A SPATIAL DECISION SUPPORT SYSTEM FOR FLOOD HAZARD IN QUANG NAM PROVINCE, VIETNAM

A thesis

submitted in partial fulfilment

of the requirements for the Degree of

PhD. of Geographic Information Science

at

Lincoln University

by

Ong Dinh Bao Tri

Lincoln University

2017

Abstract of a thesis submitted in partial fulfilment of the requirements for the Degree of PhD. of Geographic Information Science

A SPATIAL DECISION SUPPORT SYSTEM FOR FLOOD HAZARD IN QUANG NAM PROVINCE, VIETNAM

by

Ong Dinh Bao Tri

The general aim of this study is to provide decision makers and planners with a spatial decision support system (SDSS), for quantifying flood hazards in order to cope with deluge situations within the Quang Nam basin of Vietnam. This flood management SDSS provides a comprehensive set of tools for rainfall and runoff modelling, hydraulic modelling and frequency analysis for the examination of the flood nature in the Quang Nam basin. This system is implemented in the context of complex aspects of hydrological features present in the province, such as relationships between the river systems, rainfall patterns and topographical features of the basin. The concrete result of the study is an IT system that is developed on the information derived from the aforementioned context using Visual Basic programming language running on top of a GIS platform. This exploration is 'state of the art' in its usage of modern science and technology in quantifying the aspects of flood hazards, including the mapping of flood areas and ranking of different levels of probability in multi-scale and temporal resolution. This analytical approach is also a reflection of a 'best practice' methodology which is a combination of enhanced local knowledge and modern technology: that is a GIS-based flood simulation and analysis, to assist decision makers to gain quick feedback through interactive spatial IT tools.

Keywords: Quang Nam province, Vietnam, Spatial decision support system (SDSS), Geographical information system (GIS), multiple regression, NOVA analysis, optimization, continuity equation, energy balance equation, gradually varied flow, Simulated Annealing, calibration, validation, frequency analysis, Normal distribution, Gumbel distribution, general extreme value distribution (GEV), Kolmogorov – Smirnov test for goodness of fit.

Acknowledgements

First and foremost, I would like to express my sincerest gratitude to Dr. Crile Doscher, who has supported me throughout my thesis with his patience and knowledge whilst giving me the room to foster my own passion. Without his encouragement and effort and without him this thesis would not have been completed. One simply could not wish for a better or friendlier supervisor.

I am extremely thankful to Dr. Magdy Mohsshen, with his expertise, he has offered much advice and insight into hydrology directing me to a new horizon, opening a new vision for my knowledge. He is not only a supervisor but also a true friend who I can share thinking not only about hydrology, mathematics and programming but also about life.

I would like to thank my fellow doctoral students for their feedback, cooperation and of course friendship. I am especially thankful to David Bird, a true friend I found here in New Zealand. I would never forget the days we work together in Forbes building lab towards our future, sharing the vision about life and work, passion and dream.

I owe my deepest gratitude to my wife, daughter and son who sacrificed part of their life while I was away from home, whilst supporting me with all their love over those long years.

This study could not have been completed without the financial support from the New Zealand Scholarship Programme which provided me with the ASEAN Scholar Awards making my long life dream comes true.

.

Contents

List of Figures

List of Tables

Chapter 1. INTRODUCTION

1.1. Introduction

In Vietnam, flooding is one of the major threats to the development of the country. In terms of affected area, severity, frequency, losses and damages to people and development, among natural hazards, flood is ranked first. Although there has been significant effort to mitigate and adapt to the damage caused by floods in Vietnam, there is still a lack of knowledge about technologies and methodologies appropriate for flood risk management. Most of the mitigation and adaptation approaches have been decided on an ad hoc basis without a concrete scientific foundation and not in a systematic manner. This open a new door for an enhanced approach, where flood risk management can be undertaken in a more comprehensive manner. There is a requirement for a more systematic approach that takes advantage of modern technologies to provide meaningful information for decision makers and planners to carry out estimation, adaptation and mitigation measures to cope with the flood hazard in Vietnam.

The general aim of this study is to provide decision makers and planers with a spatial decision support system (SDSS) for quantifying the flood hazard physical conditions, which are the fundamental information for the decision makers and planners to formulate actions to prevent the hazards, to adaptize and mitigate the effects of floods to reduce losses and adverse consequences on the environment, using Quang Nam province as a case study.

The system is implemented in the context of complex aspects of hydrological features of the province, such as the relationships between the river systems, rainfall patterns. The concrete result of the study is a computer based system that is developed on the information derived from the above mentioned contexts. This should be the application of modern science and technology in quantifying the aspects of flood hazards, including the mapping of flood areas in multi-scale of temporal resolution. This is also a reflection of best practice approach which is a combination of enhanced local knowledge and modern technology: a GIS-based flood simulation and analysis, to support the decision makers to gain quick knowledge in making decision in flooding risk management through interactive spatial tools.

1.2. Aim and objectives

As highlighted above, there is a lack of knowledge about technologies and methodologies appropriate for flooding risk management in Quang Nam province specifically and in Vietnam as a whole. For Quang Nam province, the current operation of the flood management system shows that it relies heavily on local experts' understanding about the nature of local hazards for operational decisions. From this, the primary aim of this study is to provide decision makers and planners with a spatial decision support system, for quantifying/ estimating/ simulating the flood hazard nature in Quang Nam province. The system will provide decision makers and planners with sufficient information to make sound decisions to control and mitigate the effects of floods on the economy and environment of the province. To achieve the above mentioned aim, the decision support system for Quang Nam basin will have the following specific objectives:

- Develop a database including attribute and geographic data that provide inputs for all the models in the tool sets.
- Develop a toolset for rainfall-runoff modelling, allowing for estimation of runoff from rainfall data.
- Develop a toolset for simulating the flood inundation extent for real-time and corresponding to "chosen" design flood scenarios.
- Develop a frequency analysis tool set which includes analysis tools for generating information for planning purposes.

- Unify these tools into an integrated toolbar as an extension to the functionality of ArcGIS with user friendly UIs (user interfaces).

The system is a demonstration case of the application of a SDSS for flood hazard management in a broader context of constrained/limited conditions of personnel capacity and financial resource in river basins of developing countries. Lessons learnt from this project will form the general guidelines for development of SDSS for flood hazard management in terms of technology used, the suitability of technologies for local capacity, system structure and development, setting up and configuration, system operation and maintenance.

Chapter 2. QUANG NAM BASIN AND FLOODING IMPACTS

2.1. Quang Nam Basin

Quang Nam basin is located in the central region of Vietnam. The basin is composed of two main rivers, the Vu Gia and the Thu Bon. It is located between the Truong Son mountain range in the west and the East Sea in the east. The basin occupies about 10,350 km^2, roughly 90% of the Quang Nam Province. It also includes Da Nang city with a population of about 876,000. Figure 2.1 shows a map of Quang Nam province and Da Nang city in central Vietnam, containing the Vu Gia – Thu Bon river networks.

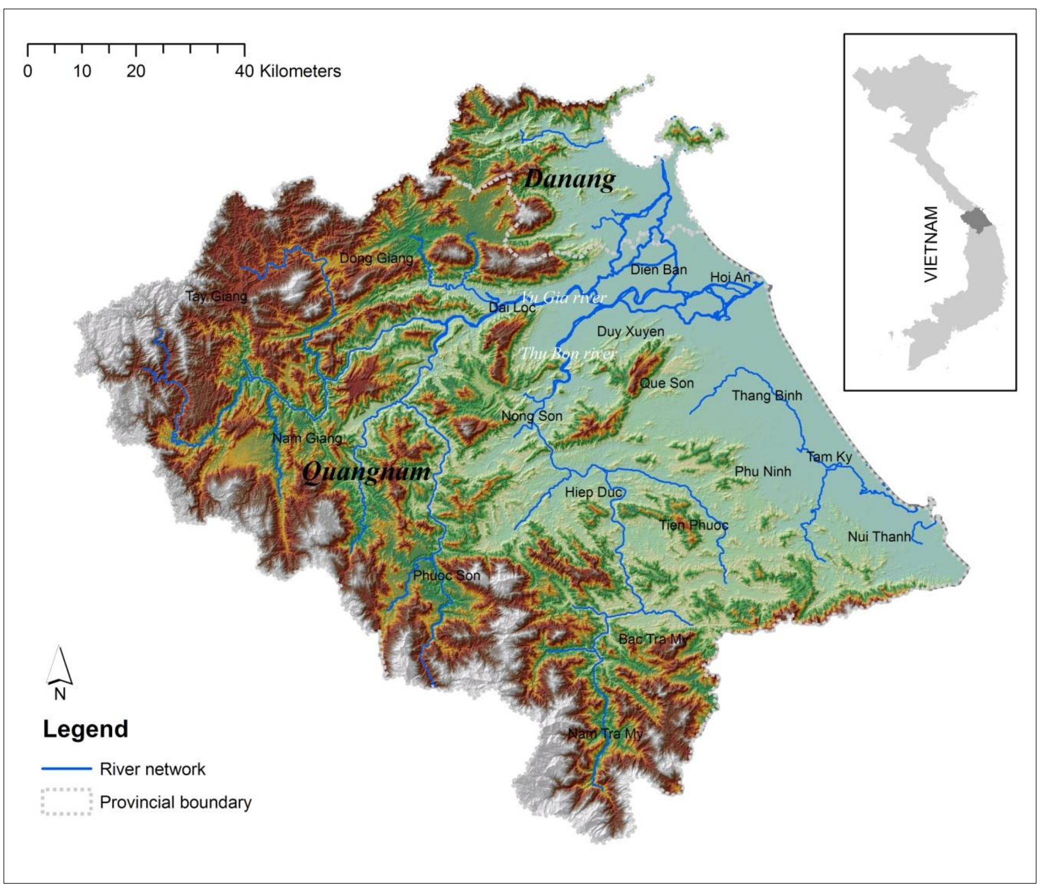


Figure 2.1 Quang Nam basin and its river network

Within the basin, there are two distinct topographic areas: the relatively narrow mountainous area with a maximum elevation of 2,600 m that features a large number of steep tributaries, and the flat coastal zone with interconnected coastal river system subjected to annual flooding.

Land use in the coastal plain includes intensive agriculture, mainly rice, recently established industrial zones that are rapidly expanding, particularly around Da Nang and Tam Ky, and tourist resources along the coast including Hoi An and the holy lands of My Son. Further upland there is a combination of subsistence agriculture and forest land, including almost 100,000 ha of the provincial Song Thanh Nature Reserve. The forest covers about 42% the area of the province. The geology is comprised mainly of metamorphic rocks, including granite, with numerous fault structures.

The Vu Gia and Thu Bon are two main rivers of the basin. The rivers originate in the highlands near the border with Laos and flow into the ocean via mouths in Da Nang city and Hoi An city. The Vu Gia and Thu Bon join approximately 36 km upstream from the coast. The Vu Gia river has many tributaries, i.e. Dak Mi (also called the Cai river), Bung, A Vuong, and Con rivers. The length of the Vu Gia river up to the mouth in Da Nang is 204 km. The Thu Bon River originates at the border of the three provinces of Quang Nam, Kon Tum and Quang Ngai, at an elevation of more than 2,000 m. It runs in a North-south direction towards Phuoc Hoi where the river changes its course to flow South-west – North-east and then West-east up to Giao Thuy before entering the sea through the Dai estuary. During the flood season the two rivers interact through the Quang Hue and Vinh Dien rivers forming a braided river delta system. This connection is often broken in the dry season.

2.2. Hydro-meteorological features of Quang Nam basin

The basin has a monsoon tropical climate with an average annual humidity is 84% (ICEM, 2008). Northeast wind occur from October to March, with an average velocity of 6 to 10 m.sec^{-1} (ICEM, 2008). From May to August, there are southern, southeast, and southwest winds, with an average velocity of 4 to 6 m.sec^{-1}. The average temperature is 25.4^{O} C while temperature in winter fluctuates between 20 to 24^{O}C (ICEM, 2008).

The monsoon season in Quang Nam and Da Nang lasts for four months, from September to December. The dry season is from January to August. In May and June there is a secondary rainfall peak, which is more pronounced towards the north-western part of the study area. Total annual rainfall varies from about 2,000 mm in central and downstream areas to more than 4,000 mm in the southern mountainous areas. Rainfall during the monsoon season accounts for 65 to 80% of total annual rainfall. The highest amount of rainfall occurs in October and November which accounts for 40 to 50% of the annual rainfall. Rainfall in the dry season represents about 20 to 35% of the total annual rainfall. The low rainfall season usually occurs from February to April, accounting for only 3 to 5% of the total annual rainfall (ICEM, 2008).

The mean annual flow volume in the basin is 19.1×10^9 m^3. Similarly to the distribution of rainfall, annual flows are distinguished by two distinct seasons (the flood season and the low-flow season). The flood season commonly starts in mid-September and ends in early January. Flows during the flood season account for 62 to 69% of the total annual water volume, while flows in the dry season comprise 22 to 38% of total annual run-off. The water volume gauged in November, the highest flow month, accounts for 26 to 31% of the total annual run-off while the driest period is April with flows of 2 to 3% of the total annual run-off (Nguyen, 2011).

The Vu Gia - Thu Bon basin is prone to natural events such as flooding and typhoons. Heavy rainfall is normally caused by northeast monsoon and typhoons. In addition, the Vu Gia – Thu Bon river system has no dike embankments or stop banks. When floods occur, large areas of the floodplain are typically inundated. As the majority of the population lives in these low lying areas, this exposes them to the potentially disastrous consequences of these events.

2.3. Flooding in Quang Nam basin

In 1964, large floods caused inundation to extensive areas in the central region of Vietnam from Quang Binh to Phu Yen. The largest historical flood event in 1971 broke dykes and caused severe inundation in many provinces in the Bac Bo river delta. In 1999, a large flood event caused inundation in Vu Gia - Thu Bon, Huong and Tra Khuc rivers leading to significant losses in human life and property in Da Nang City, and in Thua Thien Hue, Quang Nam, and Quang Ngai provinces. With a maximum rainfall of 120 mm/hour, the event produced serious flooding of the central coastal provinces of Vietnam killing 324 people and causing an economic loss of US$ 112 million (Huynh *et al.*, 1999).

According to the Institute of Geography (2010), in Quang Nam province, floods can be categorized by timing as early, main and late floods. Early floods occur from September to mid-October each year and are usually small to medium in magnitude. Around 30% of floods are early floods and occur most often in the Vu Gia - Thu Bon river system. Early floods are usually limited in duration and their extent is localised. During late October and November, the main flood period, heavy rains frequently cause floods. Early flooding saturates the soil, reducing the capacity of the soil to absorb more during this main flooding period. Most of the runoff from rain is then directly discharged into the river system. The combination of steep terrain, a dense river network and intense heavy rain leads to extensive flooding in Quang Nam. Late floods happen between December and January, accounting for 30% of the floods in a year. Rains caused by the northeast monsoons are the main source of water of floods in this period. The late floods are usually low magnitude in both severity and inundation extent.

In the Thu Bon – Vu Gia basin, due to the low drainage capacity of the downstream reaches of rivers, inundation usually last longer in the lower reaches while in the upper and middle reaches of the rivers, the intensity of heavy rains combined with steep terrain and narrow river beds lead to a rapid change in flood at an average of 20-50 cm/hour (Institute of Geography, 2010).

The hydrological features of Vu Gia - Thu Bon catchment make it vulnerable to flooding hazard causing severe damage for local populations and economies. The flood event in November 1964, caused significant flooding over the entire catchment. This historical flood event killed more than 6,000 people and destroyed many villages. The one week flood event of November 1999 caused severe inundation in the downstream districts of Quang Nam province and Da Nang city. The water overflowed the roads, railways. Many locations and villages were isolated. Traffic and communication were interrupted and almost all of the design capacities of lakes within the catchment were exceeded. The flood killed 118 people and caused a VN$ 758 billion of damage to the local economy. The flood event in 2007 inundated 125 out of 233 communes of Quang Nam province, affecting approximately 200,000 households. Communication and power supplies were interrupted in many areas. Most roads were blocked and transportation along highway 1A was obstructed for 40 hours. The number of people killed was 47 and infrastructure was damaged drastically. The event caused an economic loss of about VN$2,000 billion. In 2009, after several days of heavy rainfall from a tropical depression, many parts of region were completely inundated: 52 people were killed, 5,200 houses collapsed, and the economic loss was around VN$3,500 billion. These loses have been very substantial in the context of the underdeveloped economy of Vu Gia - Thu Bon catchment (Nguyen, 2011; RETA 6470, 2011; Vu *et al.*, 2011).

Chapter 3. LITERATURE REVIEW

3.1. Spatial decision support systems as a soft approach to flood hazard

In the context of flood management, a hard (structural) approach is one of the traditional options. However, this approach requires significant government investment for the construction of infrastructure inside the flood plain, and may also cause adverse social and ecological impacts (Jason *et al.*, 2005). As an example, it has been estimated that one-third of all flood disasters in the USA were caused by levee failures (National Research Council, 1982). In some countries, non-structural (soft) approaches have been adopted as a measure for reducing the impacts of floods. In general, non-structural approaches use a wide range of measures to reduce the vulnerability of population centres and properties, including infrastructure. The measures can be the optimization of upstream land management practices, planning and management of flood plains (i.e. restrictions imposed on development inside flood plain areas) and planning for disasters proactively, including emergency response systems, evacuation schemes, etc (Jason *et al.*, 2005). In such a situation, flood risk management decision support systems (DSS) could be used as a tool for providing input information for the above mentioned wide range of tasks. In practice, this could include a number of types of analysis such as risk assessment, technical and cost problems, loss analysis, and prevention or mitigation analyses (Chen, 2011).

3.2. Flood management decision support systems

Generally, when talking about decision support systems, it is frequently implied that they are computer based systems which are able to improve the effectiveness of the decision making process. One of the general formal definitions made by Loucks and daCosta (1991) is that DSSs are "computer based tools having interactive, graphical, and modelling characteristics to address specific problems and assist individuals in their study and search for a solution to their management problems". It is necessary to note that, in general as well as in a flood management context, flood decision support systems improve the decision making process not by prescribing a particular course of action, but by providing data displays, analytical results, and model outputs on critical information (Jason, 2005). Simonovic (1999) recommended the following description of a decision support system in the context of water resources decision making: “It is a system that allows decision-makers to combine personal judgment with computer outputs through a computer interface to produce information for the decision-making process”. This means that such systems can assist decision making for different types of problems: structured, semi-structured and unstructured. They are an integral part of the decision-maker's approach to problem identification and solution (modified after Parker and Al-Utabi, 1986; Thierauf, 1988; Simonovic and Savic, 1989 cited in Simonovic, 1999).

In reality, different studies have developed different tool sets for coping with different situations. Therefore, decision support systems tend to be described differently in terms of both functionality and scale.

The objective of a decision support system in flood modelling could be quite specific, such as the flood warning decision support system for Sacramento, California by David and Member (2001). In this system, the authors describe a flood warning decision support system used to increase warning lead time for Sacramento County, California. Real time rainfall depths and water levels are transmitted to an emergency operation centre. The real time data are used to assist flood threat recognition visually via rainfall pattern surface fitting or automatically by the setting of warning rules based on the knowledge and experience of local flood plain and emergency management experts.

A decision support system could be understood as a theoretical formulation for making selection of decision alternatives. Jason *et al.* (2007) developed a multi-criteria decision support system (MCDSS) with application to the 11 – 12 September 2000 Tokai floods in Japan. The MCDSS was used to enhance stakeholder communication and improve emergency management resource allocation by explicitly making links between flood knowledge, assumptions and choices. Through the MCDSS, stakeholder satisfaction increased, lives were saved, and flood management costs were reduced, thereby increasing decision-making effectiveness, efficiency and transparency. All of the processes are centred around the analytic hierarchy process technique.

A typical example of decision support system in flood management is the one developed by Honghai and Altinakar (2011) for flood damage assessment by integrating flood management within the framework of a GIS, based on two dimensional flood simulations (loosely coupled with the CCHE2D cohesive sediment transport model in freshwater). The system can interact with classified remotely sensed layers and other GIS vector layers like zoning layers, survey database and census blocks for flood damage calculations and loss of life estimations. It is customizable for inputting user defined criteria, such as stage-damage curves. The result can be graphically displayed.

Shim *et al.* (2002) proposed a prototype spatial decision support system (SDSS) for real time flood control for a multi-reservoir system. The SDSS was based on a geographic information system, a database management system, a real-time meteorological and hydrological data monitoring system, a model-base system for simulation and operational optimization, and is interacted with via a graphical user interface. Spatially distributed forecasted flows are updated by the model-base within a real-time flood forecasting module as the flood event progresses. In real time, optimal gate control strategies are updated based on the basin wide discharges using a dynamic programming module. The application of an SDSS in the Han river basin of Korea for the severe 1995 flood event indicated that the integrated operational strategies generated by the SDSS for flood control substantially reduced downstream flood impacts while conserving sufficient water for later use.

A decision support system in a flood management context could be comprehensively applied for the management of a large geographical scale river basin, like the one which is being developed for the Red river basin (Simonovic, 2013). A detailed outline of the plan for the development of a Red River Basin Decision Support System has been proposed. The system is envisaged to support decision makers and stakeholders in making decisions during the planning, flood response and recovery stages with an emphasis on flood prediction and monitoring, emergency response and public involvement. The architecture of the system includes a number of components:

1) A distributed Database: to ensure that all data such as topographic, land use, hydrologic, hydraulic, environmental, and economic are accessible to all users to provide support for flood management activities. The system will include a database, with back ends distributed at various agencies, to allow for access via simple protocols. Maintaining and updating database components are assumed to be undertaken by different agencies. The setting up of the database will be based on the needs and capabilities of providing data from stakeholders and the interoperability with other components of the system. Data in the database will be used for planning and design for flood protection, real time flood emergencies and flood recovery.

2) A model base component including a number of descriptive and predictive modelling tools such as hydrologic, hydraulic, economic, and environmental models which are required to support decision making in the basin:

- Hydrologic models are used to forecast runoff in a river by combining precipitation and other inputs. It is noted that the model base component will include the existing hydrologic forecasting tools and allow for the future integration of newly developed tools
- Hydraulic models are used to route flood volume in the Red River Basin, and to calculate overland flow. The use of these models can be for a) real time forecasting (flood level, time of peaks, hydrograph calculation and inundation mapping: backwater at critical locations, incorporate infrastructure changes such as breaches and blow outs, "what if" analyses or for b) Planning and design: post flood

analyses for infrastructure evaluation and mitigation design, evaluate effects of flood operations, analyse peak reduction alternatives, sensitivity analyses, define the needs for data and monitoring. Existing one-dimensional models and one coupled with two-dimensional models (MIKE-11, MIKE-21) will be the main components of the system model base component.

- Economic models: will be used in economic assessment of structural and non-structural flood damage reduction measures. Some examples of economic models include: expected annual flood damage (EAD) (U.S. Army Corps of Engineers, 1989); structure inventory for damage (SID) analysis (U.S. Army Corps of Engineers, 1989a); and the flood damage analysis (FDA) (U.S. Army Corps of Engineers, 1988).
- Environmental models: such as spill response models and habitat evaluation models for assisting the assessment of environmental consequences of flooding. All the models currently in use in the basin will be adopted.

3) A user interface: users will access the virtual database, model base modules via a multi-level interface.

Todini (2000) described a highly comprehensive decision support system for flood planning and management that takes advantage of high performance computer platforms. The objectives of the decision support system are evaluation of inundation risk and emergency management aiming at forecasting catastrophic food events and mitigating economic, social and environmental impacts. The system allows for the 1) Identification of high risk areas and expected damage estimation for planning purposes. 2) Real time forecasting and simulation of floods and inundation through analysis of real time meteorological data at different time and space scales. 3) Assessment of the effectiveness of decisions aiming at mitigating social, economic and environmental impacts in the planned or real-time scenarios.

The system is being developed based on the knowledge, experience and lessons learnt from previous projects and existing systems, includes: i) a real time flood forecasting package developed under the European food forecasting operational real time system project (EFFORTS, 1988-1991), (ET&P Srl, 1992; Todini *et al*., 1997); ii) a system for flood risk mitigation and control (AFORISM, 1990-1993), (University of Bologna, 1996); iii) an open architecture decision support system for environmental impact assessment, planning and management (ODESSEI) (EUREKAEU487, 1994-1997), providing an open architecture framework for this decision support system to adopt; iv) flood forecasting in urban areas (TELFLOOD, 1996-1998) and v) and a study on the relationship between limited area meteorological models rainfall forecasts and food forecasting models (Previsione Regionale Idro-Meteorologica Accoppiata per la Valutazione delle Emergenze e del Rischio di Alluvione: PRIMAVERA, 1997-1998).

The comprehensive view of the system includes: i) a set of relational and mathematical models: a rainfall extremes statistical model, a semi-distributed rainfall-runoff model, a one dimensional food routing model and a combined one-two dimensional inundation model, a socio-economic and environmental impact assessment model; ii) a knowledge-based system, allowing for managing data flows to and from the database, executing the mathematical models, comparing the scenarios, and guiding users in making decisions. iii); a database management system, used to manage the historical series of meteo-climatic measures (e.g. rainfall, temperature), watershed entity data (rivers, sub-basins) and the socio-economic and environmental data; iv) a geographical information system (GIS) allowing for the management of spatial data, supported by analysis tools. The interoperation between the GIS and the models is undertaken by knowledge-based system procedures and; v) a user interface, which is a sophisticated and easy to use interface facilitating the visualization of complicated information, supporting qualitative analysis and interpretation of results. The system allows for hydrological and hydraulic analyses, evaluation and the preparation of flood risk scenario, planning of structural and non- structural measures and forecasting of flood in a real time manner providing possibility of assessment of the effectiveness of different intervention scenarios.

3.3. Role of GIS in flood management decision support system

For flood modelling, there are three broad approaches to integrating GIS (Clark, 1998): 1) Data pre-processing where spatial data and non-spatial data are processed and stored in suitable format for further processing and modelling; 2) Direct support for hydrologic/ hydraulic modelling within the GIS, and; 3) Post-processing and mapping of data. Bridging between GIS and other existing software applications and databases for flood management and simulation can alleviate the load of data management for many applications. GIS has been successfully coupled with hydrological and hydraulic models, including CASC2D, SWAT, HEC-RAS, HEC-HMS (Ogden *et al.*, 2001) and KINEROS (Miller *et al.*, 2002c). Typical specific tasks may be, for instance, creating flood risk maps, performing cost–benefit analyses for alternatives, and creating a flood decision support framework (Vermieran and Watson, 2001). An example can be given by Brimicombe and Bartlett (1996) where flood risk was assessed using hydraulic models coupled with the geographic information system (GIS) and digital elevation models (DEM) to map the areas and depths of inundation. This type of flood risk assessment provides information on the probability of flood occurrence, magnitude of the event, location, and depth of the inundation for flood management. Xihua and Bengt (2002) developed a model for predicting flood inundation and risk using GIS and a hydrodynamics model at Eskiltuna in Sweden. In this study, information for emergency planning was obtained via an integrated methodology for flood prediction using GIS and hydrodynamics modelling. DEMs and other relevant data layers (e.g. real estate, building and river channels) were prepared in GIS. These were then used as inputs for MIKE21 in a specific format required by MIKE21. The outputs were then transferred back to GIS for visualization and further analysis. It is obvious that GIS can add a new dimension to decision support systems in a flood management context, making the solutions to the realistic problems more accessible.

In recent years, together with the development of information technology, the development of GIS technology has gained momentum and advanced itself into new levels allowing for supporting the integration of different tools for solution of problems in a diverse disciplines intrinsically. Commercial GIS software packages are now mostly designed as highly interoperable systems, allowing for high levels of customization. In many cases, users can also extend the functionality of the software so that they can better suit the users' requirements. These are normally done via interface customization, application language development (e.g. Python, VBA, MapBasic), or via tools developed outside a GIS and plugged in as intrinsic components e.g. extensions or toolbars in ArcGIS. This provides for a high level of integration and interoperability of spatial decision support systems where spatial component play a fundamental role in the systems.

3.4. State of the art of spatial decision support systems

A brief description of the state of the art of the spatial decision support system in a flood management context, in terms of architecture, typically comprises the following components: Firstly, databases, these databases could be centralized or distributed and either relational or object-oriented. The databases are used for storing, processing and manipulation of meteorological (e.g. rainfall, climate conditions), hydro-geological (e.g. soil types, landuse types), administrative data (e.g. population distribution, administrative boundaries) to comprehensively describe the study area hydrologically and administratively. Spatial databases could be used to store GIS and remote sensing data, which is the backend of GIS spatial analyst and visualization and hydrologic, hydraulic models.

Secondly, a set of inter-linking flood, environmental and socio-economic models that is tailored to suit the situation requirement for the specific application area. Therefore, this may be different for different areas with different hydrologic features and management requirements. Typical models may, in general, include rainfall-runoff simulation models, discharge models, hydraulic models for quantifying, mapping and simulation of flood properties (e.g. peak runoff, discharge, flood time span) and inundation area. Many of these models were developed based on the Navier-Stokes equations with different levels of simplification applied for different hydrological and hydraulic processes. Needless to say, these models need to be tailored and calibrated so that they specifically address the hydrologic/hydraulic problems within the application area. Depending on the management needs, the models may also comprise environmental, ecological, and socio-economics models. These models are developed based on a wide range of decision making techniques such as optimization techniques (linear programming, non-linear optimizations techniques), artificial neutral network (ANN), analytical hierarchy process (AHP) or the statistics techniques. The environmental and socio-economic models are used in the assessment of the management alternatives for both planning ("what if" analyses) and real time scenario analyses in such situations as flood event evacuation, reservoir operation, mitigation activities. This second component is the core and challenging part of a flood management decision support system.

The third component of a decision support system is a computer graphical user interface. In modelling flood, environmental, and socio-economic problems, the problems are formulated mathematically or physically in computer programs with different levels of complexity. The decision making process may involve different stakeholders, with different backgrounds, who have to work together towards best management solutions using these models by interacting with each other and with the computer programs. An intuitive graphical computer interface is a crucial component. Recent computer science developments (e.g. COM component architecture, plug-in technology, event driven programming paradigm) allow for robust interaction mechanisms between the programs within the computer platform itself and with users via graphical user interfaces. Therefore, the design of an intuitive graphical user interface relies heavily on the skills of the developers rather than the extent of the capability of information technology. Typical interactions via computer graphical interface by users in a flood management context can be database management activities such as editing and updating spatial and non-spatial data, querying data for information, undertaking spatial analysis operations, spatial querying on maps, visualizing maps, constructing models and executing them, extracting spatial and non-spatial reports. The interfaces determine the efficiency and scope of the results of these interactions (Al-Sabhan *et al.*, 2003).

3.5. Challenges

Despite the highly developed levels and ubiquity of the application of spatial decision support systems in flood management, there are still several challenges which need to be addressed. Miller *et al*. (2004) identified three main concerns over the development of flood management decision support systems including: 1) Interoperability: The involvement of stakeholders into the process of making decision in the context of flood management is comprehensive. This leads to a requirement of interoperability in terms of information technology: differences in the use of programming platforms, operating systems, and databases can make effective communication difficult. Creating a centralized database repository containing watershed management data for decision making is an option. However, the problems of data sharing, conversion and updating need to be addressed. 2) Accessibility: this is one of the major concerns in the context of rural areas in flood plain of Vietnam. 3) Security: if decision support systems are hosted online, security is always a concern. Precautions need to be taken to assure application security.

Chapter 4. METHODOLOGY

The methodology will be discussed in a general sense for each objective. More specific details will be discussed in the individual toolset development sections later in this thesis.

In this thesis, the general approach for the development of the tool sets in the SDSS is taking the advantage of a high level programming language (VB.net) to build modelling tools on top of a GIS platform (ArcGIS). This approach gives the SDSS the following advantages:

- All the modelling tools can be unified into a single system, for the sake of convenience and user friendliness. With this approach, the problem of installing and maintaining multiple software applications on the same computer system could be avoided.
- The graphical support of VB.net programming language allow for building a user friendly user interface (IU). Through the customization of the interface, the developer can dictate how the users interact with the system, and to what level system and the data to be accessed. The interface could also be tailored to suit the technical capacity of users.
- By building the modelling modules on top of a GIS platform, the tools directly interact with the GIS tools and the data tier below (relational databases and geo-databases), the problems of software communication and data compatibility can be avoided, addressing the problem of interoperability. No pre or post data processing is needed.
- GIS is an excellent tool for presentation and communication, and by taking the advantage of the capability of a GIS platform, the modelling tools can communicate with user in a more intuitive manner, in map form.

- Using the programming language, additional functionalities that are not available in the off the shelf software and context specific assumptions can be integrated and tested. In this thesis, with the aid of VB.net, Graph Theory was employed to automatically assemble the system of flow equations, and the Simulated Annealing optimization technique was applied in automatic calibrating the Manning's roughness coefficients for the general channel networks. Without the flexibility of the programming language, these new functionalities would not have been possible.

4.1. Objective 1. Develop a database

This component includes structuring the schema for the databases and prepare the data itself for the system. The database architecture is a relational database with map capability.

This SDSS is a data driven system where models in the system are structured, calibrated, validated and implemented on the basis of data. This study will identify and collect available data for modelling processes in the system. Generally, the requirement of data for the modelling process includes data of the following categories:

- Basic spatial data such as rain gauge, flow gauge and weather station locations, a digital elevation model (DEM), river network data, river cross sectional profiles and administration boundaries.
- Rainfall-runoff data including time series data on rainfall and runoff for historic events.
- Flood inundation data on extent and water depths of flood inundation of historical flood events. This data is used for validation of the hydraulic model.
- Socio-economic conditions data on socio-economic conditions for the province, relating to impacts of flood events in history.

Main data sources are provided by the government agencies such as Provincial Centre for Hydro-meteorological Forecasting, Regional Centre for Hydro-meteorological Forecasting and Provincial Committee for Flood and Cyclone Prevention, the Department of Natural Resource and Environment, Department of Science and Technology, Department of Agriculture and Rural Development. Further information on data used in this study will be discussed in more detail in the system component chapters of this thesis.

4.2. Objective 2. Develop a toolset for rainfall-runoff modelling

This component is the basis for prediction of the runoff. Outputs from this component are used for real-time simulation of flood inundation and for planning scenario analysis. This objective includes the following sub-objectives:

- Identify the rainfall-runoff model for the system. A review of existing rainfall-runoff models was undertaken with a focus on the models that have been used by local government agents, researchers and projects within the province and elsewhere. This step resulted in an identified model which performs effectively for Quang Nam basin.
- Develop the rainfall-runoff toolset. From the selected model, a toolset for rainfall-runoff modelling was then developed and integrated into the SDSS as ArcGIS tools using the VisualBasic.net programming language.
- Calibrate the rainfall-runoff model. This included calibrating the models to establish a set of applicable parameters for Quang Nam basin from rainfall and runoff data.
- Validate the model. To make sure the applicability of the toolset and the calibrated model, validation of the model was carried out against historical data.

These steps were undertaken to assure that the selected model will be feasible to apply to the area in terms of personnel capacity, data availability, applicability and accuracy. In terms of time efficiency, it is expected that this system will be used by decision makers, planners and managers for planning purposes, long-term and short-term decision making, and in real-time situations such as flood action plans. In terms of accuracy, the model was calibrated and tested against real data. A goodness of fit test was used to assess the performance of the model. The calibration and testing were done as comprehensively as possible assuring that the models can give accurate forecasting values. It is also an argument that the decision support system, generally, and the rainfall-runoff models specifically, have to be built on the basis of the current resource capacity of the agencies within the basin. This means that the system has to run on the data collected from current existing meteorological monitoring systems of Quang Nam province with the current conditions of resource investment and personnel capacity. All of these must be done such that the complexity of the models and software operation should be within the capacity of staff of the province, assuming that the staff will receive sufficient training. The integrated component of rainfall-runoff modelling was developed as a toolkit for forecasting runoff of the basin. The outputs from this step can be used as the inputs of the hydraulic modelling process.

4.3. Objective 3. Develop a toolset for simulating the flood inundation

The implementation of hydraulic modelling in this project is to identify the inundation area and water depth at different locations in the flood plain. This is one of the fundamental components for modelling the impacts of floods on the economic aspects of the hazard. Similar to the rainfall-runoff modelling process, this component was developed on the basis of a review of the existing models with a focus on the applicable and feasible models for the area. The following sub objectives were identified:

- Review hydraulic models: A review of hydraulic models for modelling the inundation of floodplain was undertaken by examining the models that have been used in the basin by government agencies, researchers and projects. Reports, research papers relating to these models were collected via the Provincial Centre for Hydro-meteorological Forecasting, the Regional Centre for Hydro-meteorological Forecasting and Provincial Committee for Flood and Cyclone Prevention. A number of criteria were considered including data requirements, accuracy and the ease of implementation.

- Develop the toolset for the SDSS: The toolset was developed as integrated tools in ArcGIS using VisualBasic.net programming language.
- Calibrate the model: The calibration of the model was carried out by using a newly adapted technique, Simulated Annealing, to generate a set of Manning's roughness coefficients that is applicable to the river channel network in Quang Nam basin.
- Validate the model: To ensure the applicability of the toolset, the validation of the model was carried out against historical datasets.

4.4. Objective 4. Develop a toolset for frequency analysis

This component includes a set of tools that allow for frequency analysis of flood events of different levels of severity. Together with tools in the hydraulic component, these tools will generate the inputs for the creation of maps of flood inundation of different probability levels. The scope of work under this module includes:

- Review the frequency analysis models and identify the applicable models for this type of analysis in Quang Nam basin.
- The toolset was developed as integrated tools in ArcGIS using VisualBasic.net programming language.
- Fit and validate the models.

In summary, it is expected that this flood management SDSS will provide a comprehensive tool set for analysis of the flood nature in Quang Nam. The system is implemented in the context of complex aspects of hydrological features of the province, such as the relationships between the river systems, rainfall patterns, and hydrologic conditions. The concrete result of the study is a SDSS that is based on the information derived from the above mentioned context. This should be a state of the art implementation of modern science and technology in quantifying the aspects of flood hazards, including the mapping of flood areas in multi-scale temporal resolution. This is also a reflection of best practice approaches which are a combination of local knowledge and modern technology: a GIS-based flood simulation and analysis system to support decision makers to gain quick information in flooding risk management through interactive spatial tools. Lessons learnt from this project will form the general guidelines for establishing of SDSS for flood hazard management in a developing country context.

Chapter 5. RAINFALL AND RUNOFF MODELLING

5.1. Introduction

One of the core components of flood risk management decision support system is the simulation of rainfall and runoff in flood prone areas. Accurate runoff modelling is becoming increasingly important for reliable decision support in the context of water resource management (Zang *et al.*, 2009). Most of the studies on these problems adopt one of the two main broad techniques, which are either hydrologic or hydraulic techniques (Smith, 1994) or a coupling of the two (Francisco *et al.*, 1998).

According to Abbott and Refsgaard (1996) hydrological models, including rainfall and runoff models, can be classified into deterministic and stochastic types. In deterministic models, a single set of input values and a single parameter set are used to generate a single set of outputs. In stochastic models, parameters are used to represent statistical distributions. Model parameters are normally generated by fitting measured data into a certain distribution. The output sets of a stochastic model are normally ranges of values, each derived from different combinations of the inputs and parameters. Each range associated with a probability of statistical certainty.

Singh (1995), on the basis of how physically processes are represented in models, classified the models into 1) empirical, regression or "black-box" models. 2) conceptual-empirical models and 3) physical based or process based models.

The empirical, regression or "black-box" models simply calibrate the relationship between rainfall and the output runoff. They do not attempt to represent the basic processes that really describe the mechanism of the relationship.

In conceptual-empirical models, the basic processes in rainfall runoff modelling such as interception, infiltration, evaporation, surface and subsurface runoff are described to some extent. However, the description of these processes are essentially calibrated input-output relationships based on the collected data, formulated to mimic the functional behaviour of the process in question.

Physically based or process based models depend on the fundamental physics and governing equations that describe the real physical processes to model the relationship between rainfall and runoff e.g. the process of water flow over and through soil and vegetation. They are intended to minimise the need for calibration by using relationships in which the parameters are, in principle, measurable physical quantities.

The selection of the model for use depends on a range of conditions such as the type of problem, required levels of accuracy, time constraints, data availability, usefulness of the model in terms of communication to users and the ease of use. All of the considerations mentioned above are useful to consider when answering the following two questions: "What is the appropriate model structure for a given type of hydrological system and a particular modelling task?" and "What is the appropriate parameter set within this structure to characterize the unique response features of a particular catchment?" (Wagener *et al.*, 2004). It is also noted that firstly, simple models (in terms of number of parameters) could give as good a performance as complex models for many purposes, and secondly, many models have been developed, but only a limited number of them are used in reality (Wagener *et al.*, 2004).

In this study, a number of popular methods for modelling rainfall runoff were considered. These methods have been assessed to see if they are suitable as tools for a decision support system in the Quang Nam situation. The considered models were the tank model, the unit hydrograph model (unit hydrograph with Clark's technique in GIS (Usul and Yilma, 2004)) and a multiple regression model. As far as the tank model is concerned, it had too many parameters that needed to be calibrated. The tool for this model was developed and tested. However, with little or no local knowledge of the interception, infiltration, evaporation, surface and subsurface runoff characteristics of Quang Nam basin, the calibration of the model became too tricky. An automatic calibration procedure was also developed and tested however it does not give reasonable results. The situation was very much similar for the preliminary research into employing unit hydrograph method. This method depends on the knowledge about the flow velocities of fluid over different type of landuses (Usul and Yilma, 2004). This makes the calibration process tedious as a trial and error method was used to assign different values of Manning's coefficients to landuse types has to be made. Although the model does not have as many parameters as the tank model, it does require some technical skills for calibration. With this in mind, after some preliminary study steps, this method was not considered for further development.

In the context of Quang Nam basin, with the availability of rainfall and runoff time series data, the current hydrological monitoring system and the current capacity of the staff resource of the province, a multiple regression model is proposed as it is a simple model which is easy to implement in the decision support system. More importantly, the validation of the model against historical data showed that multiple regression gave good results, as will be discussed below.

5.2.. Regression model

The following discussion about multiple regression is an adapted from Haan (1997). Regression models include two parts, namely the predicted portion and residual portion. The predicted portion has the characteristic that can be attributed to all the observations which is considered as a group within a parametric framework. The residual portion is the difference between the observed and predicted values. The general form of a regression model is as follows:

$$y_i = f(x_i, \beta) + e_i, i = 1,2, \dots n \qquad (5.1)$$

Where, n is the number of observations, y_i is the ith observation, $x_i = (x_{1i}, x_{2i}, \dots, x_{ki})$ is the predictor variable vector relating to y_i, $\beta = (\beta_0, \beta_1, \dots, \beta_p)$ is the parameter vector and e_i is the error associated with ith observation.

In the matrix form, the above model can be written as:

$$Y = X\beta + \varepsilon \tag{5.2}$$

The estimation of the parameter vector β using least square method is derived as:

$$\hat{\beta} = (X^T X)^{-1} X^T Y \tag{5.3}$$

The predicted model is:

$$\hat{Y} = X\hat{\beta} \tag{5.4}$$

so that the residual is given as:

$$e = Y - \hat{Y} \tag{5.5}$$

5.3. ANOVA for multiple regression

Multiple regression attempts to fit a regression equation for the response variable using several explanatory variables. Analysis of variance (ANOVA) provides information about levels of variability within a regression model and form the basis for tests of significance. The basis concept of a regression line is: data = fit + residual. This can be formally written as:

$$\sum_{i=1}^{n} (y_i - \bar{y})^2 = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 + \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{5.6}$$

This may be written as SST = SSM + SSE. Where SST is the total sum of squares, SSM is model sum of squares and SSE is error sum of square.

5.3.1. Square of multiple correlation coefficient R^2

R^2 is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression. The definition of R^2 is fairly straight-forward; it is the percentage of the response variable variation that is explained by a linear model:

$$R^2 = SSM/SST \tag{5.7}$$

Which is equal to the ratio of the model sum of squares to the total sum of squares. This formalize the interpretation of R^2 as explaining the fraction of variability in the data explained by the regression model. R^2 is a statistical measure of how close the data are to the fitted regression line. R^2 is always between 0 and 100%: 0% indicates that the model explains none of the variability of the response data around its mean. 100% indicates that the model explains all the variability of the response data around its mean. In general, the higher the R^2, the better the model fits your data. The square root of R^2 is called the multiple correlation coefficient and represents the correlation between the observations and the fitted values.

5.3.2. The sample variance

$$s_y^2 = \frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{n-1} = \frac{SST}{DFT} \tag{5.8}$$

DFT: total degrees of freedom.

5.3.3. Mean square model MSM

$$MSM = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{DFM} = \frac{SSM}{DFM} \tag{5.9}$$

DFM is the degree of freedoms of the model.

5.3.4. Mean square error MSE

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-2} = \frac{SSE}{DFE} \tag{5.10}$$

The estimate of the variance about the population regression line.

For p explanatory variables, the model degrees of freedom (DFM) are equal to p, error degrees of freedom (DFE) are equal to (n-p-1), and the total degrees of freedom (DFT) are equal to (n-1), the sum of DFM and DFE. The typical Anova table is as follows:

Table 5. 1 Anova table

Source	Degrees of freedom	Sum of squares	Mean square	F
Model	p	$\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$	SSM/DFM	MSM/MSE
Error	n-p-1	$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$	SSE/DFE	
Total	n-1	$\sum_{i=1}^{n}(y_i - \bar{y})^2$	SST/DFT	

The test statistic MSM/MSE has a F(p, n-p-1) distribution. The null hypothesis and alternative hypothesis:

$H0: \beta_1 = \beta_2 = \beta_3 \dots = \beta_n = 0;$

$H1: at\ least\ one\ of\ the\ parameter\ \beta_j \neq 0, j = 1, 2, 3 \dots p.$

Larger value of the test statistic provide evidence against the null hypothesis.

5.4. Model validation

Nash-Sutcliffe efficiency (NSE) is a normalized statistic that determines the relative magnitude of the residual variance ("noise") compared to the measured data variance ("information") (Nash and Sutcliffe, 1970). NSE indicates how well the plot of observed versus simulated data fits the 1:1 line. NSE is computed as shown in equation:

$$NSE = 1 - \left(\frac{\sum_{i=1}^{n}(Y_i^{obs} - Y_i^{cal})^2}{\sum_{i=1}^{n}(Y_i^{obs} - Y^{mean})^2} \right) \qquad (5.11)$$

where *obs* is observation and *cal* is calculated, Y_{mean} is the mean of observed data for the constituent being evaluated, and n is the total number of observations.

NSE ranges between −∞ and 1.0 (1 inclusive), with NSE = 1 being the optimal value. Values between 0.0 and 1.0 are generally viewed as acceptable levels of performance, whereas values smaller than 0.0 indicate that the mean observed value is a better predictor than the simulated value, which indicates unacceptable performance (Krause *et al*., 2005). NSE is very commonly used, and provides extensive information on reported values. Sevat and Dezetter (1991) found NSE to be the best objective function for reflecting the overall fit of a hydrograph.

Moriasi *et al*. (2007) proposed that NSE values should exceed 0.5 in order for models to be judged as satisfactory for hydrologic simulation performed on a monthly time step and appropriate relaxing and tightening of the standard to be performed for daily and annual time step evaluation, respectively. The poorest results generally occurred for daily predictions, although this was not universal (Grizzetti *et al*., 2005).

5.5. Toolset development

5.5.1. Rainfall-runoff database

The data used in this rainfall and runoff study were obtained from the Central Area Station for Meteorology and Hydrology of Vietnam. The data was then organized in an Access (2007) database. The data input to this database included two type of data: rainfall (in mm) and discharge (in m^3/sec) which were structured into two tables "Rain" and "Q" with similar schema: Rain{Time, Station, Rain} and Q{Time, Station, Q}. These data were then available to the rainfall-runoff toolset via queries that were designed specifically for two discharge stations in Nong Son and Thanh My. More detailed description of data and data structures used in this study will be provided in section VI.

5.5.2. Regression and validation tool development

In this study, a tool set for modeling the relationship between rainfall and runoff was developed using VisualBasic.net and implemented as an ArcGIS tool. The theory on multiple regression behind this toolset is provided in section II above. Detailed implementation of this tool using VisualBasic.net is provided in appendix 5.1. Algebraic/matrix operations in the program are undertaken by using an open source dynamic library GeneralMatrix.dll (Paul, 2004). The graph control used to display data in this tool is ZedgraphControl (JChampion, 2007), a free open source graph control.

The tool has a data loading interface, a regression interface, a graph interface and a validation interface. The data loading interface allows for loading rainfall-runoff data from the Access database. To load data into the tool, information about the event needs to be specified. This information includes start and end time of the event, and the stations involved (one or more rain gauge stations and one discharge station). Data for one or more events can be loaded.

After loading the data, users can run the regression to obtain the model information. Figure 5.1 shows the model building interface. The graphical description of the model output is also given by the tool, as shown in the Figure 5.2.

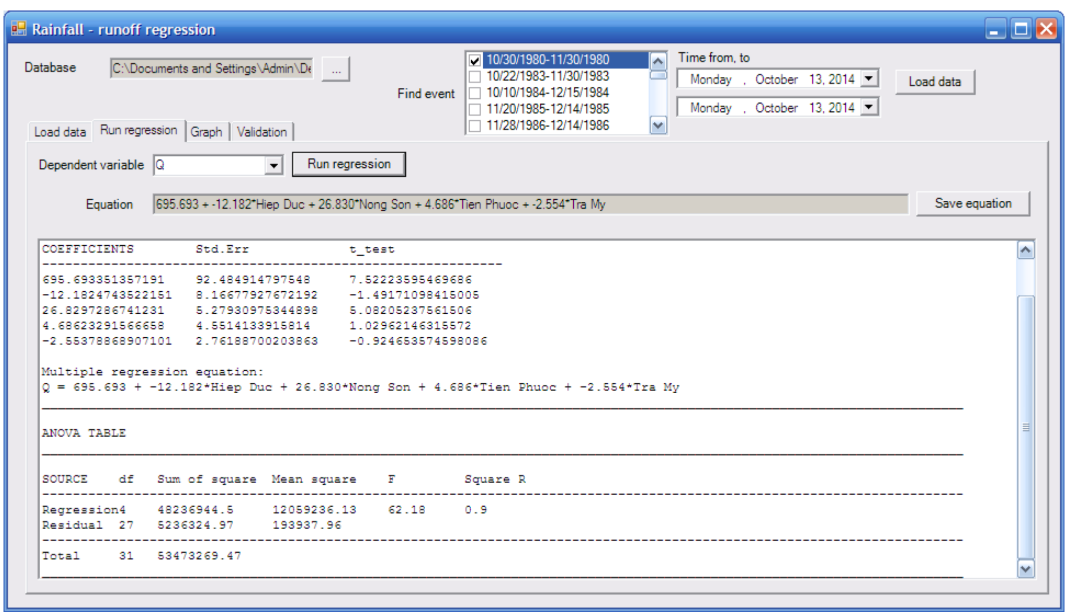


Figure 5.1 Multiple regression fitting interface example

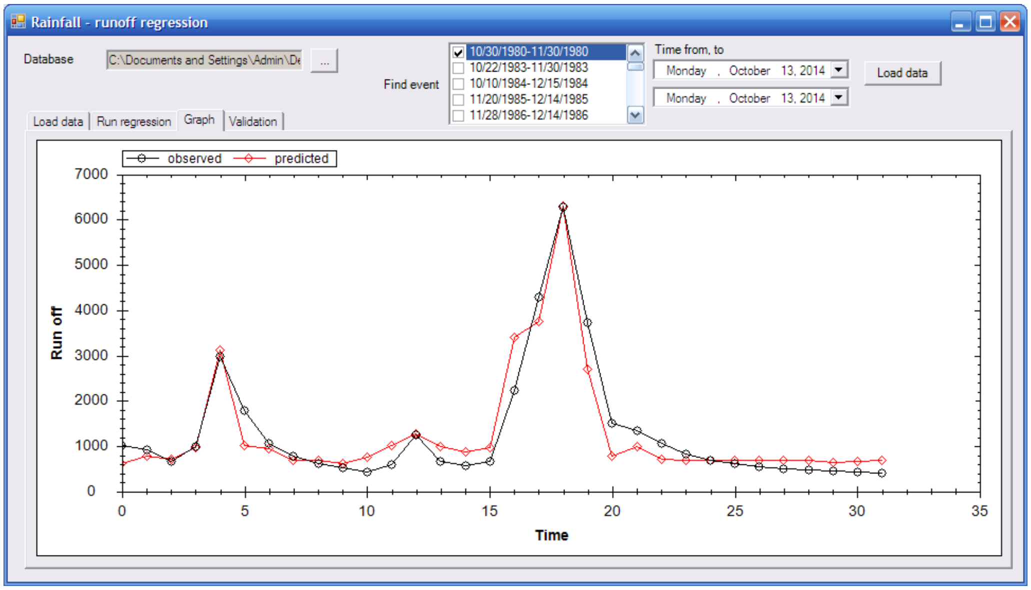


Figure 5.2 Multiple regression graph interface example

The validation function of the tool is shown in Figure 5.3. The validation function allows for the validation of the data from an observed event against a fitted model. The fitted models had to be pre-built and saved as regression equations in the database in the model fitting step. The validation criteria used in this tool is the Nash-Sutcliffe efficient efficiency (NSE), see M***odel validation*** section above.

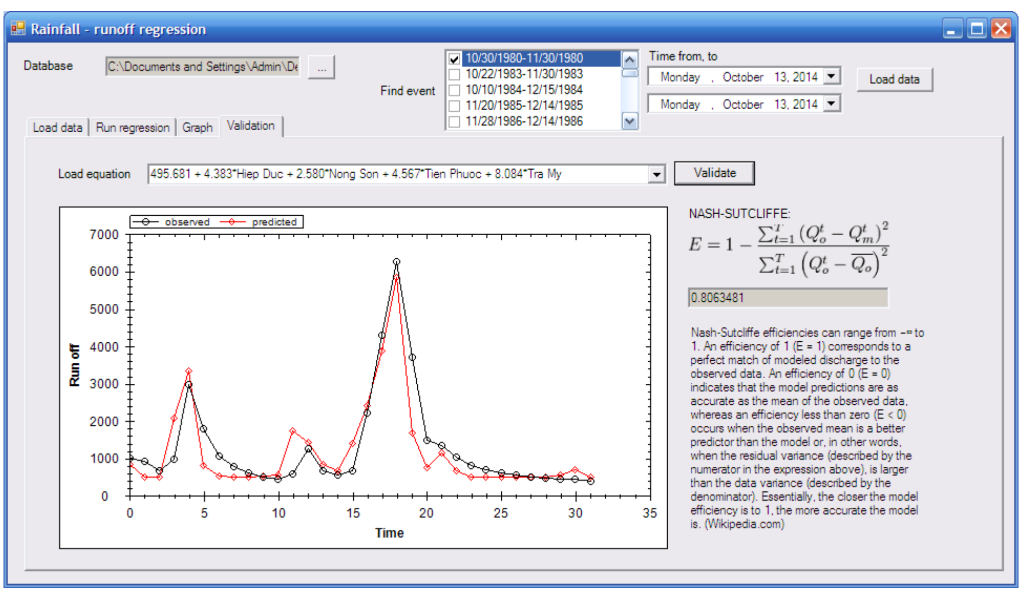


Figure 5.3 Multiple regression validation interface example

5.6. Building rainfall runoff models for Quang Nam basin

5.6.1. Rain gauge, discharge stations in sub-basins and data structures

For a detailed description of the study area please refer to chapter 2 of this thesis. The brief information about the study area in this section is only relevant for the following discussion about modeling rainfall and runoff within the basin using the regression method.

The Vu Gia - Thu Bon river system originates on the eastern side of the Truong Son mountain range. The system is formed by two main rivers: The Vu Gia and the Thu Bon. The Vu Gia has many tributaries. The Thu Bon originates at the borders of the three provinces of Quang Nam, Kon Tum and Quang Ngai at an elevation of more than 2,000 m. It enters the sea through the Dai estuary and the river mouth in Da Nang. Figure 5.4 shows Vu Gia – Thu Bon river system with 15 gauge stations. Of these 15 stations, only two, Nong Son and Thanh My, have discharge measures.

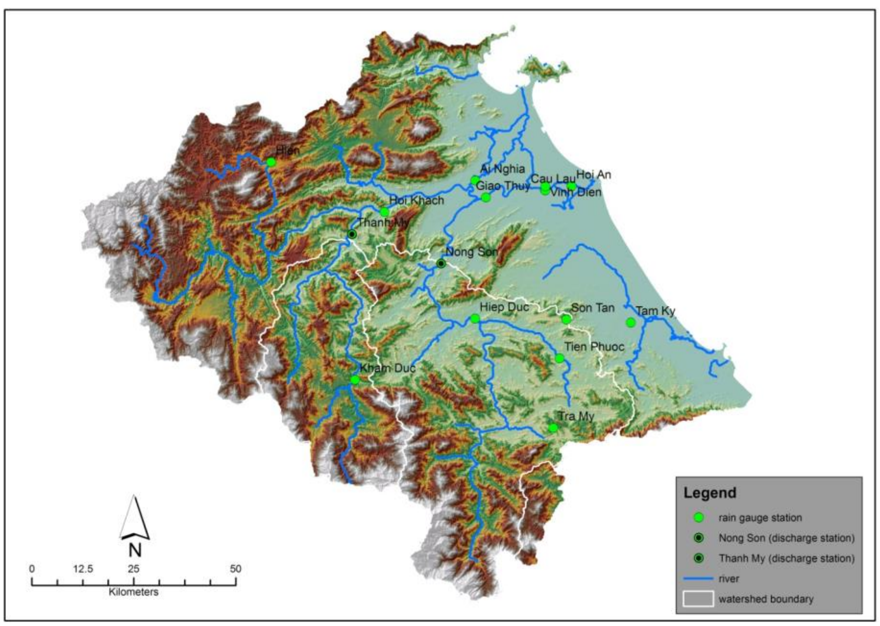


Figure 5.4 Quang Nam basin with the structure of the rain and flow gauge network

The rain gauge stations that can be used to model the discharge at each station in Nong Son and Thanh My have to be within the sub-basin that collects the runoff for each of the two above mentioned stations. A GIS was used to delineate the sub-basin boundaries of the catchments that collect water for the river system. The process starts with repairing the Quang Nam DEM by filling holes and missing data points in the raster so that these defections cannot obstruct the water flow on the surface. The next steps of the process calculate the flow direction map and flow accumulation map from the DEM. The flow direction map is a raster where each pixel in that raster contains an integer value indicating the flow direction of water flow out of that pixel and is the input for generating the flow accumulation map. In the flow accumulation map, each pixel contains a value indicating the area upstream of that pixel that drains water into it. The final step in the sub-basin delineation process is using the flow direction, flow accumulation maps and the locations of Nong Son and Thanh My stations to trace the pixel within the DEM of Quang Nam that contribute water to each of the two discharge stations in Nong Son and Thanh My. The collections of these pixels form the areas of the two sub-basins. All the above processing operations are undertaken in ArcGIS. Figure 5.5 illustrates the process of delineating the two sub-basin boundaries.

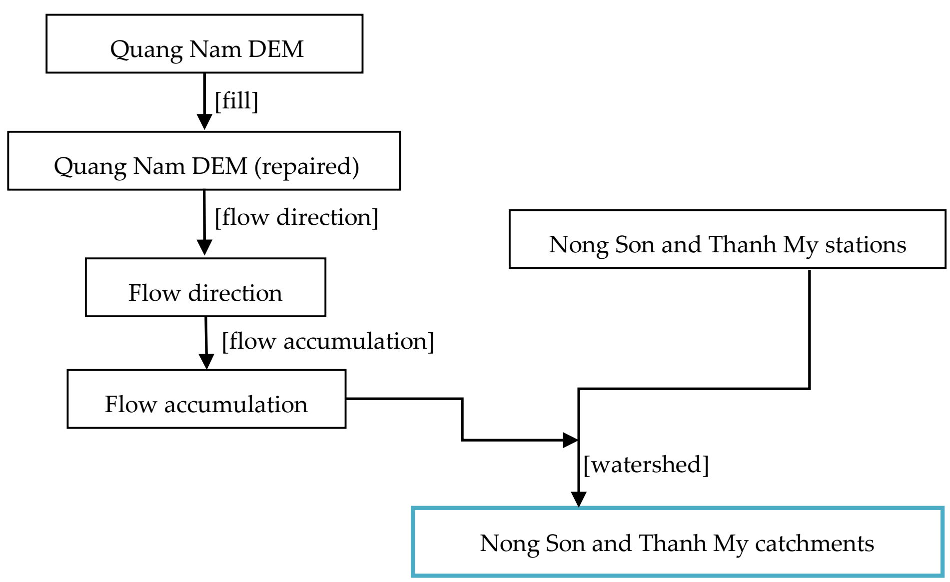


Figure 5.5 Procedure for delineating Nong Son and Thanh My catchment boundaries

Figure 5.4 shows the delineated boundaries of the watersheds that drain water into the two locations. The result shows that rain gauge stations in Hiep Duc, Nong Son, Tien Phuoc and Tra My are within Nong Son watershed and Kham Duc (Phuoc Son) rain gauge station is within the watershed that collects water to Thanh My rain-discharge station.

The input data included two types: six-hourly rainfall (in mm) and discharge (in m^3/sec). The data were organized into two tables: “Rain” and “Q” with similar schema: Rain{Time, Station, Rain} and Q{Time, station, Q} in an Access relational database. Discharge data were only available at Nong Son and Thanh My stations. As shown by the GIS processing above, rain gauge stations Tra My, Tien Phuoc, and Hiep Duc are upstream of Nong Son rain-discharge station and Kham Duc (Phuoc Son) rain gauge station is upstream of Thanh My rain-discharge station (See Figure 5.4). To prepare the data for building and validating the rainfall runoff models, database queries were designed to reflect this up-downstream structure of the gauge system. It relates rainfall from Hiep Duc, Nong Son, Tien Phuoc and Tra My to Nong Son discharge. Similarly, a query was also designed for relating rainfall from Kham Duc (Phuoc Son) and Thanh My to Thanh My discharge.

5.6.2. Model specification

A multiple linear regression model was used to estimate the discharge at Nong Son and Thanh My stations. For both Nong Son and Thanh My sub-basins, the following flood data series was used for building and validating the models:

Table 5.2 Data series for rainfall runoff regression fitting and validation for Quang Nam basin

Series id	From date	To date	Model fitting	Model validating
1	30-Oct-80	30-Nov-80		X
2	22-Oct-83	30-Nov-83	X	
3	10-Oct-84	15-Dec-84	X	
4	20-Nov-85	14-Dec-85	X	
5	28-Nov-86	14-Dec-86	X	
6	05-Oct-88	12-Nov-88	X	
7	17-May-89	30-May-89	X	

8	10-Oct-90	25-Nov-90	X	
9	10-Oct-91	25-Dec-91	X	
10	14-Oct-92	25-Nov-92		X
11	04-Oct-93	05-Nov-93	X	
12	17-Oct-96	09-Dec-96	X	
13	09-Sep-97	20-Nov-97	X	
14	09-Oct-98	20-Dec-98	X	
15	14-Oct-99	20-Dec-99		X
16	30-Sep-00	30-Dec-00	X	
17	15-Oct-01	25-Dec-01	X	
18	07-Oct-03	03-Dec-03	X	
19	25-Sep-04	25-Dec-04		X
20	05-Oct-05	03-Dec-05		X
21	27-Oct-07	10-Dec-07	X	

The dataset was divided into two sets that were used for model building and validating. The majority of the dataset (16 data series for 16 years) was used for fitting the regression model and events for five years were used for validating the model (see Table 5.2).

5.6.3. Discharge estimation at Nong Son station

With the above model specifications and data, the regression model for estimating the discharge at Nong Son station by using the rain data from gauges Hiep Duc, Nong Son, Tien Phuoc, Tra My was fitted as:

$$Q = 533.036 + 9.537 * \text{Hiep Duc} + 0.382 * \text{Nong Son} + 0.255 * \text{Tien Phuoc} + 8.077 * \text{Tra My} \quad (5.12)$$

The model statistics showed a good model fit between observed data and model prediction. A significant level of variability in the predicted values is explained by the observed data values ($R^2 = 0.58$). Figure 5.6 and 5.7 show the information of the regression model for estimation of discharge at Nong Son station.

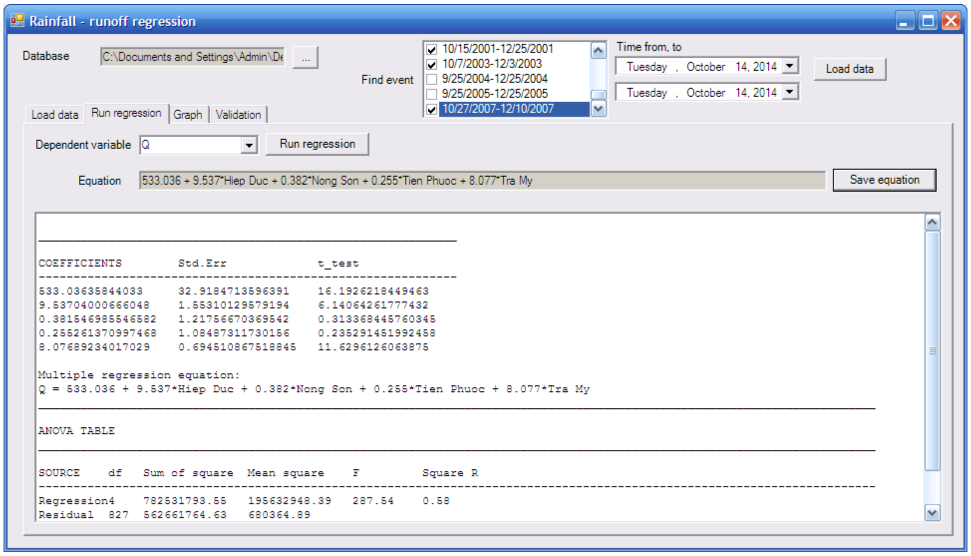


Figure 5.6 Rainfall runoff regression model information for Nong Son station

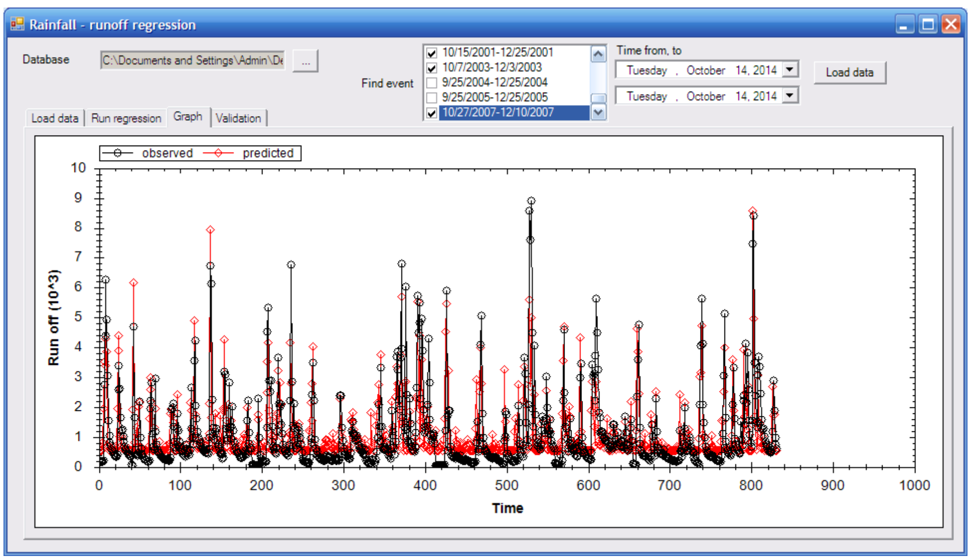


Figure 5.7 Graphical display of the runoff regression result for Nong Son station

5.6.4. Validation for Nong Son discharge estimation model

Using the above model to validate against the five data series of events in 1980, 1992, 1999, 2004, 2005 gave good agreement (see Table 5.3 and Figs 5.8 to 5.12). Nash-Sutcliff efficient (NSE) for these validations are given in the following table:

Table 5.3 Nong Son station discharge validation result using Nash-Sutcliffe efficient

Series id	From date	To date	NSE
1	30-Oct-80	30-Nov-80	0.749
10	14-Oct-92	25-Nov-92	0.590
15	14-Oct-99	20-Dec-99	0.718
19	25-Sep-04	25-Dec-04	0.594
20	05-Oct-05	03-Dec-05	0.613

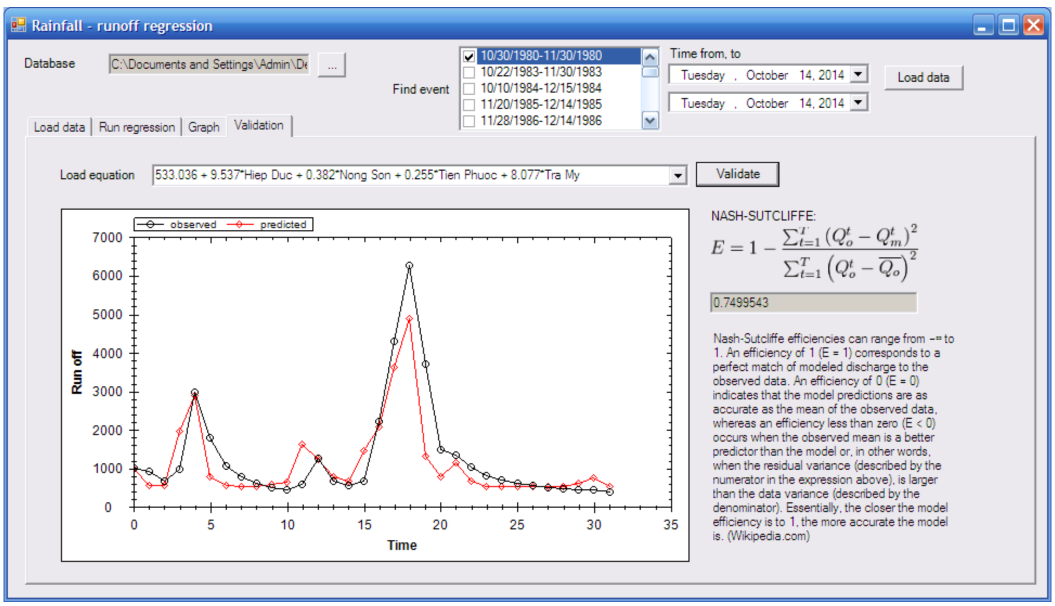


Figure 5.8 Rainfall runoff validation information for flood events in 1980 for Nong Son station

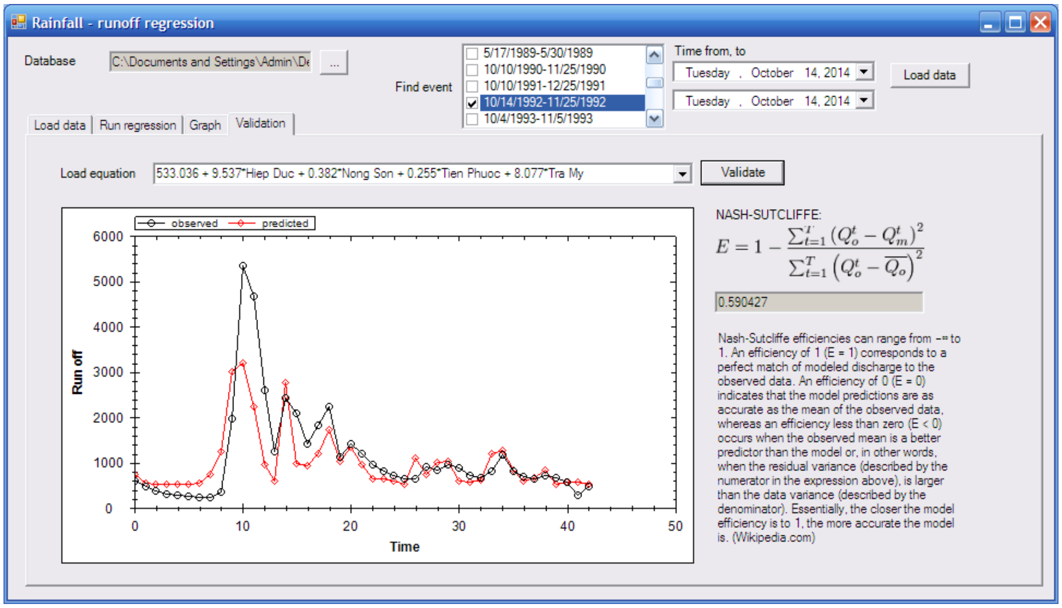


Figure 5.9 Rainfall runoff validation information for flood events in 1992 for Nong Son station

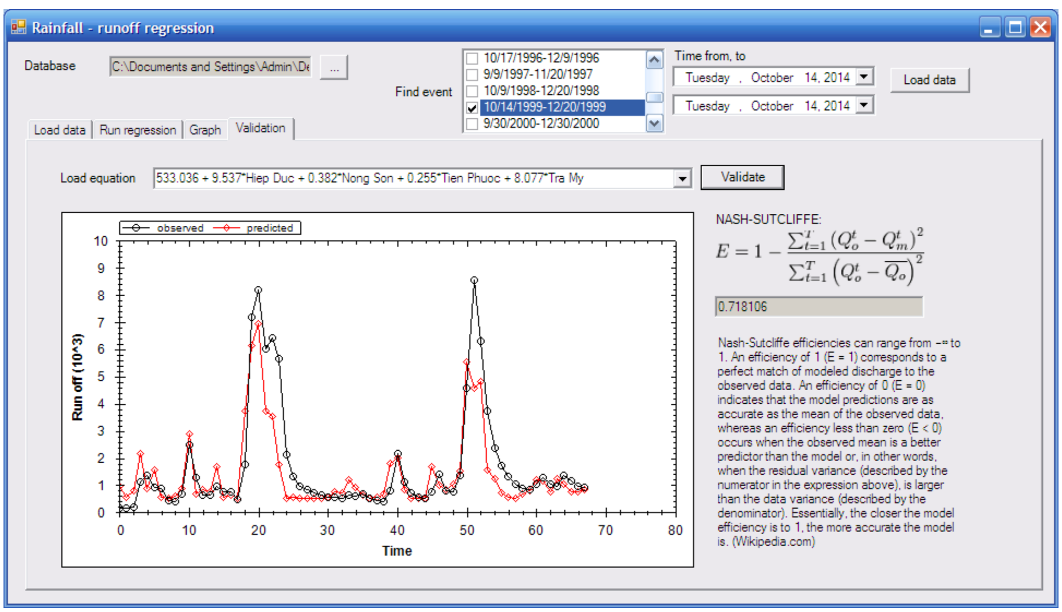


Figure 5.10 Rainfall runoff validation information for flood events in 1999 for Nong Son station

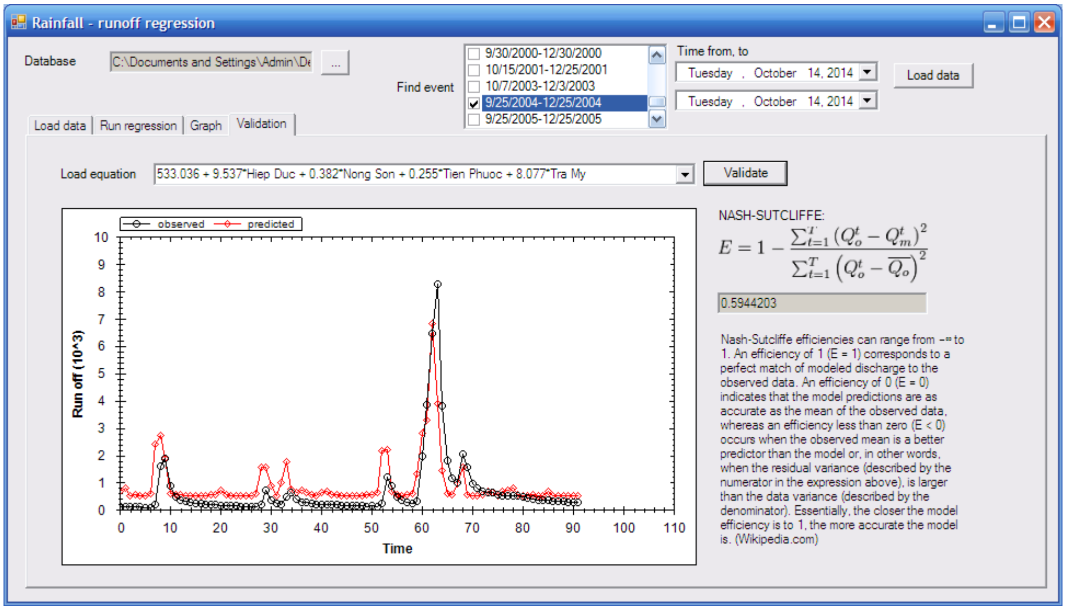


Figure 5.11 Rainfall runoff validation information for flood events in 2004 for Nong Son station

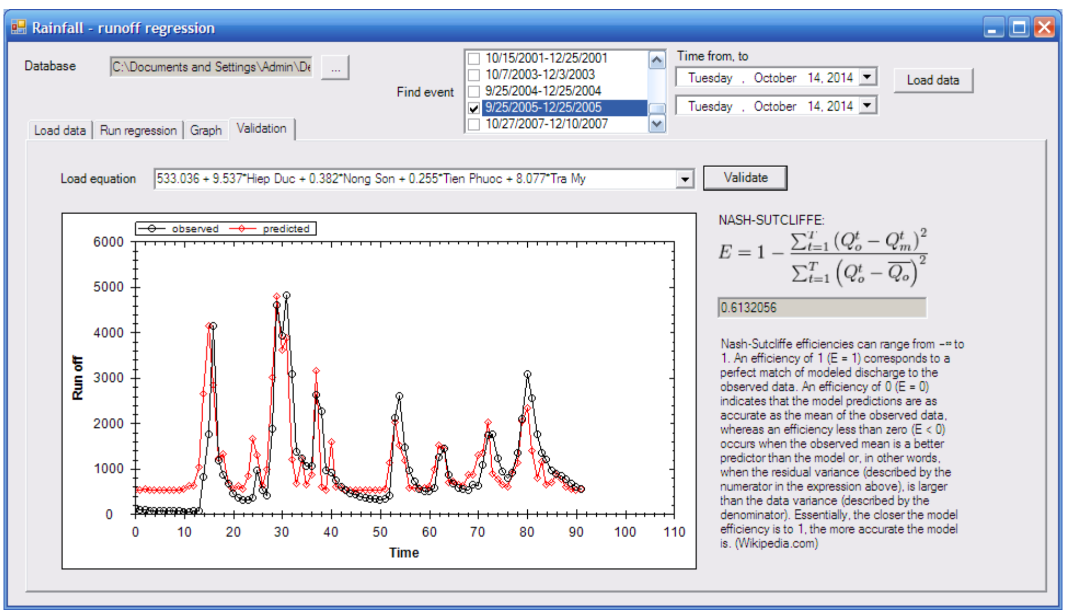


Figure 5.12 Rainfall runoff validation information for flood events in 2005 for Nong Son station

5.6.5. Discharge estimation modeling at Thanh My station

For estimating discharge at Thanh My station, rain data series at Phuoc Son and Thanh My and discharge at Thanh My station were used (see Figure 5.4). The regression model for Thanh My was also structured in the same manner as that of Nong Son above. Similar to the case of Nong Son, the dataset was divided into two parts for model building and validating. The majority of the dataset (16 data series for 16 years) was used for setting up the regression model and events for five years were used for validating the model (see Table 5.2).

With the above model structure specifications, the regression model for estimating the discharge at Thanh My station by using the rain data from gauges Phuoc Son and Thanh My was fitted as:

$$Q\ (m^3.sec^{-1}) = 226.078\ +\ 5.357 * \text{Phuoc Son}\ +\ 3.588 * \text{Thanh My} \qquad 5.13$$

The model showed a good fit between data observed and model prediction. A significant level of variability in the predicted values is explained by the observed data values (R^2 = 0.60). Figure 5.13 and 4.14 show the information of the regression model for Thanh My station discharge estimation.

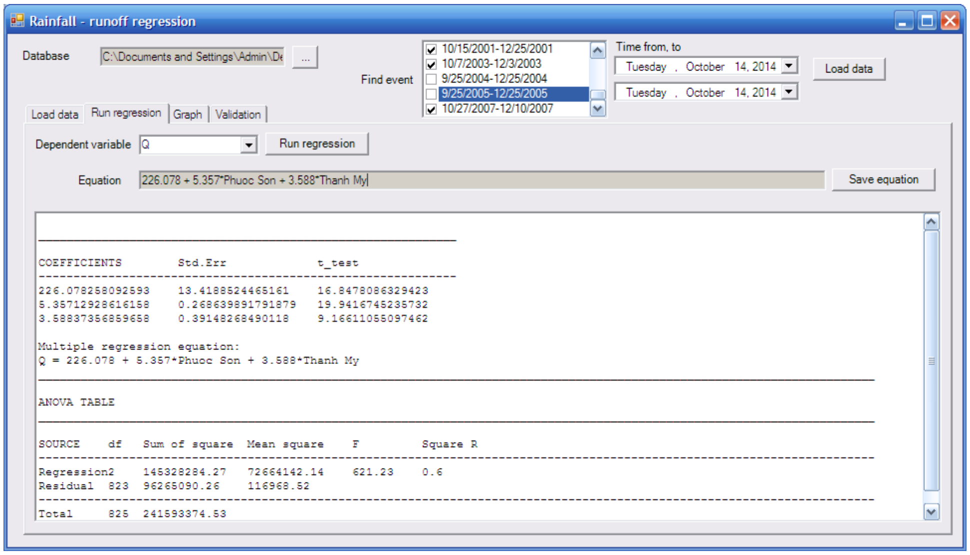


Figure 5.13 Rainfall runoff regression model information for Thanh My station

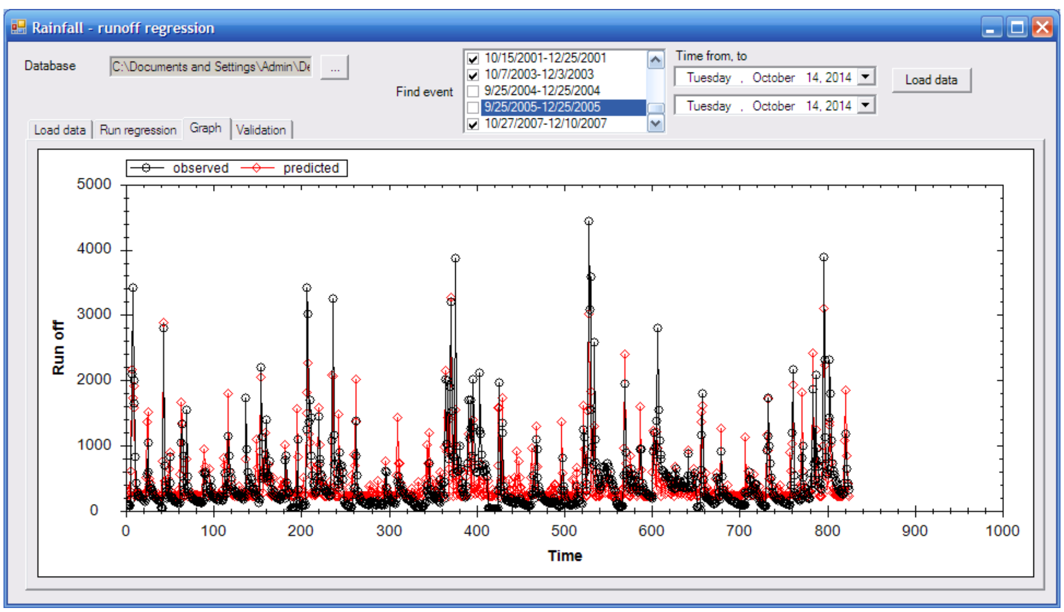


Figure 5.14 Graphical display of the runoff regression result for Thanh My station

5.6.6. Validation for Thanh My discharge estimation model

Using the above model to validate against the five data series of events in 1980, 1992, 1999, 2004, 2005 gave good agreement (see Table 5.4 and Figs. from 5.15 to 5.19). This shows that the model can be used to predict discharge with satisfaction. The Nash-Sutcliff efficiency (NSE) for these validations are given in the following table:

Table 5.4 Thanh My station discharge validation result using Nash-Sutcliff efficient

Series id	From date	To date	NSE
1	30-Oct-80	30-Nov-80	0.851
10	14-Oct-92	25-Nov-92	0.514
15	14-Oct-99	20-Dec-99	0.769
19	25-Sep-04	25-Dec-04	0.589
20	05-Oct-05	03-Dec-05	0.367

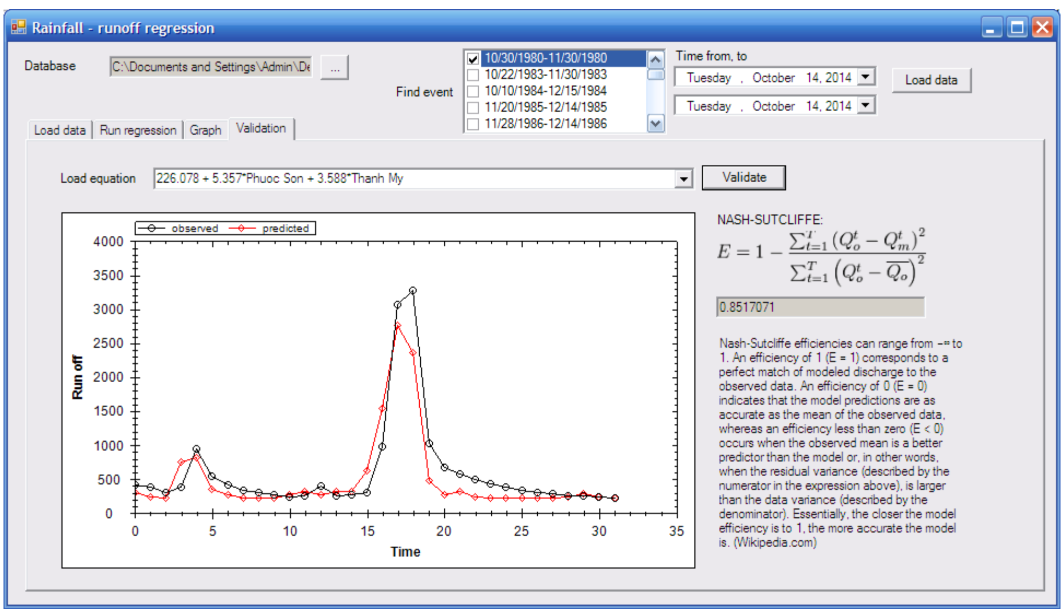


Figure 5.15 Rainfall runoff validation information for flood events in 1980 for Thanh My station

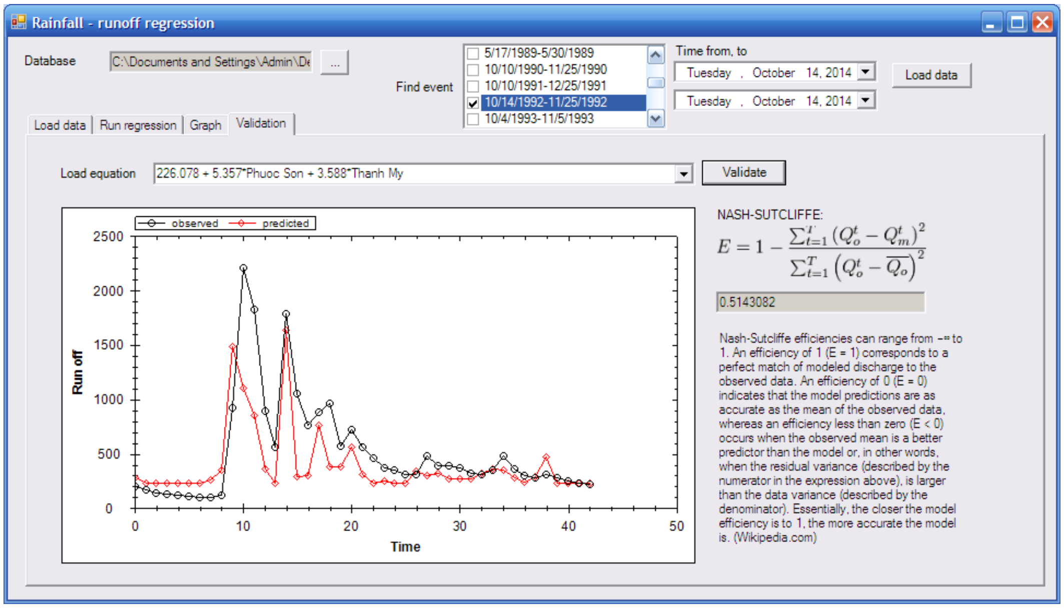


Figure 5.16 Rainfall runoff validation information for flood events in 1992 Thanh My station

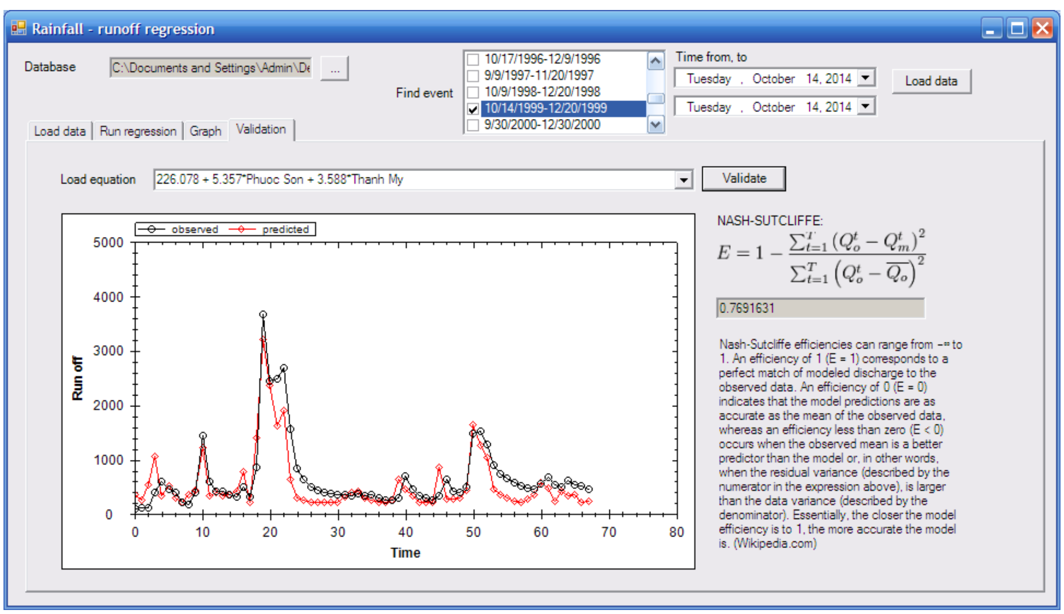


Figure 5.17 Rainfall runoff validation information for flood events in 1999 Thanh My

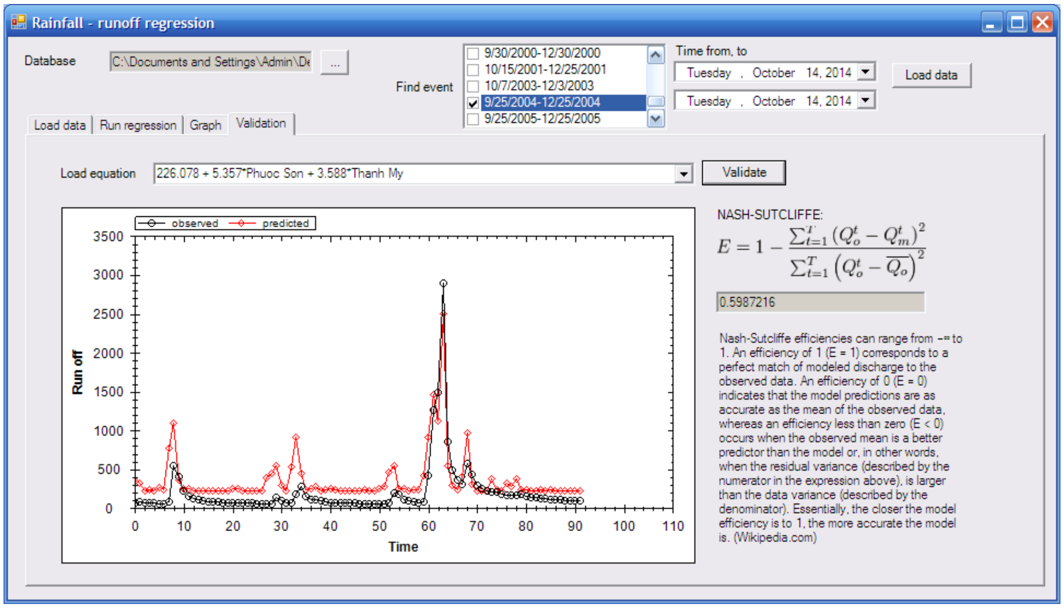


Figure 5.18 Rainfall runoff validation information for flood events in 2004 Thanh My

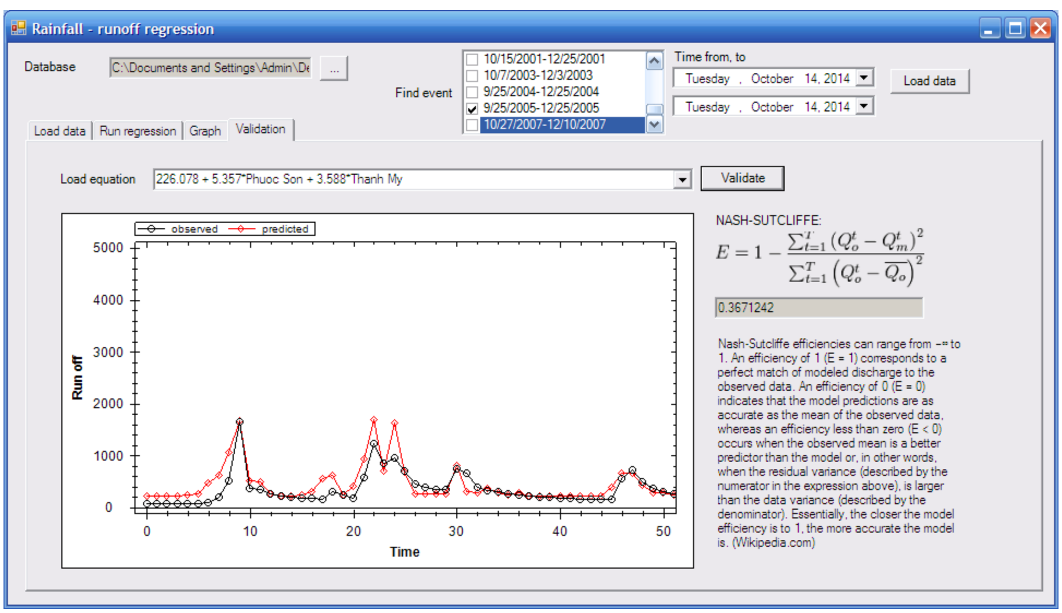


Figure 5.19 Rainfall runoff validation information for flood events in 2007 Thanh My station

5.7. Use case demonstration

With the forecasted information about the rainfall values, estimations of the runoff can be made by using the calibrated rainfall runoff models (equations 5.12 and 5.13). In this section, procedures for generating runoff estimation from forecasted rainfall are demonstrated.

In this synthetic case, the runoff values at two stations Nong Son and Thanh My will be generated from synthetic forecasted rainfall values at upstream rain gauge stations. Table 5.5 shows the synthetic forecasted rainfall values and corresponding estimated runoff values at Nong Son and Thanh My stations. These runoff values were calculated by using the calibrated regression models above (equations 5.12 and 5.13).

Table 5.5 Demonstration of runoff estimation at Nong Son and Thanh My stations

Station	Synthetic forecast rainfall ($mm.h^{-1}$)	Estimated runoff ($m^3.sec^{-1}$)
Hiep Duc	370	
Nong Son	280	6739
Tien Phuoc	500	
Tra My	300	
Phuoc Son	370	2437
Thanh My	64	

The estimated runoff values at Nong Son and Thanh My in table 5.4, in turn, could be used for generating an inundation map for the study area. These values of runoff are similar to those used as the boundary conditions for generating the flood inundation map of the event in 2008, so the inundation map generated by these synthetic inputs should be the one shown in figure 6.25. Chapter 6 will discuss in details about generating flood inundation maps.

This suggests that the information from rainfall forecast at the rain gauge stations within the basin could be used to estimate the runoff values at Nong Son and Thanh My. From these runoff values, an estimation of inundation could be made, and used as a reference or an input for flood hazard protection actions such as communicating visually with local people, preparation before a flood event, etc. However, the use cases are not limited to what discussed in this sections. The toolset can be used in many other situations as desired by planners, decision makers.

5.8. Summary

The regression tool was used to develop discharge estimation models for Nong Son and Thanh My stations. The models are as follows:

1. The regression model for estimating the discharge at Nong Son station by using the rain data from gauges Hiep Duc, Nong Son, Tien Phuong, Tra My:

$$\mathrm{Q}\ (\mathrm{m}^3.\mathrm{sec}^{-1}) \ = \ 533.036 \ + \ 9.537 * \mathrm{Hiep\ Duc} \ + \ 0.382 * \mathrm{Nong\ Son} \ + \ 0.255 * \mathrm{Tien\ Phuoc} \ + \ 8.077 * \mathrm{Tra\ My}$$

The model shows a good fit between data observed and model prediction. Significant level of variability in the predicted values are explained by the observed data values (R^2 = 0.58). Nash-Sutcliff efficiency criteria also validate that the model is useful for estimation of discharge at Nong Son. Most of the validation result are good (NSE from 0.594-0.749).

2. The regression model for estimating the discharge at Thanh My station by using the rain data from gauges Phuoc Son and Thanh My:

$$Q(\mathrm{m}^3.\mathrm{sec}^{-1}) = 226.078 \ + \ 5.357 * \mathrm{Phuoc\ Son} \ + \ 3.588 * \mathrm{Thanh\ My}$$

The model shows good fit between data observed and model prediction. Significant level of variability in the predicted values are explained by the observed data values (R^2 = 0.60). Nash-Sutcliff efficiency criteria also validate that the model is useful for estimation of discharge at Nong Son. Most of the validation result are good, NSE ranges within 0.589-0.851, except for the case of 2005 which has an NSE of 0.367.

Judging against NSE, ranging between $-\infty$ and 1.0. Values between 0.0 and 1.0 are generally viewed as acceptable levels of performance, whereas values smaller 0.0 indicate that the mean observed value is a better predictor than the simulated value, which indicates unacceptable performance (Krause *et al.,* 2005). Moriasi *et al.* (2007) proposed that NSE values should exceed 0.5 in order for models to be judged as satisfactory for hydrologic simulation performed on a ***monthly time step*** and appropriate relaxing and tightening of the standard to be performed for daily and annual time step evaluation, respectively. The implication then is that, the above to models can be used for estimating the discharge for Nong Son and Thanh My stations using upstream rain gauge data with satisfactory level of accuracy.

From the runoff estimation it can be seen that, by using the rainfall measure or forecast or design rainfall values at the rain gauge stations within each sub-basin, the runoff at Nong Son and Thanh My stations could be estimated. The estimated values of runoff at Nong Son and Thanh My could be useful in a number of situations. In real time forecasting situations, from the forecast values of rainfall at stations within Nong Son and Thanh My sub-basins, the runoff values at Nong Son and Thanh My could be estimated. From these values, by using a hydraulic model (Chapter 6) the estimated flood plain could be generated, which could be useful for real time strategies such as evacuation, resource allocation, development real time action plans, and other real time mitigation strategies. By using design values of rainfall, the corresponding design flood plains could also be generated (More discussion on this is given in chapter 6). This could be the useful information for planning purposes such as resource allocation planning, landuse planning, infrastructure development planning, industrial development, agricultural development planning, tourism development planning, urban development etc.

However, due to the fact that, the development of this tool depends on the analysis between the runoff at Nong Son and Thanh My with the rainfall at 6 upstream rain gauge stations within Quang Nam basin (Hiep Duc, Nong Son, Tien Phuoc, Tra My, Phuoc Son, Thanh My), it is not possible to be sure that this is applicable to other basins. The application of this tool set to other basins should be tested to be sure that there is linear relationship between rainfall values in the watershed with the runoff values at the pour points of the basin. Even for the case of Quang Nam basin, the study only focused on fitting regression models for large flood events, therefore, the regression equations above are only applicable for estimating runoffs for big flood events.

Chapter 6. DEVELOPING A TOOLSET FOR SIMULATING GRADUAL VARIED FLOW IN CHANNEL NETWORKS

6.1. Introduction

The problem of gradually varied flows in channel networks has been investigated in many studies such as Akan and Yen (1981), Chaudhry and Schulte (1986), Chaudhry (1993), Nguyen and Kawano (1995), Sen and Garg (2002). Most of the work focused on how to efficiently solve the system of equations resulting from the continuity, momentum or energy relationships between cross-sections of a channel network. Some of the methods focused on certain types of networks: looped, dendritic, parallel, etc. For example, Wylie (1972) proposed a method for calculating the flows around a group of islands by considering the problem as flow in a single channel where the total length of the flow is equal to the total length of the flows in the reaches. Chaudhry and Schulte (1986) proposed a finite difference method for calculating steady flow in parallel channels. Nguyen and Kawaco (1995) proposed a double sweep method coupled with a special node numbering scheme for simultaneously solving the gradually varied flow equations in dendritic channel networks. Reddy and Bhallamudi (2004) proposed an algorithm for calculating the flow depth and discharge on a cyclic network by classifying the computational procedures into initial and boundary value problems and identifying the linking path for solutions from individual channels. Most of the studies were carried out as hypothesis testing models of small scale networks with prismatic/ well behaved geometry of the flow cross-section (Islam *et al.* 2005, Sen and Garg 2002, Reddy and Bhallamudi 2004, Ghulam *et al.* 2012). This simplifies the computational effort, i.e. the partial derivative calculation of energy/momentum equations with respect to flow depths and discharges, and narrows the practical application of the results at the same time. Furthermore, most of the methods required complicated node numbering schemes. From an implementation perspective, these become barriers for computer model development. In addition to these, the solution of the gradually varied flow problems is not only about solving the system of equations efficiently but also about assembling the system of equations automatically. Using special node numbering schemes or manually setting up the system of equations make the applicability of the study results to real world problems challenging. An automatic assembling scheme for general channel networks would be ideal for the simultaneous solution of flow variables. This helps to open a door to the implementation of simultaneous solution algorithms in practical computer programs instead of confining them to research settings.

In this chapter, a comprehensive approach to the problem of steady state gradually varied flow is proposed and implemented as part of the wider aims of this research. Specifically, the following topics will be discussed:

- Governing equations for gradually varied flow and general solution approach.
- Graph theory application in automatically assembling gradually varied flow equations.
- Data models for coupling solution algorithm with a GIS platform. There are two types of data model that will be discussed in this sections. They are raster data for elevation and vector data for channel cross-sections.
- A numerical algorithm for solving gradually varied flow and software development (hydraulic modelling toolset).
- Application of Simulated Annealing (SA) in calibrating Manning's roughness coefficients for a general channel network of a basin scale.
- Application of the toolset to a case study in Quang Nam basin.

6.2. Governing equations

A slowly varying water depth or discharge with respect to time and over a considerable length of channel can be approximated to be steady gradually varied flow (Prasuhn, 1992; Subramanya, 1991). In many practical applications, the flow in open channels can be considered as steady and gradually varied. The solution of steady flow equations can arise from computing the flow distribution and water surface profile in a water channel network (Szymkiewicz, 2010). The following section briefly discusses the equations used for simultaneous solution to the system of equations; detailed derivations of the equations are given in Chaudhry (1993) and Nabi *et al.* (2012). The following terms are used in the discussion: a normal reach is a channel reach between one upstream cross-section and one downstream cross-section; junctions include split and combining junctions. A split junction is a junction where there is one upstream cross-section and more than one downstream cross-sections, while a combining junction is the inverse. An upstream end is where water enters into the channel network. A downstream end is the pour point where water exits the network. The following derivation of the governing equations for the gradual varied flow is adopted mainly from Chaudhry (2007) and Nabi *et al.* (2012).

6.2.1. Energy balance and continuity equations in a normal reach

The energy balance equation in a normal reach, between cross-sections i and j is:

$$y_i + z_i + \frac{\alpha_i Q_i^2}{2gA_i^2} = y_j + z_j + \frac{\alpha_j Q_j^2}{2gA_j^2} + 0.5.L\left(\frac{n_i^2|Q_i|Q_i}{A_i^2 R_i^{\frac{4}{3}}} + \frac{n_j^2|Q_j|Q_j}{A_j^2 R_j^{\frac{4}{3}}}\right) \tag{6.1}$$

Where, α is the energy correction coefficient at the cross-section:

$$\alpha_i = \frac{\sum \frac{K_m^3}{A_m^2} (\sum A_m)^2}{(\sum K_m)^3} \tag{6.2}$$

and

$$K_m = \frac{1}{n} A_m R_m^{2/3} \tag{6.3}$$

In (6.3) z is the channel bed elevation measured from the reference datum (m), y is the flow depth (m), A is the flow area at the cross-section (m^2), V is the flow velocity ($m.sec^{-1}$), Q is discharge ($m^3.sec^{-1}$) and g is gravitational acceleration ($m.sec^{-2}$). In 6.2 and 6.3, *m* denotes a sub-area of the cross-section and K is the conveyance factor. The continuity equation for a normal reach is:

$$Q_i = Q_j \tag{6.4}$$

6.2.2. Energy balance and continuity equation at a junction

For simplicity, in this study, cross-sections at junctions are set up close together so that the friction loss at junctions can be neglected (Szymkiewicz, 2010). The energy balance equation between the upstream node and one of the downstream nodes for the case of split flow or between one of the upstream nodes and the downstream node for the case of combined flow is then:

$$-y_i + y_j + -\frac{\alpha_i Q_i^2}{2gA_i^2} + \frac{\alpha_j Q_j^2}{2gA_j^2} = z_i - z_j \tag{6.5}$$

The continuity equation at a split/combined flow junction is:

$$\sum Q_i = \sum Q_o \tag{6.6}$$

Where i indicates inflow discharges and o indicates outflow discharges. The available equations at any junction are equal to the number of channels joining at that junction (Nabi *et al.*, 2012), no matter how many upstream and downstream channels.

6.2.3. The boundary conditions

For subcritical flow, the boundary conditions include the flow depth at the downstream end nodes and the discharges at the upstream end nodes of the channel network. For example, for a configuration with one upstream channel and one downstream channel, the boundary conditions are:

$$Q_1 = Q_{up} \tag{6.7}$$

$$y_n = y_{dn} \tag{6.8}$$

where Q_{up} is the discharge into the most upstream node of the channel network and y_{dn} is the flow depth of the most downstream node of the channel network. In general cases, the number of boundary condition equations is equal to the total number of upstream nodes and downstream nodes of the network.

6.3. Equation formulation for a general channel network and solution method

6.3.1. General formulation

In a channel network, at any cross-section, i, the flow depth y_i and discharge Q_i may be unknown. For a network of M channels where each channel i has N_k reaches (k is the channel number), the total number of unknowns is equal to $2\sum_{i=1}^{M}(N_k + 1)$.

At each reach, we can write a set of two equations, including continuity and energy equations, so the total number of equations for these reaches are $2\sum_{k=1}^{M} N_k$. The remaining 2M equations will be given as junction equations or boundary conditions.

From the above formulation of the continuity and energy equations for channel reaches, junctions, and boundary conditions for a network of M channels and each channel has N_k reaches, we can form a system of $2\sum_{i=1}^{M}(N_k + 1)$ equations with $2\sum_{i=1}^{M}(Ni + 1)$ variables which can be solved using the Picard method (1929).

6.3.2. Picard iterative scheme

In this study, preliminary numerical testing solutions of the gradually varied flow equations using Newton iterative method were not successful for the complicated channel networks such as the one in Quang Nam basin. The solutions failed to converge if the initial assumed solutions were not close enough to the converged solutions. This conclusion coincides with the observations of poor global convergence properties by Press *et al.* (1992) and Szymkiewicz (2010).

In the Picard iterative scheme for solving the system of non-linear equations, flow variables (y_i, Q_i) are solved iteratively via the system of nonlinear equations in the form of $A \cdot x = b$ where A is the coefficient matrix, x is the vector of flow variables and b is the intercept vector. The system of equations is set up using continuity and energy balance equations between different type of nodes as described above. In this study, an improved Picard method adopted from Szymkiewicz (2010) was used to solve the system, as follows:

$$A^* \cdot x^{(k+1)} = b \quad (6.9)$$

Where k is the index of iteration and

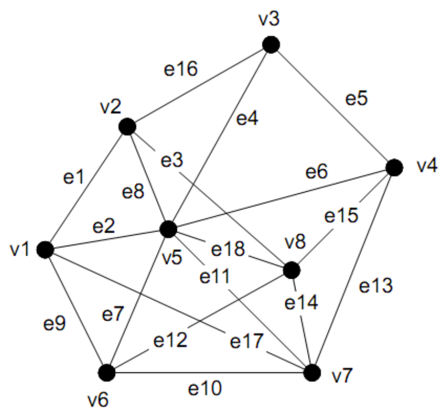
$$A^* = A(0.5(x^{(k)} + x^{(k-1)})) \quad (6.10)$$

is the modified matrix of coefficients. This means that to calculate the vector x in iteration k+1, the matrix A is calculated using the arithmetic average value of x from two proceeding iterations. For k = 1, $A^* = A(x^{(0)})$.

6.4. Assembling the system of equations using Graph Theory

In this section, a systematic approach for assembling the system of equations for general channel networks with multiple inflow/outflow boundaries, loops, combining and splitting junctions is proposed. The following discussion relating to graph theory, which is adapted from Steen (2010), is relevant for representing a general channel network only. Extensive discussion about graph theory could be found in Steen (2010).

A graph is a collection of vertices that are connected to each other via edges. Each edge in a graph joins exactly two vertices. The following example (given in Steen 2010) is an example of a graph.



$$V(G) = \{v_1, \ldots, v_8\}$$
$$E(G) = \{e_1, \ldots, e_{18}\}$$

$e_1 = \langle v_1, v_2 \rangle$	$e_{10} = \langle v_6, v_7 \rangle$
$e_2 = \langle v_1, v_5 \rangle$	$e_{11} = \langle v_5, v_7 \rangle$
$e_3 = \langle v_2, v_8 \rangle$	$e_{12} = \langle v_6, v_8 \rangle$
$e_4 = \langle v_3, v_5 \rangle$	$e_{13} = \langle v_4, v_7 \rangle$
$e_5 = \langle v_3, v_4 \rangle$	$e_{14} = \langle v_7, v_8 \rangle$
$e_6 = \langle v_4, v_5 \rangle$	$e_{15} = \langle v_4, v_8 \rangle$
$e_7 = \langle v_5, v_6 \rangle$	$e_{16} = \langle v_2, v_3 \rangle$
$e_8 = \langle v_2, v_5 \rangle$	$e_{17} = \langle v_1, v_7 \rangle$
$e_9 = \langle v_1, v_6 \rangle$	$e_{18} = \langle v_5, v_8 \rangle$

Figure 6.1 An example of a graph with 8 vertices and 18 edges

Formally, a graph is defined as follows (Steen 2010):

Definition 1 *A graph G consists a collection V of vertices and a collection of edges E, for which we write G = (V, E). Each edge* $e \in E$ *is said to join two vertices, which are called its end points. If e join* $v_i, v_j \in V$*, we write* $e = \langle v_i, v_j \rangle$*. Vertex* v_i *and* v_j *in this case are said to be adjacent. Edge e is said to be incident with vertices* v_i *and* v_j*, respectively.*

Definition 2 *For any graph G and vertex* $v \in V(G)$*, the neighbour set N(v) of v is the set of vertices (other than v) adjacent to v, that is* $N(v) \overset{\text{def}}{=} \{w \in V(G) | w \neq v, \exists e \in E(G): e = \langle w, v \rangle\}$

Definition 3 *A directed graph or digraph D consists of a collection vertices V, and a collection of arcs A, for which we write* $D = (V, A)$*. Each arc* $a = \langle \overrightarrow{u, v} \rangle$ *is said to join vertex* $u \in V$ *to another (not necessarily distinct) vertex v. Vertex u is called the tail of a, whereas v is its head.*

Definition 4 *Consider a directed graph D and vertex* $v \in V(D)$*. The in-neighbor set* $N_{in}(v)$ *of v consists of the adjacent vertices having an arc with v as its head. Likewise, the out-neighbor set* $N_{out}(v)$ *consists of the adjacent vertices having an arc with v as its tail. Formally:*

$$N_{in}(v) \overset{\text{def}}{=} \{w \in V(D) | w \neq v, \exists a = \langle \overrightarrow{w, v} \rangle : a \in A(D)\} \quad (6.11)$$

$$N_{out}(v) \overset{\text{def}}{=} \{w \in V(D) | w \neq v, \exists a = \langle \overrightarrow{v, w} \rangle : a \in A(D)\} \quad (6.12)$$

The set of neighbours $N(v)$ of vertex v is simply the union of its in-neighbors and out-neighbors, i.e., $N(v) \stackrel{\text{def}}{=} N_{in}(v) \cup N_{out}(v)$.

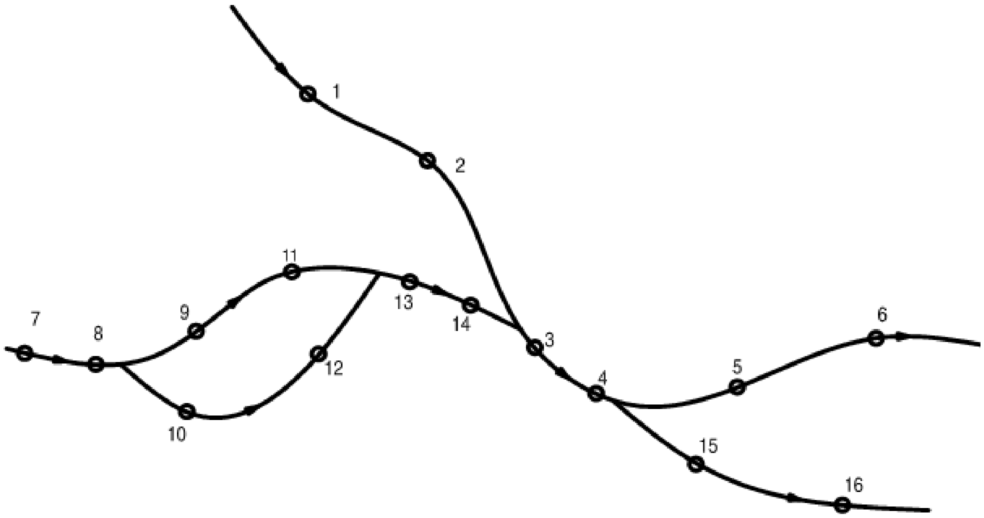
In computation, the representation of a graph could be made via an adjacent matrix. Consider a graph G with n vertices and m edges. Its adjacent matrix is a square matrix of $n \times n$ elements, $A[i,j]$ denoting the number of edges joining vertices v_i and v_j. For a directed graph, an adjacent matrix, A, in which $A[i,j]$ is equal to the number of arcs joining vertex v_i to vertex v_j. The following properties hold for a directed graph:

- A digraph D is strict if and only if for all i and j, $A[i,j] \leq 1$ and $A[i,j] = 0$. There can be at most one arc joining two vertices v_i and v_j, and no arc joining a vertex to itself.
- For each vertex $v_i, \sum_j A[i,j] = \delta_{out}(v_i)$ and $\sum_j A[j,i] = \delta_{in}(v_i)$. The sum of entries in row i corresponds to the number of out-arcs, whereas the sum of entries in column i equal to the number of in-arcs.

However, from a computation perspective, the representation of directed graph using an adjacent matrix could be memory inefficient. An equivalent alternative is an adjacent list. It is better to discuss about adjacent list via an example, shown in Figure 6.2.

It is possible to conceptually consider a general channel network as an analogy to a digraph in Graph Theory where each cross-section is a vertex, and a reach between any two vertices is a directional edge. The direction is the flow direction. Figure 6.2 shows an example of a channel network and its corresponding adjacent list. In the adjacent list, each vertex, v, links to its adjacent vertices $N_{out}(v)$ in the flow direction.

With this representation, no special method for encoding the river cross-sections is necessary. All the information needed are the unique identifications of river cross-sections and adjacent downstream cross-section ids. This method allows for a fully comprehensive representation of any river network in a simple data structure. Later in this chapter, in section V.3, it can be seen that all this information can be conveniently captured while digitizing river cross-sections.



1 → 2
2 → 3
3 → 4
4 → 5
5 → 6
6 → End
7 → 8
8 → 9 →
10
9 → 11
10 → 12
11 → 13
12 → 13
13 → 14
14 → 3
4 → 5→
15
5 → 6
15 → 16
16 →
End

Figure 6.2 Channel network and corresponding adjacent list data structure

To implement the network in a computer program, the above mentioned structure has to be precisely mapped into a data structure that allows for storing such a graph. In this study, a VB.Net OOP (Object Oriented Programming) class (Michael, 2005) was designed to store the structure and to process the related operations. The main data member of this class is an adjacent list data structure (a jagged array of arrays in VB language). With this structure, the hierarchy relationships among channel cross-sections and configuration structures (upstream boundary, downstream boundary, split junction, combined junction, simple reach) of a channel network could be described, stored and can be retrieved later. This information is fundamental for assembling flow equations at every reach of the channel network into a system of equations. No matter how many junctions and reaches a general channel network may have, no matter how complicated a system maybe (systems with loop, split, combined flows, with multiple upstream and downstream boundaries), all its simulation processes can be automated. Specifically, with information stored in the adjacent list, the suitable energy and continuity equations for each type of channel configuration structures, boundary conditions could be accurately identified and assembled into a non-linear system of the form A.x=b. Details about this are presented in the proposed algorithm in section V.6.

6.5. Implementation

VisualBasic.Net was used to develop a hydraulic modelling toolset that runs within a GIS platform (ArcGIS) as a graphical data presentation tier.

6.5.1. Elevation data

In this study, a digital elevation model (DEM) was used as the topographic platform for the simulation. The raster was processed so that the information about river bed topography was integrated into this DEM. Detail about this integration is discussed in the case study in Quang Nam basin, in sections 6.3 and 6.4.

6.5. 2. Channel cross-sections

The information about the channel network is stored in a polyline feature class. The polyline feature class has the following schema:

Table 6.1 River cross-section data structure

Field	Data type	Description
XY	Text	Cross-section information. Channel bed information is stored in pairs of distance-elevation, starting from the beginning of the polyline. These pairs are separated by % sign. # are used to mark the locations of main channel and the over bank areas. Example: 0.0000;193.0000%217.2544;192.0000%434.5089;118.0000%651.7633;75.0000#869.0178;41.0000%1086.2720;16.0000%1303.5270;6.0000%1520.7810;51.0000#1738.0350;95.0000%1955.2900;147.0000
Z	Text	The lowest elevation in the cross-section .
Avg_Mng	Text	Average Manning values for left bank area, main channel, and right bank area. Example: 0.035#0.025#0.034

Code	Text	A unique number used to identify the cross-section.
Y	Double	The initial condition of flow depth at each cross-section.
Downst_CSs	Text	A list of the downstream cross-sections of the current cross-section. This information allows for building up the structure of the channel system into an adjacent list.
Distances	Text	Distances from the current cross-section to the downstream cross-sections.
boundary_y	Double	Only for downstream cross-sections. Contain the boundary conditions of flow depths at the end cross-sections.
boundary_q		Only for upstream cross-sections. Contain the boundary conditions of discharges at the upstream end cross-sections.
Predict_y	Double	Predicted flow depths, which are the results of the simulation.
Water_elev	Double	Water_elev = z + Predict_y, which are also the results of the simulation.

A tool in ArcMap was developed to generate the channel cross-sections from a digital elevation model (DEM). The information about cross-section profile is captured via digitizing the cross-sections on the DEM. When the mouse is dragged over the DEM to create a polyline of the cross-section, the tool will sample the DEM along this polyline at a user specified interval and the information will be processed and stored in the feature class attribute table.

The digitizing process should be made from downstream to upstream so that the information about the downstream cross-sections of the currently digitized cross-section could be made available. The tool also allows for identifying the downstream cross-sections and corresponding distances of the currently digitized cross-section by pointing to those cross-sections during the cross-section digitizing process.

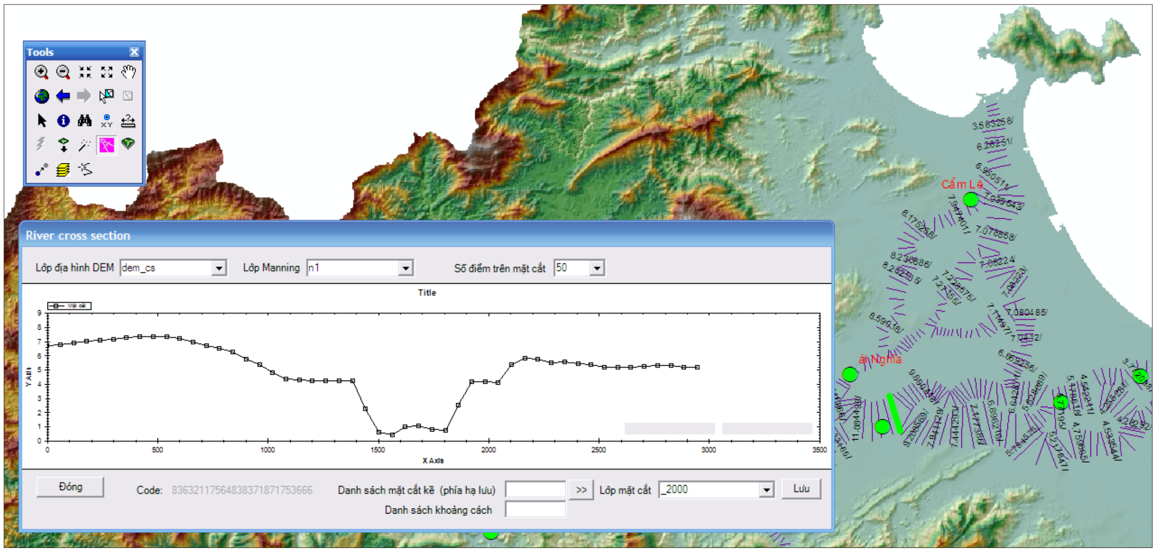


Figure 6.3 River cross-section editing tool

6.5.3. Computing parameters at a cross-section

From the geometry information of a cross-section together with Manning's values for left bank, right bank and main channel, the parameters for each channel cross-section, corresponding to a flow depth, need to be computed in order to simulate the flow along a channel. These parameters include: flow area (A), wetted perimeter (P), hydraulic radius (R=A/P), conveyance (K), friction slope (Sf), energy correction coefficient (α), top width (B), flow velocity (v), energy head (H). Formulas in Chaudhry (2007) were used to calculate these parameters.

A summary of these calculations in an OOP class is given in the following class diagram. Appendix 6.1 gives a more detailed implementation of the class in VB.Net language.

<<RiverCrossSection>>
⊸ **PointList** As String ⊸ **Manning** As String ⊸ **Q** As Double ⊸ **FlowDepth** As Double

New(ByVal PointList As String, ByVal ManningList As String, ByVal flowDepth As Decimal, Q As Decimal)

CalculateCSProperties()

← **K**() As Decimal

← **B**() As Decimal

← **R**() As Decimal

← **A**() As Decimal

← **WP**() As Decimal

← **Sf**() As Decimal

← **Alfa**() As Decimal

← **H**() As Decimal

Figure 6.4 RivercrossSection class

6.5.4. Processing channel network hierarchy

As mentioned above, from an adjacent list of a channel network, it is necessary to identify if a cross-section is an upstream or downstream end, participating in a channel joining, splitting junction or a normal reach so that appropriate continuity and energy equations can be derived for each reach in the network. The implementation of these operations was undertaken via an OOP class. The summary of the functionalities of the class is given in the diagram in Figure 6.5.

<<RiverNetwork>>
⊸ **AdjList** As String()() As String
←**GetDownstreamCSs**(ByVal CrossSectionID As String) As String()'get the downstream cross-section list of a cross-section

←**GetUpstreamCSs**(ByVal CrossSectionID As String) As String()'get the upstream cross-section list of a cross-section ←**UpstreamEndCSs**() As String()'get the upstream ends of the network

Figure 6.5 RiverNetwork class diagram

The OOP RiverNetwork Class summarises the implementation of the searching functionalities on a channel network (an adjacent list of cross-sections) to count and identify a list of downstream cross-sections of the current stream cross-section, so that we know if the current cross-section participates in a splitting junction or not. This class also allows us to identify and count a list of upstream cross-sections so that we know if the current cross-section is in a junction of combining flows or not. The class also allows for generating a list of upstream ends. Information about downstream ends is available in the adjacent list (the cross-sections that point to no other cross-sections are downstream ends).

Generating a list of downstream cross-sections for a specific cross-section via the GetDownstreamCSs function is just a matter of identifying the location of the cross-section and returning the array of linked nodes, $N_{out}(v)$. To generate a list of upstream cross-sections of a specific cross-section via GetUpstreamCSs function, all the elements in the jagged array are scanned through nested loops for the existence of the node. If the node exists in an array of linked nodes (other than the first element) the first element of that array will be returned. $N_{in}(v)$ is formed during the scan. The UpstreamEndCSs function generates a list of upstream nodes by scanning through the jag array in nested loops. If there are no node links to a node, $N_{in}(v) = \emptyset$, then that node is identified as an upstream end node.

6.5. 5. Simulating gradually varied flows in a general channel network

The following pseudo code was used to set up and solve the system of equations for simulating subcritical gradual varied flow in a general channel network. Note that the matrix A in the algorithm is the coefficient matrix. The elements in the left half of the coefficient matrix (columns 1..N) are used to store the coefficients for flow depth variables and the right half (columns N+1..2N) for discharge variables. Implementation of the algorithm is given in appendix 6.4.

- **Setup a matrix A (2N x 2N) and a vetor B (2N).**
- **Instantiate the Network object with the adjacent list of cross-sections from the feature class.**

Begin: Scan through the cross-section list, one at a time. For each cross-section, get the list of the downstream cross-sections from the network object

If nothing in the list, it is a downstream end:

> Apply the boundary condition of the stream end flow depth to write the energy equation to matrix A and vector B for this boundary.

If it found more than one downstream cross-section, then this is a split flow:

> Scan through the downstream cross-section list; write the following equations into matrix A and vector B: 1)for each downstream cross-section write the energy equations. 2)Write the continuity equation for the split flow junction.

If it found 1 downstream cross-section, it is may be in a normal reach or it may participate into a combine junction, then further check this:

Get the list of upstream cross-section of the current cross-section:

> If the number in this list is 1 then it is in a normal reach with one upstream cross-section and one down stream cross-section:
>
> Between these two cross-sections, write continuity and energy equations into matrix A and vector B.
>
> If the number in this list is > 1 then it is in a combine flow junction:
>
> Scan through the uptream cross-section list, write the following equations into matrix A and vector B; 1) for each upstream cross-section

```
    write the energy equations. 2) write the continuity equation for the
    combine flow junction.
Go back to :Begin- until all the cross-sections are scaned.

Retrieve the upstream end cross-sections from the network object. For each up
stream end cross-section, write one continuity equation.

Invert the system using improved Picard method. The solver will terminate when
the convergence condition is satisfied.
```

6.6. Case study

6.6.1. The river network and the basin

A detailed description of the study area is given in chapter 2. The following brief description of Quang Nam basin is used to serve the purpose of providing relevant context for the discussion about hydraulic simulation in the basin.

The Vu Gia - Thu Bon river system originates from the Truong Son mountain range of Vietnam and runs eastwards to the sea. In the downstream reaches, riverbanks become low, allowing overflow onto fields during the flood season. The Vu Gia - Thu Bon river system has two main sub tributaries, the Vu Gia and Thu Bon Rivers. The Vu - Gia River runs to the mouth in Da Nang city with a total length of 204 km. The Thu Bon River runs in a north-south direction then changes its course to flow South-west – North-east and then West-east up to Giao Thuy before entering the sea through the Dai estuary. The total length of the river is 152 km. Figure 6.6 shows the river system in the study area.

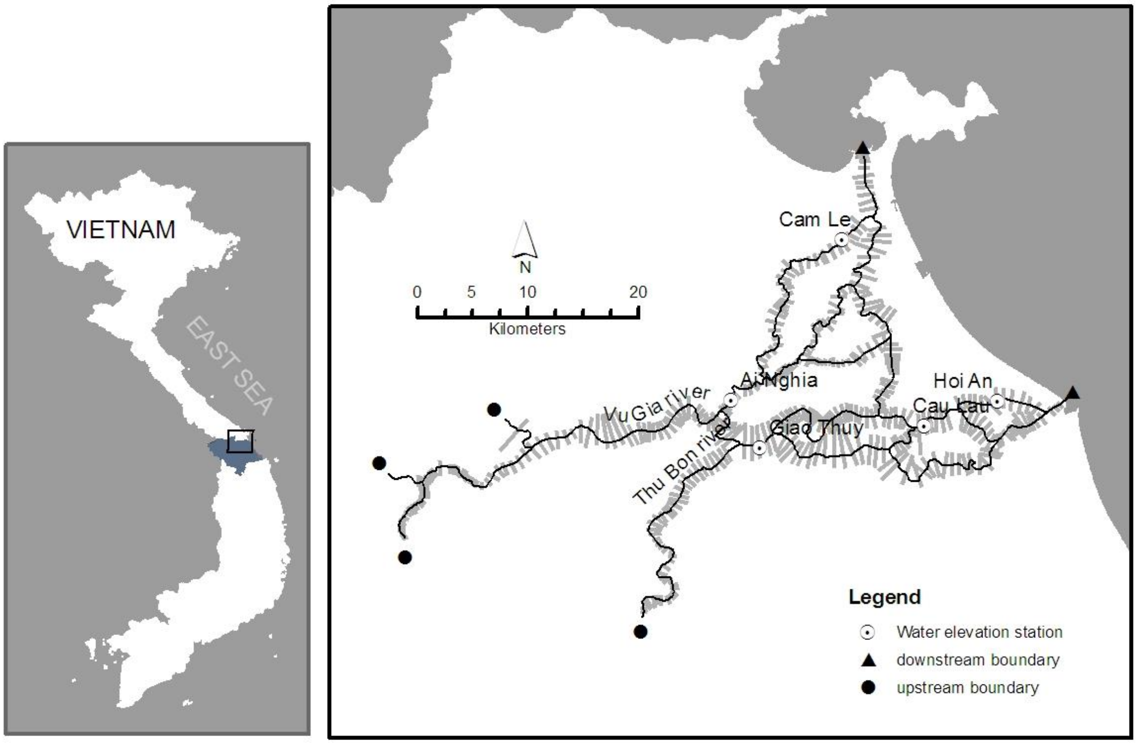


Figure 6.6 Study area, Quang Nam basin of Vietnam

6.6.2. Geographical data and processing

In this case study, the elevation and cross-section data were provided by the Department of Science and Technology of Quang Nam. Elevation data was provided in the form of digital elevation model (DEM) in ESRI raster format, with a 25 meter resolution. River cross-sections were provided in the form of polyline feature class in ESRI vector format (GDB feature class). Both data layers are in Universal Transverse Mecator 48-North (UTM-48 N) coordinate system.

6.6.3. Rectifying the cross-section data

The objective of the data processing is to prepare a raster layer with information about channel bed topography integrated, due to the fact that cross-section data and elevation data are measured at different times, elevations and river bed stages change over time. In addition, the capturing of these cross-section and elevation data was done using different technologies. As a consequence, elevations in the DEM data and cross-sectional data did not match each other. Without a visualization tool, this issue is normally undetectable and ignored.

To fix this problem, a tool was developed to:

- Visually examine the difference between cross-sectional data and elevation data from a DEM.
- Blend the cross-sections into the DEM elevations, so that when carrying out hydraulic modelling, abrupt changes in flood maps could be avoided.

The objective of the adjustment process was to make the elevations of cross-section ends match with elevations of the DEM while keeping elevations within the river channel unchanged. As an example, Figure 6.7 below describes the adjustment process. The continuous line is the DEM elevation. It can be seen that the cross-section (cross marked) is quite different from the DEM elevation. After adjustment, the cross-section (circle marked) blend well into the DEM elevation on both banks while the main channel is kept untouched.

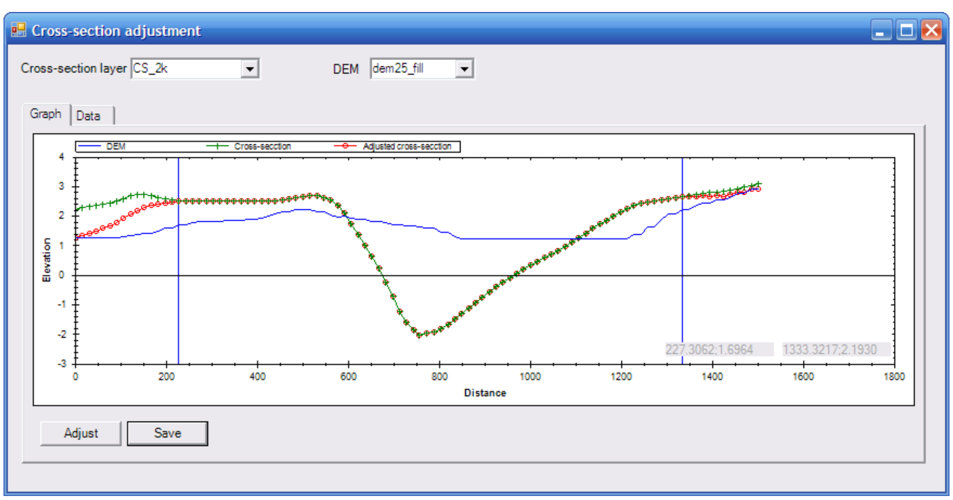


Figure 6.7 Tool for adjusting river cross-section

To do this, a weighted linear combination algorithm was implemented. The following formula shows the algorithm:

$$h_{adjusted} = \frac{x}{totalDis} \times h_{raster} + \frac{totalDis - x}{totalDis} \times h_{cross-section} \qquad (6.13)$$

For each river bank, *totalDis* is the distance from the end point of the cross-section to the point from that we want to leave unaltered. The adjusted elevation of a point that is x meters from the end point on a river cross-section is a weighted linear combination of the elevation from the DEM and the elevation of the cross-section at that point before adjustment. Using this tool, all the cross-sections in the dataset were checked and rectified with the elevation data.

6.6.4. Generating the river bed mesh

To prepare the DEM for the hydraulic modelling, it is necessary to integrate the channel bed topographic data into the DEM elevation data. Current interpolation algorithms in ArcGIS™ do not allow for interpolation of the topographic feature from cross-sections because they do not take into consideration the flow direction. A tool was developed to facilitate the interpolation of the channel bed topography from cross-sections. The tool generates a mesh by linking elevations from cross-sections. A simple algorithm for this tool was employed. Firstly the tool detects the lowest points of an adjacent pair of cross-sections and links of these points together to form the central line of the channel. On the right and the left of the central line, mesh lines are added in a linear manner, with a density proportional to the left and right bank areas. Figure 6.8 shows an example of the resulting mesh for channels from cross-sections.

From this, a triangulated irregular network (TIN) model of the channel bed topography is generated. This TIN was then rasterized and integrated into the DEM of Quang Nam basin to form the complete elevation model for hydraulic simulation. Note that the DEM was resampled to a resolution of 15 meter using bilinear interpolation method before integration.

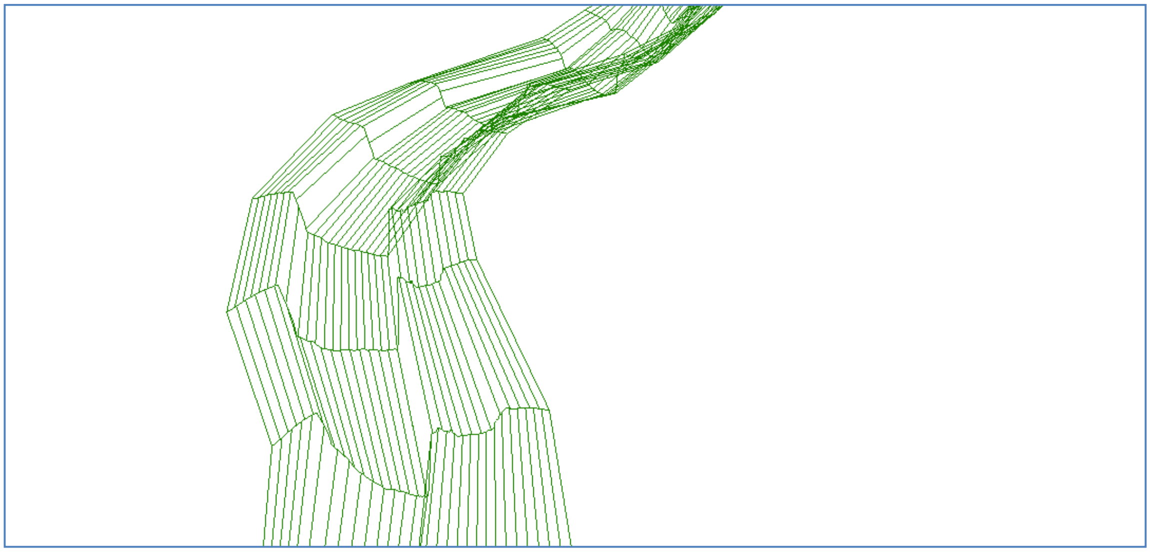


Figure 6.8 An example of channel bed mesh

From this, a set of 363 cross-sections was generated using the tool described in section V.3, to be used to simulate the water elevations in channels.

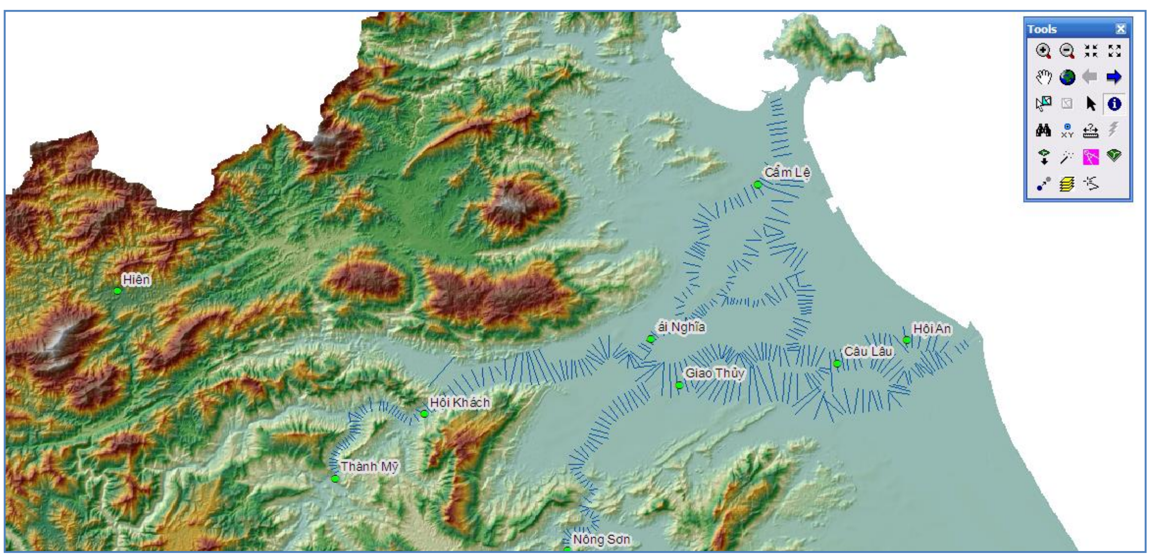


Figure 6.9 Channel cross-sections of Vu Gia – Thu Bon river system

6.6. 5. Hydrological data

Time series data including discharge at 2 stations (Nong Son and Thanh My), water elevations at 5 stations (Ai Nghia, Giao Thuy, Cau Lau, Hoi An and Cam Le) and sea level data at one station (Son Tra) were provided by the Department of Science and Technology of Quang Nam. Time series data have a temporal resolution of 6 hours.

To run the simulation for steady state gradually varied flow problem, the required upstream boundary condition is the discharge, and the downstream boundary condition is the known water level. For Quang Nam basin, there are only two discharge stations, Nong Son and Thanh My. The peak flow of the events at these two stations was used. The flows from Song Kon river (catchment area: 2413 ha) and Song Bac – Song Nam river (catchment area: 624 ha) were estimated as proportions of discharge of the near by sub-basin (Vu Gia basin, Thanh My station, catchment area of 1408 ha). Discharge from Song Kon river was 1.714 of that of Vu Gia basin, and discharge from Song Bac- Song Nam river was 0.443 of that of Vu Gia basin. For the downstream boundary condition, the highest water levels in Son Tra station were used. Discharge and water elevation data for large events in 1998, 1999, 2007, 2008, 2009 were used for model calibration and validation. The data are given in Table 6.2.

Table 6.2 Peak discharges and water elevations at stations in Quang Nam basin for flood events in 1998, 1999, 2007 and 2009

Events	Discharge	Sea level in Son Tra (m)	Water elevations at stations (m)
1998 (19-24 Dec.)	Nong Son: 10600 Thanh My: 7000	0.93	Cau Lau: 5.09, Hoi An: 2.99, Giao Thuy: 18.14, Ai Nghia: 10.37, Cam le: 3.31
1999 (5-10 Dec)	Nong Son: 10600 Thanh My: 2690	0.87	Cau Lau: 4.54, Hoi An: 2.62, Giao Thuy: 15.47, Ai Nghia: 9.43, Cam le: 2.50

2007 (10-15 Nov.)	Nong Son: 10600 Thanh My: 5280	0.94	Cau Lau: 5.39, Hoi An: 3.28, Giao Thuy: 17.67, Ai Nghia: 10.36, Cam Le: 3.98
2008 (15-21 Oct.)	Nong Son: 6710 Thanh My: 2400	0.75	Cau Lau: 3.94, Hoi An: 1.15, Giao Thuy: 8.81, Ai Nghia: 9.10, Cam Le: 2.16
2009 (23-30 Sep.)	Nong Son: 9000 Thanh My: 2710	1.65	A set of 150 survey data points of this year was used for calibration.

6.6.6. Calibration and validation

Simultaneously solving systems of flow equations in general channel networks can be challenging. In addition, the practical application of the solution results is complicated due to the simultaneous nature of the problem and depends on the calibration of the model. Calibration in the context of general channel networks means finding a set of Manning's roughness coefficients for every channel cross-section of the network, including main channels and banks, which makes the simulated water elevations and discharge rates best match the observed data within an acceptable error. To illustrate the complexity considering a simple case, assume that a channel network has M channels, and there are N discrete Manning's roughness coefficients to choose from, and also assume that at each channel cross-section, one Manning value is used for the main channel and one value is used for left and right banks. With this configuration, at each cross-section, there are N^2 ways of choosing Manning's roughness coefficients, and for the whole channel network, the number of possible configurations isN^{2M}. For a simpler scenario, one Manning's value for each channel, this figure is N^M. For a channel network on a scale of a few dozen channels, these numbers become extremely large, even for the case of discrete Manning's values. From this argument, an exhaustive scanning algorithm is not practically feasible.

In most applications of simulating flow variables in channel networks, a trial and error approach is adopted, which can only be applied for the case of single channel or simple network simulations. This approach is popular and can be found in a number of recent studies such as, for example, in Hameed and Ali (2013). The authors used a manual approach for calibrating the Manning's roughness coefficients for the events in 2008 for the Hilla River in Jordan. Parhi *et al.* (2012) used various single Manning's roughness coefficients to simulate the discharge values of an event in 2003 for the Mahanadi River, India. Timbadiya *et al.* (2011) firstly carried out the calibration for a single value for a whole river reach in the Lower Tapi River in India. Subsequently, different values were used to refine the results to match the observed data for events in 1998 and 2003. Usul and Turan (2006), starting from the Manning's values of 0.033 for the main channel and 0.06 for the flood plain, applied a trial and error technique to adjust the Manning's values (within a range of 0.020–0.075 for the riverbed and 0.040–0.14 for the floodplains) to match the observed rating curves with the simulated ones for an event of 1991 on the Ulus river in Turkey. Kidson *et al.* (2002) used this approach to calibrate the Manning's roughness coefficients for a 1.5 km reach of the Mae Chaem River in Thailand for an event in 2001. Although the application of this approach was effective in these cases, it cannot be applied for the calibration of complicated general channel networks as the behavior of the water elevations and discharges is not predictable. In a general channel network, the adjustment of the Manning's roughness coefficients will not only cause changes in discharge and water elevations of the reaches that are being adjusted but also in the reaches that are not adjusted in an unmanageable manner.

In the context of a general channel network, a more systematic approach is necessary. An alternative approach to the problem is integrating the numerical procedure of the first approach with an optimization model (Das, 2004). By doing this, the optimization facilitates the calibration procedure towards an optimal set of parameters more quickly. The application of such an approach can be found in Fread and Smith (1978). The authors used a modified Newton-Raphson scheme to find a set of Manning's values as a function of water elevations and discharges by minimizing the absolute sum of the difference between observed and simulated stage and discharge values. Ramesh *et al.* (2000) applied a quadratic programming algorithm (Powell, 1974) to optimize the Manning's value sets for a hypothetical single channel case and a three branch network with a wide rectangular cross-section. The finite difference approximation of the governing equations was embedded into the optimization as equality constraints. Das (2004) used a projected augmented Lagrangian method to solve the optimized model formulated for a steady state flow based on a hypothetical closed-loop channel network given in Chaudhry (1993). The test result gave simulated discharges within $\pm 17\%$ of the expected values, but in some channels the discrepancy went up to 75%. The author also confirmed the high sensitivity of water profiles to the Manning's roughness coefficients in the channel network. Optimization algorithms can be found in other hydrologic and hydraulic studies such as in Luce and Cundy (1994) who used Shuffle Complex Evolution algorithm to automatically calibrate rainfall-runoff models and Muleta and Nicklow (2005) applied a genetic algorithm in calibrating SWAT, a semi-distributed watershed model. Recently, Reshma *et al.* (2015) used Multi-objective Genetic Algorithm (MGA) to calibrate Manning's roughness coefficients for overland flow for different land use types of a rainfall runoff model for eleven rainfall events of Harsul watershed in India. The results showed a reasonable agreement between the observed and simulated hydrographs.

In this study, Simulated Annealing (SA) was used to test the capability to calibrate the Manning's roughness coefficients for a general channel network of a basin scale. Simulated Annealing, which was introduced by Metropolis *et al.* (1953), analytically mimics the slow cooling process of materials in a heat bath and has been used to simulate complex systems. Kirkpatrick *et al.* (1983) applied the concept to optimization problems in computer system design and the travelling salesman problem. Simulated Annealing has been introduced into hydrology in a few studies such as in Dougherty and Marryott (1991). The authors demonstrated the flexibility of the method and its potential for solving groundwater management problems via a number of applications to problems of groundwater flow and remediation strategies. Marryott *et al.* (1993) used Simulated Annealing in combination with a field-scale flow and transport simulation model for an unconfined aquifer to determine pumping schedules for a pump-and-treat remediation system at a site in central California. Mauldon *et al.* (1993) used Simulated Annealing to model the flow in fractured rock.

The specific objective of this optimization is to test the applicability and performance of SA for estimating the Manning's values for the steady state gradually varied flow in a general channel network. The simultaneous solution process of the flow equations system, solved by an improved Picard's method (1929) was used to evaluate the objective function for the optimizer. With this approach, a number of advantages can be anticipated: i) since SA is a global optimization algorithm, it allows for avoiding local extrema of the objective function, ii) SA does not require any modification to the process of evaluation of the objective function. Therefore, the calibration process and the objective function evaluation could be independently coded. This makes SA easy to be integrated into other calibration processes and, iii) computationally, SA is simple and easy to implement.

6.6.6.1. Problem formulation

The formulation of the problem is presented via a general channel network. The Manning's values for each channel was assumed to be (1) uniform and constant along each channel in the network and, (2) the flow is steady state, gradually varied and subcritical. The optimization was carried out in a way that minimizes the objective function, which is the root mean square error (RMSE) between the observed and simulated water elevations "y" at the cross-sections. Formally, the optimization is:

$$Minimize: RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_{s,i} - y_{o,i}\right)^2} \qquad (6.14)$$

Where o is an observed and s is a simulated data value at cross-section i where the observed data value is available. The objective function is evaluated for each simulation run. An iterative solver using an improved Picard's method (1929), that was adopted by Szymkiewicz (2010), was used to evaluate the objective function for each set of Manning's values during the optimization process.

6.6.6.2. Simulated Annealing (SA)

The basic idea behind SA algorithms is searching for feasible solutions using analogies to the annealing process that converge to an optimal solution. Starting from a given configuration, at a high temperature, T, a system is subjected to an elementary modification. If the modification transforms the system into a new one with a lower "energy" level, for a minimization problem, or a higher "energy level", for a maximization problem, the modification is accepted; alternatively, if it causes an increase in the value of the objective function, for a minimization problem or decrease in the value of the objective function, for a maximization problem, it may also be accepted, but with a probability of $\exp(-\Delta E/T)$, where ΔE is the change in the value of the objective function. By repeating this rule of acceptance, a sequence of configurations is generated, constituting a Markov chain. It is possible to show that when the chain is of "sufficient" length, the system can reach thermodynamic equilibrium at the temperature considered. In other words, this leads us to a Boltzmann distribution of the energy states at this temperature. At high temperature, when $\exp(-\Delta E/T)$ is close to 1, almost all the transformations are accepted, and SA behaves like a random walk algorithm. When the temperature is lowered and $\exp(-\Delta E/T)$ is close to zero, fewer worse transformations are accepted, and SA resembles an iterative improvement algorithm. At an intermediate temperature, the SA intermittently authorizes the transformations that degrade the objective function, giving opportunities for the objective function to escape local extrema (Dréo *et al.*, 2006). SA is a global optimization algorithm where the acceptance of a worse move with a certain probability allows for the feasible solutions to escape the local extrema. This is different from a pure greedy hill climbing algorithm, which may tend to get stuck in local maximum/minimum solutions (Kumar, 2008).

In this study, each of the n channels in the channel network is given an identification number from 0 to n-1. The bed friction in each channel was assumed to be uniform, and a single value of Manning's coefficient was used for each channel. Initially, each channel was given a Manning value of 0.04. From this initial configuration, the SA algorithm randomly perturbed the Manning's values within a range of 0.03 to 0.05 (Prasuhn, 1992) for a number of channels in the network and then evaluated the objective function for the network. If the objective function was smaller than the previous objective function then the new Manning's configuration was accepted as the current friction status of the system. If not, the new configuration was accepted at a certain probability. This probability depended on 1) the "temperature", which is the total number of iterations minus the current iteration (i) and 2) the value of the objective function. The algorithm is as shown in Figure 6.10.

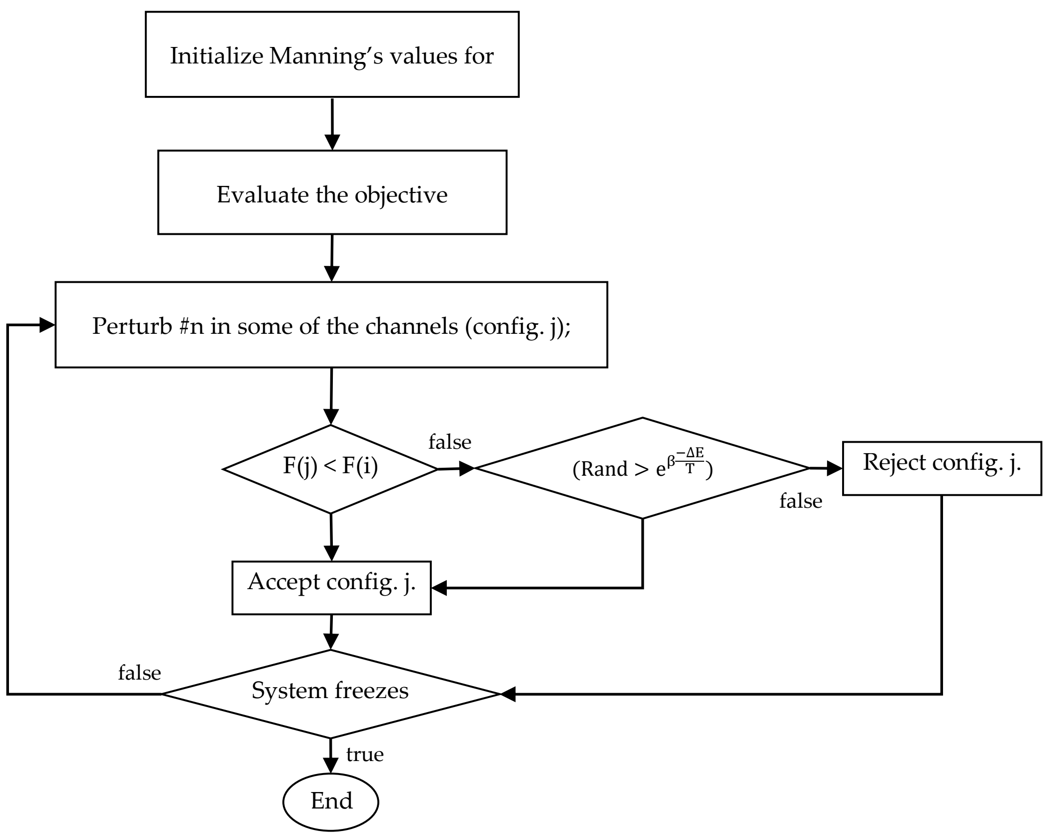


Figure 6.10 Simulated annealing algorithm for calibrating roughness coefficients for the channel network in Quang Nam

In Figure 6.10, F(i) is the value of the objective function at iteration i; ΔE is the change in the objective function value from configuration i to configuration j; β is the cooling control factor, which is used to control the rate at which the system is cooled down. Originally, the probability for accepting a worse move is $\exp(\frac{-\Delta E}{K_B T})$ where K_B is the Boltzman's constant (Laarhoven *et al.*, 1987). A large value of β makes the system cool faster, allowing less opportunity for a worse move, with the system tending to give simple hill climbing solutions, while a small value of β gives random search solutions.

In this study, objective function evaluations and Simulated Annealing functionalities are coded in VB.Net modules using VisualStudio™ 2008 as an extension to ArcGIS 9.3. This allows for the program to read geometry information of the cross-sections and the boundary conditions which are stored in a feature class of a personal geodatabase.

6.6.6.3. Testing SA technique

The application of SA in calibrating Manning's roughness coefficients is carried out for synthetic cases to see how SA can replicate known sets of Manning's roughness coefficients for different channel network configurations. Calibrations were carried out for synthetic systems of 5 channels (Figure 6.11), 9 channels (Figure 6.14) and 29 channels (Figure 6.17) with known Manning's values. For each of these synthetic channel systems, the water surface profile was obtained for an assumed set of values for Manning's roughness coefficients and the boundary conditions. The calibrations were carried out against these water surface profiles to examine the capability of the simulated annealing approach to replicate the "known" Manning's values. To assess the replicability of Manning's values of the algorithm, the following similarity formula, adopted from Gonzalez-Abril *et al.* (2014), is used. Given two one-dimensional datasets N_n and N_c representing the known and calibrated Manning's values. The distance between two interval I_1, I_2 derived from N_n and N_c is given by:

$$d_W(I_1, I_2) = \sqrt{(\Delta c, \Delta r)W(\Delta c, \Delta r)^T} \quad (6.15)$$

Where $c = 0.5(N_{max} + N_{min})$ is the center point and $r = 0.5(N_{max} - N_{min})$ is the radius of each dataset (Borelian notation) derived from N_n and N_c. W is the weight matrix. Then the similarity between the two datasets is given as:

$$K_w^l = \frac{\#((X \cup Y) \cap (I_X^l \cap I_Y^l))}{\#(X \cup Y)} \cdot \frac{1}{1 + d_w(I_X^l, I_Y^l)} \quad (6.16)$$

Where #A denotes the cardinality of set A. In this calculation l_2-distance was used, $I_X^2 = (\bar{X} \pm l.S_X)$. This similarity gives a value of 1 if N_n and N_c are identical in all respects and 0 if N_n and N_c are distinct in all respect.

Boundary conditions and Simulated Annealing parameters for synthetic cases are shown in Table 6.3 and 6.4, using flows at the upper boundary and water surface elevation at the lower boundary.

Table 6.3 Synthetic boundary conditions for the cases of 5, 9 and 29-reach systems

Boundary location (see maps)	5-reach system		9-reach system		29-reach system	
	Upper boundary (m3.sec-1)	Lower boundary(m)	Upper boundary (m3.sec-1)	Lower boundary (m)	Upper boundary (m3.sec-1)	Lower boundary (m)
A	1200	-	3000	-	2400	-
B	1100	-	3000	-	6170	-
C	3600	-	-	0.87	-	0.87
D	-	0.87	-	-	-	0.87
E	-	-	-	-	4000	-
F	-	-	-	-	1000	-

Table 6.4 The SA parameters for the synthetic cases of 5, 9 and 29-reach systems

system	Number of iterations	B	Number of perturbs
5 channels	100	200	1
9 channels	1000	250	3
29 channels	1000	250	3

6.6.6.3.1 System of 5 channels

Figure 6.11 5-channel synthetic case

The result of the optimization for 5 channels case is given in Figure 6.12. Out of 100 runs, 23 moves were rejected and 77 moves, including better moves and worse moves, were accepted. The process froze at an RMSE of approximately 0.25 m. No better solutions were found, and all the worse solutions were rejected afterwards. Solution 26 gives the lowest RMSE of 0.13 m.

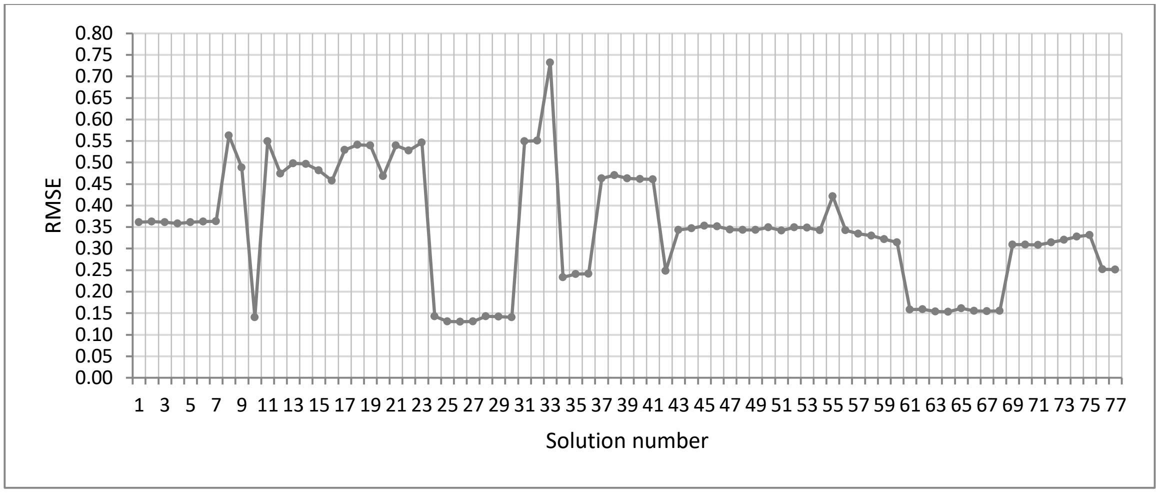


Figure 6.12 Simulated annealing solutions for the 5-channel synthetic case

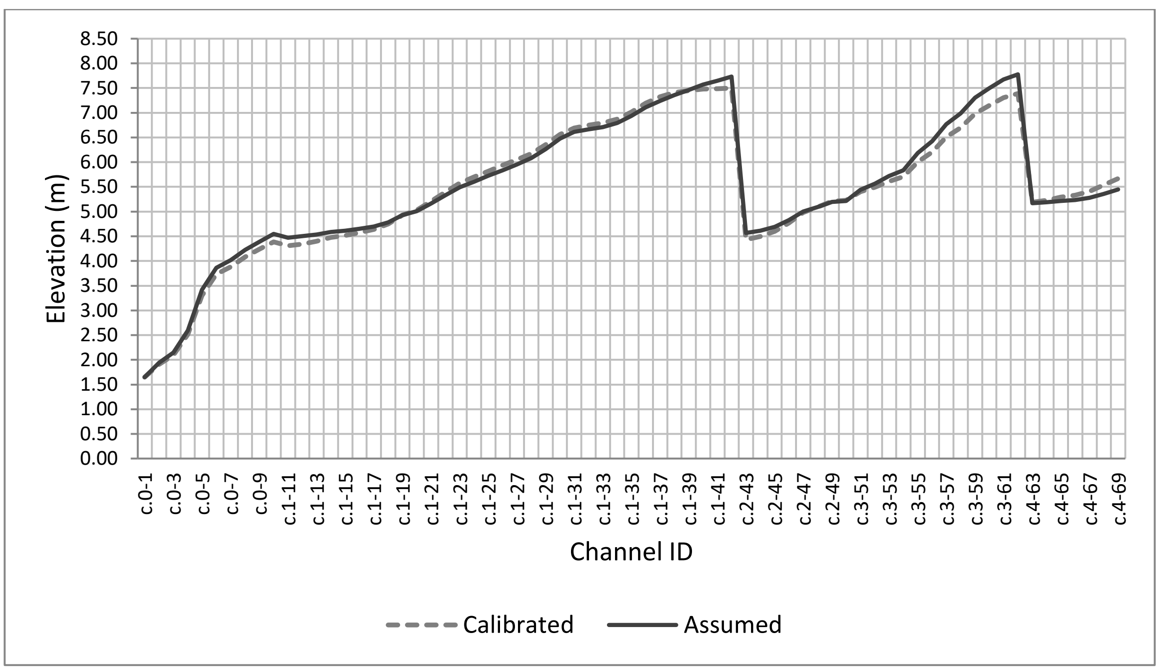


Figure 6.13 Assumed and calibrated water surface profiles for 5-channel synthetic case

The assumed and calibrated water surface profiles are given in Figure 6.13. Generally, the calibrated water surface profile fits well into the assumed water surface profile. The assumed and calibrated Manning's values for 5 channels are given in Table 6.5. The similarity (Eq. 6.16) between the assumed and calibrated Manning's value sets achieved was 1, although there was a slight different in the known and calibrated Manning's values at channel 3.

Table 6. 5 Assumed and calibrated Manning's values for 5-channel synthetic case

Channel ID	Assumed	Calibrated
0	0.041	0.041
1	0.045	0.045
2	0.043	0.043
3	0.041	0.043
4	0.034	0.034

VI.6.3.2. System of 9 channels

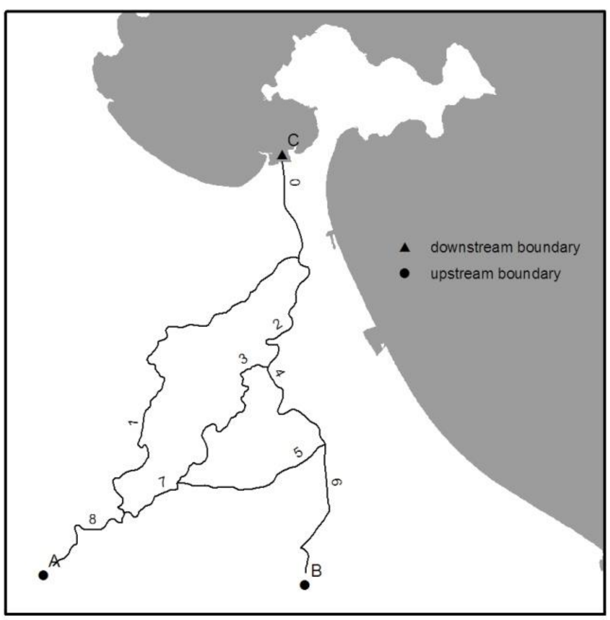


Figure 6.14 9-channel synthetic case

The result of the optimization for 9 channels case is given in Figure 6.15. Out of 1000 runs, 169 moves were rejected and 831 moves, including better moves and worse moves, were accepted.

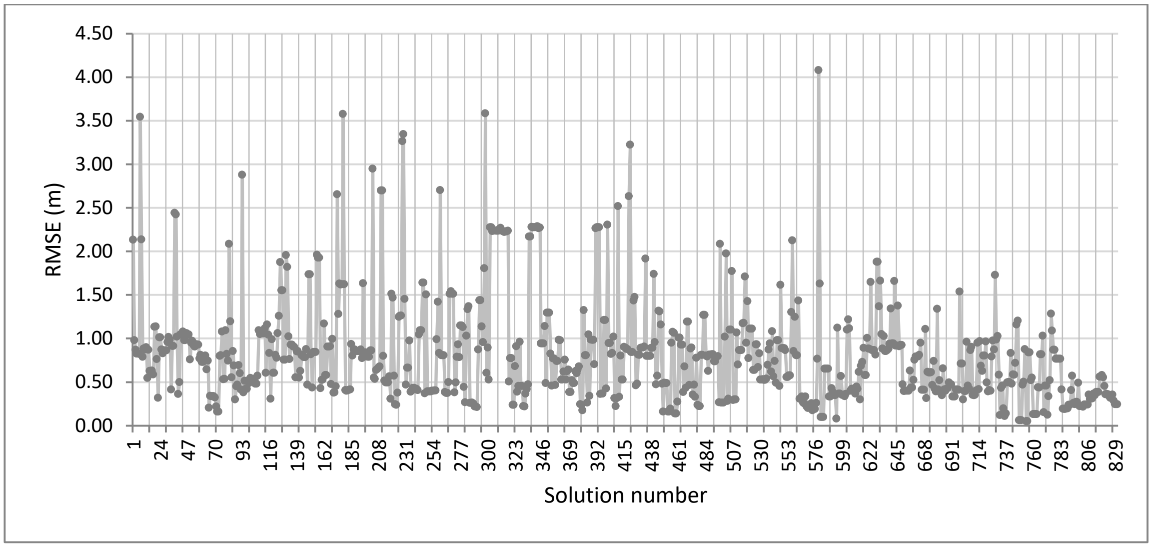


Figure 6.15 Simulated annealing solutions for the 9-channel synthetic case

The process froze at an RMSE of approximately 0.26 m. No better solutions were found, and all the worse solutions were rejected afterwards. Solution number 754 gives the lowest RMSE of 0.05 m.

The assumed and calibrated water surface profiles are given in Figure 6.16. Generally, the calibrated water surface profile fits well into the assumed water surface profile. The assumed and calibrated Manning's values for 9 channels are given in Table 6.6. The achieved similarity (Eq. 6.16) between the assumed and calibrated Manning's value sets is 0.998.

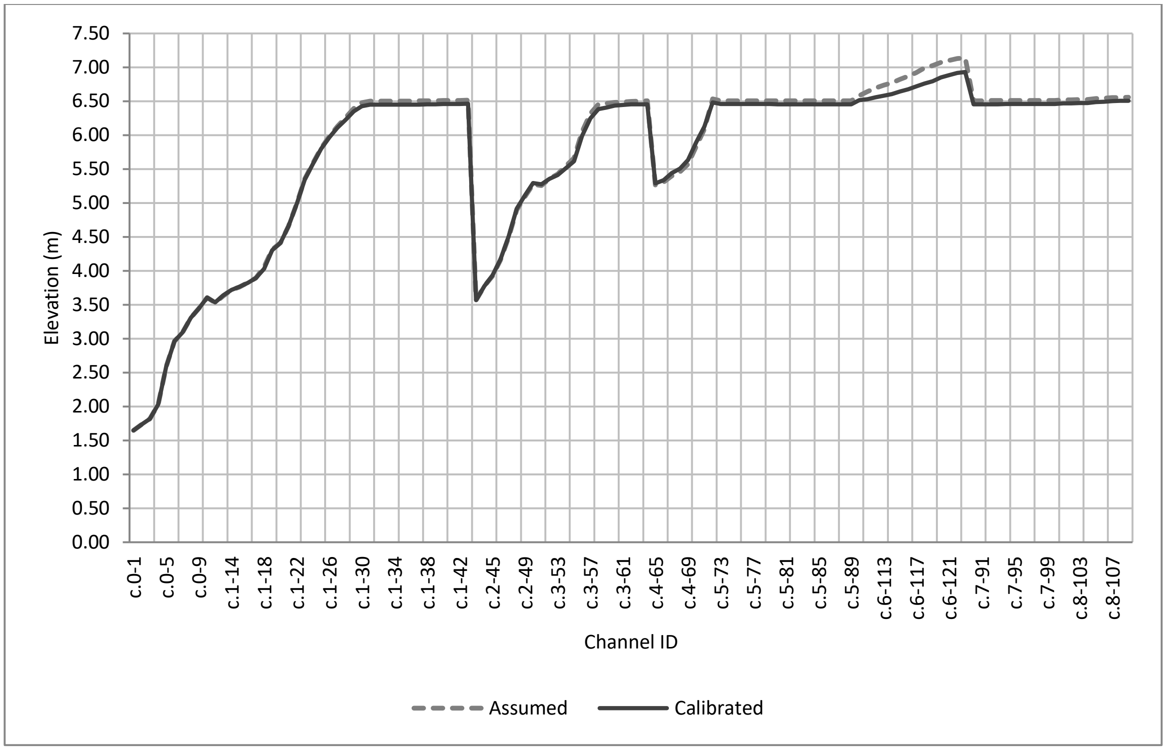


Figure 6.16 Assumed and calibrated water surface profiles for 9-channel synthetic case

Table 6.6 Assumed and calibrated Manning's values for 5-channel synthetic case

Channel ID	Assumed	Calibrated
0	0.032	0.032
1	0.034	0.034
2	0.044	0.045
3	0.044	0.042
4	0.034	0.036
5	0.042	0.038
6	0.039	0.036
7	0.050	0.048
8	0.046	0.045

6.6.6.3.3. System of 29 channels

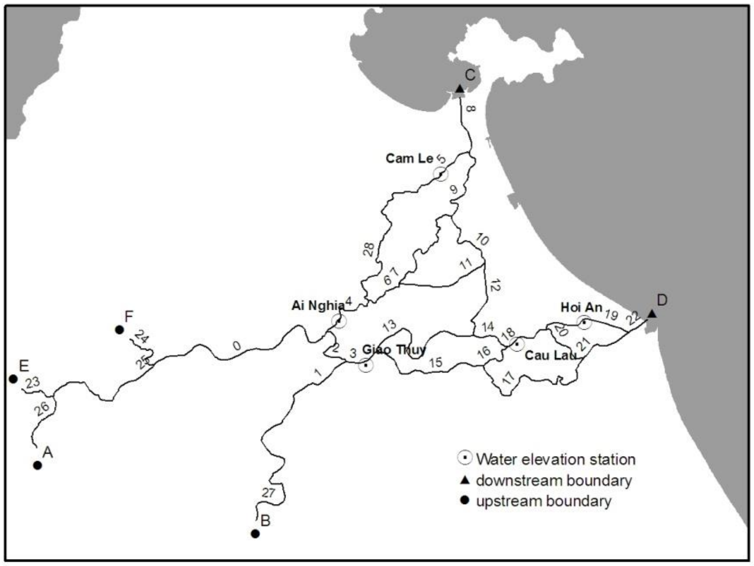


Figure 6.17 29-channel synthetic case

The result of the optimization for 29 channels case is given in Figure 6.17. Out of 1000 runs, 296 moves were rejected and 704 moves, including better moves and worse moves, were accepted. The process froze at an RMSE of approximately 1.47 m. No better solutions were found, and all the worse solutions were rejected afterwards. Solution 638 gives the lowest RMSE of 0.96 m.

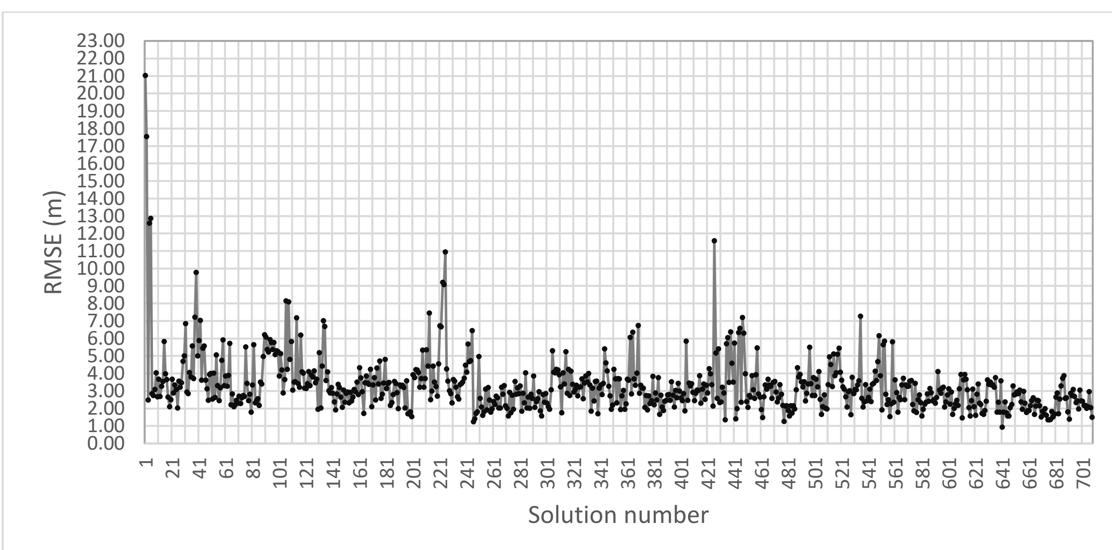


Figure 6.18 Simulated annealing solutions for the 29-channel synthetic case

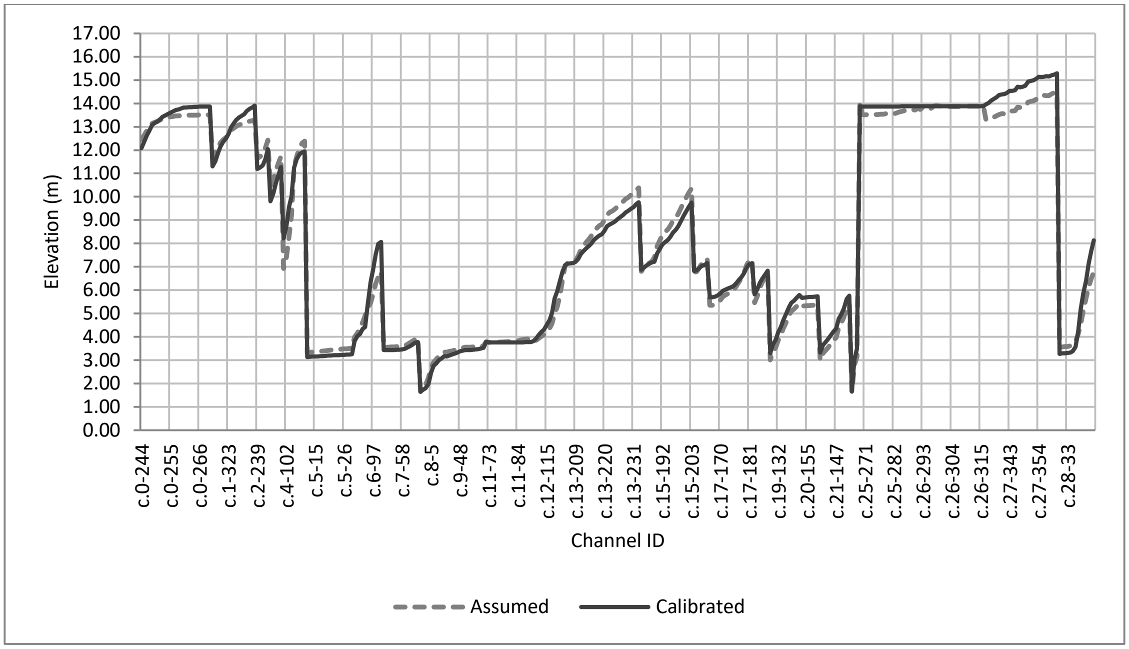


Figure 6.19 Assumed and calibrated water surface profiles for 29-channel synthetic case

The assumed and calibrated water surface profiles are given in Figure 6.19. Generally, the calibrated water surface profile fits well into the assume water surface profile. The assumed and calibrated Manning's values for 29 channels are given in Table 6.7. The relative similarity (Eq. 6.16) between the assumed and calibrated Manning's value sets is 0.877.

Table 6.7 Assumed and calibrated Manning's values for 29-channel synthetic case

Channel ID	Assumed	Calibrated	Channel ID	Assumed	Calibrated
0	0.042	0.042	15	0.042	0.040
1	0.036	0.040	16	0.040	0.042
2	0.042	0.040	17	0.044	0.042
3	0.040	0.042	18	0.040	0.040
4	0.044	0.038	19	0.038	0.040

5	0.040	0.040	20	0.038	0.040
6	0.040	0.038	21	0.040	0.042
7	0.044	0.040	22	0.040	0.042
8	0.044	0.040	23	0.038	0.038
9	0.040	0.042	24	0.038	0.038
10	0.038	0.040	25	0.040	0.040
11	0.040	0.042	26	0.044	0.042
12	0.040	0.038	27	0.036	0.038
13	0.038	0.040	28	0.040	0.038
14	0.042	0.038			

In summary, the application of SA in calibrating the Manning's roughness coefficients for the synthetic cases showed a good agreement between assumed and calibrated Manning's roughness coefficients for different systems of channel configuration.

6.6.6.4. Calibration

The application of SA in calibrating Manning's roughness coefficients is carried out for a channel network in Quang Nam province to demonstrate the practical use of SA in calibrating the roughness coefficients for a basin scale problem. The channel network in this study has 29 channels with 363 cross-sections.

To optimize the Manning roughness coefficient for Quang Nam basin channel system, 150 survey data points of water elevations of the event during November 26th to December 5th 2009 were used. These data points were used to interpolate an observed water surface. An average water elevation for every cross-section was extracted from this surface and used as observed water elevation at the cross-section. For the boundary conditions, the peak discharge rate at Nong Son, B (9,000 m^3sec^{-1}), and Thanh My, A (2,710 m^3sec^{-1}) were used. Discharge values at Song Kon (F) and Song Boung (E) were estimated from the discharge rate of the nearby station, Thanh My, based on the catchment areas. The water level at the downstream boundary condition was 0.87 m. The following parameters for the Simulated Annealing process were used: 1000 for initial temperature and 250 for β to control the cooling rate of the process. During the optimization, roughness coefficients were kept within the feasible range of 0.03 to 0.05 (Prasuhn 1992). The result of the optimization is given in Figure 6.20. The process froze at an RMSE of approximately 0.51 m, at solution 736.

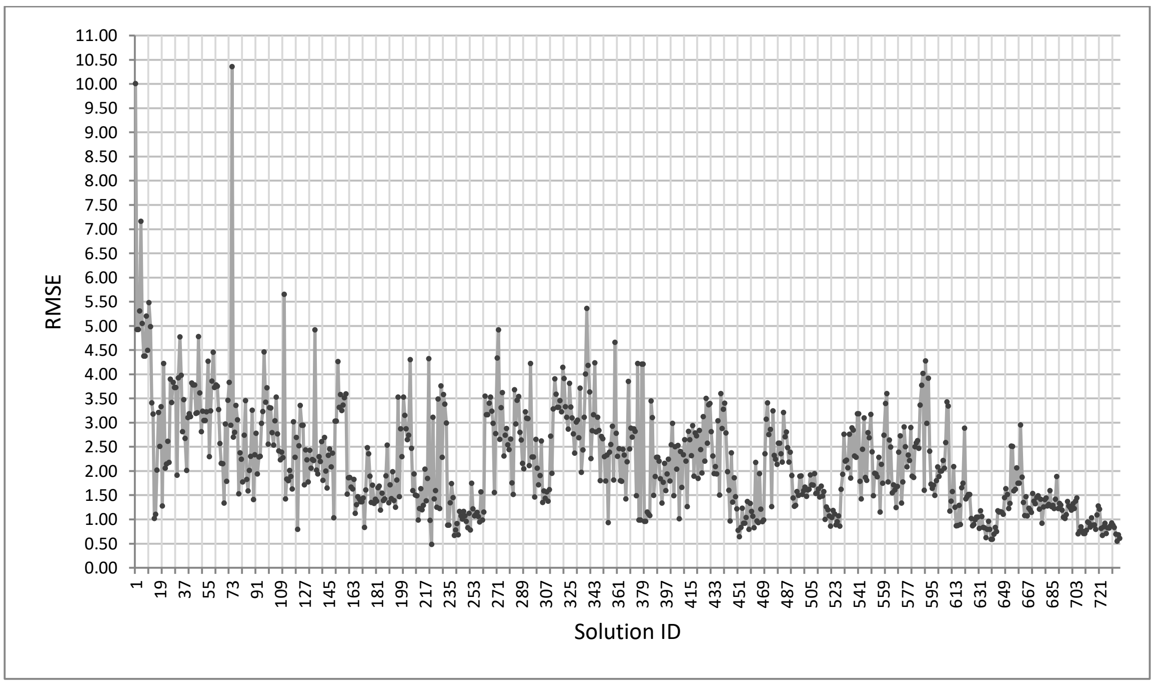


Figure 6.20 Simulated Annealing solutions for calibrating the Manning's roughness coefficients for Quang Nam basin

The best solution was found at solution 222, RMSE = 0.48 m. The "optimized" Manning's values for each channel in the network is given in Table 6.8.

Table 6.8 The best calibrated Manning's roughness coefficients for Quang Nam channel network

Channel ID	Manning's value	Channel ID	Manning's value
0	0.0435	15	0.0418
1	0.0361	16	0.0402
2	0.0418	17	0.0426
3	0.0402	18	0.0410
4	0.0426	19	0.0394
5	0.0410	20	0.0394
6	0.0410	21	0.0402
7	0.0426	22	0.0402
8	0.0426	23	0.0385
9	0.0410	24	0.0369
10	0.0394	25	0.0402
11	0.0410	26	0.0426
12	0.0402	27	0.0353
13	0.0394	28	0.0402
14	0.0418		

Observed and simulated water elevations for all the cross-section are given in Figure 6.21. Generally, the simulation gives good agreement between observed and simulated water elevations. The maximum discrepancy of 1.7 m happened at cross-section c.25-274.

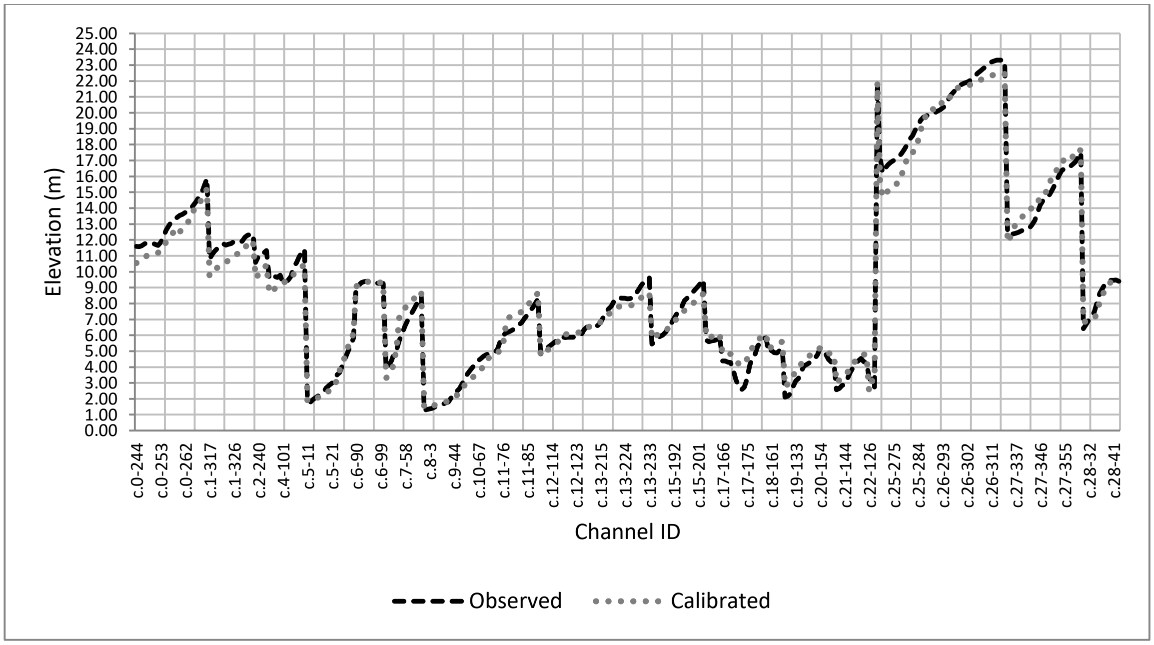


Figure 6. 21 Simulated and observed water elevations for the event in 2009 in Quang Nam basin

6.6.6.5. Validation

Using the Manning's roughness coefficients from the calibration, with the boundary conditions of the event in 1998, 1999, 2007, 2008, the simulations of the flood inundation were carried out. Figures 6.22 to 6.25 show the inundation maps of the corresponding events.

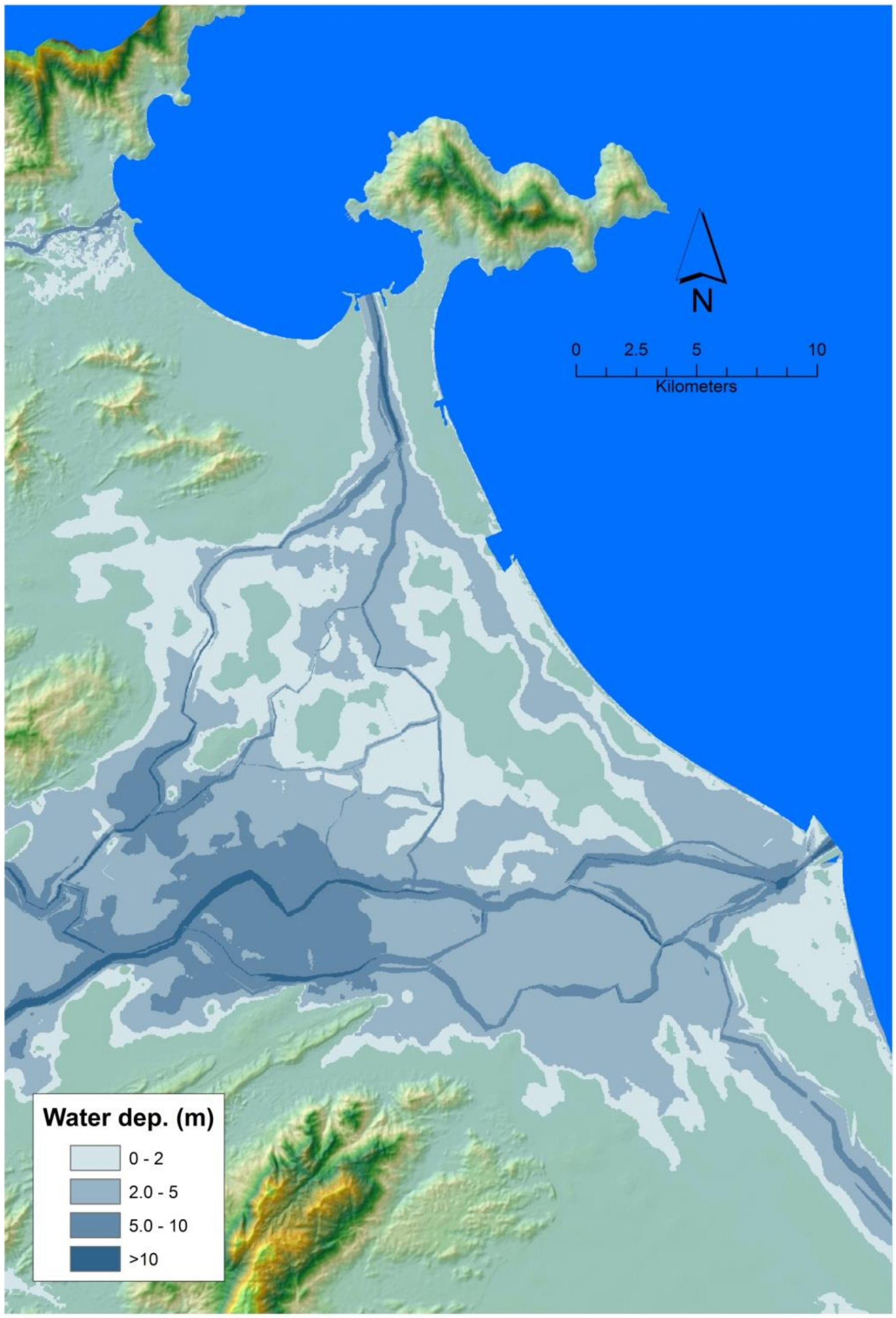


Figure 6.22 Flood inundation for the event in 1998 in Quang Nam basin

Figure 6.23 Flood inundation for the event in 1999 in Quang Nam basin

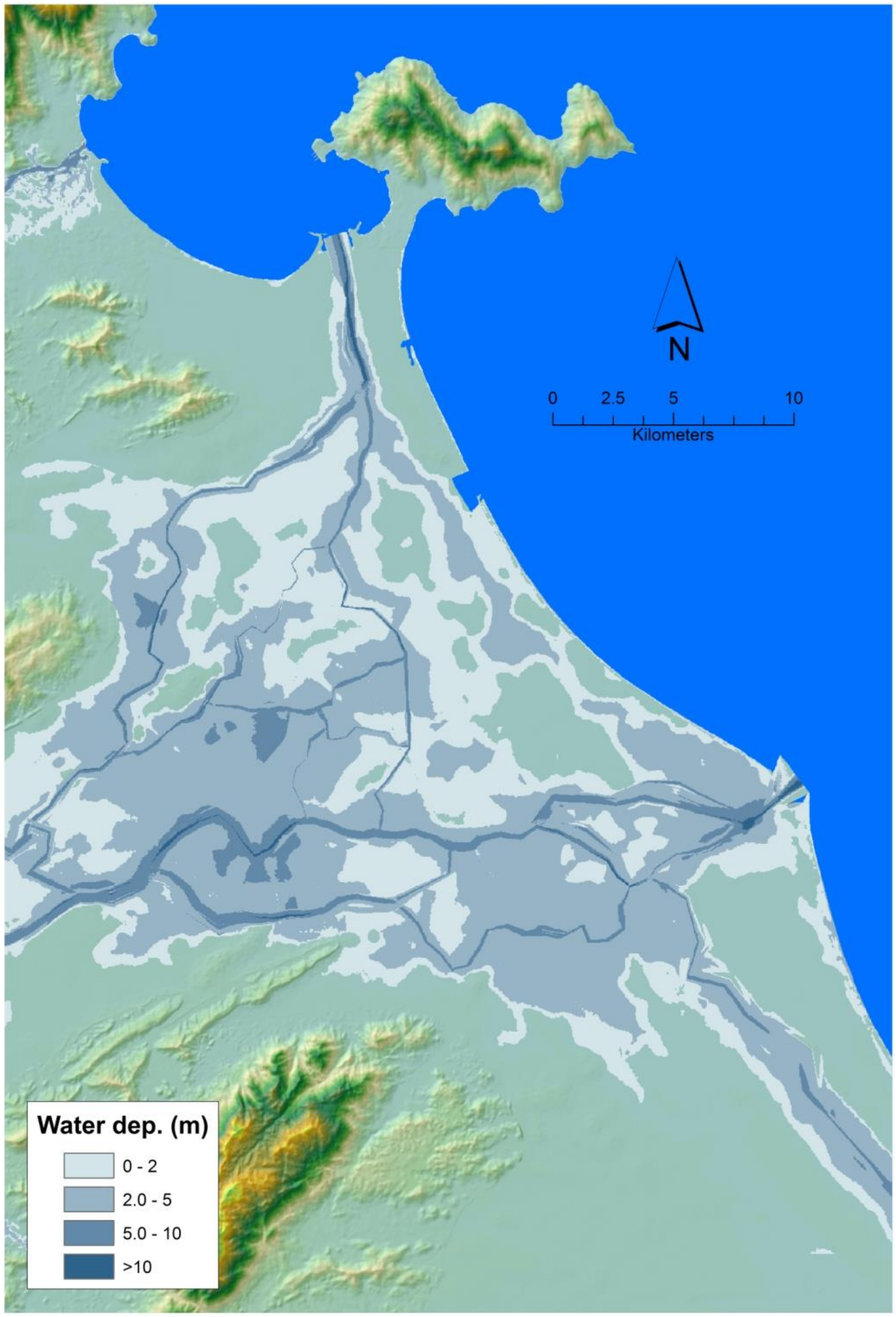


Figure 6.24 Flood inundation for the event in 2007 in Quang Nam basin

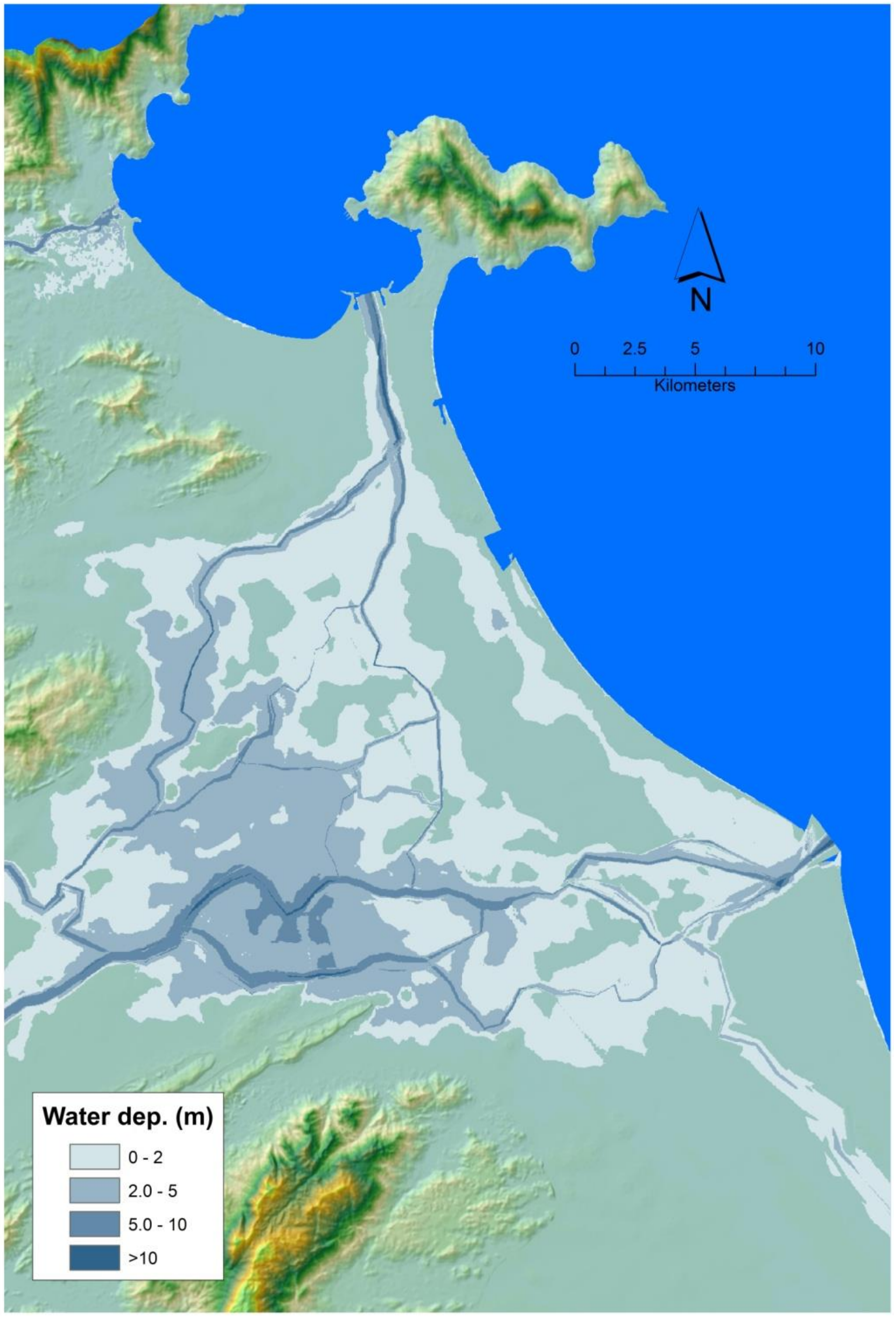


Figure 6.25 Flood inundation for the event in 2008 in Quang Nam basin

To validate the simulation results, water elevations of these events at each cross-section in the channel network in Quang Nam province were calculated. The corresponding values at the five observed stations were used to compare with the simulated water elevations. The simulated water elevations at the above mentioned five stations are given in Table 6.9 and Figure 6.26.

Table 6. 9 Water level validation result for the events in 1998, 1999, 2007, 2008 for Quang Nam basin

	Simulated (m)	Observed (m)	Difference (m)
1998			
Ai Nghia	11.63	10.37	1.26
Giao Thuy	10.39	9.41	0.98
Cau Lau	6.51	5.09	1.42
Hoi An	4.69	3.31	1.38
Cam Le	4.60	2.99	1.61
1999			
Ai Nghia	11.48	9.43	2.05
Giao Thuy	10.06	8.90	1.16
Cau Lau	5.88	4.54	1.34
Hoi An	3.25	2.50	0.75
Cam Le	4.10	2.62	1.48
2007			
Ai Nghia	11.51	10.37	1.14
Giao Thuy	10.18	9.41	0.77
Cau Lau	6.38	5.09	1.29
Hoi An	4.61	3.31	1.30
Cam Le	4.53	2.99	1.54
2008			
Ai Nghia	9.23	9.10	0.13
Giao Thuy	9.01	8.81	0.20
Cau Lau	5.24	3.94	1.30
Hoi An	1.80	1.15	0.65
Cam Le	3.65	2.16	1.49

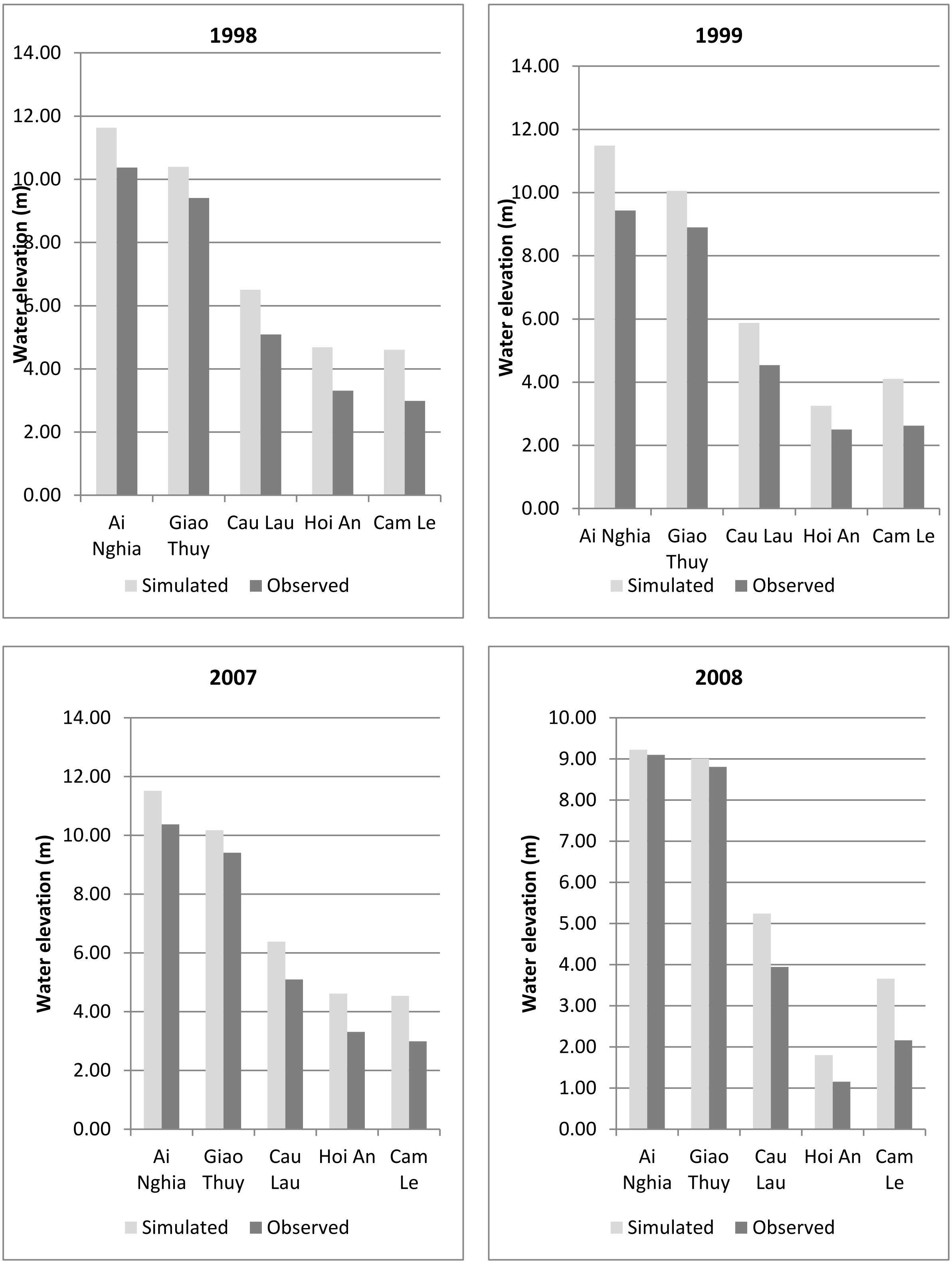


Figure 6. 26 Water level validation results for flood events in 1998, 1999, 2007, 2008 in Quang Nam basin

It can be seen that, the discrepancies in the water elevations between observed and simulated values at 5 observed stations within the flood plain are within a reasonable range (see Table 6.9). On a basin scale, the simulation result is practical.

6.7. Conclusion

The study shows that, tight coupling between a hydraulic model and a GIS is possible. This demonstrates that spatial, physical phenomena can be modelled and simulated in a GIS platform. This makes the simulation results more communicative and visually appealing.

A channel network can be considered as an analogy of a directed graph. This allow for automatically assembly flow equation systems, bringing the simulation of flows in the general channel networks out of the lab settings, opening a new door to software development in this field.

Simulated Annealing can be used to systematically calibrate the Manning's roughness coefficients for a general channel network of a basin scale. This new approach makes the flow modelling procedure for channel networks, especially complicated channel network, less laborious and the requirement for the knowledge of the roughness properties of specific basins is not necessary.

The case study in Quang Nam province of Vietnam shows that the simulation result is reasonable and practical. It could be used for water level prediction and flood plain modelling for flood hazard management in the basin.

Due to the fact that, the theory behind the tool set is based on the gradually varied flow (GVF) equations, although it performs well in estimating the inundation boundary for the basin of Quang Nam, it is limited in the sense that it does not provide inundation estimation for the unsteady state scenario. To make the tool set robust, future development of the tool set should integrate the unsteady state flow model to address this limitation.

Chapter 7. FREQUENCY ANALYSIS FOR QUANG NAM BASIN MAXIMUM ANNUAL FLOWS

7.1. Introduction

The main objective of this component is to design and integrate into the SDSS a tool set that allows decision makers and planners to calculate the frequencies of flood events corresponding to different return periods. This information has the important implications in flood management such as in developing flood mitigation and adaptation plans. In specific situations we want to know how to communicate with local people about the magnitude of a coming event in an intuitive language, perhaps by comparing the event with a similar event in the history based on the return period or the magnitude, so that people can prepare themselves appropriately; we want to know how the flood plain of 100 year event looks like so that decision makers/planners can have specific information for carrying out land use and residential development planning sustainably. In specific situations we may want to know the magnitude of a n-year return period event so that structural works that can be built to withstand such events and protect the community and their agricultural/commercial/industrial activities. This chapter will provides information on the frequency analysis for extreme flows in Nong Son and Thanh My stations within the basin.

7.2. Methodology

The toolset is developed to provide information about the distribution of extreme events using common statistical distributions for extreme events. Specifically, the toolset was developed to carry out frequency analysis by fitting the historical data into three distributions: Normal, Gumbel and Generalized Extreme Event distributions (GEV). The toolset also provides tools to help with performing the Kolmogorov – Smirnov test for the goodness of fit (GoF). The following section discusses the techniques and numerical approximations used to fit the data into the above mentioned distributions and from these fitted models, to estimate the magnitude of events with certain return periods. Technical aspects of Kolmogorov – Smirnov test are also discussed.

7.2.1. Normal distribution fitting

A normal distribution has the following probability density function (PDF):

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \; for \; \infty < x < +\infty \qquad 7.1$$

The mean of the distribution, μ, and the standard deviation of the distribution σ are the two parameters defining a specific normal distribution. Therefore, the distribution is commonly denoted as *N(μ,σ)*. For convenience, a standard normal distribution N(0, 1) is often used instead of the normal distribution N(μ,σ). N(0, 1) has the following PDF.

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}} \; for \; \infty < z < +\infty \qquad 7.2$$

All the calculations have to be done with z-scores $z = \frac{x-\mu}{\sigma}$ instead of directly with x.

The cumulative distribution function for a standard normal random variable is denoted as follows:

$$F(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-t^2/2} \, dt \qquad 7.3$$

This function gives the cumulative probability P(z < Z) of a standard normal random variable; *t* is a dummy variable. Estimations of the parameters of the model from a data series are:

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i \qquad 7.4$$

and

$$\hat{\sigma} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_{i-}\hat{\mu})^2} \qquad 7.5$$

II.1.1. Cumulative normal distribution function approximation

$f(z) = \frac{1}{\sqrt{2\pi}}e^{-\frac{z^2}{2}}$ is the standard normal probability density function. Let

$$F(z) = P(Z > z) = \int_z^\infty \frac{1}{\sqrt{2\pi}}e^{-\frac{t^2}{2}}dt \qquad 7.6$$

For z > 0, F(z) can be approximated as (Abramowitz and Stegun, 1964):

$$F(z) = f(z)(b_1t + b_2t^2 + b_3t^3 + b_4t^4 + b_5t^5) + \epsilon(z) \qquad 7.7$$

Where

$$t = \frac{1}{1+rz}, r = 0.2316419 \qquad 7.8$$

b1 = 0.31938153	b3 = 1.781477937	b5 = 1.330274429
b2 = -0.356563782	b4 = -1.821255978	

The accuracy of the approximation is: $\epsilon(p) < 7.5.10^{-8}$. This approximation can be used to calculate the exceedance probability and return period of an extreme event given the event magnitude.

7.1.2. Invert CDF function approximation

The problem here is given a p-value (P(Z > z)), find a z value that satisfies (7.6). Numerically, the invert function of the CDF of the standard normal distribution is computed by an approximation by Abramowitz and Stegun (1964)

$$z_p = t - \frac{c_0 + c_1t + c_2t^2}{1 + d_1t + d_2t^2 + d_3t^3} + \epsilon(p) \qquad 7.9$$

Where,

$$t = \sqrt{ln\frac{1}{p^2}} \qquad 7.10$$

c0 = 2.515517	d1 = 1.432788
c1 = 0.802853	d2 = 0.189269
c2 = 0.010328	d3 = 0.001308

The accuracy of the approximation is: $\epsilon(p) < 4.5.10^{-4}$

This approximation allows us to calculate the magnitude of an event given an exceedance probability or a return period t.

7.2.2. Gumbel distribution fitting

The Gumbel distribution, also known as an Extreme Value type I (EV1) distribution, has the following cumulative distribution function:

$$F(x) = e^{-e^{(-\frac{x-\xi}{\alpha})}} \quad 7.11$$

And probability density function:

$$f(x) = \alpha^{-1}e^{(-\frac{x-\xi}{\alpha})}e^{-e^{(-\frac{x-\xi}{\alpha})}} \quad 7.12$$

where μ is the location parameter and α is the scale parameter of the distribution. The parameters of the distribution function can be estimated as follows:

Scale parameter: $\alpha = \sqrt{6}\sigma/\pi$

Location parameter: $\xi = \mu - 0.557$

Then the magnitude of an extreme event with a return period of T can be calculated as:

$$Q_T = \xi + \alpha y \quad 7.13$$

Where, $y = -\ln\left(-\ln\left(1 - \frac{1}{T}\right)\right)$.

7.2.3. General Extreme Value (GEV) distribution fitting

The GEV distribution has the following probability density function (Hosking and Wallis, 1997):

$$f(x) = \frac{1}{\sigma}\left(1 + \xi(\frac{x-\mu}{\sigma})\right)^{(-1/\xi)-1} . e^{-\left(1+\xi(\frac{x-\mu}{\sigma})\right)^{-1/\xi}} \qquad 7.14$$

The cumulative distribution function of GEV is:

$$F(x) = e^{-\left(1+\xi\frac{x-\mu}{\sigma}\right)^{-1/\xi}} \qquad 7.15$$

The distribution has the following parameters:

$$k = 7.8590c + 2.9554c^2 \qquad 7.16$$

In which: $c = \frac{2}{3+\tau_3} - \frac{ln2}{ln3}$

$$\alpha = \frac{\lambda_2 k}{(1-2^{-k})\Gamma(1+k)} \qquad 7.17$$

$$\xi = \lambda_1 - \frac{\alpha\left(1-\Gamma(1+k)\right)}{k} \qquad 7.18$$

Where Γ is the Gamma function, which is $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$. In this tool set development, the Gamma function is calculated by using the *gammafunction* function in ALGLIB function library by Sergey (1999).

The estimation of the parameters of the distribution is done by an L-Moment method (Cunnane, 1989). The method is based on the probability weighted moments (PWMs). L-Moments use weighted linear combinations of data that have been arranged in ascending order. The method is less sensitive to data outliners and is nearly unbiased (Rowinski, 2001).

$$M_1 = \frac{1}{N}\sum_{i=1}^{N} Q_i \qquad 7.18$$

where, $$M_2 = \frac{1}{N}\sum_{i=1}^{N} \frac{i-1}{N-1} Q_i \quad 7.19$$

$$M_3 = \frac{1}{N}\sum_{i=1}^{N} \frac{(i-1)(i-2)}{(N-1)(N-2)} Q_i \quad 7.20$$

$$M_4 = \frac{1}{N}\sum_{i=1}^{N} \frac{(i-1)(i-2)(i-3)}{(N-1)(N-2)(N-3)} Q_i \quad 7.21$$

where N is the sample size, Q is the data value, and i is the ranking in ascending order. From this, L-Moments equations are defined as (Cunnane 1989):

$$\lambda_1 = L1 = M_1 \quad 7.22$$

$$\lambda_2 = L2 = 2M_2 - M_1 \quad 7.23$$

$$\lambda_3 = L3 = 6M_3 - 6M_2 + M_1 \quad 7.24$$

$$\lambda_4 = L4 = 20M_4 - 30M_3 + 12M_2 - M_1 \quad 7.25$$

4 L-Moments λ_1, λ_2, λ_3, λ_4 are used to derived other useful ratios which are L-CV (τ_2), L-skewness (τ_3), and L-Kurtosis (τ_4):

$$\tau_2 = \frac{L2}{L1}, \tau_3 = \frac{L3}{L2}, \tau_4 = \frac{L4}{L2} \quad 7.26$$

τ_2 is similar to the normal coefficient of variation (CV) measuring how the data set varies from the mean. The larger τ_2 value, the larger the variation. τ_3 measures the lack of symmetry in the distribution. A negative value indicates a long left tail of the probability density function, while a positive value indicates a long right tail. In the GEV, a positive τ_3 is expected as the extreme events occur in the right tail of the distribution. τ_4 is described as a measurement of "peakedness" of the distribution (Hosking and Wallis, 1997).

When all the model parameters have been estimated, the magnitude of an event with a return period of T can be calculated as:

$$Q_T = \xi + \left(\frac{\alpha}{k}\right)\left\{1 - \left(-\log\left(\frac{T-1}{T}\right)\right)^k\right\} \quad 7.27$$

7.2.4. Kolmogorov - Smirnov test for goodness of fit

The test consists of measuring the largest difference between the theoretical frequency, $F(x)$, and empirical frequency, $\hat{F}(x)$. It verifies whether an experimental distribution can be considered as identical to a reference one. The null and alternative hypothesis of the test are:

$$H0: \hat{F}(x) = F(x) \; \forall x$$

$$H1: \hat{F}(x) \neq F(x)$$

If the sample of interest, n values of x_i, is sorted in ascending order, each observation of rank order r has a corresponding empirical frequency of $\hat{F}(x_r) = n/r$. The statistics d^+ is defined as follows (Sachs, 1984):

$$d^+ = \max\left[\frac{1}{n} - F(x_1), \frac{2}{n} - F(x_2), \ldots, \frac{n}{n} - F(x_n)\right] \quad 7.28$$

And the statistic $d^-$is

$$d^- = \max\left[F(x_1) - \frac{1-1}{n}, F(x_2) - \frac{2-1}{n}, \ldots, F(x_n) - \frac{n-1}{n}\right] \quad 7.29$$

For a bilateral test the test statistic d is the maximum of d^- and d^+ thus $d = \max\{d^-, d^+\}$. The critical value for a sample of n > 35, given by Sachs (1984), is $1.358/\sqrt{n}$.

7.3. Results and discussion

7.3.1. Time series data

In Quang Nam basin, flow data are only available at two stations which are Nong Son and Thanh My. The data used for this analysis is a time series of flow data of 6 hourly measured at the two stations for a period of 37 years. The data were provided by the Department of Science and Technology of Quang Nam province. This is a very long series of data entries. The entries were imported into an Access database table named "Q" having the following schema: Q{Station, Time, Q}.

To conduct the frequency analysis and to ensure the independence constraint in the data series to be statistically analysed, the maximum annual series were extracted from this series (Haan, 1977). The frequency analysis tool developed for this study allows for quick extraction the maximum annual series from the Q table in the database. The maximum annual series extracted from the database for Nong Son and Thanh My stations for the interval from 1976 to 2012 is shown in table 7.1.

Table 7.1 Maximun annual flows for Nong Son and Thanh My stations

Year	NongSon Max Q ($m^3.sec^{-1}$)	Thanh My Max Q ($m^3.sec^{-1}$)	Year	NongSon Max Q ($m^3.sec^{-1}$)	Thanh My Max Q ($m^3.sec^{-1}$)
1976	4308	832	1995	5130	2030
1977	3390	1210	1996	6790	3870
1978	2320	836	1997	5890	1960
1979	2500	1270	1998	8920	4440
1980	6270	3280	1999	8560	3666
1981	4540	3160	2000	5620	2800
1982	1760	1440	2001	4760	1790
1983	6270	3410	2002	2200	1254
1984	4680	2800	2003	5630	2160
1985	4590	1140	2004	8260	2900
1986	6730	1730	2005	4810	1640
1987	2120	875	2006	4600	2310
1988	3200	2200	2007	8410	3880
1989	2280	1090	2008	5620	1870
1990	6760	3420	2009	7030	4540

1991	3490	1380	2010	6570	1610
1992	5360	2210	2011	6730	2660
1993	3360	1230	2012	1750	614
1994	3650	901			

7.3.2. Frequency analysis tool set development

The tool set for frequency analysis was developed using the VB.Net language and was integrated into the SDSS as a tool running inside of ArcMap. It allows for performing Normal, Gumbel and GEV distribution fittings and Kolmogorov – Smirnov hypothesis test for goodness of fit. The tool also allows data to be loaded from the database, extracting the maximum annual series via SQL language. The theory behind the tool is provided in the methodology section above and the detail implementation of the tool is given in appendix 7.3 (only the main parts of the code are given in the appendix)

7.3.3. Frequency analysis

7.3.3.1. Nong Son station maximum annual flow

Table 7.2, 7.3 and fig. 7.1 show the result of the frequency analysis for Nong Son station. Visually, the data series fit very well into each of the 3 selected distributions, though the Normal distribution and GEV distribution outperformed the Gumbel distribution. Toward longer return periods, the Gumbel distribution diverged more from the empirical measures.

Table 7.2 Parameters of Normal, Gumbel and GEV distributions for maximum annual flow at Nong Son station

```
------------------------------------------------------------------
Normal distribution: N~( M = 4996.160, Sig = 2016.400)
------------------------------------------------------------------
Gumbel distribution: G~( Alfa = 1572.182, Beta = 4120.457)
------------------------------------------------------------------
GEV distribution: G~( Alfa = 2025.840, K = 0.252, Psi = 4240.120)
------------------------------------------------------------------
```

Table 7.3 Nong Son station frequency analysis result

Return period (year)	Gumbel Q m³.sec-1	Norm. Q m³.sec-1	GEV. Q m³.sec-1
2	4696	4996	4949
5	6478	6693	6770
10	7658	7580	7720
20	8790	8312	8477
25	9149	8526	8690
50	10255	9137	9273
100	11352	9687	9759
200	12446	10190	10165

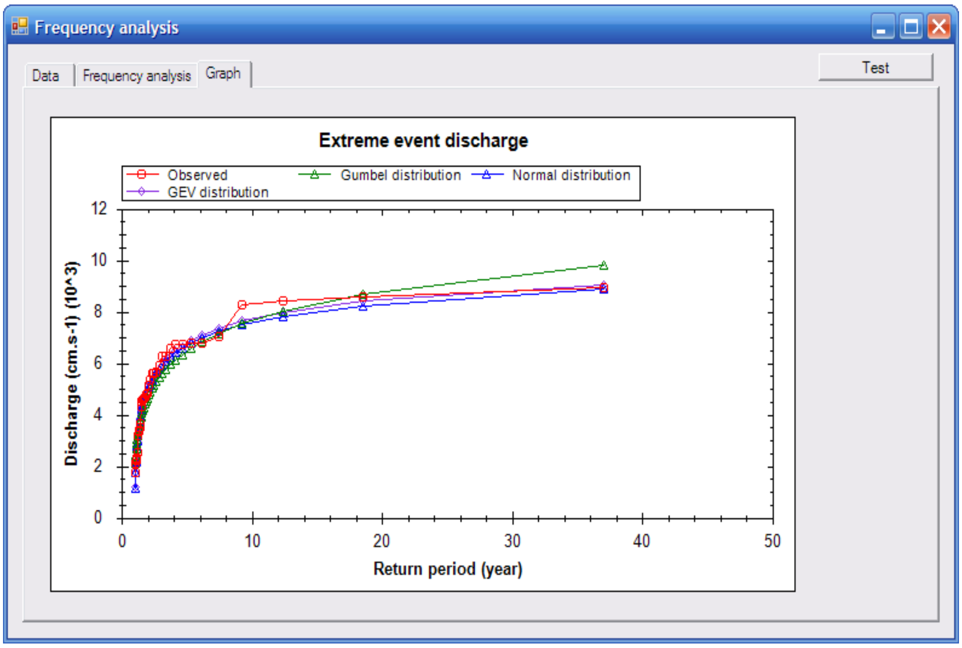


Figure 7.1 Nong Son maximum annual flow distribution fitting

The hypothesis testing for the three distributions (null hypothesis H_o: the data is conformal to the selected distribution and alternative hypothesis H_1: the data is not conformal to the selected distribution) was carried out using Kolmogorov – Smirnov test. The results of the tests are given in Table 7.4. Detailed results of the test are given in appendix 7.1. The results show that Kolmogorov – Smirnov test does not have enough evidence to reject any of the distributions.

Table 7.4 Hypothesis test for distribution of maximum annual flow in Nong Son station

```
-------------------------------------------------------------------------------------
Hypothesis test
Ho: The sample data is normal distributed N~( M = 4996.16, Sig = 2016.40)
H1: The sample data is not normal distributed
Kolmogorov - Smirnov test statistic D = 0.0813
With Alfa = 5% significance level, D.05 = 0.2236
There is not enough evidence to reject Ho
-------------------------------------------------------------------------------------
Hypothesis test
Ho: The sample data is Gumbel distributed G~( Alfa = 1572.182, Beta = 4120.457)
H1: The sample data is not Gumbel distributed
Kolmogorov - Smirnov test statistic D = 0.1286
With Alfa = 5% significance level, D.05 = 0.2236
There is not enough evidence to reject Ho
-------------------------------------------------------------------------------------
Hypothesis test
Ho: The sample data is GEV distributed G~( Alfa = 2025.840, K = 0.252, Psi = 4240.120)
H1: The sample data is not GEV distributed
Kolmogorov - Smirnov test statistic D = 0.1559
With Alfa = 5% significance level, D.05 = 0.2236
There is not enough evidence to reject Ho
-------------------------------------------------------------------------------------
```

7.3.3.2. Thanh My station maximum annual flow

For Thanh My station, table 7.5, 7.6 and fig. 7.2 show the result of the frequency analysis. Visually, the data series fit very well into each of the 3 selected distributions. Visually, none of the distributions seems to significantly outperform others although Gumbel and GEV distributions seem to be closer to the empirical distribution.

Table 7.5 Parameters of Normal, Gumbel and GEV distributions for maximum annual flow at Thanh My station

```
----------------------------------------------------------------
Normal distribution: N~( M = 2173.19, Sig = 1092.02)
----------------------------------------------------------------
Gumbel distribution: G~( Alfa = 851.446, Beta = 1698.934)
----------------------------------------------------------------
GEV distribution: G~( Alfa = 919.158, K = 0.021, Psi = 1661.298)
----------------------------------------------------------------
```

Table 7.6 Thanh My station frequency analysis result

Return period (year)	Gumbel Q m3.sec-1	Norm. Q m3.sec-1	GEV. Q m3.sec-1
2	2010	2173	1996
5	2976	3092	3018
10	3615	3572	3681
20	4227	3969	4308
25	4422	4085	4505
50	5021	4415	5105
100	5615	4713	5692
200	6208	4986	6269

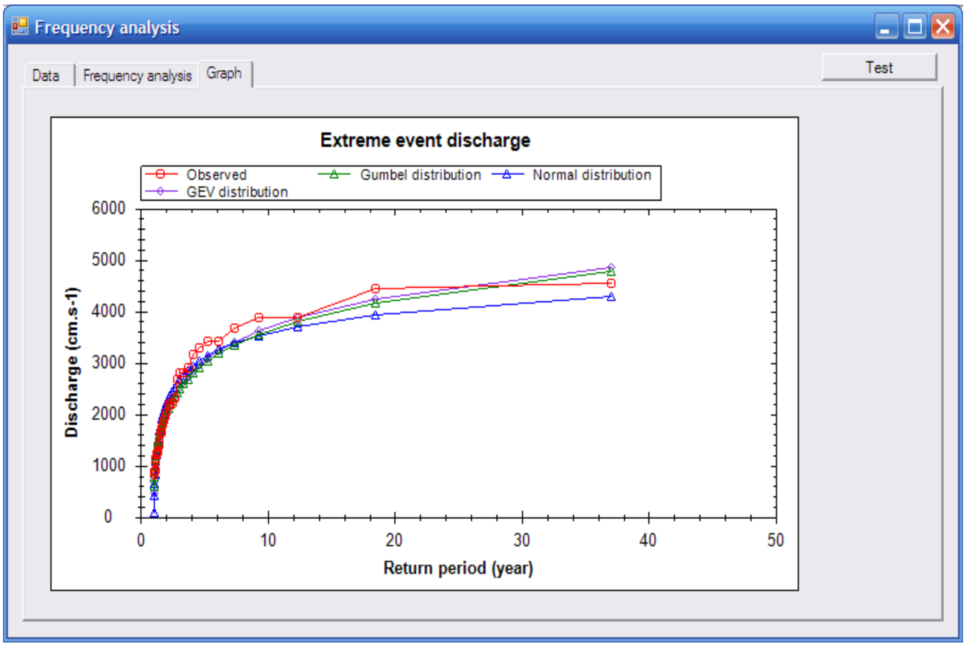


Figure 7.2 Thanh My maximum annual flow distribution fitting

The results of the Kolmogorov – Smirnov tests are given in the following listing. Detailed results of the test are given in appendix 7.2. The results show that Kolmogorov – Smirnov test does not have enough evidence to reject any of the distributions.

Table 7.7 Hypothesis test for distribution of maximum annual flow in Thanh My station

```
-----------------------------------------------------------------------------------
Hypothesis test
Ho: The sample data is normal distributed N~( M = 2173.19, Sig = 1092.02)
H1: The sample data is not normal distributed
Kolmogorov - Smirnov test statistic D = 0.1082
With Alfa = 5% significance level, D.05 = 0.2236
There is not enough evidence to reject Ho
-----------------------------------------------------------------------------------
Hypothesis test
Ho: The sample data is Gumbel distributed G~( Alfa = 851.446, Beta = 1698.934)
H1: The sample data is not Gumbel distributed
Kolmogorov - Smirnov test statistic D = 0.1062
With Alfa = 5% significance level, D.05 = 0.2236
There is not enough evidence to reject Ho
-----------------------------------------------------------------------------------
Hypothesis test
Ho: The sample data is GEV distributed G~( Alfa = 919.158, K = 0.021, Psi = 1661.298)
H1: The sample data is not GEV distributed
Kolmogorov - Smirnov test statistic D = 0.0815
```

```
With Alfa = 5% significance level, D.05 = 0.2236
There is not enough evidence to reject Ho
-------------------------------------------------------------------------------------
```

The implication then is that, for Quang Nam basin, maximum annual flows at Nong Son and Thanh My stations could be modelled using any one of the distributions: Normal, Gumber or GEV distributions. Kolmogorov – Smirnov tests did not have enough evidence to reject any of the distributions.

7.3. 3. Use case demonstration

With the information about the extreme flows generated by the frequency analysis toolset, estimations of inundation can be made. In this section, procedures for generating an inundation map for the study area are demonstrated.

In this synthetic case, an inundation map resulting from extreme flows of the 100 year return period will be generated. From the above frequency analysis results for Nong Son and Thanh My stations, the extreme flows for a return period of 100 years are 11,352 $cm.sec^{-1}$ for Nong Son and 5,615 $cmsec^{-1}$ for Thanh My station (from the Gumbel distribution). Using the lower boundary conditions of the flood event during 1-10th November 2007 and the calibrated Manning's roughness coefficients given in chapter 6, the hydraulic model was run. The boundary conditions of the run are given in table 7.8.

Table 7.8 Boundary conditions for the extreme flow use case

Station	Flows of 100 years return period (Gumbel distribution) ($cm.sec^{-1}$)	Lower boundary condition (m)
Nong Son	11,352	-
Thanh My	5,615	-
Son Tra	-	0.94
Cua Dai	-	0.94

Figure 7.3 shows the flood inundation map generated by boundary conditions of a 100 year return period flows at Nong Son and Thanh My station in comparison with the inundation extent of 2007 event.

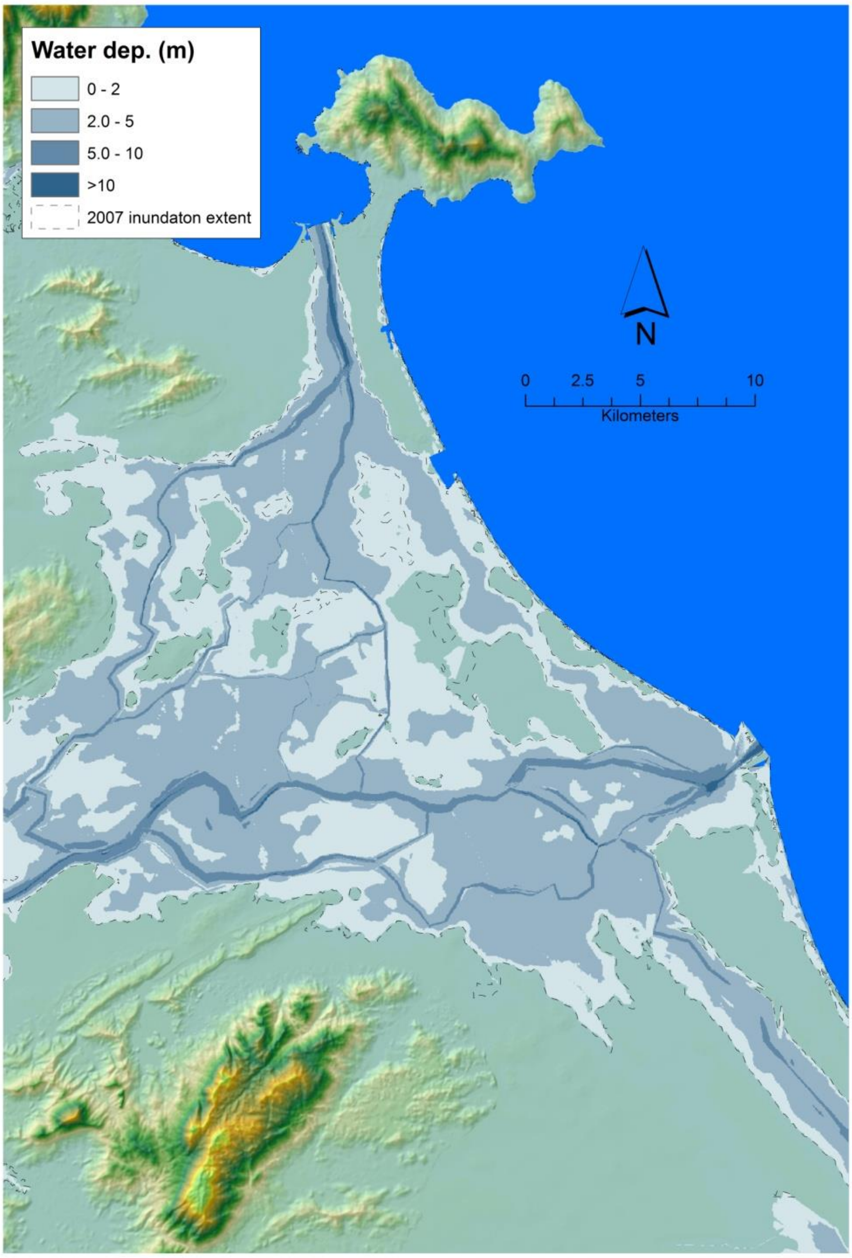


Figure 7.3 Estimation of inundation caused by a 100 year return period flow in Quang Nam basin

There is no formal way to validate the accuracy of this inundation map because data for such a validation is not available. However, extreme flows of 100 year return period were selected for the demonstration because the flows at Nong Son and Thanh My for the event during 1-10th November 2007 are relatively close to the corresponding flows of 100 year return period for Gumbel distribution. This creates a chance to compare the inundation of the two cases.

By visually comparing the inundation map of the event during 1-10th November 2007 (see Fig. 6.24 in chapter 6) with this 100 year return period inundation map generated in this synthetic case, it can be seen that the two inundation extents show a high degree of agreement.

This suggests that the information from frequency analysis results can be used for generating inundation maps which, in turn, can be used as a reference or an input for planning purposes. However, the use cases are not limited to what discussed in this sections. The toolset can be used in many other situations as desired by planners, decision makers.

Although Normal, Gumbel and GEV distributions seems to fit well with the extreme event data for the Quang Nam basin, to expand the scope of applicability of the tool set to make it general, other distributions such as the Log-normal distribution (LN), which can be used for the hydrological processes with positive skewness in the data (Vogel and Wilson 1996), Log Pearson type-3 (LP3) distribution, which is also popular in flood-frequency analysis (Huynh and Thambirajah 1984), should be integrated.

Chapter 8. Conclusion

The SDSS for Quang Nam basin was developed as an extension to the functionality of a general GIS platform (ArcGIS). The VisualBasic.Net programming language was used to develop tools and integrate them into the platform. The decision support system implements tools that help decision makers and planners to carry out estimations of physical characteristics of flood phenomenon within the basin of Quang Nam.

This approach brings about a number of appealing features for the SDSS. It makes use of the advantage of the visual presentation tier of a generic GIS thus making the simulation results more communicative. The tools were designed with the capability to transform the simulation results directly and conveniently into maps. This removes the burden of post processing steps where intermediate procedures are conducted to map the simulated attributes into meaningful information such as maps, graphs and attribute tables after modeling. This is useful especially in the context of non-technical personnel requirement as is the case in Quang Nam province.

With the aid of VB.Net and the event driven programming paradigm, the SDSS was developed with a set of graphical user interfaces (GUIs) and the interaction between the users and the SDSS is event driven. This helps to encapsulate the complexity of technical aspects of the decision support algorithms and gives users an intuitive experience. This reduces the cumbersome nature of the complexity in physical modeling procedures, helping planners and decision makers to focus more on the importance of the modeled problems themselves rather than on the technical issues.

The decision support system was developed to assist with the modeling of the physical aspects of the flood management process, which are rainfall and runoff modeling, frequency analysis and hydraulic modeling. These are the most common technical problems occurring in the flood management of a basin.

For the rainfall and runoff problem, the tool set was developed based on the multiple regression technique. The tool set was used to build the discharge estimation model for Nong Son and Thanh My stations. For the Nong Son station, the discharge is estimated from rainfall measures from 4 upstream rain gauge stations; Hiep Duc, Nong Son, Tien Phuoc and Tra My. For the Thanh My station, the discharge was estimated from rainfall measures using 2 upstream rain gauge stations, they are Phuoc Son and Thanh My. The models showed good 'fit' between observed data and model predictions. Significant levels of variability in the predicted values are explained by the observed data values (R^2 = 0.58 for Nong Son, and R^2 = 0.6 for Thanh My). The Nash-Sutcliffe efficiency criteria also validated that the models are useful for estimation of discharge at Nong Son (NSE from 0.594-0.749) and Thanh My (NSE ranges within 0.589-0.851, the exception being the case of 2005 which has an NSE of 0.367). The NSE values indicated that the performance of the models are acceptable (values between 0.0 and 1.0 are generally viewed as acceptable levels of performance (Krause et al. 2005). According to Moriasi et al. (2007), NSE values should exceed 0.5 in order for models to be judged as satisfactory for hydrologic simulation performed on a monthly time step and appropriate relaxing and tightening of the standard to be performed for daily and annual time step evaluation, respectively. The implication then is that, the models can be used for estimating the discharge for Nong Son and Thanh My stations using upstream rain gauge data with a satisfactory level of accuracy.

A frequency analysis tool set was developed and integrated into the decision support system to help with the frequency analyses at Nong Son and Thanh My stations. This toolset was developed to fit historical flow data of extreme flood events within the basin into statistical distributions. Specifically, the tool set can be used to conduct frequency analysis by fitting the historical data into three popular distributions: Normal, Gumbel and Generalized Extreme Event distributions (GEV). The toolset also provides tools for performing the Kolmogorov – Smirnov test for the goodness of fit (GoF). With a time series of 37 years from 1976 to 1994, the frequency analysis for annual maximum flow at Nong Son and Thanh My were performed. For Nong Son station, Normal distribution and GEV distribution outperformed the Gumbel distribution. Toward longer return periods, the Gumbel distribution diverged more from the empirical measures. For Thanh My station, none of the distributions significantly outperformed others although the Gumbel and GEV distributions seem to be closer to the empirical distribution. The Kolmogorov – Smirnov hypothesis test results showed the Normal, Gumbel and GEV distributions significantly (95%) fit the measured data for both of the stations. This implies that any of the above mentioned distributions can be used to fit the frequency of maximum annual flows at Nong Son and Thanh My.

Hydraulic modelling is an important tool set in this SDSS. The tool set was developed to generate representations of flooded areas for different boundary conditions via simulating gradually varied flow (GVF) in a general channel network. In this component, two important methods have been developed as techniques to simultaneously solve systems of flow equations which contribute to the flood simulation literature. They change the way researchers assemble the flow equations and calibrate the Manning's roughness coefficients for the general channel networks.

Firstly, graph theory was used to automatically assemble the flow equations. By treating a general channel network as a directed graph, the channel network can be systematically mapped to a data model which can be used for storing and retrieving channel network structures. With the application of adjacency lists to store a general channel network in computer memory, a general channel network with multiple boundary conditions, multiple loops, split and combined flow configurations, can be systematically modelled. From these, continuity equations and energy balance equations can programmatically be assembled and solved automatically. This method has eliminated the need for manually assembling the flow equations in the solving flow calculations in general channel networks. No matter how complicated a channel network is, no matter how many loops, split and combined junctions a system may have, no matter how many boundary conditions are present, the system of equations can be automatically assembled. With this method, no special network node coding scheme is necessary. It can be seen that the approach opens a new door for software development that brings the simultaneous solution of flow equations out of the lab setting. The simultaneous solution of flow equations does not have to be confined in research experiments with certain types of network configurations of well behaved cross-section geometries.

Secondly, in calibrating the Manning's roughness coefficients for general channel networks, although the trial and error approach is quite popular, it cannot be applied for the calibration of complicated general channel networks as the behavior of the water elevations and discharges is not predictable. In a general channel network, the adjustment of the Manning's roughness coefficients will not only cause changes in discharge and water elevations of the reaches that are subject to such adjustments, but also in the reaches that have not changed in an unmanageable manner. With the introduction of simulated annealing (a global optimization technique) into the calibration of Manning's roughness coefficients for a general channel network the traditional trial and error approach is no longer necessary. This approach has proven to be practical even for the basin scale channel network of Quang Nam. This innovation can release the researchers from tedious trial and error work and eliminate the need for local knowledge about the roughness characteristics of a specific basin.

The simulation and validation results showed good agreement between simulated and observed data. On a basin scale, the simulation result is practical.

In developing this decision support system, the relevant concept that was adopted is a computer based system that is able to improve the effectiveness of the existing decision making process. This follows a formal definition made by Loucks and daCosta (1991) which is "computer based tools having interactive, graphical, and modelling characteristics to address specific problems and assist individuals in their study and search for a solution to their management problems". This decision support system should be viewed as a system that improves the decision making process not by prescribing particular courses of actions, but by providing data displays, analytical results, and model outputs on the critical information (Jason 2005). It is a system that allows decision-makers to combine personal judgment with computer outputs through a computer interface to produce information for the decision-making process (Slobodan 1999). This means that such systems can assist decision making in different types of problems: structured, semi-structured and unstructured. Therefore, the implication then is that the system should not be viewed as an all in one system that can give answers to all type of questions but only to some questions that may be raised by decision makers or planners. The system is not intended to be comprehensive and can be subjected to further future development and extension to incorporate more functionality.

This decision support system is basin specific. Firstly, when designing this decision support system, the main objective is to develop a set of tools that is utilized by decision makers and planners of the Quang Nam province. Although the algorithms are generic in nature, namely Multiple Regression, Normal distribution, Gumbel distribution, GEV distribution, continuity and energy balance equations, this SDSS may not be suitable for other basins. The selection of algorithms for the integration into the decision support system was based on the performance tests for problems in Quang Nam basin. Therefore, a multiple regression, which is suitable for rainfall runoff problem for the Quang Nam basin, may not suitable for others. Similarly, the proposed distributions, which perform very well for maximum annual flow frequency analysis in Quang Nam, may not apply to others. In addition to this, the system parameters were calibrated specifically for the Quang Nam basin alone. It should not be considered as a generic system that could be customized for any basin by just plugging in new databases and recalibrating the parameters.

From the current status of the SDSS, it is obvious that to make it the system more applicable in broader situations, beyond Quang Nam basin, more effort should be considered to further expand the functionality of the system. Specifically, the following suggestions for future improvement to this SDSS can be made:

- For the frequency analysis tool set, beside the Normal, Gumbel and GEV distribution that have been integrated in the tool set, other distributions such as Log-normal distribution (LN), which can be used for the hydrological processes with positive skewness in the data (Vogel and Wilson 1996), Log Pearson type-3 (LP3) distribution, which is also popular in flood-frequency analysis (Huynh and Thambirajah 1984), should be integrated to make the tool set more robust in broader context;

- In this thesis, the rainfall runoff tool set only considered one specific process of the black box approach, the multiple regression. Although the model performs well for the case study basin, the tool set can be improved by the integration of other models reflecting of quasi-physical or physical processes of the rainfall-runoff phenomenon, and more importantly, the integration of these new models should be undertaken with the development of user friendly calibration tools to provide decision makers and planners with intuitive tools to model the runoff in more situations;
- In this thesis, the steady state flow model was employed to model the flow in the channel network. However, to make the system more comprehensive, the unsteady state flow model could also be integrated. The integration of the unsteady state flow model would allow for more modeling complicated situations where flow parameters are changed with time;
- Besides the application of Simulated Annealing in calibrating the Manning's roughness coefficients for channel networks, other optimization techniques may also be checked to explore more about the suitability and efficiency for this class of problem.

With these improvement, the system would be more applicable for other basins, not just Quang Nam. However, again, for the problem of flood management, applicable models need to be tested, and basin specific parameters need to be calibrated, and fine-tuned, no matter what basin is used, and attribute data and geographic data also need to be prepared.

REFERENCE

Abbott, M.B. and Refsgaard J.C. (Eds.) (1996). *Distributed Hydrological Modelling*. Kluwer Academic Pub.

Akan, A.O., and Yen, B.C., "Diffusion-Wave Flood Routing in Channel Networks," *Journal of the Hydraulics Division*, ASCE, Vol. 107, No. HY6, pp. 719-732, 1981.

Bochkanov S. (1999). *ALGLIB [Internet]*. Nizhny Novgorod, Russia: ALGLIB Project; c1999-2015 [cited 2016 April 30]. Available from: http://www.alglib.net/.

Brimicombe AJ, Bartlett JM (1996). *Linking geographic information systems with hydraulic simulation modelling for flood risk assessment: the Hong Kong approach.* In: Goodchild MF (ed) GIS and environmental modelling. Oxford University Press, New York

Chaudhry M. H. and Schulte A. (1986). Computation of steady state, gradually varied flows in parallel channels. *Can. J. Civ. Eng.* 13(1): 39–45

Chaudhry, M. H. (1993). *Open-Channel Flow.* Prentice Hall, Englewood Cliffs. NJ.

Chaudhry, M.H. and Schulte, A.M. (1986), Computation of Steady State G.V.F. in Parallel Channels, *Canadian Jour. of Civ. Engr*, 13, No. 1, pp. 39—45.

Chaudhry, M.H., *Open-Channel Flow,* Second Edition, Springer, New York, NY, 2007

Clark, M. J. (1998). Putting water in its place: a perspective on GIS in hydrology and water management, *Hydrological Processes*, 12, pp. 823–834.

Cunnane, C. (1989). "Statistical distributions for flood frequency analysis." *Operational Hydrol.* Rep. No.33, WMO-No. 718, World Meteorological Organization, Geneva, Switzerland.

Das, A. (2004). Parameter Estimation for Flow in Open-Channel Networks. *Journal of Irrigation and Drainage Engineering*, 130(2), 160-165. doi:doi:10.1061/(ASCE)0733-9437(2004)130:2(160)

David T. Ford, Member (2001). Flood-warning decision-support system for Sacramento, *Journal of Water Resources Planning and Management/ July/ August*

Department of Natural Resources and Environment, Quang Nam Province (2011). The Vu Gia - Thu Bon Basin. Unpublished manuscript.

Dougherty, D. E., & Marryott, R. A. (1991). Optimal Groundwater Management: 1. Simulated Annealing. *Water Resources Research,* 27(10), 2493-2508. doi:10.1029/91WR01468

Dréo, J., Petrowski, A., Siarry, P., & Taillard, E. (2006). *Metaheuristics for hard optimization: methods and case studies*. Springer Science & Business Media.

ESRI. (2007). ArcGIS Desktop: Release 9.2. Redlands, CA: Environmental Systems Research Institute.

ET&P Srl (1992). The Fuchun River project: a computer based real-time system. Final Report Research Contract CI13-0004-I, EC China Cooperation.

Ezio Todini. (1999). An operational decision support system for food risk mapping, forecasting and management. *Urban Water* 1 (1999) 131-143

Fread, D. L., & Smith, G. G. (1978). Calibration technique for 1-D unsteady flow models. *J. Hydr. Div.*, ASCE, 104(7), 1027-1043.

Ghulam Nabi, H. Reddy, Habib-ur-Rehman. (2012). Gradually Varied Flow Computation in Series, Tree Type and Looped Compound Channel Networks. *Pak. J. Engg. & Appl. Sci.* Vol. 11, July (p. 47-59).

Gonzalez-Abril, L., Gavilan, J. M., & Morente, F. V. (2014). Three Similarity Measures between One-Dimensional Data Sets. *Revista Colombiana de Estadistica,* Volume 37(no. 1), pp.77-92.

Grizzetti, B., F. Bouraoui, and G. De Marsily. 2005. Modelling nitrogen pressure in river basins: A comparison between a statistical approach and the physically-based SWAT model. *Phys. Chem. Earth*, Parts A/B/C 30(8-10): 508-517.

Haan, C. T. (1977). *Statistical methods in hydrology*. Ames: Iowa State University Press.

Hameed, L. K., & Ali, S. T. (2013). Estimating of Manning's Roughness Coefficient for Hilla River through Calibration Using HEC-RAS Model. *Jordan Journal of Civil Engineering*, 7(1).

Honghai Q., Altinakar M.S. A (2011). GIS-based decision support system for integrated flood management under uncertainty with two dimensional numerical simulations. *Environmental Modelling & Software* 26 (2011) 817-821

Hosking, J. R. M. and J. R. Wallis. (1997). *Regional Frequency Analysis.* Cambridge University Press.

Huynh, L.B., Thi, N.V., Long, B.D. and Mai, D.T., 1999. Flood Disaster Study. Disaster Management Unit, UNDP Project VIE/97/002, Hanoi, Vietnam, 302 pp.

Huynh Ngoc Phien, Thambirajah Jiv Ajirajah, Applications of the log Pearson type-3 distribution in hydrology (1984), *Journal of Hydrology,* Volume 73, Issues 3–4, Pages 359-372, ISSN 0022-1694, https://doi.org/10.1016/0022-1694(84)90008-8. http://www.sciencedirect.com/science/article/pii/0022169484900088)

ICEM. (2008). Strategic Environmental Assessment of the Quang Nam Province Hydropower Plan for the Vu Gia - Thu Bon River Basin, Prepared for the ADB, MONRE, MOITT & EVN, Hanoi, Viet Nam. Unpublished manuscript.

Institute of Geography. (2010). Contract for consultants' services to carry out surveys and hydrodynamic flood modelling in the Vu Gia - Thu Bon river basin in Quang Nam province. Unpublished manuscript. (p. 130). Hanoi, Vietnam.

IPCC. (2007). Climate Change 2007: Climate Change Impacts, Adaptation and Vulnerability. Summary for Policymakers. Working Group II Contribution to the

Intergovernmental Panel on Climate Change Fourth Assessment Report. 6th April 2007.

Islam A., Raghuwanshi N., Singh R., and Sen D. (2005). Comparison of Gradually Varied Flow Computation Algorithms for Open-Channel Network. *J. Irrig. Drain Eng.*, 131(5), 457–465.

Jason K. Levy , Chennat Gopalakrishnan & Zhaohui Lin (2005). Advances in Decision Support Systems for Flood Disaster Management: Challenges and Opportunities, *International Journal of Water Resources Development*, 21:4, 593-612

JChampion. (2007). *A flexible charting library for .NET.* Available at: https://www.codeproject.com/articles/5431/a-flexible-charting-library-for-net [Accessed 2 Jun. 2014].

Johnson N. I., Kotz S. and Balakrishnan N. (1994). *Continuous univariate distributions*. John Wiley & sons.

Kidson, R., Richards, K., & Carling, P. (2002). Hydraulic Model Calibration Using a Modern Flood Event: The Mae Chaem River, Thailand. *Proceedings of the PHEFRA Workshop*, Barcelona, 16-19 October 2002, pp. 171-176.

Kirkpatrick S., Gelatt C. D., & Vecchi M. P. (1983). Optimization by Simulated Annealing. *Science*, 220(4598), 671-680. doi:10.1126/science.220.4598.671

Krause P., Boyle D. P., and Bäse F.: Comparison of different efficiency criteria for hydrological model assessment, *Adv. Geosci.*, 5, 89-97, doi:10.5194/adgeo-5-89-2005, 2005.

Kumar, E. (2008). *Artificial Intelligence: I.K.* International Publishing House Pvt. Limited.

Kyu-Cheoul Shim, Darrell G. Fontane, M.ASCE, and John W. Labadie, M.ASCE. Spatial Decision Support System for Integrated River Basin Flood Control. *J. Water Resour. Plann. Manage.* 2002.128:190-201. DOI: 10.1061/(ASCE)0733-9496(2002)128:3(190)

Laarhoven, V., J, P., Aarts, & H, E. (1987). Simulated annealing: theory and applications (Vol. 37): *Springer Science & Business Media*.

Levy, Jason K., Jens Hartmann, Kevin W. Li, Yunbi An, and Ali Asgary, (2007). Multi-Criteria Decision Support Systems for Flood Hazard Mitigation and Emergency Response in Urban Watersheds. *Journal of the American Water Resources Association (JAWRA)* 43(2):346-358. DOI: 10.1111/j.1752-1688.2007.00027.x

Luce, C. H., & Cundy, T. W. (1994). Parameter identification for a runoff model for forest roads Water. *Resources and Research*, 30 ((4) (1994)), pp. 1057–1069.

M. Abramowitz and I.E. Stegun (ed.) (1964), *Handbook of Mathematical Functions With Formulas, Graphs and Mathematical Tables*. National Bureau of Standards, Washington.

Mark P.J. DeGagnea, Glen G. Douglas, Henry R. Hudson, Slobodan P. Simonovic (1996). A decision support system for the analysis and use of stage-discharge rating curves. *Journal of Hydrology*, 184, 225-241

Marryott R. A., Dougherty D. E., & Stollar R. L. (1993). Optimal groundwater management: 2. Application of simulated annealing to a field-scale contamination site. *Water Resources Research*, 29(4), 847-860. doi:10.1029/92WR02801

Mauldon A. D., Karasaki K., Martel S. J., Long J. C., Landsfeld M., Mensch A., & Vomvoris S. (1993). An inverse technique for developing models for fluid flow in fracture systems using simulated annealing. *Water Resour. Res*, 29(11), 3775-3789.

Metropolis N., Rosenbluth A. W., Rosenbluth M. N., Teller A. H., & Teller E. (1953). Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6), 1087-1092. doi:doi:http://dx.doi.org/10.1063/1.1699114

Michael McMillan. (2005). *Data Structures and Algorithms Using Visual Basic.NET*, Cambridge University Press

Miller, Ryan C., D. Phillip Guertin, and Philip Heilman. (2004). Information Technology in Watershed Management Decision Making. *Journal of the American Water Resources Association (JAWRA)*, 40(2):347-357.

Miller S.N., D.J. Semmens R.C. Miller, M. Hernandez, D.C.Goodrich, W.P. Miller, W.G. Kepner, and D. Ebert (2002). GIS-Based Hydrologic Modelling: The Automated Geospatial Watershed Assessment Tool. *Proceedings of the Second Federal Interagency Hydrologic Modelling Conference*, Las Vegas Nevada, CDROM, 12 pp.

Moriasi D. N, Arnold J. G., Van Liew M. W., Bingner R. L., Harmel R. D., Veith T. L. (2007). Model Evaluation Guidelines for Systematic Quantification of Accuracy in Watershed Simulation. *American Society of Agricultural and Biological Engineers* ISSN 0001–2351. Vol. 50(3): 885–900

Access. (2007). Microsoft.

Muleta M. K., & Nicklow J. W. (2005). Sensitivity and uncertainty analysis coupled with automatic calibration for a distributed watershed model. *Journal of Hydrology*, 306 (2005), pp. 127–145.

Nabi G., Reddy H., & Habib-ur-Rehman. (2012). Gradually Varied Flow Computation in Series, Tree Type and Looped Compound Channel Networks *Pak. J. Engg. & Appl. Sci.*, Vol. 11(July., 2012 (p. 47-59)

Nash J. E., and J. V. Sutcliffe. (1970). River flow forecasting through conceptual models: Part 1. A discussion of principles. *J. Hydrology*, 10(3): 282-290.

National Research Council (1982). *Levee Policy for the National Flood Insurance Policy* (Washington, DC: National Academies Press).

Nguyen, B. Q. (2011). Assessing impacts of climate change on disasters related to stream flow (flood and drought) in Quang Nam, Viet Nam. Unpublished manuscript.

Nguyen Q. K. and Kawano H. (1995). Simultaneous Solution for Flood Routing in Channel Networks. *JHE, ASCE,* Vol. 121, No. 10, pp. 744-750.

Ogden F.L., J. Garbrecht, P.A. DeBarry, and L.E. Johnson, (2001). GIS and Distributed Watershed Models. II: Modules, Interfaces and Models. *Journal of Hydrologic Engineering* 6(6):515-523.

P. Krause1 D. P. Boyle, and F. Base (2005). Comparison of different efficiency criteria for hydrological model assessment. *Advances in Geosciences*, 5, 89–97, 2005. SRef-ID: 1680-7359/adgeo/2005-5-89.

Parhi P. K., Sankhua R. N., & Roy G. P. (2012). Calibration of Channel Roughness for Mahanadi River, (India) Using HEC-RAS Model. *Journal of Water Resource and Protection*, Vol 04(Iss 10), pp 847-850

Parker B.J. and G.A. Al-Utabi, (1986). Decision Support Systems: The Reality that Seems to be Hard to Accept? *OMEGA lnternational Journal of Manaoement Science* (14),2.

Paul Selormey (2004). *DotNetMatrix: Simple Matrix Library for .NET.* Available at: https://www.codeproject.com/Articles/5835/DotNetMatrix-Simple-Matrix-Library-for-NET. [Accessed 2 Jun. 2014].

Picard, E. (1929). *Traité d'analyse.* Paris: Gauthier-Villars.

Powell, M. J. D. (1974). *Introduction to Constrained Optimization in: Numerical Methods for Constrained Optimization*, P.E. Gill and W. Murray (eds.). Academic Press, London and New York.

Prasuhn, A. L. (1992). *Fundamentals of Hydraulic Engineering*. New York, NY: Oxford University Press.

Press W. H., Teukolsky S. A., Vetterling W. T., and Flannery B. P. (1992). *Numerical Recipes in FORTRAN*, 2nd ed., Cambridge Univ. Press, New York.

R. Szymkiewicz. (2010)., *Numerical Modeling in Open Channel Hydraulics*, Springer, Amsterdam, The Netherlands

Ramesh R., Datta B., Bhallamudi S., & Narayana A. (2000). Optimal Estimation of Roughness in Open-Channel Flows. *Journal of Hydraulic Engineering*, 126(4), 299-303. doi:doi:10.1061/(ASCE)0733-9429(2000)126:4(299)

Reddy H. P. and Bhallamudi S. M. (2004) Gradually varied flow computation in cyclic looped channel networks. *J. Irrig. Drain. Eng.* 130(5): 424–431

Reshma T., Reddy K. V., Pratap D., & Agilan V. (2015, 28-29 December 2014). Optimization of Manning's Roughness Coefficients for a watershed using Multi-Objective Genetic Algorithm. *Paper presented at the meeting of the International Conference on Modeling Tools for Sustainable Water Resources Management Department of Civil Engineering*, Hyderabad.

RETA 6470. (2011). Investment, Managing water in Asia's river basins: Charting progress and facilitating - The Vu Gia - Thu Bon Basin. Unpublished manuscript.

Sachs L. (1984). *Applied statistics: A handbook of techniques.* New York: Springer-Verlag.

Sen D. and Garg N. (2002). Efficient Algorithm for Gradually Varied Flows in Channel Networks. *J. Irrig. Drain Eng.*, 128(6), 351–357.

Sevat E., and A. Dezetter. (1991). Selection of calibration objective functions in the context of rainfall-runoff modeling in a Sudanese savannah area. *Hydrological Sci. J.* 36(4): 307-330.

Simonovic S.P. and D.A. Savic (1989). lntelligent Decision Support and Reservoir Management and Operations. *ASCE Journal of Computing in Civil Engineering*, 3, (4):367-385.

Singh V. P. (Ed.) (1995). *Computer Models of Watershed Hydrology.* Water Resources Publications, Highlands Ranch, Colorado, 1130pp

Slobodan P. Simonovic (1999). Decision Support System for Flood Management in Red River Basin, *Canadian Water Resources Journal*, 24:3, 203-223

Smith D.I. (1994). Flood damage estimation-a review of urban stage-damage curves and loss functions. *Water SA* 20(3):231–238

Subramanya K. (1991). *Flow in Open Channels*. New Delhi, India: Tata McGraw-Hill Publishing Co. Ltd.

Szymkiewicz R. (2010). *Numerical modeling in open channel hydraulics.*

Thierauf R.J. (1988). *User-Oriented Decision Support Systems*, Prentice Hall, Englewood Cliffs, NJ.

Timbadiya, P. V., Patel, P. L., & Porey, P. D. (2011). Calibration of HEC-RAS model on prediction of flood for lower Tapi River, India. *Journal of Water Resource and Protection (JWARP)*, Vol.3(11)(p.805(7)).

Todini E., Marsigli M., Pani G., & Vignoli R. (1997). Operational real-time ¯flood forecasting systems based on EFFORTS. *In Proceedings of the RIBAMOD Workshop.*

U.S. Army Corps of Engineers (1988). *FDA: Flood Damage Analysis Package, User's Manual*, CPD-59, Hydrologic Engineering Center. Davis, CA.

U.S. Army Corps of Engineers (1989). *EAD: Expected Annual Flood Damage Computation, User's Manual*, Hydrologic Engineering Center, Davis, CA.

U.S. Army Corps of Englneers, (1989a). *SID: Structure lnventory for Damage Analysis, User's Manual,* CPD-41, Hydrologic Engineering Center, Davis, CA.

University of Bologna (1996). AFORISM, a comprehensive forecasting system for food risk mitigation and control. Final Report of Contract EC-EPOC-CT90-0023.

Usul N. and Yilmaz M. (2004). Estimation of instantaneous unit hydrograph with clark's technique in GIS, *METU, Civil Engg department*, Ankara, Turkey.

Usul N., & Turan B. (2006). Flood forecasting and analysis within the Ulus Basin, Turkey, using geographic information systems. *Natural Hazards*, 39(2), 213-229. doi:10.1007/s11069-006-0024-8

Vermieren J. C. & Watson C. C. (2001). New technology for improved storm risk assessment in the Caribbean, *Disaster Management*, 6, pp. 191–196.

Vogel R. M., and I. Wilson (1996), Probability distribution of annual maximum, mean and minimum streamflows in the United States, *J. Hydrol. Eng.*, 1(2), 69 – 76.

Vu, T. T., Nguyen, L., Hoang, T., Bui, T., Nguyen, M., & Nguyen, T. (2011). Solutions for flood and drought prevention and mitigation in Quang Nam. Unpublished manuscript.

Wagener, Thorsten, Wheater Howard, and Gupta, Hoshin: (2004). *Rainfall-Runoff Modelling in Gauged and Ungauged Catchments*. Imperial College Press, 332p., ISBN 1-860944-66-3.

Wylie E. B. (1972). Water surface profiles in divided channels. *Journal of Hydraulic Research*, International Association for Hydraulic Research, 10(3), pp.

Xiahua Yang and Bengt Rystedt (2002). Predicting Flood and Risk Using GIS and Hydrodynamic Model: A Case Study at Eskilstuna, Sweden. *Idian Cartographer.*

Yi-Ru Chen, Chao-Hsien Yeh & Bofu Yu (2011). Integrated application of the analytic hierarchy process and the geographic information system for flood risk assessment and flood plain management in Taiwan. *Nat Hazards*, 59:1261–1276. DOI 10.1007/s11069-011-9831-7.

Zang, Xuesong and Raghavan Srinivasan, (2009). GIS-Based Spatial Estimation: A Comparison of Geostatistical Approaches. *Journal of American Water Resources Association (JAWRA)*, 45(4):894-906. DOI:10.111/j.1752-1688.2009.00335.x

Appendix

Appendix 5.1. Rainfall runoff code

```
Imports ESRI.ArcGIS.ArcMapUI
Imports ESRI.ArcGIS.Geodatabase
Imports ESRI.ArcGIS.Carto
Imports ESRI.ArcGIS.CartoUI
Imports ESRI.ArcGIS.Display
Imports ESRI.ArcGIS.SystemUI
Imports ESRI.ArcGIS.Geometry
Imports System.Runtime.InteropServices
Imports ESRI.ArcGIS.esriSystem
Imports System.Drawing
Imports ESRI.ArcGIS.ADF
Imports System.Threading
Imports System.Windows.Forms
Imports ESRI.ArcGIS.DataSourcesRaster
Imports ZedGraph

Public Class Regression_frm
  Public MxDoc As IMxDocument
  Dim pMap As IMap
  Dim Key As String = "Software\WWFVN\ Quang NamFlood"
  Private Q() As Single
```

```
  Private Sub Regression_frm_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    On Error Resume Next
    Dim regKey As CRegistry
    regKey = New CRegistry
    Dim databasePath As Object
    regKey.ReadValue(regKey.HKeyLocalMachine, Key, "DB", databasePath)
    Me.txtDatabasePath.Text = databasePath

    'populate stations list
    Dim tableStations As DataTable
    tableStations = DataAccess.SelectCols("Stations", "Station", "Station <> """)
    Me.lstStations.DataSource = tableStations
    Me.lstStations.ValueMember = "Station"
    Me.lstStations.DisplayMember = "Station"

    'populate event list
    Dim eventTable As DataTable
    eventTable = DataAccess.SelectCols("Event_q", "[FromDate], [ToDate], [Interval]", "")
    Me.lstEventTime.DataSource = eventTable
    Me.lstEventTime.ValueMember = "Interval"
    Me.lstEventTime.DisplayMember = "Interval"
    'Me.cmbEventTime.SelectedIndex = -1
```

```
    'populatte validation equations
    Dim equationTable As DataTable
    equationTable = DataAccess.SelectCols("equations", "[Equation]", "[Equation] <> """)
    Me.cmbEquation.DataSource = equationTable
    Me.cmbEquation.ValueMember = "Equation"
    Me.cmbEquation.DisplayMember = "Equation"
    'Me.cmbEquation.SelectedIndex = -1

  End Sub
```

```
  Private Sub cmdOpenDatabase_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdOpenDatabase.Click
    Me.OpenFileDialog1.FileName = ""
    Dim rs As DialogResult
    rs = Me.OpenFileDialog1.ShowDialog()
    If rs = Windows.Forms.DialogResult.OK Then
      Me.txtDatabasePath.Text = OpenFileDialog1.FileName
      Dim regKey As CRegistry
      regKey = New CRegistry
      regKey.WriteValue(regKey.HKeyLocalMachine, Key, "DB", Me.txtDatabasePath.Text)
    End If
  End Sub
```

```
  Private Sub cmdLoadData_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles cmdLoadData.Click
    'On Error Resume Next

    Dim stations As String = ""
    For Each item As Object In Me.lstStations.CheckedItems
      If stations.Length = 0 Then
        stations = "[" & item("Station").ToString & "]"
      Else
        stations = stations & ", [" & item("Station").ToString & "]"
      End If
    Next

    Dim queryDate As String
    If Me.lstEventTime.CheckedItems.Count <> 0 Then
      For i As Integer = 0 To Me.lstEventTime.CheckedItems.Count - 1
        If i = 0 Then
          queryDate = "(Time >= #" & Me.lstEventTime.CheckedItems.Item(i)("Interval").Split("-")(0) & _
          "# AND Time <= #" & Me.lstEventTime.CheckedItems.Item(i)("Interval").Split("-")(1) & "#)"
        Else
          queryDate = queryDate & " OR (Time >= #" &
Me.lstEventTime.CheckedItems.Item(i)("Interval").Split("-")(0) & _
          "# AND Time <= #" & Me.lstEventTime.CheckedItems.Item(i)("Interval").Split("-")(1) & "#)"
        End If
      Next
    Else
      queryDate = "(Time >= #" & Me.dateFrom.Value & "# AND Time <= #" & Me.dateTo.Value & "#)"
    End If
```

```
    Dim table As String
    Dim fields As String
    If stations.Contains("Thanh My") Then
      table = "RO_ThanhMy_q"
      fields = "Time, " & stations & ", Q"
    ElseIf stations.Contains("Nong Son") Then
      table = "RO_NongSon_q"
      fields = "Time, " & stations & ", Q"
    End If

    Dim queryString As String
    queryString = ""
    Dim qRFROQuery As DataTable
    qRFROQuery = DataAccess.SelectCols(table, fields, queryDate)

    Dim cols As Integer = qRFROQuery.Columns.Count
    Dim rows As Integer = qRFROQuery.Rows.Count

    Me.DataGridView1.DataSource = qRFROQuery

    Me.DataGridView1.AutoResizeColumns(DataGridViewAutoSizeColumnsMode.AllCells)

    '--------prepare the variable lists
    Me.cmbDependentVariable.Items.Clear()

    For i As Integer = 0 To Me.DataGridView1.ColumnCount - 1
      Me.cmbDependentVariable.Items.Add(Me.DataGridView1.Columns.Item(i).HeaderText)
    Next

    'Prepare Q vector for the shifting control
    Dim numRows As Integer
    'it seems that Microsoft add one more row to the end (in edit mode), so -1
    numRows = Me.DataGridView1.RowCount - 1
    ReDim Q(numRows - 1)
    Dim colIdx As Integer
    For i = 0 To Me.DataGridView1.ColumnCount - 1
      If Me.DataGridView1.Columns.Item(i).HeaderText = "Q" Then
        colIdx = i
        Exit For
      End If
    Next
    For i As Integer = 0 To numRows - 1
      Q(i) = Me.DataGridView1.Rows(i).Cells(colIdx).Value
    Next

  End Sub
```

```
  Private Sub cmdRunRegression_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdRunRegression.Click
    On Error Resume Next
```

```
Dim numStations As Integer
numStations = Me.lstStations.CheckedItems.Count
Dim numRows As Integer
'it seems that Microsoft add one more row to the end (in edit mode), so -1
numRows = Me.DataGridView1.RowCount - 1

Dim A(numRows - 1)() As Double
Dim Y(numRows - 1)() As Double

Dim k As Integer
Dim curStation As String

For i As Integer = 0 To numRows - 1
  A(i) = New Double(numStations) {}
  Y(i) = New Double(0) {}
  k = 0
  curStation = Me.lstStations.CheckedItems.Item(k)("Station")

  For j As Integer = -1 To Me.DataGridView1.Columns.Count - 1
    If j = -1 Then
      A(i)(0) = 1
    Else
      If Me.DataGridView1.Columns.Item(j).HeaderText = curStation Then
        A(i)(k + 1) = Me.DataGridView1.Rows(i).Cells(j).Value
        k = k + 1
        curStation = Me.lstStations.CheckedItems.Item(k)("Station")
ElseIf Me.DataGridView1.Columns.Item(j).HeaderText =   Me.cmbDependentVariable.Text Then
        Y(i)(0) = Me.DataGridView1.Rows(i).Cells(j).Value
      End If
    End If
  Next
Next

Dim A_ As New DotNetMatrix.GeneralMatrix(A)
Dim Y_ As New DotNetMatrix.GeneralMatrix(Y)
Dim b As New DotNetMatrix.GeneralMatrix(numStations + 1, 1) 'number of rows and columns
b = A_.Transpose.Multiply(A_).Inverse.Multiply(A_.Transpose).Multiply(Y_)

'===================== Model diagnosis ====================

'Avg Y
Dim AvgY As Double = 0
For i = 0 To numRows - 1
  AvgY = AvgY + Y(i)(0)
Next
AvgY = AvgY / numRows

'Avg predicted Y
Dim AvgPreY As Double
Dim preY As DotNetMatrix.GeneralMatrix
preY = A_.Multiply(b)

For i = 0 To numRows - 1
```

```
  AvgPreY = AvgPreY + preY.Array(i)(0)
Next
AvgPreY = AvgPreY / numRows

'Total sum of square
Dim SST As Double
For i = 0 To numRows - 1
  SST = SST + (Y(i)(0) - AvgY) ^ 2
Next
SST = Format(SST, "0.00")

'Error sum of square
Dim SSE As Double
For i = 0 To numRows - 1
  SSE = SSE + (Y(i)(0) - preY.Array(i)(0)) ^ 2
Next
SSE = Format(SSE, "0.00")

'Regression sum Square
Dim SSR As Double
SSR = SST - SSE
SSR = Format(SSR, "0.00")

'F statistic
Dim F As Double
F = (SSR / numStations) / (SSE / (numRows - numStations - 1))
F = Format(F, "0.00")

'R statistic
Dim r As Double
r = SSR / SST
r = Format(r, "0.00")

'Model standard error
Dim s2 As Double
Dim s As Double
s2 = SSE / (numRows - numStations - 1)
s = s2 ^ 0.5
s = Format(s, "0.00")

'Covarian Matrix
Dim c As DotNetMatrix.GeneralMatrix
c = A_.Transpose.Multiply(A_).Inverse

'Co-efficient standard deviations
Dim s_b() As Double
ReDim s_b(numStations)
For i = 0 To numStations
  s_b(i) = s * (c.Array(i)(i) ^ 0.5)
Next

't_value
Dim t_value() As Double
```

```
    ReDim t_value(numStations)
    For i = 0 To numStations
      t_value(i) = b.Array(i)(0) / s_b(i)
    Next

    '====================== Results output ====================
    Me.lstRs.Items.Clear()
    Me.lstRs.Items.Add("")
    Me.lstRs.Items.Add(FmtText("________________________________", 60))
    Me.lstRs.Items.Add("")
    Me.lstRs.Items.Add(FmtText("COEFFICIENTS", 20) & FmtText("Std.Err", 20) & FmtText("t_test", 20))
    Me.lstRs.Items.Add(FmtText("-------------------------------------------------------------------", 60))

    Dim eqn As String = Me.cmbDependentVariable.Text & " = "
    For i = 0 To numStations
      Me.lstRs.Items.Add(FmtText(b.Array(i)(0), 20) & FmtText(s_b(i), 20) & FmtText(t_value(i), 20))
      If i = 0 Then
        eqn = eqn & Format(b.Array(i)(0), "0.000")
      Else
        eqn = eqn & " + " & Format(b.Array(i)(0), "0.000") & "*" & Me.lstStations.CheckedItems(i -
1)("Station")
      End If
    Next
    'write equation to the textbox
    Me.txtEquation.Text = eqn.Substring(4)

    Me.lstRs.Items.Add("")
    Me.lstRs.Items.Add("Multiple regression equation:")
    Me.lstRs.Items.Add(eqn)

    Me.lstRs.Items.Add(FmtText("________________________________", 120))
    Me.lstRs.Items.Add("")
    Me.lstRs.Items.Add("ANOVA TABLE")
    Me.lstRs.Items.Add(FmtText("________________________________", 120))
    Me.lstRs.Items.Add("")
    Me.lstRs.Items.Add(FmtText("SOURCE", 10) & FmtText("df", 5) & FmtText("Sum of square", 15) &
FmtText("Mean square", 15) & FmtText("F", 10) & FmtText("Square R", 10))
    Me.lstRs.Items.Add(FmtText("-------------------------------------------------------------------", 120))
    Me.lstRs.Items.Add(FmtText("Regression", 10) & FmtText(numStations, 5) & FmtText(SSR, 15) &
FmtText(Format(SSR / numStations, "0.00"), 15) & FmtText(F, 10) & FmtText(r, 10))
    Me.lstRs.Items.Add(FmtText("Residual", 10) & FmtText(numRows - numStations - 1, 5) &
FmtText(SSE, 15) & FmtText(Format(SSE / (numRows - numStations - 1), "0.00"), 15))
    Me.lstRs.Items.Add(FmtText("-------------------------------------------------------------------", 120))
    Me.lstRs.Items.Add(FmtText("Total", 10) & FmtText(numStations + (numRows - numStations - 1), 5)
& FmtText(SST, 15))
    Me.lstRs.Items.Add(FmtText("________________________________", 120))


    '=====================Drawing the graph ====================
    Dim PairPoints As New PointPairList
    Dim PredictedPairs As New PointPairList
    For i As Integer = 0 To numRows - 1
      For j As Integer = 0 To Me.DataGridView1.ColumnCount - 1
```

```
      If Me.DataGridView1.Columns(j).HeaderText = Me.cmbDependentVariable.Text Then
        PairPoints.Add(i, Me.DataGridView1.Rows.Item(i).Cells(j).Value)
        PredictedPairs.Add(i, preY.Array(i)(0))
      End If
    Next
  Next

  Me.Graph1.GraphPane.CurveList.Clear()
  Me.Graph1.GraphPane.AddCurve("observed", PairPoints, Color.Black, SymbolType.Circle)
  Me.Graph1.GraphPane.AddCurve("predicted", PredictedPairs, Color.Red, SymbolType.Diamond)
  Me.Graph1.GraphPane.Title.Text = ""
  Me.Graph1.GraphPane.XAxis.Title.Text = "Time"
  Me.Graph1.GraphPane.YAxis.Title.Text = "Run off"
  Me.Graph1.AxisChange()
  Me.Graph1.Refresh()

End Sub
```

```
Private Function FindGridColum(ByVal headerText As String, ByRef grid As DataGridView) As Integer
  FindGridColum = -1
  For i As Integer = 0 To grid.Columns.Count - 1
    If grid.Columns.Item(i).HeaderText = headerText Then
      Return i
      Exit For
    End If
  Next
End Function
```

```
Private Function FmtText(ByVal input As String, ByVal length As Integer) As String
  Dim concat As String = "                "
  Dim tmp As String
  tmp = input & concat
  Return tmp.Substring(0, length)
End Function
```

```
Function norm(ByVal z As Double) As Double
  Dim Q As Double
  Q = z * z
  If Math.Abs(z) > 7 Then
    norm = (1 - 1 / Q + 3 / (Q * Q)) * Math.Exp(-Q / 2) / (Math.Abs(z) * (3.14159 / 2) ^ 0.5)
  Else
    norm = ChiSq(Q, 1)
  End If
End Function
```

```
Function ChiSq(ByVal X As Double, ByVal n As Double) As Double
  Dim Q As Double
  Dim p As Double
```

```
    Dim k As Double
    Dim t As Double
    Dim a As Double

    '---------------------------------------------------------------------------------
    If X > 1000 Or n > 1000 Then

      Q = norm((((X / n) ^ (1 / 3)) + 2 / (9 * n) - 1) / ((2 / (9 * n)) ^ 0.5)) / 2

      If X > n Then
        ChiSq = Q
      End If

      ChiSq = 1 - Q
    End If
    '---------------------------------------------------------------------------------

    p = Math.Exp(-0.5 * X)

    If n Mod 2 = 1 Then
      p = p * (2 * X / 3.14159) ^ 0.5
    End If

    k = n

    Do While k >= 2
      p = p * X / k
      k = k - 2
    Loop

    t = p
    a = n

    Do While t > 0.000000000000001 * p
      a = a + 2
      t = t * X / a
      p = p + t
    Loop

    ChiSq = 1 - p
  End Function
```

```
  Private Sub VScrollBar1_Scroll(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.ScrollEventArgs) Handles VScrollBar1.Scroll

    Dim shift As Integer
    shift = VScrollBar1.Value

    Dim colIdx As Integer
    For i = 0 To Me.DataGridView1.ColumnCount - 1
      If Me.DataGridView1.Columns.Item(i).HeaderText = "Q" Then
        colIdx = i
```

```
      Exit For
    End If
  Next

  For i As Integer = 0 To Me.DataGridView1.RowCount - 2
    Try
      Me.DataGridView1.Rows(i).Cells(colIdx).Value = Q(i - shift)
    Catch
      Me.DataGridView1.Rows(i).Cells(colIdx).Value = 0
    End Try
  Next
End Sub
```

```
 Private Sub cmdSaveEquation_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles cmdSaveEquation.Click
   DataAccess.RunGeneralCommand("INSERT INTO equations (Name, Equation) VALUES ('" & _
              Me.dateFrom.Value.ToString & "', '" & Me.txtEquation.Text & "')")
 End Sub
```

```
 Private Sub cmdValidate_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles cmdValidate.Click
   Dim equation() As String
   equation = Me.cmbEquation.Text.Split("+")
   For i As Integer = 0 To equation.GetUpperBound(0)
     equation(i) = equation(i).Trim
   Next
   Dim matrix(,) As Single
   ReDim matrix(Me.DataGridView1.Rows.Count - 2, equation.GetUpperBound(0))
   For i As Integer = 0 To Me.DataGridView1.Rows.Count - 2
     For j As Integer = 0 To equation.GetUpperBound(0)
       If j = 0 Then
         matrix(i, j) = equation(0)
       Else
         For k As Integer = 0 To Me.DataGridView1.ColumnCount - 1
           If DataGridView1.Columns.Item(k).HeaderText = equation(j).Split("*")(1).Trim Then
                               matrix(i, j) = Me.DataGridView1.Rows(i).Cells(k).Value *
                               equation(j).Split("*")(0).Trim
           End If
         Next
       End If
     Next
   Next

   'Add 2 Q columns
   ReDim Preserve matrix(matrix.GetUpperBound(0), matrix.GetUpperBound(1) + 2)
   For i As Integer = 0 To matrix.GetUpperBound(0)
     For j As Integer = 0 To Me.DataGridView1.ColumnCount - 1
       If Me.DataGridView1.Columns(j).HeaderText = "Q" Then
         matrix(i, matrix.GetUpperBound(1) - 1) = Me.DataGridView1.Rows(i).Cells(j).Value
       End If
     Next
```

```
    'calculated Q, last column
    For k As Integer = 0 To matrix.GetUpperBound(1) - 2
      matrix(i, matrix.GetUpperBound(1)) = matrix(i, matrix.GetUpperBound(1)) + matrix(i, k)
    Next
   Next

    'NASH
   Dim Qm_avg As Single
   Dim upper As Single
   For i As Integer = 0 To matrix.GetUpperBound(0)
    Qm_avg = Qm_avg + matrix(i, matrix.GetUpperBound(1) - 1)
                upper = upper + (matrix(i, matrix.GetUpperBound(1) - 1) - matrix(i,
                matrix.GetUpperBound(1))) ^ 2
   Next
   Qm_avg = Qm_avg / (matrix.GetUpperBound(0) + 1)
   Dim lower As Single
   For i As Integer = 0 To matrix.GetUpperBound(0)
    lower = lower + (matrix(i, matrix.GetUpperBound(1) - 1) - Qm_avg) ^ 2
   Next
   Me.txtNash.Text = 1 - (upper / lower)


   '=====================Drawing the graph ===================
   Dim PairPoints As New PointPairList
   Dim PredictedPairs As New PointPairList
   For i As Integer = 0 To matrix.GetUpperBound(0)
    PairPoints.Add(i, matrix(i, matrix.GetUpperBound(1) - 1))
    PredictedPairs.Add(i, matrix(i, matrix.GetUpperBound(1)))
   Next

   Me.graph2.GraphPane.CurveList.Clear()
   Me.graph2.GraphPane.AddCurve("observed", PairPoints, Color.Black, SymbolType.Circle)
   Me.graph2.GraphPane.AddCurve("predicted", PredictedPairs, Color.Red, SymbolType.Diamond)
   Me.graph2.GraphPane.Title.Text = ""
   Me.graph2.GraphPane.XAxis.Title.Text = "Time"
   Me.graph2.GraphPane.YAxis.Title.Text = "Run off"
   Me.graph2.AxisChange()
   Me.graph2.Refresh()
  End Sub
End Class
```

Appendix 6.1. RiverCrossSection class

```
Public Class RiverCrossSection
   'Point list in this format 12.34;56.78%(12.34;56.78%12.34;56.78)%12.34;56.78
   'This will be stored in a feature classs field
   Private stPointList As String 'directly from feature class
   Public sglWaterElevation As Single
   Public Q As Single

   Private stManning As String
```

```
Private n0, n1, n2 As Single
Public n As Single

Private stLOB As String 'like this 12.34;56.78%12.34;56.78%12.34;56.78
Private stMainChannel As String
Private stROB As String

Private K0, K1, K2 As Single
Public K As Single
Private K0_, K1_, K2_ As Single
Public K_ As Single

Private B0, B1, B2 As Single
Public B As Single
Private B0_, B1_, B2_ As Single
Public B_ As Single

Private R0, R1, R2 As Single
Public R As Single
Private R0_, R1_, R2_ As Single
Public R_ As Single

Public z As Single
Public z_ As Single

Private A0, A1, A2 As Single
Public A As Single
Private A0_, A1_, A2_ As Single
Public A_ As Single

Private WP0, WP1, WP2 As Single
Public WP As Single
Private WP0_, WP1_, WP2_ As Single
Public WP_ As Single

Public Sf As Single
Public Sf_ As Single
Public Alfa As Single
Public Alfa_ As Single
Public flowDepth As Single 'this is not water elevation but flow depth
Public flowDepth_ As Single
Public V As Single 'Average velocity
Public V_ As Single
Public H As Single 'Energy budget/head
Public H_ As Single

'These are for derivatives
Public dR0dy, dR1dy, dR2dy As Single
Public dPdy As Single 'derivative of WP respect to flow depth
Public dRdy As Single
Public dAlfady As Single
Public dSfdy As Single
Public dHdy As Single
```

```
Public Sub New(ByVal PointList As String, ByVal ManningList As String, _
            ByVal Discharge As Single, ByVal flowDepth As Single)

    stPointList = PointList

    Dim Parts(0 To 2) As String
    Parts = stPointList.Split("#")
    stMainChannel = Parts(1)
    'adding overlapping point
    stLOB = Parts(0) & "%" & stMainChannel.Split("%")(0)
    'adding overlapping point
    stROB = stMainChannel.Split("%")(stMainChannel.Split("%").Count - 1) & "%" & Parts(2)

    stManning = ManningList
    n0 = stManning.Split("#")(0)
    n1 = stManning.Split("#")(1)
    n2 = stManning.Split("#")(2)

    Q = Discharge

    Dim tmpChannel() As String
    tmpChannel = stPointList.Replace("#", "%").Split("%")
    Dim tmpZ As Single
    tmpZ = Val(tmpChannel(0).Split(";")(1)) 'initialize as the first depth
    For i As Integer = 1 To tmpChannel.GetUpperBound(0)
        If Val(tmpChannel(i).Split(";")(1)) < tmpZ Then
            tmpZ = Val(tmpChannel(i).Split(";")(1))
        End If
    Next
    z = tmpZ
    z_ = z
    Me.flowDepth = flowDepth
    sglWaterElevation = z + flowDepth

End Sub
```

```
Private Function ReshapeBank(ByVal stBank As String, ByVal inWaterElevation As Single) As String
    'extend banks on left and right
    Dim tmpArr() As String
    tmpArr = stBank.Split("%")

    Dim tmpStBank As String
    tmpStBank = stBank

    'extend left side
    If Val(tmpArr(0).Split(";")(1)) < inWaterElevation Then
        tmpStBank = tmpArr(0).Split(";")(0) & ";" & _
                    Format(inWaterElevation, "0.0000") & "%" & tmpStBank
    End If
    'extend right side
```

```
        If Val(tmpArr(tmpArr.Count - 1).Split(";")(1)) < inWaterElevation Then
            tmpStBank = tmpStBank & "%" & tmpArr(tmpArr.Count - 1).Split(";")(0) & ";" & _
            Format(inWaterElevation, "0.0000")
        End If

        'insert intersection points:
        'detect heights using array then insert into string
        tmpArr = tmpStBank.Split("%") 're-initialize the array after the above extending step
        Dim i As Integer
        Dim sglLeftElev As Single
        Dim sglRightElev As Single
        Dim sglXLeft As Single
        Dim sglXRight As Single
        For i = 0 To tmpArr.Count - 2
            sglXLeft = tmpArr(i).Split(";")(0)
            sglLeftElev = tmpArr(i).Split(";")(1)
            sglXRight = tmpArr(i + 1).Split(";")(0)
            sglRightElev = tmpArr(i + 1).Split(";")(1)
            Dim sglMidX As Single
            If sglLeftElev < inWaterElevation And inWaterElevation < sglRightElev Then
                'interpolate a position
                sglMidX = sglXLeft + ((sglXRight - sglXLeft) * (inWaterElevation - sglLeftElev)) / (sglRightElev
- sglLeftElev)
                'insert to the right
                tmpStBank = tmpStBank.Replace(tmpArr(i), _
                tmpArr(i) & "%" & Format(sglMidX, "0.0000") & ";" & Format(inWaterElevation, "0.0000"))
            ElseIf sglLeftElev > inWaterElevation And inWaterElevation > sglRightElev Then
                'interpolate a position
                sglMidX = sglXLeft + ((sglXRight - sglXLeft) * (sglLeftElev - inWaterElevation)) / (sglLeftElev -
sglRightElev)
                'insert to the right
                tmpStBank = tmpStBank.Replace(tmpArr(i), _
                tmpArr(i) & "%" & Format(sglMidX, "0.0000") & ";" & Format(inWaterElevation, "0.0000"))
            End If
        Next

        'collapse points higer than water elevation
        tmpArr = tmpStBank.Split("%") 're-initialize the array after the above insert step
        Dim tmpElev As Single
        For i = 0 To tmpArr.Count - 1
            tmpElev = Val(tmpArr(i).Split(";")(1))
            If tmpElev > inWaterElevation Then
                tmpStBank = tmpStBank.Replace(Format(tmpElev, "0.0000"), _
                        Format(inWaterElevation, "0.0000"))
            End If
        Next

        Return tmpStBank

    End Function
```

```
    Private Sub CalculateSubCSProperties(ByVal LOB_Main_ROB_012 As Byte, _
```

```
        ByVal stGeomPart As String)
'This function is intended for one sub-section of a cross-section (LOB, ROF, Main channel)
'This is an internal function, the public one is for the whole cross-section

Dim tmpGeom As String
tmpGeom = stGeomPart
tmpGeom = ReshapeBank(tmpGeom, sglWaterElevation) 'reshape it according to water elevation
Dim arrGeom() As String
arrGeom = tmpGeom.Split("%")

Dim i As Integer
Dim leftX, rightX, leftY, rightY As Single
Dim tmpArea, tmpLength, tmpB As Single

For i = 0 To arrGeom.Count - 2

    leftX = Val(arrGeom(i).Split(";")(0))
    rightX = Val(arrGeom(i + 1).Split(";")(0))
    leftY = Val(arrGeom(i).Split(";")(1))
    rightY = Val(arrGeom(i + 1).Split(";")(1))
    'area + wp + top width
    If rightY < sglWaterElevation Or leftY < sglWaterElevation Then
        tmpArea = tmpArea + _
        0.5 * (((sglWaterElevation - rightY) + (sglWaterElevation - leftY)) * (rightX - leftX))
        tmpLength = tmpLength + ((rightX - leftX) ^ 2 + (rightY - leftY) ^ 2) ^ 0.5
        tmpB = tmpB + Math.Abs(rightX - leftX)
    End If
Next

Select Case LOB_Main_ROB_012
    Case 0
        'Flow area
        A0 = tmpArea
        'top width
        B0 = tmpB

        '--the vertical part on the right of the LOB bank
        leftX = Val(arrGeom(arrGeom.Count - 2).Split(";")(0))
        rightX = Val(arrGeom(arrGeom.Count - 1).Split(";")(0))
        leftY = Val(arrGeom(arrGeom.Count - 2).Split(";")(1))
        rightY = Val(arrGeom(arrGeom.Count - 1).Split(";")(1))

        If leftX = rightX Then
            tmpLength = tmpLength - Math.Abs(rightY - leftY)
        End If
        WP0 = tmpLength

        'hydraulic radius
        If Single.IsNaN(A0 / WP0) Or Single.IsInfinity(A0 / WP0) Then
            R0 = 0
        Else
            R0 = A0 / WP0
        End If
```

```
    'conveyance
    Dim tmpK As Single
    tmpK = (1 / n0) * A0 * (R0 ^ (2 / 3))
    If Single.IsNaN(tmpK) Or Single.IsInfinity(tmpK) Then
        K0 = 0
    Else
        K0 = tmpK
    End If
Case 1
    'Flow area
    A1 = tmpArea
    'top width
    B1 = tmpB

    '--The vertical part on the left of the main channel
    leftX = Val(arrGeom(0).Split(";")(0))
    rightX = Val(arrGeom(1).Split(";")(0))
    leftY = Val(arrGeom(0).Split(";")(1))
    rightY = Val(arrGeom(1).Split(";")(1))

    If leftX = rightX Then
        tmpLength = tmpLength - Math.Abs(rightY - leftY)
    End If

    '--The vertical part on the right of the main channel
    leftX = Val(arrGeom(arrGeom.Count - 2).Split(";")(0))
    rightX = Val(arrGeom(arrGeom.Count - 1).Split(";")(0))
    leftY = Val(arrGeom(arrGeom.Count - 2).Split(";")(1))
    rightY = Val(arrGeom(arrGeom.Count - 1).Split(";")(1))

    If leftX = rightX Then
        tmpLength = tmpLength - Math.Abs(rightY - leftY)
    End If
    WP1 = tmpLength

    'hydraulic radius
    If Single.IsNaN(A1 / WP1) Or Single.IsInfinity(A1 / WP1) Then
        R1 = 0
    Else
        R1 = A1 / WP1
    End If

    'Conveyance
    Dim tmpK As Single
    tmpK = (1 / n1) * A1 * (R1 ^ (2 / 3))
    If Single.IsNaN(tmpK) Or Single.IsInfinity(tmpK) Then
        K1 = 0
    Else
        K1 = tmpK
    End If
Case 2
    'Flow area
```

```
        A2 = tmpArea
        'top width
        B2 = tmpB

        '--The vertical part on the left of the ROB
        leftX = Val(arrGeom(0).Split(";")(0))
        rightX = Val(arrGeom(1).Split(";")(0))
        leftY = Val(arrGeom(0).Split(";")(1))
        rightY = Val(arrGeom(1).Split(";")(1))

        If leftX = rightX Then
            tmpLength = tmpLength - Math.Abs(rightY - leftY)
        End If
        WP2 = tmpLength

        'hydraulic radius
        If Single.IsNaN(A2 / WP2) Or Single.IsInfinity(A2 / WP2) Then
            R2 = 0
        Else
            R2 = A2 / WP2
        End If

        'conveyance
        Dim tmpK As Single
        tmpK = (1 / n2) * A2 * (R2 ^ (2 / 3))
        If Single.IsNaN(tmpK) Or Single.IsInfinity(tmpK) Then
            K2 = 0
        Else
            K2 = tmpK
        End If
    End Select

End Sub
```

```
Private Sub CalculateSubCSProperties_dy(ByVal LOB_Main_ROB_012 As Byte, _
        ByVal stGeomPart As String, ByVal newWaterElevation As Single)
    'This function is intended for one sub-section of a cross-section (LOB, ROF, Main channel)
    'This is an internal function, the public one is for the whole cross-section

    Dim tmpGeom As String
    tmpGeom = stGeomPart
    tmpGeom = ReshapeBank(tmpGeom, newWaterElevation)
    Dim arrGeom() As String
    arrGeom = tmpGeom.Split("%")

    Dim i As Integer
    Dim leftX, rightX, leftY, rightY As Single
    Dim tmpArea, tmpLength, tmpB As Single

    For i = 0 To arrGeom.Count - 2
        leftX = Val(arrGeom(i).Split(";")(0))
        rightX = Val(arrGeom(i + 1).Split(";")(0))
```

```
    leftY = Val(arrGeom(i).Split(";")(1))
    rightY = Val(arrGeom(i + 1).Split(";")(1))
    'area + wp + top width
    If rightY < newWaterElevation Or leftY < newWaterElevation Then
        tmpArea = tmpArea + _
        0.5 * (((newWaterElevation - rightY) + (newWaterElevation - leftY)) * (rightX - leftX))
        tmpLength = tmpLength + ((rightX - leftX) ^ 2 + (rightY - leftY) ^ 2) ^ 0.5
        tmpB = tmpB + Math.Abs(rightX - leftX)
    End If
Next

Select Case LOB_Main_ROB_012
    Case 0
        'Flow area
        A0_ = tmpArea
        'top width
        B0_ = tmpB

        '--the vertical part on the right of the LOB bank
        leftX = Val(arrGeom(arrGeom.Count - 2).Split(";")(0))
        rightX = Val(arrGeom(arrGeom.Count - 1).Split(";")(0))
        leftY = Val(arrGeom(arrGeom.Count - 2).Split(";")(1))
        rightY = Val(arrGeom(arrGeom.Count - 1).Split(";")(1))

        If leftX = rightX Then
            tmpLength = tmpLength - Math.Abs(rightY - leftY)
        End If
        WP0_ = tmpLength

        'hydraulic radius
        If Single.IsNaN(A0_ / WP0_) Or Single.IsInfinity(A0_ / WP0_) Then
            R0_ = 0
        Else
            R0_ = A0_ / WP0_
        End If

        'conveyance
        Dim tmpK As Single
        tmpK = (1 / n0) * A0_ * (R0_ ^ (2 / 3))
        If Single.IsNaN(tmpK) Or Single.IsInfinity(tmpK) Then
            K0_ = 0
        Else
            K0_ = tmpK
        End If
    Case 1
        'Flow area
        A1_ = tmpArea
        'top width
        B1_ = tmpB

        '--The vertical part on the left of the main channel
        leftX = Val(arrGeom(0).Split(";")(0))
        rightX = Val(arrGeom(1).Split(";")(0))
```

```
    leftY = Val(arrGeom(0).Split(";")(1))
    rightY = Val(arrGeom(1).Split(";")(1))

    If leftX = rightX Then
        tmpLength = tmpLength - Math.Abs(rightY - leftY)
    End If

    '--The vertical part on the right of the main channel
    leftX = Val(arrGeom(arrGeom.Count - 2).Split(";")(0))
    rightX = Val(arrGeom(arrGeom.Count - 1).Split(";")(0))
    leftY = Val(arrGeom(arrGeom.Count - 2).Split(";")(1))
    rightY = Val(arrGeom(arrGeom.Count - 1).Split(";")(1))

    If leftX = rightX Then
        tmpLength = tmpLength - Math.Abs(rightY - leftY)
    End If
    WP1_ = tmpLength

    'hydraulic radius
    If Single.IsNaN(A1_ / WP1_) Or Single.IsInfinity(A1_ / WP1_) Then
        R1_ = 0
    Else
        R1_ = A1_ / WP1_
    End If

    'Conveyance
    Dim tmpK As Single
    tmpK = (1 / n1) * A1_ * (R1_ ^ (2 / 3))
    If Single.IsNaN(tmpK) Or Single.IsInfinity(tmpK) Then
        K1_ = 0
    Else
        K1_ = tmpK
    End If
Case 2
    'Flow area
    A2_ = tmpArea
    'top width
    B2_ = tmpB

    '--The vertical part on the left of the ROB
    leftX = Val(arrGeom(0).Split(";")(0))
    rightX = Val(arrGeom(1).Split(";")(0))
    leftY = Val(arrGeom(0).Split(";")(1))
    rightY = Val(arrGeom(1).Split(";")(1))

    If leftX = rightX Then
        tmpLength = tmpLength - Math.Abs(rightY - leftY)
    End If
    WP2_ = tmpLength

    'hydraulic radius
    If Single.IsNaN(A2_ / WP2_) Or Single.IsInfinity(A2_ / WP2_) Then
        R2_ = 0
```

```
        Else
            R2_ = A2_ / WP2_
        End If

        'conveyance
        Dim tmpK As Single
        tmpK = (1 / n2) * A2_ * (R2_ ^ (2 / 3))
        If Single.IsNaN(tmpK) Or Single.IsInfinity(tmpK) Then
            K2_ = 0
        Else
            K2_ = tmpK
        End If
    End Select

End Sub
```

```
Public Sub CalculateCSProperties()

    CalculateSubCSProperties(0, stLOB)
    CalculateSubCSProperties(1, stMainChannel)
    CalculateSubCSProperties(2, stROB)

    'flow area
    A = A0 + A1 + A2

    If A = 0 Then
        MsgBox("xay ra truong hop Zero doi voi A, RiverCrossSection")
    End If

    'Wetted perimeter
    WP = WP0 + WP1 + WP2
    If WP = 0 Then
        MsgBox("xay ra truong hop Zero doi voi WP, RiverCrossSection")
    End If

    'hydraulic radius
    R = A / WP

    'conveyance
    K = K0 + K1 + K2
    If Single.IsNaN(K) Or Single.IsInfinity(K) Then
        MsgBox("xay ra truong hop /Zero hoac Infinity doi voi K, RiverCrossSection")
    End If

    'Friction slope
    Sf = (Q / K) ^ 2
    If Single.IsNaN(Sf) Or Single.IsInfinity(Sf) Then
        MsgBox("xay ra truong hop /Zero hoac Infinity doi voi Sf, RiverCrossSection")
    End If

    'Alfa coeficient
    Dim tmpSigma As Single
```

```
If K0 <> 0 And A0 <> 0 Then
   tmpSigma = tmpSigma + (K0 ^ 3) / (A0 ^ 2)
End If
If K1 <> 0 And A1 <> 0 Then
   tmpSigma = tmpSigma + (K1 ^ 3) / (A1 ^ 2)
End If
If K2 <> 0 And A2 <> 0 Then
   tmpSigma = tmpSigma + (K2 ^ 3) / (A2 ^ 2)
End If
Alfa = ((A ^ 2) * tmpSigma) / (K ^ 3)

'top width
B = B0 + B1 + B2

'Flow depth (not the water elevation); this had been calculated in the constructor
'flowDepth = sglWaterElevation - z

'Flow velocity
V = Q / A

'rep. Manning's n
n = ((WP0 * (n0 ^ (3 / 2)) + WP1 * (n1 ^ (3 / 2)) + WP2 * (n2 ^ (3 / 2))) / WP) ^ (2 / 3)

'Energy budget at CS
H = z + flowDepth + (Alfa * (V ^ 2)) / (2 * 9.78)


'=====================================================================
Dim dy As Single = 10 ^ (-8 / 3)
CalculateSubCSProperties_dy(0, stLOB, sglWaterElevation + dy)
CalculateSubCSProperties_dy(1, stMainChannel, sglWaterElevation + dy)
CalculateSubCSProperties_dy(2, stROB, sglWaterElevation + dy)

'flow area
A_ = A0_ + A1_ + A2_

If A_ = 0 Then
   MsgBox("xay ra truong hop Zero doi voi A, RiverCrossSection")
End If

'Wetted perimeter
WP_ = WP0_ + WP1_ + WP2_
If WP_ = 0 Then
   MsgBox("xay ra truong hop Zero doi voi WP, RiverCrossSection")
End If

'hydraulic radius
R_ = A_ / WP_

'conveyance
K_ = K0_ + K1_ + K2_
If Single.IsNaN(K_) Or Single.IsInfinity(K_) Then
   MsgBox("xay ra truong hop /Zero hoac Infinity doi voi K, RiverCrossSection")
```

```
        End If

        'Friction slope
        Sf_ = (Q / K_) ^ 2
        If Single.IsNaN(Sf_) Or Single.IsInfinity(Sf_) Then
            MsgBox("xay ra truong hop /Zero hoac Infinity doi voi Sf_, RiverCrossSection")
        End If

        'Alfa coeficient
        Dim tmpSigma_ As Single
        If K0_ <> 0 And A0_ <> 0 Then
            tmpSigma_ = tmpSigma_ + (K0_ ^ 3) / (A0_ ^ 2)
        End If
        If K1_ <> 0 And A1_ <> 0 Then
            tmpSigma_ = tmpSigma_ + (K1_ ^ 3) / (A1_ ^ 2)
        End If
        If K2_ <> 0 And A2_ <> 0 Then
            tmpSigma_ = tmpSigma_ + (K2_ ^ 3) / (A2_ ^ 2)
        End If
        K_ = K0_ + K1_ + K2_
        Alfa_ = ((A_ ^ 2) * tmpSigma_) / (K_ ^ 3)

        'top width
        B_ = B0_ + B1_ + B2_

        z_ = z

        'Flow depth (not the water elevation)
        flowDepth_ = flowDepth + dy

        'Flow velocity
        V_ = Q / A_

        'Energy budget at CS
        H_ = z_ + flowDepth_ + (Alfa_ * (V_ ^ 2)) / (2 * 9.78)

        'derivatives
        dR0dy = (R0_ - R0) / dy
        dR1dy = (R1_ - R1) / dy
        dR2dy = (R2_ - R2) / dy
        dPdy = (WP_ - WP) / dy
        dRdy = (R_ - R) / dy
        dAlfady = (Alfa_ - Alfa) / dy
        dSfdy = (Sf_ - Sf) / dy
        dHdy = (H_ - H) / dy
    End Sub
End Class
```

Appendix 6.3. RiverNetwork class

```
Public Class RiverNetwork
    'Adjacent list (stored in a jagged array) of river system hierarchy
    Public AdjList()() As String = New String()() {}
```

```
Public Function GetDownstreamCSs(ByVal CrossSectionID As String) As String()
    'get the downstream cross-section list of a cross-section
    'this is for determine if there is a split of flow from this cross-section
    'return nothing if it is a downstream cross secction

    For i As Integer = 0 To AdjList.GetUpperBound(0)
        If CrossSectionID = AdjList(i)(0) Then
            'downstream case
            If AdjList(i).Count = 1 Then
                Return Nothing
                Exit For
            End If

            Dim tmpDownstreamCSs(AdjList(i).GetUpperBound(0) - 1) As String
            For j As Integer = 0 To tmpDownstreamCSs.GetUpperBound(0)
                tmpDownstreamCSs(j) = AdjList(i)(j + 1)
            Next
            Return tmpDownstreamCSs
            Exit For
        End If
    Next
End Function
```

```
Public Function GetUpstreamCSs(ByVal CrossSectionID As String) As String()
    'get the upstream cross-section list of a cross-section
    'this is for determine if there is a combine flow into this cross-section
    'return nothing if it is a upstream cross secction

    Dim tmpCSs(0) As String
    tmpCSs(0) = "-1"

    For i As Integer = 0 To AdjList.GetUpperBound(0)
        If AdjList(i).Count > 1 Then
            For j As Integer = 1 To AdjList(i).GetUpperBound(0)
                If AdjList(i)(j) = CrossSectionID Then
                    If tmpCSs(0) = "-1" Then
                        tmpCSs(0) = AdjList(i)(0)
                    Else
                        ReDim Preserve tmpCSs(tmpCSs.Count)
                        tmpCSs(tmpCSs.GetUpperBound(0)) = AdjList(i)(0)
                    End If
                    Exit For
                End If
            Next
        End If
    Next

    If tmpCSs(0) = "-1" Then
        Return Nothing
    Else
        Return tmpCSs
```

```
    End If
  End Function
```

```
  Function UpstreamEndCSs() As String()
    Dim tmp(0) As String
    tmp(0) = "-99999"
    Dim CS As String
    For i As Integer = 0 To AdjList.GetUpperBound(0)
      CS = AdjList(i)(0)
      If IsUpstreamEnd(CS) Then
        If tmp(0) = "-99999" Then
          tmp(0) = CS
        Else
          ReDim Preserve tmp(tmp.Count)
          tmp(tmp.GetUpperBound(0)) = CS
        End If
      End If
    Next
    Return tmp
  End Function
```

```
  Private Function IsUpstreamEnd(ByVal id As String) As Boolean
    'check if a cross-section is an upstream end cross-section

    Dim chk As Boolean = True

    For i As Integer = 0 To AdjList.GetUpperBound(0)
      If AdjList(i).Count > 1 Then
        For j As Integer = 1 To AdjList(i).GetUpperBound(0)
          If AdjList(i)(j) = id Then
            chk = False
            GoTo e
          End If
        Next
      End If
    Next
    Return chk

    Exit Function
e:
    Return chk
  End Function

End Class
```

Appendix 6.4. Procedures for simulating subcritical gradual varied flow in river network

```
Imports ESRI.ArcGIS.ArcMapUI
```

```
Imports ESRI.ArcGIS.Geodatabase
Imports ESRI.ArcGIS.Carto
Imports ESRI.ArcGIS.CartoUI
Imports ESRI.ArcGIS.Display
Imports ESRI.ArcGIS.SystemUI
Imports ESRI.ArcGIS.Geometry
Imports System.Runtime.InteropServices
Imports ESRI.ArcGIS.esriSystem
Imports System.Drawing
Imports ESRI.ArcGIS.ADF
Imports System.Threading
Imports System.Windows.Forms
Imports ESRI.ArcGIS.DataSourcesRaster
Imports DotNetMatrix.GeneralMatrix
Imports ESRI.ArcGIS.Geoprocessing
Imports ESRI.ArcGIS.SpatialAnalystTools


Public Class Graduate_subcritical_simulation_frm

  Public MxDoc As IMxDocument
  Dim pMap As IMap
  Dim Key As String = "xxx\xxx\Quang NamFlood"
  Dim pMxDoc As IMxDocument
  'we dont want to re-visit upstream cross-section in a combined junction
  Public visitedCSList() As String
```

```
Private Sub NetworkSimulation()
    'On Error Resume Next
    'get CS layer
    pMap = MxDoc.FocusMap
    Dim pLayer As ILayer
    Dim pFLayer As IFeatureLayer

    For i As Integer = 0 To pMap.LayerCount - 1
      pLayer = pMap.Layer(i)
      If TypeOf (pLayer) Is IFeatureLayer And pLayer.Name = Me.cmbCrossSectionLayer.Text Then
        pFLayer = pLayer
        Exit For
      End If
    Next

    Dim pFClass As IFeatureClass
    pFClass = pFLayer.FeatureClass
    Dim CSCount As Integer = pFClass.FeatureCount(Nothing)
    Dim pFCursor As IFeatureCursor
    pFCursor = pFClass.Search(Nothing, True)
    Dim pF As IFeature
    pF = pFCursor.NextFeature

    Dim CSAdjList()() As String = New String(CSCount - 1)() {}
```

```
Dim CSGeo(CSCount - 1) As String
Dim Dis()() As String = New String(CSCount - 1)() {}
Dim CSIDs(CSCount - 1) As String
Dim n(CSCount - 1) As String
Dim q(CSCount - 1) As Single
Dim y(CSCount - 1) As Single
Dim YQ(2 * CSCount - 1)() As Double 'solution vector
Dim bnd_y(CSCount - 1) As Single 'boundary condition
Dim bnd_q(CSCount - 1) As Single 'boundary condition

Dim tmpSt As String
For i As Integer = 0 To CSCount - 1
   tmpSt = pF.Value(pF.Fields.FindField("code"))
   CSIDs(i) = tmpSt
   CSGeo(i) = pF.Value(pF.Fields.FindField("XY"))
   n(i) = pF.Value(pF.Fields.FindField("Avg_Mng"))
   Dis(i) = New String() {}
   Dis(i) = CType(pF.Value(pF.Fields.FindField("distances")), String).Split(",")
   q(i) = pF.Value(pF.Fields.FindField("q")) 'initial condition
   y(i) = pF.Value(pF.Fields.FindField("y")) 'initial condition

   If IsNumeric(pF.Value(pF.Fields.FindField("boundary_y"))) Then
      bnd_y(i) = pF.Value(pF.Fields.FindField("boundary_y"))
   End If
   If IsNumeric(pF.Value(pF.Fields.FindField("boundary_q"))) Then
      bnd_q(i) = pF.Value(pF.Fields.FindField("boundary_q"))
   End If

   'The adjacency list
   If Len(Trim(pF.Value(pF.Fields.FindField("downst_CSs")))) <> 0 Then
      tmpSt = tmpSt & "," & Trim(pF.Value(pF.Fields.FindField("downst_CSs")))
   End If
   CSAdjList(i) = New String() {}
   CSAdjList(i) = tmpSt.Split(",")
   pF = pFCursor.NextFeature
Next

'replace with zero to N, so that we can order the equation in Jacobian Matrix
Dim current As String
For i As Integer = 0 To CSCount - 1
   current = CSIDs(i)
   For j As Integer = 0 To CSCount - 1 'scan all and replace
      For k As Integer = 0 To CSAdjList(j).GetUpperBound(0)
         If CSAdjList(j)(k) = current Then
            CSAdjList(j)(k) = i
         End If
      Next
   Next
Next

'SIMULATION
Dim Jacob()() As Double = New Double(2 * CSCount - 1)() {}
Dim F()() As Double = New Double(2 * CSCount - 1)() {}
```

```
For i = 0 To (2 * CSCount - 1)
    Jacob(i) = New Double(2 * CSCount - 1) {}
    F(i) = New Double(0) {}
Next

Dim curIdx As Integer = 0
Dim net As New RiverNetwork
net.AdjList = CSAdjList
'normal reaches
Dim uID As String
Dim dId As String
Dim iteration As Integer
Dim uCS As RiverCrossSection
Dim dCS As RiverCrossSection
Dim L As Single

Do While iteration < 20
    curIdx = 0 'pointer pointing to the current row in the Jacobian matrix(for writing)
    ReDim visitedCSList(0) 'for tracking the upstream of combine flow
    visitedCSList(0) = "-9999"

    For i As Integer = 0 To CSCount - 1
        uID = i
        Dim dCSs() As String
        dCSs = net.GetDownstreamCSs(uID)

        If dCSs Is Nothing Then 'downstream end
            F(curIdx)(0) = y(uID) - bnd_y(uID)
            Jacob(curIdx)(uID) = 1
            curIdx = curIdx + 1

        ElseIf dCSs.Count > 1 Then 'split flow
            For cntDownstCSs As Integer = 0 To dCSs.Count - 1
                dId = dCSs(cntDownstCSs)
                'energy equation
                F(curIdx)(0) = y(uID) - y(dId)
                Jacob(curIdx)(uID) = 1
                Jacob(curIdx)(dId) = -1
                'continuity (Q accummulation)
                curIdx = curIdx + 1
            Next

            'continuity equation (ONE only) F = Q1 - Q2 - Q3...
            Dim tmpQSplit As Single
            tmpQSplit = q(uID)

            Jacob(curIdx)(uID + CSCount) = 1 'upstream cont. element
            For cntDownstCSs As Integer = 0 To dCSs.Count - 1
                dId = dCSs(cntDownstCSs)
                Jacob(curIdx)(dId + CSCount) = -1 'downstream cont. element
                tmpQSplit = tmpQSplit - q(dId)
            Next
            F(curIdx)(0) = tmpQSplit 'the cont. equation
```

```
        curIdx = curIdx + 1

    ElseIf dCSs.Count = 1 Then
        dId = net.GetDownstreamCSs(uID)(0)
        Dim uCSs() As String
        uCSs = net.GetUpstreamCSs(dId)

        If uCSs.Count = 1 Then 'normal channel
            If Visited(uID) = False Then 'we only need to track this for combine flows
                'two equations for each reach
                uCS = New RiverCrossSection(CSGeo(uID), n(uID), q(uID), y(uID))
                uCS.CalculateCSProperties()
                dCS = New RiverCrossSection(CSGeo(dId), n(dId), q(dId), y(dId))
                dCS.CalculateCSProperties()
                'energy balance equation
                L = Dis(uID)(0)
                F(curIdx)(0) = dCS.H - uCS.H + 0.5 * L * (uCS.Sf + dCS.Sf)
                Jacob(curIdx)(uID) = -uCS.dHdy + 0.5 * L * uCS.dSfdy
                Jacob(curIdx)(dId) = dCS.dHdy + 0.5 * L * dCS.dSfdy
                Jacob(curIdx)(uID + CSCount) = 0.5 * L * uCS.dSfdQ
                Jacob(curIdx)(dId + CSCount) = 0.5 * L * dCS.dSfdQ
                curIdx = curIdx + 1
                'continuity equation
                F(curIdx)(0) = uCS.Q - dCS.Q
                Jacob(curIdx)(uID + CSCount) = 1
                Jacob(curIdx)(dId + CSCount) = -1
                curIdx = curIdx + 1
            End If
        ElseIf uCSs.Count > 1 Then 'combine flow
            'y upstream = y downstream
            Dim tmpQ As Single
            For j As Integer = 0 To uCSs.Count - 1
                uID = uCSs(j)
                If Visited(uID) = False Then
                    F(curIdx)(0) = y(uID) - y(dId)
                    Jacob(curIdx)(uID) = 1
                    Jacob(curIdx)(dId) = -1
                    tmpQ = tmpQ + q(uID) 'accumulate upstream Qi
                    ReDim Preserve visitedCSList(visitedCSList.Count)
                    visitedCSList(visitedCSList.GetUpperBound(0)) = uID
                    curIdx = curIdx + 1
                End If
            Next
            'Continuity equation: (e.g. F = EQi-Qd=0)
            'this time we can not check visitedCSList.GetUpperBound(0)) = uID
            'anymore so we have to check for none zero Q
            If tmpQ <> 0 Then
                F(curIdx)(0) = tmpQ - q(dId)
                For k As Integer = 0 To uCSs.Count - 1
                    uID = uCSs(k)
                    Jacob(curIdx)(uID + CSCount) = 1 'upst derivatives (e.g. Q1+Q2+Q3-Q4=0)
                Next
                Jacob(curIdx)(dId + CSCount) = -1 'ONE downdstream derivative (Q4)
```

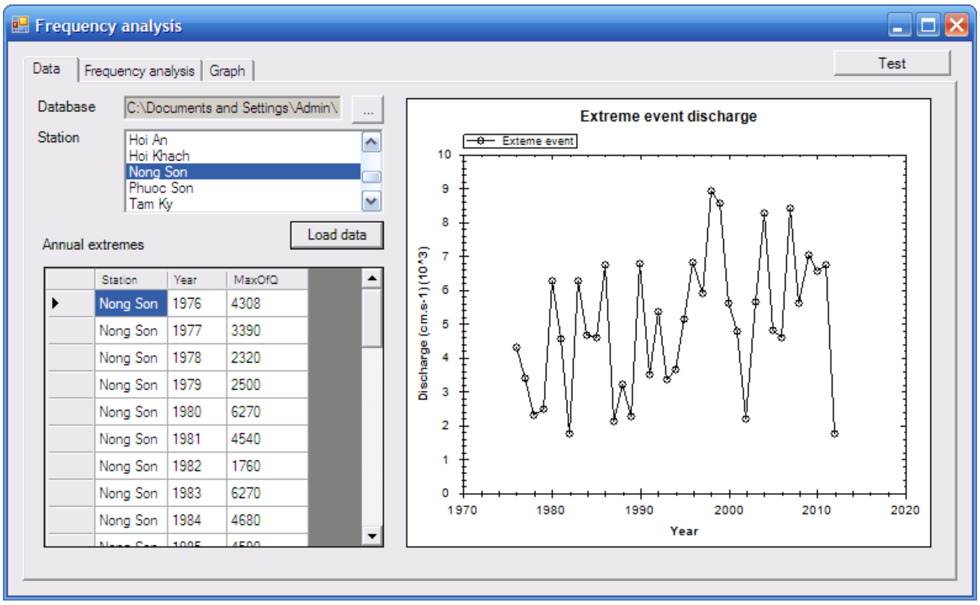
```
                tmpQ = 0 'reset this to avoid confusing
                curIdx = curIdx + 1
            End If

        End If
    End If
Next

'upstream ends
Dim upstreamEnds() As String = net.UpstreamEndCSs()
Dim tmpId As String
For i As Integer = 0 To upstreamEnds.GetUpperBound(0)
    tmpId = upstreamEnds(i)
    F(curIdx)(0) = q(tmpId) - bnd_q(tmpId)
    Jacob(curIdx)(tmpId + CSCount) = 1
    curIdx = curIdx + 1
Next

'Inverting the system
Dim DelYQ As New DotNetMatrix.GeneralMatrix(CSCount, 1)
DisplayMatrix(Jacob)

        DelYQ = ((New DotNetMatrix.GeneralMatrix(Jacob)).Inverse).Multiply(New
        DotNetMatrix.GeneralMatrix(F).Multiply(-1))

'update back
For i = 0 To CSCount - 1
    y(i) = y(i) + DelYQ.Array(i)(0)
    q(i) = q(i) + DelYQ.Array(i + CSCount)(0)
Next

'DisplayMatrix(DelYQ.Array)
displayArray(y)

If isConvergent(DelYQ.Array, 0.01) Then
    MsgBox("Converges after " & iteration & " runs")
    pFCursor = pFClass.Update(Nothing, True)
    pF = pFCursor.NextFeature
    For i As Integer = 0 To CSCount - 1
        pF.Value(pF.Fields.FindField("predict_y")) = y(i)
        pFCursor.UpdateFeature(pF)
        pF = pFCursor.NextFeature
    Next
    Exit Sub
End If

iteration = iteration + 1
Loop

pFCursor = pFClass.Update(Nothing, True)
pF = pFCursor.NextFeature
pFCursor = pFClass.Update(Nothing, True)
pF = pFCursor.NextFeature
```

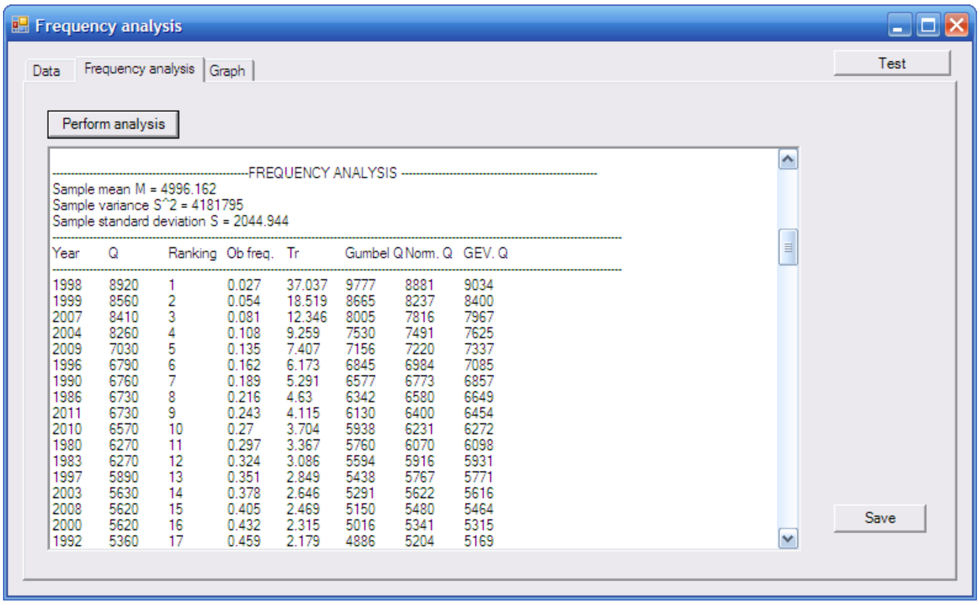
```
        For i As Integer = 0 To CSCount - 1
            pF.Value(pF.Fields.FindField("predict_y")) = y(i)
            pFCursor.UpdateFeature(pF)
            pF = pFCursor.NextFeature
        Next
        MsgBox("The result was calculated although" & _
        "Not convergent after 20 iteration", MsgBoxStyle.Critical)
    End Sub
```

```
    Private Function Visited(ByVal id As String) As Boolean
        'check if an upstream cross-section had been visited previously
        'this is for checking to avoid duplicate calculation in combine flow junctions
        On Error GoTo e
        If Me.visitedCSList Is Nothing Then
            Visited = False
            Exit Function
        End If

        Visited = False
        For i As Integer = 0 To visitedCSList.GetUpperBound(0)
            If Me.visitedCSList(i) = id Then
                Visited = True
                Exit For
            End If
        Next

        Exit Function
e:
        MsgBox(Err.Description)
    End Function
```

```
    Function isConvergent(ByRef delY()() As Double, ByVal tolerance As Single) As Boolean
        Dim tmp As Boolean = True
        For i As Integer = 0 To delY.GetUpperBound(0)
            If Math.Abs(delY(i)(0)) > Math.Abs(tolerance) Then
                tmp = False
                Exit For
            End If
        Next
        Return tmp
    End Function

End Class
```

Appendix 7.1. Frequency analysis for Maximum annual flow at Nong Son station





------------------- FREQUENCY ANALYSIS -------------------

Sample mean M = 4996.162

Sample variance S^2 = 4181795

Sample standard deviation S = 2044.944

Year	Q	Rank	Ob fq.	Tr	G.blQ	Nrm.Q	GEV.Q
1998	8920	1	0.027	37.037	9777	8881	9034
1999	8560	2	0.054	18.519	8665	8237	8400
2007	8410	3	0.081	12.346	8005	7816	7967
2004	8260	4	0.108	9.259	7530	7491	7625
2009	7030	5	0.135	7.407	7156	7220	7337
1996	6790	6	0.162	6.173	6845	6984	7085
1990	6760	7	0.189	5.291	6577	6773	6857

1986	6730	8	0.216	4.63	6342	6580	6649
2011	6730	9	0.243	4.115	6130	6400	6454
2010	6570	10	0.27	3.704	5938	6231	6272
1980	6270	11	0.297	3.367	5760	6070	6098
1983	6270	12	0.324	3.086	5594	5916	5931
1997	5890	13	0.351	2.849	5438	5767	5771
2003	5630	14	0.378	2.646	5291	5622	5616
2008	5620	15	0.405	2.469	5150	5480	5464
2000	5620	16	0.432	2.315	5016	5341	5315
1992	5360	17	0.459	2.179	4886	5204	5169
1995	5130	18	0.486	2.058	4761	5067	5024
2005	4810	19	0.514	1.946	4634	4925	4875
2001	4760	20	0.541	1.848	4513	4787	4730
1984	4680	21	0.568	1.761	4396	4651	4587
2006	4600	22	0.595	1.681	4279	4512	4442
1985	4590	23	0.622	1.608	4164	4370	4296
1981	4540	24	0.649	1.541	4048	4225	4146
1976	4308	25	0.676	1.479	3931	4074	3993
1994	3650	26	0.703	1.422	3814	3920	3836
1991	3490	27	0.73	1.37	3697	3761	3675
1977	3390	28	0.757	1.321	3575	3591	3505
1993	3360	29	0.784	1.276	3450	3413	3329
1988	3200	30	0.811	1.233	3317	3218	3136
1979	2500	31	0.838	1.193	3177	3005	2928
1978	2320	32	0.865	1.156	3028	2771	2702
1989	2280	33	0.892	1.121	2862	2500	2443
2002	2200	34	0.919	1.088	2670	2174	2137
1987	2120	35	0.946	1.057	2435	1753	1748
1982	1760	36	0.973	1.028	2105	1118	1175
2012	1750	37	1	1	-Inf	NaN	-Inf

-	-	-	-	2	4696	4996	4949
-	-	-	-	5	6478	6693	6770
-	-	-	-	10	7658	7580	7720
-	-	-	-	20	8790	8312	8477
-	-	-	-	25	9149	8526	8690
-	-	-	-	50	10255	9137	9273
-	-	-	-	100	11352	9687	9759
-	-	-	-	200	12446	10190	10165

KOLMOGOROV - SMIRNOV HYPOTHESIS TEST FOR NOMAL DISTRIBUTION FITTING

Q	F0	F1	Fn-1	D+	D-
1750.0	0.0537	0.0270	0.0000	0.0267	0.0537
1760.0	0.0543	0.0541	0.0270	0.0002	0.0272
2120.0	0.0769	0.0811	0.0541	0.0042	0.0228
2200.0	0.0828	0.1081	0.0811	0.0253	0.0017
2280.0	0.0890	0.1351	0.1081	0.0462	0.0191
2320.0	0.0922	0.1622	0.1351	0.0699	0.0429
2500.0	0.1079	0.1892	0.1622	0.0813	0.0543
3200.0	0.1865	0.2162	0.1892	0.0297	0.0027

3360.0	0.2086	0.2432	0.2162	0.0347	0.0077
3390.0	0.2129	0.2703	0.2432	0.0574	0.0304
3490.0	0.2275	0.2973	0.2703	0.0698	0.0427
3650.0	0.2522	0.3243	0.2973	0.0721	0.0451
4308.0	0.3664	0.3514	0.3243	0.0151	0.0421
4540.0	0.4105	0.3784	0.3514	0.0321	0.0592
4590.0	0.4202	0.4054	0.3784	0.0148	0.0418
4600.0	0.4221	0.4324	0.4054	0.0103	0.0167
4680.0	0.4377	0.4595	0.4324	0.0218	0.0053
4760.0	0.4534	0.4865	0.4595	0.0331	0.0061
4810.0	0.4632	0.5135	0.4865	0.0503	0.0233
5130.0	0.5265	0.5405	0.5135	0.0141	0.0129
5360.0	0.5716	0.5676	0.5405	0.0040	0.0311
5620.0	0.6215	0.5946	0.5676	0.0269	0.0539
5620.0	0.6215	0.6216	0.5946	0.0001	0.0269
5630.0	0.6234	0.6486	0.6216	0.0253	0.0017
5890.0	0.6712	0.6757	0.6486	0.0045	0.0226
6270.0	0.7362	0.7027	0.6757	0.0335	0.0605
6270.0	0.7362	0.7297	0.7027	0.0065	0.0335
6570.0	0.7825	0.7568	0.7297	0.0257	0.0527
6730.0	0.8051	0.7838	0.7568	0.0213	0.0483
6730.0	0.8051	0.8108	0.7838	0.0057	0.0213
6760.0	0.8091	0.8378	0.8108	0.0287	0.0017
6790.0	0.8132	0.8649	0.8378	0.0517	0.0247
7030.0	0.8434	0.8919	0.8649	0.0485	0.0214
8260.0	0.9472	0.9189	0.8919	0.0283	0.0553
8410.0	0.9548	0.9459	0.9189	0.0088	0.0359
8560.0	0.9614	0.9730	0.9459	0.0116	0.0155
8920.0	0.9742	1.0000	0.9730	0.0258	0.0012

Hypothesis test
Ho: The sample data is normal distributed N~(M = 4996.16, Sig = 2016.40)
H1: The sample data is not normal distributed
Kolmogorov - Smirnov test statistic D = 0.0813
With Alfa = 5% significance level, D.05 = 0.2236
There is not enough evidence to reject Ho

KOLMOGOROV - SMIRNOV HYPOTHESIS TEST FOR GUMBEL DISTRIBUTION FITTING

Q	F0	F1	Fn-1	D+	D-
1750.0	0.0109	0.0270	0.0000	0.0161	0.0109
1760.0	0.0112	0.0541	0.0270	0.0428	0.0158
2120.0	0.0282	0.0811	0.0541	0.0529	0.0259
2200.0	0.0336	0.1081	0.0811	0.0745	0.0475
2280.0	0.0398	0.1351	0.1081	0.0953	0.0683
2320.0	0.0432	0.1622	0.1351	0.1190	0.0920
2500.0	0.0606	0.1892	0.1622	0.1286	0.1015
3200.0	0.1660	0.2162	0.1892	0.0502	0.0232
3360.0	0.1975	0.2432	0.2162	0.0458	0.0187
3390.0	0.2036	0.2703	0.2432	0.0666	0.0396
3490.0	0.2246	0.2973	0.2703	0.0727	0.0456
3650.0	0.2595	0.3243	0.2973	0.0648	0.0378

4308.0	0.4117	0.3514	0.3243	0.0603	0.0873
4540.0	0.4650	0.3784	0.3514	0.0866	0.1136
4590.0	0.4762	0.4054	0.3784	0.0708	0.0979
4600.0	0.4785	0.4324	0.4054	0.0461	0.0731
4680.0	0.4963	0.4595	0.4324	0.0369	0.0639
4760.0	0.5139	0.4865	0.4595	0.0274	0.0544
4810.0	0.5247	0.5135	0.4865	0.0112	0.0382
5130.0	0.5909	0.5405	0.5135	0.0503	0.0773
5360.0	0.6347	0.5676	0.5405	0.0672	0.0942
5620.0	0.6803	0.5946	0.5676	0.0857	0.1127
5620.0	0.6803	0.6216	0.5946	0.0586	0.0857
5630.0	0.6819	0.6486	0.6216	0.0333	0.0603
5890.0	0.7229	0.6757	0.6486	0.0472	0.0743
6270.0	0.7751	0.7027	0.6757	0.0724	0.0994
6270.0	0.7751	0.7297	0.7027	0.0453	0.0724
6570.0	0.8101	0.7568	0.7297	0.0534	0.0804
6730.0	0.8268	0.7838	0.7568	0.0430	0.0701
6730.0	0.8268	0.8108	0.7838	0.0160	0.0430
6760.0	0.8298	0.8378	0.8108	0.0080	0.0190
6790.0	0.8327	0.8649	0.8378	0.0321	0.0051
7030.0	0.8546	0.8919	0.8649	0.0373	0.0103
8260.0	0.9307	0.9189	0.8919	0.0117	0.0388
8410.0	0.9368	0.9459	0.9189	0.0092	0.0178
8560.0	0.9423	0.9730	0.9459	0.0306	0.0036
8920.0	0.9539	1.0000	0.9730	0.0461	0.0191

Hypothesis test
Ho: The sample data is Gumbel distributed G~(Alfa = 1572.182, Beta = 4120.457)
H1: The sample data is not Gumbel distributed
Kolmogorov - Smirnov test statistic D = 0.1286
With Alfa = 5% significance level, D.05 = 0.2236
There is not enough evidence to reject Ho

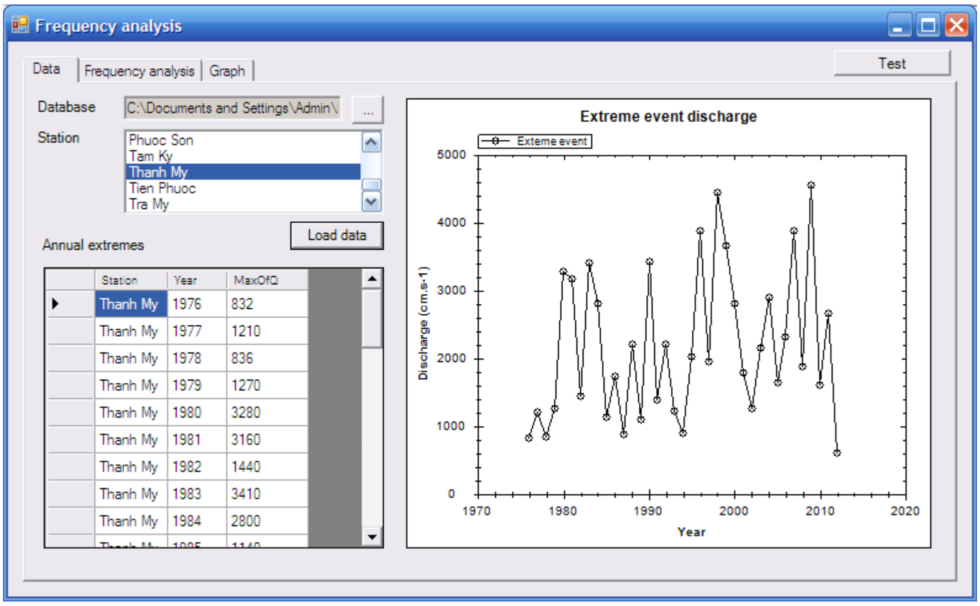
KOLMOGOROV - SMIRNOV HYPOTHESIS TEST FOR GEV DISTRIBUTION FITTING

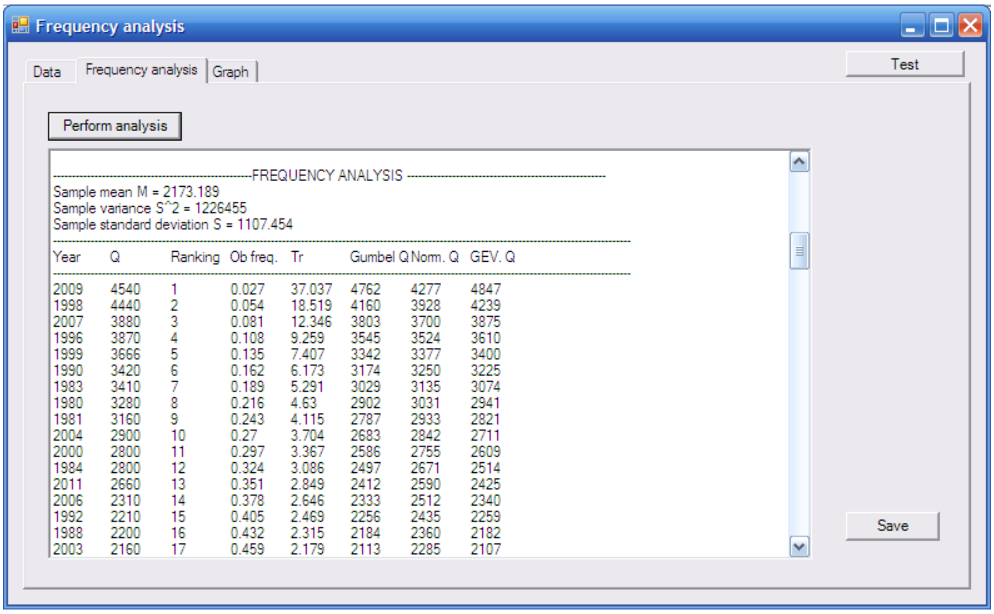
Q	F0	F1	Fn-1	D+	D-
1750.0	0.0129	0.0270	0.0000	0.0142	0.0129
1760.0	0.0133	0.0541	0.0270	0.0408	0.0137
2120.0	0.0344	0.0811	0.0541	0.0467	0.0196
2200.0	0.0410	0.1081	0.0811	0.0671	0.0401
2280.0	0.0483	0.1351	0.1081	0.0869	0.0598
2320.0	0.0522	0.1622	0.1351	0.1100	0.0830
2500.0	0.0719	0.1892	0.1622	0.1173	0.0903
3200.0	0.1768	0.2162	0.1892	0.0394	0.0124
3360.0	0.2051	0.2432	0.2162	0.0381	0.0111
3390.0	0.2105	0.2703	0.2432	0.0597	0.0327
3490.0	0.2288	0.2973	0.2703	0.0685	0.0415
3650.0	0.2584	0.3243	0.2973	0.0659	0.0389
4308.0	0.3802	0.3514	0.3243	0.0288	0.0558
4540.0	0.4212	0.3784	0.3514	0.0428	0.0698
4590.0	0.4298	0.4054	0.3784	0.0244	0.0514
4600.0	0.4315	0.4324	0.4054	0.0009	0.0261

4680.0	0.4451	0.4595	0.4324	0.0144	0.0127
4760.0	0.4585	0.4865	0.4595	0.0280	0.0010
4810.0	0.4667	0.5135	0.4865	0.0468	0.0198
5130.0	0.5172	0.5405	0.5135	0.0233	0.0037
5360.0	0.5510	0.5676	0.5405	0.0165	0.0105
5620.0	0.5867	0.5946	0.5676	0.0079	0.0191
5620.0	0.5867	0.6216	0.5946	0.0350	0.0079
5630.0	0.5880	0.6486	0.6216	0.0607	0.0336
5890.0	0.6208	0.6757	0.6486	0.0549	0.0278
6270.0	0.6642	0.7027	0.6757	0.0385	0.0115
6270.0	0.6642	0.7297	0.7027	0.0656	0.0385
6570.0	0.6947	0.7568	0.7297	0.0620	0.0350
6730.0	0.7098	0.7838	0.7568	0.0740	0.0469
6730.0	0.7098	0.8108	0.7838	0.1010	0.0740
6760.0	0.7126	0.8378	0.8108	0.1253	0.0982
6790.0	0.7153	0.8649	0.8378	0.1496	0.1226
7030.0	0.7360	0.8919	0.8649	0.1559	0.1289
8260.0	0.8187	0.9189	0.8919	0.1002	0.0732
8410.0	0.8266	0.9459	0.9189	0.1193	0.0923
8560.0	0.8341	0.9730	0.9459	0.1389	0.1119
8920.0	0.8505	1.0000	0.9730	0.1495	0.1224

Hypothesis test
Ho: The sample data is GEV distributed G~(Alfa = 2025.840, K = 0.252, Psi = 4240.120)
H1: The sample data is not GEV distributed
Kolmogorov - Smirnov test statistic D = 0.1559
With Alfa = 5% significance level, D.05 = 0.2236
There is not enough evidence to reject Ho

Appendix 7.2. Frequency analysis for maximum annual flow at Thanh My station





-----------------FREQUENCY ANALYSIS ------------------

Sample mean M = 2173.189

Sample variance S^2 = 1226455

Sample standard deviation S = 1107.454

Year	Q	Rank	Ob frq.	Tr	GblQ	NormQ	GEVQ
2009	4540	1	0.027	37.037	4762	4277	4847
1998	4440	2	0.054	18.519	4160	3928	4239
2007	3880	3	0.081	12.346	3803	3700	3875
1996	3870	4	0.108	9.259	3545	3524	3610
1999	3666	5	0.135	7.407	3342	3377	3400
1990	3420	6	0.162	6.173	3174	3250	3225
1983	3410	7	0.189	5.291	3029	3135	3074
1980	3280	8	0.216	4.63	2902	3031	2941
1981	3160	9	0.243	4.115	2787	2933	2821
2004	2900	10	0.27	3.704	2683	2842	2711
2000	2800	11	0.297	3.367	2586	2755	2609
1984	2800	12	0.324	3.086	2497	2671	2514
2011	2660	13	0.351	2.849	2412	2590	2425
2006	2310	14	0.378	2.646	2333	2512	2340
1992	2210	15	0.405	2.469	2256	2435	2259
1988	2200	16	0.432	2.315	2184	2360	2182
2003	2160	17	0.459	2.179	2113	2285	2107
1995	2030	18	0.486	2.058	2045	2211	2034
1997	1960	19	0.514	1.946	1977	2135	1960
2008	1870	20	0.541	1.848	1911	2060	1890
2001	1790	21	0.568	1.761	1848	1986	1822
1986	1730	22	0.595	1.681	1785	1911	1754
2005	1640	23	0.622	1.608	1722	1834	1686
2010	1610	24	0.649	1.541	1660	1755	1619
1982	1440	25	0.676	1.479	1596	1674	1550

1991	1380	26	0.703	1.422	1533	1590	1482
1979	1270	27	0.73	1.37	1469	1504	1413
2002	1254	28	0.757	1.321	1403	1412	1341
1993	1230	29	0.784	1.276	1336	1316	1268
1977	1210	30	0.811	1.233	1264	1210	1189
1985	1140	31	0.838	1.193	1188	1095	1106
1989	1090	32	0.865	1.156	1107	968	1018
1994	901	33	0.892	1.121	1017	821	919
1987	875	34	0.919	1.088	913	645	805
1978	836	35	0.946	1.057	786	417	665
1976	832	36	0.973	1.028	607	73	467
2012	614	37	1	1	-Inf	NaN	-Inf

0	0	0	0	2	2010	2173	1996
0	0	0	0	5	2976	3092	3018
0	0	0	0	10	3615	3572	3681
0	0	0	0	20	4227	3969	4308
0	0	0	0	25	4422	4085	4505
0	0	0	0	50	5021	4415	5105
0	0	0	0	100	5615	4713	5692
0	0	0	0	200	6208	4986	6269

KOLMOGOROV - SMIRNOV HYPOTHESIS TEST FOR NOMAL DISTRIBUTION FITTING

Q	F0	F1	Fn-1	D+	D-
614.0	0.0767	0.0270	0.0000	0.0496	0.0767
832.0	0.1097	0.0541	0.0270	0.0556	0.0827
836.0	0.1104	0.0811	0.0541	0.0293	0.0563
875.0	0.1173	0.1081	0.0811	0.0092	0.0362
901.0	0.1220	0.1351	0.1081	0.0131	0.0139
1090.0	0.1606	0.1622	0.1351	0.0015	0.0255
1140.0	0.1720	0.1892	0.1622	0.0171	0.0099
1210.0	0.1889	0.2162	0.1892	0.0273	0.0003
1230.0	0.1939	0.2432	0.2162	0.0494	0.0223
1254.0	0.2000	0.2703	0.2432	0.0703	0.0433
1270.0	0.2041	0.2973	0.2703	0.0932	0.0662
1380.0	0.2338	0.3243	0.2973	0.0905	0.0635
1440.0	0.2510	0.3514	0.3243	0.1004	0.0733
1610.0	0.3030	0.3784	0.3514	0.0754	0.0483
1640.0	0.3127	0.4054	0.3784	0.0927	0.0657
1730.0	0.3424	0.4324	0.4054	0.0900	0.0630
1790.0	0.3628	0.4595	0.4324	0.0966	0.0696
1870.0	0.3906	0.4865	0.4595	0.0958	0.0688
1960.0	0.4226	0.5135	0.4865	0.0909	0.0639
2030.0	0.4478	0.5405	0.5135	0.0927	0.0657
2160.0	0.4952	0.5676	0.5405	0.0724	0.0454
2200.0	0.5098	0.5946	0.5676	0.0848	0.0578
2210.0	0.5134	0.6216	0.5946	0.1082	0.0811
2310.0	0.5499	0.6486	0.6216	0.0988	0.0718
2660.0	0.6721	0.6757	0.6486	0.0036	0.0235
2800.0	0.7170	0.7027	0.6757	0.0143	0.0413
2800.0	0.7170	0.7297	0.7027	0.0127	0.0143
2900.0	0.7472	0.7568	0.7297	0.0096	0.0174

3160.0	0.8169	0.7838	0.7568	0.0331	0.0602
3280.0	0.8446	0.8108	0.7838	0.0338	0.0608
3410.0	0.8713	0.8378	0.8108	0.0335	0.0605
3420.0	0.8732	0.8649	0.8378	0.0084	0.0354
3666.0	0.9142	0.8919	0.8649	0.0223	0.0493
3870.0	0.9399	0.9189	0.8919	0.0210	0.0480
3880.0	0.9410	0.9459	0.9189	0.0050	0.0221
4440.0	0.9810	0.9730	0.9459	0.0081	0.0351
4540.0	0.9849	1.0000	0.9730	0.0151	0.0119

Hypothesis test
Ho: The sample data is normal distributed N~(M = 2173.19, Sig = 1092.02)
H1: The sample data is not normal distributed
Kolmogorov - Smirnov test statistic D = 0.1082
With Alfa = 5% significance level, D.05 = 0.2236
There is not enough evidence to reject Ho

KOLMOGOROV - SMIRNOV HYPOTHESIS TEST FOR GUMBEL DISTRIBUTION FITTING

Q	F0	F1	Fn-1	D+	D-
614.0	0.0280	0.0270	0.0000	0.0010	0.0280
832.0	0.0628	0.0541	0.0270	0.0087	0.0357
836.0	0.0636	0.0811	0.0541	0.0175	0.0095
875.0	0.0719	0.1081	0.0811	0.0362	0.0091
901.0	0.0779	0.1351	0.1081	0.0573	0.0302
1090.0	0.1294	0.1622	0.1351	0.0327	0.0057
1140.0	0.1454	0.1892	0.1622	0.0437	0.0167
1210.0	0.1694	0.2162	0.1892	0.0469	0.0198
1230.0	0.1765	0.2432	0.2162	0.0668	0.0397
1254.0	0.1852	0.2703	0.2432	0.0851	0.0580
1270.0	0.1911	0.2973	0.2703	0.1062	0.0792
1380.0	0.2335	0.3243	0.2973	0.0908	0.0638
1440.0	0.2578	0.3514	0.3243	0.0935	0.0665
1610.0	0.3295	0.3784	0.3514	0.0489	0.0218
1640.0	0.3424	0.4054	0.3784	0.0630	0.0359
1730.0	0.3813	0.4324	0.4054	0.0511	0.0241
1790.0	0.4072	0.4595	0.4324	0.0523	0.0253
1870.0	0.4413	0.4865	0.4595	0.0452	0.0181
1960.0	0.4791	0.5135	0.4865	0.0345	0.0074
2030.0	0.5077	0.5405	0.5135	0.0328	0.0058
2160.0	0.5589	0.5676	0.5405	0.0087	0.0183
2200.0	0.5740	0.5946	0.5676	0.0206	0.0064
2210.0	0.5777	0.6216	0.5946	0.0439	0.0169
2310.0	0.6139	0.6486	0.6216	0.0347	0.0077
2660.0	0.7237	0.6757	0.6486	0.0480	0.0750
2800.0	0.7600	0.7027	0.6757	0.0573	0.0844
2800.0	0.7600	0.7297	0.7027	0.0303	0.0573
2900.0	0.7835	0.7568	0.7297	0.0267	0.0538
3160.0	0.8354	0.7838	0.7568	0.0517	0.0787
3280.0	0.8554	0.8108	0.7838	0.0446	0.0716
3410.0	0.8746	0.8378	0.8108	0.0367	0.0637
3420.0	0.8759	0.8649	0.8378	0.0111	0.0381

3666.0	0.9055	0.8919	0.8649	0.0136	0.0407
3870.0	0.9249	0.9189	0.8919	0.0060	0.0330
3880.0	0.9257	0.9459	0.9189	0.0202	0.0068
4440.0	0.9608	0.9730	0.9459	0.0122	0.0149
4540.0	0.9651	1.0000	0.9730	0.0349	0.0079

Hypothesis test
Ho: The sample data is Gumbel distributed G~(Alfa = 851.446, Beta = 1698.934)
H1: The sample data is not Gumbel distributed
Kolmogorov - Smirnov test statistic D = 0.1062
With Alfa = 5% significance level, D.05 = 0.2236
There is not enough evidence to reject Ho

KOLMOGOROV - SMIRNOV HYPOTHESIS TEST FOR GEV DISTRIBUTION FITTING

Q	F0	F1	Fn-1	D+	D-
614.0	0.0421	0.0270	0.0000	0.0150	0.0421
832.0	0.0832	0.0541	0.0270	0.0291	0.0562
836.0	0.0841	0.0811	0.0541	0.0030	0.0301
875.0	0.0934	0.1081	0.0811	0.0147	0.0123
901.0	0.0999	0.1351	0.1081	0.0352	0.0082
1090.0	0.1542	0.1622	0.1351	0.0079	0.0191
1140.0	0.1705	0.1892	0.1622	0.0187	0.0083
1210.0	0.1944	0.2162	0.1892	0.0219	0.0052
1230.0	0.2014	0.2432	0.2162	0.0418	0.0148
1254.0	0.2100	0.2703	0.2432	0.0603	0.0333
1270.0	0.2158	0.2973	0.2703	0.0815	0.0545
1380.0	0.2568	0.3243	0.2973	0.0675	0.0405
1440.0	0.2800	0.3514	0.3243	0.0714	0.0443
1610.0	0.3473	0.3784	0.3514	0.0310	0.0040
1640.0	0.3594	0.4054	0.3784	0.0461	0.0190
1730.0	0.3953	0.4324	0.4054	0.0371	0.0101
1790.0	0.4192	0.4595	0.4324	0.0403	0.0133
1870.0	0.4505	0.4865	0.4595	0.0359	0.0089
1960.0	0.4851	0.5135	0.4865	0.0284	0.0014
2030.0	0.5114	0.5405	0.5135	0.0292	0.0022
2160.0	0.5582	0.5676	0.5405	0.0094	0.0177
2200.0	0.5721	0.5946	0.5676	0.0225	0.0045
2210.0	0.5755	0.6216	0.5946	0.0461	0.0191
2310.0	0.6088	0.6486	0.6216	0.0399	0.0128
2660.0	0.7107	0.6757	0.6486	0.0350	0.0620
2800.0	0.7450	0.7027	0.6757	0.0423	0.0694
2800.0	0.7450	0.7297	0.7027	0.0153	0.0423
2900.0	0.7674	0.7568	0.7297	0.0106	0.0377
3160.0	0.8177	0.7838	0.7568	0.0339	0.0610
3280.0	0.8375	0.8108	0.7838	0.0266	0.0537
3410.0	0.8566	0.8378	0.8108	0.0187	0.0458
3420.0	0.8580	0.8649	0.8378	0.0069	0.0201
3666.0	0.8882	0.8919	0.8649	0.0037	0.0234
3870.0	0.9086	0.9189	0.8919	0.0104	0.0167
3880.0	0.9095	0.9459	0.9189	0.0365	0.0095
4440.0	0.9481	0.9730	0.9459	0.0249	0.0021
4540.0	0.9530	1.0000	0.9730	0.0470	0.0200

Hypothesis test
Ho: The sample data is GEV distributed G~(Alfa = 919.158, K = 0.021, Psi = 1661.298)
H1: The sample data is not GEV distributed
Kolmogorov - Smirnov test statistic D = 0.0815
With Alfa = 5% significance level, D.05 = 0.2236
There is not enough evidence to reject Ho

Appendix 7.3. Frequency analysis code

```
Public Class FreqAnalysis_frm
   Public MxDoc As IMxDocument
   Dim pMap As IMap
   Dim Key As String = "Software\WWFVN\Quang NamFlood"

      Private Sub cmdOpenDatabase_Click(ByVal sender As System.Object, ByVal e As
      System.EventArgs) Handles cmdOpenDatabase.Click
      Me.OpenFileDialog1.FileName = ""
      Dim rs As DialogResult
      rs = Me.OpenFileDialog1.ShowDialog()
      If rs = Windows.Forms.DialogResult.OK Then
         Me.txtDatabasePath.Text = OpenFileDialog1.FileName
         Dim regKey As CRegistry
         regKey = New CRegistry
         regKey.WriteValue(regKey.HKeyLocalMachine, Key, "DB", Me.txtDatabasePath.Text)
      End If
   End Sub

      Private Sub FreqAnalysis_frm_Load(ByVal sender As System.Object, ByVal e As
      System.EventArgs) Handles MyBase.Load
      Dim regKey As CRegistry
      regKey = New CRegistry
      Dim databasePath As Object
      regKey.ReadValue(regKey.HKeyLocalMachine, Key, "DB", databasePath)
      Me.txtDatabasePath.Text = databasePath

      'populate stations list
      Dim tableStations As DataTable
      tableStations = DataAccess.SelectCols("Stations", "Station", "Station <> ''")
      Me.lstStations.DataSource = tableStations
      Me.lstStations.ValueMember = "Station"
      Me.lstStations.DisplayMember = "Station"
   End Sub


   Private Sub cmdLoadData_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles cmdLoadData.Click
      'On Error Resume Next

      Dim station As String = Me.lstStations.SelectedValue.ToString
```

```
    Dim querySt As String
            querySt = "SELECT Q.Station, Year([Time]) AS [Year], Max(Q.Q) AS MaxOfQ FROM Q
       GROUP BY Year([Time]), Q.Station HAVING (((Q.Station)='" & station & "'))"
    MsgBox(querySt)

    Dim qRFROQuery As DataTable
    qRFROQuery = DataAccess.RunGeneralSelectCommand(querySt)

    Dim cols As Integer = qRFROQuery.Columns.Count
    Dim rows As Integer = qRFROQuery.Rows.Count

    Me.DataGridView1.DataSource = qRFROQuery

    Me.DataGridView1.AutoResizeColumns(DataGridViewAutoSizeColumnsMode.AllCells)

    'The graph
    Dim Points As PointPairList
    Points = New PointPairList
    Dim X As String
    Dim Y As String
    For i As Integer = 0 To Me.DataGridView1.RowCount - 2 'last row is empty
      X = Me.DataGridView1.Item(getColumnIndexByName(Me.DataGridView1, "Year"), i).Value
      Y = Me.DataGridView1.Item(getColumnIndexByName(Me.DataGridView1, _
              "MaxOfQ"), i).Value
      Points.Add(X, Y)
    Next

    Me.Graph1.GraphPane.AddCurve("Exteme event", Points, Color.Black, SymbolType.Circle)
    Me.Graph1.GraphPane.Title.Text = "Extreme event discharge"
    Me.Graph1.GraphPane.XAxis.Title.Text = "Year"
    Me.Graph1.GraphPane.YAxis.Title.Text = "Discharge (cm.s-1)"
    Me.Graph1.AxisChange()
    Me.Graph1.Refresh()

  End Sub

    Friend Function getColumnIndexByName(ByRef dgv As DataGridView, ByRef colName As String)
    As Integer
    For Each column As DataGridViewColumn In dgv.Columns
      If column.Name = colName Then Return column.Index
    Next
    Try
      Throw New Exception("Column Name not Found")
    Catch ex As Exception
      MessageBox.Show(colName & ": " + ex.Message)
    End Try
    Return -1
  End Function

    Private Sub cmdPerform_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles cmdPerform.Click
    Dim A(,) As Single
    Dim B(7, 7) As Single
```

```
    'year (0), Q (1), rank (2), raw freq (3), Tr (4), Gumbel Q (5), Gumbel fq (6), Normal Q (7),
   GEV(8)
ReDim A(Me.DataGridView1.RowCount - 2, 8)

Me.ListBox1.Items.Add("")
Me.ListBox1.Items.Add("----------------------------------------- " & _
              "DATA ------------------------------------")
Me.ListBox1.Items.Add("Year" & vbTab & "Max. discharge (cms-1)")
For i As Integer = 0 To Me.DataGridView1.RowCount - 2
  A(i, 0) = Me.DataGridView1.Item(getColumnIndexByName(Me.DataGridView1, "Year"),
  i).Value
  A(i, 1) = Me.DataGridView1.Item(getColumnIndexByName(Me.DataGridView1, "MaxOfQ"),
  i).Value
  Me.ListBox1.Items.Add(A(i, 0) & vbTab & A(i, 1))
Next

Me.ListBox1.Items.Add("")
Me.ListBox1.Items.Add("-------------------------------------------------" & _
"FREQUENCY ANALYSIS ----------------------------------------------------")

'sorting the data
Dim tmp1 As Single
Dim tmp2 As Single
For i As Integer = 0 To A.GetUpperBound(0)
  tmp1 = A(i, 1)
  tmp2 = A(i, 0)
  For j As Integer = i + 1 To A.GetUpperBound(0)
    If A(j, 1) > tmp1 Then
      tmp1 = A(j, 1)
      tmp2 = A(j, 0)

      A(j, 1) = A(i, 1)
      A(j, 0) = A(i, 0)

      A(i, 1) = tmp1
      A(i, 0) = tmp2
    End If
  Next
Next

'add ranking, raw excedance freq, T and sum
Dim sum As Single
For i As Integer = 0 To A.GetUpperBound(0)
  'ranking
  A(i, 2) = i + 1
  'raw observed frequency
  A(i, 3) = Format(A(i, 2) / A.GetLength(0), "#.###")
  'raw return period
  A(i, 4) = Format(1 / A(i, 3), "#.###")
  sum = sum + A(i, 1)
Next
```

```
'Mean, variance and standard deviation
Dim mean As Single
Dim variance As Single
Dim std As Single
mean = sum / A.GetLength(0)
Me.ListBox1.Items.Add("Sample mean M = " & mean)
For i As Integer = 0 To A.GetUpperBound(0)
   variance = variance + (A(i, 1) - mean) ^ 2
Next
variance = variance / (A.GetUpperBound(0) - 1)
Me.ListBox1.Items.Add("Sample variance S^2 = " & variance)

std = variance ^ 0.5
Me.ListBox1.Items.Add("Sample standard deviation S = " & std)

'Gumbel distribution fitting
Me.ListBox1.Items.Add("-----------------------------------------------" & _
                    "-----------------------------------------------")
Me.ListBox1.Items.Add("Year" & vbTab & "Q" & vbTab & _
                "Ranking" & vbTab & "Ob freq." & vbTab & "Tr" & _
                vbTab & "Gumbel Q" & vbTab & "Norm. Q" & vbTab & "GEV. Q")

Me.ListBox1.Items.Add("-----------------------------------------------" & _
                    "-----------------------------------------------")

B(0, 4) = 2
B(1, 4) = 5
B(2, 4) = 10
B(3, 4) = 20
B(4, 4) = 25
B(5, 4) = 50
B(6, 4) = 100
B(7, 4) = 200

'Gumbel distribution
Dim V() As Single
ReDim V(A.GetUpperBound(0))
For i As Integer = 0 To V.GetUpperBound(0)
   V(i) = A(i, 1)
Next
Dim GumbelDis As GumbelDistributionFit
GumbelDis = New GumbelDistributionFit(V)

For i As Integer = 0 To A.GetUpperBound(0)
   'Gumbel Q
   A(i, 5) = Int(GumbelDis.Qt(A(i, 4)))
Next

For i As Integer = 0 To B.GetUpperBound(0)
   'Q
   B(i, 5) = Int(GumbelDis.Qt(B(i, 4)))
Next
```

```
'Normal distribution
Dim NormalDis As NormalDistributionFit
NormalDis = New NormalDistributionFit(V)
For i As Integer = 0 To A.GetUpperBound(0)
   A(i, 7) = Int(NormalDis.Qt(A(i, 4)))
Next

For i As Integer = 0 To B.GetUpperBound(0)
   B(i, 6) = Int(NormalDis.Qt(B(i, 4)))
Next

'GEV distribution
Dim GEVDis As GEVDistributionFit
GEVDis = New GEVDistributionFit(V)
For i As Integer = 0 To V.GetUpperBound(0)
   A(i, 8) = Int(GEVDis.Qt(A(i, 4)))
Next
For i As Integer = 0 To B.GetUpperBound(0)
   B(i, 7) = Int(GEVDis.Qt(B(i, 4)))
Next

'Display result
For i As Integer = 0 To A.GetUpperBound(0)
   Me.ListBox1.Items.Add(A(i, 0) & vbTab & A(i, 1) & vbTab & _
           A(i, 2) & vbTab & A(i, 3) & vbTab & A(i, 4) & _
           vbTab & A(i, 5) & vbTab & A(i, 7) & vbTab & A(i, 8))
Next



Me.ListBox1.Items.Add("---------------------------------------------" & _
              "---------------------------------------------")
For i As Integer = 0 To B.GetUpperBound(0)
   Me.ListBox1.Items.Add(B(i, 0) & vbTab & B(i, 1) & vbTab & _
                 B(i, 2) & vbTab & B(i, 3) & vbTab & B(i, 4) & _
                 vbTab & B(i, 5) & vbTab & B(i, 6) & vbTab & B(i, 7))
Next

'-------------------- drawing the graph
Dim tmp3 As Single
Dim tmp4 As Single
Dim tmp5 As Single
Dim tmp6 As Single
Dim tmp7 As Single
Dim tmp8 As Single
For i As Integer = 0 To A.GetUpperBound(0) - 1
   tmp1 = A(i, 4)
   For j As Integer = i + 1 To A.GetUpperBound(0) - 1
      If A(j, 0) < tmp1 Then
         tmp1 = A(j, 4)
         A(j, 4) = A(i, 4)
         A(i, 4) = tmp1
```

```
            tmp2 = A(j, 1)
            A(j, 1) = A(i, 1)
            A(i, 1) = tmp2

            tmp3 = A(j, 2)
            A(j, 2) = A(i, 2)
            A(i, 2) = tmp3

            tmp4 = A(j, 3)
            A(j, 3) = A(i, 3)
            A(i, 3) = tmp4

            tmp5 = A(j, 4)
            A(j, 4) = A(i, 4)
            A(i, 4) = tmp5

            tmp6 = A(j, 5)
            A(j, 5) = A(i, 5)
            A(i, 5) = tmp6

            tmp7 = A(j, 6)
            A(j, 6) = A(i, 6)
            A(i, 6) = tmp7

            tmp8 = A(j, 7)
            A(j, 7) = A(i, 7)
            A(i, 7) = tmp8
        End If
    Next
Next

Me.ListBox1.Items.Add("---------------------------------------------" & _
                      "---------------------------------------------")

Dim Points1 As PointPairList
Points1 = New PointPairList
Dim Points2 As PointPairList
Points2 = New PointPairList
Dim Points3 As PointPairList
Points3 = New PointPairList
Dim Points4 As PointPairList
Points4 = New PointPairList
Dim x As Single
Dim y As Single
For i As Integer = 0 To A.GetUpperBound(0) - 1
    X = A(i, 4)
    Y = A(i, 1)
    Points1.Add(X, Y)

    X = A(i, 4)
    Y = A(i, 5)
    Points2.Add(X, Y)
```

```
            X = A(i, 4)
            Y = A(i, 7)
            Points3.Add(X, Y)

            X = A(i, 4)
            Y = A(i, 8)
            Points4.Add(X, Y)
        Next

        Me.Graph2.GraphPane.AddCurve("Observed", Points1, Color.Red, SymbolType.Circle)
        Me.Graph2.GraphPane.AddCurve("Gumbel distribution", Points2, Color.Green,
        SymbolType.Triangle)
        Me.Graph2.GraphPane.AddCurve("Normal distribution", Points3, Color.Blue,
        SymbolType.Triangle)
        Me.Graph2.GraphPane.AddCurve("GEV distribution", Points4, Color.BlueViolet,
        SymbolType.Diamond)

        Me.Graph2.GraphPane.Title.Text = "Extreme event discharge"
        Me.Graph2.GraphPane.XAxis.Title.Text = "Return period (year)"
        Me.Graph2.GraphPane.YAxis.Title.Text = "Discharge (cm.s-1)"
        Me.Graph2.AxisChange()
        Me.Graph2.Refresh()

        '=========================================================
        'hypothesis test
        '=========================================================

        Dim test As KSTest
        test = New KSTest(V)
        Dim result() As String
        result = test.NormalDisKSTestResult
        For i As Integer = 0 To result.GetUpperBound(0)
            Me.ListBox1.Items.Add(result(i))
        Next

        result = test.GumbelDisKSTestResult
        For i As Integer = 0 To result.GetUpperBound(0)
            Me.ListBox1.Items.Add(result(i))
        Next

        result = test.GEVDisKSTestResult
        For i As Integer = 0 To result.GetUpperBound(0)
            Me.ListBox1.Items.Add(result(i))
        Next
    End Sub

    Private Sub cmdSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles cmdSave.Click
        Me.SaveFileDialog1.FileName = ""
        Dim rs As DialogResult
        rs = Me.SaveFileDialog1.ShowDialog()
        Dim fileFullName As String
        If rs = Windows.Forms.DialogResult.OK Then
```

```
        fileFullName = Me.SaveFileDialog1.FileName
      Else
        Exit Sub
      End If

      Dim objWriter As New System.IO.StreamWriter(fileFullName, False)

      Dim i As Integer
      Dim st As String
      For i = 0 To Me.ListBox1.Items.Count - 1
        st = Me.ListBox1.GetItemText(Me.ListBox1.Items.Item(i))
        objWriter.WriteLine(st)
      Next
      objWriter.Close()
    End Sub
```

```
Imports System.Math

Public Class GEVDistributionFit
    Public N As Integer
    Public M100, M110, M120, M130 As Single
    Public Lam1, Lam2, Lam3, Lam4 As Single
    Public t2 As Single 'L-CV
    Public t3 As Single 'L-Skewness
    Public t4 As Single 'L-Kurtosis
    'parameters
    Public k As Double
    Public c As Double
    Public Alfa As Double
    Public pSy As Double
    Public Mean As Single
    Public StandardDev As Single

    Public Sub New(ByVal A() As Single)
      'sort the data
      Dim tmp As Single
      N = A.GetUpperBound(0) + 1
      For i As Integer = 0 To N - 2
        tmp = A(i)
        For j As Integer = i + 1 To N - 1
          If A(j) < tmp Then
            tmp = A(j)
            A(j) = A(i)
            A(i) = tmp
          End If
        Next
      Next

      'mean
      For i As Integer = 0 To N - 1
        Mean = Mean + A(i)
      Next
      Mean = Mean / N
```

```
    'standard deviation
    For i As Integer = 0 To N - 1
        tmp = tmp + (A(i) - Mean) ^ 2
    Next
    tmp = tmp / (N - 1)
    tmp = tmp ^ 0.5
    StandardDev = tmp

    M100 = Mean

    For i As Integer = 0 To N - 1
        tmp = i / (N - 1)
        tmp = tmp * A(i)
        M110 = M110 + tmp
    Next
    M110 = M110 / N

    For i As Integer = 0 To N - 1
        tmp = i * (i - 1)
        tmp = tmp / ((N - 1) * (N - 2))
        tmp = tmp * A(i)
        M120 = M120 + tmp
    Next
    M120 = M120 / N

    For i As Integer = 0 To N - 1
        tmp = i * (i - 1) * (i - 2)
        tmp = tmp / ((N - 1) * (N - 2) * (N - 3))
        tmp = tmp * A(i)
        M130 = M130 + tmp
    Next
    M130 = M130 / N

    Lam1 = M100
    Lam2 = 2 * M110 - M100
    Lam3 = 6 * M120 - 6 * M110 + M100
    Lam4 = 20 * M130 - 30 * M120 + 12 * M110 - M100

    t2 = Lam2 / Lam1
    t3 = Lam3 / Lam2
    t4 = Lam4 / Lam2

    c = (2 / (3 + t3)) - (Math.Log(2) / Math.Log(3))
    k = 7.859 * c + 2.9554 * (c ^ 2)
    Dim gama As Double
    gama = XAlglib.gammafunction(1 + k)
    'scale parameter
    Alfa = (Lam2 * k) / ((1 - (2 ^ (-k))) * gama)
    pSy = Lam1 - (Alfa * (1 - gama)) / k

End Sub

Public Function Qt(ByVal t As Single) As Single
```

```
        Dim tmp As Single
        tmp = pSy + (Alfa / k) * (1 - (-Math.Log(1 - 1 / t)) ^ (k))
        Return tmp
    End Function

    Public Function CummulativeProb(ByVal x As Single) As Single
        Return Exp(-(1 + k * ((x - pSy) / Alfa)) ^ (-1 / k))
    End Function

End Class

Public Class GumbelDistributionFit
    Public N As Integer
    Public Mean As Single
    Public StandardDev As Single
    Public Alfa As Single
    Public Beta As Single

    Public Sub New(ByVal A() As Single)
        'sort the data
        Dim tmp As Single
        N = A.GetUpperBound(0) + 1
        For i As Integer = 0 To N - 2
            tmp = A(i)
            For j As Integer = i + 1 To N - 1
                If A(j) < tmp Then
                    tmp = A(j)
                    A(j) = A(i)
                    A(i) = tmp
                End If
            Next
        Next

        'mean
        For i As Integer = 0 To N - 1
            Mean = Mean + A(i)
        Next
        Mean = Mean / N
        'standard deviation
        For i As Integer = 0 To N - 1
            tmp = tmp + (A(i) - Mean) ^ 2
        Next
        tmp = tmp / (N - 1)
        tmp = tmp ^ 0.5
        StandardDev = tmp

        Alfa = ((6 ^ 0.5) * StandardDev) / 3.14159
        Beta = Mean - 0.557 * Alfa

    End Sub

    Public Function Qt(ByVal t As Single) As Single
        Dim x As Single
```

```
        Dim y As Single
        y = -Log(-Log(1 - (1 / t)))
        x = Beta + Alfa * y
        Return x
    End Function

    Public Function Frequency(ByVal t As Single) As Single
        'cummulative distribution function
        Return Exp(-Exp((Beta - Qt(t)) / Alfa))
    End Function

    Public Function Frequency2(ByVal Qt As Single) As Single
        'cummulative distribution function
        Return Exp(-Exp((Beta - Qt) / Alfa))
    End Function
End Class

Public Class NormalDistributionFit
    Dim Mean As Single
    Dim StandardDev As Single
    Dim N As Integer

    Public Sub New(ByVal A() As Single)
        'sort the data
        Dim tmp As Single
        N = A.GetUpperBound(0) + 1
        For i As Integer = 0 To N - 2
            tmp = A(i)
            For j As Integer = i + 1 To N - 1
                If A(j) < tmp Then
                    tmp = A(j)
                    A(j) = A(i)
                    A(i) = tmp
                End If
            Next
        Next

        'mean
        For i As Integer = 0 To N - 1
            Mean = Mean + A(i)
        Next
        Mean = Mean / N
        'standard deviation
        For i As Integer = 0 To N - 1
            tmp = tmp + (A(i) - Mean) ^ 2
        Next
        tmp = tmp / (N - 1)
        tmp = tmp ^ 0.5
        StandardDev = tmp


    End Sub
```

```
Public Function Qt(ByVal t As Single) As Single
    Dim Cv As Single = StandardDev / Mean
    Dim K As Single 'frequency factor
    Dim z As Single
    z = 1 - (1 / t)
    K = InvertNorm(z)
    Return Int(Mean * (1 + Cv * K))
End Function


Public Function InvertNorm(ByVal p As Single) As Single
    'Calculate a value with a given accumulative probability p?

    Dim c0 As Single = 2.6539620026016846 '2.515517
    Dim c1 As Single = 1.5615337002120804 '0.802853
    Dim c2 As Single = 0.061146735765196999 '0.010328
    Dim d1 As Single = 1.9048751828364987 '1.432788
    Dim d2 As Single = 0.45405553644423352 '0.189269
    Dim d3 As Single = 0.0095477453270689447 '0.001308

    Dim t As Single
    If 1 - p < 0.5 Then
        t = Log(1 / ((1 - p) ^ 2)) ^ 0.5
    Else
        t = Log(1 / (p ^ 2)) ^ 0.5
    End If

    Dim xp As Single
    xp = t - (c0 + c1 * t + c2 * (t ^ 2)) / (1 + d1 * t + d2 * (t ^ 2) + d3 * (t ^ 3))

    If 1 - p >= 0.5 Then
        xp = -xp
    End If

    Return xp
End Function

Public Function NormDist(ByVal x As Double) As Double
    'aproximating the cummulative probability P(x<Xt)
    'in normal distribution

    Dim t As Double
    Const b1 = 0.31938153
    Const b2 = -0.356563782
    Const b3 = 1.781477937
    Const b4 = -1.821255978
    Const b5 = 1.330274429
    Const p = 0.2316419
    Const c = 0.39894228

    If x >= 0 Then
        t = 1.0# / (1.0# + p * x)
        NormDist = (1.0# - c * Math.Exp(-x * x / 2.0#) * t * (t * (t * (t * (t * b5 + b4) + b3) + b2) + b1))
```

```
        Else
            t = 1.0# / (1.0# - p * x)
            NormDist = (c * Math.Exp(-x * x / 2.0#) * t * (t * (t * (t * (t * b5 + b4) + b3) + b2) + b1))
        End If
    End Function

End Class
```

```
Imports System.Math
Public Class KSTest
    '===========================================================================
    'KOLMOGOROV - SMIRNOV TEST FOR DISTRIBUTION CONFORMAL
    '===========================================================================

    Public Mean, StandardDev As Single
    Public N As Integer
    Private A_() As Single
    Public NormalTestTable(,) As Single
    Public GumbelTestTable(,) As Single
    Public GEVTestTable(,) As Single
    Public GumbelBeta, GumbelAlfa As Single
    Public GEVAlfa, GEV_k, GEVPsy As Single
    Public NormalKSTestStatistic As Single
    Public GumbelKSTestStatistic As Single
    Public GEVKSTestStatistic As Single

    Public Sub New(ByVal A() As Single)
        A_ = A

        'sort A_
        Dim tmp As Single
        N = A_.GetUpperBound(0) + 1
        For i As Integer = 0 To N - 2
            tmp = A_(i)
            For j As Integer = i + 1 To N - 1
                If A_(j) < tmp Then
                    tmp = A_(j)
                    A_(j) = A_(i)
                    A_(i) = tmp
                End If
            Next
        Next

        'mean
        For i As Integer = 0 To N - 1
            Mean = Mean + A_(i)
        Next
        Mean = Mean / N

        'standard deviation
        For i As Integer = 0 To N - 1
            tmp = tmp + (A_(i) - Mean) ^ 2
        Next
```

```
    tmp = tmp / (N - 1)
    tmp = tmp ^ 0.5
    StandardDev = tmp
End Sub

'===========================================================================
'TEST FOR NORMAL DISTRIBUTION CONFORMAL
'===========================================================================

Public Function NormalDisCumulativeProbabilityArray() As Single()
    'return cummulative probability list of Normal distribution
    Dim Z() As Single
    ReDim Z(A_.GetUpperBound(0))
    For i As Integer = 0 To A_.GetUpperBound(0)
        Z(i) = (A_(i) - Mean) / StandardDev
    Next
    Dim B(Z.GetUpperBound(0)) As Single
    Dim tmp As NormalDistribution
    tmp = New NormalDistribution
    For i As Integer = 0 To B.GetUpperBound(0)
        B(i) = tmp.NormDist(Z(i))
    Next

    Return B
End Function

Public Function CumulativeFrequencyArray()
    Dim S(A_.GetUpperBound(0)) As Single
    For i As Integer = 0 To S.GetUpperBound(0)
        S(i) = (i + 1) / N
    Next
    Return S
End Function

Public Function Normal_Data_Theory_Observed_AccProbTable() As Single(,)
    Dim X(A_.GetUpperBound(0), 2) As Single
    Dim A() As Single
    Dim B() As Single
    A = NormalDisCumulativeProbabilityArray()
    B = CumulativeFrequencyArray()
    For i As Integer = 0 To A_.GetUpperBound(0)
        X(i, 0) = A_(i)
        X(i, 1) = A(i)
        X(i, 2) = B(i)
    Next
    Return X
End Function

Public Function Normal_KS_DStatistic() As Single
    'Kolmogorov - Smirnov test statistic
    Dim X(A_.GetUpperBound(0), 5) As Single
    Dim A() As Single
    Dim B() As Single
```

```
        A = NormalDisCumulativeProbabilityArray()
        B = CumulativeFrequencyArray()
        For i As Integer = 0 To A_.GetUpperBound(0)
            X(i, 0) = A_(i)
            X(i, 1) = A(i)
            X(i, 2) = B(i)
            If i <> 0 Then
                X(i, 3) = X(i - 1, 2)
            Else
                X(i, 3) = 0.0
            End If
            X(i, 4) = Abs(X(i, 2) - X(i, 1))
            X(i, 5) = Abs(X(i, 1) - X(i, 3))
        Next

        Dim max1 As Single = X(0, 4)
        Dim max2 As Single = X(0, 5)
        For i As Integer = 0 To X.GetUpperBound(0)
            If X(i, 4) > max1 Then
                max1 = X(i, 4)
            End If
            If X(i, 5) > max2 Then
                max2 = X(i, 5)
            End If
        Next

        ReDim NormalTestTable(X.GetUpperBound(0), X.GetUpperBound(1))
        NormalTestTable = X

        If max1 > max2 Then
            Return max1
        Else
            Return max2
        End If

    End Function

    Function NormalDisKSTestResult() As String()
        'this step really carries out the test
        NormalKSTestStatistic = Normal_KS_DStatistic()

        Dim X(2) As String
        X(0) = ""
        X(1) = "KOLMOGOROV - SMIRNOV HYPOTHESIS TEST FOR NOMAL DISTRIBUTION
FITTING"
        X(2) = "Q" & vbTab & "F0" & vbTab & "F1" & vbTab & "Fn-1" & vbTab & "D+" & vbTab & "D-"
        Dim tmp As String
        For i As Integer = 0 To NormalTestTable.GetUpperBound(0)
            tmp = ""
            For j As Integer = 0 To NormalTestTable.GetUpperBound(1)
                If j = 0 Then
                    tmp = tmp & Format(NormalTestTable(i, j), "0.0")
                Else
```

```
                tmp = tmp & vbTab & Format(NormalTestTable(i, j), "0.0000")
            End If
        Next
        ReDim Preserve X(X.Count)
        X(X.GetUpperBound(0)) = tmp
    Next

    ReDim Preserve X(X.Count)
    X(X.GetUpperBound(0)) = "--------------------------------------------------------"

    ReDim Preserve X(X.Count)
    X(X.GetUpperBound(0)) = "Hypothesis test"

    ReDim Preserve X(X.Count)
    X(X.GetUpperBound(0)) = "Ho: The sample data is normal distributed N~( M = " & _
    Format(Mean, "0.00") & ", Sig = " & _
    Format(StandardDev, "0.00") & ")"

    ReDim Preserve X(X.Count)
    X(X.GetUpperBound(0)) = "H1: The sample data is not normal distributed"

    ReDim Preserve X(X.Count)
    X(X.GetUpperBound(0)) = "Kolmogorov - Smirnov test statistic D = " &
Format(Normal_KS_DStatistic(), "0.0000")

    ReDim Preserve X(X.Count)
    X(X.GetUpperBound(0)) = "With Alfa = 5% significance level, D.05 = " & Format(1.36 / (N ^ 0.5),
"0.0000")

    If Normal_KS_DStatistic() <= 1.36 / (N ^ 0.5) Then
        ReDim Preserve X(X.Count)
        X(X.GetUpperBound(0)) = "There is not enough evidence to reject Ho"
    Else
        ReDim Preserve X(X.Count)
        X(X.GetUpperBound(0)) = "There is not enough evidence to accept Ho"
    End If

    ReDim Preserve X(X.Count)
    X(X.GetUpperBound(0)) = "----------------------------------------------"

    ReDim Preserve X(X.Count)
    X(X.GetUpperBound(0)) = ""

    Return X
End Function

'==========================================================================
'TEST FOR GUMBEL DISTRIBUTION CONFORMAL
'==========================================================================

Public Function GumbelDisCummulativeProbabilityArray() As Single()
    'return cummulative probability list of Gumbel distribution
```

```
    Dim Z() As Single
    ReDim Z(A_.GetUpperBound(0))

    Dim tmp As GumbelDistributionFit
    tmp = New GumbelDistributionFit(A_)
    GumbelBeta = tmp.Beta
    GumbelAlfa = tmp.Alfa

    For i As Integer = 0 To Z.GetUpperBound(0)
        Z(i) = tmp.Frequency2(A_(i))
    Next

    Return Z
End Function

Public Function Gumbel_Data_Theory_Observed_AccProbTable() As Single(,)
    Dim X(A_.GetUpperBound(0), 2) As Single
    Dim A() As Single
    Dim B() As Single
    A = GumbelDisCummulativeProbabilityArray()
    B = CumulativeFrequencyArray()
    For i As Integer = 0 To A_.GetUpperBound(0)
        X(i, 0) = A_(i)
        X(i, 1) = A(i)
        X(i, 2) = B(i)
    Next
    Return X
End Function

Public Function Gumbel_KS_DStatistic() As Single
    'Kolmogorov - Smirnov test statistic
    Dim X(A_.GetUpperBound(0), 5) As Single
    Dim A() As Single
    Dim B() As Single
    A = GumbelDisCummulativeProbabilityArray()
    B = CumulativeFrequencyArray()
    For i As Integer = 0 To A_.GetUpperBound(0)
        X(i, 0) = A_(i)
        X(i, 1) = A(i)
        X(i, 2) = B(i)
        If i <> 0 Then
            X(i, 3) = X(i - 1, 2)
        Else
            X(i, 3) = 0.0
        End If
        X(i, 4) = Abs(X(i, 2) - X(i, 1))
        X(i, 5) = Abs(X(i, 1) - X(i, 3))
    Next

    Dim max1 As Single = X(0, 4)
    Dim max2 As Single = X(0, 5)
    For i As Integer = 0 To X.GetUpperBound(0)
        If X(i, 4) > max1 Then
```

```
            max1 = X(i, 4)
         End If
         If X(i, 5) > max2 Then
            max2 = X(i, 5)
         End If
      Next

      ReDim GumbelTestTable(X.GetUpperBound(0), X.GetUpperBound(1))
      GumbelTestTable = X

      If max1 > max2 Then
         Return max1
      Else
         Return max2
      End If

   End Function

   Function GumbelDisKSTestResult() As String()
      'this step really carries out the test
      GumbelKSTestStatistic = Gumbel_KS_DStatistic()

      Dim X(2) As String
      X(0) = ""
      X(1) = "KOLMOGOROV - SMIRNOV HYPOTHESIS TEST FOR GUMBEL DISTRIBUTION
FITTING"
      X(2) = "Q" &vbTab & "F0" & vbTab & "F1" & vbTab & "Fn-1" & vbTab & "D+" & vbTab & "D-"
      Dim tmp As String
      For i As Integer = 0 To GumbelTestTable.GetUpperBound(0)
         tmp = ""
         For j As Integer = 0 To GumbelTestTable.GetUpperBound(1)
            If j = 0 Then
               tmp = tmp & Format(GumbelTestTable(i, j), "0.0")
            Else

               tmp = tmp & vbTab & Format(GumbelTestTable(i, j), "0.0000")
            End If
         Next
         ReDim Preserve X(X.Count)
         X(X.GetUpperBound(0)) = tmp
      Next

      ReDim Preserve X(X.Count)
      X(X.GetUpperBound(0)) = "--------------------------------------------"

      ReDim Preserve X(X.Count)
      X(X.GetUpperBound(0)) = "Hypothesis test"

      ReDim Preserve X(X.Count)
      X(X.GetUpperBound(0)) = "Ho: The sample data is Gumbel distributed G~( Alfa = " & _
      Format(GumbelAlfa, "0.000") & ", Beta = " & _
      Format(GumbelBeta, "0.000") & ")"
```

```
    ReDim Preserve X(X.Count)
    X(X.GetUpperBound(0)) = "H1: The sample data is not Gumbel distributed"

    ReDim Preserve X(X.Count)
        X(X.GetUpperBound(0)) = "Kolmogorov - Smirnov test statistic D = " &
        Format(Gumbel_KS_DStatistic(), "0.0000")

    ReDim Preserve X(X.Count)
        X(X.GetUpperBound(0)) = "With Alfa = 5% significance level, D.05 = " & Format(1.36 / (N ^
        0.5), "0.0000")

    If Normal_KS_DStatistic() <= 1.36 / (N ^ 0.5) Then
      ReDim Preserve X(X.Count)
      X(X.GetUpperBound(0)) = "There is not enough evidence to reject Ho"
    Else
      ReDim Preserve X(X.Count)
      X(X.GetUpperBound(0)) = "There is not enough evidence to accept Ho"
    End If

    ReDim Preserve X(X.Count)
    X(X.GetUpperBound(0)) = "---------------------------------------------"

    ReDim Preserve X(X.Count)
    X(X.GetUpperBound(0)) = ""

    Return X
  End Function

  '=========================================================================
  'TEST FOR GEV DISTRIBUTION CONFORMAL
  '=========================================================================

  Public Function GEVDisCummulativeProbabilityArray() As Single()
    'return cummulative probability list of GEV distribution
    Dim Z() As Single
    ReDim Z(A_.GetUpperBound(0))

    Dim tmp As GEVDistributionFit
    tmp = New GEVDistributionFit(A_)
    GEVAlfa = tmp.Alfa
    GEV_k = tmp.k
    GEVPsy = tmp.pSy

    For i As Integer = 0 To Z.GetUpperBound(0)
      Z(i) = tmp.CummulativeProb(A_(i))
    Next
    Return Z
  End Function

  Public Function GEV_Data_Theory_Observed_AccProbTable() As Single(,)
    Dim X(A_.GetUpperBound(0), 2) As Single
    Dim A() As Single
    Dim B() As Single
```

```
    A = GEVDisCummulativeProbabilityArray()
    B = CumulativeFrequencyArray()
    For i As Integer = 0 To A_.GetUpperBound(0)
        X(i, 0) = A_(i)
        X(i, 1) = A(i)
        X(i, 2) = B(i)
    Next
    Return X
End Function

Public Function GEV_KS_DStatistic() As Single
    'Kolmogorov - Smirnov test statistic
    Dim X(A_.GetUpperBound(0), 5) As Single
    Dim A() As Single
    Dim B() As Single
    A = GEVDisCummulativeProbabilityArray()
    B = CumulativeFrequencyArray()
    For i As Integer = 0 To A_.GetUpperBound(0)
        X(i, 0) = A_(i)
        X(i, 1) = A(i)
        X(i, 2) = B(i)
        If i <> 0 Then
            X(i, 3) = X(i - 1, 2)
        Else
            X(i, 3) = 0.0
        End If
        X(i, 4) = Abs(X(i, 2) - X(i, 1))
        X(i, 5) = Abs(X(i, 1) - X(i, 3))
    Next

    Dim max1 As Single = X(0, 4)
    Dim max2 As Single = X(0, 5)
    For i As Integer = 0 To X.GetUpperBound(0)
        If X(i, 4) > max1 Then
            max1 = X(i, 4)
        End If
        If X(i, 5) > max2 Then
            max2 = X(i, 5)
        End If
    Next

    ReDim GEVTestTable(X.GetUpperBound(0), X.GetUpperBound(1))
    GEVTestTable = X


    If max1 > max2 Then
        Return max1
    Else
        Return max2
    End If

End Function
```

```
Function GEVDisKSTestResult() As String()
    'this step really carries out the test
    GEVKSTestStatistic = GEV_KS_DStatistic()

    Dim X(2) As String
    X(0) = ""
    X(1) = "KOLMOGOROV - SMIRNOV HYPOTHESIS TEST FOR GEV DISTRIBUTION FITTING"
    X(2) = "Q" & vbTab & "F0" & vbTab &"F1" & vbTab & "Fn-1" & vbTab & "D+" & vbTab & "D-"
    Dim tmp As String
    For i As Integer = 0 To GEVTestTable.GetUpperBound(0)
        tmp = ""
        For j As Integer = 0 To GEVTestTable.GetUpperBound(1)
            If j = 0 Then
                tmp = tmp & Format(GEVTestTable(i, j), "0.0")
            Else

                tmp = tmp & vbTab & Format(GEVTestTable(i, j), "0.0000")
            End If
        Next
        ReDim Preserve X(X.Count)
        X(X.GetUpperBound(0)) = tmp
    Next

    ReDim Preserve X(X.Count)
    X(X.GetUpperBound(0)) = "----------------------------------------------"

    ReDim Preserve X(X.Count)
    X(X.GetUpperBound(0)) = "Hypothesis test"

    ReDim Preserve X(X.Count)
    X(X.GetUpperBound(0)) = "Ho: The sample data is GEV distributed G~( Alfa = " & _
    Format(GEVAlfa, "0.000") & ", K = " & _
    Format(GEV_k, "0.000") & ", Psi = " & _
    Format(GEVPsy, "0.000") & ")"

    ReDim Preserve X(X.Count)
    X(X.GetUpperBound(0)) = "H1: The sample data is not GEV distributed"

    ReDim Preserve X(X.Count)
        X(X.GetUpperBound(0)) = "Kolmogorov - Smirnov test statistic D = " &
        Format(GEV_KS_DStatistic(), "0.0000")

    ReDim Preserve X(X.Count)
        X(X.GetUpperBound(0)) = "With Alfa = 5% significance level, D.05 = " & Format(1.36 / (N ^
        0.5), "0.0000")

    If Normal_KS_DStatistic() <= 1.36 / (N ^ 0.5) Then
        ReDim Preserve X(X.Count)
        X(X.GetUpperBound(0)) = "There is not enough evidence to reject Ho"
    Else
        ReDim Preserve X(X.Count)
        X(X.GetUpperBound(0)) = "There is not enough evidence to accept Ho"
    End If
```

```
        ReDim Preserve X(X.Count)
        X(X.GetUpperBound(0)) = "---------------------------------------------"

        ReDim Preserve X(X.Count)
        X(X.GetUpperBound(0)) = ""

        Return X
    End Function

End Class
```

```
Imports System.Math

Public Class NormalDistribution

    Public Function InvertNorm(ByVal p As Single) As Single
        'Calculate a value with a given accumulative probability p?

        Dim c0 As Single = 2.6539620026016846 '2.515517
        Dim c1 As Single = 1.5615337002120804 '0.802853
        Dim c2 As Single = 0.061146735765196999 '0.010328
        Dim d1 As Single = 1.9048751828364987 '1.432788
        Dim d2 As Single = 0.45405553644423352 '0.189269
        Dim d3 As Single = 0.0095477453270689447 '0.001308

        Dim t As Single
        If 1 - p < 0.5 Then
            t = Log(1 / ((1 - p) ^ 2)) ^ 0.5
        Else
            t = Log(1 / (p ^ 2)) ^ 0.5
        End If

        Dim xp As Single
        xp = t - (c0 + c1 * t + c2 * (t ^ 2)) / (1 + d1 * t + d2 * (t ^ 2) + d3 * (t ^ 3))

        If 1 - p >= 0.5 Then
            xp = -xp
        End If

        Return xp
    End Function

    Public Function NormDist(ByVal x As Double) As Double
        'aproximating the cummulative probability P(x<Xt)
        'in normal distribution

        Dim t As Double
        Const b1 = 0.31938153
        Const b2 = -0.356563782
        Const b3 = 1.781477937
        Const b4 = -1.821255978
```

```
        Const b5 = 1.330274429
        Const p = 0.2316419
        Const c = 0.39894228

        If x >= 0 Then
            t = 1.0# / (1.0# + p * x)
            NormDist = (1.0# - c * Math.Exp(-x * x / 2.0#) * t * (t * (t * (t * (t * b5 + b4)
            + b3) + b2) + b1))
        Else
            t = 1.0# / (1.0# - p * x)
            NormDist = (c * Math.Exp(-x * x / 2.0#) * t * (t * (t * (t * (t * b5 + b4) + b3) +
            b2) + b1))
        End If
    End Function

End Class
```