FERMENTATION CONTROL AND MODELING WITH APPLICATION IN
PRODUCTION OF THERMO-STABLE ALPHA-AMYLASE USING
RECOMBINANT BACILLUS SUBTILIS

A dissertation presented to
the faculty of
the Fritz J. and Dolores H. Russ
College of Engineering and Technology of Ohio University

In partial fulfillment
of the requirements for the degree
Doctor of Philosophy

Hanjing Huang
November 2002
This dissertation entitled
FERMENTATION CONTROL AND MODELING WITH APPLICATION IN
PRODUCTION OF THERMO-STABLE ALPHA-AMYLASE USING
RECOMBINANT Bacillus subtilis

BY
Hanjing Huang

has been approved for
the Department of Chemical Engineering
and the Russ College of Engineering and Technology by

Tingyue Gu
Associate Professor of Chemical Engineering

Dennis Irwin
Dean, Russ College of Engineering and Technology
ABSTRACT

HUANG, HANJING. Ph.D. November 2002. Chemical Engineering

Fermentation Control and Modeling with Application in Production of Thermo-stable 
Alpha-amylase using Recombinant *Bacillus subtilis* (171pp.)

Director of Dissertation: Tingyue Gu

*Bacillus subtilis* is an attractive host for the production of heterologous proteins. Fed-batch fermentation can attain a high cell density by avoiding substrate inhibition and accumulation of inhibitory metabolites such as acetate. In a fed-batch fermentation, the concentration of glucose or the specific growth rate is commonly used as an index in controlling the addition of the substrate (i.e., glucose) in order to maintain a stable, low concentration of the substrate during the entire process. A computer controlled system is required to monitor or control the limiting index when the specific fed-batch strategy is applied to the high cell density culture.

In this work, a computer controlled system using LabVIEW software was established. It achieved flexible and automatic monitoring and control of process parameters.

A dual exponential fed-batch culture strategy was applied to minimize the formation of acetate and to control the glucose concentration and specific growth rate at the predetermined values. Due to low solubilities of tyrosine and tryptophan in Feed Stream 1 containing concentrated glucose and other nutrients, tyrosine and tryptophan were removed from Feed Stream 1 and were dissolved in 14.4% ammonium water to
form Feed Stream 2. By dual feeding both Stream 1 and Stream 2 at different exponential feed rates, the cells grew exponentially and a high cell density of 24.2 g/l and a final alpha-amylase activity of 71.4 U/ml were achieved. The overall biomass yield was 0.39 g cell/g glucose. In comparison, for the batch culture with the initial glucose concentration of 8 g/l, the final cell density was 2.3 g/l, and alpha-amylase concentration was 1.5 U/ml. The corresponding biomass yield was 0.28 g cell/g glucose.

A mathematic model was developed to investigate the inherent relationships between growth, substrate consumption, differentiation and product formation. The model includes three distinguishable cell states and the transition from the vegetative phase to sporangium and finally to mature spore. An age-based population balance model was applied to describe the maturity of sporangium toward the formation of spores. The model was able to predict the transient behavior of *B. subtilis* in both batch and fed-batch cultures satisfactorily.

Approved: Tingyue Gu
Associate Professor of Chemical Engineering
ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my dissertation advisor, Professor Tingyue Gu for his help and guidance. I would like to thank Stocker Visiting Professor Dr. Murray Moo-Young and Professor Darin Ridgway for their advising. I would also like to thank Drs. Jianfeng Xu, Fang Liu and Dara O’Donnell for sharing their experiences with me and making suggestions, graduate student Liping Wang for her help and suggestions, and technician Jim Caesar for fixing the equipment.

My appreciations go to Drs. W.-J. Russell Chen and Bhavin Mehta for serving as my committee members, and Ms. Edie Chalfant, the secretary of the Department of Chemical Engineering for clerical help.

Finally, I would like to thank my parents for their love, encouragement and support.
TABLE OF CONTENTS

ABSTRACT .......................................................................................................................3
ACKNOWLEDGMENTS .......................................................................................................5
LIST OF TABLES .............................................................................................................9
LIST OF FIGURES .........................................................................................................10
LIST OF SYMBOLS .......................................................................................................12

CHAPTER 1 – INTRODUCTION ..............................................................................15

CHAPTER 2 – LITERATURE REVIEW ....................................................................21

2.1 Strain ..................................................................................................................21
   2.1.1 Biology of *B. subtilis* ...............................................................................21
   2.1.2 Plasmid stability of *B. subtilis* .................................................................24
   2.1.3 Methods for enhancing plasmid stability ....................................................25

2.2 Fermentation control ..........................................................................................26
   2.2.1 Limitations of the built-in fermenter control software .................................26
   2.2.2 LabVIEW software ...................................................................................27
   2.2.3 Serial communication between the bioreactor and computer .....................28
   2.2.4 Fed-batch fermentation ..............................................................................30
      2.2.4.1 Feed forward control .........................................................................32
      2.2.4.2 Closed-loop feedback control .............................................................33

2.3 Metabolic pathway of *B. subtilis* ....................................................................37

2.4 Modeling ............................................................................................................37

CHAPTER 3 – MATERIALS AND METHODS ..................................................41
3.1 Strain and plasmid ................................................................. 41
3.2 Fermenter ............................................................................. 41
3.3 Culture media and conditions .............................................. 43
3.4 Assays .................................................................................. 45
  3.4.1 Cell concentration ............................................................ 45
  3.4.2 Spore number and nonspore number ............................ 45
  3.4.3 Glucose assay ................................................................. 46
  3.4.4 Acetate assay ................................................................. 46
  3.4.5 Alpha-amylase assay ...................................................... 46
  3.4.6 Glucose standard curve .................................................. 47
  3.4.7 Determination of the fraction of plasmid-bearing cells ...... 47

CHAPTER 4 – SETUP OF SERIAL COMMUNICATION BETWEEN
COMPUTER AND BIOREACTOR .............................................. 50
  4.1 Fermenter settings .............................................................. 50
  4.2 Communication port setting ............................................... 50
  4.3 LabVIEW program ......................................................... 52

CHAPTER 5 – A SEGREGATED MODEL FOR Bacillus Subtilis ........ 59
  5.1 Spore germination .............................................................. 59
  5.2 Segregated model .............................................................. 63
    5.2.1 Batch culture with cell lysis ........................................... 69
    5.2.2 Batch and constant feed .............................................. 75
    5.2.3 Parameter sensitivity analysis ..................................... 76
LIST OF TABLES

Table                                          Page

3.1. Medium composition for *B. subtilis* fermentation ........................................44

4.1. Pin assignments of the 9-pin RS-422 interface on the B. Braun fermenter’s control unit ..................................................................................................................53

4.2. The assignments of the telephone wires to the corresponding pins of the 9-pin RS-422 connector on the fermenter’s control unit ......................................................54

5.1. Summary of experimentally determined parameters ...........................................72

5.2. Parameter sensitivity analysis ...........................................................................79

6.1. Media composition of nutrient-limited batch culture .........................................90

6.2. Nutrient yield coefficients from batch culture in shake flasks ..........................91
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1. Enzymatic conversion of starch to high-fructose corn syrup (Zeman et al., 1985)</td>
<td>16</td>
</tr>
<tr>
<td>2.1. Life cycle of <em>B. subtilis</em> (after Fordyce et al., 1996)</td>
<td>22</td>
</tr>
<tr>
<td>2.2. Image of <em>B. subtilis</em> in which developing spores are the bright spots shown in this image (Deacon, 2002)</td>
<td>23</td>
</tr>
<tr>
<td>2.3. Simplified scheme for <em>B. subtilis</em> metabolism (Jeong et al., 1990)</td>
<td>38</td>
</tr>
<tr>
<td>3.1. Picture of 22-liter B. Braun Bioreactor</td>
<td>42</td>
</tr>
<tr>
<td>3.2. Standard curve for determining the glucose concentration from OD value</td>
<td>48</td>
</tr>
<tr>
<td>4.1. RJ-11 telephone jack pin assignments on the Model 263F converter</td>
<td>51</td>
</tr>
<tr>
<td>4.2. Front panel of LabVIEW program file Dualexponential.VI used for dual exponential feed to the fermenter</td>
<td>57</td>
</tr>
<tr>
<td>4.3. Block diagram of LabVIEW program file Dualexponential.VI used for dual exponential feed to the fermenter</td>
<td>58</td>
</tr>
<tr>
<td>5.1. OD at 660 nm vs. time for batch spore germination</td>
<td>61</td>
</tr>
<tr>
<td>5.2. Spore concentration vs. time for batch spore germination</td>
<td>62</td>
</tr>
<tr>
<td>5.3. Batch culture with cell lysis</td>
<td>70</td>
</tr>
<tr>
<td>5.4. Time course of dissolved oxygen in batch culture with cell lysis</td>
<td>71</td>
</tr>
<tr>
<td>5.5. Sporulation rate vs. specific vegetative growth rate</td>
<td>74</td>
</tr>
<tr>
<td>5.6. Simulated results of a batch culture with constant feed rate</td>
<td>77</td>
</tr>
<tr>
<td>6.1a. Effect of initial concentration of ammonium sulfate on cell growth</td>
<td>83</td>
</tr>
<tr>
<td>6.1b. Effect of initial concentration of ammonium sulfate on alpha-amylase production</td>
<td>83</td>
</tr>
</tbody>
</table>
6.2a. Effect of initial concentrations of trp, tyr and phe on cell growth .........................84
6.2b. Effect of initial concentrations of trp, tyr and phe on alpha-amylase production.....84
6.3a. Effect of initial concentration of sodium acetate on cell growth ..........................85
6.3b. Effect of initial concentration of sodium acetate on alpha-amylase production......85
6.4. Exponential fed-batch culture with dual feeding.....................................................93
LIST OF SYMBOLS

e: spore concentration [cells/l]

E: alpha-amylase concentration [u/l]

F: volumetric feed rate of Feed Stream 1 containing glucose [l/hr]

F₁: volumetric feed rate for Feed Stream 1 [l/hr]

F₂: volumetric feed rate for Feed Stream 2 [l/hr]

h: step size

k₁: constant [hr⁻¹]

k₂: constant

k₃: constant [hr⁻¹]

k₄: death rate [hr⁻¹]

k₅: alpha-amylase production constant [u/cells/hr]

kᵢ: inhibition constant [g/l]

kₛ: saturation constant [g/l]

qₒ₂: specific oxygen consumption rate [g·(g·hr)⁻¹]

qₙₒ₂: specific carbon dioxide production rate [g·(g·hr)⁻¹]

Qₒ₂: oxygen consumption rate [g·(g·hr)⁻¹]

Qₙₒ₂: carbon dioxide production rate [g·(g·hr)⁻¹]

S: glucose concentration in the broth [g/l]

S₀: glucose concentration in the feed [g/l]

Sₙₑ: concentration of the nutrient other than glucose (such as ammonium sulfate) in the feed [g/l]
$S_N$: concentration of the nutrient other than glucose (such as ammonium sulfate) in the fermenter [g/l]

t: time [hr]

t_e: time between Stage 0 and Stage 2 of sporulation [hr]

$u(1/2)$: y value at the midpoint

$V_0$: broth volume when feed starts [l]

$V$: broth volume at time t [l]

$X_0$: the cell mass concentration when feed starts [g/l]

$X$: the cell mass concentration in the broth at time t [g/l]

$x_e$: spore number density [cells/l]

$x_s$: sporangium age density function [cells/hr/l]

$x_s(t)$: total sporangium density [cells/l]

$x_v$: vegetative cell number density [cells/l]

$x_{nonspore}$: nonspore number density [cells/l]

$Y_{X/S}$: biomass yield on glucose [g/g]

$Y_{X/S_N}$: biomass yield on nutrient [g/g]

$Y_{x/v}$: vegetative cell biomass yield [cells/g]

Greek Letters

$\mu$: specific growth rate [hr$^{-1}$]

$\mu_g$: specific germination rate [hr$^{-1}$]

$\mu_{max}$: maximum specific growth rate [hr$^{-1}$]
\( \mu_v \): specific growth rate of vegetative cell [hr\(^{-1}\)]

\( \mu_s \): sporulation rate [hr\(^{-1}\)]

\( \nu \): specific substrate uptake rate [hr\(^{-1}\)]

\( \tau \): age of sporangium [hr]

\( \tau_m \): spore mature time [hr]
CHAPTER 1
INTRODUCTION

With the development of modern biotechnology, the food industry has undergone great changes. There are many reports about the genetic engineering enzymes that have been used safely in the food industry. Enzymes are the biocatalysts for catalyzing biochemical reactions. There are many enzymes available in commercial applications. Alpha-amylase, glucoamylase, and glucose isomerase are used together for starch conversion, and they account for more than one third of the total sales. They are mainly used in the production of high-fructose corn syrup (HFCS) (Fig. 1.1) (Zeman et al., 1985).

Large quantities of glucose-containing syrups are manufactured by hydrolysis of cornstarch using enzymes. First, a starch-water mixture is catalyzed by alpha-amylase at temperatures near the boiling point of water. “Alpha-amylases can randomly hydrolyze the $\alpha 1\rightarrow4$ glycosidic linkages of amylose, amylopectin, and related polysaccharides.” (Zeman et al., 1985). In the production of HFCS, in order to reduce the by-product formation in the hydrolyzates and refining costs, a thermo-stable alpha-amylase that can withstand temperatures of more than 100 °C is particularly desirable (Zeman et al., 1985).

Bacillus stearothermophilus can produce the thermo-stable alpha-amylase, while Bacillus subtilis can only produce alpha-amylase that is not thermo-stable thus limiting its use in industry. B. stearothermophilus however, is more difficult to culture than
Starch
   ↓
  Alpha-amylase
   ↓
Liquefied starch
   ↓
  Glucoamylase
   ↓
Dextrose
   ↓
  Glucose isomerase
   ↓
High fructose corn syrup

Fig. 1.1. Enzymatic conversion of starch to high-fructose corn syrup (Zeman et al., 1985)
B. subtilis. In order to increase the productivity of thermo-stable alpha-amylase, B. subtilis was selected as the host for the production of thermo-stable alpha-amylase. The advantages of B. subtilis as a host for the production of heterologous proteins are as follows:

1) It can grow quickly.
2) It is non-pathogenic, and it has been granted as safe by the US Food and Drug Administration.
3) Since the strain has the ability to secrete large quantities of enzyme directly into the fermentation broth, there is no need to break the cells and remove viscous nucleic acids in order to recover the enzyme (Mountain, 1989).

The combination of the respective advantages of B. subtilis and B. stearothermophilus has been achieved through genetic engineering. Mielenz et al. (1985) constructed this kind of recombinant B. subtilis successfully, and this culture is available as ATCC No. 31784.

A fermenter provides a favorable environment for culturing B. subtilis. There are three common operational modes: batch, continuous and fed-batch. There is no addition of nutrient or removal of product in batch culture, and it has only limited use in the pharmaceutical industry (Rani et al., 1999). A steady state will be achieved as the nutrients are continuously added and products are continuously removed during continuous operation. For fed-batch culture, one or more nutrients are added to the
fermenter and there is no removal of product from the fermenter. Fed-batch culture is more desirable than batch culture when the concentration of a nutrient affects the productivity and yield.

High-density cultivation of cells may be the most effective method to obtain high concentration of heterologous products. Fed-batch fermentation is commonly used in high-density recombinant cell fermentation, as this strategy can attain a high cell density by avoiding substrate inhibition and the accumulation of inhibitory metabolite such as acetic acid (Lee et al., 1999). Since both overfeeding and underfeeding of nutrients are harmful to cell growth and product formation, development of a suitable feeding strategy is important in fed-batch cultivation. Various feeding strategies have been developed to control the nutrient concentration within the optimal range, and have been applied to high cell density culture of several microorganisms such as *Escherichia coli* (Lee, 1996), yeast (Yang et al., 1997), *B. subtilis* (Park et al., 1992), and *Alcaligenes eutrophus* (Ryu et al., 1997).

The implementation of a feeding strategy is usually achieved via computer control. A computer control system can realize the monitoring and control of process values. Conventional data acquisition and control software can only cope with a predefined number of tasks. LabVIEW is more flexible and can take on many complicated tasks (Kellerhals et al., 1999). LabVIEW software as standard data acquisition and control software has been successfully used in several biochemical engineering applications, such as mammalian cell cultures (Stoll et al., 1996), baker’s
yeast (Gregory et al., 1994), *E. coli* (Turner et al., 1994) and *Pseudomonas putida* KT2442 fermentations (Kellerhals et al., 1999).

For recombinant *B. subtilis*, there are only a few published papers on high cell density cultures. Park et al. (1992) used a glucose sensor together with an adaptive control strategy to control the glucose concentration at 1 g/l during the fed-batch culture, and a final cell density of 186 g/l was achieved. The recombinant product in their work was an intracellular enzyme. Martinez et al. (1998) used an exponential fed-batch culture, and the results were unsatisfactory due to the characteristics of their *B. subtilis* strain. The lower the specific growth rate, the higher the cell density and the lower the heterologous protein production in their case. A final cell density of about 40 g/l was achieved by controlling the specific growth rate at 0.05 hr⁻¹ during the exponential feeding. Vuolanto et al. (2001) reported a cell mass of 56 g/l for the fed-batch culture of recombinant *B. subtilis* BD170 in a semi-defined medium with glucose and yeast extract feeding. During the fed-batch phase, glucose concentration in the growth medium was maintained constant (1 - 2 g/l) by on-line glucose monitoring and by manually controlling the glucose feed rate. Since glucose sensor is expensive, exponential fed-batch culture is more attractive in high cell density cultures.

The production of extracellular enzymes by *Bacillus* is associated with differentiation (sporulation), and these enzymes are known as the secondary metabolites. Efforts have been made to model the production of secondary metabolites by *Bacillus*. In the work by Ollis et al. (1983), the exotoxin production by *Bacillus thuringiensis* is described by the Luedeking-Piret equation, and spore formation is a function of the first
order of the delayed biomass concentration. Shene *et al.* (1999) proposed a simple model for unstructured microbial growth and metabolite production with glucose repression.

A typical structured single-cell model describing the transition from vegetative growth to sporulation incorporates thirty-nine non-linear differential equations and almost two hundred parameters (Jeong *et al.*, 1990). A segregated model which divides cells into three groups and contains an implicit population balance model for sporangia was first proposed by Dawes and Thornley (1970). They studied the growth and sporulation in continuous culture. Their segregated model was further developed by Fordyce *et al.* (1996), who included a two-compartment substrate consumption and extended the application of their model from steady-state culture to transient growth. However, no secondary metabolite production was included in their model.

The objectives of this work are to:

1) Create an interface between a personal computer and a 22-liter bioreactor’s control system, and write LabVIEW programs to achieve on-line monitoring and control of process values.

2) Achieve a high cell density and a high productivity by exponential fed-batch culture.

3) Establish a model to describe the product formation kinetics. The model can be used for further optimization and control.
CHAPTER 2
LITERATURE REVIEW

2.1 Strain

There are three kinds of strains, wild-type, mutant and recombinant. Recombinant strains contain plasmid inserted through genetic engineering.

2.1.1 Biology of *B. subtilis*

*B. subtilis* is a spore forming bacterium. Three types of cell exist when culturing *B. subtilis*. They are vegetative cells, sporangia, and spores. The life cycle of *B. subtilis* is shown in Fig. 2.1.

Vegetative cells are gram positive and are rod shaped as shown in Fig. 2.2. When the nutrients are sufficient, vegetative cells can grow exponentially. When the nutrients are depleted, some of the cells will sporulate to form the sporangia, while other cells will die.

During the sporangium stage, the cell can be further divided into seven stages based on the differences in morphology and biochemistry. Many secondary metabolites such as alpha-amylase and protease are produced during Stage 0 and Stage 2 of sporangium. During Stage 5, the cell gains the refractivity, and becomes visible using phase-contrast light microscopy. During Stage 6, the spore develops into a mature
Fig. 2.1. Life cycle of *B. subtilis* (after Fordyce *et al.*, 1996).

- **Alpha-amylase**
- **Sporangium**
- **Sporulation cycle**
- **Germination**
- **Vegetative cell**
- **Vegetative Growth**
Fig. 2.2. Image of *B. subtilis* in which developing spores are the bright spots shown in this image (Deacon, 2002).
spore, obtaining the resistance properties against heat, chemicals, desiccation and radiation due to its extra outer layers. During Stage 7 the mother cells undergo lysis and release mature spores. The entire sporulation process takes approximately seven hours (Doi, 1989). However, other researchers reported that the time for B. subtilis to achieve maturity is in the range of eight to ten hrs (Nicholson et al., 1990; Losick et al., 1984).

The B. subtilis spore morphology is ellipsoidal and approximately 0.3 - 0.5 μm in diameter, which is quite different from the rod-shaped vegetative cell morphology. When the spore is exposed to a favorable environment, it will germinate to form a vegetative cell again (Fordyce, 1992).

2.1.2 Plasmid stability of B. subtilis

Recombinant plasmids with high copy numbers can lead to overproduction of recombinant plasmid encoded proteins. This “gene dosage effect” is often reduced because of plasmid instability (Wei et al., 1989). On the other hand, when the random plasmid partitioning mechanism happens at cell division, the probability of forming the plasmid-free cells would be reduced due to high copy number (Summers et al., 1984). The problem of plasmid instability appears to be universal; it is encountered in E. coli, B. subtilis as well as in yeast.

Segregational instability can lead to a drop in copy numbers, even the loss of the entire plasmid from the cells. Recombinant B. subtilis strains containing an antibiotic resistance marker gene in the plasmid will grow in the presence of the antibiotic, while
the plasmid-free cells will be eliminated. The segregational aspect of stability is affected by several factors including host and plasmid genotype and culture conditions such as temperature, pH of the medium, aeration, composition of the growth medium, and dilution rate (O’Kennedy et al., 1999).

It is well documented that recombinant B. subtilis also suffers from structural instability (Gurakan et al., 1998). Structural instability of plasmids may be caused by deletion, insertion or rearrangement of DNA. This will happen if the modified cells (including cells with modified plasmids or plasmid-free cells) have growth advantage over the parental cells. Due to structural instability, the modified cell retains the antibiotic resistance gene but loses the alpha-amylase gene from the plasmid. Such plasmid-bearing cells can grow in the presence of antibiotic but are unable to produce the heterologous protein (Gurakan et al., 1998).

2.1.3 Methods for enhancing plasmid stability

The improvement of plasmid stability can be achieved by several methods. One method is the selection of more stable clones over the less stable clones based on plasmid evolution. The more stable clones can be isolated from long-term selective batch and chemostat cultures. Antibiotic (positive) selection systems may be more effective as compared to auxotrophic (negative) selection because there can be an increase of selective pressure for plasmid-bearing cells with antibiotic selection (O’Kennedy et al., 1999). However, the higher level of selective pressure is not suitable for the selection of
structurally unstable strains. The total elimination of structural instability has been achieved in batch, fed-batch, and continuous culture experiments for *B. subtilis* TN106 [pAT5] by plasmid evolution in nonselective medium (Wei *et al.*, 1989). Other operating strategies were also developed to overcome the plasmid instability problem, including cell immobilization (Zhang *et al.*, 1997), oscillation of culturing conditions (Argyropoulos *et al.*, 1997), feeding strategy (Cheng *et al.*, 1997) and change of culture medium (Brigidi *et al.*, 1997).

### 2.2 Fermentation control

#### 2.2.1 Limitations of the built-in fermenter control software

The built-in fermenter control software in the 22-liter B. Braun Biostat C (B. Braun Biotech International GMBH, Melsungen, Germany) control unit used in this work can display the process values when controlling the fermenter, but its capabilities are limited. The limitations of the software that came with the fermenter are:

a. It cannot display the process parameters in a real-time graphical window.

b. It cannot store the process parameters as a computer file for later use.

c. It cannot adjust the process parameters automatically. For example, its software cannot update the set point of the feed rate. The set point must be changed manually.
d. It cannot arrange the scheduling of events with flexibility, such as the time and duration of feed in a fed-batch culture.

Thus, there is a need for more powerful software for automatic monitoring and control of the key process parameters.

2.2.2 LabVIEW software

Fermentation is a very complicated process, and feedback control of fermentation is a useful strategy to guarantee a successful run. The on-line analysis instruments and sensors reflect valuable on-line information of the microorganisms, and a control program written by engineers responds to the feedback information. LabVIEW software is a standard software program for data acquisition and control. “It supports IEEE488 (GPIB), RS232/422, and VXI, as well as plug-in A/D, D/A, and digital I/O boards” (Jamal, 1994). This work utilized LabVIEW to achieve flexible control and monitoring of the 22-liter B. Braun Biostat C fermenter.

LabVIEW (LabVIEW for WINDOWS, Version 4.1) ran on a personal computer (Pentium, 90MHz, 32 MB RAM). LabVIEW programs are called virtual instruments (VI) because their appearance and operation simulate actual instruments. A single VI has three main parts:

1) The front panel imitates the panel of an instrument. The front panel provides a user an interface for data inputs and outputs. The user can input data using a
mouse and keyboard, and then view the results produced by the user’s program on
the screen.

2) The block diagram is the actual executable program that contains the source code.
   It is written in LabVIEW’s graphical programming language G. The block
diagram is composed of icons that represent lower-level VIs, built-in functions
and program control structures. Wires are used to connect the icons together,
indicating the flow of data in the block diagram.

3) An icon represents a VI used in another VI. It is like a subroutine, and is called a
   subVI. The icon can be used as top-level program or subprogram within other
   programs. The connector defines the inputs and outputs of the VI. The icons and
   connectors realize the data flow between the VIs.

   With the features above, the users can divide a large task into several small tasks.
   Each small task is represented by a subVI, and each subVI can be debugged separately.
   Finally, a whole program can be formed by the combination of subVIs (Wells, 1994).

2.2.3 Serial communication between the bioreactor and computer

   The hardware platform (DFC-2) of the B. Braun fermenter’s control unit is named
the substation. Data are transmitted between the computer and the fermenter’s control
unit according to the master-slave principle. The computer can control the data
transmission to a substation via commands in the form of a telegram under a serial
communication protocol. A telegram mainly consists of command, value and checksum.

Commands are available as:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR</td>
<td>“data request”</td>
</tr>
<tr>
<td>DS</td>
<td>“data set”</td>
</tr>
<tr>
<td>SU</td>
<td>“setup”</td>
</tr>
</tbody>
</table>

Before sending data to the fermenter or requesting data from fermenter, the host computer must initialize the corresponding substation (in this case the fermenter’s control unit), the initialization command sends parameters to the substation and thus adapts the data transmission to the requirements. Without initialization, the substation sends an error telegram after receiving a data request (DR) or a data set (DS).

The telegram for the substation initialization is as follows:

*1:SU:0:0:PV:5:2:1030@

The interpretation of this telegram is illustrated in Appendix A. Calc Chk Sum.VI can calculate the checksum of the telegram and is illustrated in Appendix B. Test 1.VI for realizing the data transmission between the computer and the bioreactor is also explained and shown in Appendix B. This program was the core subroutine of all the monitoring and/or control programs.
2.2.4 Fed-batch fermentation

It is reported that at a higher growth rate, a large quantity of propionic acid is formed which is the most harmful by-product for the *B. subtilis* cell. At a lower growth rate, more acetate is accumulated in the medium that is harmful for *B. subtilis* (Park *et al.*, 1992; Snay *et al.*, 1989). Carbon dioxide can also affect cell growth in high cell density culture. High partial pressure of carbon dioxide decreases growth rate and stimulates acetate formation. Therefore, the increase of the pressure in a fermenter enhances both oxygen transfer and the detrimental effect of carbon dioxide (Lee, 1996).

Formation of acetate in *E. coli* cultures not only occurs under fully aerobic conditions with an excess carbon source, but it also forms under anaerobic conditions. These two mechanisms are often referred to as the overflow metabolism and the mixed-acid fermentation, respectively. When *E. coli* is grown under fully aerobic conditions, the formation of acetate typically takes place at high growth rates and/or high glucose uptake rates (Akesson *et al.*, 1999).

High cell density culture is an attractive means of achieving high product concentration. However, researchers cannot obtain high cell density (greater than 10 g/l) with batch culture alone, because the cells suffer from substrate inhibition and catabolite inhibition. It is well documented that catabolic acid by-products will accumulate in the fermentation broth during the batch culture, and they will inhibit cell growth and recombinant protein production (Yee *et al.*, 1993). In order to attain high cell densities by
the use of glucose as carbon source, fed-batch culture is preferred to minimize acetate production and to eliminate substrate inhibition (Yee et al., 1993).

In fed-batch cultures, in order to restrict the formation of the acetate, the feed rate for glucose should be controlled. In the early stage of cultivation, the cell density is rather low, so the glucose feed rate is low due to the low glucose consumption rate. In this period, no acetate is formed. As the cells grow, the feed rate and oxygen consumption can be increased and finally the oxygen consumption exceeds the maximum oxygen transfer capacity even when the stirrer speed is at its maximum, and an anaerobic condition will occur, so the feed rate should be reduced (Akesson et al., 2001). Many other researchers used air and/or pure oxygen to maintain dissolved oxygen concentration greater than 20% of air saturation to avoid anaerobic condition during the later part of the high cell density culture (Yee et al., 1993; Vuolanto et al., 2001; DeLisa et al., 1999).

When the anaerobic condition is avoided, acetate present in the media can also be reassimilated by the cells (Akesson et al., 2001; vande Walle et al., 1998). With the decrease of growth rate at the later stage of aerobic culture, the pH in the broth starts to increase due to the accumulation of ammonia, depletion of amino acid and the consumption of acetate. Robbins et al. (1989) hypothesized that ammonia was produced by the extensive deamination of amino acids (Robbins et al, 1989; Jensen et al., 1990).

High cell density cultures producing a recombinant protein or metabolite have only been reported since 1975. Many feeding strategies have been developed to avoid substrate inhibition and the accumulation of acetate. There are two principal strategies for
the control of the nutrient feed: the open-loop (feed forward) control and the closed-loop (feedback) control (Luli et al., 1988).

2.2.4.1 Feed forward control

(a) Pulse feed

The nutrients can be added intermittently at predetermined times. Sometimes, in order to overcome some of the effects of a sudden increase of osmolarity, a semi-continuous mode of nutrient addition is adopted for the subsequent fed-batch culture. Semi-continuous mode means the time of addition and the duration are predetermined (Elias et al., 2000). Pulse feed sometimes is combined with other feeding strategy such as exponential feed to solve the solubility problem (Yee et al., 1993).

(b) Constant feed

The nutrients can also be added at a constant feed rate. When the substrate is limited, the cells can grow linearly. The specific growth rate continuously decreases. When the substrate is in excess, the cells can grow exponentially. However, no quasi-steady state is achieved (Yamane et al., 1984).

(c) Linear feed

For the linear feeding, although the cell can grow to a high cell density, the substrate concentration cannot be controlled at the set point (Luli et al., 1988).
(d) Exponential feed

Exponential fed-batch culture has been used widely in the high cell density culture of yeast and bacteria. Wilms et al. (2001) reported that 100 g/l cell dry weight was achieved by using exponential feeding strategy, and 3.8 g/l recombinant L-N-carbamoylase in *E. coli* was produced using a positively regulated promoter. van Hoek et al. (2000) reported a biomass concentration of 130 g/l for baker’s yeast by exponential feeding.

(e) Feed based on model

In order to maximize product concentration at a fixed time, the optimal feed rates for glucose and ammonium hydroxide are determined by the non-singular control algorithm using a proposed model. By applying these optimal feed rates for glucose and ammonium hydroxide in the fed-batch culture, it was possible to achieve a high final cell concentration of 141 g/l and a high polyhydroxybutyrate concentration of 105 g/l after 40 hr in *E. coli* fermentation (Lee et al., 1997).

### 2.2.4.2 Closed-loop feedback control

(a) Constant dissolved oxygen concentration (DO-stat)

DO will rise above the set point when the nutrients are depleted, and DO will drop below the set point when nutrients are in excess. The glucose feed rate is used to
maintain a constant dissolved oxygen level, namely when the DO becomes higher than its set point, nutrients are fed to the fermenter at a predetermined rate. Akesson et al. (1999) pointed out that the specific growth rate could be kept more or less constant by combining the DO-stat (closed-loop control, with DO as the control variable and the carbon feeding rate as the manipulated variable) with an open-loop control for the stirrer speed, which was increased according to a predetermined exponential profile. If the stirrer speed was kept constant, dissolved oxygen decreased due to the increased oxygen demand when the feed rate increased exponentially based on the constant specific growth rate.

(b) Constant pH (pH-stat)

pH-stat control is only suitable for a medium that will be sensitive to pH changes due to the accumulation of acid by-products or the consumption of the medium components. Kim et al. (1992) reported a final cell mass of 117 g/l for recombinant E. coli with 400 g/l glucose, 100 g/l yeast extract and 100 g/l tryptone in the feed by pH-stat fed-batch. In their work, the pH of the medium rose above 7.2 due to the secretion of ammonium into the fermentation broth. The feeding solution was supplied at a predetermined time interval (0.5 to 4 min). After feeding, the pH dropped due to the production of organic acids and was neutralized with 2 N NaOH to pH 7.1 (Kim et al., 1998). Wang et al. (1998) reported a final cell density of 156 g/l for metabolically engineered E. coli in a defined medium by the pH-stat fed-batch culture. When glucose was depleted and pH rose rapidly to 6.91 due to the consumption of acetic acid in the
fermentation broth, a certain amount of feeding solution was added as a pulse. When pH dropped below 6.8, the broth pH was controlled at 6.8 by the addition of 28% ammonia water.

(c) Carbon dioxide evolution rate (CER)

The carbon dioxide in the off-gas can be continuously measured with an acoustic gas monitor. The carbon dioxide evolution rate is roughly proportional to the rate of consumption of the carbon source. The control strategy based on the on-line measurement of carbon dioxide evolution rate was used to control the substrate feed rate. (Taherzadeh et al., 2000).

(d) Substrate concentration control

Autoclavable glucose biosensors based on the immobilization of glucose oxidase have been used during fed-batch fermentation of *E. coli*. A maximum cell density of 85 g/l was reached (Phelps et al., 1995).

(e) Cell concentration

The nutrient feeding rate can be determined from the cell concentration, which is measured on-line using a laser turbidimeter. The information about the cell populations, cell size, cell volume, biomass and biomass concentration, can also be obtained by *in situ* microscopy (Bittner et al., 1998). There is also a report of on-line estimation of biomass and acetate from base consumption and conductivity measurements in high cell density
culture of *E. coli* (Hoffmann *et al.*, 2000). Vicente *et al.* (1998) reported on-line estimation of biomass through pH control analysis in aerobic yeast fermentation systems. In the future, the on-line coupling of modern monitoring techniques such as intrinsic fluorescence spectroscopy, fluorescence-activated cell sorting and flow cytometry, as well as *in situ* microscopy of high cell density culture, will provide more detailed information about dense populations (Riesenberg *et al.*, 1999).

For both the pH-stat and DO-stat methods, there is a need to determine the nutritional requirement of the culture, and the feed rate of the substrate. In a defined medium, DO-stat responds more rapidly to nutrient depletion than pH-stat. While in complex medium, pH-stat responds more rapidly to nutrient depletion than DO-stat. The DO change is not significant after the depletion of the carbon source since the cells will utilize the complex substrates (yeast extract or peptone) (Lee, 1996). pH-stat and DO-stat feeding methods are not available for the control of substrate concentration at low levels. In recent years, an on-line enzymatic glucose analyzer has often been used for control of glucose concentration. This analyzer enables control of glucose concentration at very low levels that can go below 0.08 g/l (Mizutani *et al.*, 1987).

The method of nutrient feeding is critical to the success of high cell density culture as it affects not only the maximum cell concentration but also the product formation. Two feeding methods were compared by Kim *et al.* (1998). They are pH-stat and exponential feeding. The pH-stat method was more effective in increasing endoglucanase productivity. For the exponential cultivation, the same amounts of glucose used in the pH-stat were supplied exponentially to the fermenter. This feeding strategy
should maintain an essentially constant growth rate and a glucose concentration of zero. However, glucose began to accumulate after 10 hrs of feeding, indicating reduced glucose consumption and a decrease in growth rate (Kim et al., 1998).

2.3 Metabolic pathway of B. subtilis

The metabolic pathway of B. subtilis is shown in Fig. 2.3. Glucose has three outlets: (1) glycolysis and the TCA cycle for providing energy, (2) the pentose phosphate pathway for main biosynthetic utilization, and (3) as a precursor for cell envelop synthesis. Amino acids are synthesized within the cell and are utilized for product formation and nucleotide synthesis. The depletion of purine nucleotide will trigger the initiation of sporulation process (Jeong et al., 1990).

In the metabolic pathway of B. subtilis, acetate is a by-product of glucose metabolism. Its level will affect the final biomass yield on glucose. The accumulation of acetate may be caused by the overloading of the TCA cycle. In order to reduce acetate formation, glucose has to be supplied slowly, so that TCA cycle can handle all the produced acetyl CoA (vande Walle et al., 1998).

2.4 Modeling

Population balance model describes the distinguishability between individuals (Fordyce, 1992). The definition of population balance model is as follows:
Fig. 2.3. Simplified scheme for *B. subtilis* metabolism (Jeong et al., 1990).
\[
\frac{\partial f}{\partial t} + \nabla \cdot (f \frac{dz}{dt}) = B - D
\]

\(f(z,t)\) is the density function, and \(t\) is the time. \(z\) defines a vector of distinguishing characteristics, such as size, age and chemical composition. \(B\) is the birth term, and \(D\) is the death term. For the age population model, age is the distinguishing characteristic. Let \(\tau\) represent age. Cell age is directly correlated with time since birth, so the rate that cells age, \(d\tau/dt\), is a constant of one. The choice of age as the descriptive vector in the population balance model will simplify the modeling as the birth term is included in the boundary condition (Fordyce, 1992). The age population balance model has been applied by Fordyce et al. (1996) for \(B.\ licheniformis\), and it is also used in describing the cell cycle for hybridoma cells (Martens et al., 1995).

In a typical batch process, cell growth shows the following phases: lag phase, logarithmic or exponential growth phase, deceleration phase, stationary phase and death phase. The famous Monod growth law is as follows:

\[
\frac{dx_v}{dt} = \mu_v x_v
\]

\[
\mu_v = \frac{\mu_{\text{max}} S}{k_s + S}
\]

The parameter \(\mu_{\text{max}}\) represents the maximum growth rate. \(S\) represents the substrate concentration. \(k_s\) represents the substrate concentration that will support a growth rate of half the maximum. Monod growth kinetics implicitly assumes that vegetative cells are the same.
Parameter estimation is an optimization problem. There are some advantages and disadvantages of the simplex method as a direct method for optimization as compared to an indirect method such as Newton’s method. One of the disadvantages of the simplex method is that it converges slower than the indirect methods.

The advantages of the simplex method are:

1) No need to calculate the gradient (gradient matrix for multi-dimension)
2) Usually no strict requirement for the initial value
3) Usually the obtained optimal value does not depend on the initial value very much, so it is more “global.”

A simplex is a geometric figure. It has vertices equal to one more than the number of dimensions. A simplex in two dimensions is a triangle. A new simplex is constructed by fixing two vertices and creating a new vertex, thus the new values for the variable and the new system response are obtained. Another construction of simplex will be carried out by letting the simplex move toward the best region of the factor space (Walters et al., 1999).
CHAPTER 3
MATERIALS AND METHODS

3.1 Strain and plasmid

*B. subtilis* (ATCC 31784) carrying plasmid pC194 was used throughout this work. Plasmid pC194 contains a thermo-stable alpha-amylase gene from *B. stearothermophilus*. The plasmid also codes resistance to the antibiotic chloramphenicol (Cm). The host strain is alpha-amylase negative *B. subtilis* 1A289, which does not produce homologous alpha-amylase, but produces and secretes heterologous thermo-stable alpha-amylase into the extracellular medium. The genotype for the host strain is phe-tyr-trp-sacA, which means that the host strain is auxotrophic for phenylalanine, tyrosine and tryptophan, and that the host strain cannot use sucrose as the sole carbon source (Mielenz *et al.*, 1985).

3.2 Fermenter

Fermentation experiments were performed in a 22-liter B. Braun bioreactor (Biostat C) with a working volume of 15 liters as shown in Fig. 3.1. pH, stirrer speed, temperature and airflow were controlled by a DFC-2 hardware process control system equipped with on-off and PID controllers. The airflow was distributed at the bottom of the bioreactor using a ring sparger. The pO2 was measured by steam sterilizable
Fig. 3.1. Picture of 22-liter B. Braun bioreactor.
electrode. Agitation and airflow rate were controlled by PID boards, according to the set point values addressed by the pO2 controller. Data and operation parameters were entered via keypad on the front panel of the control unit.

3.3 Culture media and conditions

The recombinant strain was maintained as spores at 4 °C on plates of LB selective agar having the following composition: NaCl 10 g/l, yeast extract 5 g/l, tryptone 10 g/l, chloramphenicol 10 mg/l (10 g/l chloramphenicol was dissolved in 100% ethanol and was stored in a –20 °C freezer), agar 15 g/l. Inoculum was prepared in the LB selective medium containing 10 mg/l chloramphenicol in shake flasks at 37 °C and 200 rpm overnight. The inoculum size was about 3% of the bioreactor working volume for the exponential fed-batch culture, 1% for the batch and constant feed culture. The batch culture was carried out at 37 °C in the B. Braun bioreactor at a stirrer speed of 400 rpm and aeration rate of 1.5 vvm. The exponential fed-batch culture was carried out at 37 °C in the B. Braun bioreactor at a stirrer speed of 400 - 1100 rpm and aeration rate of 1.0 - 2.0 vvm to maintain the dissolved oxygen concentration greater than 20% of air saturation. A 5% (v/v) antifoam 289 was used for elimination of foaming. A defined selective medium containing 10 mg/l chloramphenicol was used for the batch phase and constant feed phase cultures. Minimal defined medium composition is given in Table 3.1. For the exponential feed phase, Feed Stream 1 contained 500 g/l glucose, 150 g/l
Table 3.1. Medium composition for the *B. subtilis* fermentation

<table>
<thead>
<tr>
<th>Nutrient</th>
<th>g/l</th>
</tr>
</thead>
<tbody>
<tr>
<td>I: ((\text{NH}_4)_2\text{SO}_4)</td>
<td>2.5</td>
</tr>
<tr>
<td>(\text{K}_2\text{HPO}_4)</td>
<td>3</td>
</tr>
<tr>
<td>(\text{KH}_2\text{PO}_4)</td>
<td>1.5</td>
</tr>
<tr>
<td>Sodium Citrate</td>
<td>1</td>
</tr>
<tr>
<td>(\text{MgSO}_4)</td>
<td>0.25</td>
</tr>
<tr>
<td>II: (\text{CaCl}_2)</td>
<td>0.1</td>
</tr>
<tr>
<td>III: (\text{MnSO}_4)</td>
<td>0.01</td>
</tr>
<tr>
<td>IV: (\text{FeSO}_4)</td>
<td>0.01</td>
</tr>
<tr>
<td>V: (\text{ZnSO}_4)</td>
<td>0.002</td>
</tr>
<tr>
<td>VI: Glucose</td>
<td>2</td>
</tr>
<tr>
<td>VII: Tyr</td>
<td>0.05</td>
</tr>
<tr>
<td>Trp</td>
<td>0.05</td>
</tr>
<tr>
<td>Phe</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Note: I through VII were autoclaved separately to prevent precipitation. They were then mixed aseptically.
(NH₄)₂SO₄, 4 g/l MgSO₄, 8.5 g/l phenylalanine, 10 mg/l chloramphenicol. Feed Stream 2 contained 17 g/l tyrosine and 17 g/l tryptophan dissolved in 14.4% (v/v) ammonium water. Feed Streams 1 and 2 were fed to the fermenter at the same time with the specific growth rate set to 0.15 hr⁻¹ after the depletion of glucose. \( Y_{x/s} \) was taken as a constant value of 0.4 gDCWgGLUC⁻¹ as determined by Martinez et al. (1998). The automatic control of dual feeding was implemented using LabVIEW on a personal computer.

3.4 Assays

3.4.1 Cell concentration

Free cell concentration was measured by the optical density (OD) method. OD was measured using a Turner Model 340 spectrophotometer (Fisher, Pittsburgh, PA) at 660 nm and was converted to dry cell weight per liter of broth (1 OD₆₆₀ unit=0.45 g/l based on literature results (Park et al., 1997; Tulin et al., 1992).

3.4.2 Spore number and nonspore number

The sample was kept in an 80 °C water bath for 10 min to kill the nonspore cells, and then both the heated and unheated samples were serially diluted using sterile saline water. The diluted samples were put on the plates, and then LB agar was poured onto the plates. The plates were incubated at 37 °C overnight.
3.4.3 Glucose assay

Glucose concentration was determined using a glucose assay kit from Sigma (kit No. 510). This kit works by oxidation of glucose to a dye that has absorbance at 505 nm. The intensity of the color of the dye that is proportional to glucose concentration is measured by a spectrophotometer.

3.4.4 Acetate assay

Acetate concentration was determined using an acetate assay kit (Catalog No. E0148261) from R-Biopharm Inc. (Marshall, Michigan). This kit works by three sequential enzyme reactions to form NADH that has absorbance at 340 nm.

3.4.5 Alpha-amylase assay

0.5 ml alpha-amylase solution and 0.5 ml starch (1%) in 0.1 M sodium acetate (pH 6.0) were mixed and allowed to react at 40 °C for 30 min. 2 ml DNS reagent were added to stop the reaction and the samples were incubated at 100 °C for 10 min. (DNS reacts with the reducing sugars released by the alpha-amylase). The control was a test tube with inactivated enzyme (boiled) or with alpha-amylase added after incubation and the DNS reagent. Reducing sugar was determined by comparing absorbance at 540 nm in
the assay solution to a standard curve of glucose solutions (1 - 10 mg/ml). One unit of alpha-amylase activity was defined as the release of 1 µmol reducing sugar from the soluble starch per minute.

The preparation of DNS reagent was as follows: Over gentle heat, dissolve 10 g 3,5-dinitrosalicylic acid in 200 ml NaOH (2 N) while gradually adding 500 ml water, then add 300 g potassium sodium tartrate. Bring the total volume up to 1 liter with water and store at room temperature in a dark colored bottle.

3.4.6 Glucose standard curve

Glucose standard curve was shown in Fig. 3.2. The relationship between OD value and standard glucose concentration was that glucose concentration equaled to 0.7 times the OD value.

3.4.7 Determination of the fraction of plasmid-bearing cells

A sample was diluted in saline to obtain a cell count between 30 - 300. LB starch plates consisting of LB agar supplemented with 1% soluble starch were used for incubation at 37 °C. When 20 mg/l Cm was added, some components of the agar chelated Cm. Furthermore, the cells were exposed only to a portion of total Cm since they grew on the agar surface. For these reasons, the Cm concentration in the agar media was higher than that in the media for submerged fermentation. 20 mg/ml Cm was found
Fig. 3.2. Standard curve for determining the glucose concentration from OD value.
experimentally to be the minimal concentration that could inhibit Cm sensitive cells (Kadam et al., 1987). Simultaneous check of the cell counts on starch agar plates and starch-Cm agar plates were used to determine the fraction. Plates were allowed to grow overnight. After counting the entire colony, these plates were exposed to iodine vapor, and were then checked for zones of hydrolysis. The starch hydrolysis zones were proved to have alpha-amylase producing ability.

Under nonselective conditions, the fraction of cells containing the plasmid is defined as:

$$f = \frac{\text{Number of cells containing plasmid}}{\text{Total number of cells}}$$

$$= \frac{\text{Cell count on starch plates containing Cm}}{\text{Cell count on starch plates without Cm}}$$

(4)

The fraction of cells with alpha-amylase producing ability is defined as:

$$f_{\text{amy}} = \frac{\text{Number of cells with alpha – amylase producing ability}}{\text{Total number of cells}}$$

$$= \frac{\text{Number of hydrolysis zones on starch plates containing Cm}}{\text{Cell count on starch plates containing Cm}} \times f$$

(5)
4.1 Fermenter settings

The 22-liter B. Braun fermenter’s control unit has to be set up for a computer. To do this, go to the MAINTENANCE menu on the fermenter control unit’s display and step down to the HOST setup, then set the following parameters: ADR=1, SPEED=9600bd, DATA=7bit, STOP=1bit, PARTY=EVEN.

4.2 Communication port setting

The connection between the fermenter and the personal computer was through a non-powered RS-232 to RS-422 signal converter (Model 263F, Telebyte, Inc., Greenlawn, NY) plugged in the computer’s serial port 2 (RS-232 port). No additional power supply was needed for the converter due to the short distance (roughly 20 ft) between the fermenter and computer. Serial port 1 was used for the computer mouse. The RS-422 interface is not as sensitive to interfering signals as the common RS-232 interface. The Model 263F non-powered RS-232 to RS-422 signal converter was equipped with a 9-pin connector on one end and a RJ-11 phone jack on the other end. The four wires of the phone jack were used with pin assignments shown in Fig. 4.1. A four-
Fig. 4.1. RJ-11 telephone jack pin assignments on the Model 263F converter.
wire telephone cable was used to connect between the converter and RS-422 connector on the fermenter’s control unit. The 9 pin RS-422 female connector’s pin assignments are shown in Tables 4.1.

The Model 263F non-powered RS-232 to RS-422 signal converter was equipped with a DTE\DCE switch selectable for reversing TD and RD. It was set to DCE in this work. The assignments of the telephone wires to the corresponding pins of the RS-422 connector on the fermenter’s control unit are shown in Table 4.2.

4.3 LabVIEW program

Test 1.VI performs bidirectional serial communication using a serial port on the computer. It first initializes the COM 2 port to the specified settings (such as port number, baud rate, data bit, stop bits, and parity), then writes a string to the port. It finally performs a read string with timeout. Figs. B3 and B4 in Appendix B are the front panel and block diagram of Test 1.VI, respectively.

Process values.VI provides a more straightforward user interface than Test 1.VI in data request. Fig. B5 in Appendix B is the front panel of Process values.VI that displays current values of the process variables. Fig. B6 in Appendix B is the block diagram of Process values.VI that converts the respective process value in data transmission range to the true value. In Fig. B6, Test 1.VI and Calc Chk Sum.VI are the subroutines of Process values.VI.

DO.VI realizes the display of DO value in the graph and saves the DO value to
Table 4.1. Pin assignments of the 9-pin RS-422 interface on the B. Braun fermenter’s control unit

<table>
<thead>
<tr>
<th>PIN</th>
<th>Signal</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>GND</td>
<td>GND</td>
</tr>
<tr>
<td>3</td>
<td>DO B (TxD−)</td>
<td>Transmit−</td>
</tr>
<tr>
<td>4</td>
<td>RI B (RxD−)</td>
<td>Receive−</td>
</tr>
<tr>
<td>5</td>
<td>RGND</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>GND</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>DO A (TxD+)</td>
<td>Transmit+</td>
</tr>
<tr>
<td>9</td>
<td>RI A (RxD+)</td>
<td>Receive+</td>
</tr>
</tbody>
</table>
Table 4.2. The assignments of the telephone wires to the corresponding pins of the 9-pin RS-422 connector on the fermenter’s control unit

<table>
<thead>
<tr>
<th>Pin and function</th>
<th>Telephone wire color and function</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 TxD−</td>
<td>Yellow R+</td>
</tr>
<tr>
<td>4 RxD−</td>
<td>Red T+</td>
</tr>
<tr>
<td>8 TxD+</td>
<td>Black R−</td>
</tr>
<tr>
<td>9 RxD+</td>
<td>Green T−</td>
</tr>
</tbody>
</table>
text or spreadsheet files. Figs. B7 and B8 in Appendix B are the front panel and block diagram of DO.VI, respectively. In Fig. B7, user can set the time interval (seconds) for each data request, and the indicator shows the DO value for each data request. In the graphical window, x-axis represents the time elapsed (seconds), y-axis represents the DO (%) value, x-axis and y-axis are default to autoscale, so the graph can autoscale continuously with the change of data. In Fig. B8, Process values.VI is a subroutine of DO.VI. The curve in the graph can be lost forever when the user closes the program, so additional program is needed to redraw the graph.

Read file.VI reads the data from the file and plots the data on a graph. Figs. B9 and B10 in Appendix B are the front panel and block diagram of Read file.VI, respectively.

During a fed-batch culture, nutrients are fed to the bioreactor by a pump. The set point of flow rate can be entered manually into the corresponding pump menu via the keypad at any time. Exponential.VI realizes the automated control of feeding flow rate by the computer. It calculates the feed rate with time and updates the set point for the flow rate. Figs. B11 and B12 in Appendix B are the front panel and block diagram of Exponential.VI, respectively.

Exponentialdo.VI realizes the monitoring of DO change and updating the feed rate at the same time. Figs. B13 and B14 in Appendix B are the front panel and block diagram of Exponentialdo.VI, respectively. In Fig. B14, DO.VI and Exponential.VI are two subroutines of Exponentialdo.VI.
Due to the solubility problem of some medium in the feed stream, these media were separated from Feed Stream 1 that contained concentrated glucose and other nutrients, and were dissolved in solvent to form Feed Stream 2. The two feed streams were pumped into the bioreactor at the same time with different exponential feed rate. Dualexponential.VI realizes the dual exponential feed to the bioreactor. Figs. 4.2 and 4.3 are the front panel and block diagram of Dualexponential.VI, respectively. Exponential.VI and Exponential1.VI are the subroutines of Dualexponential.VI. Exponential1.VI is basically the same as Exponential.VI except that the command telegram *1:DS:0:0:CS:5: is changed to *1:DS:0:0:CS:6:, and the command telegram :CM:5:1: is changed to :CM:6:1:. In telegram :CM:5:1:, 5 represents the pump labeled “substrate 1,” and 6 represents the pump labeled “substrate 2.”
Fig. 4.2. Front panel of LabVIEW program file Dualexponential.VI used for dual exponential feed to the fermenter.
Fig. 4.3. Block diagram of LabVIEW program file Dualexponential.VI used for dual exponential feed to the fermenter.
In this work, spore germination kinetics was evaluated first, then a segregated model was proposed to investigate the inherent relationship between growth, substrate consumption, cell differentiation and product formation in *B. subtilis* fermentation in a 22-liter B. Braun bioreactor. The segregated model included three distinguishable cell states and the transition from vegetative phase to sporangium and finally to mature spore. An age-based population balance model was applied to describe the maturity of sporangia toward the formation of spores. Parameters in the model were determined by fitting the model with experimental data. The model was able to predict the transient behavior of *B. subtilis* in both batch and fed-batch cultures.

### 5.1 Spore germination

The dormant spore will undergo germination when it is exposed to a low concentration of germinant chemicals such as L-alanine. Germination takes place in a series of sequential steps that may be divided in three separable events: (1) activation, which prepares conditions for germination; (2) germination, which loses all the properties of spore and (3) outgrowth, which transforms a germinated spore into a vegetative cell (Lara-Reyna *et al.*, 2000).
Since spore germination causes the loss of spores from the culture and enhances the vegetative cell population, an experiment was done to study the spore germination kinetics. Results are shown in Fig. 5.1. In Fig. 5.1, the OD value at time zero was specified as 100%. From Figs. 5.1 and 5.2, after a period of 5 hrs of delay, a certain portion of spores began to lose their heat resistance property which indicated the start of germination. Then, after another 2 hrs of delay, the germinated spore underwent outgrowth with the formation of a vegetative cell. This result is consistent with the findings of other groups. Fordyce et al. (1992) found the germination was subject to a time delay of 15 hrs or more. However, Dawes et al. (1970) did not observe a delay prior to germination for *B. subtilis*. This was due to the different criteria for judging the spore. In their work (Dawes et al., 1970), germination could be represented by first order kinetics. The rate equation for the change in total spore concentration $e$ at time $t$ due to germination was

$$\frac{de}{dt} = -\mu_s e$$ (6)

The germination rate is below 0.1 hr$^{-1}$ (Dawes et al., 1970) which is rather low. Due to the low spore concentration as compared to the vegetative cell concentration, and the time delay for germination and outgrowth, the germination kinetics contributed little to the overall fermentation. Germination kinetics was neglected in the model developed in this work.
Fig. 5.1. OD at 660 nm vs. time for batch spore germination.
Fig. 5.2. Spore concentration vs. time for batch spore germination.
5.2 Segregated model

Due to the application of selective pressure in the fermentation and the elimination of structural instability from the inoculum, plasmid instability is neglected in the model. The vegetative cell growth model uses Monod growth kinetics. Assuming that only the vegetative cell will consume glucose, the substrate consumption kinetics can be applied. The specific rate of sporulation is defined as the rate of the loss of vegetative cell mass due to sporulation. Schaeffer et al. (1965) found that the sporulation rate is related to the composition of carbon and nitrogen sources, and higher glucose concentration will lead to a lower sporulation rate with the same maximum specific growth rate. Dawes et al. (1970) postulated that the sporulation rate is inversely linear to the specific vegetative growth rate within the dilution rate from 0.05 to 0.38 hr⁻¹, which corresponds to specific vegetative growth rates from 0.1 to 0.42 hr⁻¹. They also implied the dependence of sporulation on the substrate concentration. Fordyce et al. (1992) found that this linear relationship between sporulation rate and specific growth rate did not hold for a wider operation condition. They proposed that the sporulation rate was the linear combination of the piecewise linear basis functions. They described the dependence of sporulation on the substrate concentration in more detail and fitted the experimental data more accurately for the transient growth. In the work by Fordyce et al. (1996), the sporulation rate was inversely linear to internal substrate concentration when the specific vegetative growth rate was above 0.15 hr⁻¹. When the specific vegetative growth rate was below 0.15 hr⁻¹, the sporulation rate was linearly related to the internal substrate concentration.
In this work, the sporulation rate is still inversely linear to the specific vegetative growth rate when the specific vegetative growth rate is above 0.10 hr\(^{-1}\). If the specific growth rate is too large to result in the negative value for the sporulation rate, the sporulation rate is set to 0. The sporulation rate is linearly related to the specific vegetative growth rate when the specific vegetative growth rate is below 0.10 hr\(^{-1}\) in this work. During the sporangium stage, the individual cells from the seven stages of development differ from each other in morphology and biochemistry. Population balance model can describe this heterogeneity. The sporangium age, namely the time since commitment to sporulation, is used to characterize the sporangium population based on stages of spore development. Defining \(\tau\) as the sporangium age, and \(X_s(\tau,t)\) as the sporangium age density function, according to the definition of a population balance model, and assuming there is no death for the sporangia, a partial differential equation describing the age distribution of the cell population can be written with a birth term included in the boundary condition.

\[
\frac{\partial(x_s V)}{\partial t} + \frac{\partial x_s}{\partial \tau} V = 0 \tag{7}
\]

\[
x_s(0,t) = \mu_s(t)x_s(t) \tag{8}
\]

By employing the method of characteristics (Haberman et al., 1998), this partial differential equation can be solved analytically and can be reduced to an ordinary delay differential equation along characteristic trajectories given by \(dt/d\tau=1\).

\[
x_s(t,\tau)V(t) = x_s(t-\tau,0)V(t-\tau) = \mu_s(t-\tau)x_s(t-\tau)V(t-\tau) \tag{9}
\]
The total number of sporangium cells is the sum of all the cells from age zero to mature time. The total viable cell number (including vegetative, sporangium and spore) and spore number can be measured. Thus, only the nonspore number can be obtained by experiment, and there is no convenient experimental method to distinguish between the vegetative and sporangium cells except using electron microscopy (Pierce et al., 1992). The mature spore formation rate is equal to the sporangium density function at mature time.

For the alpha-amylase production kinetics, the glucose repression of alpha-amylase production is shown implicitly in Eq. (22), as high glucose concentration will inhibit the formation of sporangium. In the simplest case, the rate of alpha-amylase synthesis is neither substrate nor age dependent, and is considered as constant. The time between Stage 0 and Stage 2 of sporulation is set to 2 hr (Doi et al., 1989).

The following equations can be written for a fed-batch operation.

Glucose balance:

\[
\frac{dS}{dt} = - \frac{\mu_x x_v}{Y_{x/s}} + \frac{F(S_0 - S)}{V}
\]  

(10)

\[
\mu_v = \frac{\mu_{max} S}{k_s + S}
\]  

(11)
Vegetative cell number balance:

\[ \frac{dx_v}{dt} = (\mu_v - \mu_s)x_v - \frac{F}{V}x_v \]  \hspace{1cm} (12)

For batch culture after glucose is consumed, Eq. (13) is obtained,

\[ \frac{dx_v}{dt} = (-k_d - \mu_s)x_v \]  \hspace{1cm} (13)

Eq. (14) is written according to Dawes et al. (1970),

\[ \mu_s = k_1 - k_2\mu_v \quad (\mu_v \geq 0.1) \]  \hspace{1cm} (14)

In this work, the sporulation rate is linearly related to the specific vegetative growth rate when the specific vegetative growth rate is below 0.10 hr\(^{-1}\), so Eq. (15) is written,

\[ \mu_s = k_3 + (\frac{k_1 - k_3}{0.1} - k_2)\mu_v \quad (\mu_v < 0.1) \]  \hspace{1cm} (15)

Sporangium number balance can be written as follows (Fredrickson et al., 1967):

\[ \frac{\partial(x_s V)}{\partial t} + \frac{\partial x_s}{\partial \tau} V = 0 \]  \hspace{1cm} (16)
Applying the boundary condition,

\[ x_s(0, t) = \mu_s(t)x_v(t) \]  

(17)

Eq. (16) can be solved analytically to give (Fordyce, et al., 1996),

\[ x_s(t, \tau)V(t) = x_s(t-\tau, 0)V(t-\tau) = \mu_s(t-\tau)x_v(t-\tau)V(t-\tau) \]

(18)

Eq. (18) leads to,

\[ x_s(t) = \int_0^{\tau/\mu_s(t-\tau)x_v(t-\tau)V(t-\tau)} \frac{V(t)}{d\tau} \]

(19)

When \( \tau = \tau_m \), \( X_s(t, \tau_m) \) can be obtained from Eq. (18), so the spore number balance can be written as follows.

\[ \frac{dx_v}{dt} = \frac{\mu_s(t-\tau_m)x_v(t-\tau_m)V(t-\tau_m)}{V(t)} - \frac{F}{V}x_e \]

(20)

\( X_{\text{nons}pore} \) is calculated from Eq. (21),

\[ x_{\text{nons}pore} = x_v + x_s(t) \]

(21)
Alpha-amylase production is,

\[
\frac{dE}{dt} = \int_0^t k_e x_s(t, \tau) d\tau - \frac{F}{V} E
\]  

(22)

Feed rate is defined as,

\[
\frac{dV}{dt} = F
\]

(23)

The model above consists of one ordinary differential equation with a delay term. The Runge-Kutta method can be used to solve such a delay differential equation. The Runge-Kutta method provides solutions only at mesh points, while the x value for the delay term may fall between two mesh points. An approximate solution between mesh points should be given. The formula for a fifth degree polynomial interpolation has been published, but it is too complicated, so Shampire suggested that y value be evaluated only at the midpoint, and then the quartic polynomial interpolation with these values be obtained (Hairer et al., 1993). Quartic polynomial interpolations performed as follows:

\[
\begin{align*}
  u(0) &= y_0, & u'(0) &= hf(x_0, y_0) \\
  u(1) &= y_1, & u'(1) &= hf(x_0 + h, y_1) \\
  u(1/2) &= y_{1/2} 
\end{align*}
\]

Parameter estimation was done using a Fortran program. First, the initial guesses of parameters were entered, the simplex subroutine called the objective function, which is the weighted square of the residuals between predicted and experimental values. The predicted values are obtained by calling the subroutine for the delay differential equation.
The subroutine for delay differential equation was obtained by modifying the subroutine provided by Hairer et al. (1993).

5.2.1 Batch culture with cell lysis

A batch experiment was performed. The time courses of the concentrations of glucose, alpha-amylase, nonspore, and spore are shown in Fig. 5.3. It can be seen that cell lysis occurred immediately after glucose was depleted, which is consistent with the report for *Bacillus amyloliquefaciens* by Ponzo et al. (1991). In their work, they noticed that no apparent degradation of alpha-amylase by protease was found during the cell lysis. An extension to the cell lysis phase was needed to study the spore formation kinetics and thermo-stable alpha-amylase production kinetics. As shown in Fig. 5.3, the thermo-stable alpha-amylase concentration shot up shortly after the depletion of glucose, while the spore formation showed a longer delay. Fig. 5.4 shows the change of dissolved oxygen with time obtained using the LabVIEW software. At 10.9 hr, the dissolved oxygen stopped going down and started to increase quickly, then maintained almost a constant value afterwards. Parameters used in model simulation were determined by fitting the model with the data, and the values for the parameters are listed in Table 5.1.

In their work, Fordyce et al. (1996) first examined the sporulation rate in batch culture, then they used a piecewise linear basis function to describe the relationship
Fig. 5.3. Batch culture with cell lysis.
Fig. 5.4. Time course of dissolved oxygen in batch culture with cell lysis. The time interval for each data reading is 120 s.
Table 5.1. Summary of experimentally determined parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_{\text{max}}$</td>
<td>0.75 (hr$^{-1}$)</td>
</tr>
<tr>
<td>$k_s$</td>
<td>0.10 (g/l)</td>
</tr>
<tr>
<td>$k_i$</td>
<td>36.6 (g/l)</td>
</tr>
<tr>
<td>$Y_{x/s}$</td>
<td>$6.0 \times 10^{11}$ (number/g)</td>
</tr>
<tr>
<td>$k_d$</td>
<td>0.166 (hr$^{-1}$)</td>
</tr>
<tr>
<td>$k_1$</td>
<td>0.003 (hr$^{-1}$)</td>
</tr>
<tr>
<td>$k_2$</td>
<td>0.005</td>
</tr>
<tr>
<td>$k_3$</td>
<td>$1.95 \times 10^{-4}$ (hr$^{-1}$)</td>
</tr>
<tr>
<td>$\tau_m$</td>
<td>8 (hr)</td>
</tr>
<tr>
<td>$k_e$</td>
<td>$2.5 \times 10^{-6}$ (U/number/hr)</td>
</tr>
</tbody>
</table>
between sporulation rate and internal substrate concentration. They found that the sporulation rate was relatively sensitive for the batch run when the maximum death rate was reached, namely the zero internal substrate concentration. At the other non-zero substrate concentrations, they set the sporulation rate to zero. This rough description of the sporulation rate resulted in the poor fit of data for their step-test. Thus, in their work the growth and sporulation parameters were re-estimated using both the batch and step-test data by putting a relatively dense linear element near the zero internal substrate concentration. They found that there existed a maximum sporulation rate near the zero internal substrate concentration equal to 0.042 hr$^{-1}$ by drawing the graph of the sporulation rate versus internal substrate concentration, and it implied that there existed a maximum sporulation rate around a dilution rate of 0.15 hr$^{-1}$ (calculated in this work for their case). The newly established sporulation rate as a function of the internal substrate concentration could fit the step-test data well. However, they did not show the fit to the batch data with the newly established sporulation rate.

In this work, the simulation result for the sporulation rate versus the specific vegetative growth rate is shown in Fig. 5.5. The lower value for the sporulation rate compared to the work of Dawes et al. (1970) may be due to the different strain and the presence of amino acids in the medium that may inhibit the sporulation (Schaeffer et al., 1965). The fitting of the batch data in this work was reasonably good, and a better fit of alpha-amylase production could be achieved by further adjusting the rate of alpha-amylase production while fixing other parameters. The unstructured glucose inhibition model proposed by Shene et al. (1999) was applied to this system as a comparison
Fig. 5.5. Sporulation rate vs. specific vegetative growth rate.
between models. From the parameter estimation and simulation results, the unstructured glucose inhibition model will result in the earlier take-off of the thermo-stable alpha-amylase concentration and much flatter curve at the later stage for the batch run with cell lysis.

5.2.2 Batch and constant feed

In the fermentation industry, fed-batch is preferable to batch or continuous culture for the production of secondary metabolites, although the production in continuous culture is the most efficient (Wiebe et al., 2000). Fed-batch culture can also avoid the enrichment of plasmid free cells that would occur in continuous culture. The appearance of non-production variants for the wild-type strain in a chemostat production system has been reported (Heineken et al., 1972). Fed-batch culture can extend the stationary phase for alpha-amylase production and provide a long duration of the sensitive range for low glucose concentration.

As shown in Fig. 5.6, after the batch phase was over, constant feeding started with a glucose concentration of 2 g/l in the feed. The feed rate was low such that the dilution rate during the fed-batch culture was below 0.12 hr⁻¹, which was the threshold for accumulation of acetate reported by Snay et al. (1989). Nonspore concentration was maintained around a constant value in the constant feed phase after the rapid increase during the batch growth phase. The much higher enzyme production rate was obtained in the constant feed phase at the low constant feed rate (0.18 lhr⁻¹) as compared to the one at
cell lysis phase in Fig. 5.3. Slight adjustment of some parameters determined from batch culture with cell lysis gave better simulation results for nonspore, glucose, alpha-amylase and spore concentrations as shown in Fig. 5.6. In Fig. 5.6, it can be seen that the model fits the data well, and it further tests the sporulation kinetics around very low substrate concentrations.

There is another kind of recombinant *B. subtilis* in which the sporulation process is blocked at Stage 2, and the recombinant protein is produced during Stage 0 to Stage 2 without spore formation. This kind of strain has been reported by Oh *et al.* (1995) for subtilisin production by *B. subtilis DB104* (Δnpr Δapr) (Em′) spoIIG (BIm′):pMK101 (Cm′) in fed-batch and continuous cultures, also in Pierce *et al.* (1992) for subtilisin production by *B. subtilis* RS7907 in fed-batch culture. The model presented in this work can also be applied to this kind of strain. The model helps to explain the increased production capabilities for this kind of asporogenous strain by extending the production phase as compared to the wild-type sporogenous strain.

### 5.2.3 Parameter sensitivity analysis

In a model, if a perturbation of some parameter does not have any effect on the model prediction, this parameter is considered a fixed parameter, otherwise this parameter is called an adjustable parameter (Fordyce *et al.*, 1996). Parameter sensitivity analysis was performed in this work. For the batch culture with cell lysis, the prediction of alpha-amylase production, nonspore concentration and spore concentration at the time
Fig. 5.6. Simulated results of a batch with constant feed rate. Feed rate was 0.18 l/hr.

Broth volume when feed started was 6 liters, and constant inlet substrate concentration was 2 g/l.
of last experimental data sample was evaluated by varying one parameter and fixing all other parameters. The results are shown in Table 5.2. Table 5.2 shows that if $a_v$ was increased by 10% and all other parameters were fixed, the simulation result for alpha-amylase concentration at the time of last experimental data would be decreased by 8.2% as compared to the original simulation result, and the simulation results for nonspore concentration and spore concentration were decreased by 15.1% and 6.7%, respectively. From Table 5.2, it can be seen that the model is less sensitive to changes in $\tau_m$, $k_3$ and $k_v$ and is most sensitive to changes in $a_v$, $Y_{x/s}$, $k_d$, $k_1$, $k_2$ and $k_e$. 
Table 5.2. Parameter sensitivity analysis

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Perturbation of parameter</th>
<th>Perturbation of alpha-amylase concentration</th>
<th>Perturbation of nonspore concentration</th>
<th>Perturbation of spore concentration</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_v$</td>
<td>+10%</td>
<td>-8.2%</td>
<td>-15.1%</td>
<td>-6.7%</td>
</tr>
<tr>
<td></td>
<td>-10%</td>
<td>+16.3%</td>
<td>+17.7%</td>
<td>+12.5%</td>
</tr>
<tr>
<td>$k_v$</td>
<td>+10%</td>
<td>+5.14%</td>
<td>+1.93%</td>
<td>+4.43%</td>
</tr>
<tr>
<td></td>
<td>-10%</td>
<td>-4.98%</td>
<td>-2.19%</td>
<td>-4.37%</td>
</tr>
<tr>
<td>$Y_{x/s}$</td>
<td>+10%</td>
<td>+10.3%</td>
<td>+10.9%</td>
<td>+10.4%</td>
</tr>
<tr>
<td></td>
<td>-10%</td>
<td>-9.97%</td>
<td>-12.2%</td>
<td>-9.89%</td>
</tr>
<tr>
<td>$k_d$</td>
<td>+10%</td>
<td>-8.2%</td>
<td>-31.6%</td>
<td>-4.3%</td>
</tr>
<tr>
<td></td>
<td>-10%</td>
<td>+9.49%</td>
<td>+46.1%</td>
<td>+4.98%</td>
</tr>
<tr>
<td>$\tau_m$</td>
<td>+10%</td>
<td>-5.3%</td>
<td>0</td>
<td>-1%</td>
</tr>
<tr>
<td></td>
<td>-10%</td>
<td>+5.79%</td>
<td>0</td>
<td>+0.6%</td>
</tr>
<tr>
<td>$k_1$</td>
<td>+10%</td>
<td>+15.2%</td>
<td>0</td>
<td>+12.9%</td>
</tr>
<tr>
<td></td>
<td>-10%</td>
<td>-10.1%</td>
<td>0</td>
<td>-9.3%</td>
</tr>
<tr>
<td>$k_2$</td>
<td>+10%</td>
<td>-5.98%</td>
<td>0</td>
<td>-5.24%</td>
</tr>
<tr>
<td></td>
<td>-10%</td>
<td>+10.3%</td>
<td>0</td>
<td>+9%</td>
</tr>
<tr>
<td>$k_3$</td>
<td>+10%</td>
<td>+5.7%</td>
<td>0</td>
<td>+5.85%</td>
</tr>
<tr>
<td></td>
<td>-10%</td>
<td>-5.7%</td>
<td>0</td>
<td>+5.8%</td>
</tr>
<tr>
<td>$k_e$</td>
<td>+10%</td>
<td>+10%</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>-10%</td>
<td>-10%</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
6.1 Carbon balance

The equation for cell growth is expressed as follows:

\[ \nu + q_{O_2} \rightarrow \mu + q_{CO_2} \]  \hspace{1cm} (24)

Carbon balance:

\[ a_1 \nu = a_2 \mu + a_3 q_{CO_2} \]  \hspace{1cm} (25)

where \( a_1 = 0.4 \) g.g\(^{-1}\) (carbon content of glucose).

The formula for the cell biomass is \( CH_{1.8}O_{0.5}N_{0.2} \) (Mignone et al., 1996). Thus,

\( a_2 = 0.49 \) g.g\(^{-1}\) (carbon content of the dry cell mass)

\( a_3 = 0.27 \) g.g\(^{-1}\) (carbon content of carbon dioxide)

Oxygen balance:

\[ A \nu = B \mu + q_{O_2} \]  \hspace{1cm} (26)

The equation for burning the biomass is as follows:

\[ CH_{1.8}O_{0.5}N_{0.2} + 1.05O_2 = CO_2 + 0.6H_2O + 0.2NH_3 \]  \hspace{1cm} (27)

The oxygen consumption for burning the biomass is

\[ B = 1.05 \times 32/24.6 = 1.37 \) (g.g\(^{-1}\))

The oxygen consumption for burning the glucose is:

\[ A = 1.067 \) (g.g\(^{-1}\)) \]
The carbon balance and oxygen balance in this case are:

\[ 0.4\nu = 0.49\mu + 0.27q_{CO_2} \]  

(28)

\[ 1.067\nu = 1.37\mu + q_{O_2} \]  

(29)

Multiplying both sides of the two equations above with cell mass concentration X gives,

\[ \frac{dX}{dt} = 11.45Q_{CO_2} - 15.89Q_{O_2} \]  

(30)

\[ \frac{dS}{dt} = 14.70Q_{CO_2} - 19.47Q_{O_2} \]  

(31)

Thus, by on-line measurements of \( Q_{O_2} \) and \( Q_{CO_2} \), the on-line information about the cell growth and substrate consumption can be obtained.

### 6.2 Effects of substrate and metabolite on cell growth and alpha-amylase production

Ammonium concentration must be kept low since at higher level it inhibits growth in \( E. \ coli \) fermentation (Thompson et al., 1985; Yee et al., 1993). The effect of initial \( (NH_4)_2SO_4 \) concentration on cell growth and alpha-amylase production for \( B. \ subtilis \) was studied in shake flask culture at 37 °C in this work. The initial concentration of \( (NH_4)_2SO_4 \) in minimal medium was varied. Alpha-amylase concentration was assayed after 26 hr incubation. In Fig. 6.1, 50 g/l \( (NH_4)_2SO_4 \) inhibited both cell growth and alpha-amylase production, while 20 g/l \( (NH_4)_2SO_4 \) did not show any inhibition.

It was reported that a high concentration of amino acid would inhibit cell growth (Mizutani et al., 1986). The effect of initial phe, trp and tyr concentrations on cell growth
and alpha-amylase production for *B. subtilis* was studied in shake flask culture at 37 °C in this work. The concentrations for phe, trp and tyr were varied from 0.05 to 1 g/l in the minimal medium simultaneously. Due to solubility limitation of tyr in the minimal medium, no test was done for 1.5 g/l phe, trp and tyr. Alpha-amylase concentration was assayed after 17 hr incubation. Fig. 6.2 shows that 1 g/l phe, trp and tyr did not inhibit cell growth.

Sodium acetate is a major metabolite in the *B. subtilis* metabolic pathway (Fig. 2.3). The effect of initial sodium acetate on cell growth and alpha-amylase production was studied in a minimal medium containing 3 g/l KH$_2$PO$_4$ and 3 g/l K$_2$HPO$_4$ in shake flasks. It was reported that the pH of the medium would affect the dissociation of sodium acetate. The lower the pH in the medium, the higher the concentration of protonated acetate, and the higher the inhibition will be (Tseng *et al.*, 1991). The effect of initial sodium acetate concentration on cell growth and alpha-amylase production for *B. subtilis* was studied in shake flask culture at 37 °C. The initial concentrations of sodium acetate in the minimal medium were varied from 0 to 3 g/l. Alpha-amylase concentration was assayed after 12 hr incubation. From Fig. 6.3, it can be seen that accumulation of 0.3 g/l sodium acetate caused a long lag phase for growth, and reduced the alpha-amylase production by about 50% compared to the control. 1 g/l sodium acetate inhibited both cell growth and alpha-amylase production seriously.
Fig. 6.1a. Effect of initial concentration of ammonium sulfate on cell growth.

Fig. 6.1b. Effect of initial concentration of ammonium sulfate on alpha-amylase production.
Fig. 6.2a. Effect of initial concentrations of trp, tyr, phe on cell growth.

Fig. 6.2b. Effect of initial concentrations of trp, tyr, phe on alpha-amylase production.
Fig. 6.3a. Effect of initial concentration of sodium acetate on cell growth.

Fig. 6.3b. Effect of initial concentration of sodium acetate on alpha-amylase production.
6.3  Exponential fed-batch culture with dual feeding

The relatively simple algorithm for exponential fed-batch is described below:

If the specific growth rate $\mu$ is constant, Eq. (32) is valid,

$$X_V = X_0 V_0 e^{\mu t}$$  \hspace{1cm} (32)

Assume that glucose amount in the broth is constant,

$$\frac{d(VS)}{dt} = 0$$  \hspace{1cm} (33)

The mass balance equation for glucose is:

$$\frac{d(VS)}{dt} = F S_0 - \frac{V \mu X}{Y_{x/s}}$$  \hspace{1cm} (34)

Substitute Eqs. (32) and (33) into Eq. (34),

$$F = \frac{\mu X_0 V_0 e^{\mu t}}{S_0 Y_{x/s}}$$  \hspace{1cm} (35)

In this work, the following characteristics of the exponential fed-batch culture were assumed:

1) There is a quasi-steady state on glucose concentration when it is essentially kept at zero.
2) $\mu$ can be maintained at an essentially constant value below the specific growth rate which will cause the accumulation of acid by-products or reduction of alpha-amylase production.

3) The specific oxygen uptake rate is low, so the DO is relatively in excess.

4) The biomass yield on glucose is essentially constant.

There are three types of media: defined, complex and semi-defined. Defined media are generally used to obtain high cell-density, as the nutrient concentrations are known and can be controlled during the culture. Whether defined medium or semi-defined medium is applied in exponential feed depends on the strain. Some recombinant strains cannot grow well in a defined medium, so a semi-defined medium is needed in order to boost product formation (Delacruz et al., 1992).

Balanced nutrients in the fermentation broth are essential for high cell density cultures. Overfeeding of mineral ions can produce high ionic strength that may affect product formation and cell growth. High solute concentration reduces the maximum dissolved oxygen concentration in a medium and lowers the rate of oxygen transfer to the medium (Blackwell, 1989; Jensen et al., 1990).

In order to make the feed medium simple, for some nutrients such as mineral ions, they are supplied as much as possible in the starting medium instead of the feed medium (Lee, 1996). For the other nutrients in the feed, a balanced feed medium should be developed. The equation for the other nutrient concentrations in the feed is as follows:

$$\frac{dS_N}{dt} = \frac{F}{V} (S_{NF} - S_N) - \frac{\mu X_0 V_0 e^{\mu t}}{V Y_{X/S_N}}$$

(36)

Substituting Eq. (35) into Eq. (36), Eq. (37) is obtained:
\[
\frac{dS_N}{dt} = \frac{\mu X_0 V_0 e^{\mu t}}{V} \left( \frac{S_{NF} - S_N}{Y_{X/S} S_0} - \frac{1}{Y_{X/S_N}} \right)
\]  

(37)

Assume that the nutrient concentration in the fermenter is constant,

\[
\frac{dS_N}{dt} = 0
\]  

(38)

Eq. (38) leads to Eq. (39),

\[
S_{NF} = \frac{Y_{X/S}}{Y_{X/S_N}} S_0 + S_N
\]  

(39)

The estimation of yield coefficients for some nutrients was carried out in shake flask culture. The media composition of the different nutrient limited batch culture is given in Table 6.1. In order to minimize the effect of carrying over of the medium from the inoculum, a small amount of inoculum (0.1%) was introduced to shake flasks. The cultures were carried out at 37 °C. The yield coefficients for different limiting nutrients can be obtained. The yield coefficients for respective limiting nutrients are shown in Table 6.2.

The nutrient yield coefficients were substituted into Eq. (33) to calculate the concentrations of nutrients other than glucose in the feed. The calculated values were used to formulate the defined minimal medium for Feed Stream 1 and Feed Stream 2 as shown in Chapter 3. The initial fermenter liquid volume was six liters. The reservoir for
Stream 1 was one liter and 250 ml for Stream 2, respectively. The feed rates for Feed Stream 1 and Feed Stream 2 are expressed as follows:

\[
F_1 = \frac{\mu X_0 V_0 e^{\mu t}}{Y_{x/s} S_0} \quad (40)
\]

\[
F_2 = \frac{F_1}{4} \quad (41)
\]

The feed rate ratio for Stream 2 to Stream 1 at any time was 1:4, which was the ratio of the feed volumes of Stream 2 to Stream 1.

After inoculation, the cells grew in the batch mode for about 9.3 hr until glucose was depleted. At this time, the cell density was 1.8 g/l. The two feed streams were then pumped into the broth at the same time at different exponential rates. After 18 hr of exponential fed-batch culture with the specific growth rate set to 0.15 hr\(^{-1}\), the exponential feed rate was reduced by setting the specific growth rate to 0.05 hr\(^{-1}\) in Eq. (40) so that DO in the broth could be above 20% (Akesson et al., 1999; Konstantinov et al., 1990). After another 2 hr of feeding, the feed rates were further reduced to constant feeding with the feed rates equal to 0.06 l/hr and 0.015 l/hr for Feed Stream 1 and Feed Stream 2, respectively.

The cell density, glucose concentration, acetate concentration, alpha-amylase concentration and specific alpha-amylase concentration plotted as a function of time are shown in Fig. 6.4. Fig. 6.4 shows that the final cell density reached 24.2 g/l and the corresponding alpha-amylase concentration reached 71.4 U/ml. The overall biomass yield
Table 6.1. Media composition of nutrient-limited batch culture

<table>
<thead>
<tr>
<th>Components</th>
<th>Units</th>
<th>Glucose</th>
<th>N</th>
<th>Mg</th>
<th>P</th>
<th>K</th>
<th>Trp</th>
<th>Tyr</th>
<th>Phe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glucose</td>
<td>g/l</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>(NH₄)₂SO₄</td>
<td>g/l</td>
<td>2.5</td>
<td>0.1</td>
<td>2.5</td>
<td>2.5</td>
<td>2.5</td>
<td>2.5</td>
<td>2.5</td>
<td>2.5</td>
</tr>
<tr>
<td>K₂HPO₄</td>
<td>g/l</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>KCl</td>
<td>g/l</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.005</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>KH₂PO₄</td>
<td>g/l</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
<td>0.025</td>
<td>-</td>
<td>1.5</td>
<td>1.5</td>
<td>1.5</td>
</tr>
<tr>
<td>Na₂HPO₄</td>
<td>g/l</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>NaH₂PO₄</td>
<td>g/l</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1.5</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MgSO₄·7H₂O</td>
<td>g/l</td>
<td>0.25</td>
<td>0.25</td>
<td>0.005</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>Trp</td>
<td>g/l</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.005</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Tyr</td>
<td>g/l</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>-</td>
<td>0.005</td>
<td>-</td>
</tr>
<tr>
<td>Phe</td>
<td>g/l</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>-</td>
<td>-</td>
<td>0.005</td>
</tr>
<tr>
<td>Sodium citrate</td>
<td>g/l</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>CaCl₂</td>
<td>g/l</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.1</td>
</tr>
<tr>
<td>MnSO₄</td>
<td>g/l</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.01</td>
</tr>
<tr>
<td>FeSO₄</td>
<td>g/l</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.01</td>
</tr>
<tr>
<td>ZnSO₄</td>
<td>g/l</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.002</td>
</tr>
</tbody>
</table>
Table 6.2. Nutrient yield coefficients from batch culture in shake flasks

<table>
<thead>
<tr>
<th>Limiting nutrients</th>
<th>Yields (g cell/g nutrient)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glucose</td>
<td>0.4</td>
</tr>
<tr>
<td>N</td>
<td>5.8</td>
</tr>
<tr>
<td>Mg</td>
<td>701</td>
</tr>
<tr>
<td>K</td>
<td>106</td>
</tr>
<tr>
<td>P</td>
<td>51</td>
</tr>
<tr>
<td>Tyr</td>
<td>83.2</td>
</tr>
<tr>
<td>Try</td>
<td>130.7</td>
</tr>
<tr>
<td>Phe</td>
<td>54.5</td>
</tr>
</tbody>
</table>
was 0.39 g cell/g glucose. For the batch culture with the initial glucose concentration of 8 g/l, the final cell density was 2.3 g/l, and the corresponding alpha-amylase concentration was 1.5 U/ml. The corresponding biomass yield was only 0.28 g cell/g glucose. In the fed-batch experiment, acetate accumulated in the batch phase was reassimilated in the beginning of the fed-batch phase (Robbins et al., 1989). Specific alpha–amylase concentration showed a little increase near the end of culture due to the reassimilation of acetate.
Fig. 6.4. Exponential fed-batch culture with dual feeding.
CHAPTER 7
SUMMARY AND RECOMMENDATIONS

7.1 Conclusions

The flexible and automatic monitoring and control of \textit{B. subtilis} fermentation using LabVIEW software were achieved by establishing a serial interface between the 22-liter B. Braun Biostat C fermenter’s control unit and a personal computer.

Effect of substrate concentration on cell growth and alpha-amylase production was studied in shake flask cultures. A concentration of 20 g/l ammonium sulfate in the culture medium did not inhibit cell growth and alpha-amylase production in the fed-batch culture of \textit{B. subtilis}. High phe, trp and tyr concentrations in the culture medium did not inhibit cell growth or alpha-amylase production.

In order to minimize the formation of toxic by-products such as acetate and achieve high cell density, a dual exponential fed-batch strategy was employed. Due to low solubilities of tyrosine and tryptophan in Feed Stream 1 containing concentrated glucose and other nutrients, tyrosine and tryptophan were removed from Feed Stream 1 and were dissolved in 14.4\% ammonium water to form Feed Stream 2. By dual feeding the two streams at different exponential feed rates, the cells grew exponentially and a high cell density of 24.2 g/l accompanied by a final alpha-amylase activity of 71.4 U/ml were achieved. The overall biomass yield was 0.39 g cell/g glucose. In contrast, for the batch culture with the initial glucose concentration of 8 g/l, the final cell density was 2.3
g/l, and alpha-amylase concentration was 1.5 U/ml. The corresponding biomass yield was only 0.28 g cell/g glucose. Obviously, the dual feeding strategy improved fermentation results considerably.

The segregated model presented in this work included three distinguishable cell states and the transition from vegetative phase to sporangium and finally to mature spore. An age-based population balance model was applied to describe the maturity of sporangia toward the formation of spores. Compared to existing models in the literature, the contributions of the model in this work are: 1) An inherent mathematic relationship between sporulation rate and specific vegetative growth rate was established. 2) Alpha-amylase production kinetics which are a linear function of the total sporangia from Stage 0 to Stage 2 was proposed, and the glucose repression of alpha-amylase production was shown implicitly in the alpha-amylase production kinetics. The model proposed in this work can successfully describe product formation in both batch and fed-batch cultures.

7.2 Recommendations

Future work can include plasmid loss kinetics in batch and fed-batch cultures of recombinant *B. subtilis* in non-selective media. The segregated model can be modified to incorporate acetate production, consumption and inhibition of cell growth and product formation. The correlation between overall biomass yield on glucose and glucose concentration can also be further studied.
Feeding strategies such as DO-stat, pH-stat and carbon dioxide evolution rate can be applied to *B. subtilis* for high cell density culture. The results should be compared with those from dual exponential fed-batch culture. The respective advantages and disadvantages of feeding strategies should be investigated.
BIBLIOGRAPHY


APPENDIX A  Telegrams for LabVIEW

Telegram Format

All telegrams must have the following format. Depending on the data group, however, the message field can change in its structure. The status field always indicates the actual status of the telegram sender.

|----|-----|:|---------|:|-----|:|-------|:|-------------|:|----|----|

SC  Start character
ADR  Unit address
SEQ  Sequence
STAT  Status of sender
CS  Check Sum
EM  End Mark

Structure of message field:

| DES | : | NO | : | VAL | : | NO | : | VAL | : | ... | DES | : | NO | : | VAL | : | ... |

DES  Designation (PV, CS...)
NO  Channel Number
VAL  Value

The following characters are required for data transmission control

*  Start character for host telegram
# Start character for substation telegram
:
@  End mark for each telegram

Unit address

The unit address here is 1. It represents COM2 of PC.

Sequence

In the case of transmission error the host system can automatically repeat a command by means of time out supervision and the sequence counter. The number of repetitions is not limited and will be determined by the host system. Sequence is the number of times this message has been sent to the substation. 0 represents the first time that the message is sent. 1 represents the second time the same message is sent. The substation stores the last sent response until the next command is received in order to be able to repeat the data (incremented sequence) if necessary.

Status of sender

The status field always indicates the actual status of the sender. Between host and substation, however, different function assignments exist. The status definition value of the host is defined as 0 here, and the status definition value of the substation is 16 here.
Check Sum

The check sum is formed by addition of all ASCII values from the start sign (* or #) up to and including the delimiter(;) directly in front of the check sum. The calculated value cannot exceed the value of 32385.

Designation

PV

String PV means process value. The direction is from substation to host.

CS

String CS means controller set point. For DFC-2 hardware the designation CS for controller set point can be set by the host only.

CM

String CM means controller mode. For DFC-2 hardware the designation CM for controller mode can be set by the host only.

There are also many other designations such as PA, BC, PS and so on for DCU hardware, but for DFC-2 hardware, it only has PV, CS, CM.

Channel number

Different designation has different definition of channel number. For the designation PV, press the “process values” key of Biostat C controller keypad, a list of process values for the corresponding parameters will appear. Temperature comes first, so channel number 1
represents temperature. The second one of the list is stirrer speed, so channel number 2 represents stirrer speed. Similarly, the corresponding parameter for each channel number till 14 can be obtained. For the designation CS and CM, press the “control loops” key, a table for controllers will appear. Temperature comes first as channel 1, so channel number 1 represents temperature. Similarly, the corresponding control variable for each channel number till 7 can be obtained.

Value

The data type could be byte, integer, real and string. The range of values for the types of data used in the data field definitions are as follows:

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte</td>
<td>0-255</td>
</tr>
<tr>
<td>Integer</td>
<td>0-32767</td>
</tr>
<tr>
<td>Real</td>
<td>-1000.0-+1000.0</td>
</tr>
<tr>
<td>String</td>
<td>&lt;=255 characters</td>
</tr>
</tbody>
</table>

The transmission of the process values with the mark “real” is standardized. The range of values has been defined as 0-100%=0-1000.0. For the designation CM, value1 means AUTO(ON). Value 0 means OFF.

Response telegrams from the substation

The contacted substation responds within the timeout limit (500 ms) with the required response telegram or with an error telegram. The substation sends an acknowledgement or error telegram after the commands SU and DS and after the command DR.
Substation sends the process data or an error telegram to the host. The acknowledgement telegrams have the following structure.

1) Start character and address
2) Reflection of received command
3) Reflection of received sequence
4) Current status field
5) Check sum and end mark

Substation error telegram

In case of an error the substation generates an error telegram itself. Since there may be different error causes, it is differentiated between logical errors, data errors and format errors. Each type of error has its specific designation. The interpretation of the error telegram was given by B. Braun’s technical manual entitled “Description of Serial Interface Between Host and Control Units of the B. Braun Biotech International GMBH” (Rev. 09/96, Sept. 1996).
APPENDIX B    LabVIEW Programs

Calc Chk Sum.VI

Calc Chk Sum.VI can calculate the checksum of a telegram. Figs. B1 and B2 are the front panel and block diagram of Calc Chk Sum.VI, respectively.

Test 1.VI

Test 1.VI performs bidirectional serial communication using a serial port on the computer. Figs. B3 and B4 are the front panel and block diagram of Test 1.VI, respectively.

Switch the Biostat C on and input *1:SU:0:0:PV:5:2:1030@ in the “Write String” box on the front panel, and run the program. In the “Read String” box, #1:SU:0:16:635@ appears at once as the response to the input string. After this initial procedure, the other commands DR and DS can be used. Input *1:DR:0:0:569@ in the “Write String” box, and run the program. Then, in the “Read String” box, a string of current process values appears. Input *1:DS:0:0:CS:2:133.3:CM:2:1:1609@ in the “Write String” box, and run the program. In the “Read String” box, #1:DS:0:16:618@ appears. At the same time the stirrer speed of the bioreactor starts from 0 to reach the set point 200 rpm.
Fig. B1. Front panel of LabVIEW program file Calc Chk Sum.VI that calculates the checksum.
Fig. B2. Block diagram of LabVIEW program file Calc Chk Sum.VI that calculates the checksum.
Fig. B3. Front panel of LabVIEW program file Test1.VI that realizes the serial communication.
Fig. B4. Block diagram of LabVIEW program file Test 1.VI that realizes the serial communication.
Process value.VI

Figs. B5 and B6 are the front panel and block diagram of Process value.VI, respectively. On pages 5 - 40 of the Biostat C operating manual, there is a table of the lower limit and upper limit of process values for temperature, stirrer speed, pH, pO₂, acid, base and so on. The range of value for the real number in data transmission field is -1000.0...+1000.0, which means the upper limit 150 °C should correspond to 1000.0, and the number received from the message should be multiplied by 0.15 to get the true number of the temperature. Similarly, the true values for the process variables can be obtained as shown in Table B1.
Table B1. Conversion between the true value and the number received from the message.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Upper limit</th>
<th>Physical unit</th>
<th>True value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stirrer speed</td>
<td>1500</td>
<td>rpm</td>
<td>M×1.5</td>
</tr>
<tr>
<td>pH</td>
<td>12</td>
<td>pH</td>
<td>M/100+2</td>
</tr>
<tr>
<td>DO</td>
<td>100</td>
<td>%</td>
<td>M/10</td>
</tr>
<tr>
<td>Acid</td>
<td>500</td>
<td>ml</td>
<td>M/2</td>
</tr>
<tr>
<td>Base</td>
<td>500</td>
<td>ml</td>
<td>M/2</td>
</tr>
<tr>
<td>Substrate</td>
<td>10000</td>
<td>ml</td>
<td>M×10</td>
</tr>
<tr>
<td>Airflow rate</td>
<td>30</td>
<td>l/min</td>
<td>M×0.03</td>
</tr>
</tbody>
</table>

Note: M is the number received from the message for an individual parameter.
Fig. B5. Front panel of LabVIEW program file Process values.VI that displays current values of the process variables.
Fig. B6. Block diagram of LabVIEW program file Process values.VI that displays current values of the process variables.
DO.VI

Figs. B7 and B8 are the front panel and block diagram of DO.VI, respectively. When the user clicks the ON/OFF button to show OFF, the loop stops, and a file dialog box pops up for the user to save the data to a file.

Read file.VI

Figs. B9 and B10 are the front panel and block diagram of Read file.VI, respectively. When the user runs the program, a dialog box prompts the user to enter a file name.

Exponential.VI

Figs. B11 and B12 are the front panel and block diagram of Exponential.VI, respectively. On the front panel, there are many controls for the user to input data. The calibrated flow rate represents the maximum flow rate that the pump can handle, which can be obtained by the calibration of pump with details in the B. Braun Biostat C operating manual. The indicator shows the current flow rate. If the user clicks the ON/OFF button on the front panel to show OFF, the loop will stop, and the pump controller will be switched to OFF.
Fig. B7. Front panel of LabVIEW program file DO.VI that displays dissolved oxygen in a graph.
Fig. B8. Block Diagram of LabVIEW program file DO.VI that displays dissolved oxygen in a graph.
Fig. B9. Front panel of LabVIEW program file Read file.VI that reads the data from a file and displays the data in a graph.
Fig. B10. Block diagram of LabVIEW program file Read file.VI that reads the data from a file and displays the data in a graph.
Fig. B11. Front panel of LabVIEW program file Exponential.VI that realizes the exponential feed to the bioreactor.
Fig. B12. Block diagram of LabVIEW program file Exponential.VI that realizes the exponential feed to the bioreactor.
Exponentialdo.VI

Figs. B13 and B14 are the front panel and block diagram of Exponentialdo.VI, respectively. DO.VI and Exponential.VI are the subroutines of Exponentialdo.VI.
Fig. B13. Front panel of LabVIEW program file Exponentialdo.VI that realizes the exponential feed to the fermenter and DO monitoring at the same time.
Fig. B14. Block diagram of LabVIEW program file Exponentialdo.VI that realizes the exponential feed to the fermenter and DO monitoring at the same time.
APPENDIX C  Fortran Program

C*  CALCULATE THE MINIMUM OF A BINDING PROBLEM   *
C*                                                                 *
C*  NX:---INPUT, THE NUMBER OF PARAMETER                        *
C*  X(NX):---THE INITIAL VALUE BEFORE WORKING, THE RESULT      *
C*  AFTER WOKING                                              *
C*  FX:---OUTPUT, THE VALUE OF MINIMUM OF OBJECTIVE FUNCTION*  *
C*  NC:---INPUT, THE NUMBER OF EQUALITY BINDING                *
C*  NIC:---INPUT, THE NUMBER OF INEQUALITY BINDING             *
C*  SIZE:---INPUT, THE LENGTH OF INITIAL SIMPLE FIGURE         *
C*  ICONT:---OUTPUT, THE TIMES FOR WHICH OBJECTIVE FUNCTION    *
C*  DESCENDED                                               *
C*  ICONTR:---OUTPUT, THE ITERATIVE TIMES OF CALCULATING      *
C*  ´NENG XING DIAN'                                         *
C*  IR:---OUTPUT, WHEN '***' IS FAILURE, IR WILL BE THE LINE   *
C*  NUMBER.                                                 *
C*  IER:---OUTPUT                                             *
C*  1: '***' SUCCESSFUL                                      *
C*  -1: '***' FAILURE                                       *
C*  IMOS:---INPUT, THE LARGEST TIME OF ITERATE                *
C*  CONVER:---INPUT, PRECISION                               *
C*  FDIFER:---OUTPUT, THE ACTUAL ERROR OF X                   *
C*  IE:---OUTPUT                                             *
C*  1: REACH THE PRECISION                                   *
C*  -1: THE ITERATE TIME > IMOS                              *
C*  SUBROUTINE OBF(X,F):---THE OBJECTIVE FUNCTION, F: THE     *
C*  VALUE OF OBJECTIVE FUNCTION                              *
C*  SUBROUTINE CONSTR(X,R):---CALCULATE THE BINDING VALUE.    *
C*  R(1)-R(NC) FOR EQUALITY                                 *
C*  R(NC+1)-R(NC+NIC) FOR INEQUALITY                         *
C*  CAUTIONS:                                                *
C*  EQUALITY MUST BE LIKE FOLLOWER:                          *
C*  F(X)=S(X):------R(1)=F(X)-S(X)=0                         *
C*  IT IS TO SAY THAT EQUAITY MUST BE TRANSFERED             *
C*  INTO                                                    *
C*  R(1)=H(X) = 0                                            *
C*  INEQUALITY BINDING MUST BE LIKE FOLLOWER:                 *
C*  3<F(X)<12:------R(NC+1)=F(X)-3 AND                        *
C*  R(NC+2)=12-F(X)                                           *
C*  IT IS TO SAY THAT INEQUALITY MUST BE                     *
C*  TRANSFER INTO                                           *
C*  R(NC+N) =G(X) > 0                                        *
C* SUBROUTINE SUMR(X,SM,SM1,NX,NC,NIC,K8):                *
C*                                                        *
C**********************************************************
IMPLICIT double precision(A-H,O-Z)
dimension T(10)
c    print*, 'input the number of parameter NX'
c    read*, NX
   NX=5
c    print*, 'input the number of equality binding NC'
c    read*, NC
   NC=0
c    print*, 'input the number of inequality binding NIC'
c    read*, NIC
   NIC=5
c    print*, 'input the length of initial simple figure'
c    read*, SIZE
   SIZE=1.0D0
c    print*, 'input the number of the largest time of
*iteration'
c    read*, IMOS
   IMOS=10000
c    print*, 'input the precision'
c    read*, CONVER
   CONVER=0.01D0
c    print*, 'input the initial estimates of parameters'
c    read*, (T(I),I=1,NX)
   T(1)=1.D0
   T(2)=1.0D0
   T(3)=1.0D0
   T(4)=1.0D0
   T(5)=1.0D0

   call FLEXTO(T,FX,NX,NC,NIC,SIZE,ICONT,
   *ICONTR,IR,IER,IMOS,CONVER,FDIFER,IE)

   print*, 'THE VALUE OF MINIMUM OF OBJECTIVE FUNCTION
*FX IS'
   print*, FX
   print*, 'THE TIMES FOR WHICH OBJECTIVE FUNCTION
*DESCENDED ICONT IS'
   print*, ICONT
   print*, 'THE ITERATIVE TIMES ICONTR IS'
SUBROUTINE FLEXTO(T, FX, NX, NC, NIC, SIZE, *ICONTR, ICONT, IR, IER, IMOS, *CONVER, FDIFER, IE)
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
DIMENSION T(NX),X1(29,20),X2(20,20),F(20), *
SR(29),S(29),R(40),
*R(40),H(20),R3(40),FLG(3)
ALPHA=1.
BETA=0.5
GAMA=2.0
STEPF=SIZE
K1=NX+1
K2=NX+2
K3=NX+3
K4=NX+4
K5=NX+5
K6=NC+NIC
K7=NC+1
K8=NC+NIC
K9=K8+1
N=NX-NC
N1=N+1
IF(N1.GE.3) GOTO 50
N1=3
N=2
50
N2=N+2
N3=N+3
N4=N+4
N5=N+5
N6=N+6
N7=N+7
N8=N+8
XN=FLOAT(N)
XNX=FLOAT(NX)
XN1=FLOAT(N1)
XNC=FLOAT(NC)
R1A=0.5*(SQRT(5.)-1.)
R2A=R1A*R1A
R3A=R2A*R1A
L5=NX+5
L6=NX+6
L7=NX+7
X7=FLOAT(K7)
L8=NX+8
L9=NX+9
ICONT=1
NCONT=1
FDIFER=2.*(XNC+1.)*STEPF
FOLD=FDIFER
CALL SURM(T,S(N1),SR(N1),NX,NC,NIC,K8)
IF(SR(N1).LT.FDIFER) GOTO 201
INF=N1
IR=1
GOTO 500
201 CALL OBF(T,FTER)
237 STEP1=STEPF*(SQRT(XNX+1.)+XNX-1.)/(XNX*SQRT(2.))
STEP2=STEPF*(SQRT(XNX+1.)-1.)/(XNX*SQRT(2.))
ETA=(STEP1+(XNX-1.)*STEP2)/(XNX+1.)
DO 4 J=1,NX
T(J)=T(J)-ETA
4 CONTINUE
DO 101 J=1,NX
101 X2(1,J)=T(J)
DO 102 I=2,N1
DO 104 J=1,NX
104 X2(I,J)=STEP2+T(J)
L=I-1
X2(I,L)=STEP1+T(L)
102 CONTINUE
I=0
9 I=I+1
DO 6 J=1,NX
6 T(J)=X2(I,J)
CALL SURM(T,S(I),SR(I),NX,NC,NIC,K8)
IF(SR(I).LT.FDIFER) GOTO 8
IR=2
INF=I
GOTO 500
8    CALL OBF(T,F(I))
IF(I.NE.N1) GOTO 9
900  ICONT=ICONT+1
IF(ICONT.EQ.IMOS)GOTO 80
FH=F(1)
LHIGH=1
DO 16 I=2,N1
IF(F(I).LT.FH) GOTO 16
FH=F(I)
LHIGH=I
16   CONTINUE
41   FL=F(1)
LOW=1
DO 17 I=1,N1
IF(FL.LT.F(I)) GOTO 17
FL=F(I)
LOW=I
17   CONTINUE
DO 86 J=1,NX
86   T(J)=X2(LOW,J)
CALL SURM(T,S(LOW),SR(LOW),NX,NC,NIC,K8)
IF(SR(LOW).LT.FDIFER) GOTO 87
IR=3
INF=LOW
GOTO 500
203  CALL OBF(T,F(LOW))
GOTO 41
87   CONTINUE
DO 19 J=1,NX
SUM2=0.0
DO 20 I=1,N1
SUM2=SUM2+X2(I,J)
19   X2(N2,J)=1./XN*(SUM2-X2(LHIGH,J))
SUM2=0.0
DO 36 I=1,N1
36   SUM2=SUM2+(X2(I,J)-X2(N2,J))**2
DO 36 J=1,NX
SUM2=SUM2+(X2(I,J)-X2(N2,J))**2
36   CONTINUE
FDIFER=(XNC+1.)/XN1*SQRT(SUM2)
IF(FDIFER.LT.FOLD) GOTO 98
FDIFER=FDIFER
98   FOLD=FDIFER
FTER=F(LOW)
NCONT=NCONT+1
IF(NCONT.LT.4*N1) GOTO 37
IF(NCONT.LT.1500) GOTO 337
FOLD=0.5*FOLD
337 NCONT=0
37 IF(FDIFER.LT.CONVER) GOTO 81
IF(LHIGH.EQ.1) GOTO 43
FS=F(1)
LSEC=1
GO TO 44
43 FS=F(2)
LSEC=2
44 DO 18 I=1,N1
IF(LHIGH.EQ.I)GO TO 18
IF(F(I).LT.FS) GOTO 18
FS=F(I)
LSEC=I
18 CONTINUE
DO 61 J=1,NX
X2(N3,J)=X2(N2,J)+ALPHA*(X2(N2,J)-X2(LHIGH,J))
T(J)=X2(N3,J)
CALL SURM(T,S(N3),SR(N3),NX,NC,NIC,K8)
89 IF(SR(N3).LT.FDIFER) GOTO 82
INF=N3
IR=4
GOTO 500
82 CALL OBF(T,F(N3))
IF(F(N3).LT.F(LOW)) GOTO 84
IF(F(N3).LT.F(LSEC)) GOTO 92
GOTO 60
92 DO 93 J=1,NX
93 X2(LHIGH,J)=X2(N3,J)
SR(LHIGH)=SR(N3)
F(LHIGH)=F(N3)
GOTO 900
84 DO 23 J=1,NX
X2(N4,J)=X2(N3,J)+GAMA*(X2(N3,J)-X2(N2,J))
T(J)=X2(N4,J)
CALL SURM(T,S(N4),SR(N4),NX,NC,NIC,K8)
IF(SR(N4).LT.FDIFER) GOTO 25
NF=N4
IR=5
GOTO 500
25 CALL OBF(T,F(N4))
IF (F(LOW) .LT. F(N4)) GOTO 92
DO 26 J = 1, NX
26 X2 (LHIGH, J) = X2(N4, J)
    F(LHIGH) = F(N4)
    SR(LHIGH) = SR(N4)
    GOTO 900
IF (F(N3) .GT. F(LHIGH)) GOTO 64
DO 65 J = 1, NX
65 X2(LHIGH, J) = X2(N3, J)
64 DO 66 J = 1, NX
66 X2(N4, J) = BETA * X2(LHIGH, J) + (1. - BETA) * X2(N2, J)
66 T(J) = X2(N4, J)
CALL SURM(T, S(N4), SR(N4), NX, NC, NIC, K8)
IF (SR(N4) .LT. FDIFER) GOTO 67
IR = 6
INF = N4
GOTO 500
CALL OBF(T, F(N4))
IF (F(LHIGH) .GT. F(N4)) GOTO 68
DO 69 J = 1, NX
DO 69 I = 1, N1
69 X2(I, J) = 0.5 * (X2(I, J) + X2(LOW, J))
I = 1
70 DO 71 J = 1, NX
71 T(J) = X2(I, J)
CALL SURM(T, S(I), SR(I), NX, NC, NIC, K8)
IF (SR(I) .LT. FDIFER) GOTO 72
IR = 7
INF = I
GOTO 500
CALL OBF(T, F(I))
IF (I .EQ. N1) GOTO 900
I = I + 1
GOTO 70
DO 73 J = 1, NX
73 X2(LHIGH, J) = X2(N4, J)
    SR(LHIGH) = SR(N4)
    F(LHIGH) = F(N4)
    GOTO 900
STEP = 0.05 * FDIFER
ICONTR = 0
ICHEK = 0
STEP1 = STEP / (XNX * SQRT(2.)) * (SQRT(XNX + 1.) + XNX - 1.)
STEP2 = STEP / (XNX * SQRT(2.)) * (SQRT(XNX + 1.) - 1.)
DO 601 J = 1, NX
138

601 X1(1,J)=T(J)
DO 602 I1=2,K1
DO 604 J=1,NX
604 X1(I1,J)=STEP2+T(J)
L=I1-1
X1(I1,L)=STEP1+T(L)
602 CONTINUE
DO 503 I1=1,K1
DO 504 J=1,NX
504 T(J)=X1(I1,J)
CALL SURM(T,S(I1),S0,NX,NC,NIC,K8)
503 CONTINUE
528 SH=S(1)
INDEX=1
DO 507 I1=2,K1
IF(S(I1).LE.SH) GOTO 507
SH=S(I1)
INDEX=I1
507 CONTINUE
SL=S(1)
KOUNT=1
DO 508 I1=2,K1
IF(SL.LE.S(I1)) GOTO 508
SL=S(I1)
KOUNT=I1
508 CONTINUE
DO 509 J=1,NX
S2=0.
DO 510 I1=1,K1
510 S2=S2+X1(I1,J)
X1(K2,J)=1./XNX*(S2-X1(INDEX,J))
X1(K3,J)=2.*X1(K2,J)-X1(INDEX,J)
509 T(J)=X1(K3,J)
CALL SURM(T,S(K3),S0,NX,NC,NIC,K8)
IF(S(K3).LT.SL) GOTO 511
IF(INDEX.EQ.1) GOTO 538
SS=S(1)
GOTO 539
538 SS=S(2)
539 DO 512 I1=1,K1
IF((INDEX-I1).EQ.0) GOTO 512
IF(S(I1).LE(SS) GOTO 512
SS=S(I1)
512 CONTINUE
IF(S(K3).GT.SS) GOTO 513
GOTO 514
511 DO 515 J=1,NX
   X1(K4,J)=X1(K2,J)+2.*(X1(K3,J)-X1(K2,J))
   T(J)=X1(K4,J)
   CALL SURM(T,S(K4),S0,NX,NC,NIC,K8)
   IF(S(K4).LT.SL) GOTO 516
   GOTO 514
513 IF(S(K3).GT.SH) GOTO 517
   DO 518 J=1,NX
   X1(INDEX,J)=X1(K3,J)
517 DO 519 J=1,NX
   X1(K4,J)=0.5*(X1(INDEX,J)+X1(K2,J))
   T(J)=X1(K4,J)
   CALL SURM(T,S(K4),S0,NX,NC,NIC,K8)
   IF(SH.GT.S(K4)) GOTO 506
   DO 520 J=1,NX
   DO 520 I1=1,K1
   X1(I1,J)=0.5*(X1(I1,J)+X1(KOUNT,J))
   DO 529 I1=1,K1
   DO 530 J=1,NX
520 T(J)=X1(I1,J)
   CALL SURM(T,S(I1),S0,NX,NC,NIC,K8)
529 CONTINUE
505 SL=S(1)
   KOUNT=1
   DO 523 I1=2,K1
   IF(SL.LT.S(I1)) GOTO 523
   SL=S(I1)
   KOUNT=I1
523 CONTINUE
   SR(INF)=SQRT(S(KOUNT))
   DO 527 J=1,NX
527 T(J)=X1(KOUNT,J)
   GOTO 526
506 DO 531 J=1,NX
531 X1(INDEX,J)=X1(K4,J)
   S(INDEX)=S(K4)
   GOTO 505
516 DO 521 J=1,NX
   X1(INDEX,J)=X1(K4,J)
521 T(J)=X1(INDEX,J)
   S(INDEX)=S(K4)
   SR(INF)=SQRT(S(K4))
   GOTO 526
514 DO 522 J=1,NX
X1(INDEX,J)=X1(K3,J)

T(J)=X1(INDEX,J)
S(INDEX)=S(K3)
SR(INF)=SQRT(S(K3))

ICONTR=ICONTR+1
DO 536 J=1,NX

X2(INF,J)=T(J)
IF(ICONTR.LT.2*K1) GOTO 550
ICONTR=0
DO 524 J=1,NX

T(J)=X1(K2,J)
CALL SURM(T,S(K2),S0,NX,NC,NIC,K8)

DIFER=0.
DO 557 I1=1,K1

DIFER=DIFER+(S(I1)-S(K2))**2
DIFER=1./(X7*XNX)*SQRT(DIFER)
IF(DIFER.GT.1D-14)GOTO 550

STEP=2D1*FDIFER
CALL SURM(T,S(K1),SR(INF),NX,NC,NIC,K8)

DO 552 J=1,NX

X1(K1,J)=T(J)
DO 553 J=1,NX
FACTOR=1.0
T(J)=X1(K1,J)+FACTOR*STEP
X1(L9,J)=T(J)
CALL SURM(T,S(L9),S0,NX,NC,NIC,K8)
T(J)=X1(K1,J)-FACTOR*STEP
X1(L5,J)=T(J)
CALL SURM(T,S(L5),S0,NX,NC,NIC,K8)

IF(S(L9).LT.S(K1)) GOTO 554
IF(S(L5).LT.S(K1)) GOTO 555
GOTO 597

IF(S(L9).LT.S(K1)) GOTO 554
IF(S(L5).LT.S(K1)) GOTO 555
GOTO 597

X1(L5,J)=X1(K1,J)
S(L5)=S(K1)
X1(K1,J)=X1(L9,J)
S(K1)=S(L9)
FACTOR=FACTOR+1.
T(J)=X1(K1,J)+FACTOR*STEP
CALL SURM(T,S(L9),S0,NX,NC,NIC,K8)
GOTO 556

X1(L9,J)=X1(K1,J)
S(L9)=S(K1)
X1(K1,J)=X1(L5,J)
S(K1)=S(L5)
FACTOR=FACTOR+1.
T(J)=X1(K1,J)-FACTOR*STEP
CALL SURM(T,S(L5),S0,NX,NC,NIC,K8)
GOTO 556

597    H(J)=X1(L9,J)-X1(L5,J)
X1(L6,J)=X1(L5,J)+H(J)*R1A
T(J)=X1(L6,J)
CALL SURM(T,S(L6),S0,NX,NC,NIC,K8)
X1(L7,J)=X1(L5,J)+H(J)*R2A
T(J)=X1(L7,J)
CALL SURM(T,S(L7),S0,NX,NC,NIC,K8)
IF(S(L6).GT.S(L7)) GOTO 568
X1(L8,J)=X1(L5,J)+(1.-R3A)*H(J)
X1(L5,J)=X1(L7,J)
T(J)=X1(L8,J)
CALL SURM(T,S(L8),S0,NX,NC,NIC,K8)
IF(S(L8).GT.S(L6)) GOTO 576
X1(L5,J)=X1(L6,J)
S(L5)=S(L6)
GOTO 575

576    X1(L9,J)=X1(L8,J)
S(L9)=S(L8)
GOTO 575

568    X1(L9,J)=X1(L6,J)
X1(L8,J)=X1(L5,J)+R3A*H(J)
T(J)=X1(L8,J)
CALL SURM(T,S(L8),S0,NX,NC,NIC,K8)
S(L9)=S(L6)
IF(S(L7).GT.S(L8)) GOTO 571
X1(L5,J)=X1(L8,J)
S(L5)=S(L8)
GOTO 575

571    X1(L9,J)=X1(L7,J)
S(L9)=S(L7)

575    IF(ABS(X1(L9,J)-X1(L5,J)).GT.1D-2*FDIFER) GOTO 597
X1(K1,J)=X1(L7,J)
T(J)=X1(L7,J)
S(K1)=S(L5)
SR(INF)=SQRT(S(K1))
IF(SR(INF).LT.FDIFER) GOTO 760
CONTINUE
ICHEK=ICHEK+1
STEP=FDIFER
IF(ICHEK.LE.2) GOTO 525
IER=-1
GOTO 185
DO 761 J=1,NX
X2(INF,J)=X1(K1,J)
T(J)=X1(K1,J)
550 IF(SR(INF).GT.FDIFER) GOTO 528
IF(SR(INF).GT.0.) GOTO 535
CALL OBF(T,FINT)
DO 639 J=1,NX
T(J)=X2(INF,J)
CALL CONSTR(T,R)
DO 540 J=K7,K8
R1(J)=R(J)
DO 541 J=1,NX
H(J)=X1(KOUNT,J)-X2(INF,J)
543 T(J)=X2(INF,J)+0.5*H(J)
CALL CONSTR(T,R)
FLG(1)=0.
FLG(2)=0.
FLG(3)=0.
DO 544 J=K7,K8
IF(R3(J).GE.0.) GOTO 544
FLG(1)=FLG(1)+R1(J)*R1(J)
FLG(2)=FLG(2)+R(J)*R(J)
FLG(3)=FLG(3)+R3(J)*R3(J)
CONTINUE
SR(INF)=SQRT(FLG(1))
IF(SR(INF).LT.FDIFER) GOTO 535
ALFA1=FLG(1)-2.*FLG(2)+FLG(3)
BETA1=3.*FLG(1)-4.*FLG(2)+FLG(3)
RATIO=BETA1/(4.*ALFA1)
DO 545 J=1,NX
T(J)=X2(INF,J)+H(J)*RATIO
CALL SURM(T,S0,SR(INF),NX,NC,NIC,K8)
IF(SR(INF).LT.FDIFER) GOTO 665
DO 549 I1=1,20
DO 548 J=1,NX
T(J)=T(J)-0.05*H(J)
CALL SURM(T,S0,SR(INF),NX,NC,NIC,K8)
IF(SR(INF).LT.FDIFER) GOTO 665
CONTINUE
665 CALL OBF(T,F0)
IF(FINT.GT.F0) GOTO 546
SR(INF)=0.
GOTO 535

546 DO 547 J=1,NX
547 X2(INF,J)=T(J)
535 CONTINUE
IER=1
DO 635 J=1,NX
635 T(J)=X2(INF,J)
GOTO(201,8,203,82,25,67,72),IR
80 IE=-1
GOTO 183
81 IE=1
183 DO 184 J=1,NX
184 T(J)=X2(LOW,J)
FX=F(LOW)
185 RETURN
END

C
SUBROUTINE SURM(T,SM,SM1,NX,NC,NIC,K8)
IMPLICIT DOUBLE PRECISION(A-H,O-Z)
DIMENSION T(NX),R(40)
K7=NC+1
SM=0.0
CALL CONSTR(T,R)
IF(NIC.EQ.0) GOTO 4
DO 1 J=K7,K8
IF(R(J).GE.0) GOTO 1
SM=SM+R(J)*R(J)
1 CONTINUE
4 IF(NC.EQ.0) GOTO 3
DO 2 J=1,NC
2 SM=SM+R(J)*R(J)
3 SM1=SQRT(SM)
RETURN
END

Subroutine CONSTR(T,R)
DOUBLE PRECISION T,R
dimension R(40),T(10)
R(1)=T(1)
R(2)=T(2)
R(3)=T(3)
R(4)=T(4)
R(5)=T(5)
subroutine OBF(T,O)
implicit double precision(k,m)
implicit double precision (A-H,O-Z)
PARAMETER (NDGL=4,NGRID=33,LWORK=8*NDGL+21+NGRID,
*WORK=20)
PARAMETER(NRDENS=1,LRCONT=5000,LRCONTI=10000000,
*LICONT=NRDENS+1)
DIMENSION Y(NDGL),WORK(LWORK),IWORK(LIWORK),T(10)

COMMON /CORER/RCONT(LRCONT)
COMMON/CORED/RCONTI(LRCONTI)
COMMON /COREI/ICONT(LICONT)
EXTERNAL FCN,SOLOUT
common/par/av,kv,z,tm,kd,ke,k1,k2,k3
av=7.5D-1
kv=1.0D-1
kd=0.166D0
tm=8.0D0
k1=T(1)*3.0D-3
k2=T(2)*5.5D-3
k3=T(3)*2.0D-4
ke=T(4)*2.5D0
z=T(5)*7.0D2

C ---DIMENSION OF THE SYSTEM
N=NDGL
C ---OUTPUT ROUTINE IS USED DURING INTEGRATION
IOUT=1
C ---INITIAL VALUES AND ENDPOINT OF INTEGRATION
RPAR=0.0D0
X=0.0D0
Y(1)=1.58D0
Y(2)=0.81D0
Y(3)=0.0D0
Y(4)=0.0D0
XEND=33.D0
C ---REQUIRED (RELATIVE AND ABSOLUTE) TOLERANCE
ITOL=0
RTOL=1.0D-5
ATOL=RTOL
C ---DEFAULT VALUES FOR PARAMETERS
DO 10 I=1,20
   IWORK(I)=0
10 WORK(I)=0.D0
C ---SECOND COMPONENT USES RETARDED ARGUMENT
   IWORK(5)=NRDENS
   ICONT(2)=2
C ---USE AS GRID-POINTS
   IWORK(6)=NGRID
   DO 12 I=1,NGRID
12 WORK(20+I)=I
   open(2, file='data4.txt')

C ---CALL OF THE SUBROUTINE RETARD
   CALL RETARD(N,FCN,X,Y,XEND,
               & RTOL,ATOL,ITOL,
               & SOLOUT,IOUT,
               & WORK,LWORK,IWORK,LRCOUT,LICONT,
               & RPAR,IPAR,IDID,O,T)

   rewind(2)
   print*,O
   Return
end

Subroutine SOLOUT(NR,X,Y,resi,T)
implicit double precision (A-H,k,O-Z)
dimension Y(10),a(10),yobs(10),
* T(10),XN(0:300)
common/par/av,kv,z,tm,kd,ke,k1,k2,k3
EXTERNAL PHI
a(1)=1.5D0
a(2)=860D0
a(3)=10.D0
a(4)=8.0D0
if(XOUT.EQ.33.0D0) Then
   XOUT=0.0
   print*,av,kv,z,kd,ke,k1,k2,k3
endif
grid=3.0D0+XOUT
If(X.GT.grid-0.01D0.and.X.LT.grid+0.01D0) Then
   uv1=av*YDELAY(1,X-0.5D0*tm,RPAR,IPAR)/(kv+YDELAY
               *(1,X-0.5D0*tm,RPAR,IPAR))
   if(uv1.GE.0.1D0)then
      kul=k1-k2*uv1
   else
ku1 = k3 + ((k1-K3)/0.1D0-k2)*uv1
endif

uv2 = av*YDELAY(1, X-0.953D0*tm, RPAR, IPAR)/(kv+YDELAY(1, X-0.953D0*tm, RPAR, IPAR))
if(uv2.GE.0.1D0)
    ku2 = k1-k2*uv2
else
    ku2 = k3 + ((k1-k3)/0.1D0-k2)*uv2
endif

uv3 = av*YDELAY(1, X-0.047D0*tm, RPAR, IPAR)/(kv+YDELAY(1, X-0.047D0*tm, RPAR, IPAR))
if(uv3.GE.0.1D0)
    ku3 = k1-k2*uv3
else
    ku3 = k3 + ((k1-k3)/0.1D0-k2)*uv3
endif

uv4 = av*YDELAY(1, X-0.769D0*tm, RPAR, IPAR)/(kv+YDELAY(1, X-0.769D0*tm, RPAR, IPAR))
if(uv4.GE.0.1D0)
    ku4 = k1-k2*uv4
else
    ku4 = k3 + ((k1-k3)/0.1D0-k2)*uv4
endif

uv5 = av*YDELAY(1, X-0.231D0*tm, RPAR, IPAR)/(kv+YDELAY(1, X-0.231D0*tm, RPAR, IPAR))
if(uv5.GE.0.1D0)
    ku5 = k1-k2*uv5
else
    ku5 = k3 + ((k1-k3)/0.1D0-k2)*uv5
endif

if(ku1.LT.0.D0)
    ku1 = 0.D0
endif

if(ku2.LT.0.D0)
    ku2 = 0.D0
endif
if(ku3.LT.0.D0)then
  ku3=0.D0
endif

if(ku4.LT.0.D0)then
  ku4=0.D0
endif

if(ku5.LT.0.D0)then
  ku5=0.D0
endif

Y(5)=Y(2)+0.569D0*YLAG(2,X-*0.5D0*tm,PHI,RPAR,IPAR)*ku1+0.237D0*YLAG(2,X-0.953D0*tm,PHI,RPAR,IPAR)*ku2+0.237D0*YLAG(2,X-0.047D0*tm,PHI,RPAR,IPAR)*ku3+0.479D0*(YLAG(2,X-0.769D0*tm,PHI,RPAR,IPAR)*ku4
+YLAG(2,X-0.231D0*tm,PHI,RPAR,IPAR)*ku5)

read(2,*)yobs(1),yobs(2),yobs(3),yobs(4)
resi=((yobs(1)-Y(1))/a(1))**2.D0+((yobs(2)-Y(5))/a(2))**2.0D0+((yobs(3)-Y(3))/a(3))**2.0D0+((yobs(4)-*4))/a(4))
***2.0D0
XOUT=XOUT+3.0D0
print*,Y(1),Y(5),Y(3),Y(4)
else
  resi=0.0D0
End if
Return
end

SUBROUTINE FCN(N,X,Y,F,RPAR,IPAR)
IMPLICIT REAL*8 (A-H,k,O-Z)
DIMENSION Y(N),F(N),S(N)
EXTERNAL PHI
common/par/av,kv,z,tm,kd,ke,k1,k2,k3

Y1L=YDELAY(1,X-tm,RPAR,IPAR)
Y2L=YLAG(2,X-tm,PHI,RPAR,IPAR)
If(Y(1).LT.0.001D0.and.Y(2).GT.z)Then
  Y(1)=0
  z=Y(2)
  print*, X, Y(2)
endif

F(1)=-av*Y(1)*Y(2)/((kv+Y(1))*z)
uv=av*Y(1)/(kv+Y(1))

If(uv.GE.0.1D0)then
  ku=k1-k2*uv
else
  ku=k3+((k1-k3)/0.1D0-k2)*uv
endif
if(ku.LT.0.D0)then
  ku=0.D0
endif
üvl=av*Y1L/(kv+Y1L)
if(uvl.GE.0.1D0)then
  kula=k1-k2*uvl
else
  kula=k3+((k1-k3)/0.1D0-k2)*uvl
endif
if(kula.LT.0.D0)then
  kula=0.D0
endif

F(3)=kula*Y2L

uv1=av*YDELAY(1,X-0.5D0*tm,RPAR,IPAR)/(kv+YDELAY*(1,X-0.5D0*tm,RPAR,IPAR))
if(uv1.GE.0.1D0)then
  ku1=k1-k2*uv1
else
  ku1=k3+((k1-k3)/0.1D0-k2)*uv1
endif

uv2=av*YDELAY(1,X-0.953D0*tm,RPAR,IPAR)/(kv+YDELAY*AY(1,X-0.953D0*tm,RPAR,IPAR))
if(uv2.GE.0.1D0)then
  ku2=k1-k2*uv2
else
ku2=k3+((k1-k3)/0.1D0-k2)*uv2
endif
uv3=av*YDELAY(1,X-0.047D0*tm,RPAR,IPAR)/(kv+YDELAY *
(1,X-0.047D0*tm,RPAR,IPAR))
if(uv3.GE.0.1D0)then
ku3=k1-k2*uv3
else
ku3=k3+((k1-k3)/0.1D0-k2)*uv3
endif
uv4=av*YDELAY(1,X-0.769D0*tm,RPAR,IPAR)/(kv+YDELAY *
(1,X-0.769D0*tm,RPAR,IPAR))
if(uv4.GE.0.1D0)then
ku4=k1-k2*uv4
else
ku4=k3+((k1-k3)/0.1D0-k2)*uv4
endif
uv5=av*YDELAY(1,X-0.231D0*tm,RPAR,IPAR)/(kv+YDELAY *
(1,X-0.231D0*tm,RPAR,IPAR))
if(uv5.GE.0.1D0)then
ku5=k1-k2*uv5
else
ku5=k3+((k1-k3)/0.1D0-k2)*uv5
endif

if(ku1.LT.0.D0)then
ku1=0.D0
endif

if(ku2.LT.0.D0)then
ku2=0.D0
endif

if(ku3.LT.0.D0)then
ku3=0.D0
endif

if(ku4.LT.0.D0)then
ku4=0.D0
endif
if(ku5.LT.0.D0) then
   ku5=0.D0
endif

Xs=0.569D0*YLAG(2,X-
*0.5D0*2.0D0,PHI,RPAR,IPAR)*ku1+0.237D0*YLAG(2,X-
*0.953D0*2.0D0,PHI,RPAR,IPAR)
*ku2+0.237D0*YLAG(2,X-
*0.047D0*2.0D0,PHI,RPAR)
*ku3+0.479D0*(YLAG(2,X-
*0.769D0*2.0D0,PHI,RPAR,IPAR)*ku4+YLAG(2,X-
*0.231D0*2.0D0,PHI,RPAR,IPAR)*ku5)

F(4)=ke*Xs
If(Y(1).GT.0.001D0)Then
   F(2)=(av*Y(1)/(kv+Y(1))-ku)*Y(2)
else
   F(2)=(-kd-ku)*Y(2)
endif

RETURN
END

FUNCTION PHI(I,T,RPAR,IPAR)
IMPLICIT REAL*8 (A-H,O-Z)
IF (I.EQ.2) PHI=RPAR
RETURN
END

SUBROUTINE RETARD(N,FCN,X,Y,XEND,
& RTOL,ATOL,ITOL,
& SOLOUT,IOUT,
& WORK,LWORK,IWORK,LIWORK,LRCONT,
& LICONT,RPAR,IPAR,IDID,O,T)
C ---------------------------------------------------------
C    NUMERICAL SOLUTION OF A SYSTEM OF FIRST ORDER DELAY
C    ORDINARY DIFFERENTIAL EQUATIONS  Y'(X)=F(X,Y(X),Y(X-
C    A),...).
C    THIS CODE IS BASED ON AN EXPLICIT RUNGE-KUTTA METHOD
C    OF
C    ORDER (4)5 DUE TO DORMAND & PRINCE (WITH STEPSIZE
C    CONTROL AND DENSE OUTPUT).
AUTHORS: E. HAIRER AND G. WANNER
UNIVERSITE DE GENEVE, DEPT. DE MATHEMATIQUES
CH-1211 GENEVE 24, SWITZERLAND
E-MAIL: Ernst.Hairer@math.unige.ch
Gerhard.Wanner@math.unige.ch

THIS CODE IS DESCRIBED IN SECTION II.17 OF THE BOOK:
E. HAIRER, S.P. NORSETT AND G. WANNER, SOLVING
ORDINARY DIFFERENTIAL EQUATIONS I. NONSTIFF PROBLEMS.
2ND EDITION.
SPRINGER SERIES IN COMPUTATIONAL MATHEMATICS,
SPRINGER-VERLAG (1993)

VERSION OF SEPTEMBER 30, 1995

INPUT PARAMETERS
--------------
N         DIMENSION OF THE SYSTEM
FCN       NAME (EXTERNAL) OF SUBROUTINE COMPUTING
          THE RIGHT-
          HAND-SIDE OF THE DELAY EQUATION, E.G.,
          SUBROUTINE FCN(N,X,Y,F,RPAR,IPAR)
          DOUBLE PRECISION X,Y(N),F(N)
          EXTERNAL PHI
          F(1)=(1.4D0-YLAG(1,X-
          1.0D0,PHI,RPAR,IPAR))*Y(1)
          F(2)=...   ETC.
FOR AN EXPLICATION OF YLAG SEE BELOW.
DO NOT USE YLAG(I,X-0.0D0,PHI,RPAR,IPAR) !
THE INITIAL FUNCTION HAS TO BE SUPPLIED
BY:
FUNCTION PHI(I,X,RPAR,IPAR)
DOUBLE PRECISION PHI,X
WHERE I IS THE COMPONENT AND X THE
ARGUMENT
X         INITIAL X-VALUE
Y(N)      INITIAL VALUES FOR Y (MAY BE DIFFERENT
FROM PHI(I,X),
IN THIS CASE IT IS HIGHLY RECOMMENDED TO
SET IWORK(6)
AND WORK(21),..., SEE BELOW)
XEND        FINAL X-VALUE (XEND > X)

RTOL,ATOL   RELATIVE AND ABSOLUTE ERROR TOLERANCES.
            THEY CAN BE BOTH SCALARS OR ELSE BOTH
            VECTORS OF LENGTH N.

ITOL        SWITCH FOR RTOL AND ATOL:
            ITOL=0: BOTH RTOL AND ATOL ARE SCALARS.
            THE CODE KEEPS, ROUGHLY, THE LOCAL ERROR
            OF
            Y(I) BELOW RTOL*ABS(Y(I))+ATOL
            ITOL=1: BOTH RTOL AND ATOL ARE VECTORS.
            THE CODE KEEPS THE LOCAL ERROR OF Y(I)
            BELOW
            RTOL(I)*ABS(Y(I))+ATOL(I).

SOLOUT      NAME (EXTERNAL) OF SUBROUTINE PROVIDING
            THE
            NUMERICAL SOLUTION DURING INTEGRATION.
            IF IOUT.GE.1, IT IS CALLED AFTER EVERY
            SUCCESSFUL STEP.
            SUPPLY A DUMMY SUBROUTINE IF IOUT=0.
            IT MUST HAVE THE FORM
            SUBROUTINE SOLOUT
            (NR,XOLD,X,Y,N,RPAR,IPAR,IRTRN)
            DOUBLE PRECISION X,XOLD,Y(N)
            ....
            SOLOUT FURNISHES THE SOLUTION "Y" AT THE
            NR-TH
            GRID-POINT "X" (THEREBY THE INITIAL VALUE
            IS
            THE FIRST GRID-POINT).
            "XOLD" IS THE PRECEEDING GRID-POINT.
            "IRTRN" SERVES TO INTERRUPT THE
            INTEGRATION. IF IRTRN
            IS SET <0, RETARD WILL RETURN TO THE
            CALLING PROGRAM.

---- CONTINUOUS OUTPUT: -----
            DURING CALLS TO "SOLOUT" AS WELL AS TO
            "FCN", A
            CONTINUOUS SOLUTION IS AVAILABLE THROUGH
            THE FUNCTION
            YLAG(I,S,PHI,RPAR,IPAR)
WHICH PROVIDES AN APPROXIMATION TO THE I-TH COMPONENT OF THE SOLUTION AT THE POINT $S$.

The value $S$ has to lie in an interval where the numerical solution is already computed. It depends on the size of $LRCONT$ (see below) how far back the solution is available.

**IOUT**

Switch for calling the subroutine `SOLOUT`:
- IOUT=0: Subroutine is never called
- IOUT=1: Subroutine is used for output.

**WORK**

Array of working space of length "LWORK". `WORK(1), ..., WORK(20)` serve as parameters for the code.

For standard use, set them to zero before calling.

"LWORK" must be at least $8 \times N + 21 + NGRID$ where $NGRID = IWORK(6)$

**LWORK**

Declared length of array "WORK".

**IWORK**

Integer working space of length "LIWORK". `IWORK(1), ..., IWORK(20)` serve as parameters for the code.

For standard use, set them to zero before calling.

"LIWORK" must be at least 20.

**LIWORK**

Declared length of array "IWORK".

**LRCONT**

Declared length of common block

$$\text{COMMON} / \text{CORER/CONT(LRCONT)}$$

Which must be declared in the calling program.

"LRCONT" must be sufficiently large. If the dense output of `MXST` back steps has to be stored, it must be at least

$$MXST \times (5 \times NRDENS + 2)$$

Where $NRDENS = IWORK(5)$ (see below).
C LICONT DECLARED LENGTH OF COMMON BLOCK
     >>> COMMON /COREI/ICONT(LICONT) <<<
     WHICH MUST BE DECLARED IN THE CALLING
     PROGRAM.
     "LICONT" MUST BE AT LEAST
     NRDENS + 1
     THESE COMMON BLOCKS ARE USED FOR STORING
     THE COEFFICIENTS
     OF THE CONTINUOUS SOLUTION AND MAKES THE
     CALLING LIST FOR
     THE FUNCTION "CONTD5" AS SIMPLE AS
     POSSIBLE.

RPAR, IPAR REAL AND INTEGER PARAMETERS (OR PARAMETER
ARRAYS) WHICH
     CAN BE USED FOR COMMUNICATION BETWEEN
     YOUR CALLING
     PROGRAM AND THE FCN, JAC, MAS, SOLOUT
     SUBROUTINES.

C----------------------------------------------------------
-------------
C
C SOPHISTICATED SETTING OF PARAMETERS
-----------------

SEVERAL PARAMETERS (WORK(1),...,IWORK(1),...) ALLOW
TO ADAPT THE CODE TO THE PROBLEM AND TO THE
NEEDS OF
THE USER. FOR ZERO INPUT, THE CODE CHOOSES
DEFAULT VALUES.

WORK(1) UROUND, THE ROUNDING UNIT, DEFAULT 2.3D-16.
WORK(2) THE SAFETY FACTOR IN STEP SIZE PREDICTION,
        DEFAULT 0.9D0.
WORK(3), WORK(4) PARAMETERS FOR STEP SIZE SELECTION
     THE NEW STEP SIZE IS CHOSEN SUBJECT TO THE
     RESTRICTION
     WORK(3) <= HNEW/HOLD <= WORK(4)
     DEFAULT VALUES: WORK(3)=0.2D0, WORK(4)=10.D0
WORK(5) IS THE "BETA" FOR STABILIZED STEP SIZE
CONTROL (SEE SECTION IV.2). LARGER VALUES OF BETA (<=0.1) MAKE THE STEP SIZE CONTROL MORE STABLE. NEGATIVE WORK(5) PROVOKE BETA=0. DEFAULT (FOR WORK(5)=0.D0) IS WORK(5)=0.04D0.

WORK(6) MAXIMAL STEP SIZE, DEFAULT XEND-X.

WORK(7) INITIAL STEP SIZE, FOR WORK(7)=0.D0 AN INITIAL GUESS IS COMPUTED WITH HELP OF THE FUNCTION HINIT

WORK(21),...,WORK(20+NGRID) PRESCRIBED POINTS, WHICH THE INTEGRATION METHOD HAS TO TAKE AS GRID-POINTS
X < WORK(21) < WORK(22) < ... < WORK(20+NGRID) <= XEND

IWORK(1) THIS IS THE MAXIMAL NUMBER OF ALLOWED STEPS. THE DEFAULT VALUE (FOR IWORK(1)=0) IS 100000.

IWORK(2) SWITCH FOR THE CHOICE OF THE COEFFICIENTS IF IWORK(2).EQ.1 METHOD OF DORMAND AND PRINCE (TABLE 5.2 OF SECTION II.5). AT THE MOMENT THIS IS THE ONLY POSSIBLE CHOICE. THE DEFAULT VALUE (FOR IWORK(2)=0) IS IWORK(2)=1.

IWORK(3) SWITCH FOR PRINTING ERROR MESSAGES IF IWORK(3).LT.0 NO MESSAGES ARE BEING PRINTED IF IWORK(3).GT.0 MESSAGES ARE PRINTED WITH WRITE (IWORK(3),*) ... DEFAULT VALUE (FOR IWORK(3)=0) IS IWORK(3)=6

IWORK(4) TEST FOR STIFFNESS IS ACTIVATED AFTER STEP NUMBER J*IWORK(4) (J INTEGER), PROVIDED IWORK(4).GT.0.
FOR NEGATIVE IWORK(4) THE STIFFNESS TEST IS NEVER ACTIVATED; DEFAULT VALUE IS
IWORK(4)=1000

IWORK(5) = NRDENS = NUMBER OF COMPONENTS, FOR WHICH
DENSE OUTPUT
IS REQUIRED (EITHER BY "SOLOUT" OR BY "FCN");
DEFAULT VALUE (FOR IWORK(5)=0) IS IWORK(5)=N;
FOR 0 < NRDENS < N THE COMPONENTS (FOR
WHICH DENSE
OUTPUT IS REQUIRED) HAVE TO BE SPECIFIED IN
ICONT(2),...,ICONT(NRDENS+1);
FOR NRDENS=N THIS IS DONE BY THE CODE.

IWORK(6) = NGRID = NUMBER OF PRESCRIBED POINTS IN THE
INTEGRATION INTERVAL WHICH HAVE TO BE GRID-
POINTS
IN THE INTEGRATION. USUALLY, AT THESE POINTS

THE SOLUTION OR ONE OF ITS DERIVATIVE HAS A
DISCONTINUITY.
DEFINE THESE POINTS IN
WORK(21),...,WORK(20+NGRID)
DEFAULT VALUE: IWORK(6)=0

------------------------------------------

OUTPUT PARAMETERS
------------------

X            X-VALUE FOR WHICH THE SOLUTION HAS BEEN
             COMPUTED
             (AFTER SUCCESSFUL RETURN X=XEND).

Y(N)         NUMERICAL SOLUTION AT X

H            PREDICTED STEP SIZE OF THE LAST ACCEPTED
             STEP

IDID         REPORTS ON SUCCESSFULNESS UPON RETURN:
             IDID= 1 COMPUTATION SUCCESSFUL,
             IDID= 2 COMPUT. SUCCESSFUL
             (INTERRUPTED BY SOLOUT)
             IDID=-1 INPUT IS NOT CONSISTENT,
             IDID=-2 LARGER NMAX IS NEEDED,
IDID=-3  STEP SIZE BECOMES TOO SMALL.
IDID=-4  PROBLEM IS PROBABLY STIFF (INTERRUPTED).
IDID=-5  COMPUT. INTERRUPTED BY YLAG

IWORK(17)  NFCN   NUMBER OF FUNCTION EVALUATIONS
IWORK(18)  NSTEP  NUMBER OF COMPUTED STEPS
IWORK(19)  NACCPT NUMBER OF ACCEPTED STEPS
IWORK(20)  NREJCT NUMBER OF REJECTED STEPS (DUE TO ERROR TEST),
           (STEP REJECTIONS IN THE FIRST STEP ARE NOT COUNTED)

----------------------------------------------------------

*** *** *** *** *** *** *** *** *** *** *** *** ***

DECLARATIONS

IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION Y(N),ATOL(*),RTOL(*),WORK(LWORK),
*IWORK(LIWORK)
DIMENSION RPAR(*),IPAR(*)
LOGICAL ARRET
EXTERNAL FCN,SOLOUT
COMMON /CORER/RCONT(10000000)
COMMON/CORED/RCONTI(10000000)
COMMON /COREI/NRDS,ICONT(1)
COMMON /POSITS/X0BEG,UROUND,HMAX,LAST,
*IPOS,IRTRN,IDIF,MXST,IPRINT

SETTING THE PARAMETERS

NFCN=0
NSTEP=0
NACCPT=0
NREJCT=0
ARRET=.FALSE.

-------- IPRINT FOR MONITORING THE PRINTING
IF(IWORK(3).EQ.0)THEN
  IPRINT=6
ELSE
  IPRINT=IWORK(3)
END IF

-------- NMAX , THE MAXIMAL NUMBER OF STEPS -----
IF(IWORK(1).EQ.0)THEN
  NMAX=100000
ELSE
   NMAX=IWORK(1)
   IF (NMAX.LE.0) THEN
      IF (IPRINT.GT.0) WRITE(IPRINT,*)
      ' WRONG INPUT IWORK(1)=', IWORK(1)
      ARRET=.TRUE.
   END IF
END IF
C -------- METH  COEFFICIENTS OF THE METHOD
IF (IWORK(2).EQ.0) THEN
   METH=1
ELSE
   METH=IWORK(2)
   IF (METH.LE.0.OR.METH.GE.2) THEN
      IF (IPRINT.GT.0) WRITE(IPRINT,*)
      ' CURIOUS INPUT IWORK(2)=', IWORK(2)
      ARRET=.TRUE.
   END IF
END IF
C -------- NSTIFF  PARAMETER FOR STIFFNESS DETECTION
NSTIFF=IWORK(4)
IF (NSTIFF.EQ.0) NSTIFF=1000
IF (NSTIFF.LT.0) NSTIFF=NMAX+10
C -------- NRDENS  NUMBER OF DENSE OUTPUT COMPONENTS
NRDENS=IWORK(5)
IF (NRDENS.LT.0.OR.NRDENS.GT.N) THEN
   IF (IPRINT.GT.0) WRITE(IPRINT,*)
   ' CURIOUS INPUT IWORK(5)=', IWORK(5)
   ARRET=.TRUE.
ELSE
   IF (NRDENS.EQ.0) NRDENS=N
C -------- CONTROL OF LENGTH OF COMMON BLOCK "CORER" ----- --
   IF (LRCONT.LT.(5*NRDENS+2)) THEN
      IF (IPRINT.GT.0) WRITE(IPRINT,*)
      ' INSUFFICIENT STORAGE FOR RCONT, MIN.
      LRCONT=', 5*NRDENS+2
      ARRET=.TRUE.
   END IF
C -------- CONTROL OF LENGTH OF COMMON BLOCK "COREI" ----- --
   IF (LICONT.LT.(NRDENS+1)) THEN
      IF (IPRINT.GT.0) WRITE(IPRINT,*)
      ' INSUFFICIENT STORAGE FOR ICONT, MIN.
      & LICONT=',NRDENS+1
ARRET=.TRUE.

ELSE
   NRDS=NRDENS
   IF (NRDENS.EQ.N) THEN
      DO 16 I=1,NRDENS
         ICONT(I)=I
   END IF
16
END IF
END IF
END IF

C -------- NGRID   NUMBER OF PRESCRIBED GRID-POINTS
NGRID=IWORK(6)
IF (NGRID.LT.0) NGRID=0

C -------- UROUND   SMALLEST NUMBER SATISFYING
C                   1.D0+UROUND>1.D0
IF(WORK(1).EQ.0.D0)THEN
   UROUND=2.3D-16
ELSE
   UROUND=WORK(1)
   IF(UROUND.LE.1.D-35.OR.UROUND.GE.1.D0)THEN
      IF (IPRINT.GT.0) WRITE(IPRINT,*)
         & WHICH MACHINE DO YOU HAVE? YOUR UROUND WAS:',WORK(1)
   END IF
END IF

C -------  SAFETY FACTOR -------------
IF(WORK(2).EQ.0.D0)THEN
   SAFE=0.9D0
ELSE
   SAFE=WORK(2)
   IF(SAFE.GE.1.D0.OR.SAFE.LE.1.D-4)THEN
      IF (IPRINT.GT.0) WRITE(IPRINT,*)
         & CURIOUS INPUT FOR SAFETY FACTOR
   END IF
END IF

C WORK(2)=',WORK(2)
ARRET=.TRUE.
END IF
END IF

C -------  FAC1,FAC2     PARAMETERS FOR STEP SIZE SELECTION
IF(WORK(3).EQ.0.D0)THEN
   FAC1=0.2D0
ELSE
   FAC1=WORK(3)
END IF
IF(WORK(4).EQ.0.D0)THEN
FAC2=10.D0
ELSE
   FAC2=WORK(4)
END IF
C --------- BETA FOR STEP CONTROL STABILIZATION -----------
IF(WORK(5).EQ.0.D0)THEN
   BETA=0.04D0
ELSE
   IF(WORK(5).LT.0.D0)THEN
      BETA=0.D0
   ELSE
      BETA=WORK(5)
      IF(BETA.GT.0.2D0)THEN
         IF (IPRINT.GT.0) WRITE(IPRINT,*)
         &' CURIOUS INPUT FOR BETA: WORK(5)=',WORK(5)
         ARRET=.TRUE.
      END IF
   END IF
END IF
C -------- MAXIMAL STEP SIZE
IF(WORK(6).EQ.0.D0)THEN
   HMAX=0.2D0
ELSE
   HMAX=WORK(6)
END IF
C -------- INITIAL STEP SIZE
H=WORK(7)
C -------- GRID WITH DISCONTINUITIES
XURO=100*UROUND*ABS(XEND)
IF (WORK(20+NGRID)-XEND.GE.XURO) THEN
   IF(IPRINT.GT.0) WRITE(IPRINT,*)
   &' WORK(20+NGRID) HAS TO BE <= XEND'
   ARRET=.TRUE.
END IF
IF (ABS(WORK(20+NGRID)-XEND).*XURO) NGRID=NGRID+1
WORK(20+NGRID)=XEND
C --------- PREPARE THE ENTRY-POINTS FOR THE ARRAYS IN WORK ----
IEGR=21
IEY1=IEGR+NGRID
IEK1=IEY1+N
IEK2=IEK1+N
IEK3=IEK2+N
IEK4=IEK3+N
IEK5=IEK4+N
IEK6 = IEK5 + N
IEYS = IEK6 + N
C ------ TOTAL STORAGE REQUIREMENT��仟------
ISTORE = IEYS + N - 1
IF (ISTORE .GT. LWORK) THEN
   IF (IPRINT .GT. 0) WRITE (IPRINT, *)
   & ' INSUFFICIENT STORAGE FOR WORK, MIN.
C LWORK = ', ISTORE
   ARRET = .TRUE.
END IF
ISTORE = 20
IF (ISTORE .GT. LIWORK) THEN
   IF (IPRINT .GT. 0) WRITE (IPRINT, *)
   & ' INSUFFICIENT STORAGE FOR IWORK, MIN.
C LIWORK = ', ISTORE
   ARRET = .TRUE.
END IF
C ------- WHEN A FAIL HAS OCCURED, WE RETURN WITH IDID = -1
IF (ARRET) THEN
   IDID = -1
   RETURN
END IF
C ------- CALL TO CORE INTEGRATOR 操仟------
IDIF = 5 * NRDENS + 2
MXST = LRCONT / IDIF
CALL RETCOR (N, FCN, X, Y, XEND, H, RTOL,
& ATOL, ITOL, SOLOUT, IOUT,
& IDID, NMAX, METH, NSTIFF, SAFE, BETA, FAC1, FAC2, NGRID,
& WORK (IEY1), WORK (IEK1), WORK (IEK2), WORK (IEK3),
& WORK (IEK4),
& WORK (IEK5), WORK (IEK6), WORK (IEYS), WORK (IEGR),
& RPAR, IPAR, NFCN, NSTEP, NACCPT, NREJCT, O, T)
WORK (7) = H
IWORK (17) = NFCN
IWORK (18) = NSTEP
IWORK (19) = NACCPT
IWORK (20) = NREJCT
C ----------- RETURN 操仟-----------
RETURN
END
SUBROUTINE RETCOR(N,FCN,X,Y,XEND,H,RTOL,
& ATOL,ITOL,SOLOUT,IOUT,
& IDID,NMAX,METH,NSTIFF,SAFE,BETA,FAC1,
& FAC2,NGRID,Y1,K1,K2,K3,
& K4,K5,K6,YSTI,GRID,RPAR,IPAR,NFCN,
& NSTEP,NACCPT,NREJCT,O,T)
C ---------------------------------------------------------
C     CORE INTEGRATOR FOR RETARD
C     PARAMETERS SAME AS IN RETARD WITH WORKSPACE ADDED
C ---------------------------------------------------------
C DECLARATIONS
C ---------------------------------------------------------
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DOUBLE PRECISION Y(N),Y1(N),K1(N),K2(N),K3(N),
&K4(N),K5(N),K6(N)
DIMENSION GRID(NGRID),YSTI(N)
DIMENSION ATOL(*),RTOL(*),RPAR(*),IPAR(*)
LOGICAL REJECT,LAST
EXTERNAL FCN
COMMON /CORER/CONT(1000)
COMMON/CORED/CONTI(1000)
COMMON /COREI/NRD,ICOMP(1)
COMMON /POSITS/X0BEG,UROUND,HMAX,IACT,
&IPOS,IRTRN,IDIF,MXST,IPRINT
C *** *** *** *** *** *** ***
C INITIALISATIONS
C *** *** *** *** *** *** ***
IF (METH.EQ.1) CALL
&CDOPRI(C2,C3,C4,C5,E1,E3,E4,E5,E6,E7,
&A21,A31,A32,A41,A42,A43,A51,A52,A53,A54,
&A61,A62,A63,A64,A65,A71,A73,A74,A75,A76,
&D1,D3,D4,D5,D6,D7)
FACOLD=1.D-4
EXPO1=0.2D0-BETA*0.75D0
FACC1=1.D0/FAC1
FACC2=1.D0/FAC2
POSNEG=SIGN(1.D0,XEND-X)
C --- INITIAL PREPARATIONS
IACT=1
IPOS=1
X0BEG=X
XEND=GRID(1)
IGRID=1
UROUND=10*UROUND
DO 3 I=0,MXST-1
CONTI(1+IDIF*I)=0.0D0
3 CONT(1+IDIF*I)=0.0D0

ATOLI=ATOL(1)
RTOLI=RTOL(1)
LAST=.FALSE.
HLAMB=0.D0
IASTI=0
HMAX=ABS(HMAX)
IRTRN=2
CALL FCN(N,X,Y,K1,RPAR,IPAR)
IRTRN=1
IORD=5

NFCN=NFCN+2
REJECT=.FALSE.
XOLD=X
IF (IRTRN.LT.0) GOTO 79
C --- BASIC INTEGRATION STEP
1 CONTINUE
H=0.05D0
If(X.GT.33.0D0)Then
Return
endif

C --- THE FIRST 6 STAGES
DO 22 I=1,N
22 Y1(I)=Y(I)+H*A21*K1(I)
CALL FCN(N,X+C2*H,Y1,K2,RPAR,IPAR)

DO 23 I=1,N
23 Y1(I)=Y(I)+H*(A31*K1(I)+A32*K2(I))
CALL FCN(N,X+C3*H,Y1,K3,RPAR,IPAR)

DO 24 I=1,N
24 Y1(I)=Y(I)+H*(A41*K1(I)+A42*K2(I)+A43*K3(I))
CALL FCN(N,X+C4*H,Y1,K4,RPAR,IPAR)

DO 25 I=1,N
CALL FCN(N,X+C5*H,Y1,K5,RPAR,IPAR)

DO 26 I=1,N
26 YSTI(I)=Y(I)+H*(A61*K1(I)+A62*K2(I)+A63*K3(I)+A64
    **K4(I)+A65*K5(I))
    XPH=X+H
    CALL FCN(N,XPH,YSTI,K6,RPAR,IPAR)

DO 27 I=1,N
27 Y1(I)=Y(I)+H*(A71*K1(I)+A73*K3(I)+A74*K4(I)+
    A75*K5(I)+A76*K6(I))
    IRTRN=1
    CALL FCN(N,XPH,Y1,K2,RPAR,IPAR)

C ------- PREPARE DENSE OUTPUT I
    NRDL=4*NDRD+IACT
    DO 400 J=1,NRD
      I=1
      &                   +D6*K6(I)+D7*K2(I))
  400 CONTINUE

C ------- COMPUTE DENSE OUTPUT I
    DO 430 J=1,NRD
      I=1
      YDIFF=Y1(I)-Y(I)
      BSPL=H*K1(I)-YDIFF
      CONTI(IACT+J)=Y(I)
      CONTI(IACT+NRD+J)=YDIFF
      CONTI(IACT+2*NRD+J)=BSPL
      CONTI(IACT+3*NRD+J)=-H*K2(I)+YDIFF-BSPL
  430 CONTINUE
    CONTI(IACT)=X
    IACT=IACT+IDIF
    CONTI(IACT-1)=H
    IF (IACT+IDIF-1.GT.MXST*IDIF) IACT=1

C -------

C ------- PREPARE DENSE OUTPUT II
    IACT=IACT-IDIF
NRDL=4*NRD+IACT
DO 40 J=1,NRD
I=ICOMP(J)
CONT(NRDL+J)=H*(D1*K1(I)+D3*K3(I)+
D4*K4(I)+D5*K5(I)+D6*K6(I)+D7*K2(I))
40 CONTINUE
C ------

C ------- COMPUTE DENSE OUTPUT II

DO 43 J=1,NRD
I=ICOMP(J)
YDIFF=Y1(I)-Y(I)
BSPL=H*K1(I)-YDIFF
CONT(IACT+J)=Y(I)
CONT(IACT+NRD+J)=YDIFF
CONT(IACT+2*NRD+J)=BSPL
CONT(IACT+3*NRD+J)=-H*K2(I)+YDIFF-BSPL
43 CONTINUE
CONT(IACT)=X
IACT=IACT+IDIF
CONT(IACT-1)=H
IF (IACT+IDIF-1.GT.MXST*IDIF) IACT=1
C ------

DO 44 I=1,N
K1(I)=K2(I)
44 Y(I)=Y1(I)
XOLD=X
X=XPH
IF (IRTRN.EQ.3) THEN
IRTRN=4
CALL FCN(N,X,Y,K1,RPAR,IPAR)
NFCN=NFCN+1
END IF
IF (IOUT.NE.0) THEN
CALL SOLOUT(NACCEPT+1,X,Y,resi,T)
O=O+resi*resi
IF (IRTRN.LT.0) GOTO 79
END IF
C ------- NORMAL EXIT
IF (IRTRN.LT.0) GOTO 75
GOTO 1
C --- FAIL EXIT
75 CONTINUE
   IDID=-5
   RETURN
76 CONTINUE
   IDID=-4
   RETURN
77 CONTINUE
   IF (IPRINT.GT.0) WRITE(IPRINT,979)X
   IF (IPRINT.GT.0) WRITE(IPRINT,*)' STEP SIZE TOO SMALL, H=',H
   IDID=-3
   RETURN
78 CONTINUE
   IF (IPRINT.GT.0) WRITE(IPRINT,979)X
   IF (IPRINT.GT.0) WRITE(IPRINT,*)&' MORE THAN NMAX =',NMAX,'STEPS ARE NEEDED'
   IDID=-2
   RETURN
79 CONTINUE
   IF (IPRINT.GT.0) WRITE(IPRINT,979)X
979 FORMAT(' EXIT OF RETARD AT X=',E18.4)
   IDID=2
   RETURN
END

C
C FUNCTION YLAG(I1,X,PHI,RPAR,IPAR)
C ---------------------------------------------------------
C     THIS FUNCTION CAN BE USED FOR CONINUOUS OUTPUT IN
C     CONECTION
C     WITH THE OUTPUT-SUBROUTINE FOR RETARD. IT PROVIDES AN
C     APPROXIMATION TO THE I-TH COMPONENT OF THE SOLUTION
C     AT X.
C ---------------------------------------------------------
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION RPAR(*),IPAR(*)
COMMON /CORER/CON(10000)
COMMON/CORED/CONI(10000)
COMMON /COREI/ND,ICOMP(1)
COMMON /POSITS/X0,UR4,HMAX,IACT,IPOS,
*IRTRN,IDIF,MXST,IPRINT
C ----- INITIAL PHASE
   COMPAR=UR4*MAX(ABS(X),ABS(X0))
   IF (X-X0.LE.COMPAR) THEN
      IF (IRTRN.LE.3) THEN
         YLAG=PHI(II,X,RPAR,IPAR)
         IF (IRTRN.EQ.2) HMAX=MIN(HMAX,X0-X)
         IF (X0-X.LE.COMPAR) IRTRN=3
         RETURN
      ELSE
         IF (X0-X.GT.COMPAR) THEN
            YLAG=PHI(II,X,RPAR,IPAR)
            RETURN
         END IF
      END IF
   END IF

C ----- COMPUTE PLACE OF II-TH COMPONENT
   I=0
   DO 5 J=1,ND
      IF (ICOMP(J).EQ.II) I=J
   5 CONTINUE
   IF (I.EQ.0) THEN
      IF (IPRINT.GT.0) WRITE (IPRINT,*)
       &           ' NO DENSE OUTPUT AVAILABLE FOR COMP.',II
      RETURN
   END IF

C ----- COMPUTE THE POSITION OF X
   IF (X-CON(IACT).LT.-COMPAR) THEN
      IF (IPRINT.GT.0) WRITE (IPRINT,*)
       &          ' MEMORY FULL, MXST = ',MXST
      IRTRN=-1
      endif
      RETURN
   END IF
   INEXT=IACT-IDIF
   IF (INEXT.LT.1)INEXT=(MXST-1)*IDIF+1
   XRIGHT=CON(INEXT)+CON(INEXT+IDIF-1)
   IF (X-XRIGHT.GT.UR4*MAX(ABS(X),ABS(XRIGHT))) THEN
       pause
       IF (IPRINT.GT.0) WRITE (IPRINT,*)
& ' DONT USE ADVANCED ARGUMENTS '  
IRTRN=-1  
RETURN  
END IF  
1 CONTINUE  
IF (X-CON(IPOS).LT.-COMPAR) THEN  
   IPOS=IPOS-IDIF  
   IF (IPOS.LT.1) IPOS=(MXST-1)*IDIF+1  
   GOTO 1  
END IF  
2 CONTINUE  
INEXT=IPOS+IDIF  
IF (INEXT.GT.(MXST-1)*IDIF+1) INEXT=1  
IF (X.GT.CON(INEXT).AND.INEXT.NE.IACT) THEN  
   IPOS=INEXT  
   GOTO 2  
END IF  
C ----- COMPUTE DESIRED APPROXIMATION  
THETA=(X-CON(IPOS))/CON(IPOS+IDIF-1)  
THETA1=1.D0-THETA  
I=I+IPOS  
YLAG=CON(I)+THETA*(CON(ND+I)+THETA1*(CON(2*ND+I)  
+THETA*(CON(3*ND+I)+THETA1*CON(4*ND+I))))  
RETURN  
END  
C  
FUNCTION YDELAY(I,X,RPAR,IPAR)  
C ---------------------------------------------------------  
-  
C THIS FUNCTION CAN BE USED FOR CONINUOUS OUTPUT IN  
C CONECTION  
C WITH THE OUTPUT-SUBROUTINE FOR RETARD. IT PROVIDES AN  
C APPROXIMATION TO THE I-TH COMPONENT OF THE SOLUTION  
C AT X.  
C ---------------------------------------------------------  
-  
IMPLICIT DOUBLE PRECISION (A-H,O-Z)  
DIMENSION RPAR(*),IPAR(*)  
COMMON /CORER/CON(1000)  
COMMON/CORED/CONI(1000)  
COMMON /COREI/ND,ICOMP(1)  
COMMON /POSITS/X0,UR4,HMAX,IACT,IPOS,IRTRN,  
*IDIF,MXST,IPRINT  
C ----- INITIAL PHASE
COMPAR=UR4*MAX(ABS(X),ABS(X0))
    IF (X-X0.LE.COMPAR) THEN
        IF (IRTRN.LE.3) THEN
            YDELAY=0.0D0
            IF (IRTRN.EQ.2) HMAX=MIN(HMAX,X0-X)
            IF (X0-X.LE.COMPAR) IRTRN=3
            RETURN
        ELSE
            IF (X0-X.GT.COMPAR) THEN
                YDELAY=PHI(II,X,RPAR,IPAR)
                RETURN
            END IF
        END IF
    END IF

C ----- COMPUTE PLACE OF II-TH COMPONENT
I=0
DO 5 J=1,ND
    IF (ICOMP(J).EQ.2) I=J
  5  CONTINUE
    IF (I.EQ.0) THEN
        IF (IPRINT.GT.0) WRITE (IPRINT,*)
          & ' NO DENSE OUTPUT AVAILABLE FOR COMP.',II
        RETURN
    END IF
C ----- COMPUTE THE POSITION OF X
X1=X-CONI(IACT)
    IF (X1.LT.-COMPAR) THEN
        IF (IPRINT.GT.0) Then
            pause
            WRITE (IPRINT,*)
            & ' MEMORY FULL, MXST = ',MXST
            IRTRN=-1
            endif
            RETURN
    END IF
    INEXT=IACT-IDIF
    IF (INEXT.LT.1) INEXT=(MXST-1)*IDIF+1
    XRIGHT=CONI(INEXT)+CONI(INEXT+IDIF-1)
    IF (X-XRIGHT.GT.UR4*MAX(ABS(X),ABS(XRIGHT))) THEN
        print*,INEXT
        pause
        IF (IPRINT.GT.0) WRITE (IPRINT,*)
        & ' DONT USE ADVANCED ARGUMENTS '

IRTRN=-1
RETURN
END IF

1 CONTINUE
IF (X-CONI(IPOS).LT.-COMPAR) THEN
   IPOS=IPOS-IDIF
   IF (IPOS.LT.1) IPOS=(MXST-1)*IDIF+1
   GOTO 1
END IF

2 CONTINUE
INEXT=IPOS+IDIF
IF (INEXT.GT.(MXST-1)*IDIF+1) INEXT=1
IF (X.GT.CONI(INEXT).AND.INEXT.NE.IACT) THEN
   IPOS=INEXT
   GOTO 2
END IF

C ----- COMPUTE DESIRED APPROXIMATION
THETA=(X-CONI(IPOS))/CONI(IPOS+IDIF-1)
THETA1=1.D0-THETA
I=I+IPOS
YDELAY=CONI(I)+THETA*(CONI(ND+I)+
   *THETA1*(CONI(2*ND+I)+THETA*
   &(CONI(3*ND+I)+THETA1*CONI(4*ND+I))))
RETURN
END

SUBROUTINE CDOPRI(C2,C3,C4,C5,E1,E3,E4,E5,E6,E7,
   *A21,A31,A32,A41,A42,A43,A51,A52,A53,A54,
   *A61,A62,A63,A64,A65,A71,A73,A74,A75,A76,
   &D1,D3,D4,D5,D6,D7)
C ---------------------------------------------------------
C     RUNGE-KUTTA COEFFICIENTS OF DORMAND AND PRINCE (1980)
C ---------------------------------------------------------
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C2=0.2D0
C3=0.3D0
C4=0.8D0
C5=8.D0/9.D0
A21=0.2D0
A31=3.D0/40.D0
A32=9.D0/40.D0
A41=44.D0/45.D0
A42=-56.D0/15.D0
A43=32.D0/9.D0
A51=19372.D0/6561.D0
A52=-25360.D0/2187.D0
A53=64448.D0/6561.D0
A54=-212.D0/729.D0
A61=9017.D0/3168.D0
A62=-355.D0/33.D0
A63=46732.D0/5247.D0
A64=49.D0/176.D0
A65=-5103.D0/18656.D0
A71=35.D0/384.D0
A73=500.D0/1113.D0
A74=125.D0/192.D0
A75=-2187.D0/6784.D0
A76=11.D0/84.D0
E1=71.D0/57600.D0
E3=-71.D0/16695.D0
E4=71.D0/1920.D0
E5=-17253.D0/339200.D0
E6=22.D0/525.D0
E7=-1.D0/40.D0

C ---- DENSE OUTPUT OF SHAMPINE (1986)
D1=-12715105075.D0/11282082432.D0
D3=87487479700.D0/32700410799.D0
D4=-10690763975.D0/1880347072.D0
D5=701980252875.D0/199316789632.D0
D6=-1453857185.D0/822651844.D0
D7=69997945.D0/29380423.D0
RETURN
END