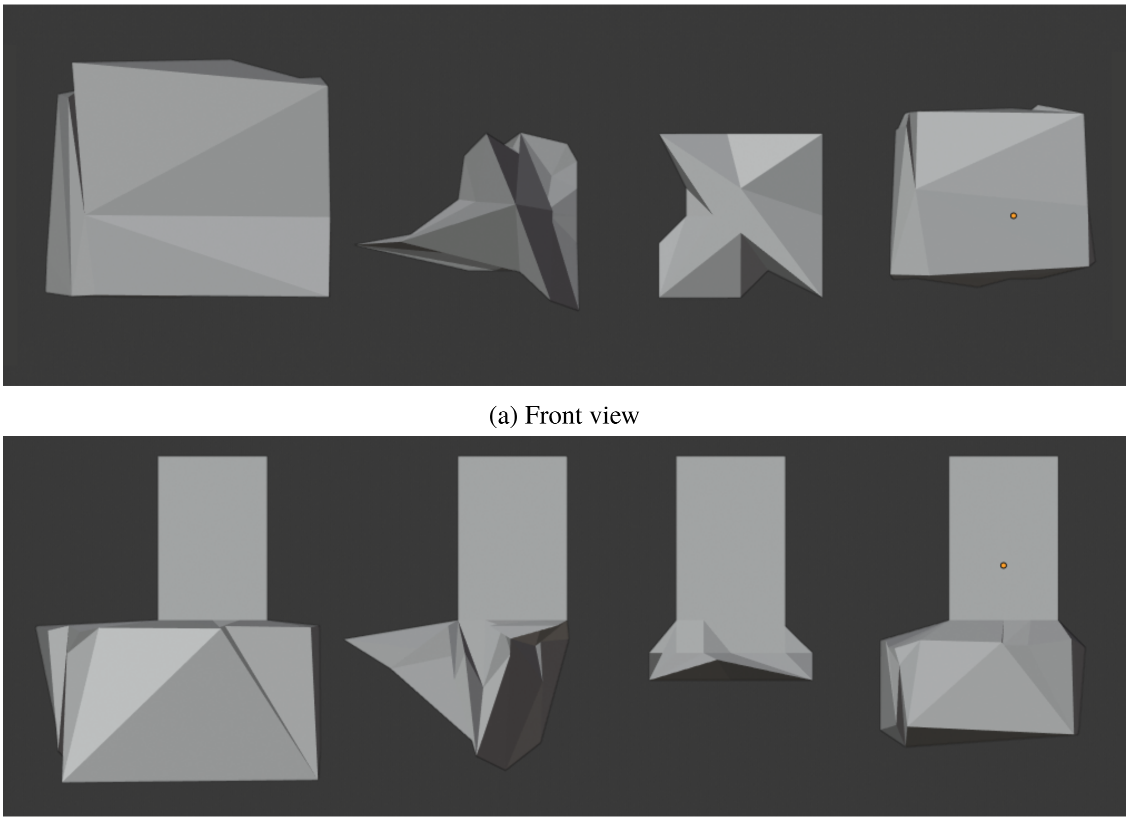


Figure 7.4. Distribution of distance of candidate solutions to the utopia point for the initial population as well as SO and MO cases.



(a) Front view

(b) Side-on view

Figure 7.5. Four shapes spanning a range of objective value combinations in the seven-objective MO.

resentative shapes spanning a large range of all objective values can be seen in fig. 7.5. Some asymmetry is seen, resulting from the program struggling to minimise the Δ_{PG} objective. There are also high-profile pointed features which protrude in the y and z axes, which can be interpreted as the algorithm attempting to place stabilising fins on the spacecraft. Fins on side and rear faces are often seen on spacecraft in VLEO [31], [32], [253], exerting strong effects on spacecraft behaviour. Although the algorithm is restricted only to the front face of the body, it attempts to place these features as far back as possible. However, it is apparent that in considering only the front face of the body it is impossible to fulfil all objectives. An incomplete picture of the characteristics of the body is painted. The limitations of considering only the front face of the body are clear. While this analysis is helpful and provides valuable insight into the spacecraft's characteristics, it ultimately paints an incomplete picture of the behaviour of the satellite as a whole.

7.3 Tail face

The TF case was optimised for the aerostablity objective list for case 1 of table 7.1. Characteristics for the initial population, SO and MO cases can be seen in fig. 7.6.

It can be seen that the minimum of $C_d A_{ref}$ found by the SO also exists in the initial population. This indicates that the lowest-drag shape was identified by the designer and included in the starting set. Upon examination, it is found that the lowest-drag shape is that which has no tail at all. It is the smallest possible shape given the configuration, with the points as close to the body as possible. This shape, however, does not exist on the Pareto front of the MO case. $C_d A_{ref}$ is again seen to increase compared to the initial population and MO cases. Thus, it is clear that although theoretically, reducing drag is advantageous, its absolute reduction is infeasible in the context of other important body characteristics.

An interesting characteristic is that the maximum feasible volume for both MO and SO cases is smaller than the volume of some shapes in the initial population. This indicates that the geometries which exceeded these volumes were infeasible in the constraints imposed. Most likely this is due to the aspect ratio constraints, which aim to maintain the feasibility of the shape for a particular launcher. As the front, pointed face protrudes from the body, it would seem that the algorithm cannot push the tail surface back to add more volume without making the shape too long for the launcher. Thus, it is successfully able to identify feasible shapes, but the constraints prevent much improvement.

It appears that the tail geometry has a much smaller effect on minimising Ω_{m_α} and Ω_{n_β} than the front geometry. The initial population of the TF case exhibits desirable values of $\Omega_{m_\alpha}, \Omega_{n_\beta} < 0$ for all cases, while for the FF case these values are either side of zero. However, the average and minimum values of the SO and MO for both objectives are more negative for the FF case than the TF case. Additionally, the TF case shows a smaller σ than FF, in other words a smaller variation was achieved through changes to the tail geometry than to the frontal face. As the initial geometries are functionally the same except flipped with respect to the direction of the flow, it is clear that these parameters are more sensitive to the frontal face geometry.

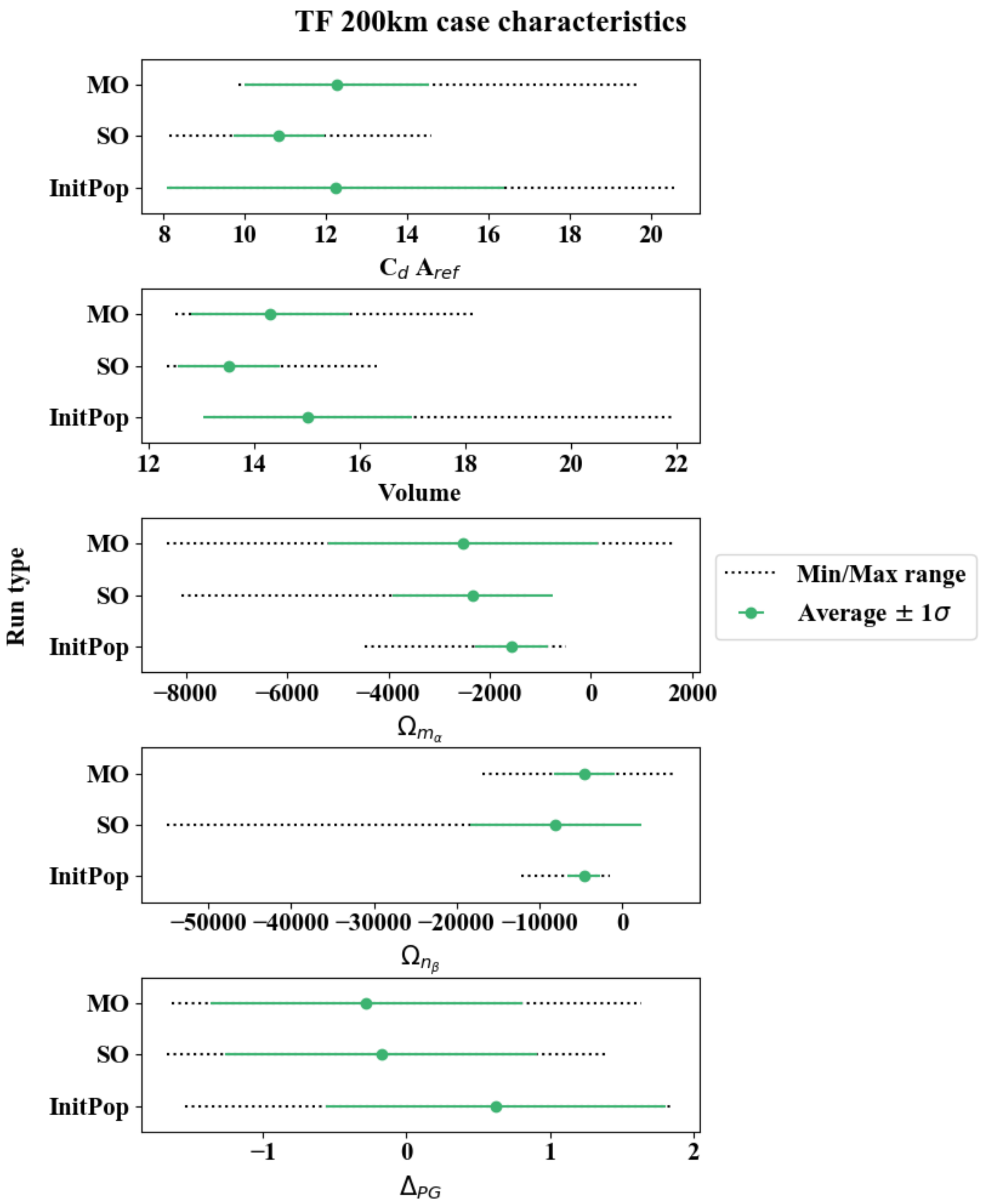


Figure 7.6. Objective values for the TF case, optimising five-objectives at 200 km.

However, tail geometry dominates in controlling the value of Δ_{PG}. While all FF cases that exhibit the desirable $\Delta_{PG} < 0$ are infeasible as explained above, most TF candidate solutions exhibit this without ”cheating“. The average of both SO and MO cases is negative, with the minimum more negative than that of the FF cases. This is consistent with the way the CoP is calculated, which relies on the barycentres of the triangular plates. For tail-less objects, the frontal faces contribute the most to the drag, and the CoP is functionally on the frontal face. In being given control of the tail, the algorithm is able to develop fin-like features which move the CoP towards the back of the body. Indeed, even the smallest protruding features have this desirable effect.

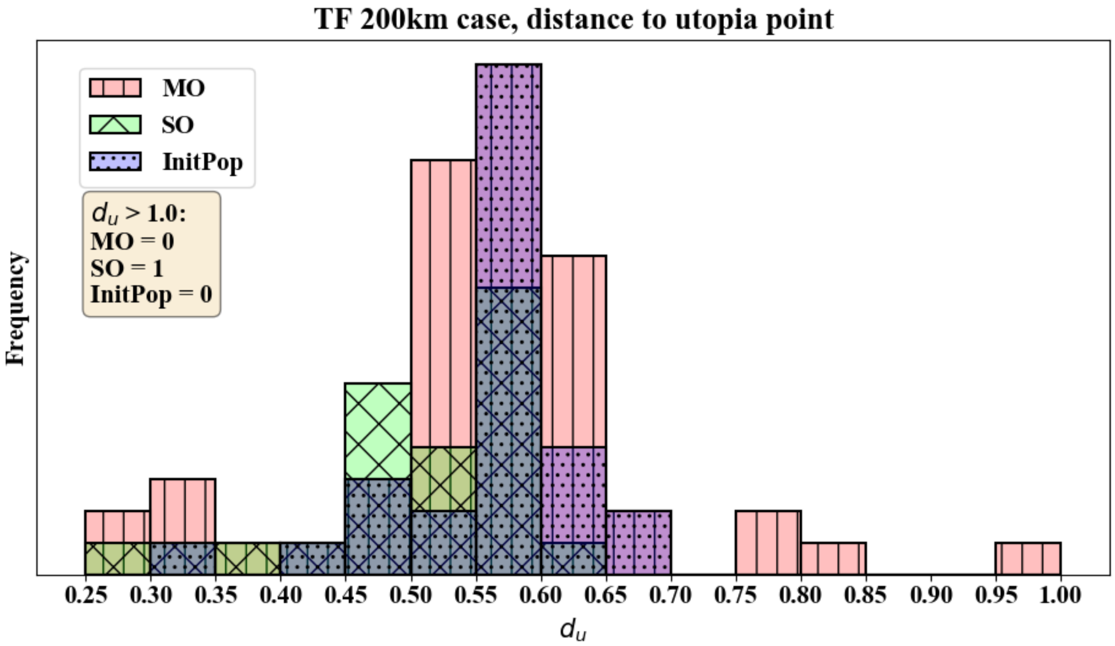


Figure 7.7. Distances to the utopia point for the three sets, for the TF case at 200 km.

Interestingly, fig. 7.7 shows that the initial population and SO is much closer to the utopia point than for the FF case. While the MO case does have the candidate solution with the smallest distance, an SO solution which minimises drag by having a flat tail comes close. This indicates that tail geometry is easier to manually design such that it achieves the desired characteristics, and that it is more simplistic in nature than the front face geometry.

The algorithm's lack of control over the front face means that while it can send the CoP towards the back of the object, it is incapable of adding mass to the front and moving the CoG forwards. Increasing the volume by pushing the rear surface back not only increases drag, but also moves the CoG backwards, worsening Δ_{PG}. The algorithm considers that the benefits of increasing volume do not outweigh the negative effects. Therefore, shapes are favoured which show a flat tail with a degree of protrusion, as in fig. 7.8. Interestingly, it is noted that symmetry is not maintained. The shapes exhibit extrusions in the positive directions of the z and y axes, but not in the negative. This is most likely due to the difference in magnitude of Ω_{m_α} and Ω_{n_β}. Ω_{n_β} is generally more negative than Ω_{m_α}, and thus it appears that the algorithm is minimising Ω_{n_β} more successfully. However, the difference in magnitude most likely stems from the factors which differ between the two calculations. Perhaps it would have been more informative to minimise only C_{m_α} and C_{n_β}, rather than Ω_{m_α} and Ω_{n_β}. Time constraints precluded this analysis, but it is hoped that it will be investigated in the future.

Because of the lesser influence of the tail geometry on the characteristics of the body, the algorithm appears to have had trouble in converging. Since geometries are closer in fitness value to each other, where for the FF case there was a clear Pareto set of superior geometries, such a set is less clear for the TF case. There is more variation in tail geometries, and they have lower convexity when compared to FF results. However, it is

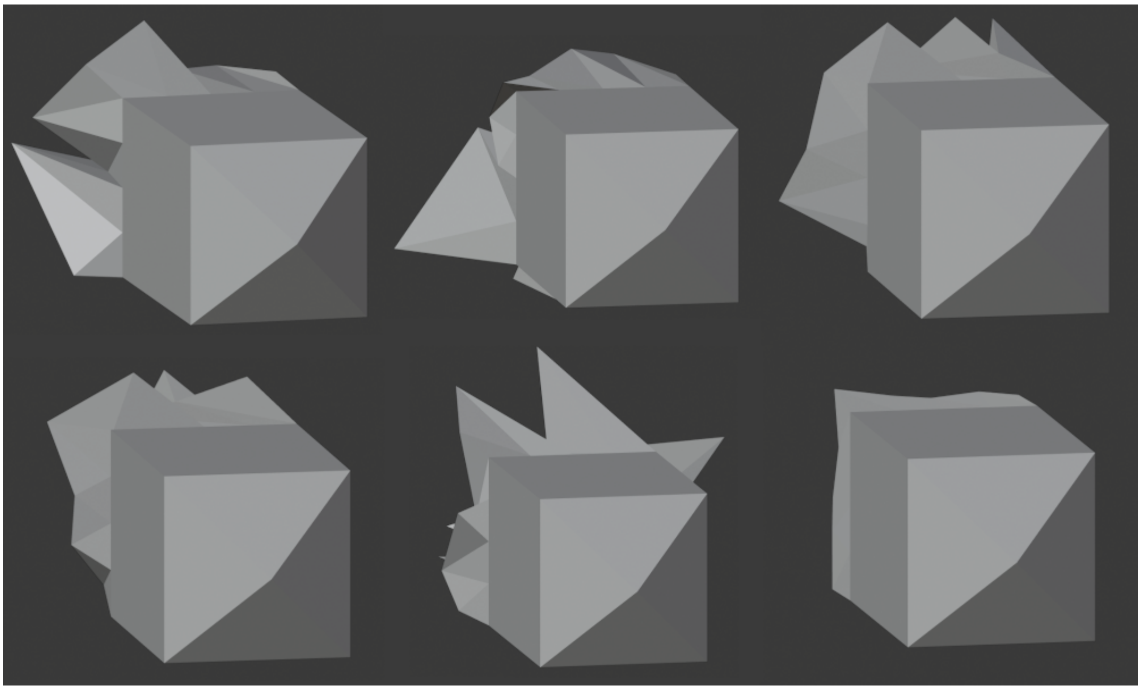


Figure 7.8. The shapes favoured by the optimisation algorithm for the TF case.

clear that the algorithm has explored the search space beyond the initial designs. This suggests that while the algorithm's exploration is good, and affords us some insight into the influence of tail geometries on the overall characteristics of the body, computational optimisation may not be the best way to design tail geometries in an isolated manner.

7.4 Front and tail

The F&T case was examined for the aerostability and neutral stability objectives lists for case 1 of table 7.1. It is hoped that by analysing both the frontal and tail geometry at the same time, a more thorough overview of the characteristics of the body can be obtained. The characteristics of the sample for the aerostability objectives list can be seen in fig. 7.9. The trends seen across the five objectives show many of the benefits of the FF and TF case, while not suffering from as many detriments. Broadly, the benefits seen are the algorithm's exploration of more of the search space than the initial population, improvements in v, Ω_{m_α} Ω_{n_β} and Δ_{PG}, a more negative distribution of Δ_{PG}, and a population closer to the utopia point as seen in fig. 7.10. As with the FF and TF cases, $C_d A_{ref}$ has been sacrificed in order to facilitate improvements in the other four objectives. This reinforces the finding that single-objective drag optimisation does not provide as significant an improvement on satellite characteristics as was previously thought, and oftentimes can be detrimental with respect to the other objectives.

However, here the variation which plagued the tail-only cases has disappeared. Instead, we see consistent results across the Pareto set, with the algorithm favouring shapes as in fig. 7.11. These shapes all have a frontal face with a pointed feature and edges extruded in the direction of the flow, which is not in the Pareto set of the FF solution as shown in fig. 7.5. The algorithm has found different solutions for the F&T case than for the FF case. Additionally, the fluctuation plaguing the tail case has disappeared - the tail is now a flat, roughly square face which protrudes past the frontal face of the body in the x and y axes. It is also now exhibiting approximate symmetry in both axes, which was not seen in the individual TF case. While the individual TF case has a large range of solutions and weak convergence, the F&T case has strongly converged under the given conditions.

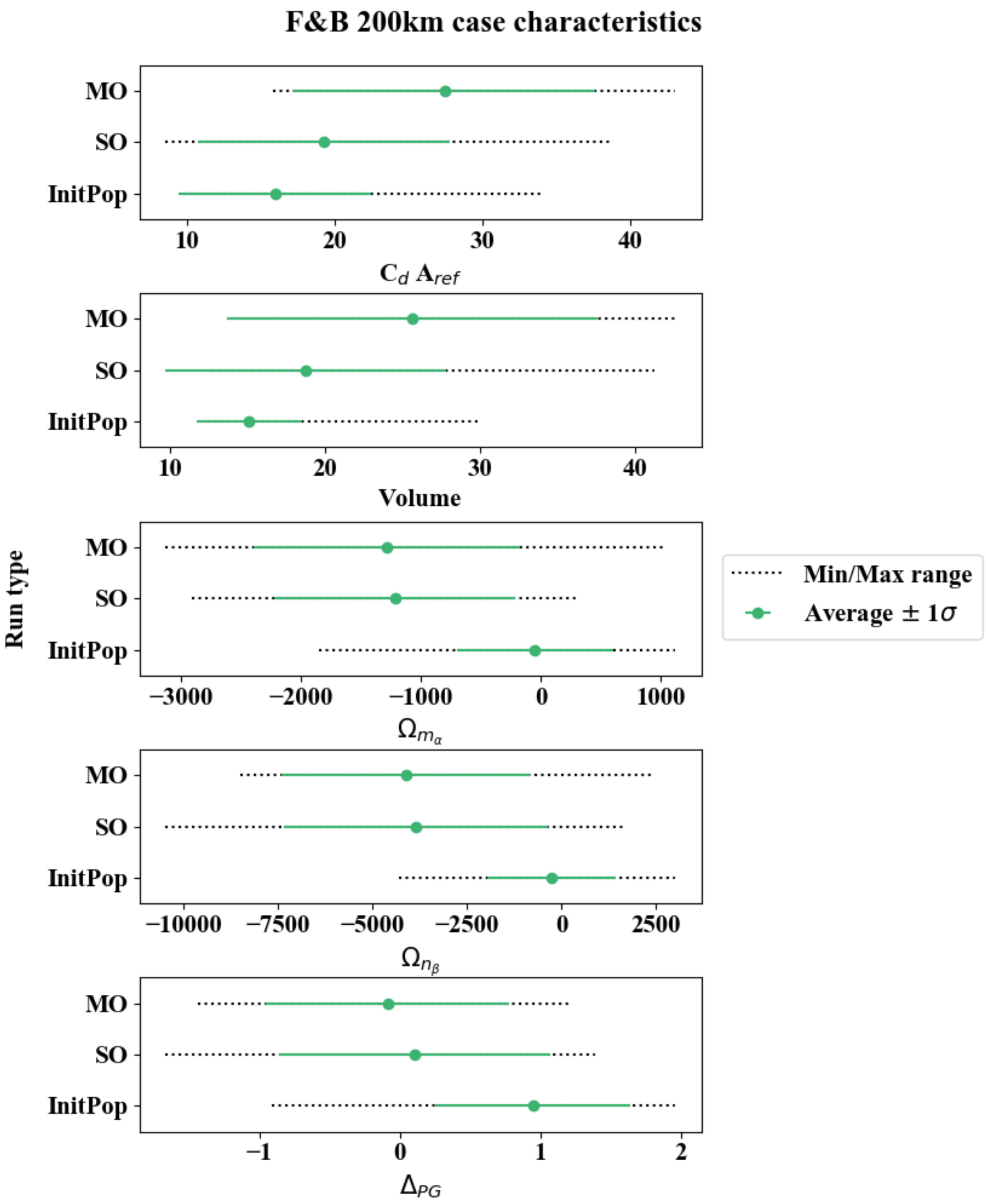


Figure 7.9. F&T case characteristics, aiming for aerostability at 200 km orbital altitude.

Thus, it can be argued that optimising the front and tail separately, and joining the two solutions together, is not representative of the behaviour of the body as a whole. Opti-

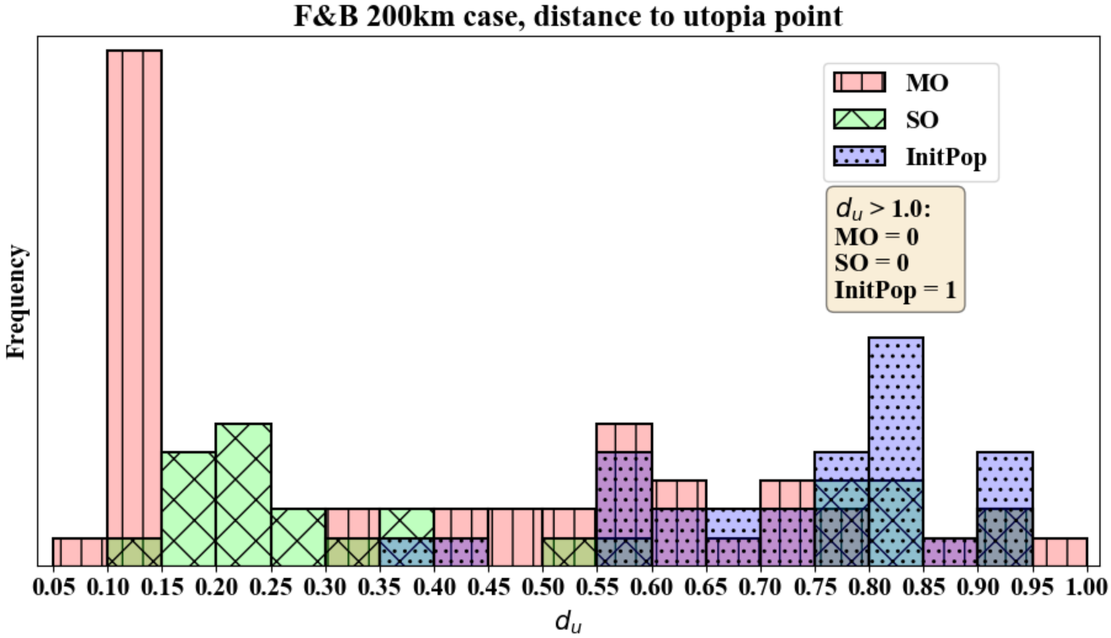


Figure 7.10. F&T distances to the utopia point, for aerostability at 200 km orbital altitude.

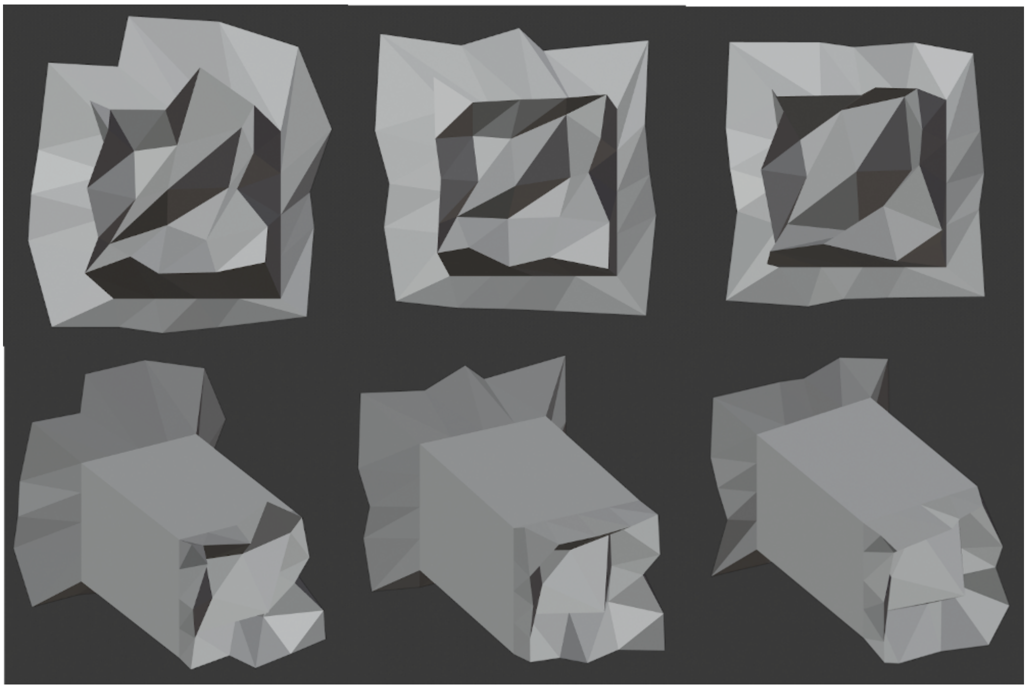


Figure 7.11. Shapes favoured by the algorithm when optimising the F&T case.

mising the entire spacecraft better considers the overall properties. While separate TF and FF optimisation can lead to some useful observations, it is more so an informative exercise, rather than being representative of the final best spacecraft body. This conclusion, alongside the observation that single-objective optimisation can often be detrimental to the other attributes of the body, is important in the face of the prevailing literature which examines individual features and objectives separately, but draws conclusions on the performance of the spacecraft as a whole. For validity, conclusions drawn from such literature must be contextualised.

Neutral stability was also examined for the F&T case. The graphs of the distribution of initial population, SO and MO across the five objectives, as well as the distances to the utopia point, are shown in Appendix A. They results follow the same trends as, and closely align with, the solutions of the WB case. However, the utopia point of the WB case is more advantageous in the individual values, indicating a better overall performance of the algorithm on the whole body than on just the front and tail surfaces. This is in line with the conclusion above that in general, analysing isolated elements of the body gives more constricted results than analysing the body as a whole. Thus, the results detailed below for the WB case for neutral stability also apply to the F&T case, while improving on the absolute value of the objectives.

7.5 Whole body

The WB case was examined for all four cases in table 7.1, for the aerostability objectives list. It was also examined for case 1 of table 7.1 for the neutral stability objectives list.

For case 1 and aiming for aerostability, similar trends are seen in fig. 7.12 when comparing initial and final geometries as those for the other optimisation cases. Broadly, for most candidate solutions $C_d A_{ref}$ is sacrificed to facilitate a desirable increase in v, Ω_{m_α} and Ω_{n_β}. Some solutions still exhibit low drag, but they tend to be poor in other objectives. The algorithm is again seen to better explore the search space than the initial, manually-created population. It is also clear once more that considering only $C_d A_{ref}$ leads to an incomplete picture of the spacecraft behaviour. While it may seem advantageous to lower $C_d A_{ref}$, it generally leads to poor spacecraft performance in other areas which may compromise overall mission efficiency.

The distribution of distances to the utopia point for the three populations is seen in fig. 7.13. As before, both SO and MO cases decrease the distance significantly when compared to the initial population. Explicitly, this proves the algorithm's ability to effectively explore the search space. This would undoubtedly afford significant benefits when used in addition to engineering expertise in the design of a spacecraft.

Results for cases 2 to 4 of table 7.1 closely follow those seen for case 1. The relevant population characteristics and distance histograms can be seen in Appendix A. The numerical values of the objectives which directly depend on atmospheric parameters, $C_d A_{ref}$, Ω_{m_α} and Ω_{n_β}, vary across altitudes. An increase in magnitude is observed for $C_d A_{ref}$ and a decrease in magnitude for Ω_{m_α} and Ω_{n_β}. However, the relative positions of the means of the population, along with the magnitude of σ in comparison to the minimum-maximum range, are the same across all altitudes. The only exception is seen in the MO case for 200 km (case 1) when looking at Ω_{m_α}, where the algorithm was able to find a better minimum value when compared to SO optimisation than for other altitudes. However, the standard deviation and relative position of the mean is very similar to those for

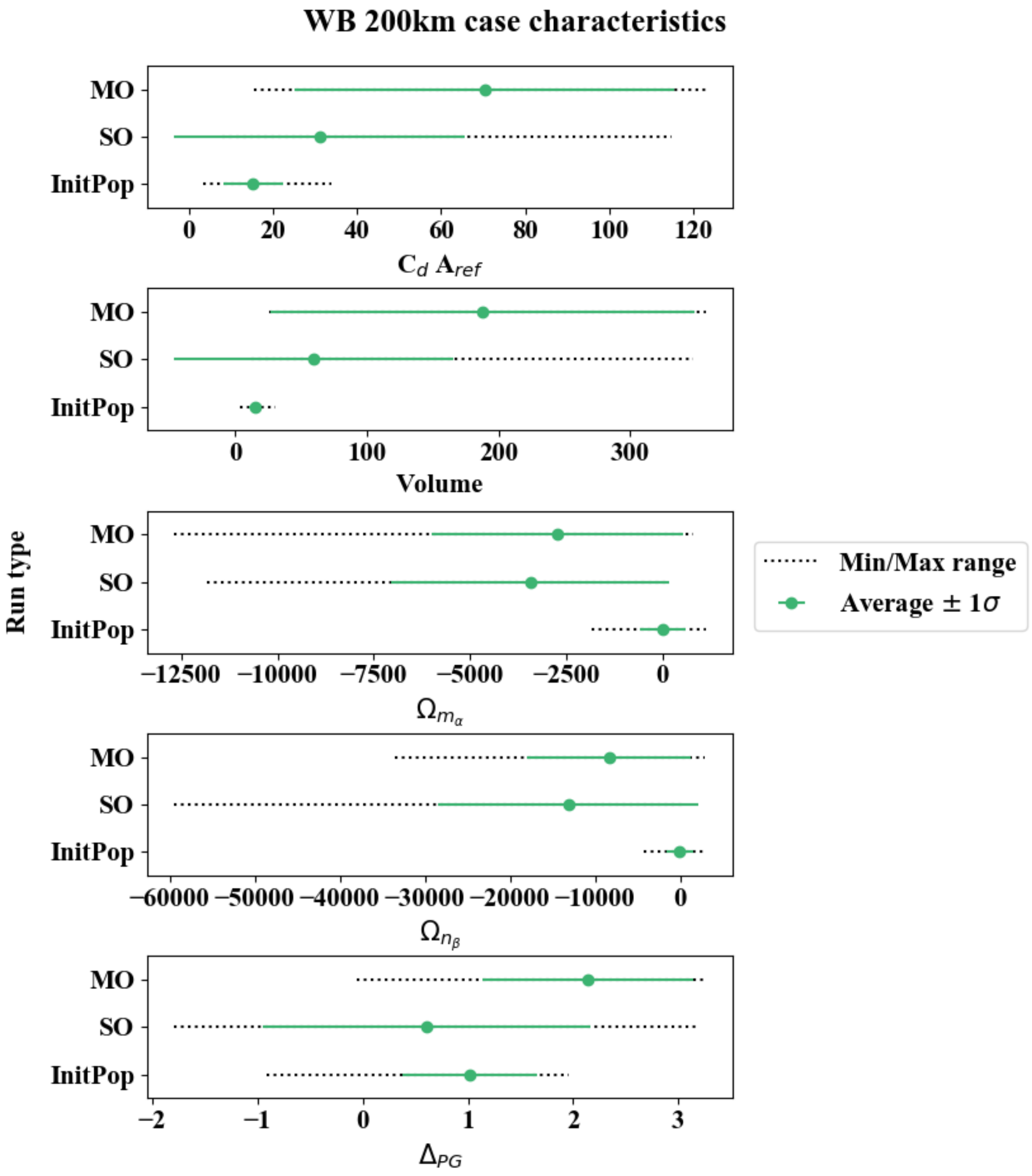


Figure 7.12. The characteristics of the WB case at 200 km aiming for aerostability. The trends seen are also representative of the results at 300 km, 400 km and 500 km.

cases 2 to 4.

As they are not dependent on atmospheric parameters, the magnitudes of v and Δ_{PG} do not vary with altitude. As expected, the population characteristics across these two objectives are identical for all altitudes. The distribution of distances across the four altitudes are also very similar, with the initial populations exhibiting $d_u > 0.7$. SO results approach the minimum distance, while MO results are the only ones to attain it. Manual examination of the results shows that geometries across all altitudes are much the same. This observed similarity indicates that the overall trends in optimisation are not dependent on altitude, although some parameters may change in magnitude. This is consistent with the formulation of the algorithm, which only seeks to minimise candidate solutions

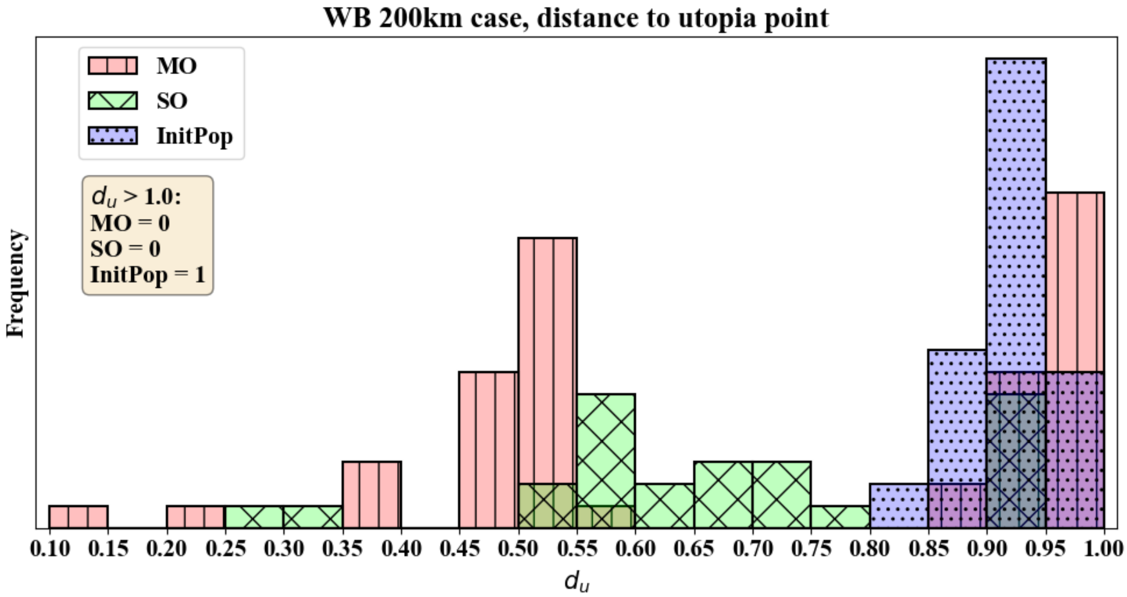
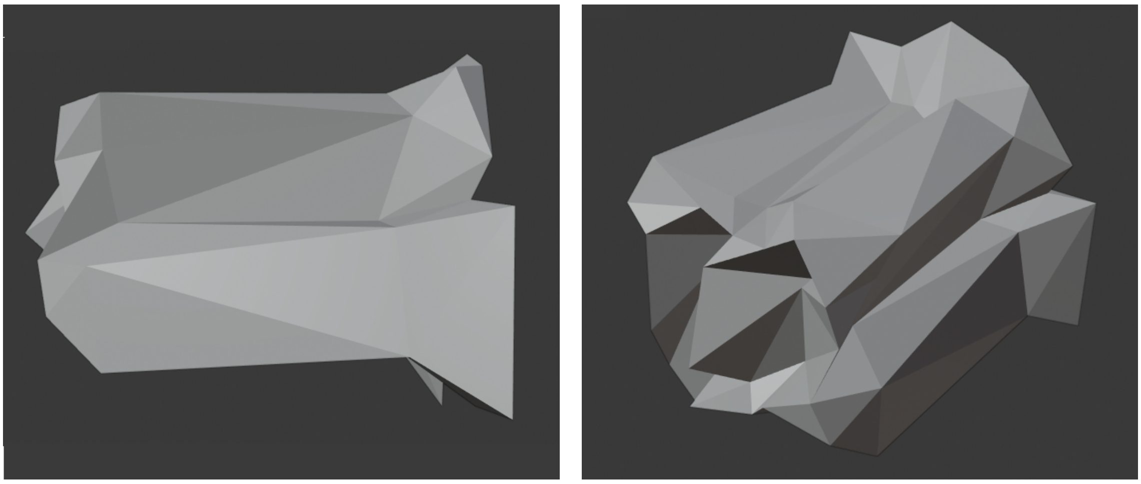


Figure 7.13. The distribution of distances to the utopia point of the WB case at 200 km.



(a) Side-on view

(b) Diagonal view

Figure 7.14. The candidate solution for which $\Delta_{PG} < 0$.

to the best of its ability, without regard for the absolute value of the individual objective values. Consequently, the results at 200 km are assumed to be representative of those for all altitudes.

The algorithm still struggles to find shapes for which $\Delta_{PG} < 0$. Figure 7.14 shows a the only shape for which this is the case. It exhibits a point on the frontal face, a thick cuboid body, an x-shaped cross section where the corners of the cuboid shapes are farther from the centre than the midpoint of the sides, and a skirt feature on its rear face. The cuboid body keeps the volume high, with the x-shaped cross section reducing drag by lowering frontal area. The skirt, slightly larger in dimension than the front face, pulls the CoP towards the back of the body and provide stability. This shape, with all three $\Delta_{PG}, \Omega_{m_\alpha}, \Omega_{n_\beta} < 0$, is the best among the set in terms of stability.

Two types of shapes are favoured by the algorithm, examples of which are shown in

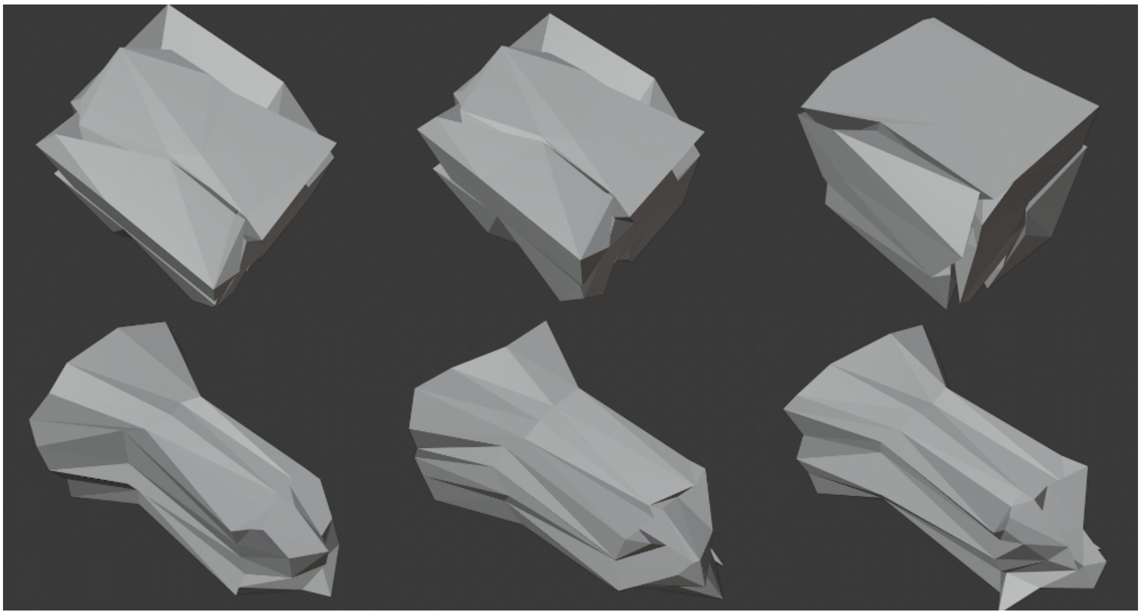


Figure 7.15. Some examples of the shapes favoured for the aerostable WB case.

fig. 7.15: large cuboid bodies with the maximum allowable dimensions and aspect ratios (top row), and elongated bodies with a degree of tapering on the frontal face and a small rear-facing skirt (bottom row). Both types of shapes exhibit good Ω_{m_α} and Ω_{n_β} and poor Δ_{PG}. Large cuboid shapes better maximise v and have slightly better Ω_{m_α} and Ω_{n_β}, while elongated shapes better minimise $C_d A_{ref}$ but have a more widely varying range of Ω_{m_α} and Ω_{n_β}, sometimes non-negative. A mission designer is then free to prioritise whichever objectives they choose, having the characteristics of the entire set available for analysis.

One salient point is that cuboid geometries make up a large proportion of the initial population. This may often be desirable, to satisfy launch, manufacturing, or transport capabilities. As the GA relies on a good quality and genetically diverse initial population to find the best solutions, finding better non-cuboid geometries may not have been possible in this case. The initial population did not exhibit a good enough genetic variation to allow the algorithm further exploration. Further simulations with a higher genetic variation, or based around a different starting shape such as a sphere or flattened saucer, could be performed as needed on a mission-by-mission basis. It is clear that the algorithm successfully captures the trade-off of objectives and provides valuable insight which cannot be gained from single-objective optimisation.

Neutral stability was also investigated, using the same initial population as the MO case for the aerostability. Results can be seen in fig. 7.16. The algorithm was able to reduce the variation of Ω_{m_α} and Ω_{n_β} dramatically to near zero, and also identified geometries for which Δ_{PG} is near zero. Drag and volume both increased, as before. As a result of using the initial population of MO case which aimed for aerostability, the distance of the initial MO population to the utopia point is in general very large. All but two individuals in the initial population and SO case have $d_u > 1.0$, while the entire MO population exhibits $d_u < 1.0$ Histograms for this case are included in Appendix A for interest. The algorithm succeeds in finding a utopia point which is close to the requirements, and lowers the distance of the solution set to this point drastically. The utility of the algorithm in identifying shapes which better match the requirements is clear.

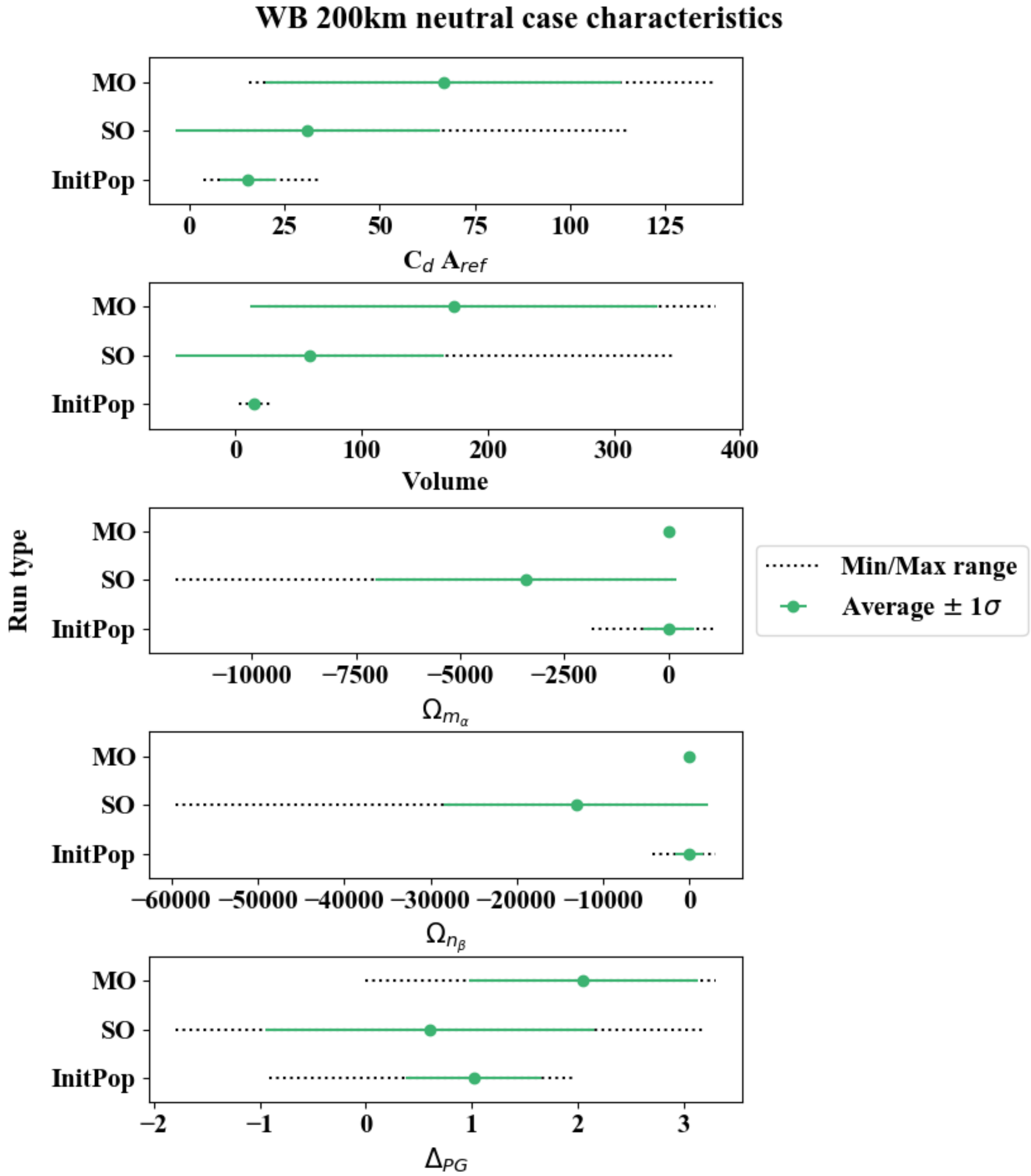


Figure 7.16. The characteristics of the WB case at 200 km aiming for neutral stability.

Figure 7.17 shows the shape favoured by the optimisation algorithm when using the neutral stability objectives list, with front-on views in the top row, and side-on views in the bottom row of the same shapes above. While these may seem similar to the bottom row of fig. 7.15, one key difference is the skirt flare, here defined by a ratio of maximum body width to maximum skirt width, r_{bs}. For aerostable objects with a skirt, $r_{bs} \sim 0.6$. Neutral stability has $r_{bs} \sim 0.4$, indicating a larger skirt width when compared to the body. The larger plate area towards the back of the object pulls the CoP backwards and

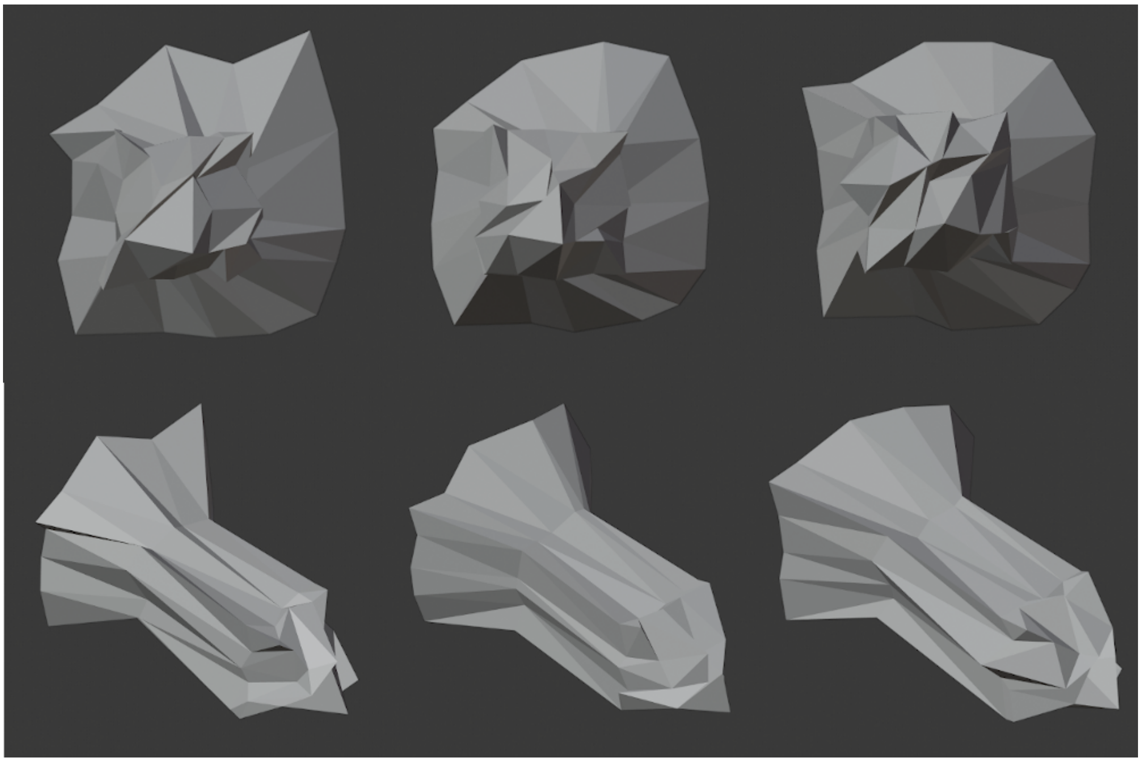


Figure 7.17. Shapes representative of the neutral stability optimisation results.

places it on top of the CoG. It also plays an important part in the magnitude of Ω_{m_α} and Ω_{n_β}, as it means that A_{ref}, significant in the calculation of the two derivatives, is very similar for the small angle approximation over which the derivative is calculated. Thus, the derivative becomes small.

Computational optimisation is particularly useful in this case. Unlike volume, and to a certain extent drag, stability parameters are difficult to control intuitively when designing spacecraft geometries by hand. Where increasing the size of an object will certainly increase its volume and most likely increase its $C_d A_{ref}$, the consequences of altering the geometry on the stability parameters are more difficult to understand. Here, where the aim is not to increase stability but to obtain a geometry that is neutral to attitude changes, it is particularly difficult to determine what a particular operation will do to a satellite body. By applying computational optimisation, the designer can save time and decrease frustration by allowing the algorithm to directly target the desired results in a systematic way. It will also more quickly disregard unsuitable designs which may have otherwise been considered. The final results will, of course, need expert refinement, but the algorithm provides additional information to the expert and helps with narrowing down candidates.

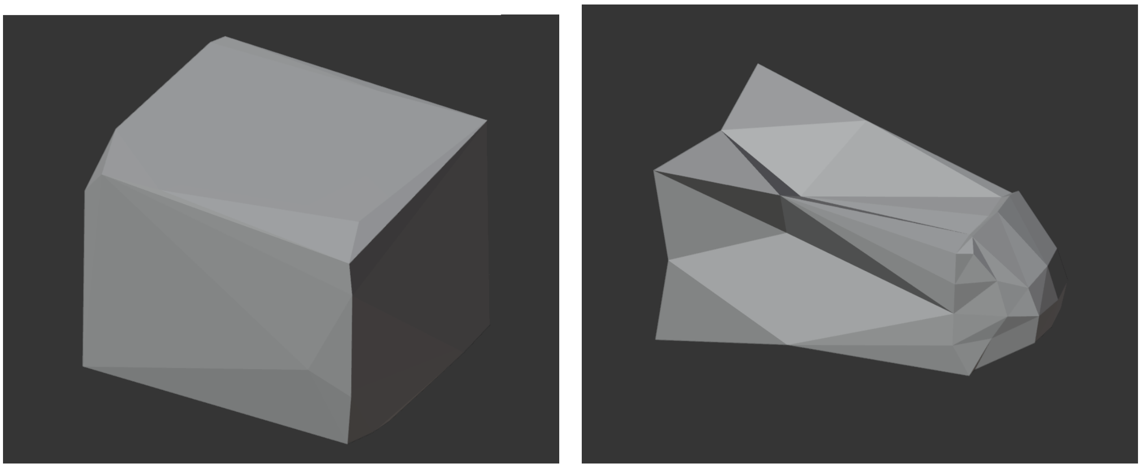
7.6 Refined results

In order to quantify the improvement of the optimised results over the initial set, some designs from the WB Pareto set optimised for 200 km altitude, aiming for aerostability, were selected for streamlining. The aim was to increase manufacturability compared to the initial more rough results. This also allows more realistic designs originating from the results to be compared to the initial set. In other words, this analysis provides a more realistic view of the performance of useful results against the initial set.

It is worth noting that the Pareto set itself is one of the benefits of using the GA for optimisation. By producing a set of solutions within the given parameters, the program allows the user to examine the trade-off between the different objectives. This is by design, as the user can then decide which objectives are important and which can be sacrificed. The range of solutions is one of the strengths of this method, not a drawback.

That having been established, in order to show the usefulness of the method, some choices have been made which encapsulate the extremes of this range for demonstrative purposes. One low-drag and one high-drag case have been chosen to be further streamlined and examined against the initial set. These are representative of each of the two types favoured by the algorithm, shown in fig. 7.15. Manual refinement was performed using Blender, with the aim of increasing shape smoothness and continuity across surfaces. Notably, this refinement relates directly to the user and their needs, and is therefore just one of the useful outcomes that could be obtained from this Pareto set. The refined cases are shown in fig. 7.18.

The difference in characteristics between the high-volume result and some reference cases is shown in table 7.2. The reference cases are the average, median, and lowest utopia distance (in other words, best) shape of the initial population. The best shape of the ini-



(a)

(b)

Figure 7.18. The result of refining two shapes from the WB Pareto set. Figure 7.18a was selected for high volume, and fig. 7.18b was selected for low drag.

Difference	Average (%)	Median (%)	Best (%)
C_dA	330	458	1109
V	825	2251	2554
Ω_{m_α}	-140	-457	-95
Ω_{n_β}	-112	-346	-36
Δ_{PG}	459	417	122

Table 7.2. The percentage difference between the characteristics of the high-drag case and three reference cases from the initial population.

difference	Average (%)	Median (%)	Best (%)
C_dA	-23	0	118
V	-37	60	81
Ω_{m_α}	-78	-48	-82
Ω_{n_β}	-82	-62	-95
Δ_{PG}	95	80	-23

Table 7.3. The percentage difference between the characteristics of the low-drag case and three reference cases from the initial population.

tial set is that in the top right corner of the collage shown in fig. 6.13. Nether the average nor median correspond to any one specific shape. They are considered representative of the characteristics of the set as a whole. It can be seen that compared to all three cases, there has been substantial increase in C_dA and Δ_{PG}, with a desirable decrease in Ω_{m_α} and Ω_{n_β}, and a desirable increase in volume. The trade-off across conflicting objectives can clearly be seen. The program has clearly found a solution which maximises volume at the cost of drag, while maintaining controllability and satisfying all specified constraints.

The same comparison was also performed for the low-drag result, and shown in table 7.3. The two controllability characteristics, Ω_{m_α} and Ω_{n_β}, show a desirable decrease compared to all cases. For the average of the set, the drag is reduced with a corresponding reduction in volume, while Δ_{PG} is increased. This is to be expected of the minimum drag case. There is no notable reduction in drag over the median of the set, however there is an increase in volume and an increase in Δ_{PG}. The most interesting comparison is that with the best shape of the initial set. Unexpectedly, the drag has increased by 118% for the optimised result over that of the initial set, bringing the numerical value of the result to 18 compared to that of the best shape, 8. Note that numerically, the input set has a range of C_dA of 4 to 115. Thus, numerically this can be considered a small increase in drag. However, there is also a large increase in volume, a desirable decrease in Δ_{PG} indicating better stability, as well as the decrease of Ω_{m_α} and Ω_{n_β}. Thus, the algorithm has identified a solution which numerically increases drag slightly, while providing better results for all four other objectives and maintaining feasibility in terms of the constraints.

Thus, the value of the optimisation is apparent. While a human designer can identify shapes with better individual objectives than the algorithm, the shapes output by the algorithm were significantly increase the suitability across all objectives compared to the initial set. The two least intuitive objectives, Ω_{m_α} and Ω_{n_β}, showed a desirable decrease in every case. However, the optimisation is as yet unable to directly provide results which are realistically suitable for production in terms of fabricating the body directly from the output CAD models. Instead, it provides a starting point, from which an engineer can consider different configurations of the body. The contribution of the algorithm to the early stages of the design process means that an engineer can better select suitable shapes to be considered for final design refinement.

7.7 Summary

The main conclusions drawn from these results can be summarised as follows:

- Both SO and MO show better exploration of the search space than the initial population.
- MO results improve on SO results, which themselves improve on the initial population.
- Tail geometry appears to have less of an influence on $C_d A_{ref}$, v, Ω_{m_α} and Ω_{n_β} than frontal face geometry. However, it controls Δ_{pg}.
- For objectives which are not intuitive, such as Ω_{m_α} and Ω_{n_β} in contrast to the more intuitive $C_d A_{ref}$ and v, the algorithm is particularly useful in addition to human design expertise in finding good candidate solutions.
- Optimising just one objective provides a very limited view of the performance of the spacecraft body. Oftentimes, the common approach of minimising drag causes a considerable loss in performance to the satellite as a whole. MO results are more balanced across the different characteristics than SO.
- Optimising just one part of a body also provides an incomplete picture of performance, and may not align with the results for that part when optimisation of the whole body is performed. For example, the geometry obtained by optimising the face and tail separately and putting the two together does not agree with that obtained when both are optimised together. Thus, the common approach seen in literature of optimising only one part of a body is incomplete.
- The algorithm often has preferred designs, with the Pareto set being comprised of variations on those themes. This indicates good algorithm convergence. Multiple combinations of GA parameters were employed for a general approach, and they all converge to similar Pareto sets. Thus, it can be seen that the results are independent of GA parameter settings, making them more robust.

- No variation was seen for optimisation across different altitudes. The magnitude of the objectives varied, but the results were identical barring the inherent stochastic variation of the GA.

- Advantageous geometries for the conditions at hand can be identified by the framework.

- The GA still exhibits some fluctuation, and requires input from a human user in refining the final design. The presented results can either be used to select one output, or to identify characteristics of interest, to be taken forwards through a further design process. Thus, such a framework is in addition to human expertise, not instead of it.

- When the feature studied does not have a large impact on the objectives, such as in the TF case, the algorithm struggles to converge and shows erratic results.

Therefore, it is predicted that the integration of computational optimisation into the mission design process will increase the range of possible solutions, and decrease labour cost by streamlining the design process. A multi-objective approach also find more suitable solutions than single-objective or manual design. The practicality and versatility of such a framework is readily apparent.

Chapter 8

Conclusions and Future Work

This research was fundamentally driven by the rapidly increasing popularity of satellite flight in VLEO, at orbital altitudes of less than 450 km. Such altitudes present multiple benefits, particularly for earth observation missions, but also suffer from the significant drawback of non-negligible atmospheric drag. This drag, which acts to de-orbit spacecraft and bring about a premature end to missions, must be mitigated in order for sustained flight to be feasible. To this end, the method proposed in this thesis is the bespoke design of the outer geometry of spacecraft through the use of multi-objective computational optimisation. While optimisation of the outer geometry is commonplace in continuous flow for bodies such as aeroplanes, cars and motorbikes, literature on the influence of external body geometry on satellite flight in VLEO is scarce. None attempts to solve the problem through such extensive multi-objective computational optimisation as was employed here. This approach has revealed important insights into the problem at hand.

Initial research indicated inconsistency between different literature sources in defining this orbital regime. To resolve this, an investigation was undertaken into the influence of atmospheric drag on spacecraft, in comparison with other perturbing forces. The analysis reinforced the significant impact of atmospheric drag on satellites below an altitude of 450 km, which is the recommended value for the upper limit of VLEO.

Designing spacecraft in VLEO to survive this high drag force is vital to avoid premature de-orbiting and failure in achieving mission objectives. In addition to drag, it is also beneficial to examine other characteristics of the body, such as payload volume, aspect ratio, and stability. Upon reviewing existing literature, it was found that most satellite design methodologies rely only on the engineering expertise of the designer. Those that do incorporate a computational framework to more thoroughly analyse the search space focus on single-objective drag optimisation, without considering supplementary characteristics. The main aim of this thesis was to analyse the interplay between multiple satellite characteristics that are important to successfully carrying out a mission. This novel approach harnesses more information about satellite behaviour than heretofore considered.

Unfortunately, drag analysis is itself non-trivial. Most existing methods are either restricted to simplistic designs, have excessive computational run-time requirements, or suffer from incomplete knowledge of their behaviour in edge cases. To solve these problems, the ADBSat panel method software for satellite drag analysis was improved, tested and validated. Good agreement was proven with existing methods of drag analysis and published literature. A new optimisation framework was then devised which implements ADBSat and uses the GA algorithm to find satellite geometries which simultaneously minimise drag and maximise volume, while maintaining stability and controllability. Results indicate that the multi-objective GA, when benefiting from a good initial population and appropriate algorithm parameters, can more thoroughly search the solution space than an engineer can by hand. It can output solutions which give the designer a broader choice of parameters to select from. In particular, parameters which are not intuitive and have a complex search space benefit greatly from a computational approach, such as stability characteristics (in contrast with the more intuitive drag and volume). Thus, the novel framework can improve understanding of parameter trade-off across different body geometries, and facilitate the design of more efficient spacecraft for flight in VLEO.

8.1 Main Contributions

The main novel scientific contributions of this work are as follows:

- **Industry trends:** The state of the spacecraft launch market to VLEO was analysed. Miniaturisation and standardisation of spacecraft components are increasingly popular, although recent science missions have benefited from bespoke engineering. A significant increase in the number of satellites in VLEO over the period 2018 to 2021 was noted: the number of satellites quintupled, equivalent to a doubling of their percentage share of satellites in orbit. The increase in investment into satellite flight at VLEO is clear. Multiple sources predict that this increase will maintain for the foreseeable future, highlighting the need for solutions to the unique problems which VLEO poses.
- **Definition of VLEO:** A literature review indicated conflicting definitions of VLEO, placing its upper bound anywhere from 160 km to 500 km. The atmospheric model which best represents the conditions in VLEO was identified, and applied to determine that the drag force exceeds other significant perturbing forces at around 450 km orbital altitude. An investigation into the fuel mass fraction needed for orbit-keeping showed that around 450 km and below, this fraction begins to be excessively large, although mitigated in some cases by drag-oriented design. Therefore, it is proven that the existing literature which uses 450 km altitude as the limit of VLEO is likely consistent with the idea of drag becoming dominant.
- **ADBSat benefits and drawbacks:** ADBSat was validated against closed-form solutions, DSMC simulations, and reported literature values. Analysis proves that it

produces a good approximation of the aerodynamic characteristics of a body in orbit. New, thorough information about its accuracy when dealing with convex shapes and shapes which exhibit self-shading was made available to the wider scientific community. Mitigation strategies are suggested for cases when it exhibits low accuracy. When these are employed, comparison to literature values suggests an error of up to 3%. As ADBSat is intended to be free and open-source, and interest in its use has already been registered from the wider scientific community, the information gained through this research will prove invaluable to future engineers.

- **Optimisation framework:** Following on from the validation of ADBSat, an innovative optimisation framework was designed which uses ADBSat for drag analysis of satellite geometries. This is the first example among the literature available of a panel method being used for drag optimisation. Additionally, three stability objectives and payload volume were also optimised. Multiple constraints have been integrated to maintain shape feasibility and ADBSat accuracy. This is the first such large-scale multi-objective multi-constraint framework to be harnessed for the purpose of spacecraft design in VLEO.
- **Knowledge of search space:** Results show that the proposed framework can identify a set of solutions which span a larger portion of the search space than the initial population. The feasibility of examining multiple objectives was proven, and the fact that it is more beneficial for satellite body design than optimising just one objective. Whereas the existing literature often treated minimising drag as the only goal, analysis indicated that this can at times be detrimental when its effect on other body characteristics is considered. Similarly, optimising only the front or rear face of the object provides a limited understanding of the body characteristics. Whole-body optimisation with a looser shape parametrisation which affords more control to the program, as undertaken here for the first time, is necessary in order to attain desirable shapes. Analysis of the output shapes provided new insight into the trade-off between the five parameters, and thus the shape of the solution space.

8.2 Future Work

This thesis has laid the ground work of developing a multi-objective optimisation framework for satellite body design. This work could be further continued through any of the following avenues:

- **ADBSat mesh quality:** A number of additional features have been identified that could be added to ADBSat to improve its functionality and user-friendliness. Firstly, there are no mesh checks for suitability, meaning that ADBSat cannot identify meshes which will output poor results, and it is up to the user to ensure a good quality CAD model. Following the recommendation of the reviewer of Sinpetru, Crisp, Roberts, *et al.* [92], in the future mesh checks could be incorporated into ADBSat. The following are recommended:

1. Check that all faces in the Wavefront file are triangular
2. Check that the mesh is watertight
3. Flag zero area faces which will cause NaN results
4. Check for degenerate vertices and faces
5. Check for convexity, using the same method as the optimisation framework
6. Flag overlapping faces

All these problems could result in an error on the results much higher than the reported 3%. It would be beneficial for the program to be able to alert the user to them. While it would be ideal to include functionality in ADBSat to automatically fix these issues, ADBSat is not intended to be a CAD program. Therefore, it would be suitable to just flag these issues and inform the user that they must further refine the mesh for accurate results.

- **ADBSat suggested α_E:** Suitable values of α_E for the Sentman GSIM can be difficult to determine. They require extensive experimentation, or fuzzy interpolation of literature data. However, some models exist which can be used to calculate an approximate value, such as the Langmuir isotherm model of Pilinski, Argrow, and Palo [122]. This particular model requires only atmospheric parameters which are readily available in ADBSat. It is proposed to integrate a model such as this into ADBSat to provide a suggested α_E for the Sentman model when the user does not know its value.

- **ADBSat solar coefficient validation:** As well as drag coefficient estimation, ADBSat can also estimate the solar coefficients of a satellite by applying the equations of Luthcke, Marshall, Rowton, *et al.* [272] to the model in much the same way as GSIM equations. This provides a useful approximation of the solar radiation force experienced by the spacecraft. However, the model has not yet undergone validation. The full validation of this model is recommended, which would add significant functionality to ADBSat.

- **Optimisation completeness:** The objectives described above are a good starting point for realistic computational satellite geometry design. However, more are needed if the framework is to output truly realistic results. Additional objectives could include payload placement and look angle considerations, aerodynamic control surfaces, thruster design and placement, solar array placement and sizing, materials cost, booms and appendages, antennae placement, and thermal analysis. Many of these could also be controlled via constraints. There is a large scope for extending the framework to cover more specific uses.

Additionally, research is currently ongoing at the University of Manchester to identify materials which promote specular reflection. As the Sentman GSIM used in this work assumes fully diffuse reflection, it would be useful to extend the work to other models which can capture some degree of specularity, such as the CLL GSIM. This would capture a wider set of reflection characteristics and may identify realistic lower-drag configurations based on the novel materials.

- **Incorporation into mission design:** To better understand the implications of these results, the next step after successful shape optimisation is to incorporate it into a mission design framework. An analysis of case studies based on common satellite missions is recommended, with a focus on the unique challenges posed by each one. This will reveal the overall effect of the optimisation on each mission. Case studies could include, for example, large optical payloads, a communications satellite, a synthetic aperture radar mission, or a small CubeSat mission among others. A complete mission analysis from inception through to testing and launch is exceedingly complex for this task. A simplified mission design is suggested which focuses on certain unique challenges posed by each mission.

8.3 Concluding Remarks

The benefits of the VLEO orbital regime are now abundantly clear. Undoubtedly, the number of missions which utilise this range of altitudes will continue to grow in future years. Now more than ever there is a need to solve the unique problems found therein.

In this thesis, we merely dip a toe into the waters of multi-objective computational optimisation for satellite geometry design. The requirement to validate ADBSat and ensure its efficacy before applying it, while of great value to the scientific community, meant that the optimisation was carried out under the weight of significant time constraints. Despite this, the framework is able to successfully examine the interplay between the considered objectives, and to explore the search space far more thoroughly than any single engineer could. The resulting candidate solutions show a range of characteristics which would enable a designer to better understand the consequences of design choices on the mission as a whole, and thus to more wisely choose a final geometry.

Computational optimisation evidently has real potential to play a powerful role in spacecraft geometry design. However, at present the framework is still in need of further development. It is hoped that in the future, this work is continued by the scientific community. While the recommendations laid out above identify a solid next step in which to take this research, ultimately, it is anticipated that future researchers will surpass them.

References

[1] N. Crisp, P. Roberts, S. Livadiotti, V. Oiko, S. Edmondson, S. Haigh, C. Huyton, L. Sinpetru, K. Smith, S. Worrall, J. Becedas, R. Domínguez, D. González, V. Hanessian, A. Mølgaard, J. Nielsen, M. Bisgaard, Y.-A. Chan, S. Fasoulas, G. Herdrich, F. Romano, C. Traub, D. Garcia-Almiñana, S. Rodríguez-Donaire, M. Sureda, D. Kataria, R. Outlaw, B. Belkouchi, A. Conte, J. Perez, R. Villain, B. Heißerer, and A. Schwalber, "The benefits of very low earth orbit for earth observation missions," *Progress in Aerospace Sciences*, vol. 117, p. 100 619, 2020, `https://doi.org/10.1016/j.paerosci.2020.100619`, ISSN: 0376-0421.

[2] P. Roberts, N. Crisp, S. Edmondson, S. Haigh, R. Lyons, V. Abrao Oiko, A. Macario Rojas, K. Smith, J. Becedas, G. González, I. Vázquez, Á. Braña, K. Antonini, K. Bay, L. Ghizoni, V. Jungnell, J. Morsbøl, T. Binder, A. Boxberger, G. Herdrich, F. Romano, S. Fasoulas, D. Garcia-Almiñana, S. Rodriguez-Donaire, D. Kataria, M. Davidson, R. Outlaw, B. Belkouchi, A. Conte, J. Perez, R. Villain, and A. Schwalber, "DISCOVERER: Radical redesign of earth observation satellites for sustained operation at significantly lower altitudes," in *68th International Astronautical Congress (IAC), Adelaide, Australia, 25-29 September 2017.*, `https://discovery.ucl.ac.uk/id/eprint/10066748/`, Sep. 2017.

[3] A. Witze, "The quest to conquer earth's space junk problem," *Nature*, vol. 561, no. 7721, pp. 24–27, 2018, `http://dx.doi.org/10.1038/d41586-018-06170-1`.

[4] J. Du, X. Lei, and J. Sang, "A space surveillance satellite for cataloging high-altitude small debris," *Acta Astronautica*, vol. 157, pp. 268–275, 2019, `https://doi.org/10.1016/j.actaastro.2019.01.003`, ISSN: 0094-5765.

[5] M. A. Teixeira, F. Goulart, and F. Campelo, "Evolutionary multiobjective optimization of winglets," in *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, ser. GECCO '16, `https://doi.org/10.1145/2908812.2908848`, Denver, Colorado, USA: Association for Computing Machinery, 2016, pp. 1021–1028, ISBN: 9781450342063.

[6] S. Eyi, J. O. Hager, and K. D. Lee, "Airfoil design optimization using the Navier-Stokes equations," *Journal of Optimization Theory and Applications*, vol. 83, 3 Dec. 1994, `https://doi.org/10.1007/BF02207637`.

[7] S. Peigin and B. Epstein, "Robust optimization of 2d airfoils driven by full navier–stokes computations," *Computers & Fluids*, vol. 33, no. 9, pp. 1175–1200, 2004, `https://doi.org/10.1016/j.compfluid.2003.11.001`, ISSN: 0045-7930.

[8] A. Shahrokhi and A. Jahangirian, "Airfoil shape parameterization for optimum navier–stokes design with genetic algorithm," *Aerospace Science and Technology*, vol. 11, no. 6, pp. 443–450, 2007, `https://doi.org/10.1016/j.ast.2007.04.004`, ISSN: 1270-9638.

[9] J. Li, M. A. Bouhlel, and J. R. R. A. Martins, "Data-based approach for fast airfoil analysis and optimization," *AIAA Journal*, vol. 57, no. 2, pp. 581–596, 2019, `https://doi.org/10.2514/1.J057129`.

[10] Z. Zhang, A. De Gaspari, S. Ricci, C. Song, and C. Yang, "Gradient-based aerodynamic optimization of an airfoil with morphing leading and trailing edges," *Applied Sciences*, vol. 11, no. 4, 2021, `https://doi.org/10.3390/app11041929`, ISSN: 2076-3417.

[11] J. Li, P. Wang, H. Dong, X. Wu, X. Chen, and C. Chen, "Shape optimisation of blended-wing-body underwater gliders based on free-form deformation," *Ships and Offshore Structures*, vol. 15, no. 3, pp. 227–235, 2020, `https://doi.org/10.1080/17445302.2019.1611989`.

[12] A. Ribeiro, A. Awruch, and H. Gomes, "An airfoil optimization technique for wind turbines," *Applied Mathematical Modelling*, vol. 36, no. 10, pp. 4898–4907, 2012, `https://doi.org/10.1016/j.apm.2011.12.026`, ISSN: 0307-904X.

[13] E. Fagan, O. De La Torre, S. Leen, and J. Goggins, "Validation of the multi-objective structural optimisation of a composite wind turbine blade," *Composite Structures*, vol. 204, pp. 567–577, 2018, `https://doi.org/10.1016/j.compstruct.2018.07.114`, ISSN: 0263-8223.

[14] A. Shourangiz-Haghighi, M. A. Haghnegahdar, L. Wang, M. Mussetta, A. Kolios, and M. Lander, "State of the art in the optimisation of wind turbine performance using CFD," *Archives of Computational Methods in Engineering*, vol. 27, 2 Apr. 2020, `https://doi.org/10.1007/s11831-019-09316-0`.

[15] M. T. Akram and M.-H. Kim, “Aerodynamic shape optimization of nrel s809 airfoil for wind turbine blades using reynolds-averaged navier stokes model—part ii,” *Applied Sciences*, vol. 11, no. 5, 2021, https://doi.org/10.3390/app11052211, ISSN: 2076-3417.

[16] P. Sharma, B. Gupta, M. Pandey, A. K. Sharma, and R. Nareliya Mishra, “Recent advancements in optimization methods for wind turbine airfoil design: A review,” *Materials Today: Proceedings*, vol. 47, pp. 6556–6563, 2021, https://doi.org/10.1016/j.matpr.2021.02.231, ISSN: 2214-7853.

[17] Union of Concerned Scientists, *UCS satellite database*, [Online; date accessed: 25/11/2021], 2021. [Online]. Available: https://www.ucsusa.org/resources/satellite-database.

[18] R. T. Nikolaevna, K. D. Sergeevna, and A. Z. Sardarovich, “Space market: Problems of commercialization, development trends, diversification,” *AIP Conference Proceedings*, vol. 2318, no. 1, p. 070 012, 2021, https://doi.org/10.1063/5.0036000.

[19] E. Rough, E. Kirk-Wade, A. Adcock, and C. Housley, *Future of the UK space industry*, UK Parliament debate pack, https://researchbriefings.files.parliament.uk/documents/CDP-2021-0006/CDP-2021-0006.pdf, 2021.

[20] C. Reigber, H. Lühr, and P. Schwintzer, “CHAMP mission status,” *Advances in Space Research*, vol. 30, no. 2, pp. 129–134, 2002, https://doi.org/10.1016/S0273-1177(02)00276-4, ISSN: 0273-1177.

[21] H. Liu, H. Lühr, V. Henize, and W. Köhler, “Global distribution of the thermospheric total mass density derived from CHAMP,” *Journal of Geophysical Research: Space Physics*, vol. 110, no. A4, 2005, https://doi.org/10.1029/2004JA010741.

[22] Y. Zhou, S. Ma, H. Lühr, C. Xiong, and C. Reigber, “An empirical relation to correct storm-time thermospheric mass density modeled by NRLMSISE-00 with CHAMP satellite air drag data,” *Advances in Space Research*, vol. 43, no. 5, pp. 819–828, 2009, https://doi.org/10.1016/j.asr.2008.06.016, ISSN: 0273-1177.

[23] B. D. Tapley, S. Bettadpur, M. Watkins, and C. Reigber, “The gravity recovery and climate experiment: Mission overview and early results,” *Geophysical Research Letters*, vol. 31, no. 9, p. L09607, 2004, https://doi.org/10.1029/2004GL019920.

[24] P. M. Mehta, C. A. McLaughlin, and E. K. Sutton, “Drag coefficient modelling for GRACE using direct simulation Monte Carlo,” *Advances in Space Research*, vol. 52, no. 12, pp. 2035–2051, 2013, https://doi.org/10.1016/j.asr.2013.08.033, ISSN: 0273-1177.

[25] M. R. Drinkwater, R. Haagmans, D. Muzi, A. Popescu, R. Floberghagen, M. Kern, and M. Fehringer, “The GOCE gravity mission: ESA’s first core Earth explorer,” in *Proceedings of the 3rd international GOCE user workshop*, https://doi.org/10.1007/978-94-017-1333-7_36, Citeseer, 2006, pp. 1–8.

[26] A. Bezdêk, J. Klokoĉník, J. Kostelecký, R. Floberghagen, and C. Gruber, “Simulation of free fall and resonances in the GOCE mission,” *Journal of Geodynamics*, vol. 48, no. 1, pp. 47–53, 2009, https://doi.org/10.1016/j.jog.2009.01.007, ISSN: 0264-3707.

[27] K. Fujita and A. Noda, “Rarefied aerodynamics of a super low altitude test satellite,” in *41st AIAA Thermophysics Conference*, https://doi.org/10.2514/6.2009-3606, 2009, p. 3606.

[28] Japan Aerospace Exploration Agency, *Tsubame transition to orbit keeping operations*, [Online; date accessed: 8/12/2021], Mar. 2019. [Online]. Available: https://global.jaxa.jp/press/2019/03/20190318a.html.

[29] R. P. Kornfeld, B. W. Arnold, M. A. Gross, N. T. Dahya, W. M. Klipstein, P. F. Gath, and S. Bettadpur, “GRACE-FO: The gravity recovery and climate experiment follow-on mission,” *Journal of Spacecraft and Rockets*, vol. 56, no. 3, pp. 931–951, 2019, https://doi.org/10.2514/1.A34326.

[30] F. Flechtner, P. Morton, M. Watkins, and F. Webb, “Status of the GRACE follow-on mission,” in *Gravity, Geoid and Height Systems*, http://dx.doi.org/10.1007/978-3-319-10837-7_15, Springer International Publishing, 2014, pp. 117–121, ISBN: 978-3-319-10837-7.

[31] N. Crisp, A. M. Rojas, P. Roberts, S. Edmondson, S. J. Haigh, B. Holmes, S. Livadiotti, V. Oiko, K. Smith, L. Sinpetru, J. Becedas, V. Sulliotti-Linner, S. Christensen, V. Hanessian, T. Jensen, J. Nielsen, M. Bisgaard, Y.-A. Chan, G. Herdrich, F. Romano, S. Fasoulas, C. Traub, D. García-Almiñana, S. Rodríguez-Donaire, M. Sureda, D. Kataria, B. Belkouchi, A. Conte, S. Seminari, and R. Villain, “Launch, operations, and first experimental results of the Satellite for Orbital Aerodynamics Research (SOAR),” in *72nd International Astronautical*

Congress (IAC), Dubai, United Arab Emirates, 25-29 September 2017., https://upcommons.upc.edu/handle/2117/359050, Oct. 2021.

[32] N. Crisp, P. Roberts, S. Livadiotti, A. M. Rojas, V. Oiko, S. Edmondson, S. Haigh, B. Holmes, L. Sinpetru, K. Smith, J. Becedas, R. Domínguez, V. Sulliotti-Linner, S. Christensen, J. Nielsen, M. Bisgaard, Y.-A. Chan, S. Fasoulas, G. Herdrich, F. Romano, C. Traub, D. García-Almiñana, S. Rodríguez-Donaire, M. Sureda, D. Kataria, B. Belkouchi, A. Conte, S. Seminari, and R. Villain, "In-orbit aerodynamic coefficient measurements using SOAR (Satellite for Orbital Aerodynamics Research)," *Acta Astronautica*, vol. 180, pp. 85–99, 2021, https://doi.org/10.1016/j.actaastro.2020.12.024, ISSN: 0094-5765.

[33] H. Heidt, J. Puig-Suari, A. Moore, S. Nakasuka, and R. Twiggs, "Cubesat: A new generation of picosatellite for education and industry low-cost space experimentation," in *AIAA Small Satellite Conference*, https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=2069&context=smallsat, 2000.

[34] A. Chin, R. Coelho, R. Nugent, R. Munakata, and J. Puig-Suari, "Cubesat: The pico-satellite standard for research and education," in *AIAA SPACE 2008 Conference & Exposition*, http://dx.doi.org/10.2514/6.2008-7734, 2008.

[35] Erik Kulu, *Nanosats database*, [Online; date accessed: 8/12/2021], 2019. [Online]. Available: https://www.nanosats.eu/.

[36] L. Hill, D. Barnhart, E. Fowler, R. Hunter, L. M. Hoag, B. Sullivan, and P. Will, "The market for satellite cellularization: A historical view of the impact of the satlet morphology on the space industry," in *AIAA SPACE 2013 Conference and Exposition*, http://dx.doi.org/10.2514/6.2013-5486, 2013.

[37] T. Jaeger, W. Mirczak, and B. Crandall, "Cellularized satellites-a small satellite instantiation that provides mission and space access adaptability," in *AIAA Small Satellite Conference*, https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=3363&context=smallsat, 2016.

[38] NovaWurks, *Reach*, [Online; date accessed: 8/12/2021], 2020. [Online]. Available: https://www.novawurks.com/company/accomplishments/.

[39] T. Buettner, A. Tanev, L. Pfotzer, A. Roennau, and R. Dillmann, "The intelligent computer aided satellite designer iCASD - creating viable configurations for modular satellites," in *2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, https://doi.org/10.1109/AHS.2018.8541366, Aug. 2018, pp. 25–32.

[40] N. Crisp, A. Macario, P. Roberts, D. Garcia-Almiñana, M. Sureda, and S. Rodriguez-Donaire, "Experimental results from the satellite for orbital aerodynamics research (soar) mission," in *Proceedings of the 73rd International Astronautical Congress*. International Astronautical Federation, 2022, pp. 1–9, http://hdl.handle.net/2117/374439, ISBN: 0074-1795.

[41] J. M. Picone, A. E. Hedin, D. P. Drob, and A. C. Aikin, "NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues," *Journal of Geophysical Research: Space Physics*, vol. 107, no. A12, pp. 15–16, 2002, https://doi.org/10.1029/2002JA009430, ISSN: 2169-9402.

[42] B. R. Bowman, W. K. Tobiska, F. A. Marcos, and C. Valladares, "The JB2006 empirical thermospheric density model," *Journal of Atmospheric and Solar-Terrestrial Physics*, vol. 70, no. 5, pp. 774–793, 2008, https://doi.org/10.1016/j.jastp.2007.10.002, ISSN: 1364-6826.

[43] B. Bowman, W. K. Tobiska, F. Marcos, C. Huang, C. Lin, and W. Burke, "A new empirical thermospheric density model JB2008 using new solar and geomagnetic indices," in *37th COSPAR Scientific Assembly*, ser. COSPAR Meeting, https://doi.org/10.2514/6.2008-6438, vol. 37, Aug. 2009, p. 367.

[44] S. Bruinsma, G. Thuillier, and F. Barlier, "The DTM-2000 empirical thermosphere model with new data assimilation and constraints at lower boundary: Accuracy and properties," *Journal of Atmospheric and Solar-Terrestrial Physics*, vol. 65, no. 9, pp. 1053–1070, 2003, https://doi.org/10.1016/S1364-6826(03)00137-8, ISSN: 1364-6826.

[45] S. L. Bruinsma, N. Sánchez-Ortiz, E. Olmedo, and N. Guijarro, "Evaluation of the DTM-2009 thermosphere model for benchmarking purposes," *Journal of Space Weather and Space Climate*, vol. 2, A04, 2012, https://doi.org/10.1051/swsc/2012005.

[46] S. Bruinsma, "The DTM-2013 thermosphere model," *Journal of Space Weather and Space Climate*, vol. 5, A1, 2015, https://doi.org/10.1051/swsc/2015001, ISSN: 2115-7251.

[47] P. M. Bainum, "A review of modelling techniques for the open and closed-loop dynamics of large space structures," in *Large Space Structures: Dynamics and Control*, https://doi.org/10.1007/978-3-642-83376-2_7, Berlin, Heidelberg: Springer Berlin Heidelberg, 1988, pp. 165–177, ISBN: 978-3-642-83376-2.

[48] D. V. McMillen, "Model "T" satellite series: Small satellites designed for scientific and commercial use," in *Proceedings of SPIE, small-satellite technology and applications*, https://doi.org/10.1117/12.45877, vol. 1495, 1991, pp. 95–102.

[49] L. J. Fogel, D. B. Fogel, and W. Atmar, "Evolutionary programming for ASAT battle management," in *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, http://dx.doi.org/10.1109/ACSSC.1993.342591, vol. 1, Nov. 1993, pp. 617–621.

[50] S. Lichten, C. Thornton, L. Young, and T. Yunck, "Concepts for automated precise low earth orbiter navigation with the global positioning system," *Telecommunications and Mission Operations Progress Report*, vol. 42-133, Jan. 1998, https://tmo.jpl.nasa.gov/progress_report/42-133/133F.pdf.

[51] M. Wade, *Vostok*, [Online; date accessed: 23/04/2019], 2019. [Online]. Available: http://www.astronautix.com/v/vostok.html.

[52] L. G. Jacchia and J. Slowey, "Atmospheric heating in the auroral zones: A preliminary analysis of the atmospheric drag of the Injun 3 satellite," *Journal of Geophysical Research (1896-1977)*, vol. 69, no. 5, pp. 905–910, 1964, https://ntrs.nasa.gov/citations/19640018811.

[53] D. R. Imbro, M. M. Moe, and K. Moe, "On fundamental problems in the deduction of atmospheric densities from satellite drag," *Journal of Geophysical Research (1896-1977)*, vol. 80, no. 22, pp. 3077–3086, 1975, https://doi.org/10.1029/JA080i022p03077.

[54] J. Park, R. S. Myong, D.-H. Kim, and S. Baek, "Aerodynamic shape optimization of space vehicle in very-low-earth-orbit," in *Proceedings of the 29th international symposium on rarefied gas dynamics*, https://doi.org/10.1063/1.4902745, vol. 1628, Dec. 2014, pp. 1331–1336.

[55] C. Hsieh, C. Pan, and M. Lo, "Determining aerothermodynamic effects on very-low-earth-orbit satellites using direct simulation monte carlo method," *International Journal of Computational Fluid Dynamics*, vol. 35, no. 5, pp. 388–411, 2021, https://doi.org/10.1080/10618562.2021.1942458.

[56] F. Romano, J. Espinosa-Orozco, M. Pfeiffer, G. Herdrich, N. Crisp, P. Roberts, B. Holmes, S. Edmondson, S. Haigh, S. Livadiotti, A. Macario-Rojas, V. Oiko, L. Sinpetru, K. Smith, J. Becedas, V. Sulliotti-Linner, M. Bisgaard, S. Christensen, V. Hanessian, T. K. Jensen, J. Nielsen, Y.-A. Chan, S. Fasoulas, C. Traub,

D. García-Almiñana, S. Rodríguez-Donaire, M. Sureda, D. Kataria, B. Belkouchi, A. Conte, S. Seminari, and R. Villain, "Intake design for an atmosphere-breathing electric propulsion system (abep)," *Acta Astronautica*, vol. 187, pp. 225–235, 2021, https://doi.org/10.1016/j.actaastro.2021.06.033, ISSN: 0094-5765.

[57] J. Virgili-Llop, P. Roberts, Z. Hao, L. Ramio, and V. Beauplet, "Very low earth orbit mission concepts for earth observation. Benefits and challenges," in *Proceedings of the 12th Reinventing Space Conference, London, United Kingdom, 2014*, https://www.researchgate.net/publication/271499606_Very_Low_Earth_Orbit_mission_concepts_for_Earth_Observation_Benefits_and_challenges, Nov. 2014.

[58] J. Virgili-Llop, H. C. Polat, and M. Romano, "Using shifting masses to reject aerodynamic perturbations and to maintain a stable attitude in very low earth orbit," in *26th AAS/AIAA Space Flight Mechanics Meeting, Napa, CA, February*, https://calhoun.nps.edu/handle/10945/50863, 2016, pp. 14–18.

[59] ——, "Attitude stabilization of spacecraft in very low earth orbit by center-of-mass shifting," *Frontiers in Robotics and AI*, vol. 6, p. 7, 2019, https://doi.org/10.3389/frobt.2019.00007, ISSN: 2296-9144.

[60] N. Crisp, P. Roberts, F. Romano, K. Smith, V. Oiko, V. Sulliotti-Linner, V. Hanessian, G. Herdrich, D. García-Almiñana, D. Kataria, and S. Seminari, "System modelling of very low earth orbit satellites for earth observation," *Acta Astronautica*, vol. 187, pp. 475–491, 2021, https://doi.org/10.1016/j.actaastro.2021.07.004, ISSN: 0094-5765.

[61] J. A. Walsh and L. Berthoud, "Reducing spacecraft drag in very low Earth orbit through shape optimisation," in *7th European Conference for Aeronautics and Aerospace Sciences (EUCASS)*, https://doi.org/10.13009/EUCASS2017-449, Sep. 2017.

[62] J. Walsh, L. Berthoud, and C. Allen, "Drag reduction through shape optimisation for satellites in very low earth orbit," *Acta Astronautica*, vol. 179, pp. 105–121, 2021, https://doi.org/10.1016/j.actaastro.2020.09.018, ISSN: 0094-5765.

[63] A. Goto, K. Umeda, K. Yukumatsu, and Y. Kimoto, "Property changes in materials due to atomic oxygen in the low earth orbit," *CEAS Space Journal*, vol. 13,

3 Jul. 2021, https://doi.org/10.1007/s12567-021-00376-2, ISSN: 868-2510.

[64] M. Leomanni, A. Garulli, A. Giannitrapani, and F. Scortecci, "Propulsion options for very low earth orbit microsatellites," *Acta Astronautica*, vol. 133, pp. 444–454, 2017, https://doi.org/10.1016/j.actaastro.2016.11.001, ISSN: 0094-5765.

[65] T. Schönherr, K. Komurasaki, F. Romano, B. Massuti-Ballester, and G. Herdrich, "Analysis of atmosphere-breathing electric propulsion," *IEEE Transactions on Plasma Science*, vol. 43, no. 1, pp. 287–294, Jan. 2015, https://doi.org/10.1109/TPS.2014.2364053.

[66] F. Romano, B. Massuti-Ballester, T. Binder, G. Herdrich, S. Fasoulas, and T. Schönherr, "System analysis and test-bed for an atmosphere-breathing electric propulsion system using an inductive plasma thruster," *Acta Astronautica*, vol. 147, pp. 114–126, 2018, https://doi.org/10.1016/j.actaastro.2018.03.031, ISSN: 0094-5765.

[67] T. Minton, T. Schwartzentruber, and C. Xu, "On the utility of coated poss-polyimides for vehicles in very low earth orbit (vleo)," *ChemRxiv*, 2021, https://doi.org/10.33774/chemrxiv-2021-x41bm.

[68] K. Fujita and A. Noda, "Aerodynamics of satellites on a super low earth orbit," *AIP Conference Proceedings*, vol. 1084, no. 1, pp. 772–777, 2008, https://doi.org/10.1063/1.3076580.

[69] A. A. Doroshkin, A. M. Zadorozhny, O. N. Kus, V. Y. Prokopyev, and Y. M. Prokopyev, "Experimental study of lora modulation immunity to doppler effect in cubesat radio communications," *IEEE Access*, vol. 7, pp. 75 721–75 731, 2019, https://doi.org/10.1109/ACCESS.2019.2919274.

[70] A. Grasso, "Feasibility study of an ultra-low altitude hyperspectral micro-satellite," *Proceedings of the International Astronautical Congress, IAC*, vol. 13, pp. 10 223–10 239, Jan. 2012, https://www.researchgate.net/publication/289656269_Feasibility_study_of_an_ultra-low_altitude_hyperspectral_micro-satellite.

[71] A. Golikov and A. Filatyev, "Integrated optimization of trajectories and layout parameters of spacecraft with air-breathing electric propulsion," *Acta Astronautica*, 2021, https://doi.org/10.1016/j.actaastro.2021.06.052, ISSN: 0094-5765.

[72] S.-t. Yu and C.-z. Fan, “Aerodynamic analysis and drag-reduction design for ultra-low-orbit satellite,” in *IOP Conference Series: Materials Science and Engineering*, `http://dx.doi.org/10.1088/1757-899X/887/1/012013`, IOP Publishing, vol. 887, 2020, p. 012 013.

[73] P. W. Fortescue and J. P. W. Stark, Eds., *Spacecraft systems engineering*, 2nd. Chichester: Wiley, 1995, `https://www.wiley.com/en-us/Spacecraft+Systems+Engineering,+4th+Edition-p-9780470750124`, ISBN: 0471952206.

[74] D. J. Pawlowski and A. J. Ridley, “Quantifying the effect of thermospheric parameterization in a global model,” *Journal of Atmospheric and Solar-Terrestrial Physics*, vol. 71, no. 17, pp. 2017–2026, 2009, `https://doi.org/10.1016/j.jastp.2009.09.007`, ISSN: 1364-6826.

[75] D. A. Vallado and D. Finkleman, “A critical assessment of satellite drag and atmospheric density modeling,” *Acta Astronautica*, vol. 95, pp. 141–165, 2014, `https://doi.org/10.1016/j.actaastro.2013.10.005`, ISSN: 0094-5765.

[76] S. Livadiotti, N. H. Crisp, P. C. Roberts, S. D. Worrall, V. T. Oiko, S. Edmondson, S. J. Haigh, C. Huyton, K. L. Smith, L. A. Sinpetru, B. E. Holmes, J. Becedas, R. M. Domínguez, V. Cañas, S. Christensen, A. Mølgaard, J. Nielsen, M. Bisgaard, Y.-A. Chan, G. H. Herdrich, F. Romano, S. Fasoulas, C. Traub, D. Garcia-Almiñana, S. Rodríguez-Donaire, M. Sureda, D. Kataria, B. Belkouchi, A. Conte, J. S. Perez, R. Villain, and R. Outlaw, “A review of gas-surface interaction models for orbital aerodynamics applications,” *Progress in Aerospace Sciences*, vol. 119, p. 100 675, 2020, `https://doi.org/10.1016/j.paerosci.2020.100675`.

[77] S. Livadiotti, “Application of orbital aerodynamics to satellite attitude control,” `https://www.research.manchester.ac.uk/portal/en/theses/application-of-orbital-aerodynamics-to-satellite-attitude-control(c751e053-99bf-4b36-9493-3019aa1ef802).html`, Ph.D. dissertation, University of Manchester, 2021.

[78] D. Bilitza, D. Altadill, V. Truhlik, V. Shubin, I. Galkin, B. Reinisch, and X. Huang, “International reference ionosphere 2016: From ionospheric climate to real-time weather predictions,” *Space Weather*, vol. 15, no. 2, pp. 418–429, 2017, `https://doi.org/10.1002/2016SW001593`.

[79] V. P. Zhaborovskyi, “Quantitative analysis of the atmospheric density models applicable for determination of artificial satellite deceleration,” *Kinematics and*

Physics of Celestial Bodies, vol. 30, no. 6, pp. 308–312, Nov. 2014, https://doi.org/10.3103/S0884591314060087, ISSN: 1934-8401.

[80] E. N. Doornbos, "Thermospheric density and wind determination from satellite dynamics," https://link.springer.com/book/10.1007/978-3-642-25129-0, Ph.D. dissertation, Delft University of Technology, Mar. 2011, ISBN: 9789090260518.

[81] J. S. Kim, J. V. Urbina, T. J. Kane, and D. B. Spencer, "Improvement of TIE-GCM thermospheric density predictions via incorporation of helium data from NRLMSISE-00," *Journal of Atmospheric and Solar-Terrestrial Physics*, vol. 77, pp. 19–25, 2012, https://doi.org/10.1016/j.jastp.2011.10.018, ISSN: 1364-6826.

[82] O. Montenbruck, *Satellite orbits: models, methods, and applications*. Berlin: Springer, 2000, https://link.springer.com/book/10.1007/978-3-642-58351-3, ISBN: 354067280X.

[83] G. Seeber, *Satellite Geodesy*, 2nd. Berlin, New York: Walter de Gruyter, 2003, https://doi.org/10.1515/9783110200089, ISBN: 9783110175493.

[84] H. D. Curtis, "Chapter 12 - introduction to orbital perturbations," in *Orbital Mechanics for Engineering Students*, Third, https://www.elsevier.com/books/orbital-mechanics-for-engineering-students/curtis/978-0-12-824025-0, Boston: Butterworth-Heinemann, 2014, pp. 651–720, ISBN: 978-0-08-097747-8.

[85] G. Xu and Y. Xu, *GPS : Theory, Algorithms and Applications*, 3rd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, https://link.springer.com/book/10.1007/978-3-540-72715-6, ISBN: 9783662503676.

[86] E. Yiğit and A. S. Medvedev, "Internal gravity waves in the thermosphere during low and high solar activity: Simulation study," *Journal of Geophysical Research: Space Physics*, vol. 115, no. A8, 2010, https://doi.org/10.1029/2009JA015106.

[87] Parsec vzw, *Space weather live*, [Online; date accessed: 30/03/2022], 2019. [Online]. Available: https://www.spaceweatherlive.com/en/solar-activity/solar-cycle.

[88] K. Lemmer, "Propulsion for cubesats," *Acta Astronautica*, vol. 134, pp. 231–243, 2017, https://doi.org/10.1016/j.actaastro.2017.01.048, ISSN: 0094-5765.

[89] I. Levchenko, K. Bazaka, Y. Ding, Y. Raitses, S. Mazouffre, T. Henning, P. J. Klar, S. Shinohara, J. Schein, L. Garrigues, M. Kim, D. Lev, F. Taccogna, R. W. Boswell, C. Charles, H. Koizumi, Y. Shen, C. Scharlemann, M. Keidar, and S. Xu, "Space micropropulsion systems for CubeSats and small satellites: From proximate targets to furthermost frontiers," *Applied Physics Reviews*, vol. 5, no. 1, p. 011 104, 2018, https://doi.org/10.1063/1.5007734.

[90] D. Krejci and P. Lozano, "Space propulsion technology for small spacecraft," *Proceedings of the IEEE*, vol. 106, no. 3, pp. 362–378, 2018, https://doi.org/10.1109/JPROC.2017.2778747.

[91] E. S. Agency, *GRACE (gravity recovery and climate experiment)*, [Online; date accessed: 31/03/2022], 2022. [Online]. Available: https://directory.eoportal.org/web/eoportal/satellite-missions/g/grace.

[92] L. A. Sinpetru, N. H. Crisp, P. C. Roberts, V. Sulliotti-Linner, V. Hanessian, G. H. Herdrich, F. Romano, D. Garcia-Almiñana, S. Rodríguez-Donaire, and S. Seminari, "ADBSat: Verification and validation of a novel panel method for quick aerodynamic analysis of satellites," *Computer Physics Communications*, vol. 275, p. 108 327, 2022, https://doi.org/10.1016/j.cpc.2022.108327, ISSN: 0010-4655.

[93] M. Schelkle, "The GRACE cold gas attitude and orbit control system," in *Spacecraft Propulsion*, https://adsabs.harvard.edu/full/2000ESASP.465..769S, vol. 465, 2000, p. 769.

[94] N. Wallace, P. Jameson, C. Saunders, M. Fehringer, C. Edwards, and R. Floberghagen, "The GOCE ion propulsion assembly - lessons learnt from the first 22 months of flight operations," in *32nd International Electric Propulsion Conference*, http://electricrocket.org/IEPC/IEPC-2011-327.pdf, 2011.

[95] F. Gini, "GOCE precise non-gravitational force modeling for POD applications," https://www.research.unipd.it/handle/11577/3423563, Ph.D. dissertation, Università degli Studi di Padova, 2014.

[96] D. Mostaza-Prieto, "Characterisation and applications of aerodynamic torques on satellites," https://www.research.manchester.ac.uk/portal/en/theses/characterisation-and-applications-of-aerodynamic-torques-on-satellites(8cff03c1-7167-47b3-9688-cd8ba65c1f3d).html, Ph.D. dissertation, The University of Manchester, 2017.

[97] J. D. Fuller and R. H. Tolson, "Improved method for the estimation of spacecraft free-molecular aerodynamic properties," *Journal of Spacecraft and Rockets*, vol. 46, no. 5, pp. 938–948, 2009, https://doi.org/10.2514/1.43205.

[98] J. Moss, K. Boyles, and F. Greene, "Orion aerodynamics for hypersonic free molecular to continuum conditions," in *14th AIAA/AHI Space Planes and Hypersonic Systems and Technologies Conference*, https://doi.org/10.2514/6.2006-8081, 2006.

[99] J. Moss, C. Glass, and F. Greene, "DSMC simulations of Apollo capsule aerodynamics for hypersonic rarefied conditions," in *9th AIAA/ASME Joint Thermophysics and Heat Transfer Conference*, https://doi.org/10.2514/6.2006-3577, 2006, p. 3577.

[100] K. Moe, M. M. Moe, and C. J. Rice, "Simultaneous analysis of multi-instrument satellite measurements of atmospheric density," *Journal of Spacecraft and Rockets*, vol. 41, no. 5, pp. 849–853, 2004, https://doi.org/10.2514/1.2090.

[101] D. Mostaza Prieto, B. P. Graziano, and P. C. Roberts, "Spacecraft drag modelling," *Progress in Aerospace Sciences*, vol. 64, pp. 56–65, 2014, https://doi.org/10.1016/j.paerosci.2013.09.001, ISSN: 0376-0421.

[102] P. M. Mehta, A. Walker, E. Lawrence, R. Linares, D. Higdon, and J. Koller, "Modeling satellite drag coefficients with response surfaces," *Advances in Space Research*, vol. 54, no. 8, pp. 1590–1607, 2014, https://doi.org/10.1016/j.asr.2014.06.033, ISSN: 0273-1177.

[103] R. M. Fredo and M. H. Kaplan, "Procedure for obtaining aerodynamic properties of spacecraft," *Journal of Spacecraft and Rockets*, vol. 18, no. 4, pp. 367–373, 1981, https://doi.org/10.2514/3.28061.

[104] Blender Online Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org.

[105] L. H. Sentman, "Free molecular flow theory and its application to the determination of aerodynamic forces," Lockheed Missiles and Space Co., Sunnyvale, CA, Tech. Rep. LMSC-448514, 1961, https://apps.dtic.mil/sti/citations/AD0265409.

[106] S. A. Schaaf and P. L. Chambre, *Flow of Rarefied Gases*. Princeton, NJ: Princeton University Press, 1961, vol. 8, https://doi.org/10.1515/9781400885800.

[107] G. Cook, "Satellite drag coefficients," *Planetary and Space Science*, vol. 13, no. 10, pp. 929–946, 1965, https://doi.org/10.1016/0032-0633(65)90150-9, ISSN: 0032-0633.

[108] J. A. Storch, "Aerodynamic disturbances on spacecraft in free-molecular flow (monograph)," The Aerospace Corporation, Tech. Rep., 2002, https://apps.dtic.mil/sti/citations/ADA410696.

[109] M. M. Moe, S. D. Wallace, and K. Moe, "Recommended drag coefficients for aeronomic satellites," in *The Upper Mesosphere and Lower Thermosphere: A Review of Experiment and Theory*, R. Johnson and T. Killeen, Eds. Washington, D. C.: American Geophysical Union (AGU), 1995, pp. 349–356, https://doi.org/10.1029/GM087p0349, ISBN: 9781118664247.

[110] K. Moe, M. M. Moe, and S. D. Wallace, "Improved satellite drag coefficient calculations from orbital measurements of energy accommodation," *Journal of Spacecraft and Rockets*, vol. 35, no. 3, pp. 266–272, May 1998, https://doi.org/10.2514/2.3350.

[111] K. Moe and M. M. Moe, "Gas-surface interactions and satellite drag coefficients," *Planetary and Space Science*, vol. 53, no. 8, pp. 793–801, 2005, https://doi.org/10.1016/j.pss.2005.03.005, ISSN: 0032-0633.

[112] C. Pardini, L. Anselmo, K. Moe, and M. Moe, "Drag and energy accommodation coefficients during sunspot maximum," *Advances in Space Research*, vol. 45, no. 5, pp. 638–650, 2010, https://doi.org/10.1016/j.asr.2009.08.034, ISSN: 0273-1177.

[113] M. D. Pilinski, K. Moe, S. E. Palo, and B. M. Argrow, "Measuring absolute thermospheric densities and accommodation coefficients using paddlewheel satellites: Past findings, present uses, and future mission concepts," *The Journal of the Astronautical Sciences*, vol. 58, no. 3, pp. 531–549, 2011, https://doi.org/10.1007/BF03321184.

[114] R. G. Lord, "Some extensions to the cercignani–lampis gas–surface scattering kernel," *Phys.Fluids A-Fluid*, vol. 3, no. 4, pp. 706–710, 1991, https://doi.org/10.1063/1.858076.

[115] P. M. Mehta, A. Walker, C. A. McLaughlin, and J. Koller, "Comparing physical drag coefficients computed using different gas-surface interaction models," *Journal of Spacecraft and Rockets*, vol. 51, no. 3, pp. 873–883, 2014, https://doi.org/10.2514/1.A32566.

[116] L. B. Thomas, “Thermal accommodation of gases on solids,” in *Fundamentals of Gas–Surface Interactions*, H. Saltsburg, J. N. Smith, and M. Rogers, Eds., https://doi.org/10.1016/B978-1-4832-2901-0.50026-3, Academic Press, 1967, pp. 346–369, ISBN: 978-1-4832-2901-0.

[117] R. H. Krech, M. J. Gauthier, and G. E. Caledonia, “High velocity atomic oxygen/surface accommodation studies,” *J. Spacecraft Rockets*, vol. 30, no. 4, pp. 509–513, 1993, https://doi.org/10.2514/3.25558.

[118] T. K. Minton, M. Tagawa, and G. M. Nathanson, “Energy accommodation in hyperthermal gas-surface collisions: Aerobraking in planetary atmospheres,” *J. Spacecraft Rockets*, vol. 41, no. 3, pp. 389–396, 2004, https://doi.org/10.2514/1.10724.

[119] K. Moe, “Absolute atmospheric densities determined from the spin and orbital decays of Explorer VI,” *Planetary and Space Science*, vol. 14, no. 11, pp. 1065–1075, 1966, https://doi.org/10.1016/0032-0633(66)90022-5.

[120] K. Moe and M. M. Moe, “Gas-surface interactions in low-earth orbit,” *AIP Conference Proceedings*, vol. 1333, no. 1, pp. 1313–1318, 2011, https://doi.org/10.1063/1.3562825.

[121] K. Moe and B. Bowman, “The effects of surface composition and treatment on drag coefficients of spherical satellites,” *Advances in the Astronautical Sciences*, vol. 123, pp. 137–152, 2006, https://www.researchgate.net/publication/287499075_The_Effects_of_Surface_Composition_and_Treatment_On_Drag_Coefficients_of_Spherical_Satellites.

[122] M. D. Pilinski, B. M. Argrow, and S. E. Palo, “Semiempirical model for satellite energy-accommodation coefficients,” *Journal of Spacecraft and Rockets*, vol. 47, no. 6, pp. 951–956, 2010, https://doi.org/10.2514/1.49330.

[123] E. K. Sutton, “Normalized force coefficients for satellites with elongated shapes,” *Journal of Spacecraft and Rockets*, vol. 46, no. 1, pp. 112–116, 2009, https://doi.org/10.2514/1.40940.

[124] L. Felicetti, F. Piergentili, and F. Santoni, “Thermosphere density and wind measurements in the equatorial region using a constellation of drag balance nanospacecraft,” *Advances in Space Research*, vol. 54, no. 3, pp. 546–553, 2014, https://doi.org/10.1016/j.asr.2013.09.008.

[125] O. Ben-Yaacov, E. Edlerman, and P. Gurfil, “Analytical technique for satellite projected cross-sectional area calculation,” *Advances in Space Research*, vol. 56, no. 2, pp. 205–217, 2015, https://doi.org/10.1016/j.asr.2015.04.004.

[126] A. Walker, P. Mehta, and J. Koller, “Drag coefficient model using the cercignani–lampis–lord gas–surface interaction model,” *Journal of Spacecraft and Rockets*, vol. 51, no. 5, pp. 1544–1563, 2014, https://doi.org/10.2514/1.A32677.

[127] G. Koppenwallner, “Comment on special section: New perspectives on the satellite drag environments of earth, mars, and venus,” *Journal of Spacecraft and Rockets*, vol. 45, no. 6, pp. 1324–1327, 2008, https://doi.org/10.2514/1.37539.

[128] G. Zuppardi, L. Morsa, R. Savino, M. Sippel, and T. Schwanekamp, “Rarefied aerodynamic characteristics of aero-space-planes: A comparative study of two gas-surface interaction models,” *European Journal of Mechanics - B/Fluids*, vol. 53, pp. 37–47, 2015, https://doi.org/10.1016/j.euromechflu.2015.04.003, ISSN: 0997-7546.

[129] J. Padilla and I. Boyd, “Assessment of gas-surface interaction models for computation of rarefied hypersonic flow,” *Journal of Thermophysics and Heat Transfer*, vol. 23, no. 1, pp. 96–105, Jan. 2009, https://doi.org/10.2514/1.36375, ISSN: 0887-8722.

[130] M. Leomanni, A. Garulli, A. Giannitrapani, and F. Scortecci, “Propulsion options for very low Earth orbit microsatellites,” *Acta Astronautica*, vol. 133, pp. 444–454, 2017, https://doi.org/10.1016/j.actaastro.2016.11.001, ISSN: 0094-5765.

[131] S. Andrews and L. Berthoud, “Characterising satellite aerodynamics in very low Earth orbit inclusive of ion thruster plume-thermosphere/ionosphere interactions,” *Acta Astronautica*, vol. 170, pp. 386–396, 2020, https://doi.org/10.1016/j.actaastro.2019.12.034, ISSN: 0094-5765.

[132] L. A. Sinpetru, N. H. Crisp, D. Mostaza-Prieto, S. Livadiotti, and P. C. Roberts, “Adbsat: Methodology of a novel panel method tool for aerodynamic analysis of satellites,” *Computer Physics Communications*, vol. 275, p. 108 326, 2022, https://doi.org/10.1016/j.cpc.2022.108326, ISSN: 0010-4655.

[133] G. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, ser. The Oxford engineering science series. Clarendon Press, 1994, https://searchworks.stanford.edu/view/2984210.

[134] C. White, M. K. Borg, T. J. Scanlon, S. M. Longshaw, B. John, D. R. Emerson, and J. M. Reese, "dsmcFoam+: An OpenFOAM based direct simulation Monte Carlo solver," *Computer Physics Communications*, vol. 224, pp. 22–43, 2018, https://doi.org/10.1016/j.cpc.2017.09.030, ISSN: 0010-4655.

[135] R. C. Palharini, C. White, T. J. Scanlon, R. E. Brown, M. K. Borg, and J. M. Reese, "Benchmark numerical simulations of rarefied non-reacting gas flows using an open-source DSMC code," *Computers & Fluids*, vol. 120, pp. 140–157, 2015, https://doi.org/10.1016/j.compfluid.2015.07.021, ISSN: 0045-7930.

[136] C. White, C. Colombo, T. J. Scanlon, C. R. McInnes, and J. M. Reese, "Rarefied gas effects on the aerodynamics of high area-to-mass ratio spacecraft in orbit," *Advances in Space Research*, vol. 51, no. 11, pp. 2112–2124, 2013, https://doi.org/10.1016/j.asr.2013.01.002.

[137] P. Parodi, D. Le Quang, F. Bariselli, S. Boccelli, Z. Alsalihi, and T. Magin, "Study of a collector-intake system for VLEO air-breathing platforms," in *International Conference on Flight vehicles, Aerothermodynamics and Re-entry Missions and Engineering (FAR), Monopoli, Italy*, https://www.researchgate.net/publication/337060417_Study_of_a_collector-intake_system_for_VLEO_air-breathing_platforms, Oct. 2009.

[138] M. D. Pilinski, B. M. Argrow, and S. E. Palo, "Drag coefficients of satellites with concave geometries: Comparing models and observations," *Journal of Spacecraft and Rockets*, vol. 48, no. 2, pp. 312–325, 2011, https://doi.org/10.2514/1.50915.

[139] T. J. Scanlon, E. Roohi, C. White, M. Darbandi, and J. M. Reese, "An open source, parallel DSMC code for rarefied gas flows in arbitrary geometries," *Computers & Fluids*, vol. 39, no. 10, pp. 2078–2089, 2010, https://doi.org/10.1016/j.compfluid.2010.07.014, ISSN: 0045-7930.

[140] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, "A tensorial approach to computational continuum mechanics using object-oriented techniques," *Journal of Computational Physics*, vol. 12, no. 6, pp. 620–631, 1998, https://doi.org/10.1063/1.168744.

[141] M. D. Weinberg, “Direct Simulation Monte Carlo for astrophysical flows - I. Motivation and methodology,” *Monthly Notices of the Royal Astronomical Society*, vol. 438, no. 4, pp. 2995–3006, Jan. 2014, https://doi.org/10.1093/mnras/stt2406.

[142] G. A. Bird, “Sophisticated DSMC,” in *Notes prepared for a short course at the DSMC07 meeting, Santa Fe, USA*, https://www.aeromech.usyd.edu.au/dsmc_gab/Resources/DSMC07notes.pdf, Sep. 2007.

[143] D. C. Wadsworth, D. B. VanGilder, and V. K. Dogra, “Gas-surface interaction model evaluation for DSMC applications,” *AIP Conf. Proc.*, vol. 663, no. 1, pp. 965–972, 2003, https://doi.org/10.1063/1.1581644.

[144] C. Cercignani and M. Lampis, “Kinetic models for gas-surface interactions,” *Journal of Computational and Theoretical Transport*, vol. 1, no. 2, pp. 101–114, 1971, https://doi.org/10.1080/00411457108231440.

[145] P. M. Mehta, C. A. McLaughlin, and E. K. Sutton, “Drag coefficient modeling for GRACE using direct simulation Monte Carlo,” *Advances in Space Research*, vol. 52, no. 12, pp. 2035–2051, 2013, https://doi.org/10.1016/j.asr.2013.08.033, ISSN: 0273-1177.

[146] D. E. Siskind, J. M. Picone, M. H. Stevens, and K. Minschwaner, “Middle and upper thermospheric odd nitrogen: 1. a new analysis of rocket data,” *J. Geophys. Res. Space Phys.*, vol. 109, no. A1, A01303, 2004, https://doi.org/10.1029/2003JA009943.

[147] D. Drob, J. Emmert, D. Siskind, and J. M. Picone, “NRLMSIS 2.0: New Formulation, New Data,” in *42nd COSPAR Scientific Assembly*, https://ui.adsabs.harvard.edu/abs/2018cosp...42E.893D/abstract, vol. 42, Jul. 2018, pp. C4.2-3–18.

[148] X. Cheng, J. Yang, C. Xiao, and X. Hu, “Density correction of NRLMSISE-00 in the middle atmosphere (20–100 km) based on TIMED/SABER density data,” *Annales Geophysicae Discussions*, vol. 2019, pp. 1–18, 2019, https://doi.org/10.5194/angeo-2019-93.

[149] Y. Dai, W. Pan, X. Hu, Z. Bai, C. Ban, H. Zhang, and Y. Che, “An approach for improving the NRLMSISE-00 model using a radiosonde at Golmud of the Tibetan Plateau,” *Meteorology and Atmospheric Physics*, vol. 132, pp. 451–459, 2020, https://doi.org/10.1007/s00703-019-00700-w.

[150] G. March, E. Doornbos, and P. Visser, “High-fidelity geometry models for improving the consistency of CHAMP, GRACE, GOCE and Swarm thermospheric density data sets,” *Advances in Space Research*, vol. 63, no. 1, pp. 213–238, 2019, `https://doi.org/10.1016/j.asr.2018.07.009`.

[151] E. K. Sutton, “Effects of solar disturbances on the thermosphere densities and winds from champ and grace satellite accelerometer data,” `https://www.proquest.com/openview/6cca9ab1067fedf8cc3eadd7adf5d82d/1?pq-origsite=gscholar&cbl=18750&diss=y`, Ph.D. dissertation, University of Colorado Boulder, 2008.

[152] B. R. Bowman, F. Marcos, K. Moe, and M. Moe, “Determination of drag coefficient values for CHAMP and GRACE satellites using orbit drag analysis,” *Advances in the Astronautical Sciences*, vol. 129, no. 1, pp. 147–166, 2008, `http://www.univelt.com/book=1786`.

[153] M. Tillerson, G. Inalhan, and J. P. How, “Co-ordination and control of distributed spacecraft systems using convex optimization techniques,” *International Journal of Robust and Nonlinear Control*, vol. 12, no. 2-3, pp. 207–242, 2002, `https://doi.org/10.1002/rnc.683`.

[154] A. Boudjemai, M. Bouanane, L. Merad, and A. S. Mohammed, “Small satellite structural optimisation using genetic algorithm approach,” in *2007 3rd International Conference on Recent Advances in Space Technologies*, `https://doi.org/10.1109/RAST.2007.4284021`, 2007, pp. 398–406.

[155] H. Kim and Y. K. Chang, “Mission scheduling optimization of sar satellite constellation for minimizing system response time,” *Aerospace Science and Technology*, vol. 40, pp. 17–32, 2015, `https://doi.org/10.1016/j.ast.2014.10.006`, ISSN: 1270-9638.

[156] S. Spangelo, J. Cutler, K. Gilson, and A. Cohn, “Optimization-based scheduling for the single-satellite, multi-ground station communication problem,” *Computers & Operations Research*, vol. 57, pp. 1–16, 2015, `https://doi.org/10.1016/j.cor.2014.11.004`, ISSN: 0305-0548.

[157] T. Savitri, Y. Kim, s. Jo, and H. Bang, “Satellite constellation orbit design optimization with combined genetic algorithm and semianalytical approach,” *International Journal of Aerospace Engineering*, vol. 2017, 2017, `https://doi.org/10.1155/2017/1235692`.

[158] X. Luo, M. Wang, G. Dai, and X. Chen, "A novel technique to compute the revisit time of satellites and its application in remote sensing satellite optimization design," *International Journal of Aerospace Engineering*, vol. 2017, 2017, `https://doi.org/10.1155/2017/6469439`.

[159] H. W. Lee, P. C. Jakob, K. Ho, S. Shimizu, and S. Yoshikawa, "Optimization of satellite constellation deployment strategy considering uncertain areas of interest," *Acta Astronautica*, vol. 153, pp. 213–228, 2018, `https://doi.org/10.1016/j.actaastro.2018.03.054`, ISSN: 0094-5765.

[160] D.-H. Cho, J.-H. Kim, H.-L. Choi, and J. Ahn, "Optimization-based scheduling method for agile earth-observing satellite constellation," *Journal of Aerospace Information Systems*, vol. 15, no. 11, pp. 611–626, 2018, `https://doi.org/10.2514/1.I010620`.

[161] H. W. Lee, S. Shimizu, S. Yoshikawa, and K. Ho, "Satellite constellation pattern optimization for complex regional coverage," *Journal of Spacecraft and Rockets*, vol. 57, no. 6, pp. 1309–1327, 2020, `https://doi.org/10.2514/1.A34657`.

[162] F. Xhafa and A. W. Ip, "Optimisation problems and resolution methods in satellite scheduling and space-craft operation: A survey," *Enterprise Information Systems*, vol. 15, no. 8, pp. 1022–1045, 2021, `https://doi.org/10.1080/17517575.2019.1593508`.

[163] H. Eschenauer, "Shape optimization of satellite tanks for minimum weight and maximum storage capacity," *Structural optimization*, vol. 1, no. 3, pp. 171–180, 1989, `https://doi.org/10.1007/BF01637337`.

[164] W. Wu, H. Huang, S. Chen, and B. Wu, "Satellite multidisciplinary design optimization with a high-fidelity model," *Journal of Spacecraft and Rockets*, vol. 50, no. 2, pp. 463–466, 2013, `https://doi.org/10.2514/1.A32309`.

[165] S.-t. Yu and C.-z. Fan, "Aerodynamic analysis and drag-reduction design for ultra-low-orbit satellite," in *IOP Conference Series: Materials Science and Engineering*, `http://dx.doi.org/10.1088/1757-899X/887/1/012013`, IOP Publishing, vol. 887, 2020, p. 012 013.

[166] F. Hild, C. Traub, M. Pfeiffer, J. Beyer, and S. Fasoulas, *Optimisation of vleo satellite geometries for drag minimisation and lifetime extension*, `https://arxiv.org/abs/2202.11779`, 2022. arXiv: `2202.11779 [cs.LG]`.

[167] S. Ruder, *An overview of gradient descent optimization algorithms*, `https://arxiv.org/abs/1609.04747`, 2017. arXiv: `1609.04747 [cs.LG]`.

[168] Y. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, *Identifying and attacking the saddle point problem in high-dimensional non-convex optimization*, `https://arxiv.org/abs/1406.2572`, 2014. arXiv: `1406.2572 [cs.LG]`.

[169] K. Sörensen, "Metaheuristics — the metaphor exposed," *International Transactions in Operational Research*, vol. 22, no. 1, pp. 3–18, 2015, `https://doi.org/10.1111/itor.12001`.

[170] L. Bianchi, M. Dorigo, L. M. Gambardella, and W. J. Gutjahr, "A survey on metaheuristics for stochastic combinatorial optimization," *Natural Computing*, vol. 8, pp. 239–287, 2 2009, `https://doi.org/10.1007/s11047-008-9098-4`.

[171] X.-S. Yang, *Engineering Optimization : An Introduction with Metaheuristic Applications*. John Wiley & Sons, Incorporated, 2010, `https://www.wiley.com/en-gb/Engineering+Optimization%3A+An+Introduction+with+Metaheuristic+Applications-p-9780470582466`, ISBN: 9780470640418.

[172] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997, `https://doi.org/10.1109/4235.585893`.

[173] F. Glover and K. Sörensen, "Metaheuristics," *Scholarpedia*, vol. 10, no. 4, p. 6532, 2015, `https://doi.org/10.4249/scholarpedia.6532`.

[174] E. B. d. M. Barbosa and E. L. F. Senne, "Improving the fine-tuning of metaheuristics: An approach combining design of experiments and racing algorithms," *Journal of Optimization*, vol. 2017, 2017, `https://doi.org/10.1155/2017/8042436`.

[175] R. Evins, "A review of computational optimisation methods applied to sustainable building design," *Renewable and Sustainable Energy Reviews*, vol. 22, pp. 230–245, 2013, `https://doi.org/10.1016/j.rser.2013.02.004`.

[176] H. Stegherr, M. Heider, and J. Hähner, "Classifying metaheuristics: Towards a unified multi-level classification system," *Natural Computing*, 2020, `https://doi.org/10.1007/s11047-020-09824-0`.

[177] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Computing Surveys*, vol. 35, no. 3, pp. 268–308, Sep. 2003, `https://doi.org/10.1145/937503.937505`.

[178] E. Talbi, *Metaheuristics: From Design To Implementation*. John Wiley & Sons, Ltd, 2009, pp. 87–307, https://doi.org/10.1002/9780470496916, ISBN: 9780470496916.

[179] J. Swan, S. Adriaensen, M. Bishr, E. Burke, J. Clark, P. De Causmaecker, J. Durillo, K. Hammond, E. Hart, C. Johnson, Z. Kocsis, B. Kovitz, K. Krawiec, S. Martin, J. Merelo, L. Minku, E. Özcan, G. Pappa, E. Pesch, P. Garcia-Sànchez, A. Schaerf, K. Sim, J. Smith, T. Stützle, V. Stefan, S. Wagner, and X. Yao, “A research agenda for metaheuristic standardization,” in *MIC 2015: The XI Metaheuristics International Conference*, https://www.researchgate.net/publication/275956641_A_Research_Agenda_for_Metaheuristic_Standardization, Jul. 2015, pp. 1–3.

[180] M. A. Lones, “Mitigating metaphors: A comprehensible guide to recent nature-inspired algorithms,” *SN Computer Science*, vol. 1, 1 2019, https://doi.org/10.1007/s42979-019-0050-8.

[181] D. Weyland, “A rigorous analysis of the harmony search algorithm: How the research community can be misled by a “novel” methodology,” *International Journal of Applied Metaheuristic Computing (IJAMC)*, vol. 1, no. 2, pp. 50–60, 2010, https://doi.org/10.4018/jamc.2010040104.

[182] I. Fister, A. Iglesias, and A. Galvez, “On detecting the novelties in metaphor-based algorithms,” in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 71–72, https://doi.org/10.1145/3449726.3459413, ISBN: 9781450383516.

[183] A. Tzanetos and G. Dounias, “Nature inspired optimization algorithms or simply variations of metaheuristics?” *Artificial Intelligence Review*, vol. 54, 3 2021, https://doi.org/10.1007/s10462-020-09893-8.

[184] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system: Optimization by a colony of cooperating agents,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996, https://doi.org/10.1109/3477.484436.

[185] M. Dorigo, “Optimization, learning and natural algorithms,” (in Italian), Ph.D. dissertation, Dipartimento di Elettronica, Politecnico di Milano, 1992.

[186] R. Mullen, D. Monekosso, S. Barman, and P. Remagnino, “A review of ant algorithms,” *Expert Systems with Applications*, vol. 36, no. 6, pp. 9608–9617, 2009, https://doi.org/10.1016/j.eswa.2009.01.020, ISSN: 0957-4174.

[187] S. Mirjalili, “Ant colony optimisation,” in *Evolutionary Algorithms and Neural Networks: Theory and Applications*. Cham: Springer International Publishing, 2019, pp. 33–42, https://doi.org/10.1007/978-3-319-93025-1_3, ISBN: 978-3-319-93025-1.

[188] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006, https://doi.org/10.1109/MCI.2006.329691.

[189] V. Selvi and R. Umarani, “Comparative analysis of ant colony and particle swarm optimization techniques,” *International Journal of Computer Applications*, vol. 5, no. 4, pp. 1–6, 2010, https://www.ijcaonline.org/volume5/number4/pxc3871286.pdf.

[190] A. Afshar, F. Massoumi, A. Afshar, and M. A. Mariño, “State of the art review of ant colony optimization applications in water resource management,” *Water Resources Management*, vol. 29, pp. 3891–3904, 11 Sep. 2015, https://doi.org/10.1007/s11269-015-1016-9.

[191] M. Mulani and V. L. Desai, “Design and implementation issues in ant colony optimization,” *International Journal of Applied Engineering Research*, vol. 13, no. 16, pp. 1–11, 2018, https://www.ripublication.com/ijaer18/ijaerv13n16_69.pdf.

[192] S. Kulluk, L. Ozbakir, and A. Baykasoglu, “Training neural networks with harmony search algorithms for classification problems,” *Engineering Applications of Artificial Intelligence*, vol. 25, no. 1, pp. 11–19, 2012, https://doi.org/10.1016/j.engappai.2011.07.006, ISSN: 0952-1976.

[193] A. Askarzadeh and E. Rashedi, “Harmony search algorithm: Basic concepts and engineering applications,” in *Recent Developments in Intelligent Nature-Inspired Computing*. Hershey, PA: IGI Global, 2017, pp. 1–36, https://doi.org/10.4018/978-1-5225-2322-2.ch001, ISBN: 978-3-319-93025-1.

[194] Z. W. Geem, J. H. Kim, and G. Loganathan, “A new heuristic optimization algorithm: Harmony search,” *SIMULATION*, vol. 76, no. 2, pp. 60–68, 2001, https://doi.org/10.1177/003754970107600201.

[195] M. Dubey, V. Kumar, M. Kaur, and T.-P. Dao, "A systematic review on harmony search algorithm: Theory, literature, and applications," *Mathematical Problems in Engineering*, vol. 2021, Apr. 2021, https://doi.org/10.1155/2021/5594267.

[196] X. Z. Gao, V. Govindasamy, H. Xu, X. Wang, and K. Zenger, "Harmony search method: Theory and applications," *Computational Intelligence and Neuroscience*, vol. 2015, Apr. 2015, https://doi.org/10.1155/2015/258491.

[197] W. Sun and X. Chang, "An improved harmony search algorithm for power distribution network planning," *Journal of Electrical and Computer Engineering*, vol. 2015, Feb. 2015, https://doi.org/10.1155/2015/753712.

[198] A. Milad, "Harmony search algorithm: Strengths and weaknesses," *Journal of Computer Engineering & Information Technology*, vol. 2, no. 1, pp. 1–7, 2013, http://dx.doi.org/10.4172/2324-9307.1000102.

[199] L. Abualigah, A. Diabat, and Z. W. Geem, "A comprehensive survey of the harmony search algorithm in clustering applications," *Applied Sciences*, vol. 10, no. 11, p. 3827, 2020, https://doi.org/10.3390/app10113827.

[200] T. K. Sharma, M. Pant, and V. P. Singh, *Improved local search in artificial bee colony using golden section search*, https://arxiv.org/abs/1210.6128, 2012. arXiv: 1210.6128 [cs.AI].

[201] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical report-tr06, Erciyes university, engineering faculty, computer …, Tech. Rep., 2005.

[202] J. R. Riley, U. Greggers, A. D. Smith, D. R. Reynolds, and R. Menzel, "The flight paths of honeybees recruited by the waggle dance," *Nature*, vol. 435, 7039 May 2005, https://doi.org/10.1038/nature03526.

[203] P.-W. TSai, J.-S. Pan, B.-Y. Liao, S.-C. Chu, *et al.*, "Enhanced artificial bee colony optimization," *International Journal of Innovative Computing, Information and Control*, vol. 5, no. 12, pp. 5081–5092, 2009, https://www.researchgate.net/profile/Jeng-Shyang-Pan/publication/228416656_Enhanced_artificial_bee_colony_optimization/links/09e41505c193deced3000000/Enhanced-artificial-bee-colony-optimization.pdf.

[204] X. Zhou, J. Lu, J. Huang, M. Zhong, and M. Wang, “Enhancing artificial bee colony algorithm with multi-elite guidance,” *Information Sciences*, vol. 543, pp. 242–258, 2021, https://doi.org/10.1016/j.ins.2020.07.037, ISSN: 0020-0255.

[205] S. Khan, M. Asjad, and A. Ahmad, “Review of modern optimization techniques,” *International Journal of Engineering Research and Technology*, vol. 4, no. 4, pp. 2278–0181, 2015, http://dx.doi.org/10.17577/IJERTV4IS041129.

[206] I. J. Jacob and P. E. Darney, “Artificial bee colony optimization algorithm for enhancing routing in wireless networks,” *Journal of Artificial Intelligence*, vol. 3, no. 01, pp. 62–71, 2021, https://doi.org/10.36548/jaicn.2021.1.006.

[207] F. S. Abu-Mouti and M. E. El-Hawary, “Optimal distributed generation allocation and sizing in distribution systems via artificial bee colony algorithm,” *IEEE Transactions on Power Delivery*, vol. 26, no. 4, pp. 2090–2101, 2011, https://doi.org/10.1109/TPWRD.2011.2158246.

[208] P. Guo, W. Cheng, and J. Liang, “Global artificial bee colony search algorithm for numerical function optimization,” in *2011 Seventh International Conference on Natural Computation*, https://doi.org/10.1109/ICNC.2011.6022368, vol. 3, 2011, pp. 1280–1283.

[209] Y.-H. Kim, K. H. Lee, and Y. Yoon, “Visualizing the search process of particle swarm optimization,” in *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO ’09, https://doi.org/10.1145/1569901.1569909, Montreal, Québec, Canada: Association for Computing Machinery, 2009, pp. 49–56, ISBN: 9781605583259.

[210] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, https://doi.org/10.1109/ICNN.1995.488968, vol. 4, 1995, 1942–1948 vol.4.

[211] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, https://doi.org/10.1109/ICEC.1998.699146, 1998, pp. 69–73.

[212] M. R. Bonyadi and Z. Michalewicz, “Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review,” *Evolutionary Computation*, vol. 25, no. 1, pp. 1–54, Mar. 2017, https://doi.org/10.1162/EVCO_r_00180, ISSN: 1063-6560.

[213] Q. Bai, “Analysis of particle swarm optimization algorithm,” *Computer and information science*, vol. 3, no. 1, p. 180, 2010, https://doi.org/10.5539/cis.v3n1p180.

[214] K. Y. Lee and J.-b. Park, “Application of particle swarm optimization to economic dispatch problem: Advantages and disadvantages,” in *2006 IEEE PES Power Systems Conference and Exposition*, https://doi.org/10.1109/PSCE.2006.296295, 2006, pp. 188–192.

[215] Autodesk, *Generative design primer*, [Online; date accessed: 28/02/2022], 2021. [Online]. Available: https://www.generativedesign.org/02-deeper-dive/02-04_genetic-algorithms/02-04-01_what-is-a-genetic-algorithm.

[216] A. M. TURING, “I.—COMPUTING MACHINERY AND INTELLIGENCE,” *Mind*, vol. LIX, no. 236, pp. 433–460, Oct. 1950, https://doi.org/10.1093/mind/LIX.236.433, ISSN: 0026-4423.

[217] A. Fraser, “Simulation of genetic systems,” *Journal of Theoretical Biology*, vol. 2, no. 3, pp. 329–346, 1962, https://doi.org/10.1016/0022-5193(62)90036-X, ISSN: 0022-5193.

[218] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992, https://dl.acm.org/doi/10.5555/531075, ISBN: 0262082136.

[219] K. A. De Jong, “An analysis of the behavior of a class of genetic adaptive systems.,” Ph.D. dissertation, University of Michigan, 1975.

[220] D. A. Coley, *An Introduction to Genetic Algorithms for Scientists and Engineers*. Singapore: WORLD SCIENTIFIC, 1999, https://doi.org/10.1142/3904.

[221] P. C. Chu, “A genetic algorithm approach for combinatorial optimisation problems,” https://spiral.imperial.ac.uk/bitstream/10044/1/11491/2/Chu-PCH-1997-PhD-Thesis.pdf, Ph.D. dissertation, Imperial College London, 1997.

[222] H. Tamaki, H. Kita, and S. Kobayashi, “Multi-objective optimization by genetic algorithms: A review,” in *Proceedings of IEEE International Conference on Evolutionary Computation*, https://doi.org/10.1109/ICEC.1996.542653, 1996, pp. 517–522.

[223] N. M. Razali, J. Geraghty, *et al.*, "Genetic algorithm performance with different selection strategies in solving tsp," in *Proceedings of the world congress on engineering*, http://www.iaeng.org/publication/WCE2011/WCE2011_pp1134-1139.pdf, vol. 2, 2011, pp. 1–6.

[224] A. Lambora, K. Gupta, and K. Chopra, "Genetic algorithm- a literature review," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, https://doi.org/10.1109/COMITCon.2019.8862255, 2019, pp. 380–384.

[225] S. Sivanandam and S. Deepa, *Introduction to Genetic Algorithms*. New York: Springer-Verlag Berlin Heidelberg, 2008, https://doi.org/10.1007/978-3-540-73190-0_10, ISBN: 9783540731900.

[226] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: Past, present, and future," *Multimedia Tools and Applications*, vol. 80, 5 2021, https://doi.org/10.1007/s11042-020-10139-6.

[227] A. K. Kar, "Bio inspired computing – a review of algorithms and scope of applications," *Expert Systems with Applications*, vol. 59, pp. 20–32, 2016, https://doi.org/10.1016/j.eswa.2016.04.018, ISSN: 0957-4174.

[228] A. J. Umbarkar and P. D. Sheth, "Crossover operators in genetic algorithms: A review.," *ICTACT journal on soft computing*, vol. 6, no. 1, 2015, https://doi.org/10.5120/ijca2017913370.

[229] MathWorks, *Genetic algorithm options*, [Online; date accessed: 02/03/2022], 2022. [Online]. Available: https://uk.mathworks.com/help/gads/genetic-algorithm-options.html#f7820.

[230] B. L. Miller and D. E. Goldberg, "Genetic algorithms, tournament selection, and the effects of noise," *Complex Systems*, vol. 9, no. 3, pp. 193–212, 1995, https://wpmedia.wolfram.com/uploads/sites/13/2018/02/09-3-2.pdf.

[231] M. Srinivas and L. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 4, pp. 656–667, 1994.

[232] S. Marsili Libelli and P. Alba, "Adaptive mutation in genetic algorithms," *Soft Computing*, vol. 4, 2 Jul. 2000, https://doi.org/10.1007/s005000000042.

[233] C. Hwang, S. Paidy, A. Masud, and K. Yoon, *Multiple Objective Decision Making — Methods and Applications: A State-of-the-Art Survey*, ser. Lecture Notes in Economics and Mathematical Systems. Springer Berlin Heidelberg, 2012, `https://books.google.co.uk/books?id=MOnoCAAAQBAJ`, ISBN: 9783642455117.

[234] F. B. Abdelaziz, "Solution approaches for the multiobjective stochastic programming," *European Journal of Operational Research*, vol. 216, no. 1, pp. 1–16, 2012, `https://doi.org/10.1016/j.ejor.2011.03.033`, ISSN: 0377-2217.

[235] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability Engineering & System Safety*, vol. 91, no. 9, pp. 992–1007, 2006, `https://doi.org/10.1016/j.ress.2005.11.018`, ISSN: 0951-8320.

[236] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization: New insights," *Structural and Multidisciplinary Optimization*, vol. 41, 6 Jun. 2010, `https://doi.org/10.1007/s00158-009-0460-7`.

[237] E. Hannan, "Nondominance in goal programming," *INFOR: Information Systems and Operational Research*, vol. 18, no. 4, pp. 300–309, 1980, `https://doi.org/10.1080/03155986.1980.11731801`.

[238] M. Tamiz and D. F. Jones, "Goal programming and pareto efficiency," *Journal of Information and Optimization Sciences*, vol. 17, no. 2, pp. 291–307, 1996, `https://doi.org/10.1080/02522667.1996.10699283`.

[239] R. Marler and J. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and Multidisciplinary Optimization*, vol. 26, 6 Apr. 2004, `https://doi.org/10.1007/s00158-003-0368-6`.

[240] J. L. J. Pereira, G. A. Oliver, M. B. Francisco, S. S. Cunha, and G. F. Gomes, "A review of multi-objective optimization: Methods and algorithms in mechanical engineering problems," *Archives of Computational Methods in Engineering*, Oct. 2021, `https://doi.org/10.1007/s11831-021-09663-x`.

[241] H. Tamaki, H. Kita, and S. Kobayashi, "Multi-objective optimization by genetic algorithms: A review," in *Proceedings of IEEE International Conference on Evolutionary Computation*, `https://doi.org/10.1109/ICEC.1996.542653`, 1996, pp. 517–522.

[242] D. Jones, S. Mirrazavi, and M. Tamiz, “Multi-objective meta-heuristics: An overview of the current state-of-the-art,” *European Journal of Operational Research*, vol. 137, no. 1, pp. 1–9, 2002, https://doi.org/10.1016/S0377-2217(01)00123-0, ISSN: 0377-2217.

[243] K. Deb and T. Goel, “Controlled elitist non-dominated sorting genetic algorithms for better convergence,” in *Evolutionary Multi-Criterion Optimization*, E. Zitzler, L. Thiele, K. Deb, C. A. Coello Coello, and D. Corne, Eds., https://doi.org/10.1007/3-540-44719-9_5, Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 67–81, ISBN: 978-3-540-44719-1.

[244] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002, https://doi.org/10.1109/4235.996017.

[245] P. S. Bharti, S. Maheshwari, and C. Sharma, “Multi-objective optimization of electric-discharge machining process using controlled elitist NSGA-II,” *Journal of Mechanical Science and Technology*, vol. 26, pp. 1875–1883, 6 Jun. 2012, https://doi.org/10.1007/s12206-012-0411-x.

[246] R. Derksen and T. Rogalsky, “Bezier-parsec: An optimized aerofoil parameterization for design,” *Advances in Engineering Software*, vol. 41, no. 7, pp. 923–930, 2010, https://doi.org/10.1016/j.advengsoft.2010.05.002, ISSN: 0965-9978.

[247] X. Wei, X. Wang, and S. Chen, “Research on parameterization and optimization procedure of low-reynolds-number airfoils based on genetic algorithm and bezier curve,” *Advances in Engineering Software*, vol. 149, p. 102 864, 2020, https://doi.org/10.1016/j.advengsoft.2020.102864, ISSN: 0965-9978.

[248] J. Lepine, J.-Y. Trepanier, and F. Pepin, “Wing aerodynamic design using an optimized nurbs geometrical representation,” in *38th Aerospace Sciences Meeting and Exhibit*, https://arc.aiaa.org/doi/pdf/10.2514/6.2000-669, 2000, p. 669.

[249] J. Li, Z. Gao, J. Huang, and K. Zhao, “Robust design of NLF airfoils,” *Chinese Journal of Aeronautics*, vol. 26, no. 2, pp. 309–318, 2013, https://doi.org/10.1016/j.cja.2013.02.007, ISSN: 1000-9361.

[250] N. P. Salunke, J. A. R. A., and S. Channiwala, “Airfoil parameterization techniques: A review,” *American Journal of Mechanical Engineering*, vol. 2, no. 4, pp. 99–102, 2014, https://www.doi.org/10.12691/ajme-2-4-1.

[251] Adam A., *Mesh voxelisation*, https://uk.mathworks.com/matlabcentral/fileexchange/27390-mesh-voxelisation, Matlab Central File Exchange. [Online; date accessed: 23/03/2022], 2022.

[252] S. Patil and B. Ravi, “Voxel-based representation, display and thickness analysis of intricate shapes,” in *Ninth International Conference on Computer Aided Design and Computer Graphics (CAD-CG’05)*, https://www.doi.org/10.1109/CAD-CG.2005.86, 2005, p. 6.

[253] D. Mostaza-Prieto and P. C. Roberts, “Methodology to analyze attitude stability of satellites subjected to aerodynamic torques,” *Journal of guidance, control, and dynamics*, vol. 39, no. 3, pp. 437–449, 2016, https://doi.org/10.2514/1.G001481.

[254] M. Chari and S. Salon, “3 - the finite difference method,” in *Numerical Methods in Electromagnetism*, ser. Electromagnetism, M. Chari and S. Salon, Eds., https://doi.org/10.1016/B978-012615760-4/50004-3, San Diego: Academic Press, 2000, pp. 105–141, ISBN: 978-0-12-615760-4.

[255] S. Rao, “Finite difference methods,” in *Encyclopedia of Vibration*, S. Braun, Ed., https://doi.org/10.1006/rwvb.2001.0002, Oxford: Elsevier, 2001, pp. 520–530, ISBN: 978-0-12-227085-7.

[256] X. Shi, R. Li, and Y. Sheng, “A new volume-based convexity measure for 3d shapes,” in *Computer Graphics International Conference*, N. Magnenat-Thalmann, C. Stephanidis, E. Wu, D. Thalmann, B. Sheng, J. Kim, G. Papagiannakis, and M. Gavrilova, Eds., http://dx.doi.org/10.1007/978-3-030-61864-3_6, Springer International Publishing, 2020, pp. 59–70, ISBN: 978-3-030-61864-3.

[257] T. Möller, “A fast triangle-triangle intersection test,” *Journal of Graphics Tools*, vol. 2, no. 2, pp. 25–30, 1997, https://doi.org/10.1080/10867651.1997.10487472.

[258] O. Tropp, A. Tal, and I. Shimshoni, “A fast triangle to triangle intersection test for collision detection,” *Computer Animation and Virtual Worlds*, vol. 17, no. 5, pp. 527–535, 2006, https://doi.org/10.1002/cav.115.

[259] H. Shen, P. A. Heng, and Z. Tang, “A fast triangle-triangle overlap test using signed distances,” *Journal of Graphics Tools*, vol. 8, no. 1, pp. 17–23, 2003, https://www.tandfonline.com/doi/abs/10.1080/10867651.2003.10487579.

[260] O. Devillers and P. Guigue, “Faster Triangle-Triangle Intersection Tests,” INRIA, Tech. Rep. RR-4488, Jun. 2002, https://hal.inria.fr/inria-00072100.

[261] C. L. Sabharwal and J. L. Leopold, “A triangle-triangle intersection algorithm,” *Computer Science & Information Technology*, vol. 5, no. 11, pp. 27–35, 2015, https://doi.org/10.5121/csit.2015.51003.

[262] ——, “A generic design for implementing intersection between triangles in computer vision and spatial reasoning,” in *Innovative Research in Attention Modeling and Computer Vision Applications*, R. Pal, Ed. IGI Global, 2016, ch. 8, pp. 195–235, https://doi.org/10.4018/978-1-4666-8723-3.ch008.

[263] D. Johnson, *Triangle triangle contact*, https://uk.mathworks.com/matlabcentral/fileexchange/38135-triangle-triangle-contact?s_tid=prof_contriblnk. [Online; date accessed: 3/04/2022], 2022.

[264] O. Grodzevich and O. Romanko, “Normalization and other topics in multi-objective optimization,” in *Proceedings of the Fields–MITACS Industrial Problems Workshop, 2006*, http://miis.maths.ox.ac.uk/miis/233/1/fmipw1-6.pdf, 2006.

[265] A. Szparaga, M. Stachnik, E. Czerwińska, S. Kocira, M. Dymkowska-Malesa, and M. Jakubowski, “Multi-objective optimization based on the utopian point method applied to a case study of osmotic dehydration of plums and its storage,” *Journal of Food Engineering*, vol. 245, pp. 104–111, 2019, https://doi.org/10.1016/j.jfoodeng.2018.10.014, ISSN: 0260-8774.

[266] M. Martínez-Iranzo, J. M. Herrero, J. Sanchis, X. Blasco, and S. García-Nieto, “Applied pareto multi-objective optimization by stochastic solvers,” *Engineering Applications of Artificial Intelligence*, vol. 22, no. 3, pp. 455–465, 2009, https://doi.org/10.1016/j.engappai.2008.10.018, ISSN: 0952-1976.

[267] P. Diaz-Gomez and D. Hougen, “Initial population for genetic algorithms: A metric approach.,” in *Proceedings of the 2007 International Conference on Genetic and Evolutionary Methods, GEM 2007*, https://www.researchgate.net/publication/220862320_Initial_Population_for_Genetic_Algorithms_A_Metric_Approach, Jan. 2007, pp. 43–49.

[268] R. R. Hill, “A monte-carlo study of genetic algorithm initial population generation methods,” in *Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future-Volume 1*, https://doi.org/10.1145/324138.324430, 1999, pp. 543–547.

[269] S. Sivanandam and S. Deepa, “Genetic algorithms,” in *Introduction to Genetic Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 15–37, https://doi.org/10.1007/978-3-540-73190-0_2, ISBN: 978-3-540-73190-0.

[270] G. Smith and S. Smith, “Automated initial population generation for genetic assembly planning,” *International Journal of Computer Integrated Manufacturing*, vol. 16, no. 3, pp. 219–228, 2003, https://doi.org/10.1080/0951192021000039602.

[271] N. H. Crisp, “A methodology for the integrated design of small satellite constellation deployment,” Ph.D. dissertation, The University of Manchester, Dec. 2016.

[272] S. Luthcke, J. Marshall, S. Rowton, K. Rachlin, C. Cox, and R. Williamson, “Enhanced radiative force modeling of the tracking and data relay satellites,” *The Journal of the Astronautical Sciences*, vol. 45, pp. 349–370, 3 Sep. 1997, https://doi.org/10.1007/BF03546409.

Appendices

Appendix A

Ancillary results

For clarity and brevity, additional graphs of results which are practically identical to those discussed in chapter 7 are not given therein. These further results are captured in the following figures. They are complementary to those discussed in-depth in chapter 7. The figures in this appendix relate to the optimisation cases as follows:

- figs. A.1 and A.2 refer to WB optimisation for case 2 of table 7.1.
- figs. A.3 and A.4 refer to WB optimisation for case 3 of table 7.1.
- figs. A.5 and A.6 refer to WB optimisation for case 4 of table 7.1.
- figs. A.7 to A.10 refer to F&B optimisation for case 1 of table 7.1, aiming for neutral stability.
- figs. A.11 to A.13 refer to WB optimisation for case 1 of table 7.1, aiming for neutral stability.

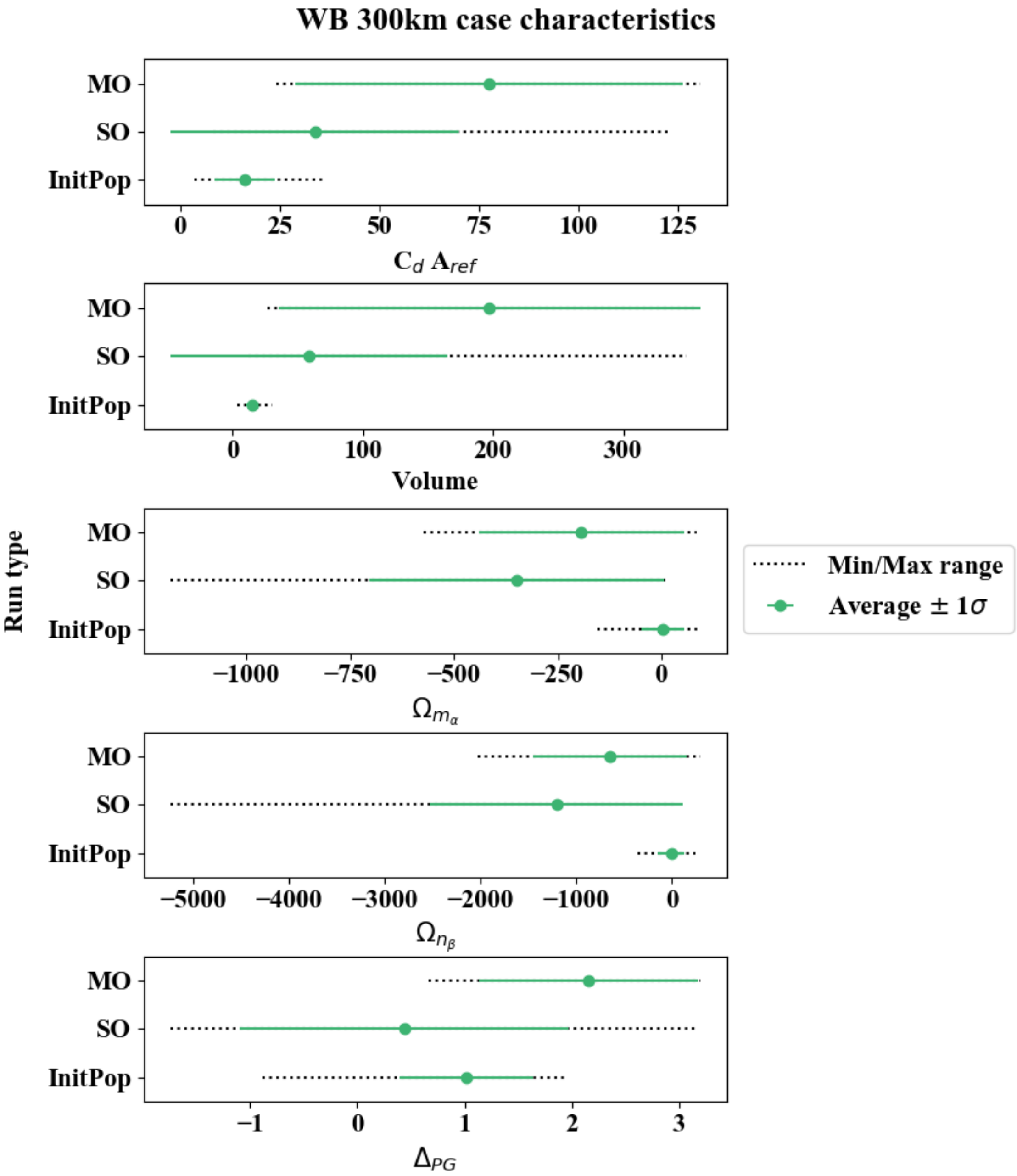


Figure A.1. Comparison of initial population, SO and MO for 300 km altitude optimisation.

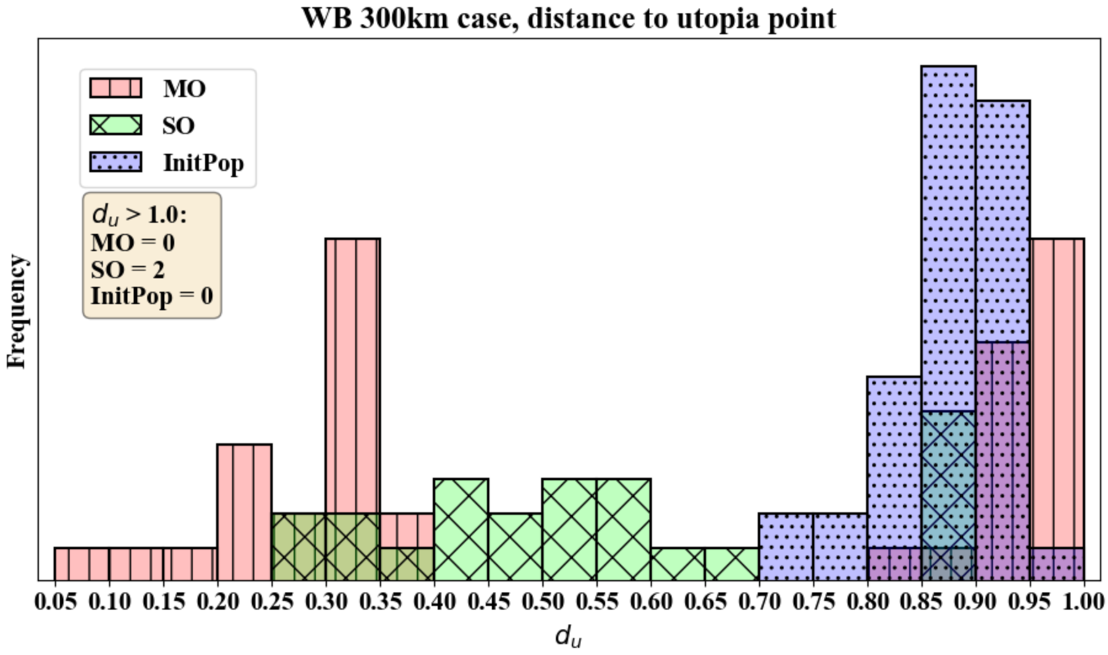


Figure A.2. Distance to utopia point for initial population, SO and MO for 300 km altitude optimisation.

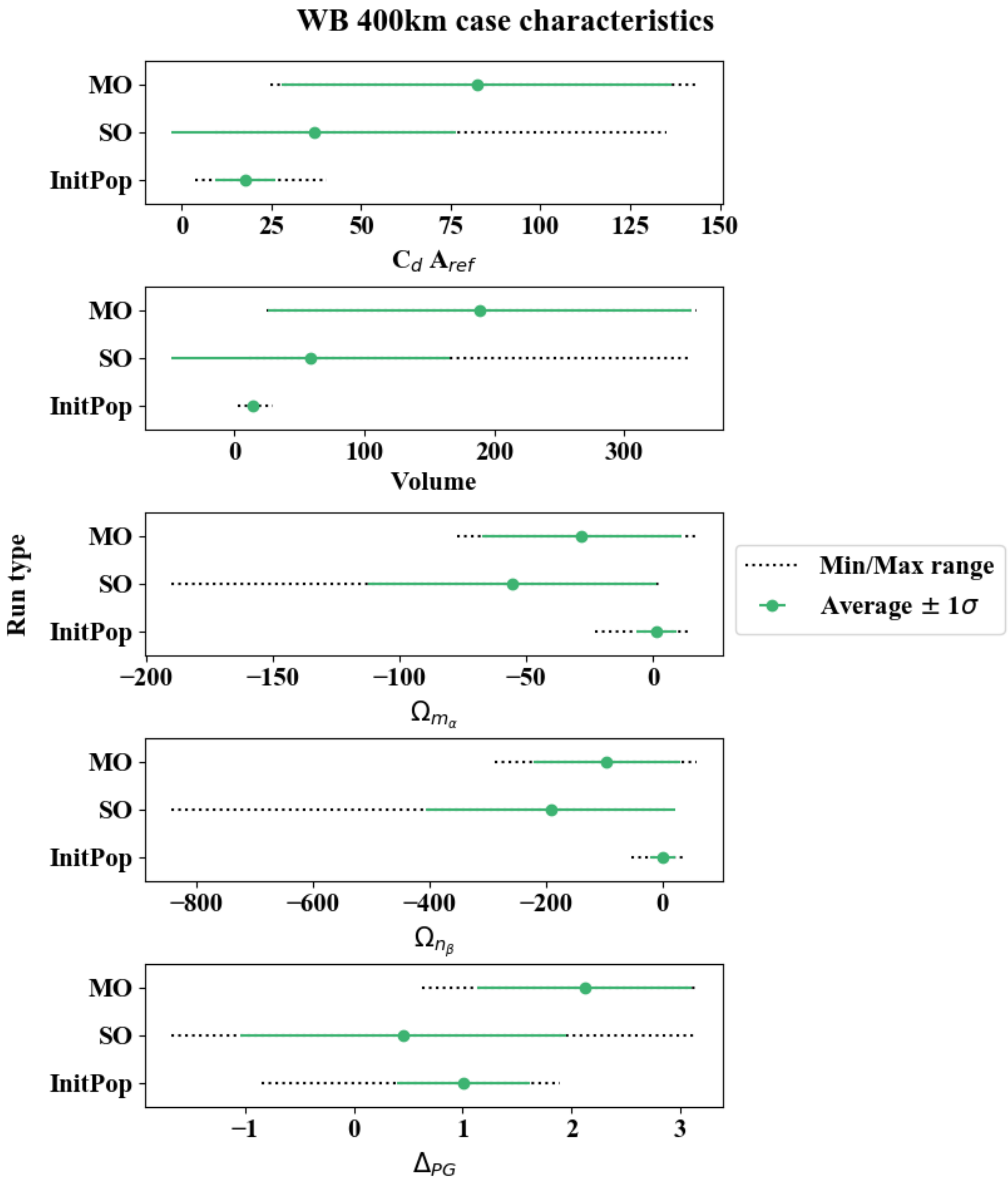


Figure A.3. Comparison of initial population, SO and MO for 400 km altitude optimisation.

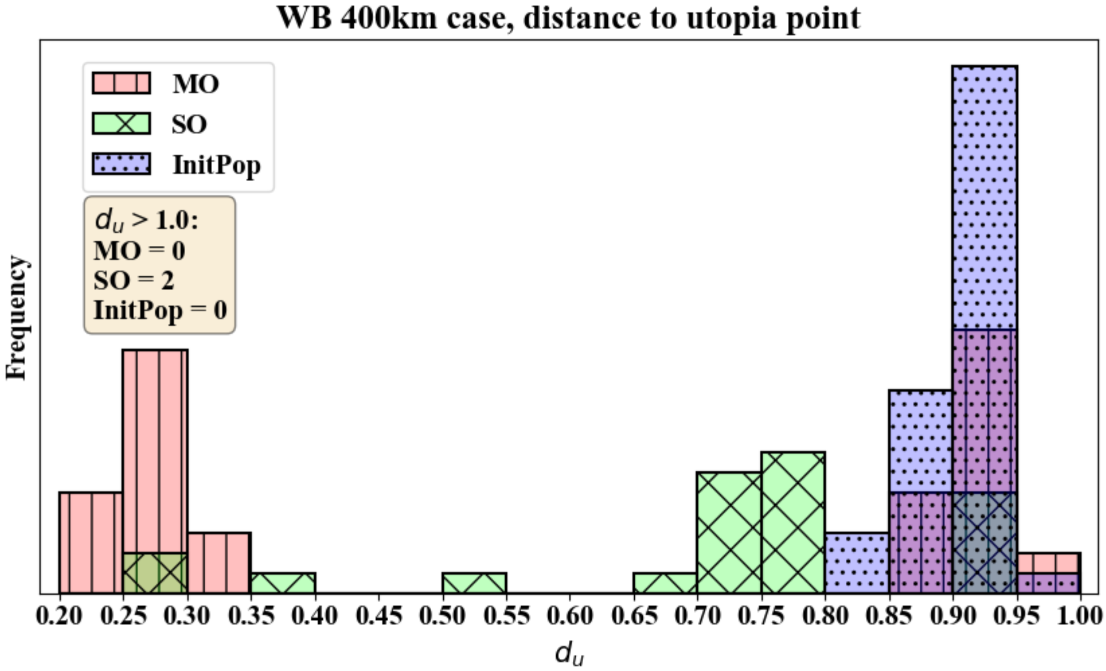


Figure A.4. Distance to utopia point for initial population, SO and MO for 400 km altitude optimisation.

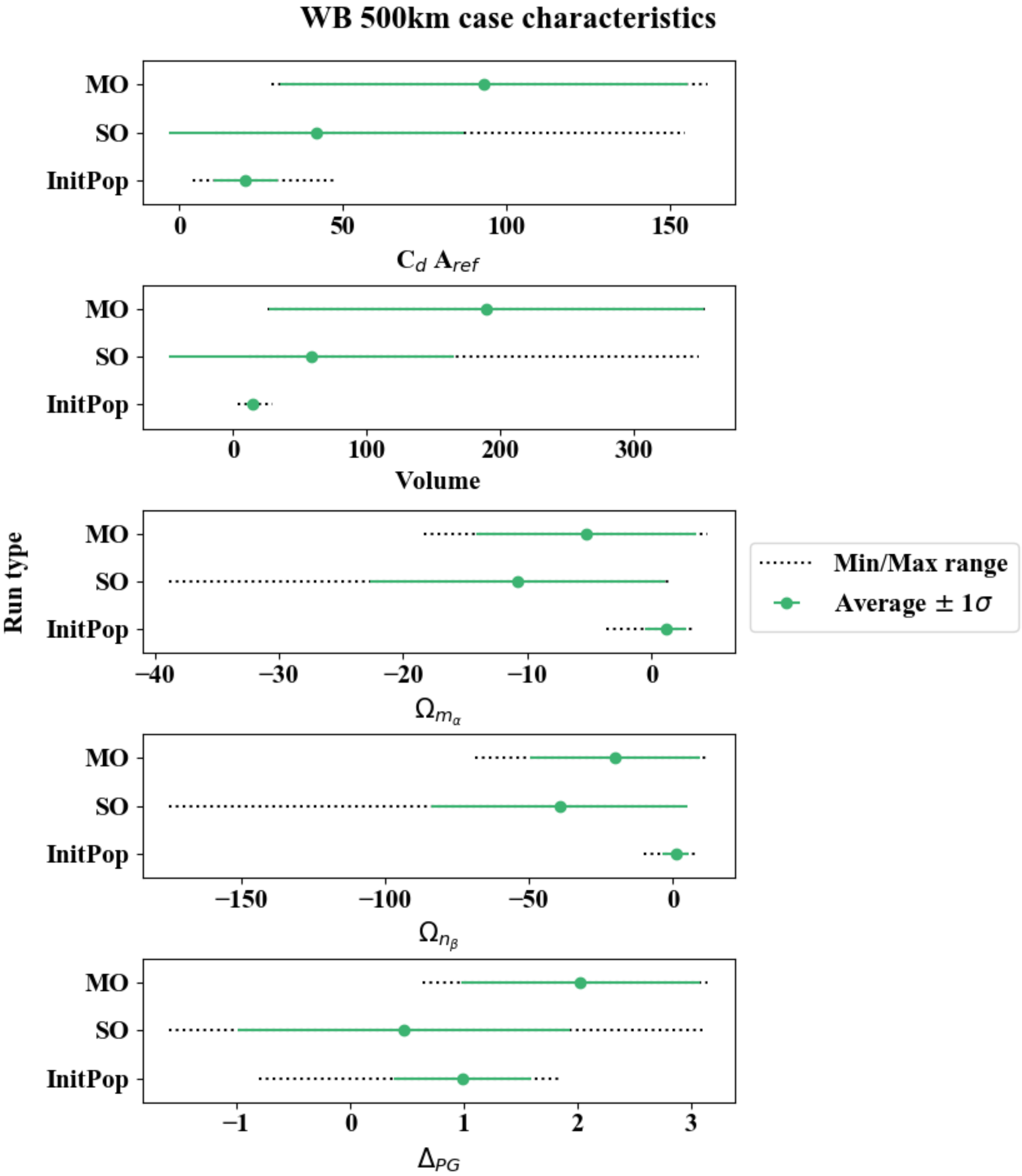


Figure A.5. Comparison of initial population, SO and MO for 500 km altitude optimisation.

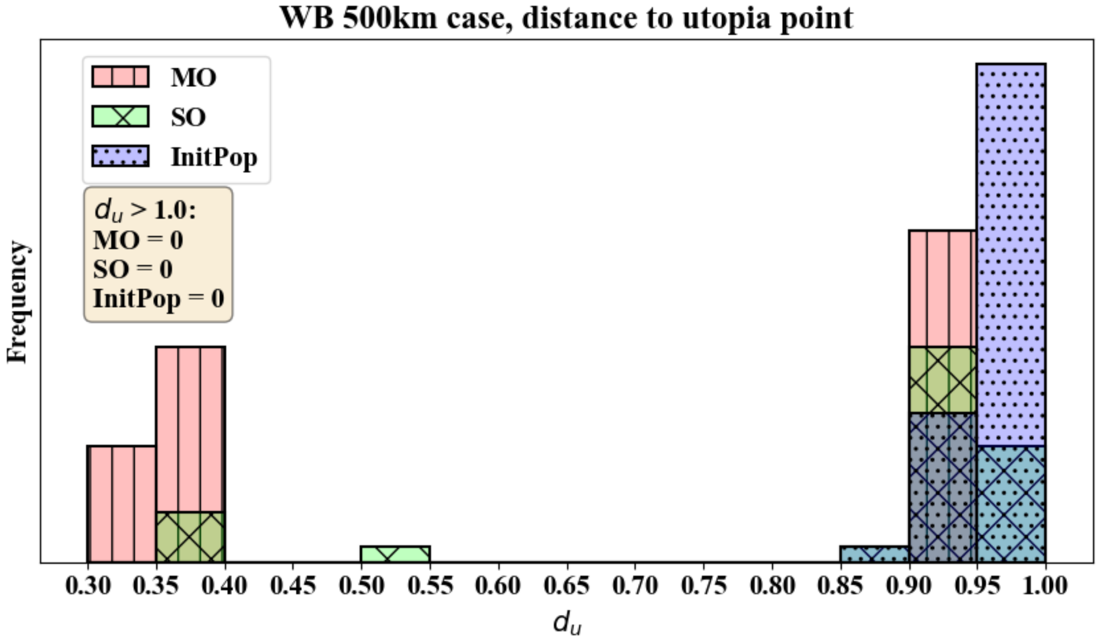


Figure A.6. Distance to utopia point for initial population, SO and MO for 500 km altitude optimisation.

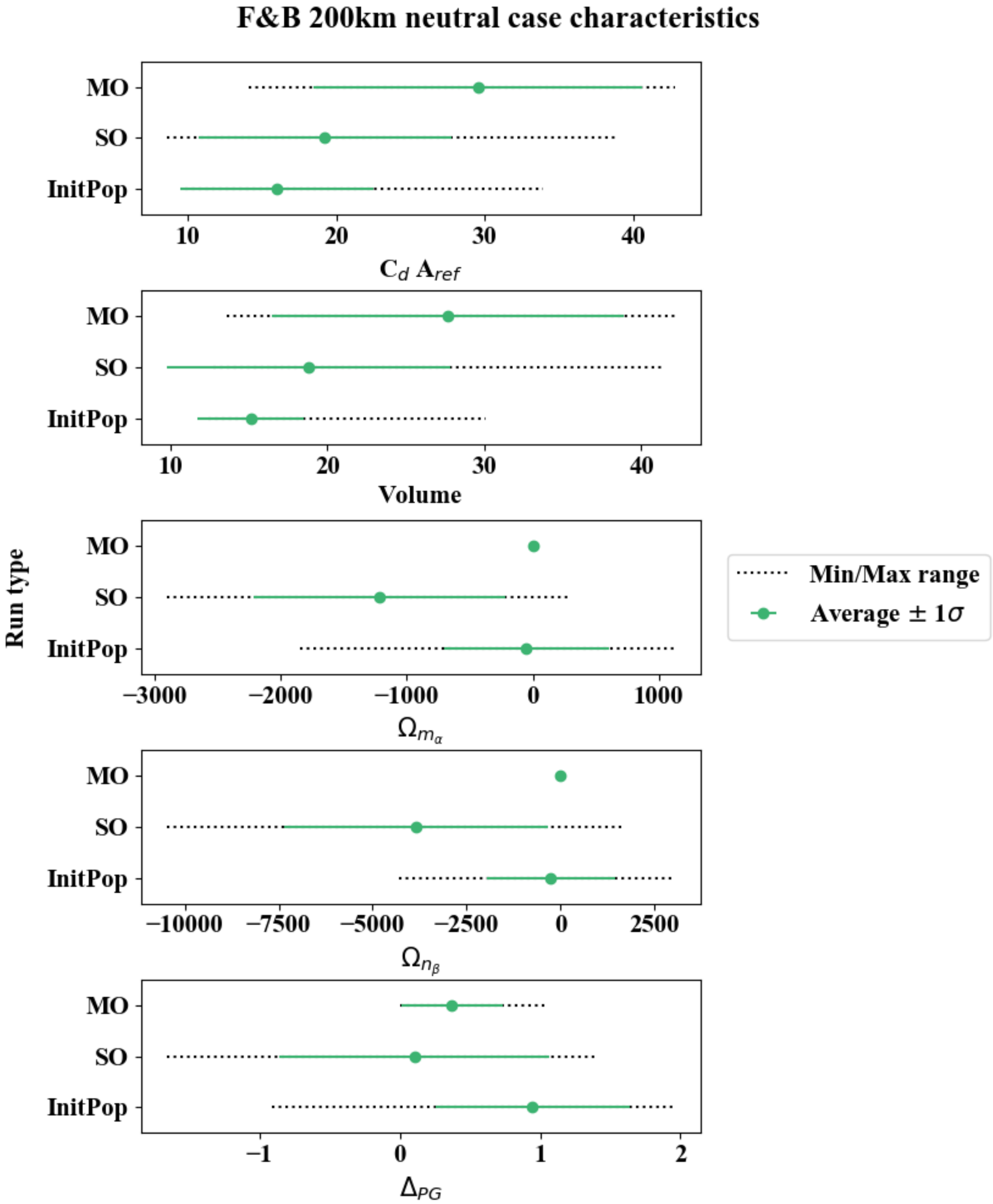


Figure A.7. Comparison for the F&T case of initial population, SO and MO for 200 km altitude optimisation, aiming for neutral stability.

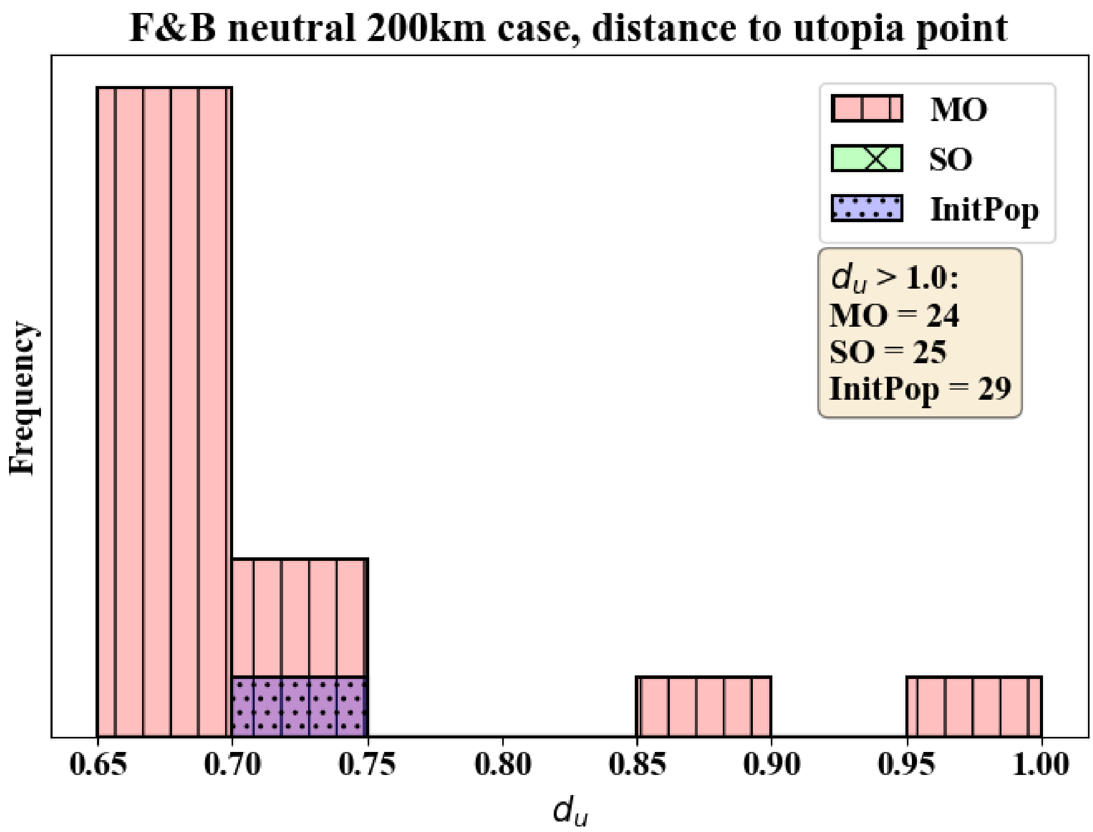


Figure A.8. Distribution of d_u for individuals for the F&B case, aiming for neutral stability, in the range $0 < d_u < 1$.

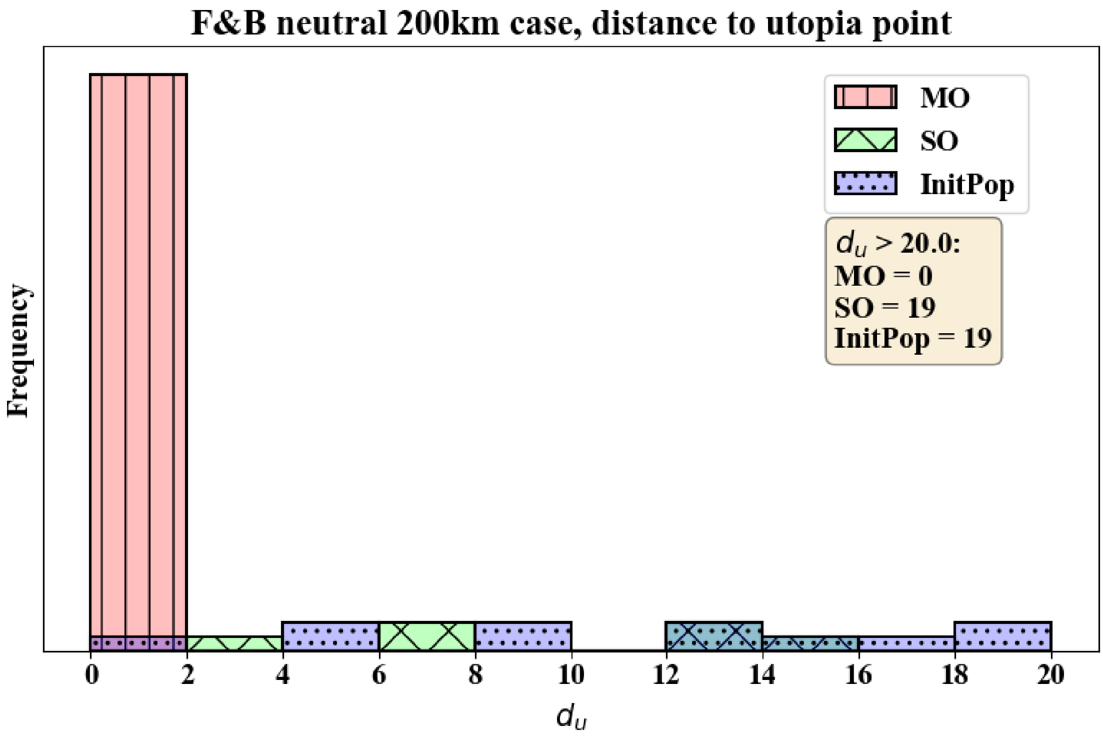


Figure A.9. Distribution of d_u for individuals for the F&B case, aiming for neutral stability, in the range $0 < d_u < 20$.

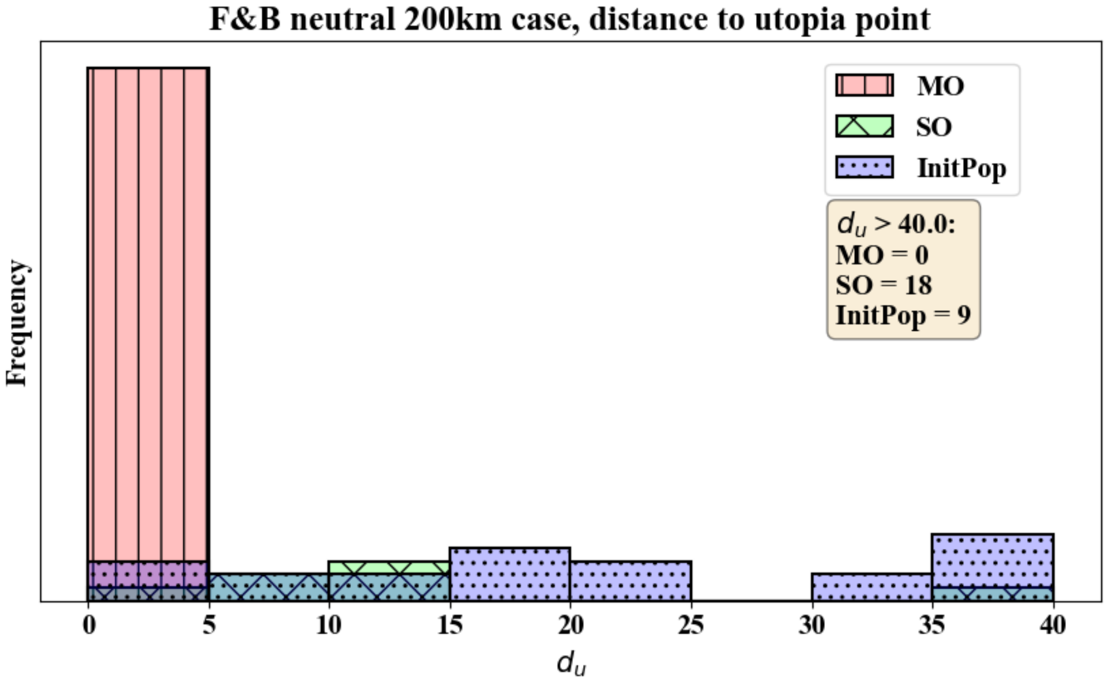


Figure A.10. Distribution of d_u for individuals for the F&B case, aiming for neutral stability, in the range $0 < d_u < 40$.

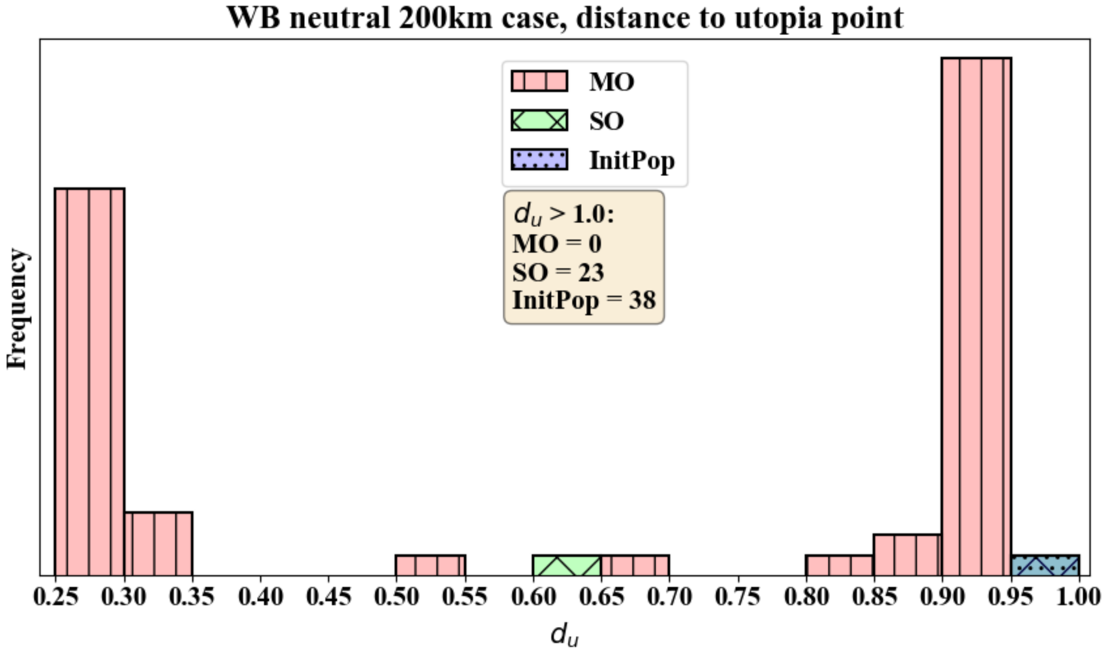


Figure A.11. Distribution of d_u for individuals for the WB case, aiming for neutral stability, in the range $0 < d_u < 1$.

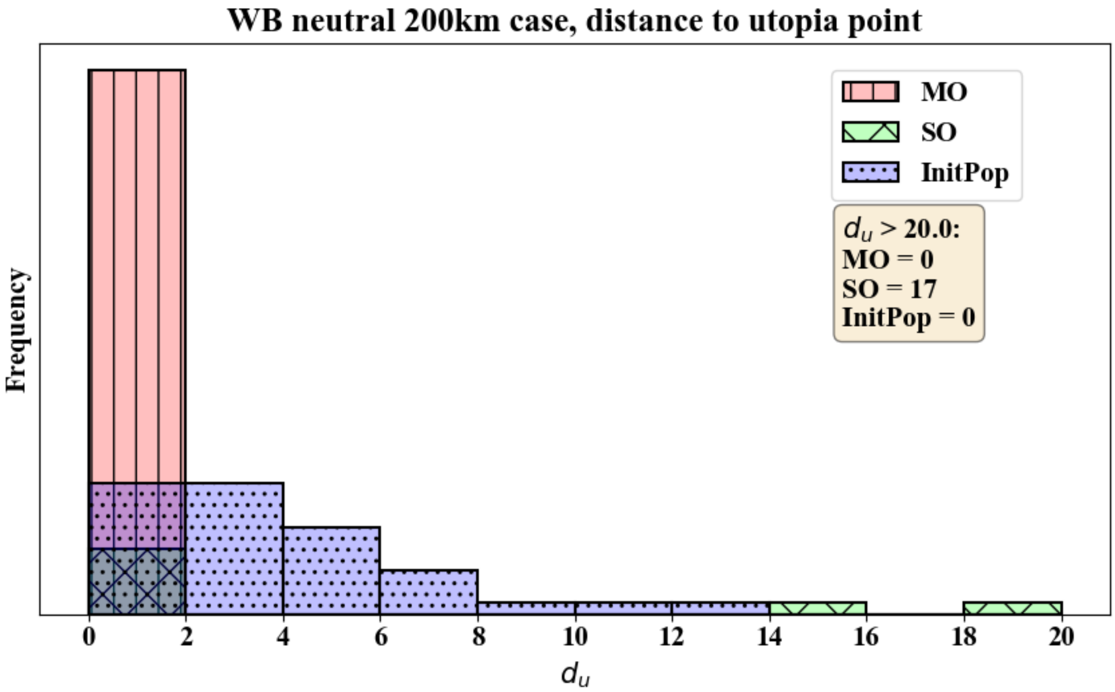


Figure A.12. Distribution of d_u for individuals for the WB case, aiming for neutral stability, in the range $0 < d_u < 20$.

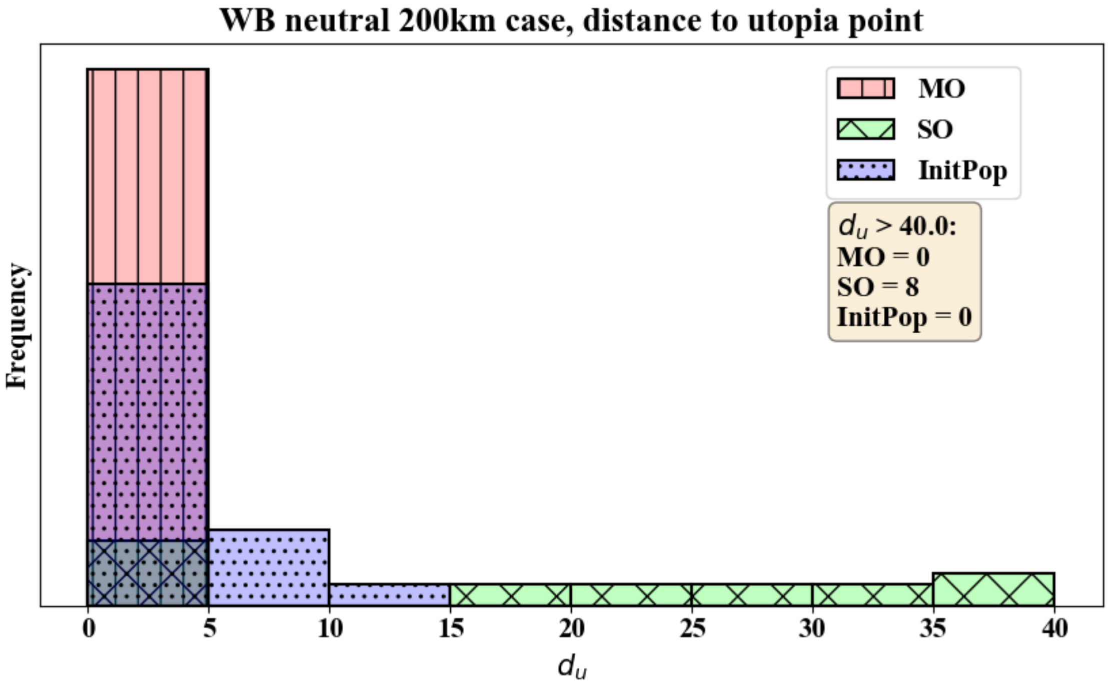


Figure A.13. Distribution of d_u for individuals for the WB case, aiming for neutral stability, in the range $0 < d_u < 40$.