The computational structure of life cycle assessment

Reinout Heijungs
Sangwon Suh

Centre of Environmental Science
Leiden University

To be published at
Kluwer Academic Publishers
Dordrecht, 2002

Preface

This books presents a complete overview of the computational aspects of life cycle assessment (LCA). Many books and articles have been written on LCA, including theoretical treatments of the entire concept, practical guidebooks to apply the technique, and concrete case studies in which LCA is applied to support decision-making with respect to environmental aspects of product alternatives. However, a good discussion of the computational structure of LCA is lacking. Knowledge is only partially documented, and what is documented is fragmented over diverse publications with mutual inconsistencies in approach, terminology and notation.

The book is the result of several years of research, along with the teaching of LCA at university classes and, not unimportantly, the development of software for LCA. This software has been designed to support the education of LCA, but it has been applied in real-world case studies as well. The name of the software is CMLCA, which is an abbreviation of Chain Management by Life Cycle Assessment. This program can easily be used to reanalyse and further explore the ideas that are outlined in this book. Another important source for this book relates to the work involved in connecting input-output analysis (IOA) to LCA. Software for this – MIET, an abbreviation of Missing Inventory Estimation Tool – is also available. Some of the basic routines have been implemented in Matlab script as well. All three pieces of software can be accessed, free of charge, through http://www.leidenuniv.nl/cml/ssp/software.html.

In developing the ideas that are written in this book, we have benefited from discussions during the last few years with Jeroen Guinée, Gjalt Huppes, René Kleijn and Ruben Huele at the Centre of Environmental Science, Leiden University, Rolf Frischknecht at ESU-services, ETH Zürich, Mark Huijbregts, formerly at the Interfaculty Department of Environmental Science, University of Amsterdam, now at the Department of Environmental Science, Nijmegen University and Wang Hongtao at Sichuan University. Igor Nikolić provided support in discovering the advanced features of type-

setting with LaTeX. The actual text, including possible omissions and errors, however, is our responsibility.

Contents

Chapter 1

Introduction

This chapter introduces the aim of this book and motivates the importance of its topic. It does so in relation to a brief introduction of life cycle assessment (LCA), in which the various types of activities are outlined as well. Finally, the structure of the book is presented, along with a reading guide.

1.1 Purpose of the book

1.1.1 Aim

This book presents and discusses the computational structure of life cycle assessment. Under the computational structure, we will capture the arithmetical rules that are involved in carrying out an LCA study. However, this book is not a book with computational recipes only. Two other aspects receive a large emphasis as well. These are the background of the computational recipes, including argumentations and proofs, even though sometimes heuristically, references to related mathematical rules, and aspects that relate to the numerical implementation of the computational recipes. For this latter, the book will not provide computer source codes, but it will concentrate on the algorithmic aspects, even though some example pieces of Matlab code are given in Appendix C. Thus the computational structure is understood here to cover the mathematical structure as well as the algorithmic structure.

The computational structure will be formulated in terms of explicit mathematical equations. It will become apparent that use of matrix algebra provides an elegant, concise and powerful formalism. One should note that the term 'matrix' in this book refers to a rigid mathematical concept (see Appendix A), that is defined in a linear space and for which operations such as multiplication, transposition and inversion are defined. Thus, Graedel's (1998, p.100) concept of matrix as a table of 5×5 cells in which the user is supposed to enter an ordinal score between 0 ("highest impact") and 4 ("lowest impact") is outside the scope of the present book.

It will be assumed that the reader has a basic knowledge of the principles, framework and terminology of LCA. Useful texts at varying levels of depth are provided by Lindfors *et al.* (1995), Curran (1996), Weidema (1997), Jensen *et al.* (1997), Hauschild & Wenzel (1998), Wenzel *et al.* (1998), UNEP (1999), Guinée *et al.* (2002), and others. However, a short overview of the basic elements of LCA is discussed in the next section. We also will, as much as reasonably possible, adhere to the ISO-standards for LCA (ISO, 1997, 1998, 2000). At certain points, departures will be necessary, and at many places, new concepts must be introduced. When appropriate, such cases will be argued.

Throughout this book, it will be assumed that data availability is not a problem. In fact, the efforts and measurement, modeling and estimation techniques that are needed to obtain data is not discussed in this book. The central theme is how the data, once available, should be processed and combined to complete an LCA study. In the first few chapters, it will moreover be assumed that data are known exactly. This will allow us to present the basic structure in terms of deterministic equations. Chapter 6 discusses extensively the topic of perturbation theory, which includes the statistical processing of stochastic data.

1.1.2 Motivation

The main motivation for writing this book is that the computational structure is an important topic for which no reference book is available. Below, we first seek to explain that indeed the topic is underemphasised, and then will demonstrate its importance.

It is a remarkable fact that there is a large number of guidebooks for applying the LCA technique, but that the computational structure of LCA is hardly addressed in these books. To some extent, this is understandable: a person charged with carrying out an LCA study needs guidelines on which data to collect, which choices to make, and how to report assumptions and results. For the calculations, he or she will rely on LCA software, of which there is a large choice on the market (Siegenthaler *et al.*, 1997). But this alleged lack of direct utility is not a decisive argument, since most guidebooks on LCA discuss the backgrounds of, say, models for ecotoxicity, even though these models are not used in an LCA, because it is only the tabulated characterisation factors that are derived from such models that are used. So, lack of direct utility when executing an LCA is not a valid reason for excluding material on the computational structure in guidebooks for LCA.

A further remarkable fact is that the computational structure is by and large overlooked by the theoretical literature on LCA as well. The equation which forms the basis for almost the entire book is

$$\mathbf{s} = \mathbf{A}^{-1}\mathbf{f} \tag{1.1}$$

in which $\mathbf{f}$ is the final demand vector, $\mathbf{A}$ is the technology matrix (and $\mathbf{A}^{-1}$ its inverse), and $\mathbf{s}$ is the scaling vector; see Sections 2.1 and 2.2 for a full explanation. In the standard literature on LCA, this equation, as well as the terms final demand vector, technology matrix and scaling vector are missing entirely. And the few sources in which the computational structure is discussed are used in a rather limited way. An example may illustrate this. In 1994, one of the authors published a paper (Heijungs, 1994) that explicitly discussed some important elements of the computational structure of LCA. It introduced a matrix formalism towards the inventory analysis, and it gave a small example system with only four unit processes with a feedback loop that needed a matrix approach for a reliable solution. Six years later, in 2000, virtually all commercially available LCA programs were still unable to reproduce these results. Some of the programs refused to perform the calculation, others gave a totally wrong answer, and still others gave results that at best approximated the exact solution.

One might think that the computational structure of LCA is a too obvious issue to discuss in scientific publications. This is suggested by the formulation in the ISO-standard for inventory analysis: “Based on the flow chart and system boundaries, unit processes are interconnected to allow calculations on the complete system. This is accomplished by normalising the flows of all unit processes in the system to the functional unit. The calculation should result in all system input and output data being referenced to the functional unit.” (ISO (1998, p.10)). The forerunner of the ISO-standard, SETAC’s Code of Practice (Consoli *et al.* (1993)), provides some more information, but is still far from being exact and operational on that topic. Fecker (1992, p.4) writes in a book with the promising title *How to calculate an ecological balance?* that “the process parameters are multiplied with the corresponding factor by which the process participates in the system." In this, he is one of the few authors that explicitly introduce the concept of scaling factors, but he does not provide a method to obtain them in a concrete situation. The report of SETAC's Working Group on Inventory Enhancement (Clift *et al.* (1998)) ignores the topic entirely. Another famous SETAC-publication (Fava *et al.* (1991, p.15)) is more explicit: "The calculation procedure is relatively straightforward ... The calculations can usually be performed by common spreadsheet software on a personal computer." This is, however, no longer true. As we will see in subsequent chapters, the theory involves concepts such as linear spaces, singular value decomposition, the pseudoinverse of a matrix, and the condition number of a matrix. Of course, there are a few texts in which the topic is addressed. For an overview, see Section 1.3.

It is the authors' experience that a good knowledge of the computational structure of LCA is important for several reasons:

- it is a prerequisite in the construction of a method that really can claim to have scientific validity;
- it is useful to gain an understanding of the logic of LCA in a university course;
- it guides the design and implementation of reliable LCA software (so proves the aforementioned failure of most commercial programs to deal with system with feedback loops);
- it may shed lead new light on established topics, such as co-product allocation;
- it enables a further exploration of advanced topics, such as uncertainty analysis.

In conclusion, the aim of this book is to provide a comprehensive description of the present state of scientific knowledge of the computational structure of LCA.

1.2 Elements of LCA

The general ISO 14040 standard (ISO, 1997, p.2) defines LCA as the "compilation and evaluation of the inputs, outputs and the environmental impacts of a product system throughout its life cycle." The LCA technique is structured along a framework with a number of steps or activities in each of these steps. There are four phases:

- goal and scope definition;
- inventory analysis;
- impact assessment;
- interpretation.

A short summary of these phases follows.

Goal and scope definition deals with the clear and unambiguous formulation of the research question and the intended application of the answer that the LCA study is supposed to provide. Important elements of the goal and scope definition are the choice of the functional unit, the selection of product alternatives to be analysed, and the definition of the reference flows for each of the alternative systems.

The inventory analysis is concerned with the construction of these product systems. These systems are composed of unit processes, like industrial production, household consumption, waste treatment, transportation and so on. System boundaries and flow charts of linked unit processes are drawn for each alternative product system, and quantitative data as well as qualitative data for representativeness, etc. are collected during this phase. For those unit processes that are multifunctional, *i.e.* that provide more than one function, an allocation step is made. A final step of the inventory analysis is the aggregation of the emissions of chemicals and the extractions of natural resources over the entire product system, in such a way that a quantitative match with the system's reference flow is achieved. The final table of these aggregated emissions and extracted is referred to as the inventory table.

The result of the inventory analysis is often a long list with disparate entries, such as carbon dioxide, nitrogen oxides, chloromethane and mercury. The impact assessment aims to convert and aggregate these into environmentally relevant items. In particular, we mention here the step of characterisation, in which the inventory results are transformed into a number of contributions to environmental impact categories, such as global warming, acidification, and ecotoxicity. We also mention the optional normalisation in which the characterisation results are related to a reference value, such as the annual global extent of these impacts. We finally mention the weighting, in which priority weights are assigned to the characterisation or normalisation results, and which may result into one final score for each alternative product system.

During the course of the LCA, many choices and assumptions must be made. Moreover, uncertainty may be introduced with every data item. The interpretation phase deals with the meaning and robustness of the information obtained and processed in the previous phases. The interpretation may include comparisons with previously published LCA studies on similar products, uncertainty and sensitivity analyses, data checks, external comments, and much more. It is also the place in which a final judgement and decision is outspoken.

In using the LCA technique for carrying out an LCA study, one may distinguish several types of activities.

- There are activities, related to the design of the system, the collection of data, the making of assumptions and choices, and so on. This, for instance, includes steps like the drawing of system boundaries, the collection of process data, the choice of allocation method, and the choice of an impact assessment method.
- There are computational activities, related to transforming or combining data items into a certain result. For instance, emission data are related to the functional unit, aggregated over all unit processes in the system, multiplied with appropriate characterisation factors, and so on.
- There are activities that relate to the procedural embedding of an LCA project. Depending on the topic of study and the intended application, different stakeholders may be involved in certain ways. For certain applications, critical review by an independent expert is essential.
- There are activities, related to the planning of the LCA. For instance, one can start with a small-size LCA, to explore the potentials and bottlenecks, and then to reiterate the steps in a more complete way. Uncertainty analyses can give rise to further reiterations.
- There are activities, related to the reporting of an LCA. All types of requirements on what to report and how to report can be imposed to obtain transparent and reproducible reports.

The ISO-standards for LCA do not clearly separate these different types of activities. However, emphasis is, apart from the presentation of framework and the definition of terms, mainly on procedural embedding and reporting. Most importantly for this book, the ISO-standards for LCA do not cover the computational structure. One can easily confirm this by observing the absence of mathematical equations. This leaves a large degree of freedom for the present book. Many new technical terms will be introduced; examples are technology matrix and final demand vector. In fact, besides a presentation of the computational structure, this book aims to propose a standard nomenclature for a number of concepts; see Appendix B. Notation is also free, as there are no reserved symbols in the LCA-community (except perhaps one older proposal by Heijungs & Hofstetter (1995)). Throughout this book a consistent notation will be used. It is summarised in Appendix B as well. A number of new non-mathematical terms are introduced; we mention in particular hollow processes (Section 3.1), brands of economic flows (Section 3.4) and sleeping processes (Section 3.8). Finally, for a few terms that do occur in the ISO-standards, we have found reason to introduce a different meaning; here we mention reference flows (Section 3.7.2) and grouping (Section 8.1.6).

As already indicated, this book discusses the computational structure of LCA, without reference to the procedural embedding and without reference to the planning aspects. This means, for instance, that this book may well describe the mathematics of comparing product alternatives on the basis of a weighting procedure, while the ISO-standards state that such an activity is not appropriate. The point is that ISO's reluctance derives from procedural grounds, while the mathematics is in itself without problems. The mathematics remains valid even when someone decides to operate outside the ISO-framework, or when the ISO-standards are changed in this respect.

1.3 Background of the book

Most method-oriented texts on LCA focus on formulating guidelines (*cf.* Guinée *et al.* (2002)). In addition to that, there are many articles and reports in which specific topics are discussed, such as models for assessing impacts of acidification or data quality. There are only few texts in which the computational structure is discussed. To the extent that they are relevant for the present book, their material has been included. Important references in this respect include Projektgemeinschaft Lebenswegbilanzen (1991), Heijungs *et al.* (1992), Möller (1992), Frischknecht *et al.* (1993), Heijungs (1994), Schmidt & Schorb (1995), Heijungs (1996), Heijungs (1997), Heijungs & Frischknecht (1998), Huele & van den Berg (1998) and Heijungs & Kleijn (2001).

In addition to that, other references that are relevant throughout the text are on linear algebra. Many texts, at various levels of sophistication and rigour, are available. Apostol (1969), Stewart (1973), Gentle (1997) and Harville (1997) provide good and accessible reviews. Albert (1972), Jennings & McKeown (1977) and Golub & Van Loan (1996) provide more specialised texts at an advanced level.

Finally, the topic of numerical analysis and computer algorithms is treated in many books, some emphasising the theoretical aspect and others providing easy-to-use computer codes. We have made use of the books by Jennings & McKeown (1977), Hamming (1986), Thisted (1988), Press *et al.* (1992) and Cheney & Kincaid (1999).

1.4 Structure of the book

1.4.1 Outline

This book discusses the computational structure of LCA. Much of the discussion will be directed to the computational aspects of inventory analysis. While many books on LCA would be structured along four core chapters, each of them dealing with one single phase of the LCA framework, this book presents the material in a different way. There is no chapter on goal and scope definition (although the reference flow is introduced in Section 2.1), and impact assessment and interpretation are treated in one single chapter (8).

Chapter 2 presents the basic computational model for inventory analysis. It introduces the representation of unit processes, economic flows and environmental flows, and it presents and solves the inventory problem: how to obtain the environmental flows associated with a functional unit.

Chapter 3 further develops the inventory analysis. We will see that the basic model falls short in many practical cases. This failure has to do with various complications that distort the ideal required for the basic model. These complications are, most importantly, cut-off and multifunctionality, the second type of complication giving rise to the allocation problem. This chapter also explores how the basic model works for a number of difficult situations.

Chapter 4 discusses advanced topics of the inventory analysis and is mainly intended for discussing very specific points. This chapter may be omitted without affecting the readability of the subsequent chapters. The same applies to Chapter 5 which ties the discussion to input-output analysis, a tool that is familiar in economics for more than sixty years and that shares certain features with LCA.

In Chapter 6, we abandon the idea of point estimates of data, and develop how the computational rules can be used to statistically deal with uncertainty. Both an analytical and a numerical treatment are included.

Chapter 7 discusses analytical explorations of the data on the basis of theoretical considerations. This leads to summary measures of the structure of the data and their dependencies.

All computational aspects beyond the inventory analysis are discussed in Chapter 8: impact assessment and interpretation.

Chapter 9 briefly explores a more general theory for LCA, in which the usual simplifying assumptions of linearity and steady state are abandoned.

A final chapter (10) is devoted to more information-technical topics: algorithms for the inversion of a matrix under special conditions, memory requirements, and so on.

Some sections contain special topics that can be omitted without distorting the readability of subsequent text. These sections are indicated with an asterisk (*).

The book assumes that the reader has a basic knowledge of matrix notation and manipulation. A concise review of matrix algebra is provided as an appendix (A). The first few chapters require a smaller background in mathematics than the chapters later on do. Especially Chapter 2, which discusses the basics, has been written in a more accessible way, to make sure that the basics can be understood by a wide audience. Chapter 3 is already more involved, and especially Chapter 6 requires quite some background.

1.4.2 Notation

In this book, a consistent notation will be employed throughout. Appendix B gives an overview of the most important symbols and the name of the concepts they represent. Furthermore, we have adhered to the convention that italic letters (like x) indicate scalars, that roman bold lowercase letters (like $\mathbf{x}$) indicate vectors and that roman bold uppercase letters (like $\mathbf{X}$) indicate matrices. A superscript T indicates the transpose of a vector or matrix, a superscript -1 the inverse of a matrix, a superscript $+$ the pseudoinverse of a matrix; see Appendix A for the definitions of these concepts. Other symbols that are placed after or on top of a symbol, like primes (x'), hats ($\hat{x}$), dots ($\dot{x}$) and tildes ($\tilde{x}$), are used to refer to another variable, and their meaning differs per occurrence. Sometimes, we will write a row vector for a column vector to save space, *e.g.* writing

$\mathbf{x} = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}^{\mathrm{T}}$ instead of $\mathbf{x} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$.