WRITING YOUR THESIS WITH LATEX WITH A VERY, VERY, VERY LONG TITLE

LIM LIAN TZE

UNIVERSITI SAINS MALAYSIA

2015

WRITING YOUR THESIS WITH LATEX WITH A VERY, VERY, VERY LONG TITLE

by

LIM LIAN TZE

**Thesis submitted in fulfilment of the requirements
for the degree of
Doctor of Philosphy**

December 2015

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF PLATES

LIST OF ABBREVIATIONS

IPS	Institut Pengajian Siswazah
PPSK	Pusat Pengajian Sains Komputer
USM	Universiti Sains Malaysia
UTMK	Unit Terjemahan Melalui Komputer

LIST OF SYMBOLS

lim	limit
θ	angle in radians

PENULISAN TESIS DENGAN LATEX

ABSTRAK

Ini merupakan abstrak Melayu untuk tesis USM. Ianya disediakan dengan sistem penyediaan dokumen LaTeX.

WRITING YOUR THESIS WITH LATEX WITH A VERY, VERY, VERY LONG TITLE

ABSTRACT

This is the English abstract of a USM thesis. It was prepared with the LaTeX document typesetting system.

CHAPTER 1

INTRODUCTION

1.1 Biosecurity and Policies

The global community faces an escalating array of biological threats that transcend traditional boundaries between nations, economic sectors, and scientific disciplines. The COVID-19 pandemic demonstrated these threats' devastating consequences, causing millions of deaths worldwide and imposing trillions of dollars in economic losses (Hulme, 2021; World Health Organization, 2021). Modern society's interconnected nature—characterized by rapid international travel and extensive trade networks—means biological threats can spread with alarming speed, requiring coordinated responses that integrate public health measures, border controls, surveillance systems, and emergency management protocols (World Trade Organization, 2024).

The World Health Organization defines biosecurity as a strategic and integrated approach to analyzing and managing relevant risks to human, animal, and plant life and health, along with associated environmental risks (World Health Organization, 2021). This holistic perspective recognizes interconnections among human health, animal health, plant health, and ecosystem integrity, often referred to as the One Health approach (Hulme, 2021). The Food and Agriculture Organization emphasizes that biosecurity encompasses policy and regulatory frameworks that analyze and manage risks in food safety, animal life and health, and plant life and health (Food and Agriculture Organization, 2023). Effective biosecurity requires coordination across multiple sectors including agriculture, veterinary medicine, public health, environmental protection, customs and border control, transportation, and increasingly, information technology and cybersecurity.

Biosecurity governance operates through a complex architecture of international agreements and national regulations. At the global level, the World Health Organization's International Health Regulations establish binding obligations for member states to develop core capacities for detecting, assessing, reporting, and responding to public health emergencies (World Health Organization, 2021). The World Organisation for Animal Health, the Codex Alimentarius Commission, and other international bodies develop standards that member states use as reference points for managing biosecurity risks (World Trade Organization, 2024). The Biological Weapons Convention prohibits the development, production, and stockpiling of biological weapons, establishing a normative foundation against the malicious use of biological agents (United Nations Office for Disarmament Affairs, 2023).

Malaysia has implemented comprehensive biosecurity policies reflecting both its exposure to biological threats and its strategic position in global trade networks. As a tropical country with extensive agricultural production and a major exporter of agricultural commodities including palm oil and rubber, Malaysia faces biosecurity risks from endemic diseases, imported pests and pathogens, and potential bioterrorism (Ministry of Health Malaysia, 2023). The country's location along the Strait of Malacca—through which approximately one-quarter of global maritime trade transits—positions it as a critical node in international supply chains and a potential entry point for biological threats (Association of Southeast Asian Nations, 2024).

Recognizing cybersecurity's growing importance to national security and economic competitiveness, Malaysia enacted the Cybersecurity Act 2024, which came into force on 26 August 2024 (Government of Malaysia, 2024). This landmark legislation establishes a comprehensive framework for managing cybersecurity risks to critical information infrastructure, including systems supporting biosecurity functions. The Act designates the National Cyber Security Agency as the lead authority for coordinating cybersecurity policy and incident response, with powers to designate systems as National Critical Information Infrastructure and to require their operators to implement cybersecurity measures, report incidents, and undergo regular audits (National Cyber Security Agency Malaysia, 2024). The legislation applies extraterritorially to cybersecurity service providers whose activities affect Malaysia's cybersecurity, reflecting cyber threats' transnational nature (CMS Law Offices, 2024).

Malaysia has also established regulatory frameworks for unmanned aerial systems. The Civil Aviation Authority of Malaysia enforces comprehensive drone regulations under the Malaysian Civil Aviation Regulations 2016, which categorize unmanned aircraft based on weight and operational characteristics (Civil Aviation Authority of Malaysia, 2024). Small Unmanned Aircraft Systems weighing up to 20 kilograms are subject to relatively permissive regulations for recreational use, while commercial operations require permits with specific requirements for operator qualifications, aircraft certification, and insurance coverage (Drone Academy Asia, 2023). Several locations have been designated as no-fly zones for security purposes, including Putrajaya, the Kuala Lumpur City Centre, Parliament House, and military installations (Global Drone Regulations Database, 2024).

Biosecurity encompasses multiple interconnected types addressing different categories of biological threats. Agricultural biosecurity protects crops, livestock, and aquaculture from pests and diseases through quarantine inspections, certification programs, pest surveillance systems, and rapid response protocols (Food and Agriculture Organization, 2023). Public health biosecurity addresses infectious disease threats to human populations through disease surveillance systems, laboratory networks, stockpiles of medical countermeasures, and health system surge capacity (World Health Organization, 2021). Environmental biosecurity protects ecosystems and biodiversity from invasive alien species and genetically modified organisms (Hulme, 2021). Laboratory biosecurity ensures secure storage and handling of dangerous pathogens through biosafety regulations and inventory management systems.

Figure 1.1 illustrates the major types of biosecurity and their relationships, with particular emphasis on cybersecurity's central role in enabling and protecting biosecurity functions across all domains.

Within the broader biosecurity landscape, cybersecurity has emerged as a critical enabling capability underpinning biosecurity measures' effectiveness across all domains. Modern biosecurity systems increasingly depend on digital technologies for surveillance, data management, communication, laboratory analysis, and coordination of response activities (European Union Agency for Cybersecurity, 2023). Disease surveillance systems collect and analyze vast quantities of data from healthcare facilities and laboratories to detect outbreaks early and track their spread. Border control systems use databases, risk assessment algorithms, and electronic cargo tracking to target inspections at high-risk shipments while facilitating legitimate trade. Labo-

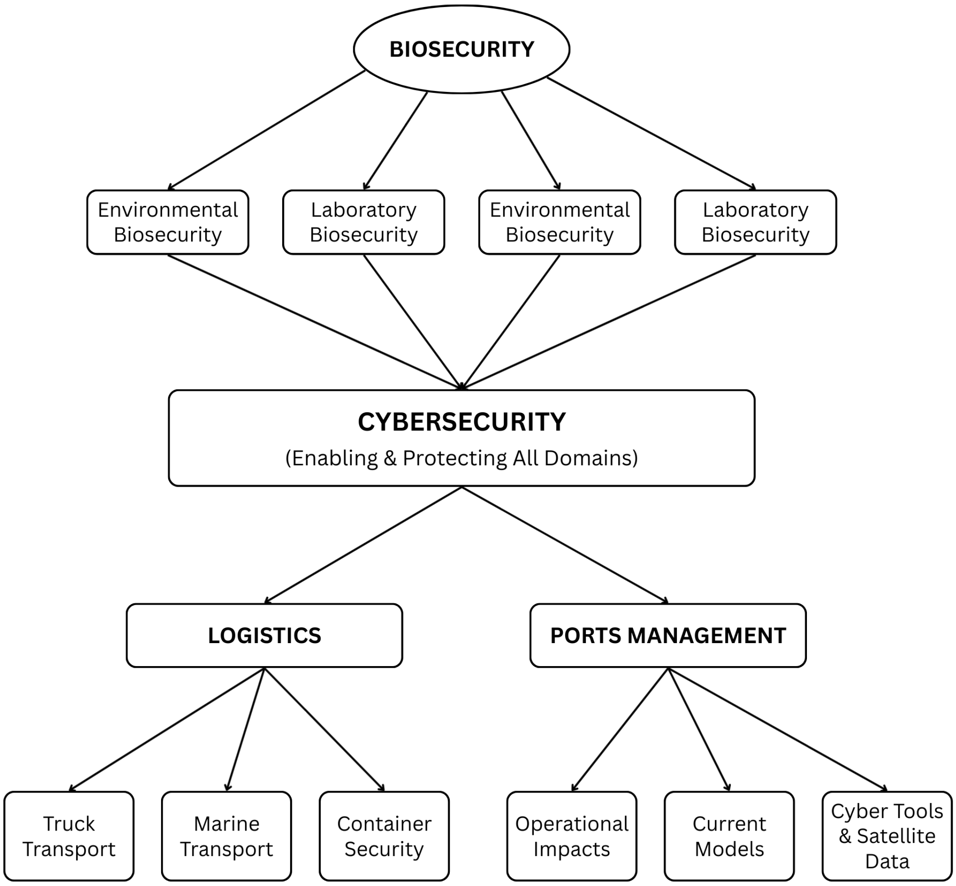

Figure 1.1: Types of Biosecurity and the Central Role of Cybersecurity

ratory information management systems maintain inventories of dangerous pathogens and track specimen transfers. Emergency management systems coordinate personnel and resource deployment during outbreaks. These digital systems create dependencies on cybersecurity—their compromise could blind surveillance systems to emerging threats, misdirect response resources, or enable theft of dangerous pathogens.

Cybersecurity intersects with biosecurity through two critical domains essential to global commerce and public health: logistics operations and ports management. Modern logistics systems rely on information technologies to coordinate goods movement across supply chains spanning multiple countries and transportation modes (Notteboom, Pallis, & Rodrigue, 2024). Truck transportation increasingly employs digital fleet management systems, electronic logging devices, and real-time cargo tracking technologies that enhance operational efficiency but create cybersecurity vulnerabilities (Department of Homeland Security, 2024). Marine transportation has adopted automatic identification systems, electronic chart display and information systems, and satellite communications that improve safety and efficiency but also create attack surfaces for cyber intrusions (International Maritime Organization, 2022). Container shipping relies heavily on digital systems including electronic cargo manifests, automated risk assessment algorithms, electronic seals, and tracking systems throughout the supply chain (World Customs Organization, 2023).

Ports management represents a particularly critical nexus of cybersecurity and biosecurity concerns given ports' role as gateways for international trade and potential entry points for biological threats. Modern ports function as complex cyber-physical systems integrating information technology, operational technology, and physical infrastructure (Notteboom et al., 2024). Terminal operating systems coordinate container movements, port community systems facilitate information sharing among multiple organizations, vessel traffic management systems monitor ship movements, and cargo screening technologies enable non-intrusive inspection for contraband including dangerous biological materials. The 2017 NotPetya ransomware attack on Maersk Line demonstrated global shipping infrastructure's vulnerability, costing approximately 300 million US dollars and forcing shutdown of 76 container terminals (Intelligence, 2018). The 2023 ransomware attack on the Port of Nagoya forced suspension of container operations for four days (Marine Digital, 2023).

Current operational models for port management increasingly recognize cybersecurity as a fundamental component of port security. Leading port authorities have adopted risk-based cybersecurity frameworks aligned with international standards including the International Maritime Organization's Guidelines on Maritime Cyber Risk Management, the National Institute of Standards and Technology Cybersecurity Framework, and ISO/IEC 27001 (International Maritime Organization, 2022; International Organization for Standardization, 2022; National Institute of Standards and Technology, 2023). Tools and technologies for port cybersecurity include network security solutions such as firewalls and intrusion detection systems, industrial control system security technologies for operational technology environments, security information and event management platforms, endpoint detection and response solutions, and vulnerability assessment tools (European Union Agency for Cybersecurity, 2023).

Terrain and satellite data play increasingly important roles in port security by providing capabilities for monitoring port facilities and tracking vessel movements.

Satellite-based automatic identification systems enable global tracking of ship movements for maritime domain awareness and security surveillance (International Maritime Organization, 2022). Synthetic aperture radar satellites provide all-weather imagery of port facilities and surrounding waters, while optical satellite imagery enables detailed monitoring of port infrastructure and vessel identification (European Space Agency, 2023). However, reliance on space-based systems creates potential vulnerabilities if satellite systems are disrupted by technical failures or deliberate attacks (Department of Homeland Security, 2024).

This thesis addresses the critical need for efficient, reliable, and secure pathfinding capabilities for autonomous vehicles operating in biosecurity-sensitive environments such as agricultural facilities, healthcare settings, transportation hubs, and border control operations. The research develops novel algorithms for grid-based pathfinding that reduce computational complexity while maintaining path quality and enabling real-time responsiveness to dynamic environment changes. Particular attention is devoted to risk-aware pathfinding that minimizes exposure to contaminated or high-risk areas, a capability essential for autonomous systems operating in biosecurity contexts where human safety and mission effectiveness depend on avoiding known or suspected biosecurity hazards.

1.2 Problem Statement

Deploying autonomous systems for grid-based navigation faces several interconnected challenges that limit their effectiveness and practical utility, particularly in time-critical and risk-sensitive applications. This section identifies four key problems that significantly constrain current autonomous navigation capabilities.

Problem 1: Real-time Computational Constraints. Standard pathfinding algorithms face a fundamental challenge when applied to large-scale or highly cluttered environments: computation time grows prohibitively as problem complexity increases. On large grids, the state space expands quadratically with environment dimensions, forcing algorithms to examine vast numbers of potential paths before finding acceptable solutions. This computational burden becomes particularly severe in risk-annotated grids commonly used for biosecurity applications, where each cell contains not only traversability information but also exposure data requiring evaluation of composite cost functions that balance distance against risk metrics. The problem intensifies in cluttered environments featuring narrow aisles or complex obstacle configurations, where algorithms must explore numerous alternative routes before identifying feasible paths. In time-critical missions such as biosecurity surveillance, rapid response to detected contamination, or emergency disinfection operations, these computational delays directly impact mission effectiveness. When autonomous systems cannot compute paths quickly enough to respond to evolving situations, the resulting hesitation or inability to react can compromise operational objectives and potentially endanger human operators or critical assets that depend on timely autonomous system deployment.

Problem 2: Limited Adaptability to Dynamic Environments. Real-world operational environments rarely remain static. Occupancy and risk information frequently change due to moving obstacles, personnel activity, updated sensor readings, new test results, and evolving contamination zones. In biosecurity contexts specifically, risk maps may require updates based on newly detected pathogens, revised quarantine boundaries, or temporal progression of contamination spread. Current planning approaches struggle with this inherent dynamism. When environmental conditions change, most systems must restart the entire planning process from the beginning, discarding all previously computed information. This complete restart incurs substantial computational overhead, especially problematic when updates occur frequently or when environments are large and complex. The delays introduced by repeated full re-planning become critical bottlenecks in time-sensitive scenarios where rapid adaptation maintains safety margins and mission effectiveness. Beyond the computational costs, frequent complete re-planning can produce inconsistent or oscillating path selections as the system responds to each new environmental state, potentially confusing human operators who must supervise autonomous operations or compromising autonomous system reliability when paths change dramatically in response to minor environmental updates. The fundamental issue is that current systems lack mechanisms to efficiently update existing plans by incorporating only locally relevant changes, instead treating every environmental modification—however small—as requiring complete plan reconstruction.

Problem 3: Significant Deployment and Integration Barriers. Despite extensive research on grid-based pathfinding algorithms and their thorough evaluation in simulation environments, a substantial gap exists between algorithmic development and practical deployment on physical autonomous vehicles. This gap manifests in multiple dimensions that collectively impede technology transfer from research to operational systems. First, there is limited availability of accessible, well-documented toolchains that integrate grid-based planning algorithms with actual vehicle control systems, particularly for scenarios requiring risk-aware navigation. Researchers and practitioners face significant barriers when attempting to bridge from abstract algorithmic implementations to systems that can command real vehicles. Second, converting abstract grid paths into executable vehicle trajectories requires careful attention to vehicle dynamics, control system constraints, and safety margins—aspects often neglected in purely algorithmic research focused on optimality metrics like path length or computation time. The practical reality of autonomous vehicle control introduces complexities that simulation environments may not capture adequately. Third, validating and testing integrated planning-execution systems typically demands access to specialized hardware, facilities, and expertise not readily available to many researchers or practitioners interested in deploying advanced pathfinding techniques. In biosecurity robotics particularly, where safe and reproducible execution is paramount for tasks such as contamination surveillance transects, targeted sampling route navigation, and systematic disinfection operations, this deployment barrier represents a critical obstacle. The absence of validated, accessible pipelines demonstrating reliable path execution even in realistic simulation environments impedes the transition from research prototypes to operational systems that biosecurity personnel could actually deploy in field conditions.

Problem 4: Unexploited Structural Regularities in Operational Environments. Many practical autonomous navigation scenarios occur in structured facilities that exhibit inherent geometric regularities, particularly horizontal symmetries in environments such as greenhouses, warehouses, hospital wards, agricultural storage facilities, and similar structured spaces. These environments typically feature parallel rows, corridors, or aisles creating symmetric patterns in their spatial organization. Classical pathfinding algorithms, however, treat each location as an independent state without recognizing or exploiting these structural regularities. This limitation results in substantial redundant computation, as algorithms independently explore symmetric regions that could potentially be processed collectively. The computational inefficiency becomes particularly pronounced in large-scale structured environments where symmetry is prevalent and predictable. Moreover, in risk-annotated grids where symmetry extends beyond geometric layout to include similar risk distributions in symmetric regions, the failure to exploit symmetry represents a missed opportunity not only for computational efficiency but also for ensuring consistent risk-cost evaluation across structurally equivalent portions of the environment. Current systems perform extensive redundant calculations exploring symmetric regions as if they were unrelated, consuming computational resources that could be allocated to more complex aspects of path planning or to enable faster response times. This fundamental inefficiency particularly constrains applications requiring real-time performance in structured environments where symmetry could provide substantial computational advantages if properly recognized and exploited.

1.3 Research Questions and Research Hypotheses

Based on the problems identified in Section 1.2, this research formulates four research questions and corresponding hypotheses that guide the development, implementation, and evaluation of the proposed pathfinding techniques.

Research Question 1 (RQ1): To what extent can constraining search to an Incremental Line Search (ILS) corridor reduce computational requirements and cumulative exposure while preserving path optimality on risk-annotated grid maps?

Research Hypothesis 1 (RH1): The ILS approach will achieve significant reductions in runtime and node expansions (expected reduction of 40-70% compared to standard A*) while maintaining path lengths within 5% of optimal solutions. Additionally, on risk-annotated grids, ILS will reduce the cumulative exposure integral (sum of risk values along the path) by prioritizing paths through the corridor that naturally avoid high-risk regions, with negligible impact on path quality as measured by standard optimality metrics.

Research Question 2 (RQ2): Can an adaptive corridor mechanism that dynamically adjusts ILS corridor width based on local obstructions and risk concentrations support efficient re-planning under dynamic environmental updates while maintaining exposure constraints?

Research Hypothesis 2 (RH2): An adaptive corridor widening strategy that expands the search space only near detected obstructions or risk spikes will sustain the computational advantages of ILS (maintaining 50-80% of the speedup observed in static scenarios) while successfully handling moderate environment dynamics. The adaptive mechanism will keep cumulative exposure within user-specified thresholds (e.g., no more than 15% increase in exposure integral) even under dynamic updates, and will achieve re-planning latencies that are 3-5 times faster than complete re-planning approaches.

Research Question 3 (RQ3): Does an open-source planning-to-flight pipeline integrating ILS with risk-aware cost functions and ArduPilot Software-In-The-Loop (SITL) simulation reliably execute grid-derived paths for autonomous vehicles?

Research Hypothesis 3 (RH3): The integrated planning-to-flight pipeline will successfully execute planned paths in ArduPilot SITL with bounded tracking errors (position error < 2 meters for waypoint following) and successful task completion rates exceeding 95% for typical biosecurity mission profiles including surveillance transects, sampling routes, and disinfection patterns. The pipeline will demonstrate reproducible execution across multiple simulation runs with consistent performance metrics, providing a validated foundation for safe hardware deployment when permitted by safety protocols and regulatory requirements.

Research Question 4 (RQ4): When horizontal symmetry exists in grid environments, does the Folding A* algorithm provide predictable constant-factor acceleration while preserving optimality and correctly handling risk-weighted costs?

Research Hypothesis 4 (RH4): On horizontally symmetric grids, Folding A* will achieve approximately 50% reduction in the effective state space (i.e., exploring roughly half the nodes compared to standard A*) with exact optimality preservation for both standard and risk-weighted cost functions. The runtime improvements will scale consistently across different grid sizes and obstacle densities, yielding speedups ranging from 1.8x to 2.2x compared to standard A* on symmetric maps. The algorithm will correctly handle risk-annotated grids by maintaining symmetry-aware risk propagation, ensuring that paths in symmetric regions are evaluated with equivalent composite costs.

1.4 Research Objectives

The overarching aim of this research is to develop, implement, and validate efficient grid-based pathfinding techniques for autonomous systems that combine computational efficiency with practical deployability, particularly in risk-sensitive biosecurity applications. This aim is achieved through four specific research objectives:

Objective 1 (O1): Design and evaluate an Incremental Line Search (ILS) framework for risk-aware pathfinding that focuses exploration within a narrow corridor centered on a direct line between start and goal positions. The framework must integrate seamlessly with standard grid planning algorithms (A*, Dijkstra, BFS), support optional risk-weighted cost functions, and demonstrate measurable reductions in runtime and node expansions while maintaining near-optimal path quality. The evaluation will encompass diverse grid scenarios including varying sizes (50x50 to 1000x1000), obstacle densities (10%-40%), and risk annotation patterns reflecting realistic biosecurity scenarios such as quarantine zones and contamination gradients.

Objective 2 (O2): Develop and validate an adaptive corridor control mechanism for ILS that dynamically adjusts corridor width based on local environmental features including obstacle concentrations and risk-level spikes. The adaptive mechanism must balance exploration efficiency with solution completeness, ensuring that narrow corridors are used in open regions while sufficient expansion occurs near obstructions or high-risk zones. The control strategy will be evaluated under dynamic environments with piecewise-static updates at various frequencies (1-10 updates per planning session), assessing its ability to maintain computational efficiency while preserving path quality and exposure constraints across update cycles.

Objective 3 (O3): Implement and formally validate the Folding A* algorithm for horizontally symmetric grid maps, providing rigorous proofs of correctness and optimality preservation for both standard and risk-weighted cost functions. The implementation must include automatic symmetry detection mechanisms, efficient folding transformations that halve the effective state space, and correct unfolding procedures to recover full-space paths. The empirical evaluation will quantify constant-factor speedup gains across diverse symmetric grid scenarios, analyzing how performance scales with grid size, obstacle density, and the degree of symmetry in the environment.

Objective 4 (O4): Build and release an open-source, end-to-end grid-planning-to-flight pipeline that integrates ILS and Folding A* algorithms with ArduPilot Software-In-The-Loop (SITL) simulation for autonomous vehicle execution. The pipeline must support risk-annotated grid inputs, automatic conversion of grid paths to vehicle waypoints with appropriate safety margins, and comprehensive logging for performance analysis. The implementation will be validated through systematic SITL experiments demonstrating successful execution of representative biosecurity mission profiles including surveillance transects, area coverage patterns, and targeted sampling or disinfection routes. The pipeline will be released under a permissive open-source license to facilitate community adoption and extension.

1.5 Scope and Limitations

This research focuses on grid-based pathfinding for autonomous navigation with specific scope boundaries and acknowledged limitations defining the results' applicability and generalizability.

Environment Representation: The research considers two-dimensional occupancy grids with optional risk layers. Grid cells are classified as either traversable or occupied, with traversable cells optionally annotated with continuous risk values normalized to [0, 1]. The evaluation encompasses grid sizes from 50x50 to 1000x1000 cells and obstacle densities from 10% to 40%. Risk distributions include uniform random patterns, localized hotspots, and structured zones reflecting quarantine areas or contamination gradients. Full three-dimensional environments with varying altitude constraints are considered out of scope.

Agent Model: The research assumes a single autonomous agent with either 4-connected or 8-connected motion primitives. For aerial vehicles, a fixed-altitude abstraction is employed. The agent is modeled as occupying a single grid cell at any discrete time step. Vehicle dynamics and continuous-time trajectory optimization are addressed at the execution layer rather than in the planning phase. Multi-agent coordination is explicitly excluded.

Environment Dynamics: The research considers moderate, piecewise-static dynamics where occupancy and risk updates occur at discrete time points with sufficient intervals for the planner to complete computation. The evaluation includes scenarios with 1 to 10 updates per planning session. Highly adversarial dynamics and continuous-time stochastic environment evolution are out of scope.

Planning Objectives: The primary objective is to find collision-free paths from start to goal subject to real-time latency constraints. For standard grids, path cost is cumulative distance based on Euclidean or Manhattan metrics. For risk-annotated grids, a composite cost function combines distance with weighted risk: $cost = distance + \lambda \cdot \sum risk$, where λ is a user-specified weight parameter. The research evaluates multiple λ values to assess sensitivity to risk-cost tradeoff preferences.

Evaluation Metrics: Performance assessment employs runtime (wall-clock computation time), nodes expanded, path length, exposure integral (sum of risk values along the path), maximum on-path risk, and re-plan latency. All experiments are conducted on controlled hardware configurations (Intel Core i7 or equivalent, 16GB RAM) to ensure reproducibility.

Input Assumptions: The research assumes that occupancy information and optional risk annotations are provided as exogenous inputs from external perception systems. The source of this information, sensor fusion techniques, and uncertainty quantification are not addressed. Start and goal positions are assumed known and specified in advance.

Limitations: The computational gains of corridor-constrained ILS may diminish on highly tortuous maps. Folding A* provides benefits only when horizontal symmetry is present and detectable. Risk-map uncertainty modeling is not explicitly addressed. The ArduPilot SITL integration provides validation in simulation but does not replace the need for comprehensive hardware testing before operational deployment.

1.6 Significance of the Research

This research makes significant contributions to both theoretical understanding and practical deployment of autonomous navigation systems, with particular relevance to biosecurity applications.

Computational Efficiency: The proposed corridor-constrained ILS approach dramatically reduces the search space requiring exploration to find high-quality paths. By constraining exploration to a narrow corridor around the direct line between start and goal, ILS enables real-time pathfinding on large-scale grids using commodity hardware, without requiring specialized processors or GPU acceleration. For time-critical applications such as emergency response or surveillance of sudden contamination events, computing paths within strict latency constraints can significantly enhance operational capability.

Enhanced Safety: Integrating risk-annotated grids with ILS and Folding A* enables autonomous systems to balance competing objectives of efficiency and safety in contaminated or hazardous environments. Explicitly treating exposure integrals and maximum on-path risk provides quantifiable safety metrics that can be incorporated into mission planning. By reducing average exposure while maintaining acceptable path lengths, the proposed techniques enable more aggressive deployment of autonomous systems in scenarios where human access would pose unacceptable safety risks.

Improved Responsiveness: The adaptive corridor mechanism specifically addresses efficient re-planning challenges under dynamic environment changes. By enabling localized corridor widening only where necessary, the adaptive approach maintains ILS computational efficiency while ensuring solution completeness under moderate dynamics. This capability is particularly valuable in biosecurity operations where risk maps are frequently updated based on new test results or revised contamination boundaries.

Predictable Performance: Folding A* makes a fundamental theoretical contribution by demonstrating how horizontal symmetry can be exploited to achieve guaranteed constant-factor state-space reductions while preserving optimality. Unlike heuristic acceleration techniques that may provide variable speedups, the symmetry-based reduction is predictable and analyzable, making it suitable for systems with hard real-time constraints or safety-critical requirements.

Practical Deployment Infrastructure: The open-source planning-to-flight pipeline addresses a critical gap between algorithmic research and practical deployment. Integration with ArduPilot SITL enables researchers and practitioners to evaluate and refine pathfinding algorithms in a realistic simulation environment. This infrastructure reduces barriers to deploying advanced pathfinding techniques on physical vehicles, potentially accelerating technology transfer from research prototypes to operational systems.

Broader Impact: While biosecurity applications provide the primary motivation, the techniques developed in this thesis have broader applicability to general autonomous navigation problems wherever grid-based planning is employed. The ILS corridor-constrained approach can benefit warehouse automation, agricultural robots navigating crop rows, search and rescue operations, and any scenario where rapid pathfinding on large grids is required. Open-source release of implementations and the planning-to-flight pipeline facilitates adoption across diverse application domains.

1.7 Organization of the Thesis

This thesis is organized into seven chapters that systematically develop, implement, and evaluate the proposed pathfinding techniques.

Chapter 1: Introduction provides foundational context including biosecurity challenges and their relationship to autonomous navigation, identification of key problems limiting current approaches, formulation of research questions and hypotheses, specification of research objectives, and definition of scope and limitations.

Chapter 2: Literature Review surveys existing knowledge on classical search algorithms (Dijkstra, BFS, A*), heuristic search techniques, incremental and anytime planning approaches, line-of-sight path shortcutting methods, symmetry exploitation techniques, and risk-sensitive routing methods. The review identifies gaps in current approaches motivating ILS and Folding A* development.

Chapter 3: Incremental Line Search Framework presents detailed design, implementation, and analysis of the ILS corridor-constrained pathfinding approach, including corridor construction algorithms, adaptive widening strategies, integration mechanisms with standard grid planners, theoretical complexity analysis, and benchmark results on risk-aware grid scenarios.

Chapter 4: Folding A* for Symmetric Grids develops the symmetry-aware planning technique, including formal definitions of horizontal symmetry, folding transformation descriptions, comprehensive correctness and optimality proofs, theoretical complexity analysis, and experimental results quantifying speedups across diverse symmetric grids.

Chapter 5: Open-Source Planning-to-Flight Pipeline documents the integrated system bridging grid-based planning with vehicle execution using ArduPilot SITL, including pipeline architecture, path-to-waypoint conversion, ArduPilot interface configuration, and experimental validation demonstrating successful execution of biosecurity mission profiles.

Chapter 6: Results and Discussion provides comprehensive comparative evaluations including comparative performance analysis of ILS versus standard algorithms, evaluation of Folding A* speedups, ablation studies isolating parameter impacts, sensitivity analyses, and practical implications for biosecurity operations.

Chapter 7: Conclusions synthesizes key findings including algorithmic innovations, theoretical results, and empirical findings, discusses limitations including assumptions and cases where proposed techniques may not provide advantages, and identifies promising avenues for future research including extensions to three-dimensional environments, integration with learning-based approaches, and field deployment on physical hardware.

CHAPTER 2

CITATIONS AND BIBLIOGRAPHY

This chapter should have been a survey on the history of TeX and LaTeX, and a comparison to conventional word processors in preparing academic documents. Due to lack of time on the author's part, and also the abundance of such discussions on the web, we look at ways to prepare the bibliography and citations instead.

2.1 The `*.bib` File

First of all, bear in mind that your bibliography file (`*.bib` files) is like a database. That means you can maintain a centralised list, and reuse it for all your publications. LaTeX will only list sources that you actually cite in the text for each document, according to the bibliography and citation style you select in each document. But you can still hack it so that your own publications are listed, even if you did not cite it.

```
@BOOK{latex:companion,
  title = {The {\LaTeX} Companion},
  publisher = {Addison-Wesley},
  year = {2004},
  author = {Frank Mittelbach and Michel Goosens and
     Johannes Braams and David Carlisle and Chris
     Rowley},
  series = {Addison-Wesley Series on Tools and
     Techniques for Computer Typesetting},
  address = {Boston, MA, USA},
  edition = {2nd}
}
```

Figure 2.1: A BibTeX Entry

As an example, in `mybib.bib` I created a BibTEX entry with JabRef, the source text of which is shown in Figure 2.1.

One thing to note about authors' names: BibTEX recognises "Mittelbach" as the last name for both `Frank Mittelbach` and `Mittelbach, Frank`. So for a name like "Lim Lian Tze", you would have to specify it as either `Lian Tze Lim` or `Lim, Lian Tze` for BibTEX to recognise "Lim" as the last name correctly. In addition, if the surname or family name of an author consists of multiple words, enclose it with braces to avoid surprises, like so: `Syed Muhammad Naquib {al-Attas}`.

2.2 Citations using the `natbib` package

The `usmthesis` package imports the `natbib` and `apacite` package which provides flexible citation mechanisms, so see its documentation for more details. On a MiKTEX installation, use the command prompt to issue `mthelp --view natbib` and `mthelp --view apacite` to access the documentation. On TeXLive, simply type `texdoc natbib` and `texdoc apacite` and the documentation will be displayed automatically, if it's found on your machine.

The basic citation commands are `\citet` and `\citep`, which stands for *textual* and *parenthetical* citation respectively. They take extra arguments, too, for adding notes in the citations.

2.2.1 Author-Year System

The default bibliography style is APA:

- `\citet{latex:companion}` → **?**
- `\citet[chap.~2]{latex:companion}` → **?**, chap. 2
- `\citep{latex:companion}` → (**?**)
- `\citep[chap.~2]{latex:companion}` → (**?**, chap. 2)
- `\citep[see also][]{latex:companion}` → (see also **?**)
- `\citep[see also][chap.~2]{latex:companion}` → (see also **?**, chap. 2)
- `\citet{latex:companion,roberts}` → **??**
- `\citep{latex:companion,roberts}` → (**??**)

You may also want to write only the author's name or year occassionally:

- `\citeauthor{latex:companion}` → **?**
- `\citeyear{latex:companion}` → **?**
- `\citeyearpar{latex:companion}` → (**?**)

2.2.2 Numeric System

If you prefer the plain, numerical system, do the following steps first:

1. In `usmthesis.cls`, search for the line `\RequirePackage[natbibapa]{apacite}` and modify it to:

   `\RequirePackage[numbers]{natbib}`
2. In `usmthesis.tex`:
   - modify the `biblography` styles to:

     `\bibliographystyle{plainnat}`

     `\bibliographystyleown{plainnat}`

or any other number system style that you prefer.

You will then get the following citation outputs:

- `\citet{latex:companion}` → Mittelbach et al. [1]
- `\citet[chap.~2]{latex:companion}` → Mittelbach et al. [1, chap. 2]
- `\citep{latex:companion}` → [1]
- `\citep[chap.~2]{latex:companion}` → [1, chap. 2]
- `\citep[see also][]{latex:companion}` → [see also 1]
- `\citep[see also][chap.~2]{latex:companion}` → [see also 1, chap. 2]
- `\citet{latex:companion,roberts}` → Mittelbach et al. [1], Roberts [3]
- `\citep{latex:companion,roberts}` → [1, 3]
- `\citeauthor{latex:companion}` → Mittelbach et al.
- `\citeyear{latex:companion}` → 2004
- `\citeyearpar{latex:companion}` → [2004]

CHAPTER 3

FIGURES, TABLES, EQUATIONS, ALGORITHMS, ETC

(This is supposed to be the design or methodology chapter. Instead, we include examples on inserting figures, tables, mathematical equations. . . i.e. things that you might want to include in your thesis.)

3.1 Inserting Figures

You can draw diagrams with special LaTeX commands, but this may take some extra time to learn. I've had some forays into the `pgf` and `tikz` packages and must say I quite like the results; but as I said, they take time to learn. If you want a faster solution, you can draw your diagrams using other applications, and saving them as graphic files (EPS, PNG, JPG, PDF).

LaTeX requires EPS (encapsulated postscript) graphic files when generating DVI output, and PNG, JPG or PDF when generating PDF output.

For exporting to EPS, try `http://www.cloudconvert.com`. It's like a Swiss knife for converting from almost any format, to almost any format.

Do note that IPS **discourages** the use of colours in your thesis, including diagrams and figures. Phographs and colour plates are exceptions to this rule: see Section 3.3.

Here's how to insert a picture with the filename `pythag.eps` or `pythag.png`. I'm going to display it here with 5cm width, and the caption "Pythagoras' Theorem".

```
\begin{figure}[hbt!]\centering
\includegraphics[width=50mm]{pythag}
\caption{Pythagoras' Theorem}\label{fig:pythagoras}
\end{figure}
```

Figure 3.1: Including a Graphics File

The result would be:

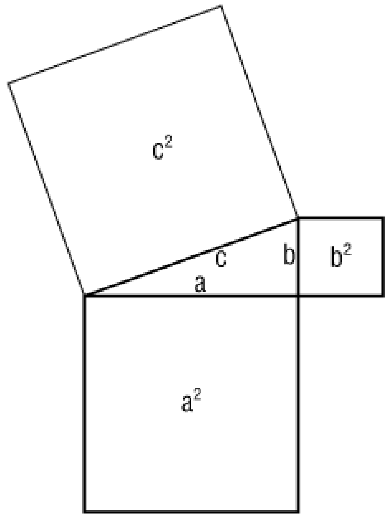

Figure 3.2: Pythagoras' Theroem

Don't specify the extension of the graphic file. The template will automatically look for the EPS or the PNG (or otherwise) versions, depending on whether `latex` or `pdflatex` was used. The `figure` environment will also ensure that that an entry is inserted into the *List of Figures* automatically – including the figure numbering, caption and page number.

In addition, the width of the included graphics can also be specified as a percentage of the text width, e.g. `width=.2\textwidth` would cause the graphics to occupy 20% of the text width.

Notice that I inserted a `\label` just after the `\caption`. This can be used for referencing the figure number, like this:

`Figure \ref{fig:pythagoras}` → Figure 3.2

This works the same for chapters, sections, tables, equations too. In `chap-intro.tex`, I labelled the Introduction chapter with `\label{chap:intro}`. I also labelled the section on inserting figures, `\label{sec:figure}`. So now I can do

`Chapter \ref{chap:intro}` → Chapter 1

`section \ref{sec:figure}` → section 3.1

Everytime the numbering of the heading changes, the reference will change automatically as well. **This is another advantage of using LaTeX**: you do not need to manually update the reference counters (nor the Table of Contents, List of Figures and Tables) whenever you add or remove figures, tables, sections or chapters.

You might also want to try out `JpgfDraw`: it is a vector graphics and drawing application (requiring Java), and can export to LaTeX code which you can paste into your LaTeX source. `JpgfDraw` is available from `http://theoval.cmp.uea.ac.uk/~nlct/jpgfdraw/index.html`.

3.2 How Do I Do Subfigures?

Here's an example on how to do subfigures (and similarly subtables):

3.3 Inserting Plates

Colour photographs are now regarded as *plates*. They must be listed in the *List of Plates* instead of the List of Figures, and should be printed in colour on glossy photo

```
\begin{figure}[hbt!]
  \begin{minipage}{.49\textwidth}
  \centering
  \subfloat[First caption]{\includegraphics[width=3cm]{
     pythag}} \label{fig:sub1}
  \end{minipage}
  \hfill
  \begin{minipage}{.49\textwidth}
  \subfloat[Second caption]{\includegraphics[width=0.8\
     textwidth]{USMScience}}\label{fig:sub2}
  \end{minipage}

  \caption{This is the main caption of the figure.}
  \label{fig:main}
\end{figure}
```

Figure 3.3: Creating subfigures within figures

(a) First caption

(b) Second caption

Figure 3.4: This is the main caption of the figure.

paper (**?**).

The `usmthesis` document class defines a new `plate` environment, as well as a corresponding `\listofplates` command. (The `\listofplates` command is already placed in the sample template file `usmthesis.tex`.) In short, all you need to do to insert a photograph or plate (as a graphics file `USMScience.{eps,png,jpg}`) is shown in Figure 3.5, and you will then get Plate 3.1 as the result.

```
\begin{plate}[hbt!]\centering
\includegraphics[width=.9\textwidth]{USMScience}
\caption{School of Computer Sciences, USM}\label{plate:
   ppsk:usm}
\end{plate}
```

Figure 3.5: Inserting a Plate

Plate 3.1: School of Computer Sciences, USM

3.4 Inserting Tables

Typesetting tables can be a little troublesome especially with complex layouts. Look up (**?**) to learn about some tips, or you can use the LaTable program (`http://www.g32.org/latable/`) to help you.

If using LaTable, when you're done designing the table, copy the whole table as LaTeX code, and paste it in your source file. (You may add additional formatting commands, like bold, italics, etc.) If this is going to be a numbered table, remember to surround it with `\begin{table}` and `\end{table}`, and give it a caption, like this:

```
\begin{table}[hbt!]\centering
\begin{tabular}{| l | c || r |}
\hline
\textbf{Name} & \textbf{Category} & \textbf{Quantity}
  \\
\hline\hline
Apple & Fruit & 10 \\
\hline
Cucumber & Vegetable & 25 \\
\hline
Daisy & Flower & 5 \\
\hline
\end{tabular}
\caption{Sample Table Only} \label{table:sample}
\end{table}
```

Figure 3.6: Typesetting Tables

Table 3.1: Sample Table Only

Name	**Category**	**Quantity**
Apple	Fruit	10
Cucumber	Vegetable	25
Daisy	Flower	5

Note also that `usmthesis` is configured such that captions for figures are placed *below* the figures, and captions for tables are placed *above* them, in accordance with the formatting guidelines.

Many of us would have had massive headaches about lining up decimal places in table columns (as mentioned in the IPS guidelines) if not for this tip from (**?**, pp. 274–276). This method uses the `dcolumn` package (already loaded by `usmthesis.cls`). Instead of using `l`,`c` or `r` as the column type in the `tabular` declaration, use

`D{`*`input sep`*`}{`*`output sep`*`}{`*`decimal places`*`}`.

The LaTeX code in Figure 3.7 will give you Table 3.2.

```
\begin{table}[htb!]\centering
\begin{tabular}{| c | D{.}{.}{2} |}
\hline
Item & \multicolumn{1}{c|}{Reading}\\\hline
A & 1.11\\\hline
B & 3.99\\\hline
C & 2.27\\\hline
\end{tabular}
\caption{A table with decimal data}
\end{table}
```

Figure 3.7: Aligning decimal data in tables

Table 3.2: A table with decimal data

Item	Reading
A	1.11
B	3.999
C	22.2

Without using dcolumn, you'd get something like this:

Table 3.3: A table with decimal data (mis-aligned)

Item	Reading
A	1.11
B	3.999
C	22.2

3.5 Full-paged, Sideways Figures and Tables

To make a figure appear on a landscape, full-page layout, put your \includegraphics command in a sidewaysfigure environment (Figure 3.8).

```
\begin{sidewaysfigure}\centering
\includegraphics[width=\textheight]{latex-win-comp}
\caption{A full-page, sideways figure}\label{fig:
   sidewaysfig}
\end{sidewaysfigure}
```

Figure 3.8: Including a sideway, full-page graphic

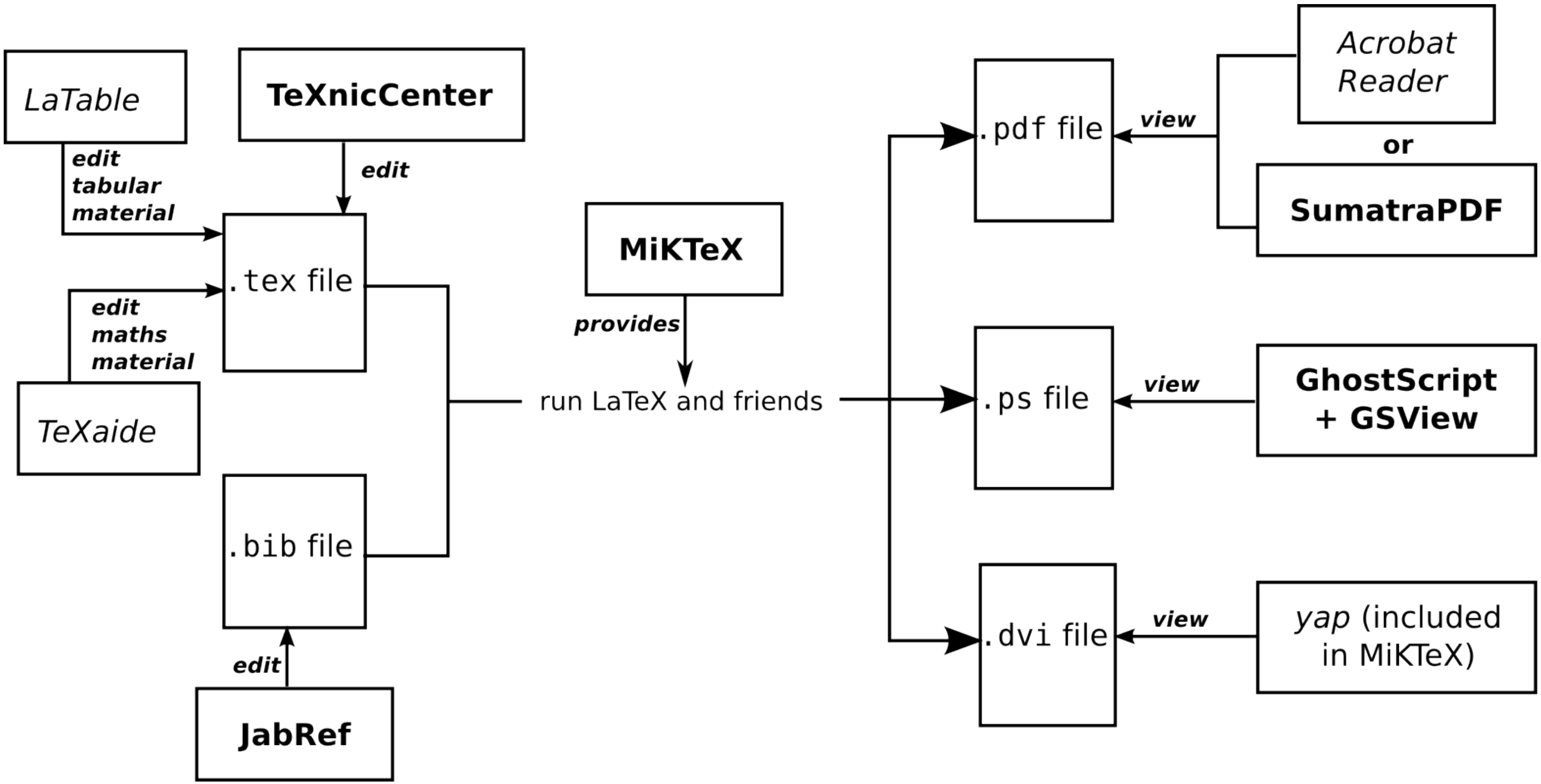

Figure 3.9: A full-page, sideways figure

The resultant figure (Figure 3.9) should appear on the next page.

For a sideways table, use the `sidewaystable` environment instead around your usual `tabular` material.

3.6 Mathematical Equations

Typesetting mathematical material is one of, if not *the*, strongest capabilities of LaTeX. After all, that was the Knuth's main motivation for creating TeX. As it is impossible to enumerate all possible mathematically-related commands and macros here, we will just give some examples. The reader is directed to the many well-written online tutorials, such as (**?**), for more elaborate examples. TeXnicCenter also provides many shortcut buttons for inserting mathematical symbols.

$$z^2 = x^2 + y^2 \tag{3.1}$$

$$\phi = \frac{1}{2}(1+\sqrt{5}) \tag{3.2}$$

$$\phi = 1 + \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{F_n F_{n+1}} \tag{3.3}$$

Equation 3.1 is the Pythagoras Theorem. (3.2) gives the golden ratio ϕ, and (3.3) relates it to the Fibonacci series.

The LaTeX code to generate the above mathematics materials are shown in Figure 3.10. As you can see, references to equations can be achieved with either `\ref` or `\eqref`.

```
\begin{equation}\label{eq:pythagoras}
z^2 = x^2 + y^2
\end{equation}

\begin{equation}\label{eq:golden:ratio}
\phi = \frac{1}{2} (1 + \sqrt{5})
\end{equation}

\begin{equation}\label{eq:golden:ratio}
\phi = \frac{1}{2} (1 + \sqrt{5})
\end{equation}
\begin{equation}\label{eq:golden:ratio:fibonacci}
\phi = 1 + \sum ^ {\infty} _ {n=1}
                \frac{ (-1) ^ {n+1} }{ F_n F_{n+1} }
\end{equation}

Equation~\ref{eq:pythagoras} is the Pythagoras Theorem.
\eqref{eq:golden:ratio} gives the golden ratio $\phi$,
   and
\eqref{eq:golden:ratio:fibonacci} relates it to the
   Fibonacci
series.
```

Figure 3.10: Typesetting Mathematical Equations

A disclaimer: if you think the mathematic equations don't look as great as all those LaTeX advocates make them out to be, that's because IPS requires Times to be used and the current offerings of free LaTeX math fonts for Times don't look great. It would've been a different picture if we used Computer Modern.

3.7 Acronyms

If you have a list of acronyms or symbols, edit the file `loa.tex` as in Figure 3.11.

You can also use this acronym list to help expand it the first time you mention it in your text. For example, the first time you use `\ac{USM}`, 'Universiti Sains Malaysia (USM)' will be the output (without the quotes). After that, all calls to `\ac{USM}` will

```
\begin{acronym}[UTMK] %% replace 'UTMK' with the
   longest acronym in your list
\acro{IPS}{Institut Pengajian Siswazah}
\acro{PPSK}{Pusat Pengajian Sains Komputer}
\acro{USM}{Universiti Sains Malaysia}
\acro{UTMK}{Unit Terjemahan Melalui Komputer}
\end{acronym}
```

Figure 3.11: The template `loa.tex` for acronyms

give 'USM' (without the quotes). For more information, see the documentation for the `acronym` package.

3.8 Typesetting Algorithms

As computer scientists, it is quite common to include algorithms and/or pseudocode. There are a number of different packages available, but unfortunately they tend not to work well together! I'm using `algorithmicx` here.

Algorithm 1 Computing $x^n, n > 0$

Require: $n \geq 0$
Ensure: $y = x^n$
  $y \Leftarrow 1$
  $X \Leftarrow x$
  $N \Leftarrow n$
  while $N \neq 0$ **do**
    if N is even **then**
      $X \Leftarrow X \times X$
      $N \Leftarrow \frac{N}{2}$ ▷ This is a comment
    else if N is odd **then**
      $y \Leftarrow y \times X$
      $N \Leftarrow N - 1$
    end if
  end while

```
\begin{algorithm}[hbt!]
\begin{algorithmic}
\Require $n \geq 0$
\Ensure $y = x^n$
\State $y \Leftarrow 1$
\State $X \Leftarrow x$
\State $N \Leftarrow n$
\While{$N \neq 0$}
\If{$N$ is even}
  \State $X \Leftarrow X \times X$
  \State $N \Leftarrow \frac{N}{2} $  \Comment{This is
     a comment}
\ElsIf{$N$ is odd}
  \State $y \Leftarrow y \times X$
  \State $N \Leftarrow N - 1$
\EndIf
\EndWhile
\end{algorithmic}
\caption{Computing $x^n, n > 0$}
\end{algorithm}
```

Figure 3.12: Typesetting Algorithms

3.9 Program Listings

You may have noticed that I used the `lstlisting` environment to typeset some of the LaTeX examples – with pretty-printing[1], too, including automatic line-breaking. For more information, see the documentation for the `listings` package: it's available online at `http://www.texdoc.net/pkg/listings`.

Just to give some simple example here. For example, to typeset a "Hello World" Java program with syntax highlighting, you can use the following code:

If you want to turn off the syntax highlighting, set `language={}`. (See the `listings` documentation for a list of programming languages for which syntax highlighting is supported.) You can also change the `basicstyle` value to get different effects: e.g. a

[1]Whether you agree that it *is* pretty is another story altogether.

```
\lstset{basicstyle=\small\ttfamily, language=Java,
   breaklines=true, columns=fullflexible, tabsize=2}
\begin{lstlisting}
public class HelloWorld {
        public static void main( String arg[] ) {
        for (int i = 0; i < 10; i++) {
                        System.out.println( "Hello
                           World!" + i);
                }
        }
}
\end{lstlisting}
```

Figure 3.13: Typesetting a Java program listing

```
public class HelloWorld {
  public static void main( String arg[] ) {
      for (int i = 0; i < 10; i++) {
    System.out.println( "Hello␣World!" + i);
   }
  }
}
```

Figure 3.14: A pretty-printed Java program listing with syntax highlighting

different font family, size or text formatting.

Here's another example for a C program:

```
\lstset{basicstyle=\sffamily, language=C, breaklines=
   true, columns=fullflexible, tabsize=2}
\begin{lstlisting}
int main() {
        int c = 0;
        c = c + 1;
        printf( "%d", c );
        return 0;
}
\end{lstlisting}
```

Figure 3.15: Typesetting a C program listing

```
int main() {
    int c = 0;
    c = c + 1;
    printf( "%d", c );
    return 0;
}
```

Figure 3.16: A pretty-printed C program listing with syntax highlighting

And here is the same C program listing *without* syntax highlighting (by setting `language={}`):

```
int main() {
    int c = 0;
    c = c + 1;
    printf( "%d", c );
    return 0;
}
```

Figure 3.17: A C program listing without syntax highlighting

CHAPTER 4

IMPLEMENTATION

Now is the time to “implement” your thesis with LaTeX. Go forth and typeset! Happy LaTeXing! ☺

4.1 Printing Your Thesis

This is *very* important. Assuming you’re printing your thesis from Acrobat Reader, make sure the following settings are chosen correctly in the Print window:

- A4 paper size is selected.
- Make sure your Printer settings is using A4 too.
- No page scaling.

Otherwise, the margins of your printed outputs may go horribly wrong. Print one or two pages first to make sure everything looks fine before printing your entire thesis.

CHAPTER 5

DISCUSSION

Just a placeholder for the discussion chapter.

CHAPTER 6

CONCLUSION

T-that's all folks. Have fun with LaTeX!

REFERENCES

Association of Southeast Asian Nations. (2024). *ASEAN health security framework 2024.* Jakarta, Indonesia.

Civil Aviation Authority of Malaysia. (2024). Malaysian civil aviation regulations 2016: Part 101 - unmanned aircraft systems [Computer software manual]. Subang, Selangor, Malaysia.

CMS Law Offices. (2024). *Malaysia's new cyber security law takes effect.* Retrieved from `https://cms-lawnow.com/en/ealerts/2024/08/malaysia-s-new-cyber-security-law-takes-effect` (Accessed: 2024-11-21)

Department of Homeland Security. (2024). *U.s. maritime trade and port cybersecurity: Analytic exchange program phase II* (Tech. Rep.). Washington, DC, USA: Department of Homeland Security.

Drone Academy Asia. (2023). *Drone laws and regulations in malaysia.* Retrieved from `https://www.droneacademy-asia.com/post/drone-laws-and-regulations-in-malaysia` (Accessed: 2024-11-21)

European Space Agency. (2023). *Satellite technology for maritime security and surveillance* (Tech. Rep.). Paris, France: European Space Agency.

European Union Agency for Cybersecurity. (2023). *Port cybersecurity: Good practices for cybersecurity in the maritime sector* (Tech. Rep.). Athens, Greece: ENISA.

Food and Agriculture Organization. (2023). *Biosecurity toolkit: A strategic and integrated approach.* Rome, Italy.

Global Drone Regulations Database. (2024). *Malaysia drone laws 2024.* Retrieved from `https://drone-laws.com/drone-laws-in-malaysia/` (Accessed: 2024-11-21)

Government of Malaysia. (2024). *Cyber security act 2024 (act 854).* Federal Government Gazette. (Gazetted on 26 June 2024, Effective 26 August 2024)

Hulme, P. E. (2021). One biosecurity: A unified concept to integrate human, animal, plant, and environmental health. *Emerging Topics in Life Sciences*, *5*(4), 539–549. doi: 10.1042/ETLS20210074

Intelligence, L. L. (2018, July). The cost of the NotPetya cyber attack on Maersk.

Lloyd's List, *2018*.

International Maritime Organization. (2022). Guidelines on maritime cyber risk management [Computer software manual]. London, United Kingdom. (MSC-FAL.1/Circ.3/Rev.2)

International Organization for Standardization. (2022). *ISO/IEC 27001:2022 information security management systems - requirements.* Geneva, Switzerland.

Marine Digital. (2023). *Cybersecurity in shipping and port technologies: Examples of cyber attacks in maritime.* Retrieved from `https://marine-digital.com/cybersecurity_in_shipping_and_ports` (Accessed: 2024-11-21)

Ministry of Health Malaysia. (2023). *National biosecurity and biosafety strategic plan 2023-2027.* Putrajaya, Malaysia.

National Cyber Security Agency Malaysia. (2024). *Implementation guidelines for the cyber security act 2024* (Tech. Rep.). Putrajaya, Malaysia: National Cyber Security Agency.

National Institute of Standards and Technology. (2023). NIST cybersecurity framework version 2.0 [Computer software manual]. Gaithersburg, Maryland, USA.

Notteboom, T., Pallis, A., & Rodrigue, J. P. (2024). Port economics, management and policy: Chapter 11.4 - port safety, security, and cybersecurity. *Routledge Studies in Transport Analysis*.

United Nations Office for Disarmament Affairs. (2023). *The biological weapons convention* (Tech. Rep.). New York, USA: United Nations.

World Customs Organization. (2023). *Guidelines for cargo screening and inspection in the context of biosecurity.* Brussels, Belgium.

World Health Organization. (2021). International health regulations (2005), third edition. *WHO*.

World Trade Organization. (2024). *Agreement on the application of sanitary and phytosanitary measures* (Tech. Rep.). Geneva, Switzerland: World Trade Organization.

APPENDICES

APPENDIX A

DATA USED

Put some test data here.

APPENDIX B

UML DIAGRAMS

Yet another dummy placeholder for appendix material.