

Aiida (May 2019)

Total Questions: 2

Most Correct Answers: #0

Least Correct Answers: #0

1. Why – if at all – do you feel safe that your DFT code solves the Kohn-Sham equations correctly? On which arguments is your confidence based?

Anon anon096d454ac7744d7b

No. Bugs and different detailed formulae to be implemented may give different values.

Anon anon14d306792cf647e5

The code was able to reproduce experimental data in the past

Anon anon34fa3fd3ee0447bb

Test test test. Check on experimentally available quantity and literature results on similar systems.

Anon anon48a7eb090bbd4f26

since they predict many experimental observations reasonably.

Anon anon4c2936a500e44897

Comparing the result in different dft codes.

Anon anon4f42b3689e324ed5

In my opinion, it is not an easy decision to make. As we normally rely on the protocols published by others and compare some parameters with experimental data.

Anon anon5f8ae494e9144cc8

Comparing the results with the experimental results if available. In the case of unavailable experimental results analogy with other calculations(materials) can be made.

Anon anon63fdbcb8f13a498b

Trust the developers that they verified their code, the responsibility lies by them.

Anon anon6448e21e4f4e4120

Regarding open source codes, I can look inside of it to check for myself what the code is actually doing.

Anon anon6747c164192a4a05

Benchmarking with other DFT codes, if possible with different basis sets/PSP/etc.

Anon anon7063778a16864cc1

The scf

Anon anon83bdc83fb247495a

It is correct, because it reproduced the results on a number of test example, which agree well with other codes.

Anon anon8cf8c2787a8d4ebc

The most important assumption is the developers of the code should report error message if something goes wrong. As a user I use cross validation between codes and comparison to known solutions (calibration).

Anon anon95bbed1eb2884254

Only by comparing it to existing, "well known" results.

Anon anon9ede18eb2a114f04

Not at all, but if you can compare to experiments

Anon anona0bb6a6d49c44cd5

Because of delta tests: comparison of different codes against the same computational problems

Anon anonba29b6eeb2df4b50

DFT codes are complex, and scientist around the world make sure they are correct enough. If I'm using an open source code I'm confident that I could spot the issue and work towards a solution.

Anon anonbdeaac3346ea4210

Benchmarks with various Materials using experimental and computational reference data

Anon anonc095aed418444618

reproducibility of existing data in the literature (possibly obtained with other codes), extensive checks on new features

Anon anonce21d349f1544606

To be honest: the popularity of the code makes me confident.

Anon anonce2c3f675f9749f7

If my code -- that does not include any source file in common with another DFT code -- reproduces the same results, I estimate that it is very unlikely that the two codes accidentally contain the same errors.

Anon anonce8d5bf34c2b49ea

Only moderately and contingently confident. Eg, uspps (ultrasoft pseudopotentials) very extensively used, known to do a bad job with many systems. Only way I can see to gain confidence is continuous benchmarking / cross validation on relevant systems. Systems too complex to manually check or even more extremely do, to mathematically prove...

Anon anoncf007a535158417d

comparison with other codes

Anon anon28eb024cc7b4800

Hope in many years of development

Anon anondb3b329d4bdf49d8

check the experimental values with dft calculations

Anon anon866aeca0454d67

It seems to give the same results that other researchers get. However, I'm not 100% confident that everything is working as it should, and as a matter of fact a couple years ago we discovered (and fixed) a bug in solving the Dirac-Kohn-Sham equation for the "small component"

Anon anonf63a0d2388eb4875

only if it is for a single element! :-)

Anon anonf825b0325e4f4c27

The agreement between different codes (<https://molmod.ugent.be/deltacodesdft>) based on different solvers and basis set, eliminates the possibility of large accidental errors at the very list.

Anon anonfc4c242bc98b48f2

- Many codes are open-source and researchers have tried to improve on the method - so major bugs would have been spotted
- Many people have used it
- If at least two codes that have been independently implemented give the same answer, this is a strong hint that they might be right

2. Were old DFT codes as good as the present ones? Why (not)?

Anon anon096d454ac7744d7b

Maybe not, depending on the bit size of variables hard coded.

Anon anon14d306792cf647e5

Did not run simulations with old codes

Anon anon19b4e16da5604610

the more a code has been on the market, the more bugs have been discovered and fixed

Anon anon1f8d1ecf44264957

probably yes! the pseudos were not so good

Anon anon291d2c64d125432a

Yes and no. It is the matter of parallelization and state-of-the-art algorithms which are available in new ones.

Anon anon34fa3fd3ee0447bb

Not as numerically accurate. Building high quality pseudopotential has become easier now and is based on more refined methods.

Anon anon48a7eb090bbd4f26

yes

Anon anon4c2936a500e44897

Pseudo potential getting more consistent.

Anon anon5f8ae494e9144cc8

No, much fewer options for extensibility.

Anon anon63fdbcb8f13a498b

Probably no, because of more complicated setup and new implementations (Ex/Corr functionals, new algorithms, ...).

Anon anon6448e21e4f4e4120

I'm not sure, but I would guess not. Probably because of the necessity of more optimization and approximations in order to cope with the scarcity of computational power.

Anon anon83bdc83fb247495a

I've never used old ones. But I'm quite convinced that modern codes are way better.

Anon anon8cf8c2787a8d4ebc

They are as good in terms of being able to solve the same numerical problems. Not because performance improvement.

Anon anon95bbbed1eb2884254

I would imagine that they would not be as good. Otherwise we would not be coming up with new ones and using improved codes.

Anon anon9ede18eb2a114f04

No reasearch went on, new features has been developed and new aspects were found to be observed.

Anon anona0bb6a6d49c44cd5

No, the old codes are:

- 1) Less optimal
- 2) Provide less features
- 3) Less scalable
- 4) Less user-friendly
- 5) Based on older pseudopotentials

Anon anona40e6fcb70424cd3

I think so, maybe the old codes lack in accuracy due computational cost.

Anon anonbdeaac3346ea4210

No, less data to fit pseudopotentials and functionals

Anon anonce21d349f1544606

I think that old DFT codes can give less accuracy because they are based on more approximations in order to run with older (slow) computers.

Anon anonce2c3f675f9749f7

Older codes (to the extent that they still exist) have the advantage of a longer history of bug tracking and fixing.

Anon anonce8d5bf34c2b49ea

No, ease of use, speed on massively parallel hardware, range of XC functionals available, other features / methods.

Numerical precision may or may not be as good. Theoretically should be better as less need to compromise for slow hardware, however systems more complex and perhaps therefore harder to test and maintain.

Anon anoncf007a535158417d

it depends on the quality of pseudos, and the reachability of cutoffs for convergence.

Anon anon28eb024cc7b4800

What is 'good'? Precise, efficient? The answer depends on this.

Anon anondb3b329d4bdf49d8

depend on how much the values that we get from each dft code are near to the experimental values...

Anon anonea866aeca0454d67

I'm working with an "old" DFT code and it's terrible to work with. It's a mess to maintain, with unpronounceable, short, ALCPSLK variable and subroutine names. It may produce correct physical results and be relatively performant, but developing it is an ordeal.

I hope newer codes are better.

Anon anoned80a984d77a4657

Aside from the already mentioned Pseudopotential quality, there has been an advancement in correlation-functionals to address specific problems with DFT (band gap or lattice parameters). Potentially, solvers have also improved in accuracy. And computation power is more widely available needing less approximations to achieve tractability.

Anon anonfc4c242bc98b48f2

No:

- Codes were not open-source but developed in-house, hard to check for others
- they were less robust (the authors knew how to fix problems, now the codes are (a bit more) high-throughput-ready
- old programming standards, that were more bug-prone (see F90-95-2003 vs F77)