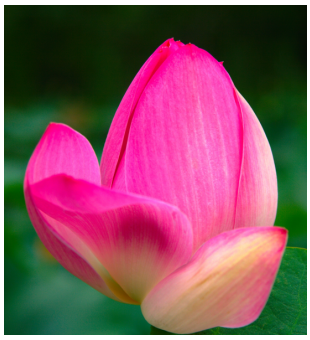

Reference Manual for Interacting Quantum Impurity Systems Simulating Toolkit

Core Developers:

Li Huang[†] and **Yilin Wang**[‡]

Key Contributors:

Zi Yang Meng[‡] and **Liang Du**[♯]

Directors and Supervisors:

Philipp Werner[†] and **Xi Dai**[‡]

[†]Department of Physics, University of Fribourg, 1700 Fribourg, Switzerland

[‡]Beijing National Laboratory for Condensed Matter Physics, and Institute of Physics, Chinese Academy of Sciences, Beijing 100190, China

[♯]Department of Physics, The University of Texas at Austin, Austin, Texas 78712, USA

Draft Version October 7, 2014

To my lovely wife X. Zhao

L. H

To my lovely girlfriend X.Y. Mao

Y.L. Wang

Copyright 2014 by Li Huang

Permission is granted to copy, distribute and/or modify *the documentation* under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

Permission is granted to copy, distribute and/or modify *the code of the package* under the terms of the GNU Public License, Version 2 or any later version published by the Free Software Foundation.

Permission is also granted to distribute and/or modify *both the documentation and the code* under the conditions of the LaTeX Project Public License, either version 1.3 of this license or (at your option) any later version.

Contents

1 INTRODUCTION **1**

1.1 What's *i*QIST ? . . . 1

1.2 Motivation . . . 1

1.3 Software architecture . . . 1

1.4 Main features . . . 1

1.5 Development history . . . 1

1.6 Policy and licences . . . 1

2 INSTALLATION **3**

2.1 Obtain . . . 3

2.2 Uncompress . . . 3

2.3 Direcrory structures . . . 3

2.4 Compiling environment . . . 3

2.5 Compiling system . . . 3

2.6 Build impurity solvers . . . 3

2.7 Build auxiliary tools . . . 3

2.8 Build documents . . . 3

2.9 Build application programming interfaces . . . 3

3 RUNNING **5**

3.1 Configure your system . . . 5

3.2 Create input files . . . 5

3.3 Execute codes . . . 5

3.4 Monitor and Profile . . . 5

4 STANDARD INPUT FILES **7**

4.1 solver.ctqmc.in . . . 7

4.2 solver.eimp.in . . . 7
4.3 solver.hyb.in . . . 7
4.4 solver.anydos.in . . . 7
4.5 solver.ktau.in . . . 7
4.6 atom.cix . . . 7

5 STANDARD OUTPUT FILES **9**
5.1 Terminal output . . . 10
5.1.1 out.dat . . . 10
5.2 File output . . . 10
5.2.1 solver.green.dat . . . 10
5.2.2 solver.green.bin . . . 10
5.2.3 solver.weiss.dat . . . 10
5.2.4 solver.hybrid.dat . . . 10
5.2.5 solver.grn.dat . . . 10
5.2.6 solver.wss.dat . . . 10
5.2.7 solver.hyb.dat . . . 10
5.2.8 solver.sgm.dat . . . 10
5.2.9 solver.hub.dat . . . 10
5.2.10 solver.nmat.dat . . . 10
5.2.11 solver.schi.dat . . . 10
5.2.12 solver.ochi.dat . . . 10
5.2.13 solver.twop.dat . . . 10
5.2.14 solver.vrtx.dat . . . 10
5.2.15 solver.hist.dat . . . 10
5.2.16 solver.prob.dat . . . 10
5.2.17 solver.kernel.dat . . . 10
5.2.18 solver.status.dat . . . 10

6 PARAMETERS **11**
6.1 isscf . . . 12
6.2 issun . . . 12
6.3 isspn . . . 12
6.4 isbin . . . 12
6.5 isort . . . 12
6.6 isvrt . . . 12

6.7 isscr . . . 12
6.8 nband . . . 12
6.9 nspin . . . 12
6.10 norbs . . . 12
6.11 ncfgs . . . 12
6.12 nzero . . . 12
6.13 nvect . . . 12
6.14 nhmat . . . 12
6.15 nfmat . . . 12
6.16 niter . . . 12
6.17 U . . . 12
6.18 Uc . . . 12
6.19 Uv . . . 12
6.20 Jz . . . 12
6.21 Js . . . 12
6.22 Jp . . . 12
6.23 lc . . . 12
6.24 wc . . . 12
6.25 mune . . . 12
6.26 beta . . . 12
6.27 part . . . 12
6.28 alpha . . . 12
6.29 lemax . . . 12
6.30 legrd . . . 12
6.31 chmax . . . 12
6.32 chgrd . . . 12
6.33 mkink . . . 12
6.34 mfreq . . . 12
6.35 nffrq . . . 12
6.36 nbfrq . . . 12
6.37 nfreq . . . 12
6.38 ntime . . . 12
6.39 nleja . . . 12
6.40 npart . . . 12
6.41 nflip . . . 12

6.42 ntherm . . . 12
6.43 nsweep . . . 12
6.44 nwrite . . . 12
6.45 nclean . . . 12
6.46 nmonte . . . 12
6.47 ncarlo . . . 12

7 AUXILIARY TOOLS 13

7.1 *J*asmine component . . . 14
7.2 *H*ibiscus component . . . 14
7.2.1 Maximum entropy method: entropy1 . . . 14
7.2.2 Maximum entropy method: entropy2 . . . 14
7.2.3 Stochastic analytical continuation: sac . . . 14
7.2.4 Analytical continuation for self-energy: swing . . . 14
7.2.5 toolbox/makechi . . . 14
7.2.6 toolbox/makedos . . . 14
7.2.7 toolbox/makekra . . . 14
7.2.8 toolbox/makescr . . . 14
7.2.9 toolbox/makesig . . . 14
7.2.10 toolbox/makestd . . . 14
7.2.11 toolbox/maketau . . . 14
7.2.12 toolbox/makeups . . . 14
7.2.13 script/pysci.py . . . 14
7.2.14 script/check.py . . . 14
7.3 Parquet component . . . 14

8 APPLICATION PROGRAMMING INTERFACES 15

8.1 Fortran binding . . . 15
8.2 Python binding . . . 15
8.3 iqist.py . . . 15

9 *i*QIST IN ACTION 17

9.1 Basic applications . . . 18
9.1.1 Hello *i*QIST ! . . . 18
9.1.2 Mott metal-insulator transition . . . 18
9.2 Advanced applications I: Complex systems . . . 18

9.2.1 General Coulomb interaction . . . 18
9.2.2 Spin-orbital coupling . . . 18
9.2.3 Crystal field splitting . . . 18
9.2.4 Retarded interaction and dynamical screening effect . . . 18
9.3 Advanced applications II: Accurate measurement of physical observables . . . 18
9.3.1 One-shot and self-consistent calculations . . . 18
9.3.2 Data binning mode . . . 18
9.3.3 Imaginary-time Green's function . . . 18
9.3.4 Matsubara Green's function and self-energy function . . . 18
9.3.5 Spin-spin correlation function and orbital-orbital correlation function . . . 18
9.3.6 Two-particle Green's function and vertex function . . . 18
9.4 Advanced applications III: post-processing procedures . . . 18
9.4.1 Analytical continuation for imaginary-time Green's function . . . 18
9.4.2 Analytical continuation for Matsubara self-energy function . . . 18
9.5 Practical exercises . . . 18
9.5.1 Orbital-selective Mott transition in two-band Hubbard model . . . 18
9.5.2 Orbital Kondo and spin Kondo effects in three-band Anderson impurity model . . . 18

10 INSIDE *i*QIST 19

10.1 Basic theory and methods . . . 20
10.1.1 Quantum impurity model . . . 20
10.1.2 Principles of continuous-time quantum Monte Carlo algorithm . . . 20
10.1.3 Hybridization expansion . . . 20
10.1.4 Physical observables . . . 20
10.1.5 Two-particle measurements and DMFT + Parquet formalism . . . 20
10.2 Implementations and optimizations . . . 20
10.2.1 Development platform . . . 20
10.2.2 Orthogonal polynomial representation . . . 20
10.2.3 Improved estimator for the self-energy function . . . 20
10.2.4 Random number generators . . . 20
10.2.5 Subspaces and symmetry . . . 20
10.2.6 Truncation approximation . . . 20
10.2.7 Lazy trace evaluation . . . 20
10.2.8 Divide-and-conquer and sparse matrix tricks . . . 20
10.2.9 Parallelization . . . 20

Appendix **21**

A.1 TODO 21

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 What's *i*QIST ?

1.2 Motivation

1.3 Software architecture

1.4 Main features

1.5 Development history

1.6 Policy and licences

Chapter 2

INSTALLATION

2.1 Obtain

2.2 Uncompress

2.3 Direcrory structures

2.4 Compiling environment

2.5 Compiling system

2.6 Build impurity solvers

2.7 Build auxiliary tools

2.8 Build documents

2.9 Build application programming interfaces

Chapter 3

RUNNING

3.1 Configure your system

3.2 Create input files

3.3 Execute codes

3.4 Monitor and Profile

Chapter 4

STANDARD INPUT FILES

4.1 solver.ctqmc.in

4.2 solver.eimp.in

4.3 solver.hyb.in

4.4 solver.anydos.in

4.5 solver.ktau.in

4.6 atom.cix

Chapter 5

STANDARD OUTPUT FILES

5.1 Terminal output

5.1.1 out.dat

5.2 File output

5.2.1 solver.green.dat

5.2.2 solver.green.bin

5.2.3 solver.weiss.dat

5.2.4 solver.hybrid.dat

5.2.5 solver.grn.dat

5.2.6 solver.wss.dat

5.2.7 solver.hyb.dat

5.2.8 solver.sgm.dat

5.2.9 solver.hub.dat

5.2.10 solver.nmat.dat

5.2.11 solver.schi.dat

5.2.12 solver.ochi.dat

5.2.13 solver.twop.dat

5.2.14 solver.vrtx.dat

Chapter 6

PARAMETERS

6.1 isscf

6.2 issun

6.3 isspn

6.4 isbin

6.5 isort

6.6 isvrt

6.7 isscr

6.8 nband

6.9 nspin

6.10 norbs

6.11 ncfgs

6.12 nzero

6.13 nvect

6.14 nhmat

Chapter 7

AUXILIARY TOOLS

7.1 $\mathcal{J}$asmine component

7.2 $\mathcal{H}$ibiscus component

7.2.1 Maximum entropy method: entropy1

7.2.2 Maximum entropy method: entropy2

7.2.3 Stochastic analytical continuation: sac

7.2.4 Analytical continuation for self-energy: swing

7.2.5 toolbox/makechi

7.2.6 toolbox/makedos

7.2.7 toolbox/makekra

7.2.8 toolbox/makescr

7.2.9 toolbox/makesig

7.2.10 toolbox/makestd

7.2.11 toolbox/maketau

7.2.12 toolbox/makeups

7.2.13 script/pysci.py

7.2.14 script/check.py

7.3 Parquet component

Chapter 8

APPLICATION PROGRAMMING INTERFACES

8.1 Fortran binding

8.2 Python binding

8.3 iqist.py

Chapter 9

iQIST IN ACTION

9.1 Basic applications

9.1.1 Hello *iQIST* !

9.1.2 Mott metal-insulator transition

9.2 Advanced applications I: Complex systems

9.2.1 General Coulomb interaction

9.2.2 Spin-orbital coupling

9.2.3 Crystal field splitting

9.2.4 Retarded interaction and dynamical screening effect

9.3 Advanced applications II: Accurate measurement of physical observables

9.3.1 One-shot and self-consistent calculations

9.3.2 Data binning mode

9.3.3 Imaginary-time Green's function

9.3.4 Matsubara Green's function and self-energy function

9.3.5 Spin-spin correlation function and orbital-orbital correlation function

9.3.6 Two-particle Green's function and vertex function

9.4 Advanced applications III: post-processing procedures

9.4.1 Analytical continuation for imaginary-time Green's function

9.4.2 Analytical continuation for Matsubara self-energy function

9.5 Practical exercises

9.5.1 Orbital-selective Mott transition in two-band Hubbard model

9.5.2 Orbital Kondo and spin Kondo effects in three-band Anderson impurity

Chapter 10

INSIDE iQIST

10.1 Basic theory and methods

10.1.1 Quantum impurity model

10.1.2 Principles of continuous-time quantum Monte Carlo algorithm

10.1.3 Hybridization expansion

10.1.4 Physical observables

10.1.5 Two-particle measurements and DMFT + Parquet formalism

10.2 Implementations and optimizations

10.2.1 Development platform

10.2.2 Orthogonal polynomial representation

10.2.3 Improved estimator for the self-energy function

10.2.4 Random number generators

10.2.5 Subspaces and symmetry

10.2.6 Truncation approximation

10.2.7 Lazy trace evaluation

10.2.8 Divide-and-conquer and sparse matrix tricks

10.2.9 Parallelization

Appendix

A.1 TODO