

A guide to the Etude database

Christie Maas
Hoang Nguyen
Peter Wilders
Cees Vuik
Sebastiaan
Bas Deelder

University of Delft
The Netherlands

14th August 2001

preface

For the last two years several people have worked on the Etude project. This internal report is written by them, and intended to assist the *database supervisors* (eg. authors, assistants) with maintaining the still being developed *Etude database* associated with the courses "Numerical Mathematics IN" and "Numerical Analysis WI". The database itself is called ITS1.

Throughout this report, we will refer to this database as "our database", which is in fact several directories within the database assigned to us. We would like to remark that the term database we use in this report might differ a little bit from the term database in other Etude guides. In those guides, a database might include user data as well.

On maintaining a database, we need to understand the way it is developed. Hence, we will mostly concentrate on the development of our database in section 3, and then discuss some proposals and relevant topics which can be useful on maintaining and updating the database in section 4. After that, we will discuss some problems we have encountered in section 6. Those three sections form the core of this report, discussing how to use Etude to implement the idea of applying information and communication technology in education in such a way that surplus value might be added to the new educational environment. Section 5 contains practical information. Hence, besides all of the technical and specific details, we will also touch on some educational aspects of Etude. Therefore, this report can also be used as a short reference for developing other databases, in particular those associated with mathematics courses.

Contents

1	A short introduction to Etude	4
2	Etude overview	6
3	Database development	9
3.1	EOS ("Etude Opgave Systeem")	9
3.2	EExpl ("Etude Beheer Systeem")	13
3.2.1	An example	13
3.3	ETCS ("Etude Toets Constructie Systeem")	15
3.3.1	General	15
3.3.2	How to actually create a Test	16
3.4	EAS ("Etude Toets Afname Systeem")	17
3.5	Resultviewer ("Etude Resultaten Viewer")	17
4	Database maintenance	21
4.1	technical and educational	21
4.2	Etude at home	21
4.3	Exercise Defaults	22
4.4	Miscellaneous	23
5	Useful tools and tips	24
5.1	Equations in Etude	24
5.1.1	L ^A T _E X	24
5.1.2	Equation editor	24
5.1.3	Our solution	25
5.2	Variables within equations	25
5.3	Pictures in Etude	26
5.4	Local data	26
5.5	Functions as answers	27
5.6	Making test results viewable	27
5.7	Rounding	28
5.8	Hardcopys	28
6	Problems	29

7	Examples	31
7.1	Introduction	31
7.1.1	Used variables in Etude	33
7.1.2	Questions and Answers	34
7.2	Optimization	36
7.3	interpoleren 3	41
A	Tables of exercises	44

Chapter 1

A short introduction to Etude

In the last decade, information and communication technology has taken a great leap forward into our society, having a tremendous amount of influence on our daily lives. The internet, e-mail and many more applications of this technology are rapidly becoming a necessity. One of these applications is *Etude*, which stands for "Educatief Toetssysteem TU-Delft" (or "Educational Test System Delft University of Technology"). At this time, we are working with Etude 2.0, but we have also worked with previous versions, and improvements on Etude 2.0 will certainly be developed.

Etude is a computerized system, designed to ease certain tedious tasks involving constructing and processing academic tests. In our terminology, an academic test can be an exam (whether midterm or final), a homework-project, or a training session. The main difference is that the use of feedback is not possible when creating or taking an exam. Because Etude can deal with certain mathematical data, such as numbers, or even mathematical formulae, it is quite attractive for "numeric" courses, such as certain areas of mathematics.

We would like to make two remarks about the construction and the processing of a test. Firstly, the construction of a test is not only making a set of exercises available, but also constructing associated feedback and hints, assigning weights to different exercises within a test, assigning weights to different questions within a exercise, choosing an appropriate *test model* according to our goal, etcetera. Such construction tasks can be carried out within Etude. Secondly, the processing of a test is not only grading the test accordingly to the weights we have assigned to the exercises/questions within that test (which can be done within Etude) but also to use the results of the test to evaluate our approach, and adjust our goal. Unfortunately, this is not (yet) implemented in Etude. However, if processing a test includes re-using and maintaining it, then Etude seems to be very handy, because we

can not only save the exercises and the complete tests (including their definition/description) in our database, but also modify their contents quickly and easily (this is the most convenient aspect of electronic documents). Furthermore, we can parameterize the variables available in Etude so that a family of exercises can be obtained from merely one exercise. This way the re-use of an exercise or test makes more sense.

Before having such accomplished, we must, of course, have a place for our database in the *Etude server* and the authority to work on it (using certain Etude tools). Within our university, this can be done by contacting the Etude Project Team [4]. After that, we can begin with developing our own database. At last, but certainly not least, keeping a test attractive is not an easy task. Both its outer look, as its inner beauty (ie. the scientific and educational quality) are very important. There is a great difference between a test on paper, and a test done on a computer, which should be taken into account. We have tried to give our tests quite good outer looks. This fact can be found here and there throughout the report. We have also still been trying to keep the scientific quality of our database being guaranteed and up to date. However, a total revision of any database, let's say within each three years, seems to be necessary in order to maintain this goal. A thorough discussion about educational quality can be found in [1], the educational guide of Etude. We would like to invite interested readers to read that valuable guide, because it is not just a guide. From our point of view, it is a book of reference, which reflects the whole idea, the concept of Etude so that practically useful rules (of thumb) therein are motivated by sound, yet interesting philosophical and theoretical background. In this way, educational quality of practical approaches can be ensured (since we know what should and what not), without any loss of creativeness (since we still have room to adjust the rules (of thumb) therein). After all, and from our point of view, educating is not merely a science, it is also an art. We will leave this general introduction to go into more technical and specific details.

For more information about historical and other interesting aspects of Etude, we would like to refer to the following website:

<http://www.icto.tudelft.nl/>

This is the website of the center of expertise ICTO (in Dutch: "Informatie en Communicatie Technologie in het Onderwijs", or Information and Communication Technology in Education") which is a special group working on educational improvements, consisting of various specialized teams, among others the friendly Etude Project Team.

For more information about our work with Etude, please visit:

<http://ta.its.tudelft.nl/WAGM/users/wilders/test>

Here you can find this report, among with some other guides and programs.

Chapter 2

Etude overview

The effect of Etude can only be visible if we go through its cycle, from the very beginning up to the very end of Etude. Hence, before going into specific details in the next section, we would like to give in this section a short description of the total functionality of Etude.

As said before, Etude is a (computerized) system, designed to ease certain tasks when constructing and processing an academic test. In other words: Etude is a CAD (Computer Aided Design) software package, aiming at the construction and the processing of academic tests. Moreover, Etude is a *client-server software package*, consisting of a central database on the Etude server (this is in fact our "home directory") and five *client programs* on every user's machine. Therefore, the installation of Etude would consist of assigning a place in the central database for each user (or group of users), as well as installing each of the five client programs on each user's machine. At this moment, the complete installation can only be carried out by the Etude Project Team, but each of the five client programs can be installed separately. The Etude Taking Test client is the only client program that can be installed separately by every student (if working at home) or by the administrator (in the computer rooms).

After having the installation completed, we can use the client programs (or just "programs" for short) as tools to develop and to manage our database. Note that the students are also database users, but they do not have the authority to work on the database. We are in fact *database supervisors*, who are able to supervise/control the development and management of their own database. Assigning such authority, along with other advanced management tasks, such as modifying user's data, adding database users, adding course code as keyword, backing up, etc., can only be carried out by the *Etude administrators*. When constructing any academic test, we first need to have a set of building materials, which are in fact exercises. The exercises in turn need to be built from other materials, which are, as we can guess, questions and sound descriptions of the problems being considered.

Both questions and problem descriptions in turn consist of text and/or image(s). However, we can safely stop with the refinement since we can hardly edit complex "text" (e.g. mathematical formulas) and images within Etude. Hence, within Etude, we will consider questions and problem descriptions as raw materials, which might be partly produced by ourselves and partly "delivered" to Etude by third parties (eg. ps2gif or Matlab graphic software for graphical illustrations).

Now that we have made a rough top-down refinement, we can use it to build our test bottom-up. Within Etude, the raw materials, ie. problem descriptions and associated questions, are used to build exercise(s) by invoking the program *EOS.exe* ("Etude Opgave Systeem" in Dutch, or "Etude Exercise System"). The constructed exercises are in turn used to build test(s) by invoking the program *ETCS.exe* ("Etude Toets Constructie Systeem" in Dutch, or "Etude Test Construction System"). When a test has been constructed, the students can take that test by invoking the program *EAS.exe* ("Etude Afname Systeem" in Dutch, or "Etude Taking Test System"). After that, their score can be viewed by invoking the program *Resulta.exe* ("Resultaat Viewer" in Dutch, or "Result Viewer"). These last two programs are very important to use as a feedback system to test and improve exercises. More details about those two programs can be found in the "Etude HTML-help" [4].

There is still one client program left. That is the program *EExpl.exe* ("Etude Beheerssysteem" in Dutch, or "Etude Explorer"). As the name suggests, this program does have more or less the same function as the famous Windows Explorer. That means we can use features such as "new", "rename", "copy" and "paste" to create new directories, rearrange files to have a well-structured, transparent database. Of course, we can provide our files or directories with short descriptions as well. The Etude Explorer will also be discussed further in the next section.

We conclude this section by the following table, which is in fact a summary of what we have discussed in this section.

Program	Full name	Short description
EOS.exe	Etude Opgave Systeem (Etude Exercise System)	Exercise construction tool
ETCS.exe	Etude Toets Constructie Systeem (Etude Test Construction System)	Test construction tool
EAS.exe	Etude Afname Systeem (Etude Taking Test System)	Taking test environment
Resulta.exe	Resultaat Viewer (Result Viewer)	Result viewer
EExpl.exe	Etude Beheerssysteem (Etude Explorer)	Exploration tool

Table 2.1: Etude client software package

Chapter 3

Database development

In this section, the tools necessary to build a solid database are explained. First, for clarity's sake, we will explain what is in the database. The database contains a lot of data (so a clear directory-structure is necessary!). There are tests, exercises, questions, feedbacks and hints. In section 3.2 we will further explain the structure, but for now it is sufficient to explain that a test contains exercises, and exercises contain questions.

Test			
Exercise		Exercise	
question	question	question	question

Table 3.1: Etude test structure

The database contains mainly exercises, so lets start with the Exercise system.

3.1 EOS ("Etude Opgave System")

In this section, we will discuss the construction of an exercise, starting with constructing the raw materials needed, which are problem descriptions and associated questions as said in section 2. In addition, we might first need to have a common lay-out for all of the exercises, in order to obtain a good outer look as well as other advantages mentioned in section 2. This can be done in EOS, by using one and the same template as a background for all exercises (remember: if the template is changed, then every exercise

made with the template will also change, but once a template is chosen for an exercise, it is not possible to change to another template!). However, it must be available at the moment we begin to construct our exercises. Hence, let's discuss the construction of a template first.

Template construction

After starting EOS.exe, we can begin to create our template by selecting the option for creating a new template available in EOS. At this moment, there are seven different kind of template frames available in Etude, which are all adaptable, so anything can be made. Also, images and/or OLE-objects (OLE: Object Linking and Embedding) can be inserted in the template. This way, we can fully make use of our CAD skill (CAD: Computer Aided Design). We would like to remark that not all image formats can be handled by Etude, especially well-known image formats under Unix/Linux such as EPS (EPS: Enhanced Postscript), because Etude is an MS Windows software package. GIF (GIF: Graphics Interchange Format) and BMP (BMP: BitMaP) are the only formats supported when working with Etude. Therefore, for Unix/Linux users like us, we might need to use certain *conversion programs*. More about conversion programs will be discussed in section 5.

Furthermore, we can reserve places in a template for the raw materials needed to create exercises. For example, we can reserve a text field for the problem description, or one or more question fields for the associated questions. In the exercise construction phase, we can then place the actual problem description and associated questions into those reserved places. However, those reserved places are fixed in the exercise construction phase, we can not manipulate them anymore. This fact might give raise to some problems, e.g. when there are not enough questions to fill in the reserved places. Therefore, we have not made use of *this* facility in templates.

After all, designing good templates is not an easy task, because a template in Etude might on one hand be considered as the visiting-card of our course and it should on the other hand serve educational purposes. Interesting aspects concerning good template designs are discussed in the educational guide of Etude [1]. In our database, we have decided keep it classical and simple, by just inserting the logo of our university and the code of our course into our template. Nevertheless, it is still elegant enough, at least from our point of view.

Having created a template, and saved it in our database of course, we can use it for the to be constructed exercises, by selecting the option for creating a new exercise using an existing template. However, it is not a must create templates first, because they are used for obtaining the advantages mentioned in section 2. We can therefore always select the option for creating a new exercise without using any template. Hence, although all of

our exercises do make use of (the same) template, the following discussion can also be applied in the case that there is no template used.

Exercise construction

As said in section 2, the raw materials for an exercise are a problem description and one or more associated questions. Both problem description and questions might consist of text and/or images, whether or not for illustrating purposes. Within Etude, we can edit simple text by the Etude built-in editor (which is about equal to MS Wordpad). But first of all, we need to select the button for adding a new text field and drag an appropriately big enough area, so that text can be edited within that field (technical details are given in the "Etude HTML-help" file [4]).

We can also insert images into an exercise as we could when dealing with templates in the previous subsection (subsection 3.1). Thus we can once again use third parties (eg. Matlab graphic software) to create the images needed and then insert them into our exercises. Unfortunately, it's a bit complicated to use formulas in Etude. More about formulas will be explained in chapter 5.

Now that we know how to deal with text and images, we can proceed further with the constructions of our exercises. As said before, we can define variables within Etude. An interesting aspect of Etude variables is that we can have deterministic as well as non-deterministic values assigned to them. Deterministic values are fixed numbers given by us, while non-deterministic values are either numbers which are randomly selected by Etude from an interval or a table specified by us. This way, a family of exercises can be obtained from merely one exercise, as mentioned in section 1. For problem descriptions, it is more or less the end of the story; but for questions, there are still several aspects to discuss.

Firstly, there are four types of questions available in Etude which we can choose to fit our formulation: *multiple choice*, *open calculating question*, or just *open question* for short, *hotspot* and *short answer*. We have not used the last two types of questions. With a *hotspot* it is possible to specify areas in a picture where the student can choose the correct area. With the last type, *short answer*, one can enter one or two words, which will be compared to the correct answer. The multiple choice questions are divided into two groups: *1 out of m* and *n out of m*. This type of questioning is very useful for variation of questions within a exercise. However, the most interesting aspect which makes Etude attractive to mathematical courses belongs to the open (calculating) questions. Indeed, mathematical expressions consisting of well-known functions, operators and constants can be evaluated by *Etude* when using this type of question. Supervisors, as we are, can provide mathematical expressions as answers to the open questions on one hand. On the other hand, students can also use mathematical expressions to an-

swer those questions, and with a little trick it is even possible to evaluate complete functions in undetermined variables. More, we, as supervisors, can provide our answers with marges and/or numbers of precision so that any well-known numerical error (such as round-off error) made by our students during the calculation phase can be taken into account.

Secondly, as mentioned in section 2, we can provide all kind of questions with accessories such as correct/wrong feedback, general feedback, and hints, even combined with other properties such as weight, or maximal number of attempts to answer the question. We even have the option to allow the students to use windows-programs such as the calculator, Matlab or Maple. As we mentioned before, educating is not merely a science, it's also an art. Therefore we strongly recommend to use those features, but we will not go into detail. However, we would like to invite readers once again to read the valuable educational guide of Etude [1], which consists of numerous interesting philosophical and theoretical background about educational aspects of Etude. For technical details, we would like to refer readers to the "Etude HTML-help" file [4].

Thirdly, the exercises within the same tests can be presented to the students in three kinds of arrangement:

1. *free choice*,
2. *fixed order*,
3. *conditional alternative question*

As the names suggest, the students are free to choose the questions they want to (or are able to) answer in "free choice", or they must answer the questions accordingly to the order given in "fixed order", or they can have alternative questions presented accordingly to their current score in "conditional alternative question". This last option is related to the *adaptive exercise model*. More information about this last model can be found in section 3.3. As a consequence, the third arrangement can only be applied and make sense if there are a lot of alternative questions to the same problem. Due to the lack of time to design such alternative questions, we could not provide the exercises in our database with this option. We have decided to take the classical approach: the students are free in answering the questions, ie. only "free choice" is present in our exercises.

After having the problem description and associated questions constructed, we can provide the new exercise with the following three features: *keyword*, *course names* (or *course code*), and *level of difficulty*. Those features can be used to compose an appropriate test. More about this fact will be discussed in section (3.3).

Before leaving the construction of an exercise to go on with an explanation of the Etude-Explorer and the construction of a test, we would like to

make a final remark on the combination of feedback and hint with multiple attempt and/or alternative answers. Unfortunately it is not possible to use variables in feedback which causes a mayor problem. We have found it is difficult to solve this problem, but there are ways around it, see chapter 6.

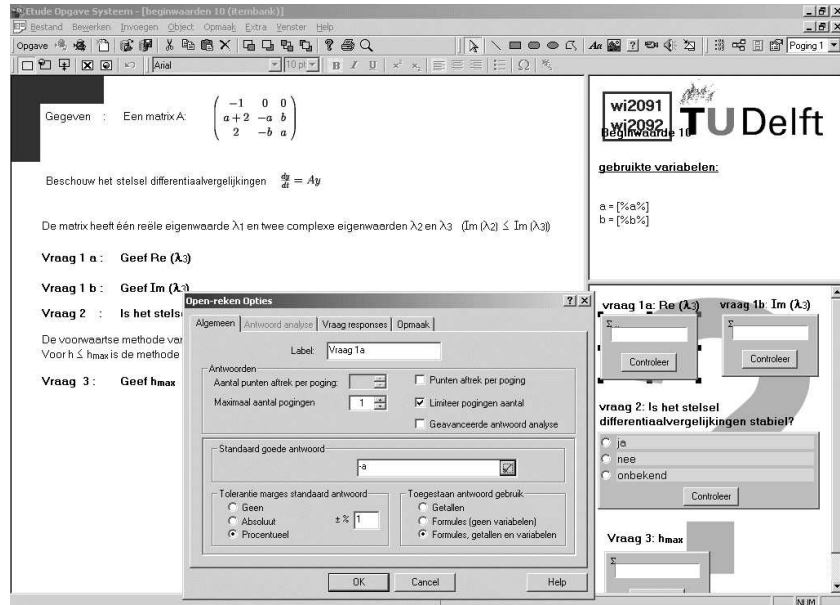


Figure 3.1: Exercise construction

3.2 EExpl ("Etude Beheer Systeem")

First of all, it is very important to create a clear directory structure within the database. The Etude explorer allows a supervisor to copy, delete etc. all items within the database. We have decided to give our database a tree-structure, which should be correlated to the structure of a given course. Therefore, our database is divided into subdirectories accordingly to the main subjects treated in that course. The subdirectories should also be divided further, in order to guarantee the transparent structure of the database. This database is located on the Etude server and is therefore accessible by everyone working on the exercises and for students to take the tests.

3.2.1 An example

We have, using the Etude Explorer, provided our database with several subdirectories, one for each course given. For example, in the directory for "wi2025in", we have created three subdirectories and named them accord-

ing to the three main subjects treated in the course "Numerical Mathematics IN": "Computational Linear Algebra", "Differential Equations", and "Nonlinear Equations and Optimization". Those three subdirectories are in turn provided with three subdirectories: "Exercise", "Hint" and "Feedback", which contain exercises, hints and feedbacks respectively as their names suggest. For the sake of transparency, the hints (feedbacks) belonging to an exercise are grouped into a directory named after that exercise and located within the directory "Hint" ("Feedback"). Naturally, this is only useful when there are a lot of hints, or feedback.

Besides, there is another subdirectory named "Assignment", located at the very first level of the directory "wi2025in", which should be used to hold all of tests.

Moreover, we have proposed and strictly followed some rules of thumb, concerning mathematical symbols/formulas, names of hints or feedbacks, fonts, etcetera, in order to keep our database more transparent and consistent. Those rules of thumb are described in detail, see section 4.3.

The current structure of our itembank *its1* is a direct result of what we have just discussed, and it is given in figure 3.2. At last, but not least, we have constructed some very simple tests illustrating the use of feedback and hint in combination with multiple attempt and/or alternative answers.

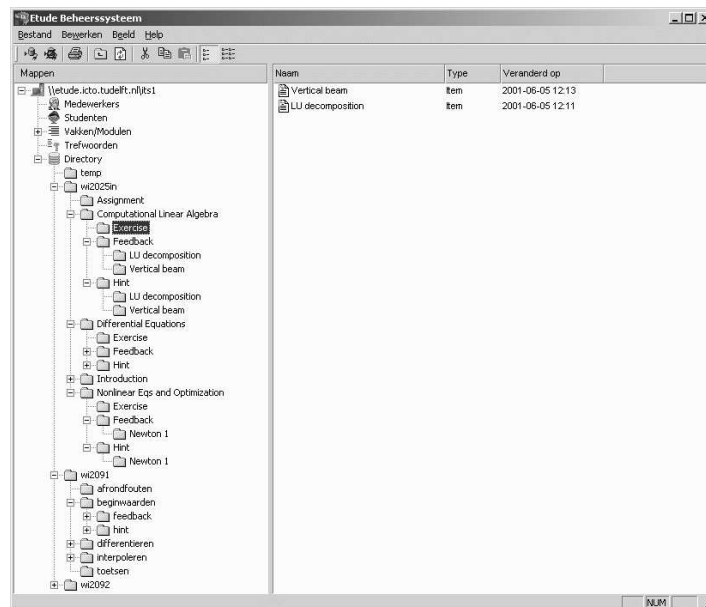


Figure 3.2: Directory structure of *its1*

3.3 ETCS (“Etude Toets Constructie Systeem”)

3.3.1 General

Having a set of exercises available in the database, we can go to the next level to construct a test. From our point of view, a test can be viewed as a selection of exercises which realizes a certain goal (in our mind). This raises two questions:

1. *What is our goal?*
2. *How and what do we select?*

This section is devoted to give an answer to those questions.

First of all, and as mentioned in section 2, there are two test models available in the current version of Etude, the *loose exercise model* and the *adaptive model* (the third model, so-called *interactive model*, will come along with the next version of Etude). As the names suggest, the exercises in an adaptive model are conditionally presented, while all of them are presented to the students in a loose exercise model. As mentioned in section 3.1, there are a lot of similarities between these two models and the three questions arrangements. Indeed, the loose exercise test model and free choice questions arrangement are analogy of each other. However, there is a subtle difference between the adaptive test model and the conditional alternative questions arrangement, although both of them are dependent on the performance of the students during the test. The main difference is that the latter is designed to give the students a second chance, while the former is designed to estimate their skill level, based on which appropriate actions can be taken. Therefore, in an adaptive test, the students might get some easier exercise, or to be more precise, an exercise with lower level of difficulty, presented. More, the students can only get exercises with higher level of difficulty presented if they do pass some exercise with certain level of difficulty assigned apriori by us. We refer readers to the educational guide of Etude [1] for theoretical background and motivation of the adaptive model, and to the “Etude HTML-help” file [4] for technical details. As a consequence of the adaptive test model, there should be plenty of exercises distributed all over the range of the level of difficulty (which is between 0 and 1) if we want to fully make use of its adaptability. This is also indicated in both of the Etude references [1] and [4]. But, as already mentioned, our database is still being under strong development, hence there are not enough exercises in our database for us to try out the very promising adaptive model.

3.3.2 How to actually create a Test

Since it doesn't matter which model we choose, the discussion in the following can be applied to both of them. After having chosen a test model according to our goal, we can reserve places in the test for the exercises being presented to the student, by adding the so-called *exercise holders* (or groups of exercise holders) to the test. For each exercise holder, we can then assign a group of exercises available in our database directly by right-clicking on a exercise holder. After that, we get a pop-up menu, and there are several possibilities: one is to browse through our database and select an exercise, or we can do it indirectly by giving certain *criteria* with respect to the three exercise features discussed in the previous subsection. In both cases, during the time of taking test, Etude will select exactly one exercise from each (non-empty) exercise holder (at random in case there is more than one exercise in that exercise holder), and present that exercise to the student. In the latter case, the selected exercise must meet the given criteria. We will clarify this by the following simple example. Suppose we have given a certain keyword as a criterion to a certain exercise holder. During the test, Etude will look for all of the exercises been provided with the given keyword, and then randomly select one out of those exercise to present to the student. As a consequence, the criterion should be met at least by one exercise, otherwise Etude will have nothing to present. More, we can use two or more criteria, such as a combination between a certain keyword and a certain range of difficulty. In this case, the selected exercise must meet all of the given criteria. However, using difficulty as criterion does not make sense in an adaptive test, because an adaptive test has already been provided with a built-in selecting mechanism with levels of difficulty as selecting criteria. We can also assign weights to the exercise holders within a test as we could when dealing with questions within an exercise. Technical details can again be found in [4].

After constructing a test, the students cannot take that test yet. We need to provide that test with a *definition* (or "afnamedefinitie" in Dutch). That mean we can add certain features to that test, such as a short description, when and how long can that test be taken, especially, who can take that test, etcetera. Note that only the students allowed by us, in the definition of taking test, can take that test (by invoking the program EAS.exe as mentioned before). All one has to do is click on the button "afnamedefinitie", and again, a pop-up menu will appear. Be aware that if you alter an exercise which has been assigned to a test, it is not necessary to recreate the test. Because a test definition only links exercises and does not contain them, it will always use the most up to date exercise at the moment of taking that test.

Now that we have gone through the development of our database, organizing it in subsection 3.2, constructing the exercises in subsection 3.1

and constructing the tests in subsection 3.3, we will devote the rest of this section to the students, who will have to take the test after all.

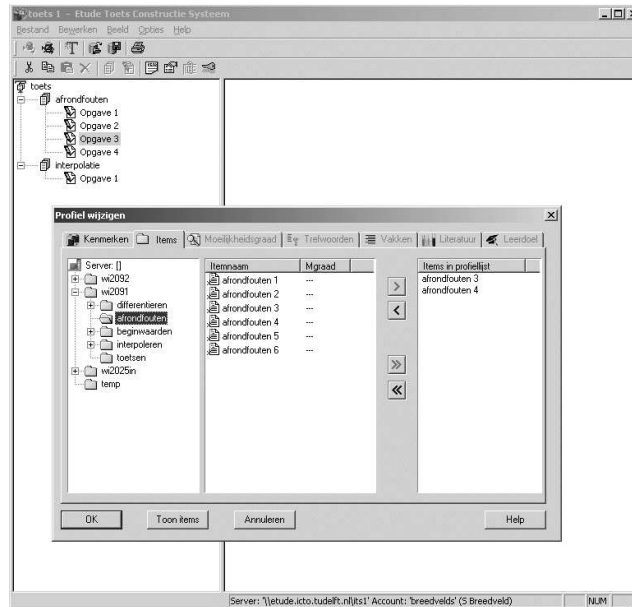


Figure 3.3: Test construction

3.4 EAS (“Etude Toets Afname Systeem”)

The EAS program is the only satellite program the students see. basically, it has to be installed on the computers the students use. They *can* start it up at all times, but they can only take the test at the times *we* have specified, with exception of tests which are always open, of course. As far as we have evaluated, the students think the program rather pleasant to work with. All they have to do is start the program, and everything after that is given in menus, so there shouldn’t be any problems there. The EAS-program can use some standard windows-programs, but we have to include that option when we construct an exercise (see also section 3.1).

3.5 Resultviewer (“Etude Resultaten Viewer”)

When the resultviewer is started, it asks about which test data has to be displayed, and delivers a table in which *all* data about the requested test is given. We can view all results, so as to get an overview, or we can look at what the individual students have done and we can print them. The most important feature of this program is that we can evaluate the tests we

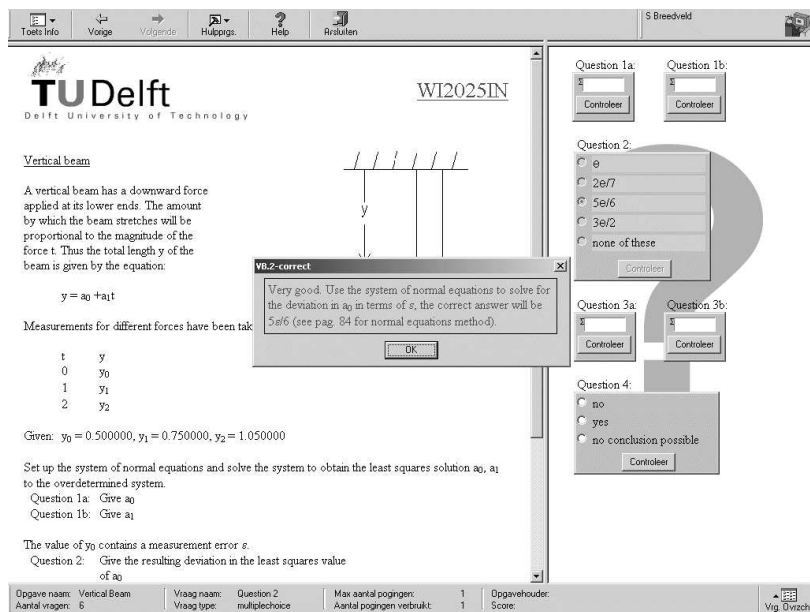


Figure 3.4: EAS at work

have made, and make adjustments accordingly. See subsection 5.6 for more information about how to use the results.

We conclude this section by the table 3.2, which is in fact a working schedule so as to make a test.

Etude Resultaat Viewer

Bestand Afname Beeld Help

toets: \\etude.icto.tudelft.nl\its1
afname: test 3

student: _____
studienummer: _____

nr.	naam	studie...	IP adres	score	datum							
1	J ...	11015...	130.161.31.124	55.78	13-03-2001, 10:44	Newton 1	34.00	Newton 1	0.00	Ne...	10...	Ne...
2	J v...	11014...	130.161.31.150	33.56	13-03-2001, 10:45	Newton 1	34.00	Newton 1	0.00	Ne...	10...	Ne...
3	M ...	11026...	130.161.31.141	66.67	13-03-2001, 10:45	Newton 1	100.00	Newton 1	0.00	Ne...	0.00	Ne...
4	M ...	11015...	130.161.31.151	88.89	13-03-2001, 10:46	Newton 1	100.00	Newton 1	100.00	Ne...	10...	Ne...
5	B ...	11024...	130.161.31.149	66.67	13-03-2001, 10:47	Newton 1	100.00	Newton 1	0.00	Ne...	10...	Ne...
6	M...	9919...	130.161.31.142	78.22	13-03-2001, 10:47	Newton 1	68.00	Newton 1	100.00	Ne...	0.00	Ne...
7	E ...	11014...	130.161.31.130	77.78	13-03-2001, 10:47	Newton 1	100.00	Newton 1	100.00	Ne...	0.00	Ne...
8	R ...	11014...	130.161.31.169	77.78	13-03-2001, 10:47	Newton 1	100.00	Newton 1	0.00	Ne...	10...	Ne...
9	P...	11015...	130.161.31.137	88.89	13-03-2001, 10:47	Newton 1	100.00	Newton 1	0.00	Ne...	10...	Ne...
10	G ...	11015...	130.161.31.134	100.00	13-03-2001, 10:47	Newton 1	100.00	Newton 1	100.00	Ne...	10...	Ne...
11	S ...	11014...	130.161.31.138	89.33	13-03-2001, 10:47	Newton 1	68.00	Newton 1	100.00	Ne...	10...	Ne...
12	J...	11015...	130.161.31.131	88.89	13-03-2001, 10:48	Newton 1	100.00	Newton 1	100.00	Ne...	10...	Ne...
13	D ...	99644...	130.161.31.126	56.00	13-03-2001, 10:48	Newton 1	68.00	Newton 1	0.00	Ne...	10...	Ne...
14	J v...	11019...	130.161.31.147	55.56	13-03-2001, 10:48	Newton 1	0.00	Newton 1	0.00	Ne...	10...	Ne...
15	B ...	11014...	130.161.31.128	88.89	13-03-2001, 10:48	Newton 1	100.00	Newton 1	100.00	Ne...	10...	Ne...
16	L B...	9123...	130.161.31.135	77.78	13-03-2001, 10:49	Newton 1	100.00	Newton 1	100.00	Ne...	10...	Ne...
17	L B...	11014...	130.161.31.125	67.11	13-03-2001, 10:50	Newton 1	68.00	Newton 1	100.00	Ne...	0.00	Ne...
18	beer	1	130.161.31.132	0.00	13-03-2001, 10:51	Newton 1	0.00	Newton 1	0.00	Ne...	0.00	Ne...
19	F...	99042...	130.161.31.127	11.11	13-03-2001, 10:54	Newton 1	0.00	Newton 1	0.00	Ne...	10...	Ne...
20	R ...	9783...	130.161.31.173	56.00	13-03-2001, 10:54	Newton 1	68.00	Newton 1	100.00	Ne...	10...	Ne...
21	S ...	184710	stigmata.twi.tudelft.nl	11.11	25-06-2001, 12:16	Newton 1	-1.00	Newton 1	100.00	Ne...	0.00	Ne...

vragen naar overzicht

Figure 3.5: Result overview

Etude Resultaat Viewer

Bestand Afname Beeld Help

toets: \\etude.icto.tudelft.nl\its1
afname: test 3

student: _____
studienummer: _____

result uit database	Eigenschap	Waarde
nonlinear equations and opti	CQuestion score	100.00
Question 3	gegeven antwoord	0.37
Question 5		
Question 4		
Question 1b		
Question 2a		
Question 2b		
Question 1a		

vragen naar overzicht

Figure 3.6: Result detailed

Program	Short description
EOS.exe	Create a template Create a exercise Create feedback and hints, and link them to the exercise
ETCS.exe	Create a test Create a definition
EAS.exe	Test the test yourself!
EOS.exe	review the exercise
EAS.exe	Have the students take the test
Resulta.exe	review the results
EExpl.exe	Everything still in the right place?
Start all over again	

Table 3.2: Working schedule

Chapter 4

Database maintenance

4.1 technical and educational

Here and there in this report, especially in the database development section (section 3), we have touched on several aspects which should be maintained. In this section, we will summarize them and discuss some other interesting aspects as well. It is *very important* for both technical as educational reasons to have a consequent database. We have said this before, but it cannot be emphasized enough.

Although computers have become more and more popular and effective, it is hard to imagine that computerized courses such as Etude tests will ever replace the classical courses due to the lack of several important aspects of a lively verbal communication. Nevertheless, computerized courses should not be ignored, because written materials in computerized courses can be easily updated. Hence we should take advantage of the computer to make Etude more valuable by attaching more educational aspects to it, so that it can become a very good complementary tool in education. This can be done, for example, by exploiting the adaptability of Etude. By doing that, we should pay more attention in design appropriate exercises, eventually with appropriate alternative questions.

The database should be reviewed every two years or so, to insure the exercises are up to date with the objectives of the given courses.

Analyzing the test results, and revising the exercises in our database on time seem also be very worthwhile. This way, we can help the students more directly, and keep them well-informed of latest developments/technologies.

4.2 Etude at home

Etude can be used at home. The students can download the Etude taking test environment from the Etude download page and install it on their own windoze computer. Newer versions of Etude even work behind a

firewall. The student does not need a 24 hour internet connection because Etude is optimized to work over a modem connection, ***which can be disconnected during the taking of the test, but you need to reconnect before hitting 'Afsluiten.***

An other surplus value of Etude is that it provides a *link* to the famous Blackboard CourseInfo. It is possible to present an Etude test to the students via Blackboard. Due to the capability to deal with numerical data of Etude, which Blackboard can not, this is a very good news and should be extensively made use of. What we have to do, is just make a short cut to an Etude test. This can be done by selecting the *definition* ("afnamedefinitie") in the Etude Explorer and select *Take Test shortcut* ("Afname Snelkoppeling" in Dutch) to save a short cut as an *.etu file, and then uploading that *.etu file to an assignment in our Blackboard online course as a hyperbolic link (more over Blackboard online course can be found in [2]). Provided that the students have the Etude taking test environment EAS.exe installed on the machine they are using, they can take that Etude test without any problem. After clicking on the link, EAS is started and the student only has to login. *** We have successfully tried out this Etude feature, although it is not quite optimal in the sense that both of the two browsers, Netscape and Internet Explorer do give an error message, while the Etude test can still be taken.***

4.3 Exercise Defaults

We have used the following defaults for our tests. It is important that all exercises have the same outer look.

- i) Font: Times New Roman 12pts
- ii) Mathematical symbol: NOT in italic NOR bold

And for the sake of consistency and transparency within Etude courses:

- iii) Hint should be in blue and should have a name that reveals the identity of the associated question and exercise (see the "structure" file).
- iv) Feedback on correct and incorrect answer should be in green and red resp. and should have a name that reveals the identities of the associated question and exercise.
- v) Overall or common feedback, that is when feedback is always given, wether the question was answered correctly or not, should have the colour black.
- vi) Exercises are made for a screen resolution of 1024x768.
- vii) When using multiple choice questions, the possible answers should not be given in the exercise, but only in the multiple choice question block. This is because the list of possible answers eats space and students want to have the entire exercise on one screen without scrolling.

4.4 Miscellaneous

At last, but not least, for further information or support about technical and educational aspects of Etude, the following two members of the Etude Project Team can be contacted:

J.F. Krook (J.F.Krook@dto.tudelft.nl) (for technical aspects)

and:

Sofia Dopper (S.M.Dopper@tbm.tudelft.nl)(for educational aspects).

Also note the Educational Guide *Het hoe en waarom van toetsen met Etude* [1] by Sofia Dopper.

There are also two internet pages which contains information about Etude. Those are:

<http://www.icto.tudelft.nl/>, the Etude project page. Here you will find information about the program itself, additional help files, download files, etcetera.

<http://ta.twi.tudelft.nl/WAGM/users/wilders/test/> Here you will find this report among with other documentation concerning Etude and the little program *middel* described in 5.6.

Chapter 5

Useful tools and tips

5.1 Equations in Etude

There are two possible ways to insert a formula in a Etude exercise. Both have advantages and disadvantages. In this chapter we will explain a bit about both. We will end with a *big* problem which occurs in both uses, and a short explanation on pictures.

5.1.1 \LaTeX

A very big drawback for Unix/Linux users, especially mathematicians, is that if we want to use existing materials to construct exercises in Etude, then we might run into the following problem of incompatibility. That is, almost, if not all of our materials are in \LaTeX format because \LaTeX is, and still tends to be, the favorite editing tool for scientists, especially mathematicians. But the final version of a \LaTeX file is a PS file (PS: Postscript), which can not be handled by Etude. The most well-known trick is using *ps2gif* or *gimp*, which is a conversion program under Unix/Linux used to convert PS files to GIF images, and then insert them into the exercises. But, it is not the end of the story yet. In [2], we have pointed out that all of the conversion programs we know are not flexible, in the sense that firstly, the result is a full page (which we can resize with a drawing program such as Paint Shop Pro), and secondly, the converted materials are hardly editable, so that changes and/or updates can hardly be directly carried out without evoking those conversion programs again.

5.1.2 Equation editor

Another possibility to import formulas, is to use microsoft software, such as the equation editor, or mathtype. For us, Equation Editor is a natural choice, because it is a built-in tool of MS Word. MS equation editor is not hard to use at all, if one doesn't want any complicated formulas. Anybody

with some basic menu-oriented skill and some knowledge of any mathematical editor (such as \LaTeX) can handle Equation Editor with ease, because it is very straight forward to use, thanks to its menu-oriented nature. Well-known mathematical symbols and notations, from which we can build up simple mathematical expressions, are present in Equation Editor. Unfortunately, when a formula becomes more complicated, so does the equation editor and if it must be extended unexpectedly, you must even construct the entire equation from scratch. One of the problems that arise, is that the equation editor is peculiar with fonts, and keeps trying to insert the "right" font ("right" according to microsoft, that is). However, (mathematical) expressions produced by Equation Editor can be resized without any significant loss of resolution, although only up to a certain level. Hence, it is no problem if we want to resize our exercises (eg. using a larger font). Readers who are familiar with MS Word, will not find any trouble when using Equation Editor.

5.1.3 Our solution

As with most things in life, the best tool is the tool you know how to use. If you can handle the Equation Editor it is probably wise to use it, but otherwise it may be even wiser to use \LaTeX . For the most part, we have used \LaTeX to create formulas. To create a formula, there has to be a \LaTeX source-file. Then `ps2gif` can be used on that file, to get a GIF-file. The GIF-file has the wrong size, so we have to resize the picture. We have used Paint Shop Pro to resize the file, but other programs can work as well, for example Photoshop. To resize a GIF-file in paint shop pro, simply open the file, select the desired size, and press the button "crop to selection". It's also possible to play with colours, but not necessary when working with GIF. If the black/white proportions are not right, use the histogram function "stretch" to correct that.

5.2 Variables within equations

Here a *big* problem arises (see also section 6). There is no possible way to insert variables used in an exercise in a formula, whether \LaTeX or microsoft equation editor is used. The only way around this problem is to use the symbols of the variables in the formula, and give the actual values in the exercise.

Be sure that you do not make the equations too complicated because the student has to calculate them before he/she could make use of it. If there is no way to simplify them, consider splitting up the exercise in separate ones where the variables have a *fixed* value. Then (re)edit the exercise with its hints and feedback and simplify the equations or values. Then construct the test using the *random order* future, described in 3.3.2.

5.3 Pictures in Etude

Pictures in Etude are very pleasant to work with. We always use GIF format, an other possibility allowed by Etude is the BMP format. We can simply import a picture by clicking on the "insert picture" button, and then selecting the appropriate picture, or we can even make simple pictures ourselves (also usable to create nice tables) by clicking and dragging lines and circles (which is also button-oriented). It is also possible to use other objects like AVI files or *sound samples*. We have not used this because we did not need it, but it can be used, though.

5.4 Local data

Of course we have more data for an exercise than that we should put in the database. Take for example pictures, hardcopy's, extra notitions and remarks, work outs or other data. You do not want to pollute the Etude database with this, because Etude is unable to handle these types of files. The database must be kept clean and only be filled with Etude data. So where do we put the rest of the data? A sensible solution for this is to create a local archive with exact the same directory structure as on the Etude server, but now with different subsubdirectories. Compare figure 5.1 with figure 3.2.

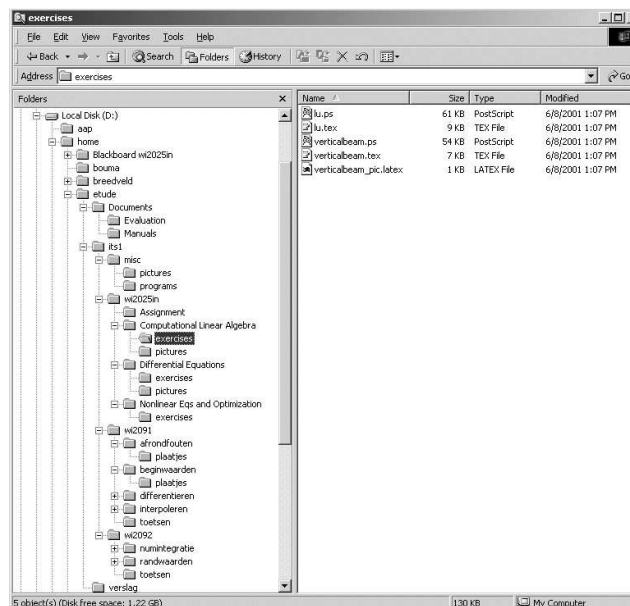


Figure 5.1: Local directory structure of its1

5.5 Functions as answers

It is not directly possible to answer a question like '*Give the function $f(x, y)$ in terms of x and y where x and y are underdetermined*'. So an answer could be ' $(x+x)*y$ ' but ' $2*x*y$ ' should be correct too. To make this work we use a little trick. By default it is possible by Etude to use given variables in the answers, so if it is given that $x = 5$, you answer could be '15', but ' $3*x$ ' is correct too, because Etude fills in the value for x and evaluates the expression. The idea in fact is simple: we force the student to give the solution in terms of x by not revealing the true value of x . You can give the variables values like $x = 12345$, $y = 42135$ or $i = 666$. The answer to the question '*Give a general expression for a complex value*' could be something like ' $x+i*y$ '. Take a look at the *Introduction* test in 7.1 for a more practical implementation.

5.6 Making test results viewable

The students have taken several tests and now the teacher has to make the results of the tests interpretable and viewable. For this purpose a simple C program has been written which should run on any computer running Unix/Linux. The program can be downloaded from [3], read the ReadMe file for installation instructions.

It works as follows: open the test in the Result Viewer and save the results by clicking on the 'Trumpet' icon. This saves the test results in a plain ASCII file. If you wish, or if needed, you can customize the data a bit by removing result lines from test sessions and the like. Do this for every test and transport these files to a Linux computer. There, assumed that the program is already installed, run:

```
middel test1.txt test2.txt test3.txt
```

This will fashion the data: if a student missed a test or screwed up one completely, he will get as mark a 1.0. A big advance is that the input files do not have to be sorted and the students that have taken the test may differ. The output is given in the form (student numbers are scrambled for privacy reasons):

StudentNumber	Test1	Test2	Test3		Average
1015***	7.8	1.0	10.0		6.3
1016***	9.4	1.0	5.6		5.3
1018***	9.4	1.0	8.9		6.4
1035***	7.8	1.7	8.9		6.1
1064***	8.8	1.0	1.0		3.6
9087***	8.1	4.3	8.9		7.1

Of course you can customize the output by simple redirection:

```
middel test1.txt test2.txt test3.txt | sort +0n | less
middel test1.txt test2.txt test3.txt | sort +0n > total.txt
middel test1.txt test2.txt test3.txt | sort +0n | lpr
```

The first will sort on the students number and give you the ability to read the data in a viewer. The second writes the data to an ASCII file 'total.txt' and the third will print the data directly (if a line printer is installed).

5.7 Rounding

A question that has been asked by many students is "What order of significance is expected?". This is certainly something that should be considered *before* constructing any test. It is very frustrating for a student when he/she fills in one decimal to few and the answer is therefore incorrect. You could set as a rule "the three most significant digits" and use this consequently, exceptions explicitly noted. If we take the example stated above, for an answer like 3.65 the precision in Etude must be set to 0.005 absolute deviation. Of course, you can make your own rules, just as long as these are clear to the student.

5.8 Hardcopys

For every exercise we have make so called *hardcopys*. These are documents which contains the entire exercise, along with any other information that is necessary to construct the exercise in Etude. If desired, one can also add an entire (mathematical) workout. Examples are given in chapter 7.

We have found it convenient when constructing a brand new exercise, to do this on plain paper first. When you think you have a nice exercise, work on the hardcopy of it. Now you know what information Etude needs, and, more important, you have a good overview. Now it is easy to review the exercise, construct proper feedback and hints, estimate the amount of time needed to make the exercise, etcetera. Only after you are completely satisfied with the exercise you can input it in Etude's EOS.

Following this path not only reduces the amount of mistakes in the exercise (even a teacher is only human, you know) and improves the quality of the exercise, it also reduces the amount of time making one. Because Etude is a windows program, it takes a lot of time to make changes afterwards. Take for example an exercise with 10 questions. A minor change in the exercise can be followed by adapting 20 feedbacks (for *correct* and *incorrect* answers) that results in a large amount of work behind the screen.

Chapter 6

Problems

Unfortunately, Etude is still in a development-stage, so there are still some problems. Some we already encountered, and found solutions for, but some not. The most important problem (in our opinion) lies with the use of feedback.

- Firstly, the feedback as it is implemented now, does not allow for variables belonging to the associated exercise to be embedded in the feedback (or anything, see section 5.2.)
- Secondly, the possibility to incorporate the correct feedback to the corresponding answer (which can be right *or* wrong is only possible in multiple-choice-questions.) For example: in the open answer question, which color is the sky?: Blue, green, or yellow, it is possible to say which is right, and which is wrong. There is no way to tell the student yellow is wrong, because that's the sun, and green is wrong, because that's the grass on the world.

As a consequence, when dealing with open-calculating-questions, if the students are allowed to have multiple attempts to answer a question, then whenever a wrong answer is given, we cannot give the student an appropriate *intermediate feedback* (which has to use the variables *and* be appropriate to the error made), which we call *multistep feedback*. If the last attempt is unsuccessful, we should give the students the correct answer. That means we should need some indication depends on the number of attempts to give the appropriate feedback. But we cannot do that in Etude. Hence, due to the these problems, we have decided to keep our feedback a bit vague, resulting in the following approach: Both feedback to correct and wrong answers is formulated in a general manner, and the correct answer (or solution) is given at the end, in the so-called *end feedback*. This way, the correct answer and its (associated) feedback can only be revealed after a student has given a correct answer or used all of the attempts allowed.

There are more problems, but fortunately those are minor ones, and often there is a way around them (good thinking before implementing anything for example). Once a template is selected for an exercise, it cannot be changed which template is used for that exercise, so if another template is selected, one has to copy and paste everything from the exercise to the new exercise which uses the correct template. The use of different fonts within an exercise is very bothersome, since Etude keeps trying to use the same font everywhere. Also, Etude only recognizes the fonts installed on the computer which it is installed on, but shows a lot more fonts in the fonts option field. We are certain other people will find more problems, but these are the ones we had to deal with.

Problems, errors and/or new (desired) futures should be reported to J.F. Krook (J.F.Krook@dto.tudelft.nl) in order to approve Etude. Bugs will be taken into account almost immediately and be solved in the next release of Etude.

Chapter 7

Examples

As mentioned before, there are currently two courses given which use Etude. In the first course, Numerical Mathematics IN, Etude is used to give exams. We have included a complete report on one of those tests, Optimization (see section 7.2). The students are free to participate (or not) with any of the tests as they please, and they have to take the test at an appropriate time. The results (when adequate and complete) are added to a conventional test. The test itself is in English.

In the second course, Numerical Analysis, we have experimented with using Etude as a homework project. We have also included a report on one of these exercises (see section 7.3) The students are free to log on a computer, and take the tests whenever they please, and do not get any reward in terms of grades. The tests are in Dutch.

These hardcopy's are very usefull when constructing and maintaining an exercise. These overviews are constructed in a way that they are compatible to Etude. Every information that you need to make the exercise in EOS is included here, answers, feedbacks, weightfactors and variables included.

7.1 Introduction

Introduction to Etude

Welcome to Etude. This exercise will give you an introduction to Etude. The exercise (including questions) is in the left frame of the screen (this frame). You can give the answers to these questions in the question dialogs in the right frame.

For each question individually you get usually 100% for a correct answer and 0% for an incorrect answer. Some questions can be partially correct or incorrect. This is the case when you use a hint or if you have a multiple choice question partially answered correct (after the last attempt). The result of separate questions will be transfered to the total score using weight factors (which are invisible for the student). Questions that are more difficult than

others can have a higher weight.

In the lower right corner of the screen you will notice a button 'Vrg. Ovrzcht.'. By pressing this button you will get an overview of the questions and your score for that question. For more information about the test, press the button up-left: 'Toets Info'. The information describes the total scores for the tests in this session. Also the remaining time can be viewed here. When you have 10 or 5 minutes left, you will be alarmed by a notice right in the lowest bar of the screen. Keep an eye on this! When you have finished a test you will also see the total score here.

An exam can consist of multiple tests. You can scroll through the available tests by clicking the left and right arrows on top of the screen. Etude closes the exam automatically if all questions are answered. By pressing the button 'Afsluiten' you can finish the exam and the application will be closed. Of course, all unanswered question will be marked as incorrect.

Additional help can be found under the 'Help' button on the upper section of the screen.

Rounding

A big frustration is still that it is not exactly clear which order of significance is expected. (Mathematical) integer numbers are usually exact. Most real numbers are rounded to the most significant part, at which you usually have a marge of 2% or a deviation of 1/100 from the exact answer. Unfortunately, there are no clear agreements about this. If you are not sure, it is of course safe to answer with as much as possible significant digits.

Variables

Etude can give you different variables. ' $a = [%a\%]$ ' where 'a' is fixed for every student and every time the exam is taken. But it is possible to give each student a different value too. Let ' $b = [%b\%]$ ' different for each student (check with your neighbour).

Given variables are oftenu used in the tests un the so-called 'parameterized test'. It is often wise to fill in the specific value directly, in particular when dealing with matrices. However, if preferred, it is possible to use parameters in your answers. If Greek letters are given, you use the Latin writing for these letters in your answer, so for a you write alpha.

There are several kind questions: open questions, multiple choice with 1 correct answer and multiple choice with at least one good answer.

Open question

With these kind of questions you can give a number as answer. For example, if the answer is 5, you fill in '5' (without quotes). But also powers of 10 and

functions are accepted. 'sin(sqrt(pi))' is correct when you give this answer, so it is not necessary for you to calculate this.

Not only standard functions are accepted, but you can also use given variables in your answer.

Sometimes expressions are asked as answers. For example, take the question 'Compute the derivative of $f(x)=x*\sin(x)$ '. Correct answers are ' $\sin(x)+x*\cos(x)$ ' or ' $\cos(x)*(tan(x)+x)$ ', while incorrect is ' $\sin(1)+\cos(1)$ '. For this type of answers you can use standard operators and functions, as described in the Etude Help, which can be accessed by clicking the 'Help' button at the upper part of the screen.

Multiple choice

There are two different kinds of these: Only one answer is correct; one and only one answer can be chosen: activate the correct (round) button. At least one answer is correct and at least one answer must be given: activate the correct (square) button(s).

With the latter kind of questions the following procedure is used for the score: every correct answer will give you $1/m$ points, where m is the amount of correct answers. every wrong or missing answer results in subtracting $1/m$ points from your current score, with the floor at 0%. So it is not wise to mark every answer as correct.

For Numerical Mathematics we apply the following rule: If answer A is correct and B too, but B implies A, then the full score will only be given if both the B and A button are activated.

subsection*Feedback With most questions you will receive a feedback after you have given the answer. If you have given a wrong answer you will get the correct answer with the feedback, usually with a work out and a reference. WRITE THIS ANSWER DOWN: it is not possible for Etude to use the given (wrong) answers for processing answers to the next questions. If you need the answer on question 1 for question 2 you HAVE to use the answer given in the feedback on question 1 (unless you gave the correct answer of course).

subsection*Hints With some questions it is possible to ask for a hint. If you press the 'Hint' button for that question it will tell you how much points it will cost you. Then you can decide whether you take

7.1.1 Used variables in Etude

Given

- $a = 153$
- $b \in [0.000, 10.000]$

Used

- $x = 34382$
- $y = 23423$

7.1.2 Questions and Answers

Open questions:

- Question 1: 5
 - Answer: 5
 - Attempts: 1
- Question 2: $\sin(\sqrt{\pi})$
 - Answer: $\sin(\sqrt{\pi})$
 - Attempts: 1
 - Tolerance: 0.002%
- Question 3: $\sin(\sqrt{\pi}) = 0.979735932$
 - Answer: $\sin(\sqrt{\pi})$
 - Attempts: 1
 - Tolerance: 0.002%
- Question 4: $a/b + 1e3$
 - Answer: $\frac{a}{b} + 1e3$
 - Attempts: 1
 - No Tolerance
- Question 5: $\sin(x) + x * \cos(x)$
 - Answer: $\sin(x) + x * \cos(x)$
 - Attempts: 1
 - No Tolerance

Multiple choice questions:

- Question 6: only one answer is correct
 - Answer: This is choice 3: CORRECT
 - Possible Answers:
 - * This is choice 1: WRONG
 - * This is choice 2: WRONG
 - * This is choice 3: CORRECT

- * This is choice 4: WRONG
 - Attempts: 1
- Question 7: multiple answers are correct
 - Answer:
 - * A: I am ill
 - * B: I have a fever
 - Possible Answers:
 - * A: I am ill
 - * B: I have a fever
 - * C: this introduction test hearsCHt
 - * D: today is yesterday for tomorrow
 - Attempts: 1
 - Points per correct answer: 50 percent

Feedback:

- Question 8: the correct answer is 7
 - Answer: 7
 - Attempts: 1
 - No Tolerance
 - Feedback Correct: Correct: consult page 666 for more information.
 - Feedback Incorrect: Incorrect: unfortunately, as proposed, the answer was 7 (consult page 666 for more information).
- Question 9: the wrong answer is 3
 - Answer: 84488
 - Attempts: 1
 - No Tolerance
 - Feedback Correct: Correct: consult page 666 for more information.
 - Feedback Incorrect: Incorrect: unfortunately, as proposed the correct answer was not 3 (consult page 666 for more information).
- Question 10: something is correct
 - Answer:
 - * This is choice 3

- * This is choice 4
- Possible Answers:
 - * This is choice 1
 - * This is choice 2
 - * This is choice 3
 - * This is choice 4
- Attempts: 1
- Feedback Overall: With these questions it is more complicated to give for every possible answercombination a feedback. It is much more easy to give ne feedback with the correct answers.
- Question 11: what is this?
 - Answer: 373
 - Attempts: 1
 - No Tolerance
 - Hint: This is a hint. To answer this question correctly, give 373 as answer.
 - Hint discount: 20%

7.2 Optimization

Excercise

Optimization

Consider the 2 dimensional function:

$$f(z) = f(x, y) = x^2 + y^2 - (x + 3\eta)y$$

$$z \in \text{mooieR}, z = \begin{pmatrix} x \\ y \end{pmatrix}$$

where $\eta = GIVEN$.

With the aid of the Steepest Descent method, we are going to determine the extreme of this function. For this, we need the gradient $\nabla f(z) = \begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \end{pmatrix}$

- Question 1:
 - a: Give \dot{z}_1 in terms of x and y .
 - b: Give \dot{z}_2 in terms of x and y .

We are starting in $z_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. To make the first step with this method, we need to determine α_0 first.

- Question 2: Give α_0
- Question 3: Now we know α_0 , determine the point z_1
 - a: x_1
 - b: y_1
- Question 4: Execute these steps again in order to determine z_2
 - a: x_2
 - b: y_2

As we know (or as we ought to know), the Steepest Descent method is fairly slow. We are going to use an other method, the Conjugate Gradient method.

Again, we start in $z_0^* = z_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, where the variables marked with an asterisk belong to the steps executed by the Conjugate Gradient method. We also denote that $-s_0 = g_0 = \nabla f(z_0^*)$ and according to this method $z_1^* = z_1$ and $g_1 = \nabla f(z_1^*)$, whose we have calculated before.

From here on we apply the Conjugate Gradient method.

- Question 5: Give β_1
- Question 6: Give the new search direction s_1
 - a: s_{1_x}
 - b: s_{1_y}
- Question 7: Give the new α_1^*
- Question 8: Give z_2^*
 - a: x_2^*
 - b: y_2^*

Used variables in Etude

Given

- $\eta =$ multiply of 4? Worst fraction is $\frac{33}{16}\eta$.

Used

- $x = 34532$
- $y = 98212$

subsection*Questions and Answers

- Question 1a: Give z_1 in terms of x and y .
 - Answer: $2x - y$
 - Feedback Correct: Good. Consult paragraph 6.3.2 for more information about partial differentiation.
 - Feedback Incorrect: Unfortunately, your answer is incorrect. z_1 is the partial derivative of $f(x, y)$ with respect to x . See paragraph 6.3.2 for more information.
- Question 1b: Give z_2 in terms of x and y .
 - Answer: $2y - (x + 3\eta)$
 - Feedback Correct: Good. Consult paragraph 6.3.2 for more information about partial differentiation.
 - Feedback Incorrect: Unfortunately, your answer is incorrect. z_2 is the partial derivative of $f(x, y)$ with respect to y . See paragraph 6.3.2 for more information.
- Question 2: Give α_0
 - Answer: $\frac{1}{2}$
 - Feedback Correct: Good. See paragraph 6.3.2 and example 6.5 for more information.
 - Feedback Incorrect: Unfortunately, your answer is incorrect. $\alpha_0 = \min_{\alpha} f(z_0 - \alpha \nabla f(z_0))$. Using the information we have had til now, α_0 turns out to be $\frac{1}{2}$. See paragraph 6.3.2 and example 6.5 for more information.
- Question 3: Now we know α_0 , determine the point z_1
 - a: x_1

- * Answer: 0
- * Feedback Correct: Good. See paragraph 6.3.2 for more information.
- * Feedback Incorrect: Unfortunately, your answer is incorrect. It should be 0. See question 3b for why.
- b: y_1
 - * Answer: $\frac{3}{2}\eta$
 - * Feedback Correct: Good. See paragraph 6.3.2 for more information.
 - * Feedback Incorrect: Unfortunately, your answer is incorrect. Each successive approximate solution is given by $z_{k+1} = z_k - \alpha_k \nabla f(z_k)$ and therefore the correct answer should be $z_1 = \begin{pmatrix} 0 \\ \frac{3}{2}\eta \end{pmatrix}$ Consult paragraph 6.3.6 for more information.
- Question 4: Execute these steps again in order to determine z_2
 - a: x_2
 - * Answer: $\frac{3}{4}\eta$
 - * Feedback Correct: Good. See paragraph 6.3.2 for more information.
 - * Feedback Incorrect: Unfortunately, your answer is incorrect. It should be $\frac{3}{4}\eta$. See question 4b for why.
 - b: y_1
 - * Answer: $\frac{3}{2}\eta$
 - * Feedback Correct: Good. See paragraph 6.3.2 for more information.
 - * Feedback Incorrect: Unfortunately, your answer is incorrect. Each successive approximate solution is given by $z_{k+1} = z_k - \alpha_k \nabla f(z_k)$ and therefore the correct answer should be $z_2 = \begin{pmatrix} \frac{3}{4}\eta \\ \frac{3}{2}\eta \end{pmatrix}$ See paragraph 6.3.6 for more information.
- Question 5: Give β_1
 - Answer: $\frac{1}{4}$
 - Feedback Correct: Correct. See paragraph 6.3.6 for more information.
 - Feedback Incorrect: Incorrect. In the Conjugate Gradient method β_{k+1} is determined by $\frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}$. While we know g_0 and g_1 , β_1 turns out to be $\frac{1}{4}$. See paragraph 6.3.6 and example 6.8 for more information.

- Question 6: Give the new search direction s_1
 - a: s_{1_x}
 - * Answer: $\frac{3}{2}\eta$
 - * Feedback Correct: Good. See paragraph 6.3.6 for more information.
 - * Feedback Incorrect: Unfortunately, your answer is incorrect. It should be $\frac{3}{2}\eta$. See question 6b for why.
 - b: s_{1_y}
 - * Answer: $\frac{3}{4}\eta$
 - * Feedback Correct: Good. See paragraph 6.3.6 for more information.
 - * Feedback Incorrect: Unfortunately, your answer is incorrect. According to the Conjugate Gradient method $s_{k+1} = -g_{k+1} + \beta_{k+1}s_k$. Therefore $s_{k+1} = \begin{pmatrix} \frac{3}{2}\eta \\ \frac{3}{4}\eta \end{pmatrix}$. Consult paragraph 6.3.6 and example 6.8 for more information.
- Question 7: Give the new α_1^*
 - Answer: $\frac{3}{4}$
 - Feedback Correct: Correct. See paragraph 6.3.6 for more information.
 - Feedback Incorrect: Incorrect. As with the Steepest Descent method, $\alpha_1^* = \min_{\alpha} f(z_1^* - \alpha \nabla f(z_1^*))$
- Question 8: Give z_2^*
 - a: x_2^*
 - * Answer: $\frac{9}{8}\eta$
 - * Feedback Correct: Good. See paragraph 6.3.6 for more information.
 - * Feedback Incorrect: Unfortunately, your answer is incorrect. It should be $\frac{9}{8}\eta$. See question 8b for why.
 - b: y_2^*
 - * Answer: $\frac{33}{16}\eta$
 - * Feedback Correct: Good. See paragraph 6.3.6 for more information.
 - * Feedback Incorrect: Unfortunately, your answer is incorrect. According to the Conjugate Gradient method each successive approximate solution is given by $z_{k+1} = z_k + \alpha_k s_k$. Filling in z_1^* , α_1^* and s_1 , z_2^* turns out to be $\begin{pmatrix} \frac{9}{8}\eta \\ \frac{33}{16}\eta \end{pmatrix}$. Consult paragraph 6.3.6 and example 6.8 for more information.

7.3 interpoleren 3

Als meer gegevens dan twee punten bekend zijn, is het handig om daar gebruik van te maken. Een methode om dat te doen is de n-de orde Lagrange Interpolatie.

Algemeen:

$$L_n(x) = \sum_{k=0}^n f(x_k) L_{kn}(x) \quad (7.1)$$

hierin is:

$$L_{kn}(x) = \frac{(x - x_0) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n)}{(x_k - x_0) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)} \quad (7.2)$$

gegeven is de functie $f(x) = \frac{1}{x}$. steunpunten x_0, x_1 en x_2 zijn gegeven.

- Vraag 1: Bepaal het 2de graads lagrange interpolatiepolynoom van $f(x)$ en bepaal daar $f(3)$ mee
- Vraag 2: Bepaal een bovengrens voor de interpolatiefout
- Vraag 3: Bepaal een bovengrens voor de benaderingsfout $\sum_{k=0}^n |L_{kn}(x)|$

Gebruikte variabelen in Etude

Gegeven

- $x_0 = 2$
- $x_1 = 2,5$
- $x_2 = 4$

Gebruikt

- $x = 3$
- $f(x_0) = \frac{1}{x_0}$
- $f(x_1) = \frac{1}{x_1}$
- $f(x_2) = \frac{1}{x_2}$
- $f^{(4)}(x) = 24x_2^{-5}$

Vragen en antwoorden

- Vraag 1: Bepaal het 2de graads lagrange interpolatiepolynoom van $f(x)$ en bepaal daar $f(3)$ mee
 - Antwoord: $f(x_0) \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + f(x_1) \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + f(x_2) \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} = 0,325$
 - Geen tolerantie
 - Weegfactor: 1
- Vraag 2: Bepaal een bovengrens voor de interpolatiefout
 - Antwoord: $|(x-x_0)(x-x_1)(x-x_2) \frac{f^{(4)}(x)}{4!}| \approx 0,000488281$
 - Hint: De totale fout wordt samengesteld uit de interpolatiefout en de benaderingsfout. In het dictaat wordt daar een uitleg over gegeven.
 - Puntaf trek: 0
 - Geen tolerantie
 - Weegfactor: 1
- Vraag 3: Bepaal een bovengrens voor de benaderingsfout $\sum_{k=0}^n |L_{kn}(x)|$
 - Antwoord: 1,25
 - Geen tolerantie
 - Weegfactor: 1

Bibliography

- [1] Sofia Dopper. *'Het hoe en waarom van toetsen met Etude', Onderwijskundige handleiding bij versie 1.1.* October 2000.
- [2] Hoang Nguyen. *A guide to the online course 'Numerical Mathematics IN'.* May 2000.
- [3] Sebastiaan. *<http://ta.twi.tudelft.nl/WAGM/users/wilders/test>.*
- [4] Etude Project Team. *The 'Etude HTML-help' file of Etude.*

Appendix A

Tables of exercises

Sorted on subject, and number of exercise, estimated time in minutes for "Numerical Analysis WI", and sorted on subject for "Numerical Mathematics IN". In the last case, each subject contains precisely one test, and it should take a student approximately one hour to finish the tests.

chapter	exercise	short description	time needed
Rounding errors	1	floating point system	10
	2	floating point arithmetic	10
	3	floating point arithmetic	5
	4	floating point arithmetic	5
	5	number-representation	10
	6	floating point arithmetic	15
Interpolation	1	inverse/linear interpolation	20
	2	linear/2-nd lagrange	20
	3	lagrange/error	20
	4	taylor (+error!!)	30
	5	hermite	30
Differential equations	1	floating point + forward differentiation	15
	2a	difference quotients - forward	10
	2b	difference quotients - central	10
	2c	difference quotients - backward	10
	3	approximation error in function	10
	4	appr. first derivative	15
Initial value problems	5	appr. second derivative	10
	1	error in proof	15
	2	error in rk3	15
	3	definition num. stability	5
	4	error	20
	5	euler/heun	20
	6	euler/trapezium	20
	7	stability euler	5
	8	stability euler	30
	9	runga kutta	25
	10	stability euler	20
	11	adams-bashforth	15
	12	stability multistep-methods	30
	13	approximation integral	15
	14	Euler forward + interpolation	15
15	RK4	10	

Table A.1: Exercises for wi2091

chapter	exercise	short description	time needed
Boundary value problems	1	discretisation	40
	2	discretisation	15
	3	error estimation	20
	4	error in break (afbreekfout)	15
	5	polar coordinates	25
Numerical Integration	1	improper integral	20
	2	multiple integral	15
	3	zeropoint	30

Table A.2: Exercises for wi2092

subject	exercise	time needed
introduction	introduction exercise	20
computational linear algebra	Vertical beam	15
	LU decomposition	25
differential equations	Boundary Value Problem	55
nonlinear equations and optimization	Newton 1	40

Table A.3: Exercises for wi2025IN