Computational challenges in huge LCA and EEIOA systems

Reinout Heijungs* & Arjan de Koning

Abstract
Life cycle assessment (LCA) and environmentally extended input-output analysis (EEIOA) are tools for supporting policies on sustainable production and consumption. Both the LCA and EEIOA model rely on manipulations with matrices, filled with data on technologies, emissions, and environmental impacts. Developments in LCA databases, e.g. at JRC-IES, and in EEIOA databases, e.g. at Eurostat, lead to the creation of ever-increasing datasets. The handling of such datasets in terms of retrieval, storage and processing urges for a need to rethink the computational paradigm. This paper sketches some of the developments.

1. Introduction
The history of LCA can be approached from many sides. For instance, one can study a shift in the topics LCA has been applied to, from packaging in the 1990s to consumer products around 2000 and onward to resource strategies in the present time. Or one can study it from the involvement of different types of actors during the different periods: companies, NGOs, national governments, international bodies. This paper will approach the history of LCA from the computational perspective. In doing so, we must distinguish three elements:

- the computational framework, i.e. the mathematics and its coding into software;
- the computational system, i.e. the hardware and the software;
- the computational content, i.e. the data.

All of these have seen tremendous developments during the past 20 years. The three fields are in principle separate, but interesting links can be shown to exist.

2. Developments of the computational framework
The early texts hardly discussed the computational framework of LCA. Texts on inventory analysis typically provided no formulas, but spent pages on discussing how to collect data, how to ensure and establish data quality, how to set system boundaries, and how to deal with allocation. Heijungs & Suh (2006) review the history of the computational framework for LCA, dating the seminal attempts to the grey literature in 1991-1992, and the first peer-reviewed paper in 1994.

The developments in the computational framework can be characterized as follows:

- the use of matrix algebra has led to a concise and flexible formulation of the computational steps involved (Heijungs & Suh, 2002);
- the link with the literature on matrix theory has enriched the analytical power of LCA, for instance by linking it to matrix perturbation theory (Heijungs, 2002).

Altogether, the explicit formulation of the computational can be recognized as a step in maturing the science of LCA.

1 Institute of Environmental Sciences (CML), Leiden University, PO Box 9518, 2300 RA Leiden, The Netherlands
Providing explicit formulas is one thing, implementing these into software is another thing. Although “scientific computation” and “scientific programming” has been a métier for many decades, its lessons for LCA have so far been neglected in the literature.

3. Developments of the computational system

The history of the computer is outside the scope of this paper, but some points should be mentioned. First and foremost, it goes without saying that present-day PCs facilitate the work of the scientist and consultant. As PCs have increased performance in terms of speed, memory capacity, and user-interaction, their use for LCA has become indispensible. There are many dedicated programs for doing LCA (JRC-IES, 2011). Some of these have been designed for special fields of application (like buildings, waste, or design), while others are general purpose programs. Some come with data (see below), while others are “empty”. Altogether there is a wide choice for anyone, from free to expensive, from simple to highly advanced, from Windows to Linux, etc. We have no picture of the performance of these tools in terms of the system size. It would be a nice addition if a standard test system (say, of 2000 unit processes) would be processed by all these tools, to address the cpu time and memory requirement needed.

4. Developments of the computational content

Until 1995, there were no databases for LCA. There were some documents with emission factors and similar information (see, e.g., BUWAL 1996), but they were provided on paper, and any user had to retype these data. Some software systems had taken this effort, and distributed their packages with databases. In fact, many software suppliers still advertise their software by their content in terms of databases, rather than by algorithms (see Figure 1). This is a bit strange situation: many databases are purchased from third parties (such as the ecoinvent consortium or JRC-IES), and can be part of any of these tools. It is like advertising a pdf-reader with the slogan “Ships with 10,000 pdfs!”.

![Figure 1.]

The two major commercial programs for LCA (GaBi and Simparo) stress the databases that are shipped with these programs.
So, let’s go back to the databases themselves. There have been a number of major innovations since 1995.

• databases have started to become available in digital format;
• databases have gone online;
• databases have adapted to the need of data exchange by using universal exchange formats and coding systems (such as CAS-numbers) for automatic matching of entities.

The result is that present-day LCAs can easily contain thousands of unit processes. The present release (v2.2) of ecoinvent has over 4,000 processes, and EIO-tables (that are sometimes used instead of process-based data, or to supplement process-based data) are starting to grow into similar numbers. For example, Exiopol’s IO-table (Tukker, 2009) contains almost 6000 rows and columns.

5. Discussion

We see three developments:
• the basic math of LCA has become mature, but more advanced analysis is underway;
• PCs can do more than ever;
• databases grow extremely large.

It is our concern that PCs will not keep track of the speed at which databases are growing and/or of the requirements that more advanced analysis is starting to pose. Naive matrix inversion is an operation of order N3. That means that doubling the size of the matrix will require eight times longer computing time. Add to this that larger data structures require more page faults, and the problem in performance will become obvious. As an illustration: the 4000 rows and columns of ecoinvent requires on our system 1 minute, while the 6000 rows and columns of Exiopol require 1 hour. That is a basic calculation, but if we go for a Monte Carlo simulations of 1,000 runs, the issue becomes nagging.

Solutions must come from different sides:
• a performance analysis of the bottlenecks of the existing software may reveal snippets of code that are sub-optimal (Heijungs & de Koning, 2010). Even the changing the order of two loops (for row and columns) may speed up performance in some cases.
• the structure of the problem at hand needs consideration. For instance, matrix inversion may be reduced to vector division (Ciroth et al., 2004), and Monte Carlo simulation may be approximated by Gaussian error propagation (Heijungs & Kleijn, 2001). Many more clever solutions must exist.
• the use of sparse matrices can save memory and in certain cases cpu-time as well. This has hardly been explored in LCA, but substantial gains might be possible (Heijungs & Suh, 2002).
• the issue of precision of algorithms should be considered. Especially for large systems of equations, solutions may be subject to extreme round-off errors.

Doing calculations has become so easy, that we tend to forget that such calculations are not just a million-fold repetitions of the well-known calculations old-style. The increase of computer capacity is starting to become insufficient in the light of the data explosion. Input-output tables for 100 sectors for 200 countries create a matrix of 20,000 rows and columns. Adding a bit of refinement, from 100 to 150 sectors, will slow-down the calculations tremendously. Size matters. It is a challenge to find ways out of this.

Bibliography

