

TRAINING SIMULATOR: BATCH POLYSTYRENE REACTOR

MOHD SYAKIR BIN SHAHIDAN

Thesis Submitted In Fulfillment Of the Requirements
For The Award of Degree in Chemical Engineering Course.

School of Chemical Engineering
UNIVERSITI SAINS MALAYSIA

2022

ACKNOWLEDGEMENT

First and foremost, I would like to express my gratitude to my family and friends for support me physically and mentally in completing this Report 1 for the thesis. Then, I also would like to express my deepest gratitude towards my supervisor, Dr Suhairi Abdul Sata, who has guided me through completing this report with his knowledge and experience on my project. To my one and only comrade under the same supervisor, Vengdasan A/L Kumaresan, he also deserves some token of appreciation as he taught me on how to install Python Programming Language in my computer to initiate my project through simulation.

TABLE OF CONTENTS

LIST OF FIGURES.....	5
ACKNOWLEDGEMENT.....	2
TABLE OF CONTENTS.....	3
ABSTRACT.....	6
CHAPTER 1: INTRODUCTION.....	7
1.1 Virtual Laboratory.....	7
1.2 Problem Statement: Case Study.....	9
1.3 Research Objectives.....	10
1.4 Scope of Case Study.....	11
1.5 Significant of Case Study	12
CHAPTER 2: LITERATURE REVIEW.....	13
2.1 Introduction of Literature Review.....	13
2.2 Implementation of Virtual Laboratory for Polymerization Cases.....	13
2.3 Implementation of Virtual Laboratory for Other Cases.....	17
2.4 Dynamic Simulation Method.....	24
CHAPTER 3: METHODOLOGY.....	27
3.1 Introduction to Methodology.....	27
3.2 Mathematical Model Development.....	27
3.3 Python Software Computational Development.....	33
CHAPTER 4: RESULTS AND DISCUSSIONS.....	38
4.1 Reaction Temperature Determination.....	38
4.2 Ability to Duplicate Result From Other Researchers.....	41

4.3 Dynamic Simulation Problem Formulation.....	44
CHAPTER 5: CONCLUSION.....	49
REFERENCES.....	51
APPENDICES.....	56
Coding A:.....	56
Coding B:.....	57
Coding C:.....	62

LIST OF FIGURES

Figure 4.1 Schematic diagram for styrene polymerization in a batch reactor	38
Figure 4.2 Reaction temperature when the timeline is at the ideal period, 3000 seconds.....	39
Figure 4.3 Reaction temperature when the temperature increase from 373 K to 374 K.....	40
Figure 4.4 Results for mass balance styrene polymerization in the batch reactor for 3000 seconds.....	41
Figure 4.5 Monomer conversion and initiator conversion with and without gel effect via gPROMS software by Ibrahim, W. H. B. W. (2012).....	43
Figure 4.6 Monomer conversion and initiator conversion without gel effect via Python software.....	43
Figure 4.7 The trend of monomer conversion, initiator conversion and initiator concentration when its initial values were being manipulated.....	46

TRAINING SIMULATOR: BATCH POLYSTYRENE REACTOR

ABSTRACT

This study is aimed to produce some sets of coding that shown same results as other researchers via different coding language. Thus, a case study on polystyrene reaction in a batch reactor was conducted by applying Ordinary Differential Equation (ODE) into a software named Python Programming Language. To ensure the success of the case study, the review on the development of virtual laboratories and simulations through literatures were a must. A lot of literatures were explaining about their initiatives towards digitalization era by constructing and applying virtual-laboratory-and-simulation-based for their studies. The efforts were supported by realizing the benefit of digitalization, which can save a lot of money, time and energy for the efficient way. ODE function was fully explored about its correct way of coding, technique and correct annotation of Python that make it differ from other languages. Case study of polystyrene reaction in the batch reactor was conducted by referring to other literatures with same-alike case study. The reaction use 2, 2' azobisisobutyronitrile catalyst (AIBN) catalyst as an initiator. The monomer conversion, initiator conversion and initiator concentration were the most important criteria that were being studied.

CHAPTER 1

INTRODUCTION

1.1 VIRTUAL LABORATORY

Process engineers and plant operators can benefit from virtual laboratory (or virtual lab) as educational tools. They offer a versatile and user-friendly way to specify the experiments to be run on the mathematical model (Jimoyiannis & Komis, 2001). Virtual labs enable users to create and run their own simulation experiments in front of the impending wave of automation and digitalization, which is predicted to supplant traditional engineering employment with advanced mechanical and digital hybrid jobs. Virtual labs allow several people to access the same tools at the same time because they are replicas of an environment. The number of users is restricted only by the number of licenses utilized, depending on the software. Even with simulated defects and statistical error appearances, the simulation of digital processing frequently comes near to the real behavior.

There are three types of virtual lab (Martin, C *et al.*, 2008). The first type is simulation of the mathematical model that describes the system's relevant features. The second is the virtual-lab view, which is an interactive user-to-model interface, and the third is narrative, which gives information about the system and how to use the virtual-lab. The virtual-lab view is designed to give users a visual depiction of the model's dynamic behavior and to make it easier for them to interact with it. The view automatically displays any change in the value of

a model variable. Any user interaction with the view alters the value of the relevant model variable in a reciprocal manner.

Virtual labs are designed to simulate the plant's real-time response so that the user may make real-time decisions. The user can perform interactive actions on the model at any point during the simulation run in these types of applications. The user can alter the model's inputs, parameters, and state variables and see how these changes affect the model's dynamic immediately. During a simulation run, any number of actions on the model can be performed. Runtime interactivity is the name for this form of interaction. In other circumstances, the purpose is to acquire the model's simulated response for a given time period and use it to compute linear approximations to the system and their frequency-domain properties, as well as to execute automatic controller synthesis (Martin, C *et al.*, 2008). In this other type of application, the user's activity initiates the simulation, which continues until it is completed. The user is not permitted to interact with the model while the simulation is running. The results of the simulation run are displayed, and a new user's action on the model is permitted. Batch interactivity is the name for this form of interaction. This paper discusses the implementation of a virtual lab with runtime and batch interaction.

Virtual labs provide a significant advantage in terms of safety. Because the entire experiment is simulated, there are no difficulties with hazardous materials and no waste management is required (S. Frerich *et al.*, 2014). Both types of laboratory experiments can be used, depending on the knowledge of the student or operator. As a result, it is possible to design and moderate individually structured learning processes. As a result, users become active

participants in their own learning, which encourages them to continue studying. Individual competencies can be developed in virtual labs. Each student is required to conduct independently planned experiments while dealing with real-world engineering difficulties and assessing their knowledge level. As a result, the jury's verdict was heavily influenced by the execution of didactical notions. Student knowledge should be augmented on an individual basis, however it was required to specify how the relevant improvement in knowledge would be rated (S. Raivo and S. Seiler 2013). Positive feedback was given to submissions that included realistic experiments with incorrect results or measurement errors. Experiments with boundary or unstable parameters also reveal limitations and are necessary in practical circumstances.

1.2 PROBLEM STATEMENT

The polymerization reaction is usually multivariable, has nonlinear features, and has a lot of temporal delays. Polymer characteristics are characterized in terms of distributions of chain length, molecular weight, and composition since polymers are made up of macromolecules of non-uniform length, hence non-uniform molecular weight and possibly non-uniform content (Babatunde A., 1994). Majority of chemical procedures were carried out by hand. The requirement for automation like simulation grew as production quantities increased and the linkages between processing units got more complex (Hagglom K., 1993).

As a result, in the 1950s, the use of basic feedback control to tackle key chemical process automation challenges became popular. However, during the following decades, there

has been a continuous development in environmental consciousness, as well as a general demand for more constant high product quality and more energy efficient use of energy (Haggbloom K., 1993). From the magic of automation mode implemented like simulation and virtual laboratory, the complex reaction with a lot of consumption on times, energies and spaces were drastically reduced.

Based on some of the materials I've read, I'd like to propose some ideas for the polymerization process using a digital computer, sometimes known as virtual labs. As a result, a case study of a virtual laboratory batch polystyrene reaction will be carried out utilizing the Python computer language. Python should be the savior when it comes to simulating this procedure, which may face a lot of issues if it is being run physically.

1.3 RESEARCH OBJECTIVES

This study outlined the following objectives:

- a) To implement the simulation and virtual laboratory on polystyrene reaction in the batch reactor via Python Programming Language.
- b) To duplicate others result via different coding medium for a bench mark of success in this study.
- c) To investigate the relationship between each initial value of monomer conversion, initiator conversion and initiator concentration towards reaction time.

1.4 SCOPES OF CASE STUDY

The scopes of case study on polymerization reaction in the batch reactor were discussed. Simulation research on polystyrene reaction in the batch reactor will be conducted using Python programming language. The software can be downloaded via bypass from Anaconda file. Python also can be coded online via www.replit.com medium without any software installation. The process simulation is using 2, 2' azobisisobutyronitrile catalyst (AIBN) as an initiator in the batch reactor. The simulation results also focused by varying each initial value of monomer conversion, initiator conversion and initiator concentration towards reaction time.

1.5 SIGNIFICANCES OF CASE STUDY

A model named Control Vector Parameterization (CVP) technique had been used by Ekpo and Mutjaba (2004) to simulate the dynamic optimization of styrene polymerization in batch reactors. Ibrahim, W. H. B. W. (2012), further improved the work from Ekpo and Mutjaba (2004) and compared the results by using gPROMS software. gPROMS enables model developers to write models of the most complex processes and their operating procedures using a sophisticated natural language, ranging from the intricate mathematical equations for individual components to the structure and operation of large complex systems made up of numerous such components.

Here, the significance of this study was about the usage of ODE function within Python software to run simulation researches on the polystyrene reaction in the batch reactor using the same model equations in Ibrahim, W. H. B. W. (2012)'s study. If the simulation researches were successful, it signified that Python software was also capable and suitable to solve any optimization problems instead of CVP technique in the future works.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter provided a literature review on the implementation of virtual laboratory and simulation as the urges of digitalization era and Industrial Revolution 4.0. The topic discussed including the implementation of virtual lab and simulation from the authors who wrote articles for polymerization processes, and well as other processes that are applicable for simulation and generation of result virtually. All the information could be obtained from articles, journals and books.

2.2 IMPLEMENTATION OF VIRTUAL LABORATORY FOR POLYMERIZATION CASES

Hosen M. A. *et al.* (2011) used MATLAB software to regulate polystyrene batch reactors using neural network based model predictive control (NNMPC). The NNMPC algorithm was employed to manage the temperature of a polystyrene (PS) batch reactor, and the controller set-point tracking and load rejection performance were examined to implement the virtual labs. Because the end use qualities of the resulting polymer are heavily dependent on temperature, it is common practice to optimize the reactor temperature profile when running

polymerization reactors. The neural network predictive controller contains a neural network model that was trained using data from the hybrid model and serves as a nonlinear plant for predicting future plant performance. The neural network plant model, which is a system identification, is the first step in building the model predictive control. The controller then uses the plant model to predict future movements. The neural network plant model predicts future plant output values based on prior inputs and plant outputs. The neural network plant model's structure was also established in the paper.

MATLAB software was used to simulate the polymerization first principle-NN kinetic model outlined. The operational conditions of the reactor and the design parameters employed in simulations are being determined. Plotting the observed values of the reactor temperature against the first principle-NN kinetic model predictions demonstrates the model's prediction capacity. Next, we'll look at how to create a neural network dynamic model for a batch polystyrene reactor. The neural network is trained using a set of reliable process data as the source. The simulation of the polystyrene polymerization process models yields training and validation data sets. For training the neural network, the controlled variables heat load (Q) and plant output reactor temperature (T) were utilized as input variables at t with two time delays (t_1 and t_2), while the plant output reactor temperature at $t+1$ was employed as an output variable. The experimental data for the operational variable ranges was used to validate the data obtained by this hybrid model.

Using the experimental setup to generate the required data is, in general, a time and resource intensive task. As a result, it was chosen to use the hybrid process model to generate

the NN training data. A total of 10,000 data records were employed in the analysis. To compute the reactor temperature, the data were generated by altering the controlled variable Q and then simulating the first principle-NN kinetic model (Hosen et al., 2010). In a 2:1:1 ratio, the set was separated into training, testing, and validation. To obtain a good performance out of the NN model, you'll need a trustworthy NN configuration, which includes the number of hidden layers and the number and type of neurons in each layer. The sigmoidal activation function is employed as the activation function in the nodes for the hidden layer in this study, while the output layer uses a linear function. The NN is trained using the Levenberg–Marquardt technique, which involves minimizing a cost function.

The construction of a model and control of an auto-refrigerated polystyrene polymerization reactor was the subject of another paper by Mokhtarname R. et al. (2021). The temperature regulation of an auto-refrigerated CSTR is accomplished by an alternate control strategy that employs a vacuum system coupled to the condenser and has yet to be discussed in the literature (i.e. to the best of our knowledge). The created model is next tested with experimental data from a real-world plant. Differential equations are implemented in the MATLAB software, and the Euler method is used to solve them. A popular control method employed in several earlier studies is also simulated to highlight the heat removal potential of this control scheme. The simulation findings suggest that the first control scheme, which is already in use in our plant, has faster dynamics and better performance. Furthermore, for the polymerization process under investigation, a nonlinear model predictive control (NMPC) is designed to provide improved temperature management while satisfying the input/output and heat exchanger capacity limitations on heat removal. Then, using some standard tuning criteria,

a comparison was done with a traditional proportional-integral (PI) controller. Through simulations, some robustness and stability evaluations of the control methods explored are also offered. The simulation results clearly show that the NMPC technique is superior in every way.

This paper's case study is a sophisticated continuous bulk process based on styrene free radical polymerization that is currently accessible in the industry. The feed preparation, the first polymerization reactor, and the second polymerization reactor are the three nodes that make up this polymerization process. Exothermic and irreversible reactions necessitate efficient heat removal (Hui and Hamielec, 1972). The feed from the recycle drum, which consists of unreacted monomer and solvent, is mixed with pure styrene monomer from the storage tank in this operation. Before entering the first reactor, the homogenous mixture is heated by passing through the heat exchanger to attain the design temperature. Ethylbenzene is the solvent utilized in this procedure. During the start-up, a total volume of reacting mixture containing monomer and ethylbenzene is heated to 415K to allow the monomer styrene to generate radicals and launch the reaction. The first and second reactors' inlet and exit mass flow rates are 6.13 ton/h. The polymerization reaction is carried out in two consecutive CSTRs of volume 21 m³ to achieve better styrene monomer conversions. Some of the heat exchangers cool the reactors, while the reactor jacket is kept at reaction temperature and does not contribute to heat removal. The operating temperature and solid content in the first and second reactors are 415K, 38 percent and 425K, 63 percent, respectively, under these conditions. Finally, the solvent and unreacted monomer in the second reactor's output are extracted, purified, and reintroduced into the feed stream.

2.3 IMPLEMENTATION OF VIRTUAL LABORATORY FOR OTHER CASES

Martin, C., *et al.* (2008) published a paper in which they used virtual labs in their research. The virtual labs were created using Sysquake and Modelica/Dymola software. The researchers can use the software to complete some of the duties. It can, for starters, define the experiment's description. Second, it is capable of simulating and linearizing Modelica models. Third, it can save a model simulation or linearization result to the Sysquake workspace, which can subsequently be used by Sysquake apps. The Sysquake interface can be used to launch the virtual-lab. The virtual labs were evaluated in three different scenarios by the researchers. The first concern is the control of a two-pipe heat exchanger. The control of an industrial boiler is the subject of the second case study, while the control of a batch chemical reactor is the subject of the third. The JARA Modelica Library was used to extract the study's key features. The interactive models for all three cases of study were created using the JARA Modelica package.

The goal of Case 1 is to investigate the relationship between temperature and axial coordinate, as well as to demonstrate the dynamic behavior of the open-loop plant using Modelica/Dymola. It also attempted to show how some linearization and control approaches could be used. As a result, the heat flows were modeled using the JARA library's thermal resistor models. Convective heat flow occurs between the gas mixture and the inner wall of the inner pipe, convective heat flow occurs between the inner pipe's wall and the water, and conduction heat flow occurs along the inner pipe's wall. The Dittus-Boelter correlation is used to calculate convective heat transport on the pipe and shell (Cutlip & Shacham, 1999). By linking JARA's pump models to the elements, the gas and liquid flows between the elements

are modeled. There are two modes of operation available: co-current flow and counter-current flow.

The goal of case study 2 is to show the closed-loop dynamics of the boiler for two different control strategies: manual control and decentralized PID. It was also meant to show how the boiler control system was put together. As a result, the interactive model of an industrial boiler was created using the JARA Modelica package. It's based on the process's mathematical model, which can be found at (Ramirez, 1989). The liquid water intake valve is at the bottom of the boiler, and the vapor outflow valve is at the top. The water inside the boiler is heated on a continuous basis. The model is divided into two control volumes, each of which contains the mass and energy balances. The first is a control volume that contains the liquid water in the boiler, and the second is a control volume that contains the vapor created. Both control volumes are linked in the boiling process model. The JARA source models are used to model the heat flow from the heater to the water, the pressure at the valve output, and the water pump. A nice example of a multivariable system is the boiler virtual-lab. The regulated variables are the water level within the boiler and the vapor output flow. The altered variables are the pump throughput and heater power. The virtual-lab allows interactive changes to the control mode, PID controller parameters, and the mass and temperature of the water inside the boiler at any moment throughout the simulation run. It can also handle changes to the mass and temperature of the vapor inside the boiler, the total inner volume of the boiler, the water inlet temperature and input flow rate, and the vapor valve opening and downstream pressure.

Case 3 was investigated in order to demonstrate the dynamic behavior of a reactor and the use of a heuristic control approach. It also attempted to demonstrate how a heuristic control policy may be used. As a result, the JARA library was used to create a model of a batch chemical reactor. In the liquid phase, reactant A undergoes an exothermic reaction to create P. A heat exchanger is included in the reactor, which can be used with steam or cooling water. The reactor's operation policies should be followed, such as filling the reactor with the reacting liquid and preheating it to a specific temperature. Allowing the reaction to run adiabatically is also critical. Then, once the maximum permitted reaction temperature has been reached or the desired conversion has been achieved, begin cooling to the desired temperature. Finally, decommission the reactor (Froment & Bischoff, 1979).

For this experiment, the user can choose between manual and automatic control interactively. The automatic control follows the previously specified operating policy. In the virtual-window, the value of the PID-controller settings, the temperatures defining the operating policy, and the desired conversion can all be modified interactively. The user can alter the value of the model state-variables, which include the temperature, A and P concentrations, and total mass of the reaction mixture at any moment throughout the simulation run. Following that, the simulation can be tweaked to adjust the reactor volume section, the heat exchanger area, the physical–chemical data of the steam and cooling water, and the input variables. The charts in the secondary window will show the time evolution of the key process variables during the experiment.

J. L. Guzmán *et al.* (2005) published a report that looked into virtual labs as well. The focus of the paper is on a case study involving greenhouse climate control. The study's goal is to make it as simple as possible for students to organize their research papers. It also aimed to make the study and comprehension of the issues involved in greenhouse modeling and climate control easier. The tool presented in this paper, in particular, comprises a structured graphic screen that provides access to a complete greenhouse climate model with a set of unique controllers. This allows students to put their theoretical knowledge into practice without having to worry about space or time limits.

This paper introduces a tool that aids in the completion of this activity. During the last few years, a large number of programs and research projects aimed at increasing education in the field of automatic control have been developed (Dormido, 2004). Many studies on interaction, web-based simulation, and virtual labs (Dormido *et al.* 2003) have opened up a wide range of possibilities as educational support for students. Some of these works, such as virtual courses (Rodríguez *et al.*, 2004), are being used in undergraduate and doctoral courses to teach greenhouse climate control. This paper proposes a virtual lab for teaching greenhouse climate control to complement prior studies and progress the subject. This program was created with the help of Easy Java Simulation (Esquembre, 2004) and Rodríguez (2002)'s study findings on greenhouse modeling and climate control.

Temperature control, humidity control, and radiation control are some of the variables involved in greenhouse climatic control. According to the authors, the software utilized is MATLAB/Simulink and Easy Java Simulation, which have a good connection. There are three

buttons, three edit fields, and one checkbox for the control settings in the first column. The buttons allow you to start, stop, and restart the simulation. The temperature and radiation set points can be changed in the edit areas. The checkbox determines whether the simulation begins during the day or at night. The control parameters for temperature control with ventilation are found in the next two columns. The three initial parameters are used to adjust the PI controller with anti-windup to control the temperature with heating in the last column.

The simulation workshop for process control education was carried out for the next publication by B. R. Young *et al.* (2001) to create fantastic virtual labs for undergraduate chemical engineers. The authors cite the software HYSYS, Aspen Dynamics, and MATLAB, which are utilized in chemical engineers' process control courses. Chemical process control necessitates a multivariable and plant-wide approach since chemical processes are characterized by high degrees of nonlinearity, process interactions, and significant dead time. In an ideal scenario, a student chemical engineer would be able to experiment on a "virtual plant". This plant would capture all of the key non-idealities that the real world imposes, allowing the student engineer to try even the most bizarre control schemes without fear of being caught.

Teachers at the University of Calgary have organized some workshops for their students. One of them is about the concepts of feedback control loops. The surge tank or separator is a common process device that can be used to study capacity-dominated process behavior. For linear and non-linear outlet valves, the level response of a single tank to changes in inflow flow and temperature is initially investigated. The temperature response of the tank

with steam heating was also investigated, as well as the addition of a plug flow pipe element to simulate dead time. The fundamental understanding of the processes is used to manage the reactions of the processes through feedback control. Examining the response to set point and disturbance changes, the influence of dead time and capacity on P, PI, and PID control of the steam duty stream is investigated. The regulation of the tank level via averaging is also investigated.

The next step is to fine-tune the capacity and dead time processes. The goal of this workshop is to determine the optimum parameters for a PI controller that is operating a capacitive process with a lot of dead time. For set point and disturbance tuning, the closed loop tuning technique of Ziegler and Nichols (J. G. Ziegler *et al.*, 1942) and the auto-tune variation tuning technique of Astrom and Hagglund (K. J. Astrom *et al.*, 1984) are used, and the results are compared. The contributions of the controllers to attenuation are also determined and compared to the previous workshop's results. Finally, the workshop provides significant assistance to students, as it is based on fundamental process models of industrial unit operations that are implemented utilizing educationally affordable and easily available commercial process simulation software.

Martin C. *et al.* (2011) published a paper regarding the creation of virtual labs for chemical process control education using Modelica and C++ software. The authors have outlined the stages involved in developing virtual labs. The interactive library, which includes Modelica classes and code needed to communicate between the model and the display, was also included in the study. The virtual-lab designer simply needs to set the length of the model-

to-view communication interval, and the code is valid for any model and view descriptions. To begin, the Modelica model must be modified to accommodate the interactive simulation. It then has two parameterized generic classes, which are the virtual-lab model and view classes. In addition, the virtual-lab designer must specify the required equations and how the model and view class variables are related in writing. The descriptions are then translated and run using Dymola. As a result, the virtual-lab view's C++ code is generated automatically during model startup. The model's and view's communication procedures are also set up. This communication is built on a client-server architecture, with the server being the C++ program generated by Dymola (Dynasim AB, 2008) and the client being the C++ software generated automatically during model configuration.

The virtual-lab user must execute the C++ program once it has been constructed. As a result, the virtual-lab view's basic layout is shown, and client-server connection is created. The model simulation then begins. There is a bi-directional flow of information between the model and the view during the simulation run. Every connection interval requires critical communication, such as sending the model simulation to the client and the data needed to refresh the view. The view then sends the updated value of the variables modified as a result of the user's interactive activity to the model simulation. The author is now researching a control scenario that exhibits the open and closed loop dynamics of a double-pipe heat exchanger. Manual and automatic PID control techniques are discussed. The heat exchanger using a PID controller. The purpose is to manipulate the liquid input flow to control the gas exit temperature. It has been using the PID controller model supplied in the standard Modelica library (Modelica Association, 2011), which was created using an approach from Aström and

Hagglund (1995). Limited output, anti-windup compensation, and set point weightings are all features of this model.

2.4 DYNAMIC SIMULATION METHODS

Dynamic simulation methods must be reviewed in this study because the focus of study was on the duplication of the related case study based on Ordinary Differential Equation (ODE) function within Python software. The reviewed sub-topics cover Python software and its ODE function.

2.4.1 PYTHON SOFTWARE

Python is a general-purpose programming language with a high level of abstraction. Its design philosophy prioritizes code readability and makes extensive use of indentation. Its language elements and object-oriented approach are aimed at assisting programmers in writing clear, logical code for both small and large projects. Python is garbage-collected and dynamically typed. It supports a variety of programming paradigms, including structured (especially procedural) programming, object-oriented programming, and functional programming. Because of its extensive standard library, it is often referred to as a "batteries included" language.