DOCUMENTATION FOR

CONAN

Version 1.6

8th March 2017

1 About CONAN

CONAN (Coefficients of One-dimensional N-Atom Networks) is an open source program for calculating local exchange coefficients for strongly interacting confined systems in one dimension released under GNU General Public License, version 3 (GPL-3.0):

A physical motivation for the code and a thorough description of the computational algorithm is given in Ref. [1]. In this reference, you can also find guides on the use of CONAN including examples with typical parameter choices.

We ask that in any scientific publication based wholly or in part on the CONAN software, the use of CONAN must be acknowledged and Ref. [1] must be cited.

2 Installation

You can use CONAN in two ways:

1. Use the pre-compiled version for Linux. It works out of the box and is run from a terminal/command prompt, see Section 3.
2. Build the program yourself using the source code. Below we give instructions on how to compile the program on Linux and Mac. Then run it as the pre-compiled version.

2.1 Compile CONAN

CONAN depends on four other libraries

- `GSL`: `http://www.gnu.org/software/gsl/`
- `GMP`: `https://gmplib.org/`
- `MPFR`: `http://www.mpfr.org/`
- `OpenMP`: `http://openmp.org/wp/`

Beside this CONAN uses the cross platform build tool `CMake` (`https://cmake.org/`).

Linux

Install the four dependencies by following the references on their websites or your preferred package manager. Make sure they are installed in your `PATH`, or update it accordingly. Note that `MPFR` dependes on `GMP`, so you want to follow above order.

Extract the project to a folder. Inside that folder, create a directory called `build` (this will seperate build files from the source files). Go into the `build` directory and call the command

```
cmake ../
```

CMake will now generate a Makefile and you can now simply call

```
make
```

Conan should be located in your [path to Conan]/build folder and can be run like the pre-compiled versions.

OSX

Install the four dependencies using the homebrew package manager. In order to use OpenMP you need to install GCC with the flag --without-multilib. Simply download and install homebrew. Now reinstall GCC

```
brew reinstall gcc --without-multilib
```

Also install CMake

```
brew install cmake
```

You are now ready to compile CONAN.

Extract the project to a folder. Inside that folder, create a directory called build (this will seperate build files from the source files). Go into the build directory and call the command

NB You may need to specify your version of GCC to CMake. If it doesn't work, go change the line set(CMAKE_C_COMPILER gcc-5) to the version of GCC you are using.

```
cmake ../
```

CMake will now generate a Makefile and you can now simply call

```
make
```

Conan should be located in your [path to Conan]/build folder and can be run like the pre-compiled versions.

3 Usage

3.1 Basic usage

The basic usage syntax for CONAN is

```
Conan [-options] [<params>]
```

Calling CONAN without any options brings up a list of possible options. CONAN expects at least two mandatory options in order to do anything useful. The first one is the potential describing the system, specified with a -V. The potential is a mathematical function on the interval [0,L], where the length is by default L=100. The second mandatory option is the number of particles trapped in that potential. This is specified with a -N. A minimal example on this could be 5 particles in the harmonic oscillator centered around the middle of the interval [0;L]. Go into the folder where Conan is located and run the following command in the terminal:

```
./Conan -V '0.1*(x-L/2)^2' -N 5
```

CONAN will immediately show the parameter values for the calculation, and when the calculation is done, the output is printed. The geometric coefficients (α_k in Ref. [1]) in units of ℓ^{-3} (where ℓ is some length unit) will is printed as

```
I 0.385186 0.564994 0.564994 0.385186
```

in the order $\alpha_1\ \alpha_2\ \ldots\ \alpha_{N-1}$. The eigenenergies $E_1\ E_2\ \ldots\ E_N$ for the non-interacting system in units of $\varepsilon = 1/2\ell^2$ (with $\hbar$ and the mass of the particles set to unity) is printed as

```
E 0.316228 0.948683 1.58114 2.21359 2.84605
```

3.2 The potential

The potential is the most important input. As given in the earlier example, it is specified as a normal mathematical expression. The program supports a long list of mathematical functions from the `math.h` library that may be combined to build the potential:

- **Arithmetic operators:** +, -, *, /, ^
- **Single parameter functions:** acos, acosh, asin, asinh, atan, atanh, cbrt, ceil, cos, cosh, erfc, erf, exp2, exp, expm1, fabs, lgamma, log10, log2, log1p, logb, log, nearbyint, rint, round, sin, sinh, sqrt, tan, tanh, tgamma, trunc
- **Two parameter functions:** atan2, copysign, pow, fdim, fmax, fmin, fmod, hypot, nextafter, remainder
- **Three parameter functions:** fma

The potential should be written as a function of the parameter `x`, corresponding to the one-dimensional position in the system. The calculations are performed on the interval `[0;L]`, so the potential should be constructed with this interval in mind.

Built in parameters As mentioned above CONAN uses the build in parameters `x` for the position and `L` for the length of the region in which it works. It is possible to specify the length `L` through the option `-L`, for instance `-L 40` would set the length to 40. See also Section 3.3.

User supplied parameters You can make CONAN calculate multiple configuration of your system by specifying your own parameters in your potential. Say you want to offset your harmonic oscillator in the x-direction and want to calculate the resulting coefficients for multiple values of the offset. You can then specify it as a parameter and supply that parameter in the end of call to CONAN:

```
Conan -V '0.1*(x-L/2+d)^2' -N 5 d=10
```

This is of course not particularly interesting since you just could have put the `10` directly in your expression. You can, however, also supply lists and it will calculate the whole range in one CONAN-run:

```
Conan -V '0.1*(x-d)^2' -N 5 d=[10,20,30]
```

You can even specify multiple parameters:

```
Conan -V 'h*(x-d)^2' -N 5 d=[10,20,30] h=[0.1,0.2,0.3]
```

The rules for the accepted lengths of the parameter lists are:

- Parameters of length 1 is always accepted: `d=4.2`
- If you have multiple parameters lists, they must agree in length.

Examples:

```
Conan -N 5 -V '0.1*(x-d)^2' d=[10,20,30] // OK
Conan -N 5 -V 'h*(x-d)^2' d=[10,20,30] h=0.1 // OK
Conan -N 5 -V 'h*(x-d)^2' d=[10,20,30] h=[0.1,0.2] // NOT OK
Conan -N 5 -V 'h*(x-d)^2' d=[10,20,30] h=[0.1,0.2,0.3] // OK
Conan -N 5 -V 'h*(x-d)^2+g*x' d=[10,20,30] h=[0.1,0.2,0.3] g=0.2 // OK
Conan -N 5 -V 'h*(x-d)^2+g*x' d=[10,20,30] h=[0.1,0.2,0.3] g=[0.2,0.3]
   // NOT OK
```

NB The numerical routines do not handle constant potentials in a stable way, unless it is 0. A constant potential does, however, only affect the energies by a constant shift relative to a zero-potential. Should you need this, we advise you simply calculate the 0 potential and add the constant the the outputted energies.

3.3 Options

Beside the potential and the number of particles, you can specify a range of parameters related to both the system considered and the numerical routines. Most of them have a shortcut and should be prefixed with a single `-`. The four last options can be specified with a double `--`. All the options are listed in Table 3.3, when relevant we also state the notation used in Ref. [1].

To get the most out of CONAN, it is important to set the various parameters correctly as to gain high precision results without spending too much unecessary computation time. Furthermore, it is important to think carefully about what is submitted to CONAN in order to interpret the result correctly within a physics context.

NB We strongly recommend that the user carefully reads Section VI and VII in Ref. [1] in order to get an idea of what potentials and parameter values should be submitted to CONAN and what the precision is on the results.

4 FAQ

The first coefficient is way bigger than expected and/or the remaining coefficients. What is wrong?
You have probably picked an insufficient bit precision. Increase it with the command `-p`. When N is sufficiently large, one cannot rely on the default value of 256, so try double it by adding `-p 512` to the command when you run CONAN.

The program works, but sometimes it returns an error or wierd results. What is wrong?
You're options for the program are probably badly chosen, only for small N are the default values sufficient, or your potential have some issues. Carefully read the guide and the examples in [1].

Will there be a Windows version?
Probably not.

I have encountered bugs, I have a feature request or general feedback. Who do I contact?
Great! Please contact Niels Jakob Søe Loft, Department of Physics and Astronomy, Aarhus University, DK-8000 Aarhus C, Denmark.

References

[1] N. J. S. Loft, L. B. Kristensen, A. E. Thomsen, A. G. Volosniev, and N. T. Zinner: *CONAN – the cruncher of local exchange coefficients for strongly interacting confined systems in one dimension*, Computer Physics Communications **209**, 171 (2016). The paper and the program source files can also be found here.

Table 1: Full list of available commands for CONAN.

Short	Long	Description	
		Mandatory! Requires 1 input.	
`-V`	`--potential`	The potential as function of `x` in units of ε. This is denoted $\tilde{V}(\tilde{x})$ in Ref. [1].	
`-N`	`--number-of-par`	The number of particles N.	
		Numerical parameters. Requires 1 input.	Default value
`-L`	`--wellwidth`	The length of the x-axis to be used in the calculation, aka the length of the expansion box in units of ℓ, denoted $\tilde{L}$.	100.0
`-b`	`--basesize`	Number of sine-functions used to as a basis to solve the Schrödinger equation, denoted N_b.	300
`-p`	`--precision`	Number of bits used to store each number, denoted p.	256
	`--abs-solver`	Absolute precision used when finding the matrix elements of the potential.	1e−9
	`--rel-solver`	Relative precision used when finding the matrix elements of the potential.	1e−9
	`--abs-final`	Absolute precision used when performing the final integration.	1e−5
	`--rel-final`	Relative precision used when performing the final integration.	1e−5
		Output commands.	Example.
`-D`	`--just-potential`	With this flag the potential will be outputted to a file in your working directory and nothing is calculated. This is mainly for plotting and debugging purposes.	`-D`
`-E`	`--just-energy`	With this flag the coeffecients are not calculated, only the energies.	`-E`
`-o`	`--filename`	If you specify a filename, the output of the program will be appended to this file. The file is created if it does not exists.	`-o output.txt`