ICE — the IBP Chooser of Equations

Philipp Kant
Humboldt-Universität zu Berlin, Institut für Physik,
Newtonstraße 15, 12489 Berlin

September 26, 2013

This is the manual of ICE, a program to choose a maximal linearly independent subset from a given set of Integration-by-Parts and/or Lorentz Invariance equations. The algorithm it implements is described in [1], and the author considers it good practice to cite that paper when the program is used in a calculation leading to a scientific publication.

1 Installation

The easiest way to compile the program from source is to use the cabal build system. It is included in the Haskell Platform[1], which is available on most modern systems, including GNU/Linux, Windows, and OS/X. If you have the Haskell Platform installed, the following commands, executed from the sourcedirectory, will install all needed libraries and compile the program

```
cabal install --only-dependencies
cabal configure
cabal build
```

This will create the executable file `./dist/build/ice/ice`.

2 Usage

ICE is run as

```
ice [OPTIONS] [FILE]
```

where the following options are available:

`-d --dumpfile=FILE` In addition to the output on stdout, print a list of newline-separated equation numbers to `FILE`. Note that the equations are zero-indexed.

[1] //www.haskell.org/platform/

`--intname=NAME` This is used to control the name of the function representing integrals in the input file. The default is `Int`.

`-i --invariants=x` Add the symbol `x` to the list of variables that appear in the polynomials.

`-b --backsub` After forward substitution, perform backward substitution in order to determine which master integrals appear in the result for each integral.

`-r --rmax=n`, `-s --smax=n` Only relevant if `--backsub` is given. Do not try to find a representation for integrals with more than `rmax` dots or more than `smax` scalar products. A system of IBP equations will typically contain some integrals with many dots and/or scalar products (more than the integrals used as seeds in the generation of the system) that can not be determined by the system, but are not master integrals and could be reduced if the system was enlarged. Discarding those before the backward substitution saves some time.

The input file `FILE` should have the following syntax:

- Each line gives one term in an equation in the form
  `Int[<indices>]*(<sum of coefficients>)`
  The brackets are mandatory.
- Equations are terminated and separated by a line consisting of only a semicolon.

For an example, see the following section.

3 Example: One-Loop Massive Self-Energy

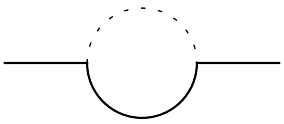

Figure 1: One-Loop massive self-energy

In order to illustrate the usage of the program, we give a simple example input file for the reduction of the diagram shown in Figure 1. The input file with the equations is found in `example/se1l.in`. It contains equations to reduce one scalar product and one dot, with invariants `d` and `m`. The command line to run ICE on this file is

```
./dist/build/ice/ice -id -im example/se1l.in
```

or

```
./dist/build/ice/ice -id -im -r1 -s1 --backsub example/se1l.in
```

In the latter case, ICE also determines which master integrals are needed to express each integral with at most one dot and/or one scalar product. The output will look similar to

```
ICE -- Integration-By-Parts Chooser of Equations
Command line arguments: Config {inputFile = "example/se1l.in"
, dumpFile = "", intName = "Int", intNumbers = False
, invariants = ["d","m"], rMax = 1, sMax = 1, backsub = True}
Number of equations: 8
Number of integrals: 8
Number of integrals within r=1, s=1: 4
Probing for p = 3036998761
Random points: [2887887967,690347565]
Number of linearly independent equations: 7
Indices of linearly independent equations (starting at 0):
5
4
1
6
0
3
2
Integrals that can be reduced with these equations:
Int[2,-1]
Int[1,-1]
Int[2,0]
Possible Master Integrals:
Int[1,0]
Doing backward substitution.
Final representations of the integrals will look like:
Int[2,-1] -> {Int[1,0]}
Int[1,-1] -> {Int[1,0]}
Int[2,0] -> {Int[1,0]}
The probability that too many equations were discarded
 is less than 9.219628172019156e-9
Timings:
Parsing and preparing equations: 0.000514s
Solving Equations: 0.000517s
```

First, ICE reports the values of the command line arguments given. Next, the number of equations and integrals, as well as the number of integrals that lie within the region given by the values of `rmax` and `smax` is listed.

After that, the actual algorithm starts. The prime number and evaluation point is given, followed by the number of linearly independent equations and their positions in the input file (starting with zero). This is the main information of interest for a subsequent run of Laporta's Algorithm. With the option `--dumpfile`, it is possible to write this list to a separate file.

Finally, we get information on which integrals were reduced and which are considered master integrals. In case the `--backsub` option is given, ICE also gives a lists of which master

integrals appear in the expression for each integral that could be reduced.

References

[1] P. Kant. “Solving Linear Equations Over Sparse Multivariate Polynomials”. In: *in preparation* ().