Programming as a semiotic system to support physics students' construction of meaning: A pilot study

Kim Svensson, Urban Eriksson, Ann-Marie Pendrill and Lassana Ouattara

National Resource Centre for Physics Education, Department of Physics, Lund University, 221 00 Lund, Sweden

E-mail: `Kim.Svensson@fysik.lu.se`

Abstract.
Programming as a tool to be used for analysing and exploring physics in an educational setting offers an unprecedented opportunity for the students to create and explore their own semiotic resources. Students may use programming to create and explore different models of physical systems. In this study a small group of upper secondary education students participated in a workshop where they learned to program physics simulations and to create their own models to implement using the programming language Python. Results from the study shows that upper secondary education students are able to create their own models of physical systems and implement them into code. The implemented models were models of hanging cloth and heat diffusion. Results were obtained by analysing video and audio recordings of the students through the lens of social semiotics.

1. Introduction

In Sweden, a push to use programming outside of the programming class has been ongoing for years. From the summer of 2017 it is mandatory to have programming elements in mathematics from year one of elementary school [1]. It does not start with coding, but with algorithmic thinking and figuring out rules and models for solving problems and then transitions into implementation and validation. Digital resources and digital competency comprises a large part of the evolving educational system in Sweden. The focus is on dynamic representations, such as animations, simulations and interactive elements where the user can interact with the representation to observe changes and variations of different aspects [2].

Physics and mathematics are closely related to each other ever since Isaac Newton's formalisation of motion, giving natural philosophers another way to investigate natural phenomena. Programming has been used in physics and physics education for a long time, but it has rarely been used explicitly to give students a new tool that they can use outside of class [3, 4, 5]. We propose a more exploratory use of programming, where students define their own models and implementations, allowing them the freedom of variation [6], both in implementation and visualisation. This approach is believed to provide the student with meaning-making opportunities through the process of creating/implementing/testing their models.

In this paper we study a small group of six 17-18 years old students, with mixed genders, in upper secondary school with an interest in physics, to see how they approach physics problems and concepts within a programming setting. The students volunteered to be part of the study after learning about it during a physics-based event at Lund University. All students came from the same school and the same class, they were friends and had no problem discussing or speaking to each other. Special focus was given to analysing their understanding of the use of programming as a tool in physics. These students were chosen because of their interest in physics and their experience/lack of experience with programming. The aim of the exercise was to make the use of programming more explicit and to give the students a new tool to use for creating and investigating models of physical phenomena. One of the overarching aim of this study is to see what level of programming proficiency is needed to use programming in this manner. Half of the students had had some form of programming experience, equivalent to a basic course in programming, before participating in the workshop.

It is through the lens of *Social Semiotics*[7] that the different aspects of programming can be analysed and given a meaning potential. This work is both theoretical and empirical in the sense that, through Social Semiotics, the strengths of programming as a tool for meaning-making can be identified and then used as a lens to study how students use, or interact with, programming in a physics environment.

2. Theory

The Social Semiotics framework is the lens used to study and explain the students reactions and actions in this pilot study. Social Semiotics was started by Michael Halliday [7] and looked at language as the main communication method. It has since grown to encapsulate many different forms of communication methods and systems [8, 9, 10]. John Airey & Cedric Linder [8] defines social semiotics as: *the study of the development and reproduction of specialised systems of meaning making in particular sections of society.* This is also the definition used in the work presented in this paper.

2.1. Social Semiotics and Programming

Programming fits very well in the social semiotics framework thanks to its ability to reproduce and develop specialised systems. The production of specialised systems, through programming, can be seen as meaning-making functions. By creating and implementing models of physical phenomena, insights into the structure of the model, its dynamics, can be obtained. We believe that programming may help the student gain specific insight into the physical phenomena that is being simulated.

In programming, there is no need for explicit communication of the students idea to other students/teachers, but only to the computer. The interaction between student and computer becomes the disciplinary communication and meaning-making activity used to create/extract meaning. The interplay between student and computer allows the student to ask their program questions and analyse the answer, such as: "What if Hooke's law is $F = -kx^2$ instead of $F = -kx$?". These kind of questions and the ability for the student to, quickly and easily, observe the results allows the student to explore and experience variation of key aspects of the students own mental-model of the physical model.

A teacher or TA should help students realise the potential of asking the program questions and analysing the results. The teacher or TA should also help the students exploration through guided questions or "what if"-scenarios.

2.2. Semiotic Resource

A semiotic resource is defined by Linder et al. as: "Anything that is used to make meaning in a disciplinary relevant manner" [8]. which includes representations, tools and activities. Within the physics discipline we have many disciplinary-specific semiotic resources such as: particle detectors, right-hand rule, Feynman diagrams and many more. Each semiotic resource have some specific disciplinary meaning for the discipline. A student must learn to use, create, read, and analyse these resources if they are to become part of the physics discipline.

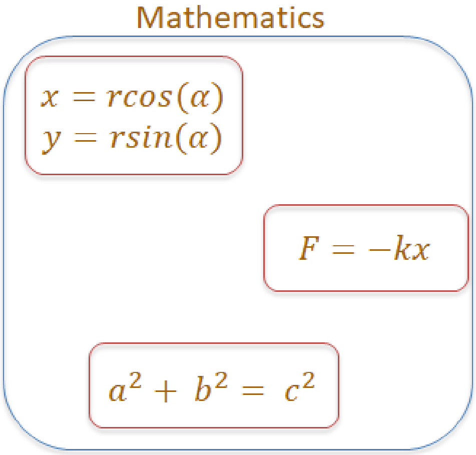

Figure 1. Different semiotic resources (red) relation to their semiotic system (blue). Each of the resources are different and are used in different ways in different scenarios, but they all belong to the same semiotic system.

2.3. Semiotic System

A semiotic system is defined by Linder et al. as: "Qualitatively different ways of communicating disciplinary relevant knowledge" [8]. See Fig 1 for a visual interpretation of the relation between semiotic systems and semiotic resources. Only when a semiotic system is used in a specialised case is a semiotic resource created/retrieved, as can be seen in Fig 1. A student must be able to extract relevant semiotic resources from different semiotic systems to solve different physics problems or to set up their own models and theories [11].

2.4. Transduction and Transductive Links

Within the physics discipline it is required to move between different semiotic resources and between semiotic systems, such as: going between a function and its graphical representation. The transformation between semiotic systems is called a *transduction.* Transductions are everywhere in communication: going from speech to gesture to drawing to speech and so forth. Each of these transductions is designed to move the focus from one semiotic resource to another with the purpose of highlighting some important aspect. Some semiotic systems aid the transduction from one semiotic system to another, these are called transductive links. Gestures are often used as transductive links between semiotic systems that are easier to extract disciplinary meaning from. Gestures allows the user to move quickly between different semiotic systems without losing focus of relevant aspects.

2.4.1. Programming as a Transductive Link Programming can be viewed as a transductive link between many different semiotic systems since it may take many different inputs and produce many different outputs. Due to the versatility of programming it can be described as a universal transductive link since it may move between many different semiotic systems.

The student constructing a program is responsible for the transduction taking place and is explicitly expressing the rules for the transduction. This provides the student control over the transduction, it allows the student to choose how to do the transduction and how to represent the new semiotic resource.

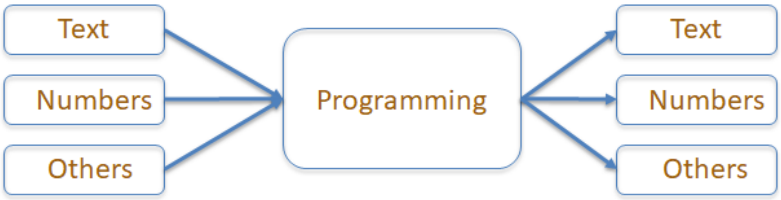

Figure 2. Programming is a universal transductive link since it can act as a stepping stone between many different semiotic systems. The student has full control over the transduction step and can choose the final products semiotic system at will.

2.5. Affordances and programming

An object or resource can have different affordances. An affordance is something a resource offers to an agent. Different resources offer different things and thus have different affordances. For example, a bottle of water affords 'Drinking' and 'Holding' and many other affordances. Within the social semiotic framework, two useful affordances have emerged:

Pedagogical Affordance: *"the aptness of a semiotic resource for the teaching and learning of some particular educational content".*

Disciplinary Affordance: *"the agreed meaning making functions that a semiotic resource fulfils for a particular disciplinary community".*

as defined by John Airey [12]. These affordances offer a way to study semiotic resources and say something qualitative about them. One of the goals of teaching would be to use resources with high pedagogical affordance and slowly transition into using resources with high disciplinary affordance. Tobias Fredlund [13] showed that different semiotic resources within the same semiotic system can have very different levels of pedagogical and disciplinary affordances.

Programming, through its power as a transductive link, allows the student to create their own semiotic resources. These new resources have their own disciplinary and pedagogical affordances which are directly, albeit unknowingly, controlled by the student. The student can thus create resources that have a balance of affordances that matches the students own ability to extract disciplinary meaning from it. Some students may spend extra time to create a resource with higher pedagogical affordance, to facilitate meaning-making, see Figure 3. Other students may extract meaning from highly abstract visualisations and instead increase the disciplinary affordance of the resource.

2.6. Coding, Visualisation and Interaction

The National Agency for Education in Sweden uses a broad definition of programming which includes: algorithmic thinking, creating dynamic representations (visualisations), producing coherent models and implementing them in code [2]. This view of programming is also the view taken in this research. We have decided to combine all of the different programming parts into the following aspects: coding, visualisation and interaction. Special focus is placed on the interplay between the three aspects and how they allow the student to open up dimensions of variation[6] of different disciplinary relevant aspects[13]. This approach is a step up from Orban et al. [14], where they specifically looked at coding and interaction. Together, coding,

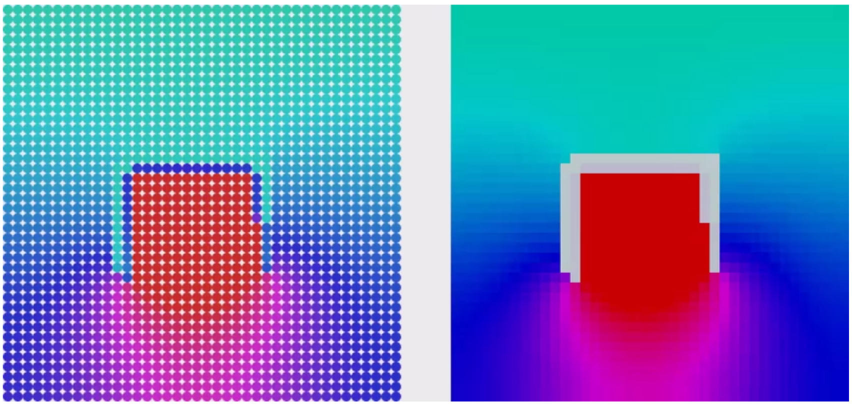

Figure 3. A standard visualisation is shown on the left. On the right is a visualisation created by a student who altered the original visualisation. The semiotic resource on the right has higher pedagogical affordance since it allows for easier extraction of relevant disciplinary information whereas the information in both resources are the same.

visualisation and interaction allow the student to code their own model, visualise it and interact with it to explore different scenarios.

Dimensions of Variation is a term from the Variation Theory of Learning [6], where learning occurs when the learner discerns variation in the object of learning. By varying the aspect the object of learning with respect to a static background, that aspect is highlighted and discerned, learning takes place. This is the main tenet behind using coding, visualisation and interaction as a package since it provides the student with the ability to vary all aspects of the simulation and its representation. It is hoped that the student realises the potential of programming and varies whatever aspect they are interested in to gain new insights into the physical phenomena they are exploring.

3. Research Questions

This research is designed to answer the questions below, but also to see if there are any programming related learning scenarios occurring during the workshop. An example of such a scenario can be seen in section 5.3 where students investigate predictions of their models using the real world, but also using an interactive simulation.

- How can Social Semiotics be used to describe learning physics using programming as a semiotic system, based on the reported experiences by the students?
- What does the participant students report about using programming to explore and learn physics?

4. Methodology

A workshop was created with the purpose of unlocking the potential of programming as a tool to understand physical phenomena for the students. To do this, the workshop relied heavily on variation theory to highlight different aspects of programming. The students were encouraged to vary different aspects of the code such as: the interaction between particles, the visualisation and the interaction with the simulation. Through these variations the student could highlight different aspects that caught their interest. The workshop was especially designed to be versatile with respect to the different kinds of physical phenomena that can be simulated. The different sections of the workshop and their purpose for unlocking the potential of programming can be seen below. Each session was two hours long and was video and audio recorded from several angles. During the sessions, smaller interviews were done to capture the students thinking about kinematics and attempts were made to observe changes to their thinking through video recordings. Each session was self contained and the content did not bleed over into the other session.

Each session was set out as a Code-Along, where the students coded along with the lecturer. At each step of the implementation, the students were encouraged to test and vary things in the code to understand how it behaved. The programming language used was Python using the Processing IDE, which allows for easy visualisation, interaction and quick iterative coding.

The sessions took place over six weeks in May and June in 2018. The first four sessions took place during the four first week of the interval and the last session, the interview session, took place during two weeks after. The gap was due to a national holiday happening on the same day and it was not expected that the participants would participate on that day. Each session took place on Saturdays before noon.

- Session 1 was designed to make the students familiar with the programming environment and introduce key aspects of the engine such as the updating loop and the ability to draw shapes in a window. The students coded along with the lecturer and constructed a circle that had its position updated between each frame, creating an animation of a ball moving. The ball was given velocity and acceleration which in turn allowed the ball to showcase ballistic motion in the window.

- Session 2 focused on taking what was created in session 1 and combining it all into a Particle-class. The Particle-class can update its position using an Euler-Cromer [15] integrator, it can show itself in the window and it can feel forces. During each timestep, the particle calculates new accelerations from the forces it feels, it then uses the acceleration to update its velocity and position. The students coded along with the lecturer and was encouraged to vary different attributes of the particles to see that particles with different values can be created but they all follow the same code. In session 2, the notion of interaction between different particles was introduced and the interaction was limited to forces.

- Session 3 was divided into two parts. First, the students were divided into groups of three and each group was tasked with coming up a model to simulate hanging cloth. The groups had thirty minutes to come up with a model and they had access to a interactive simulation of hanging cloth but they had not access to the code for the simulation. The group then presented their models and discussed. In the second part of the session, a hanging cloth simulation was created by the lecturer with the students coding along. The dire

- Session 4 instructed the students to come up with, and implement, their own models for heat diffusion and then compare their models with the textbook formula for heat diffusion. The lecturer helped with programming questions and advice regarding potential pitfalls. The of this approach was to create an environment where data, about the students ability to formulate their own models and testing them, could be obtained. The students had to figure out how to represent thermal energy and heat between different particles, how to update the attributes and how to visualise it. See Fig. 3.

- Session 5 was an interview session consisting of individual interviews as well as a group interview. Questions were designed to highlight their vision of programming as a tool to investigate physics, what they can use it for and what they want to use it for in the future.

In essence: The workshop was designed to create a solver for ordinary second-order differential equations such as the kinematic equations from Newtons laws of motion. The setup of the solver allows the student to easily access different parts of the interaction and change them to create new models and simulations. The easy interpretation of the implementation of the solver is created to place the attention of the student on other aspects of the program.

4.1. Analysis

To analyse the video interviews and workshop sessions a qualitative analysis method is used. By transcribing the videos, with a special focus on disciplinary relevant events, categories about the students conceptual understanding and its relationship to programming can be inferred. The student's relationship to programming and their ideas about future use of programming can be seen from the categories.

The categories have yet to be finalised but will follow from the theory of Phenomenography [16], where a phenomenon is experienced by agents and their experience about the phenomenon is categorised into qualitatively different chunks.

5. Results

All students managed to follow along during the sessions regardless of their programming experience prior to taking part in the workshop. Three of the six students had some prior knowledge of programming and three of the students had close to no experience of programming.

When asked about what programming offers or differs from a normal physics education situation two of them said:

Student 1

> [Programming] has given me, that I can take a phenomenon or problem or ... **anything** ... from physics. Implement it and visualise it and ... **figure out answers** and **see if I've done it correctly.**

Student 2

> ... something else I thought about... **that programming gives another angle on the physics**. Often, you have exercises you have to solve, and that is the case in programming as well, if we would simulate a pendulum, but its much more... vague. There are different ways to do it. **Instead of just solving something, you create.**

These answers shows that the students have seen the potential and the use of programming in physics and how they may apply programming to investigate physics. The second student also separated programming from normal exercises within physics education in the sense that in programming you do not solve a particular problem, but you create a whole system capable of solving many exercises.

5.1. Programming Proficiency

All students managed to follow along in all the sessions. The students with prior programming knowledge began using programming at home or in school for smaller projects. The students all agreed that an introductory course in basic programming would be good, and they also said that they do not think anything more were needed to use programming in the way they had in the sessions.

5.2. Dimensions of Variation

One student created a small simulation of a Frisbee and said that the direction of forces had become much more important than before. The student had had to think much more carefully about the directions of forces than they had before this small project. A new dimension of variation had opened up when implementing the model for the Frisbee, namely that the direction of forces can change.

Another student did some programming ahead of time, because they realised where the session was headed and realised that they could implement it themselves. At the end of the session it turned out that they had written a correct solution, but used a different approach than what was shown by the lecturer. The student then asked if it was acceptable to write different solutions. [The answer is, of course, Yes!]. This opened up a dimension of variation for this student: the ability to vary the solution, or to vary the approach or implementation of the idea.

A third student varied the visualisation during the last session to make it more visually appealing, see Fig 3. This opened up yet another dimension of variation: namely the ability to represent data in different ways to highlight different aspects. During the second session, the colour of particles where coupled to different aspects such as its position, velocity and acceleration, this coupling of the visual to variables gave the student a way to showcase different aspects that they were focused on.

5.3. Making and testing predictions

During the third session, the students were tasked with constructing a model for hanging cloth. During this task, they had access to an interactive simulation of a hanging cloth to act as inspiration.

The interactive simulation took on an unexpected role of being the validation medium for their models. Before the students model were implemented, the students made predictions about the behaviour of their model and came up with scenarios to test using the interactive simulation.

Student 4	”Can you throw the curtain above...?”
Student 1	”Yes, but it can go down. Then **it [their model] does not work** if we can pull it down.”

The student used the interactive nature of the simulation to test their prediction and to find new aspects that they had not thought about. These interactions made them rethink or adapt their own model to fit with the interactive simulations behaviour.

Another example of implementing models and making predictions come from the fourth session as Student 5 manages to get their simulation to work:

Student 5	”YES! **It does what I want!**” [Student 5 puts their hands up the air, they also stand up and clap their hands]

Apart from being a celebratory occasion, this also showcases that Student 5 had expectations of their model. They understood what they wanted from the model and could visualise the behaviour of the model in their head. When the implemented model was visualised Student 5 could immediately identify that they had implemented it correctly. Student 5 thus used the visualisation of the simulation to validate the behaviour of the simulation versus how they expected in to work.

5.4. Getting the full picture

Student 4 realised that the dynamics of the cloth simulation would drop out as long as the base interaction between particles were implemented correctly. As one group of students were discussing their model, the problem with interactivity (dragging the mouse across the hanging cloth) came up:

Student 1	”Shall we start wondering about what happens when we throw in a ball?” [Student 1 picks up an eraser and moves it towards the drawing on the whiteboard.]
Student 4	”...actually, I think if we just have a **good simulation at the start...**”

Student 4 implies that the simulation will handle the interaction with the ball without problems if the base of the simulation is good. This insight is true and goes even deeper, student 4 realised that the phenomenons they had observed in the interactive simulation or the phenomenons they expected to observe was the result of the basic interactions between individual particles.

Another student, Student 3, also realised that the model used to implement the physics can be represented in different ways. Student 3 realised that the particles used in the simulation are just the ”physical background” which handles the simulation and that they can be represented in various ways on the screen depending on what they wish to highlight.

Student 5	”It is the particle inside that is good to have for the distance...” [Student 5 gesticulates and pulls out a thread (in the air).]
Student 3	”No, we have to have something, the particles are only there for the thing, then we we may place squares over them to make it look nice. **These round things are only there for the physical background.**”

6. Discussion

Social Semiotics, and its semiotic resources and semiotic systems, provides well adapted tools to investigate programming as a tool for learning within physics education. By identifying the interaction between student and computer as communication, the tools of social semiotics can be applied such as variation theory and transduction. Programming is especially well suited to exploit variation theory due to its well defined structure, by changing single variables or small bits of code and observing the effects of this variation, concepts or connections can be discerned which would be hard to discern from just a formula. Programming also allows student to make well defined transductions between different semiotic systems. The transductions require the student to unpack the semiotic resource they are moving from one semiotic system to another. The unpacking of semiotic resource reveals the inner structure of the semiotic resource that is being transduced, allowing for discernment of its various important parts.

Within this study, it has been shown that students with interest can use programming to create, implement and visualise their own models of physical phenomena. However, they all agree that a basic knowledge of programming would help them to easier implement their own ideas or models. The students also said that implementing physical models into code highlighted different aspects, that they had not focused on when solving normal physics exercises, such as the direction of forces when implementing a model of a Frisbee. Programming also allowed them to rapidly change their models based on the visual feedback from the simulation. This created a feedback-loop where the students could iterate and test different aspects of their model using variation of different aspects.

7. Conclusions

Students with an interest in physics and programming were able to see the potential of programming as tool to be used when learning/investigating physics and physical phenomena. Some programming knowledge were needed to apply the programming to their own ideas, but only the basics of programming knowledge such as: If-statements, For-loops, Variables, Lists/Arrays. Knowledge of classes help, but is not needed to produce the simulations.

Programming allows the student to open up many different Dimensions of Variation to explore. Since the student is the programmer, they choose the dimensions of variation themselves and thus focus on the aspect they wish to understand.

Students can use the interactive nature of the simulations to test predictions and to construct new scenarios were they can not predict the results.

Programming fits well into the Social Semiotics framework and programming is a powerful tool when looked at as a transductive link between semiotic systems where students are allowed to create their own representations through transduction from one semiotic system to another.

8. Acknowledgements

The authors would like to thank Helen Edström and Moa Eriksson for comments on this article. We would also like to thank the Uppsala Physics Education Research group for all their comments and useful discussions. Special thanks to Vattenhallen Science Center in Lund for providing the locales.

References

[1] Regeringskansliet. (2017). ”Stärkt digital kompetens i skolans styrdokument”.

[2] Skolverket (National Agency for Education). (2017). ”Programmering i matematik och teknik i grundskolan.”

[3] Winograd, T., & Flores, F. (1986). Using Computers. Computer, 34, 551–562. https://doi.org/10.1063/1.881202

[4] Redish, E. F., & Wilson, J. M. (1993). Student programming in the introductory physics course—Muppet. American Journal of Physics, 61(3), 222–232.

[5] Caballero, M. D., Obsniuk, M. J., & Irving, P. W. (2017). Teaching Computation in Introductory Physics using Complex Problems. Retrieved from http://arxiv.org/abs/1709.05493

[6] Marton, F. (2015). Necessary conditions of learning. New York: Routledge.

[7] Halliday, M. A. K. (1978). Language as Social Semiotic. London: Edward Arnold.

[8] Airey, J., & Linder, C. (2017). Social Semiotics in University Physics Education. In D. F. Treagust, R. Duit, & H. E. Fischer (Eds.), Multiple Representations in Physics Education (pp. 95-122). Cham: Springer International Publishing.

[9] Fredlund, T. (2015). Using a Social Semiotic Perspective to Inform the Teaching and Learning of Physics. Uppsala: Acta Universitatis Upsaliensis.

[10] Fredlund, T., Linder, C., & Airey, J. (2015). A social semiotic approach to identifying critical aspects. International Journal for Lesson and Learning Studies, 4(3), 302–316. https://doi.org/10.1108/IJLLS-01-2015-0005

[11] Eriksson, U. (2014). Reading the sky: From starspots to spotting stars.

[12] Airey, J. Social Semiotics in Higher Education. In SACF Singapore-Sweden Excellence Seminars, Swedish Foundation for International Cooperation in Research in Higher Education (STINT) , 2015.

[13] Fredlund, T., Linder, C., Airey, J., & Linder, A. (2014). Unpacking physics representations: Towards an appreciation of disciplinary affordance. Physical Review Special Topics - Physics Education Research, 10(2). https://doi.org/10.1103/PhysRevSTPER.10.020129

[14] Orban, C., Teeling-Smith, R. M., Smith, J. R. H., & Porter, C. D. (2018). A Hybrid Approach for Using Programming Exercises in Introductory Physics, 831. https://doi.org/10.1119/1.5058449

[15] A. Cromer, Stable solutions using the Euler Approximation, American Journal of Physics, 49, 455 (1981),

[16] Marton, F. (1981). Phenomenography - describing world around us conceptions. Instructional Science, 10, 177–200. https://doi.org/10.1007/BF00132516