



*i*QIST 用户指南

User's Guide for **I**nteracting **Q**uantum **I**mpurity
Systems Simulating **T**oolkit

黄理

表面物理与化学国家重点实验室

王义林 杜亮 戴希

中国科学院物理研究所

凝聚态物理国家实验室

To my lovely wife X. Zhao

L. H

To my lovely girlfriend X.Y. Mao

Y.L. Wang

Copyright 2012 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.

目 录

1	<i>i</i>QIST 简介	1
1.1	什么是 <i>i</i> QIST ?	1
1.2	<i>i</i> QIST 的项目背景	2
1.3	<i>i</i> QIST 的组织架构	4
1.4	<i>i</i> QIST 的主要功能	7
1.5	<i>i</i> QIST 的版本历史	8
1.6	<i>i</i> QIST 的使用条款	9
2	<i>i</i>QIST 的安装配置	11
2.1	获取 <i>i</i> QIST	11
2.2	解压缩 <i>i</i> QIST	11
2.3	<i>i</i> QIST 的目录结构	12
2.4	<i>i</i> QIST 组件的编译链接	13
2.4.1	基本编译环境	13
2.4.2	编译过程详解	14
2.5	<i>i</i> QIST 组件的安装设置	16
3	<i>i</i>QIST 的运行	17
3.1	搭建 <i>i</i> QIST 运行平台	17
3.2	创建 <i>i</i> QIST 输入文件	18
3.3	<i>i</i> QIST 组件的运行	19
3.4	<i>i</i> QIST 组件的监测	20
4	<i>i</i>QIST 的标准输入文件	21
4.1	solver.ctqmc.in	21

4.1.1	<i>Azalea</i> //solver.ctqmc.in	21
4.1.2	<i>Gardenia</i> //solver.ctqmc.in	24
4.1.3	<i>Narcissus</i> //solver.ctqmc.in	27
4.1.4	<i>Begonia</i> //solver.ctqmc.in	31
4.1.5	<i>Lavender</i> //solver.ctqmc.in	33
4.2	solver.eimp.in	37
4.3	solver.hyb.in	38
4.4	solver.anydos.in	39
4.5	solver.ktau.in	40
4.6	atom.cix	41
5	<i>iQIST</i> 的标准输出文件	43
5.1	终端输出	43
5.1.1	out.dat	43
5.2	文件输出	48
5.2.1	solver.green.dat	48
5.2.2	solver.green.bin	49
5.2.3	solver.weiss.dat	49
5.2.4	solver.hybri.dat	50
5.2.5	solver.grn.dat	50
5.2.6	solver.wss.dat	51
5.2.7	solver.hyb.dat	51
5.2.8	solver.sgm.dat	52
5.2.9	solver.hub.dat	53
5.2.10	solver.nmat.dat	53
5.2.11	solver.schi.dat	54
5.2.12	solver.ochi.dat	54
5.2.13	solver.twop.dat	55
5.2.14	solver.vrtx.dat	56
5.2.15	solver.hist.dat	56
5.2.16	solver.prob.dat	57
5.2.17	solver.kernel.dat	58
5.2.18	solver.status.dat	58

6	<i>i</i>QIST 配置文件详解	59
6.1	isscf	59
6.2	issun	60
6.3	isspn	60
6.4	isbin	61
6.5	isort	62
6.6	isvrt	63
6.7	isscr	64
6.8	nband	65
6.9	nspin	65
6.10	norbs	65
6.11	ncfgs	66
6.12	nzero	66
6.13	nvect	66
6.14	nhmat	67
6.15	nfmat	67
6.16	niter	67
6.17	U	68
6.18	U_c	68
6.19	U_v	68
6.20	J_z	69
6.21	J_s	69
6.22	J_p	69
6.23	lc	69
6.24	wc	70
6.25	mune	70
6.26	beta	71
6.27	part	71
6.28	alpha	71
6.29	lemax	72
6.30	legrd	72
6.31	chmax	72

6.32	chgrd	73
6.33	mkink	73
6.34	mfreq	73
6.35	nffrq	73
6.36	nbfrq	74
6.37	nfreq	74
6.38	ntime	74
6.39	nleja	75
6.40	npart	75
6.41	nflip	75
6.42	ntherm	77
6.43	nsweep	77
6.44	nwrite	77
6.45	nclean	78
6.46	nmonte	78
6.47	ncarlo	78
7	iQIST 附属工具程序	81
7.1	Jasmine 组件	81
7.2	Hibiscus /hibiscus-entropy 组件	85
7.3	Hibiscus /hibiscus-stochastic 组件	88
7.4	Hibiscus /hibiscus-swing 组件	92
7.5	Hibiscus /hibiscus-toolbox 组件	94
7.5.1	makechi	94
7.5.2	makedos	94
7.5.3	makekra	95
7.5.4	makescr	95
7.5.5	makesig	96
7.5.6	makestd	96
7.5.7	maketau	96
7.5.8	makeups	97
7.6	pysci.py 程序	97

8	<i>i</i> QIST 典型应用场景	99
8.1	基本应用	99
8.1.1	初识 <i>i</i> QIST	99
8.1.2	Mott 金属 – 绝缘体相变	106
8.2	进阶应用 I: 新的物理	111
8.2.1	广义相互作用项	111
8.2.2	自旋 – 轨道耦合作用	115
8.2.3	晶体场劈裂效应	120
8.2.4	动态屏蔽效应	123
8.3	进阶应用 II: 物理量的精确测量	127
8.3.1	单步执行模式	127
8.3.2	data binning 模式	127
8.3.3	虚时格林函数	129
8.3.4	虚频格林函数与电子自能函数	130
8.3.5	自旋 – 自旋关联函数与轨道 – 轨道关联函数	130
8.3.6	双粒子格林函数与顶角函数	133
8.4	进阶应用 III: 后处理步骤	138
8.4.1	虚时格林函数的解析延拓	138
8.4.2	虚频电子自能函数的解析延拓	141
8.5	<i>i</i> QIST 实战应用	143
8.5.1	两带 Hubbard 模型中掺杂与晶体场相互作用导致的轨道选择性 Mott 相变	145
8.5.2	三带 Anderson 杂质模型中的轨道 Kondo 效应和自旋 Kondo 效应	146
	附录	151
A.1	如何获得 <i>i</i> QIST 的最新版本?	151
A.2	如何获得本文档的最新版本?	151
A.3	<i>i</i> QIST 的开发者是谁?	151
A.4	<i>i</i> QIST 的维护者是谁?	151
A.5	如何获得技术支持?	152
A.6	如何正确选取 <i>i</i> QIST 的组件?	152
A.7	什么时候可以使用正交多项式方法?	152
A.8	如何激活内核多项式功能?	152

A.9 如何固定占据数，搜索费米面?	153
A.10 如何计算高阶关联函数?	153
A.11 如何考虑不同的晶格态密度?	153
A.12 出现 segment fault 错误时应该怎么办?	154
A.13 如何使用 gfortran 编译 QIST 组件程序?	154
参考文献	155

插图索引

1.1	<i>i</i> QIST 软件包各组件的继承关系	5
2.1	<i>i</i> QIST 软件包的目录结构	12
6.1	量子杂质模型求解器组件的四种运行模式	62
6.2	分而治之的计算策略示意图	76
8.1	单带半满 Hubbard 模型的虚时格林函数 $G(\tau)$	104
8.2	单带半满 Hubbard 模型的虚频格林函数的虚部 $\Im G(i\omega)$	104
8.3	单带半满 Hubbard 模型的电子自能函数的虚部 $\Im \Sigma(i\omega)$	105
8.4	图形微扰展开序列的阶数分布 P_H	106
8.5	虚时格林函数 $G(\tau)$ 随着 U 的变化	107
8.6	虚频格林函数的虚部 $\Im G(i\omega)$ 随着 U 的变化	108
8.7	电子自能函数的虚部 $\Im \Sigma(i\omega)$ 随着 U 的变化	108
8.8	准粒子权重 Z 随着 U 的变化	109
8.9	双占据数 D 随着 U 的变化	109
8.10	单粒子谱函数 $A(\omega)$ 随着 U 的变化	110
8.11	广义相互作用的两带半满 Hubbard 模型的虚时格林函数 $G(\tau)$	113
8.12	广义相互作用的两带半满 Hubbard 模型的虚频格林函数的虚部 $\Im G(i\omega)$	114
8.13	广义相互作用的两带半满 Hubbard 模型的图形微扰展开序列的阶数分布 P_H	114
8.14	包含自旋 – 轨道耦合的三带 Hubbard 模型的虚时格林函数 $G(\tau)$	118
8.15	包含自旋 – 轨道耦合的三带 Hubbard 模型的虚频格林函数的虚部 $\Im G(i\omega)$	119
8.16	包含自旋 – 轨道耦合的三带 Hubbard 模型的图形微扰展开序列的阶数分布 P_H	119
8.17	包含晶体场劈裂的两带 Hubbard 模型的虚时格林函数 $G(\tau)$	122
8.18	包含晶体场劈裂的两带 Hubbard 模型的虚频格林函数的虚部 $\Im G(i\omega)$	122

8.19 plasmon pole 模型与 ohmic 模型示意图	124
8.20 包含动态屏蔽效应的单带半满 Hubbard 模型的虚频格林函数的虚部 $\Im G(i\omega)$	126
8.21 包含动态屏蔽效应的单带半满 Hubbard 模型的电子自能函数的虚部 $\Im \Sigma(i\omega)$	126
8.22 单带半满 Hubbard 模型的自旋 – 自旋关联函数 $\chi_{\text{spin}}(\tau)$	132
8.23 单带半满 Hubbard 模型的轨道 – 轨道关联函数 $\chi_{\text{orb}}(\tau)$	133
8.24 总的双粒子格林函数 $\chi_{\text{tot}}^{mn}(\omega, \omega', \nu)$	135
8.25 可约双粒子格林函数 $\chi_0^{mn}(\omega, \omega', \nu)$	136
8.26 不可约双粒子格林函数 $\chi_{\text{irr}}^{mn}(\omega, \omega', \nu)$	136
8.27 全顶角函数 $\mathcal{F}^{mn}(\omega, \omega', \nu)$	137
8.28 单带半满 Hubbard 模型的谱函数 $A(\omega)$	140
8.29 单带半满 Hubbard 模型实频电子自能的实部 $\Re \Sigma(\omega)$	144
8.30 单带半满 Hubbard 模型实频电子自能的虚部 $\Im \Sigma(\omega)$	144
8.31 总占据数 N 与晶体场 Δ 的相图	147
8.32 能带的占据数随着晶体场变化的情况	148
8.33 局域磁矩随着晶体场变化的情况	148
8.34 三带 Anderson 杂质模型中杂质的轨道磁化系数 χ_{orb}^{-1} 随着温度的变化	150
8.35 三带 Anderson 杂质模型中杂质的自旋磁化系数 χ_{spin}^{-1} 随着温度的变化	150

表格索引

3.1	量子蒙特卡洛杂质求解器输入文件需求表	18
4.1	<i>Azalea</i> 组件 solver.ctqmc.in 文件的输入格式	22
4.2	<i>Gardenia</i> 组件 solver.ctqmc.in 文件的输入格式	25
4.3	<i>Narcissus</i> 组件 solver.ctqmc.in 文件的输入格式	29
4.4	<i>Begonia</i> 组件 solver.ctqmc.in 文件的输入格式	32
4.5	<i>Lavender</i> 组件 solver.ctqmc.in 文件的输入格式	35

第一章

*i*QIST 简介

本章将让用户对 *i*QIST 软件包有一个初步的了解，适合此前从未接触过 *i*QIST 的用户阅读。在第2章至第6章，将着重描述 *i*QIST 的量子杂质求解器组件的具体用法，资深用户可以选择性阅读。在第7章，将介绍 *i*QIST 的辅助工具组件的用法，资深用户可以选择性阅读。第8章中包含了若干个深入浅出的计算实例以及项目实战解析，可供用户学习参考。在本文文末还提供了—个简短的附录，收集了用户经常会遇到的问题并给出解答。

1.1 什么是 *i*QIST ？

*i*QIST 是 Interacting Quantum Impurity Systems Simulating Toolkit (相互作用量子杂质系统模拟工具包) 的英文字头缩写。顾名思义，*i*QIST 是由我们独立开发的，若干个用于处理不含时相互作用量子杂质模型的软件的集合。这些软件包括了十余种量子杂质模型求解器以及配套的前处理软件和后处理程序，是动力学平均场理论框架中最为核心的部分。

*i*QIST 软件包的主要部分是黄理博士¹在中国科学院物理研究所理论室 T03 组攻读理论物理专业博士研究生期间 (2008.9 ~ 2012.5) 的研究成果之一。此软件包的开发是在戴希研究员的倡议和指导下进行的，王义林博士与杜亮博士对此亦有不少直接的贡献。*i*QIST 的开发耗费了我们无数的精力与时间，付出了极大的代价，目前正是科研成果的收获期。从情理而言，闭源是保护知识产权的最佳途径，但是为了促进相关研究领域的发展，同时扩大 *i*QIST 的学术影响力，我们做出了一个艰难的决定：将 *i*QIST 在受控下开源公开发布。我们希望此举能收到抛砖引玉的作用，同时也热切期盼收到用户的反馈与建议，推动 *i*QIST 的进一步发展。

¹亦即本用户手册的第一作者。

1.2 iQIST 的项目背景

目前动力学平均场理论是研究强关联电子系统最为强有力的方法之一。动力学平均场理论的核心要点是将相互作用的广义晶格模型自治映射到量子杂质模型中，通过迭代求解量子杂质模型，即可获取原始的广义晶格模型的性质¹⁻³。动力学平均场理论隐含着这样一个假设：假定原始晶格模型的维数趋于无穷大，那么电子的自能函数是一个与动量无关的量，亦即，

$$\Sigma(k, \omega) = \Sigma(\omega). \quad (1.1)$$

虽然动力学平均场理论仅在无穷维度系统中才是严格成立的，但是对于有限维系统，动力学平均场理论的计算结果仍然是值得信赖的。关于动力学平均场的理论细节以及诸多应用实例请参考相关的综述文献，此处不再赘述。

如上所述，动力学平均场方法的关键步骤就是迭代求解量子杂质模型。经过数十年的发展，目前已经有许多种方法可以用于求解量子杂质模型，这些方法亦被称之为量子杂质求解器。一般而言，量子杂质求解器可以分为两大类：解析型与数值型。典型的解析型量子杂质求解器包括：Hubbard-I 近似 (HIA)⁴、非相交近似 (NCA)⁵、单相交近似 (OCA)^{6,7}、涨落 - 交换近似 (FLEX) 等等，这些量子杂质求解器的优点是快速，效率非常高，缺点则是不够精确，适用范围有限。而典型的数值型量子杂质求解器则包括数值重整化群方法 (NRG)⁸⁻¹⁰、严格对角化方法 (ED)¹¹、量子蒙特卡洛方法 (QMC) 等等，这些量子杂质求解器的优点是精确可靠，缺点是计算效率低，适用范围也不够广。时至今日，最重要，最主流的量子杂质求解器显然就是量子蒙特卡洛方法。基于量子蒙特卡洛方法的量子杂质求解器可以划分为两大类：分立时间量子蒙特卡洛杂质求解器以及连续时间量子蒙特卡洛杂质求解器。

分立时间量子蒙特卡洛算法 (亦即 Hirsch-Fye 算法)¹²⁻¹⁴ 在早期的动力学平均场理论中应用得最为广泛。Hirsch-Fye 量子蒙特卡洛算法的技术要点是将虚时间轴 $[0, \beta)$ 均分为 M 个时间片段 $\Delta\tau = \beta/M$ ，对于每个时间片段 i 都应用 Hubbard-Stratonovich 变换。以单带 Anderson 杂质模型为例：

$$e^{-\Delta\tau U[n_\uparrow n_\downarrow - (n_\uparrow + n_\downarrow)/2]} = \frac{1}{2} \sum_{s_i = \pm 1} e^{\lambda s_i (n_\uparrow - n_\downarrow)}, \quad (1.2)$$

其中：

$$\lambda = \operatorname{arccosh} \left[\exp \left(\frac{1}{2} \Delta\tau U \right) \right]. \quad (1.3)$$

式中 s_i 被称为附加的 Ising 场变量。这样量子杂质问题转化为随时间变化的附加 Ising 场上的无相互作用费米子模型，可以用蒙特卡洛算法严格求解。Hirsch-Fye 量子蒙特卡洛算法既能处理强关联电子体系，也能处理弱关联电子体系，均能获得十分精确的结果。它的主要缺点是计算速度比较慢，而且只能获得虚时格林函数，为了获得有物理意义的结果，必须先对其进行解析延拓。主要存在着三个困难限制了 Hirsch-Fye 算法的应用：首先 Hirsch-Fye 算法要求对虚时间轴进行均匀分立化，这样会引入 $O(\tau^2)$ 量级的系统误差。为了减少系统误差， $\Delta\tau$ 需要尽可能的小，但是计算量的增长是 $O(M^3)$ 的量级，因此这是个两难的选择，需要在计算精度和计算时间之间做出折衷的选择²。其次，当相互作用强度较大，或者

²N. Blumer 等人曾提出一种外推方法，将普通的蒙特卡洛计算结果外推至 $\tau \rightarrow 0$ 的情况 (参见 [arXiv:0712.1290](https://arxiv.org/abs/0712.1290))。

系统温度较低时，Hirsch-Fye 算法要达到平衡比较困难。最后，当相互作用的形式比较复杂时³，附加 Ising 场变量的数目将急剧增加，这也会给蒙特卡洛抽样算法带来巨大的困难¹⁴。

连续时间量子蒙特卡洛 (continuous-time quantum Monte Carlo, CT-QMC) 算法的基本思想是避免时间分立化步骤，直接对图形微扰展开项进行抽样。最早利用这一想法的可能是 Handscomb 方法^{15,16}以及它的推广——随机序列展开 (stochastic series expansion, SSE) 算法¹⁷。SSE 算法将配分函数依据 βH 因子进行 Taylor 展开。这一算法成功地用于量子磁体 (quantum magnets) 的模拟计算。但是 SSE 算法要求哈密顿量的频谱是束缚的，因此在模拟玻色系统时，需要对 Hilbert 空间进行截断，在模拟费米系统时，负符号问题十分严重。

当前使用的连续时间量子蒙特卡洛杂质求解器主要是受到 Prokof'ev 等人^{18,19}工作的启发。他们开创性的研究表明：通过随机抽样配分函数的图形微扰展开项，玻色型的晶格模型可以用连续时间量子蒙特卡洛算法求解。他们将连续时间量子蒙特卡洛算法推广至任意存在着连续变量的系统，又称之为图形量子蒙特卡洛 (diagrammatic quantum Monte Carlo, DQMC) 方法^{20,21}。在图形量子蒙特卡洛方法中，完全消除了由于时间分立化以及 Suzuki-Trotter 分解所带来的系统误差，因此计算效率较之分立时间量子蒙特卡洛方法有巨大的提升。应用该方法已经完全解决了与 unfrustrated 的玻色型晶格模型有关的计算问题。

连续时间量子蒙特卡洛算法在玻色型晶格问题上的成功促使人们将其拓展至费米型的晶格问题中去。这方面的先导工作是由 Rombouts 等人²²完成的。对于玻色型的体系，所有的展开图都有相同的符号，然而对于费米型的体系，由于反对易关系的原因，不同的图形其符号并不一样，或正或负，因此每次都抽样单个的图会导致十分严重的负符号问题。解决之道是把同类的图合并，每次抽样一系列的图，在数值上表现为对行列式 (determinant) 的操作。不幸的是 Rombouts 及其合作者在他们所研究的晶格模型中仍然发现了极为严重的负符号问题。实际上他们所使用的算法²²仅能处理密度 - 密度形式的相互作用。这样糟糕的结果使得人们对于连续时间量子蒙特卡洛算法在费米型晶格问题上的应用持悲观的态度，很快这一算法便乏人问津了。

连续时间量子蒙特卡洛算法的转机出现于 2004 年与 2005 年之间²³。随着动力学平均场理论的深入发展，人们迫切需要一种适应范围广，性能优越且稳定的量子杂质模型求解器。人们很快就意识到，量子杂质模型的负符号问题远没有量子晶格模型那么严重，在某些特殊情况下，负符号问题甚至不再存在，因此人们开始回头重新审视连续时间量子蒙特卡洛算法。Rubtsov 等人首先提出了弱耦合展开算法 (CT-INT)^{24,25}，这是一个突破性的进展。紧接着，Werner 与 Millis 等人提出了与之对应的强耦合展开算法 (CT-HYB)²⁶⁻³¹。Gull 等人结合了附加场技术以及弱耦合展开算法，提出了第三种连续时间量子蒙特卡洛杂质求解器 (CT-AUX)³²。Otsuki 与 Werner 等人还将弱耦合展开算法推广至 Coqblin-Schrieffer 模型以及 Kondo 模型中 (CT-J)³³。

由于避免了时间轴分立化的步骤，因此连续时间量子蒙特卡洛算法中没有引入系统性的误差，比分立时间量子蒙特卡洛算法更为精确可靠，并且其计算速度要比分立时间量子蒙特卡洛算法快很多，能够有

³例如包含自旋翻转项或者是对跃迁项时。

效处理多带体系。由于具有上述优点，虽然仍需要对连续时间量子蒙特卡洛杂质求解器的计算结果进行解析延拓以获得真实的物理可观测量，它已经成为现代动力学平均场理论中最为主流的量子杂质模型求解器^{34,35}，在许多领域都得到了广泛的应用²³，同时也取得了巨大的成功。

应用连续时间量子蒙特卡洛算法，人们能够精确地模拟 Kondo 晶格模型³⁶；首次定量研究具有旋转不变性相互作用的多带模型^{24,27,31,37-40}；能够高效求解多格点的杂质模型，为在动力学平均场理论框架内包含空间涨落效应奠定了坚实的基础⁴¹⁻⁴⁸；能够支持在密度泛函理论结合动力学平均场方法的框架内快速模拟计算实际的关联电子材料的物理性质⁴⁹。连续时间量子蒙特卡洛算法还能够高效地测量四点(高阶)关联函数(four-point correlation function)，这对于研究系统的磁化系数，相边界十分重要。而且四点关联函数也是对偶费米子方法^{50,51}与动力学顶角近似⁵²等动力学平均场理论扩展的关键组成部分，因此不难预见，连续时间量子蒙特卡洛算法在这些新领域将会大放异彩。在纳米科学领域，连续时间量子蒙特卡洛算法被用于研究过渡金属团簇在金属表面的物理属性⁵³。一些在以前由于方法上的限制而无法研究的物理问题现在也得到了初步解决，例如重费米子材料中的准粒子动力学以及热力学交叉点⁵⁴⁻⁵⁷。对于费米子在光学晶格中的问题也已开始理论研究^{58,59}。将连续时间量子蒙特卡洛算法拓展至非平衡体系实时系统中的研究⁶⁰⁻⁶⁵也在紧锣密鼓地进行当中。

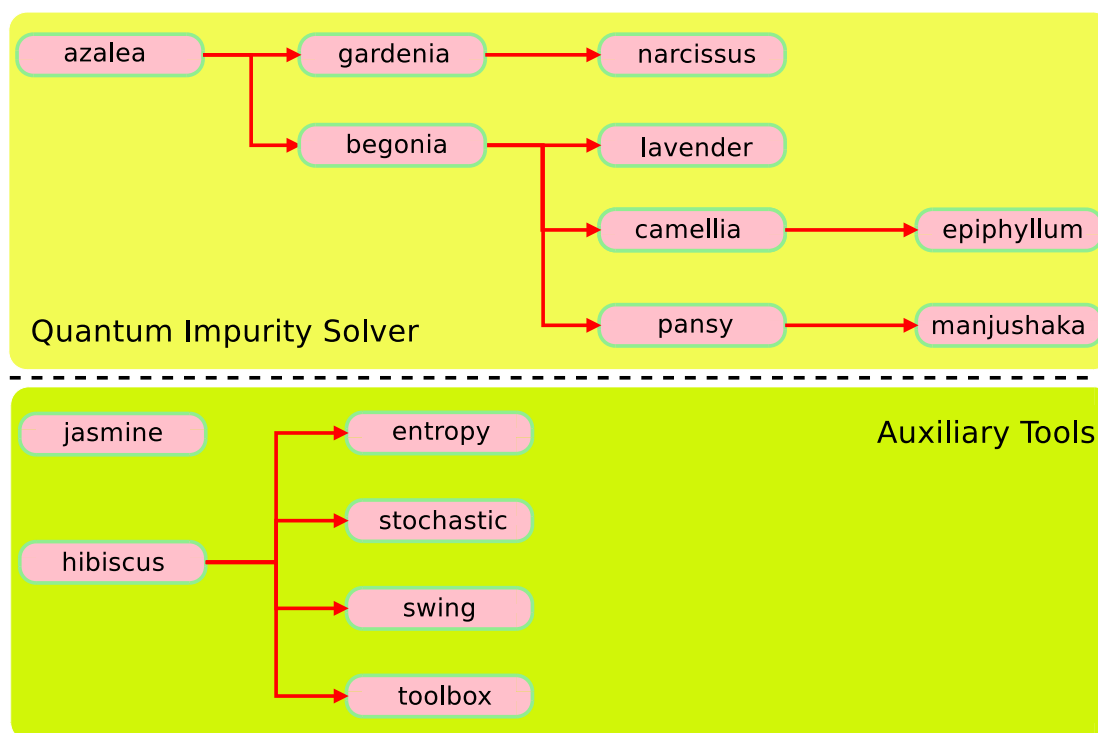
应该特别指出的是，虽然连续时间量子蒙特卡洛算法极大地拓展了量子杂质模型求解器的适用范围，取得了巨大的成功，但是它们并不能解决费米子符号问题。就目前所知，符号问题是物理的，不可避免的。符号问题的严重性与否将最终决定连续时间量子蒙特卡洛算法的适用边界。

早在 2008 年之前，黄理就已经独立实现了分立时间量子蒙特卡洛杂质求解器(Hirsch-Fye 算法，Daisy 项目)以及弱耦合版本的连续时间量子蒙特卡洛杂质求解器(CT-INT 算法，Sakura 项目)。2009 年中，在戴希研究员的推动促进下，他对连续时间量子蒙特卡洛杂质求解器做了深入的调研，了解到强耦合版本的连续时间量子蒙特卡洛杂质求解器无论是在适用范围还是在计算效率上都更具优势，因此决定开发强耦合版本的程序，并重新整理此前开发的程序。2009 年末，开发工作正式开始，经过近三年时间的努力，在王义林与杜亮等同学的协助下，一共开发出十余种量子蒙特卡洛杂质求解器，并配套研发了一系列的前处理与后处理工具程序，形成了一套比较完备的工具链。2012 年中，黄理博士研究生毕业，离开中科院物理研究所，回到位于四川绵阳的中国工程物理研究院工作。他在工作之余，重新审视此前所做的项目，尤其是他所开发的多种量子杂质模型求解器，觉得应该将它们融合为一个有机的整体，以便发挥更大的作用，这就是 iQIST 项目的由来。

1.3 iQIST 的组织架构

在 iQIST 软件包中，我们不但实现了最为成熟的 Hirsch-Fye 量子蒙特卡洛杂质求解器，还实现了最为先进的连续时间量子蒙特卡洛杂质求解器(包括 CT-INT 与 CT-HYB 算法)以及相应的前处理与后处理脚本程序。iQIST 软件包的框架十分庞大，涵括了许许多多的组件⁴，但是由于人力资源以及时间的

⁴当前版本的 iQIST 软件包包含了如下组件: Azalea、Gardenia、Narcissus、Begonia、Lavender、Camellia、Epiphyllum、Pansy、Manjushaka、Sakura、Daisy、Jasmine、Hibiscus 等等。

图 1.1: *iQIST* 软件包各组件的继承关系

关系，许多组件并不是十分成熟完备，暂不适宜公开发布，仅在开发团队内部进行测试。因此目前公开发布的 *iQIST* 软件包仅仅包含了如下组件 (参见图1.1)⁵:

Azalea

Azalea 是采用 CT-HYB 算法的连续时间量子蒙特卡洛杂质求解器，是所有 CT-QMC 杂质求解器的原型，也是 *iQIST* 中第一个开发完毕的连续时间量子蒙特卡洛杂质求解器。该杂质求解器基于 segment 表象算法²⁶，仅仅支持具有密度 – 密度相互作用形式的哈密顿量，运行性能极其稳定高效。

Gardenia

Gardenia 是采用 CT-HYB 算法的连续时间量子蒙特卡洛杂质求解器，它是在 *Azalea* 的基础之上发展而来。它相对于 *Azalea* 的最大改进就是可以选择采用正交多项式表象以及内核多项式表象⁶⁶来改善电子自能函数以及虚时格林函数的测量精度。同时 *Gardenia* 的功能也大为扩充了，能够测量自旋 – 自旋关联函数、轨道 – 轨道关联函数、双粒子格林函数以及顶角函数等等。*Gardenia* 的运行效率要明显弱于 *Azalea*，但是相对于 Hirsch-Fye 算法 (亦即 *Daisy* 组件) 而言，仍属十分高效。

⁵目前暂不公开发布的组件有: *Camellia*、*Epiphyllum*、*Pansy*、*Manjushaka*、*Sakura*、*Daisy*。欲知详情或者是申请软件抢先试用，请联系黄理 (huangli712@gmail.com)。

Narcissus

Narcissus 是采用 CT-HYB 算法的连续时间量子蒙特卡洛杂质求解器，它是在 *Gardenia* 的基础之上发展而来。它相对于 *Gardenia* 的最大改进是支持动态屏蔽效应²⁹ 以及 Hubbard-Holstein 模型²⁸ 的计算。但是在 *Narcissus* 中不再支持 *Gardenia* 与 *Azalea* 杂质求解器都支持的 GLOBAL SWAP 算法⁶，这意味着 *Narcissus* 杂质求解器不支持研究极端高温体系。在考虑动态屏蔽效应或是研究 Hubbard-Holstein 模型时，*Narcissus* 的运行效率要明显弱于 *Gardenia*。但是在研究普通系统时，*Narcissus*、*Gardenia* 与 *Azalea* 组件的运行效率没有明显的差别。总体而言，在运行效率方面， $Azalea \geq Gardenia \geq Narcissus$ 。

Begonia

Begonia 是采用 CT-HYB 算法的连续时间量子蒙特卡洛杂质求解器，它也是在 *Azalea* 的基础之上发展而来。该杂质求解器基于广义矩阵表象²⁷，支持任意相互作用形式的哈密顿量，运行性能较之 *Azalea* 有较大的降低。*Begonia* 采用了稀疏矩阵技术以及分而治之的策略来提高计算效率，但是并没有采用好量子数以及分块矩阵乘法技术³¹，因此 *Begonia* 并不适用于研究较大的体系，五带模型就是它的极限了。

Lavender

Lavender 是采用 CT-HYB 算法的连续时间量子蒙特卡洛杂质求解器，它是在 *Begonia* 的基础之上发展而来。它相对于 *Begonia* 的最大改进就是可以选择采用正交多项式表象以及内核多项式表象⁶⁶ 来改善虚时格林函数的测量精度。同时 *Lavender* 还支持双粒子格林函数以及顶角函数的测量。经过修正的 *Lavender* 还可以用于研究含自旋 - 轨道耦合效应的系统。同 *Begonia* 类似，*Lavender* 也不适用于研究较大的体系，五带 Hubbard 模型是它的极限。

Jasmine

Jasmine 是前处理程序，主要作用是对角化原子问题，产生 atom.cix 文件，与 *Begonia*、*Lavender*、*Camellia*、*Epiphyllum*、*Pansy*、*Manjushaka* 等程序配合使用。目前 *Jasmine* 并不支持自旋 - 轨道耦合效应 (SOC)，如果用户的模型需要考虑 SOC，那么必须采用杜亮博士提供的专用版本的 rambutan 程序。

Hibiscus

Hibiscus 是一大类前处理与后处理程序的集合，须与各种量子蒙特卡洛杂质求解器配合使用。它包含 entropy、stochastic、swing 与 toolbox 等四个组件。其中 entropy 组件是最大熵程序⁶⁷，用于虚时格林

⁶该算法由戴希研究员提出。

函数的解析延拓⁷。stochastic 组件是随机解析延拓程序⁶⁸，同样用于虚时格林函数的解析延拓。swing 组件用于电子自能函数的解析延拓⁸。toolbox 组件包含了十余个前/后处理小程序。

1.4 *iQIST* 的主要功能

iQIST 项目的主要开发语言为 Fortran 90 语言，有部分后处理程序应用了 Python⁹。开发平台为 GNU/Linux 系统，采用 Bazaar 版本控制软件来控制开发进程。*iQIST* 的主要程序均支持 MPI 并行计算，所有程序都经过了高度的优化，追求运行速度的极限。*iQIST* 的主要程序均经过反复测试，其可靠性也得到多次验证。

iQIST 的量子杂质求解器组件¹⁰可以求解如下的量子杂质模型：

- 1 ~ 7 带 Hubbard 模型 (*Begonia* 与 *Lavender* 仅仅支持五带及五带以下的系统)
- 任意强度的 Coulomb 相互作用 U 以及 Hund 交换作用 J
- 动态屏蔽效应 (仅仅 *Narcissus* 支持)
- Hubbard-Holstein 模型 (仅仅 *Narcissus* 支持)
- Bethe 晶格，可扩充至其它晶格模型
- 密度 – 密度相互作用
- 具有旋转不变性的广义相互作用 (除了 *Azalea*、*Gardenia* 与 *Narcissus* 外均支持)
- 自旋 – 轨道耦合作用 (仅仅 *Lavender* 支持¹¹)
- 系统温度 $0.5 < \beta < 400$ (仅仅 *Azalea* 与 *Gardenia* 支持 $\beta < 5$)
- Krylov 子空间迭代算法 (仅仅 *Camellia* 与 *Epiphyllum* 支持)
- Newton-Leja 多项式插值算法 (仅仅 *Camellia* 与 *Epiphyllum* 支持)
- 子空间分块算法 (仅仅 *Pansy* 与 *Manjushaka* 支持)
- 正交多项式算法 (除了 *Azalea*、*Begonia* 与 *Pansy* 外均支持)
- 内核多项式算法 (除了 *Azalea*、*Begonia* 与 *Pansy* 外均支持)
- MPI 并行加速

⁷即由 $G(\tau)$ 得到 $G(\omega)$ 。

⁸即由 $\Sigma(i\omega)$ 得到 $\Sigma(\omega)$ 。

⁹指 *Hibiscus* 组件中的 swing 程序。

¹⁰仅指基于 CT-HYB 算法的量子杂质求解器组件，因此 *Daisy* 与 *Sakura* 组件不包括在内。

¹¹源程序需要进行额外的修改。

- OpenMP 并行加速 (仅仅 *Epiphyllum* 支持)
- 可独立运行也可与 LDA + DMFT 接口程序联用

*i*QIST 的量子杂质求解器组件¹²可以输出如下的观测量:

- 虚时格林函数 $G(\tau)$, 虚频格林函数 $G(i\omega)$
- 虚时 weiss 函数 $\mathcal{G}(\tau)$, 虚频 weiss 函数 $\mathcal{G}(i\omega)$
- 虚时杂化函数 $\Delta(\tau)$, 虚频杂化函数 $\Delta(i\omega)$
- 虚频自能函数 $\Sigma(i\omega)$
- 原子虚频格林函数 $G_{\text{atom}}(i\omega)$, 原子虚频自能函数 $\Sigma_{\text{atom}}(i\omega)$
- 轨道占据数 $\langle n_\alpha \rangle$, 双占据数 $\langle n_\alpha n_\beta \rangle$
- 动能 E_{kin} 、势能 E_{pot} 、总能 E_{tot} 、平均有效磁矩 $\langle S_z \rangle$
- 微扰展开项分布概率 P_{H}
- 原子组态分布概率 P_{T}
- 自旋 – 自旋关联函数 $\langle S_z(0)S_z(\tau) \rangle$ (仅仅 *Gardenia* 与 *Narcissus* 支持)
- 轨道 – 轨道关联函数 $\langle N_\alpha(0)N_\alpha(\tau) \rangle$ (仅仅 *Gardenia* 与 *Narcissus* 支持)
- 双粒子虚时格林函数 g_2 (仅仅 *Gardenia*, *Narcissus* 与 *Lavender* 支持)
- 顶角函数 γ_4 (仅仅 *Gardenia*, *Narcissus* 与 *Lavender* 支持)

1.5 *i*QIST 的版本历史

v0.01, 2004 年, 开始项目初始开发。

v0.02, 2005 年, 实现单带 Hirsch-Fye 量子蒙特卡洛杂质求解器。

v0.03, 2006 年, 实现多带 Hirsch-Fye 量子蒙特卡洛杂质求解器, *Daisy* 组件的前身。

v0.04, 2007 年, 实现弱耦合连续时间量子蒙特卡洛杂质求解器, *Sakura* 组件的前身。

v0.05, 2009 年, 实现强耦合连续时间量子蒙特卡洛杂质求解器, *Azalea* 组件与 *Hibiscus* 组件。

v0.06, 2010 年, 实现强耦合连续时间量子蒙特卡洛杂质求解器, *Begonia* 组件与 *Jasmine* 组件。

v0.07, 2011 年, 实现了 *Gardenia*、*Narcissus* 与 *Lavender* 等组件。

¹²仅指基于 CT-HYB 算法的量子杂质求解器组件, 因此 *Daisy* 与 *Sakura* 组件不包括在内。

v0.10, 2012 年 8 月, 提出创建 *iQIST* 软件包, 开始编写本文档。

v0.11, 2012 年 10 月, 用户文档基本成型。

v0.12, 2012 年 11 月, 用户文档分发相关用户修订校阅。

v0.13, 2012 年 12 月, *iQIST* 软件包基本成型, 准备公开发布。

1.6 *iQIST* 的使用条款

如果你需要使用 *iQIST*, 那么必须遵守如下用户条款 (十分宽松的, 不必担心):

- *iQIST* 是开源软件, 用户可以自由编译、链接、使用、复制、拷贝、分发、修改、破坏、删除它。
- 用户可以任意修改、复制、调用 *iQIST* 的代码片段或者是函数。
- 如果用户修改了 *iQIST* 的源代码, 希望能及时反馈回开发者处。
- 如果用户发现了任何的 bug, 或者是有任何的建议, 欢迎与开发者联系。
- 如果用户需要在 *iQIST* 的基础上重新发布修改过的程序, 那么禁止使用 *iQIST* 或者是类似的名称。
- 尽管开发者对 *iQIST* 的可靠性有足够的信心, 但是开发者并不能保证计算结果的绝对正确性。
- 开发者一般不对用户提供无偿的技术支持。
- 对于用户使用 *iQIST* 所造成的任何直接/间接后果, 开发者一概不予负责。
- 如果用户在工作中使用了 *iQIST*, 希望能够引用如下的论文或者是把开发者列为共同作者。

Li Huang, Yilin Wang, Liang Du and Xi Dai,

User's Guide for Interacting Quantum Impurity Systems Simulating Toolkit, 2012

- 上述条款的最终解释权为黄理 (huangli712@gmail.com) 所有。

第二章

*i*QIST 的安装配置

本章介绍如何获取、编译、安装与设置 *i*QIST 软件包的各个组件，在下一章将介绍如何运行 *i*QIST。

2.1 获取 *i*QIST

也许您已经由中国科学院物理研究所理论室 T03 组的某位学生那里或者是其它途径获得了 *i*QIST 的某个组件的源程序，甚至还积累了不少糟糕的或者是良好的用户体验。但是需要郑重提醒您的是: *i*QIST 的最新的最全的最权威的最官方的绝对无删减版本只能通过下述途径获取。

email: huangli712@gmail.com

或者是

email: huangli712@yahoo.com.cn

出于某种考虑，我们一般不将 *i*QIST 对外完整发布，而只是以组件的形式独立发布 *i*QIST 的某个或者是某些部分。如果您有特别的需求，请与我们联系。

请注意，如果您已经获取了 *i*QIST 的源代码，未经作者书面或者口头许可，建议 (仅仅是建议!) 您勿随意以任何方式扩散此程序，请尊重作者的辛苦付出。用户需要遵守的协议请参阅本文的第1.6节。

如果您已经是 *i*QIST 的用户，建议您将您的名字、工作单位以及通联方式发送给我们，以便我们做用户统计。这并不是强制性的，仅仅只是一个建议。

2.2 解压缩 *i*QIST

*i*QIST 的所有源文件、文档、例子都包含在名为 iqist.tar.gz 的压缩文件中，用一般的解压缩工具可将其解开。以 Linux 系统为例，您可以用下述命令将 iqist.tar.gz 文件解压缩。

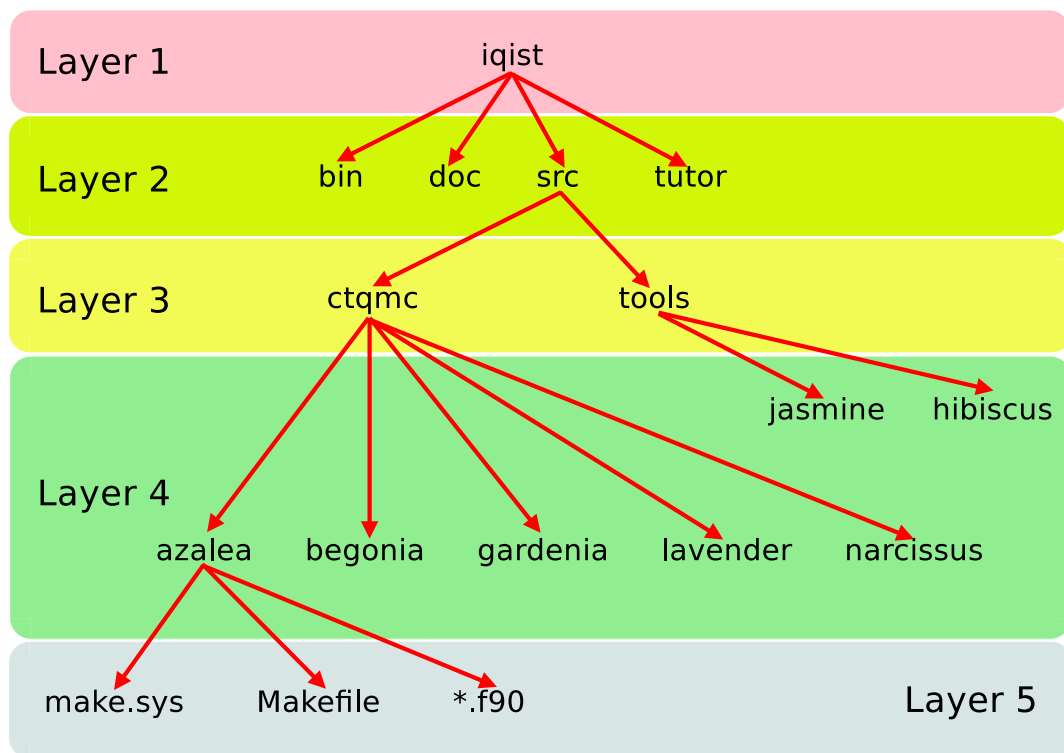


图 2.1: iQIST 软件包的目录结构

```
$ tar xvfz iqist.tar.gz
```

如果您获取的仅仅是 iQIST 的某个组件的源文件，例如 `azalea.tar.gz`¹，那么也可以用相同的命令对其进行解压：

```
$ tar xvfz azalea.tar.gz
```

2.3 iQIST 的目录结构

iQIST 的目录结构如图 2.1 所示。每个文件夹具体存放的内容如下：

- bin: 存放编译后可执行程序
- doc: 存放用户手册与说明文档
- src: 存放不同的组件的源程序
- tutor: 教程，提供简单的例子

¹ Azalea 组件的源程序。

在 `src` 文件夹中，又内含 `ctqmc` 与 `tools` 两个子文件夹。在 `ctqmc` 文件夹中，包含着 *Azalea*、*Gardenia*、*Narcissus*、*Begonia* 与 *Lavender* 等组件的源程序，每个组件都有单独的文件夹。在 `tools` 文件夹中，包含着 *Jasmine* 与 *Hibiscus* 组件的源程序，在此目录下，还有 `pysci.py` 脚本程序。

在组件的文件夹中，有三类型文件，分别是：`make.sys`，`Makefile`，与 `*.f90`²。其中 `*.f90` 为源代码文件，不可修改。`Makefile` 为编译主控文件，由 `make` 程序处理，不可修改。而 `make.sys` 文件为编译设置文件，需要用户修改以适合系统的当前环境。

警告! 除非您确信自己正在做些什么，并且由此可能带来的后果，否则请不要修改任何 *iQIST* 组件的源代码。

2.4 *iQIST* 组件的编译链接

iQIST 目前并不提供一个统一的编译脚本，各个组件是分别编译链接的，也就是说，用户无法简单地敲一个命令就把 *iQIST* 编译完毕。但是用户也无须感到恐惧，*iQIST* 的编译链接是十分简单的。

2.4.1 基本编译环境

我们首先来看 *iQIST* 的编译环境。在正式编译 *iQIST* 之前，用户必须确保自己的系统符合以下最低要求：

- 操作系统为 UNIX/Linux 或者是 Mac OS X，尽量升级为最新或者是较新的版本
- 如果操作系统为 Mac OS X，请务必提前安装 Xcode 套件
- 如果操作系统为 UNIX/Linux，请务必提前安装 GCC 套件
- Python 要求版本 2.6+，安装 `scipy`, `numpy`, `matplotlib` 库³
- 安装 `f2py`⁴
- 安装 Intel Fortran Compiler，要求版本 11.0+⁵
- 安装 MPI 支持，MPICH1、MPICH2、OPENMPI，或者是各种定制的 MPI 实现均可⁶

²在 *Hibiscus* 组件的目录下，还存在着以 `py` 为扩展名的 Python 脚本程序。

³参见 `scipy` 与 `numpy` 的主页 (URL:<http://www.scipy.org/>) 与 `matplotlib` 的主页 (URL:<http://matplotlib.org/>)。

⁴该工具负责将 Python 脚本与 Fortran 子程序耦合链接在一起 (URL:<http://www.scipy.org/F2py>)，*Hibiscus* 组件的 `swing` 程序需要用到它。

⁵关于 Fortran 编译器，我们推荐的是 Intel 提供的 Fortran 编译器。从理论上说，我们的 Fortran 程序没有使用任何 Intel Fortran 独有的语法，GNU 的 `gfortran` 或者 PGI 的 `pgf90` 应该也是可用的，但是我们并没有对此做过测试，欢迎用户提供测试报告。将 *iQIST* 移植到更多的软硬件平台上是我们努力的目标之一。

⁶推荐采用 ANL 开发的 MPICH2 的稳定版本 (URL:<http://www.mcs.anl.gov/research/projects/mpich2/>)。

- 安装 BLAS 与 LAPACK 数学库⁷
- 安装 vim、gnuplot 等工具程序⁸

2.4.2 编译过程详解

下面我们来编译 iQIST 的各个组件。此处以 .Azalea 组件为例来说明编译过程，其余组件的编译过程是完全类似的。首先进入到 .Azalea 组件的源代码目录：

```
$ cd iqist/src/ctqmc/azalea
```

然后使用 vim 编辑器编辑 make.sys 文件，设定编译环境：

```
$ vim make.sys
```

典型的 make.sys 文件如下所示：

```
1 # fortran compiler and linker
2 #-----
3 F90      = mpif90
4 LINKER = $(F90)
5
6 # fortran preprocessor options, common setting
7 #-----
8 MPI      = -DMPI
9 OMP      = #-openmp
10 FPP      = -fpp
11 CPP      = $(FPP) $(MPI) $(OMP)
12
13 # machine tuning options, just for my laptop: iris system
14 #-----
15 GPROF    = #-pg
16 CHECK    = -warn all #-check all -traceback -g
17 CDUMP    = -vec-report2 -openmp-report2 -nogen-interfaces
18 LEVEL    = -O3 -xHost -unroll-aggressive -align all
```

⁷关于线性代数库有很多的选择。最简单最省事的选择是直接采用 Intel 提供的 MKL 数学函数库。性能最差的是选择 netlib 上所提供的 LAPACK 和 BLAS 的标准参考实现库 (URL:<http://www.netlib.org/lapack/>)。如果追求性能的极致的话，BLAS 的实现建议选择 GotoBLAS2 或者是 OpenBLAS。在此隆重地向各位用户推介国人自主开发的 OpenBLAS。因为 GotoBLAS2 已经无人维护，它不支持最新的处理器平台以及最新的指令集。因此除非用户的机器仍旧是一些老古董，否则 GotoBLAS2 已经不是一个最好的选择了。在 GotoBLAS2 1.1.3BSD 版本的基础上，中国科学院软件研究所并行计算实验室的张先轶等人分支出一个新的版本，继续对其进行优化改进，并命名为 OpenBLAS(URL:<https://github.com/xianyi/OpenBLAS>)，OpenBLAS 修正了不少 GotoBLAS2 中的 bug，新支持 LLVM/clang 编译器。除了最新的 Intel Sandy Bridge 平台外，OpenBLAS 对于国产的龙芯 3A/3B 有着很好的支持。

⁸参见 vim 的主页 (URL:www.vim.org) 与 gnuplot 的主页 (URL:www.gnuplot.info)。

```

19 MTUNE = -mtune=core2
20
21 # flags for compiler and linker
22 #-----
23 FFLAGS = -c $(CPP) $(CHECK) $(CDUMP) $(LEVEL) $(MTUNE) $(GPROF)
24 LFLAGS = -static $(OMP) $(GPROF)
25
26 # linear algebra library , lapack and blas
27 #-----
28 LIBS = -L. -llapack -lblas #-lguide
29 LIBS = -L. -llapack -lgoto #-lguide

```

我们对上面的 `make.sys` 文件稍作解释。

1. `#` 符号后面出现的内容表示注释，会被忽略掉。
2. 第 3 行设定 Fortran 90 编译器，第 4 行设定链接器。
3. 第 8 行打开 MPI 并行支持，如果只编译串行程序，可以注释掉该行。
4. 第 9 行打开 OpenMP 并行支持，目前公布的 *iQIST* 组件暂不支持该技术，不要改动它。
5. 第 10 行打开 Intel Fortran Compiler 内置的预处理器，这是必须的，不要改动它。如果你的编译器不属于 Intel Fortran Compiler，那么此选项可能会有所不同。
6. 第 15 行控制是否在运行时刻采样监测运行性能，如果打开该选项将会降低性能。
7. 第 16 行，`-warn all` 表示在编译时刻输出所有的警告信息；`-check all` 表示在运行时刻输出所有监测信息，如果遇到数组越界这类型的异常会报错退出，该选项会严重降低运行效率；`-traceback` 选项表示在输出运行时刻出错信息时向后追溯；`-g` 选项表示将调试符号包含在可执行程序中，那么当程序报错时，会给出具体出错的行号。如果你的编译器不属于 Intel Fortran Compiler，那么上述选项可能会有所不同。
8. 第 18 行，`-O3`、`-xHost` 等选项都是用来设置针对当前平台产生最优化的代码。如果你的编译器不属于 Intel Fortran Compiler，那么上述选项可能会有所不同。
9. 第 19 行设置针对 `core2` 架构产生最优的代码，如果你的处理器不属于 Intel 出品，或者即使是 Intel 出品但是也不属于 `core2` 架构的，必须注释掉该行或者是查阅编译器手册，将 `core2` 替换为合适的取值。我们的开发平台为 `core2` 架构，因此将其设为缺省值。如果你的编译器不属于 Intel Fortran Compiler，那么上述选项可能会有所不同。
10. 第 23 行设置编译参数。
11. 第 24 行设置链接参数，`-static` 表示产生静态程序，Mac OS X 系统不支持此选项。
12. 第 28 ~ 29 行，设置 BLAS 数学库以及 LAPACK 数学库的路径并将它们加入到链接选项中。选用高度优化的 BLAS 与 LAPACK 库，例如 Intel MKL、GotoBLAS2、OpenBLAS 将会显著提高 *iQIST*

的运行效率。注意: 实际上只有第 29 行的设置起作用, 第 28 行的设置被覆盖掉了。

编辑好 `make.sys` 文件后, 退出 `vim` 程序。然后在命令行中用 `make` 命令编译:

```
$ make
```

编译成功后, 将会产生名为 `ctqmc` 的可执行程序。如果编译失败, 请检查出错信息, 重新编辑 `make.sys` 文件调整编译环境。然后在命令行中输入 `make clean` 命令清除编译痕迹:

```
$ make clean
```

接下来可再次用 `make` 命令进行编译。

注意: 我们在上文所述的编译设置仅仅只是一个示例, 仅仅只适用于开发者所使用的软硬件系统。为了能够正确编译 *iQIST*, 用户须根据实际情况对 `make.sys` 中的缺省设置进行微调。

2.5 *iQIST* 组件的安装设置

按照上节中的介绍, 依次编译好 *iQIST* 的各个组件后, 用户可将可执行程序统一拷贝至 `iqist/bin` 目录下, 然后将 `iqist/bin` 目录加入到系统路径中, 这样就完成了所有的安装设置工作。注意到各个量子杂质求解器组件的可执行程序都是 `ctqmc`, 在拷贝到 `iqist/bin` 目录之前请自行重命名以防止覆盖。

第三章

*i*QIST 的运行

本章讲述如何运行 *i*QIST 的量子杂质求解器组件。

3.1 搭建 *i*QIST 运行平台

*i*QIST 可以在多种软硬件平台上执行。经过测试，*i*QIST 至少可以在下述硬件平台上正常运行：

- HP 6730b laptop + Ubuntu Linux 10.04
- MacPro Workstation + Mac OS X 10.6
- Linux Cluster + Cent OS 5.0
- DAWN4000 System
- SHENTENG7000 System
- TIANHE-1 System

上述硬件平台从笔记本到工作站，从小集群到大型机，涵盖的范围十分广，并且软件平台也是多种多样，由此可见 *i*QIST 的移植性适用性非常好。

在编译安装成功 *i*QIST 后，无须特别的设置，即可执行它的各种组件。但是在运行 *i*QIST 的量子杂质求解器组件之前，请确保以下条件均具备：

1. 确保 Intel Fortran Compiler 的运行时刻库在系统路径中：

```
$ source /opt/intel/Compiler/11.1/072/bin/ifortvars.sh intel64
```

表 3.1: 量子蒙特卡洛杂质求解器输入文件需求表

	<i>Azalea</i>	<i>Gardenia</i>	<i>Narcissus</i>	<i>Begonia</i>	<i>Lavender</i>
solver.ctqmc.in	Y ^a	Y	Y	Y	Y
solver.eimp.in	Y	Y	Y	Y	Y
solver.hyb.in	Y	Y	Y	Y	Y
solver.anydos.in	Y	Y	Y	Y	Y
solver.ktau.in	N ^b	N	Y	N	N
atom.cix	N	N	N	Y ^c	Y

^aY 表示该文件是可选的。

^bN 表示该文件将被忽略。

^cY 表示该文件是必须的。

如果用户没有使用 Intel Fortran Compiler，而是使用 gfortran 或者是 PGI Fortran Compiler，那么这一步可以略过。

2. 确保 MPI 实现的可执行程序在系统路径中:

```
$ export PATH=/usr/local/mpich/bin:$PATH
```

如果用户打算在串行模式下执行量子杂质模型求解器，那么可以略过此步骤。

3. 确保 MPI 环境的守护进程已经在后台运行:

```
$ mpd &
```

如果用户打算在串行模式下执行量子杂质模型求解器，那么可以略过此步骤。此外，由于实现机制不一样，某些 MPI 实现并不要求启动守护进程，此时也可以略过此步骤。具体情况请参阅 MPI 实现的相关文档。

4. 确保系统中的 SSH 服务已经打开。如果用户打算在串行模式下执行量子杂质模型求解器，那么可以略过此步骤。

3.2 创建 *iQIST* 输入文件

iQIST 的量子杂质模型求解器组件需要如下几种输入文件:

- solver.ctqmc.in, 包含所有控制参数，参见第4.1节
- solver.eimp.in, 包含杂质能级 E_{imp} 以及对称矩阵，参见第4.2节

- solver.hyb.in, 包含初始的杂化函数 $\Delta(i\omega)$, 参见第4.3节
- solver.anydos.in, 包含晶格模型的态密度 $N(\epsilon)$, 参见第4.4节
- solver.ktau.in, 包含双积分推迟作用函数 $\mathcal{K}(\tau)$, 参见第4.5节
- atom.cix, 包含原子组态的信息, 参见第4.6节

那么很显然, 对于不同的量子杂质模型求解器, 所需要的输入文件是完全不同的。具体的对应情况请参见表3.1。

对于 solver.ctqmc.in 文件与 solver.eimp.in 文件, 你可以利用 vim 程序或者其它任意字处理软件手工编辑标准输入文件, 使其符合实际需求, 也可以利用 pysci.py 程序辅助产生全新的 solver.ctqmc.in 文件与 solver.eimp.in 文件¹。

solver.hyb.in 文件通常由上次的输出结果 (solver.hyb.dat) 转换而来, 或者是由其它程序产生, 无法手工编辑。

solver.anydos.in 文件可以由 *Hibiscus* 组件的 toolbox/makedos 程序产生, 也可以由其它程序产生, 无法手工编辑²。

solver.ktau.in 文件可以由 *Hibiscus* 组件的 toolbox/makescr 程序产生, 无法手工编辑³。

atom.cix 文件可以由 *Jasmine* 组件产生, 也可以由杜亮博士提供的原子问题程序 rambutan 产生, 不能手工编辑⁴。对于 *Begonia* 与 *Lavender* 组件而言, atom.cix 文件是必须的。除了这个特例以外, 其它任意输入文件对于任意的量子杂质模型求解器而言都是可选的输入, 可以不提供。

3.3 *i*QIST 组件的运行

以串行模式运行量子杂质模型求解器的命令为:

```
$ nohup ctqmc < /dev/null > out.dat &
```

以并行模式运行量子杂质模型求解器的命令为:

```
$ nohup mpiexec -n np ctqmc < /dev/null > out.dat &
```

其中 nohup 程序让量子杂质模型求解器不会被挂起, 忽略系统发送的 hangup 信号; mpiexec 表示 MPI 启动程序, np 表示计算核心的数目, 建议最好保持 2 的倍数; </dev/null 表示从空设备中读取输入文件; > out.dat 表示将输出重定向至 out.dat 文件中; & 符号让量子杂质模型求解器在后台执行。

如果你所用的服务器需要使用队列系统提交计算任务, 那么上述命令须做出相应的改动。

¹关于 pysci.py 程序的细节, 请参阅第7.6节。

²关于 makedos 程序的细节, 请参阅第7.5节。

³关于 makescr 程序的细节, 请参阅第7.5节。

⁴关于 *Jasmine* 组件的细节, 请参阅第7.1节。关于 rambutan 程序的问题, 请与杜亮博士联系 (mailto:duliang@iphy.ac.cn)。

3.4 *iQIST* 组件的监测

保持对 *iQIST* 组件，尤其是量子杂质求解器组件的运行时刻监测是十分重要的。可以通过下述方法监测 *iQIST* 组件的运行情况。

1. 使用 `top` 命令监测 *iQIST* 组件的进程信息，包括 CPU 占用率，内存占用，运行时间等等。在正常情况下，每个 `ctqmc` 进程的 CPU 占用率应该在 100% 左右。

2. 在当前工作目录中执行下述命令：

```
$ tail -f out.dat
```

随时注意 *iQIST* 组件的运行输出有无异常。每隔一段时间，量子杂质求解器组件会自检自身的运行状态，如果发现异常会报错退出。

3. 在编译 *iQIST* 组件时启用 `-pg` 选项，那么 *iQIST* 组件在运行时会自动搜集各个子程序的执行信息，包括调用次数，执行时间等等。接着你可以在任意时刻强制终止 *iQIST* 组件的执行⁵，那么在当前工作目录下会自动产生一个名为 `gmon.out` 的文件。然后利用 `gprof` 程序可以探测执行的瓶颈在何处。此前我们曾经频繁使用此方法来优化 *iQIST* 组件的执行效率。

```
$ gprof -b ctqmc
```

4. 随时观察 `solver.green.dat` 与 `solver.hist.dat` 等输出文件，看当前的计算结果是否合理。

5. 随时观察 `solver.status.dat` 文件，看计算结果是否合理，普通用户不建议使用此方法，因为我知道你看不懂。

⁵通过 `CTRL+C` 或者是 `CTRL+D` 命令。

第四章

iQIST 的标准输入文件

本章将描述量子杂质求解器组件的输入文件的格式，在第5章将描述输出文件的格式。

4.1 solver.ctqmc.in

solver.ctqmc.in 文件是所有的量子杂质求解器组件都需要的可选的配置文件，它可以设置一些关键的计算参数。所有的量子杂质求解器组件都预置了一套完整的计算参数，因此即使用户不提供 solver.ctqmc.in 文件，量子杂质求解器组件也能正常执行。但是如果当前目录下存在 solver.ctqmc.in 文件，那么 solver.ctqmc.in 文件中设置的参数将会覆盖掉缺省的参数。这是我们提请用户需要特别注意的地方。此外，出于兼容性的考虑，对于不同的量子杂质求解器组件，其对应的 solver.ctqmc.in 文件的格式十分类似，仅有十分微小的差异。下面我们将给出不同量子杂质求解器组件的标准 solver.ctqmc.in 文件输入格式。

注意: 用户可以通过 pysci.py 程序产生标准的 solver.ctqmc.in 文件，具体步骤请参阅第7.6节。

4.1.1 Azalea //solver.ctqmc.in

Azalea 组件的标准 solver.ctqmc.in 文件如下：

```
1
2 AZALEA: continuous time quantum Monte Carlo quantum impurity solver
3
4 2          ! non-self-consistent (1) or self-consistent mode (2)
5 2          ! without symmetry (1) or with symmetry mode (2)
6 1          ! spin projection, PM (1) or AFM mode (2)
```

```

7 2          ! without binning      (1) or with binning      mode (2)
8
9 1          ! number of bands
10 2         ! number of spin projection
11 2         ! number of orbitals (= nband * nspin)
12 4         ! number of atomic states
13 20        ! maximum number of DMFT + CTQMC self-consistent iterations
14
15 4.00      ! U : average Coulomb interaction
16 4.00      ! Uc: intraorbital Coulomb interaction
17 4.00      ! Uv: interorbital Coulomb interaction , Uv = Uc-2*Jz for t2g system
18 0.00      ! Jz: Hund's exchange interaction in z axis (Jz = Js = Jp = J)
19 0.00      ! Js: spin-flip term
20 0.00      ! Jp: pair-hopping term
21
22 2.00      ! chemical potential or fermi level
23 8.00      ! inversion of temperature
24 0.50      ! coupling parameter t for Hubbard model
25 0.70      ! mixing parameter for self-consistent engine
26 ~~~~~
27 1024      ! maximum perturbation expansions order
28 8193      ! maximum number of matsubara frequency
29
30 128       ! maximum number of matsubara frequency sampling by quantum impurity solver
31 1024      ! number of time slice
32 20000     ! flip period for spin up and spin down states
33 200000    ! maximum number of thermalization steps
34 200000000 ! maximum number of quantum Monte Carlo sampling steps
35 20000000  ! output period
36 100000    ! clean update period
37 100       ! how often to sampling the gmat and nmat
38 100       ! how often to sampling the gtau and prob
39 ~~~~~

```

具体的输入格式如表4.1所示。

表 4.1: *Azalea* 组件 solver.ctqmc.in 文件的输入格式

变量名称	所在行号	数据类型	作用参考
isscf	4	integer	第6.1节
issun	5	integer	第6.2节
isspn	6	integer	第6.3节
isbin	7	integer	第6.4节
nband	9	integer	第6.8节
nspin	10	integer	第6.9节
norbs	11	integer	第6.10节
ncfgs	12	integer	第6.11节
niter	13	integer	第6.16节
U	15	real(dp)	第6.17节
Uc	16	real(dp)	第6.18节
Uv	17	real(dp)	第6.19节
Jz	18	real(dp)	第6.20节
Js	19	real(dp)	第6.21节
Jp	20	real(dp)	第6.22节
mune	22	real(dp)	第6.25节
beta	23	real(dp)	第6.26节
part	24	real(dp)	第6.27节
alpha	25	real(dp)	第6.28节
mkink	27	integer	第6.33节
mfreq	28	integer	第6.34节

nfreq	30	integer	第6.37节
ntime	31	integer	第6.38节
nflip	32	integer	第6.41节
ntherm	33	integer	第6.42节
nsweep	34	integer	第6.43节
nwrite	35	integer	第6.44节
nclean	36	integer	第6.45节
nmonte	37	integer	第6.46节
ncarlo	38	integer	第6.47节

4.1.2 *Gardenia* //solver.ctqmc.in

Gardenia 组件的标准 solver.ctqmc.in 文件如下：

```

1  =====
2  GARDENIA: continuous time quantum Monte Carlo quantum impurity solver
3  =====
4  2          ! non-self-consistent (1) or self-consistent mode (2)
5  2          ! without symmetry (1) or with symmetry mode (2)
6  1          ! spin projection, PM (1) or AFM mode (2)
7  2          ! without binning (1) or with binning mode (2)
8  5          ! normal measurement (1) or legendre polynomial (2) or chebyshev polynomial (3)
9  1          ! without vertex (1) or with vertex function (2)
10 =====
11 1          ! number of bands
12 2          ! number of spin projection
13 2          ! number of orbitals (= nband * nspin)
14 4          ! number of atomic states
15 20         ! maximum number of DMFT + CTQMC self-consistent iterations
16 =====
17 4.00       ! U : average Coulomb interaction
18 4.00       ! Uc: intraorbital Coulomb interaction
19 4.00       ! Uv: interorbital Coulomb interaction, Uv = Uc-2*Jz for t2g system

```

```

20 0.00      ! Jz: Hund's exchange interaction in z axis (Jz = Js = Jp = J)
21 0.00      ! Js: spin-flip term
22 0.00      ! Jp: pair-hopping term
23
24 2.00      ! chemical potential or fermi level
25 8.00      ! inversion of temperature
26 0.50      ! coupling parameter t for Hubbard model
27 0.70      ! mixing parameter for self-consistent engine
28 ~~~~~
29 24        ! maximum order for legendre polynomial
30 20001     ! number of mesh points for legendre polynomial
31 24        ! maximum order for chebyshev polynomial
32 20001     ! number of mesh points for chebyshev polynomial
33
34 1024      ! maximum perturbation expansions order
35 8193      ! maximum number of matsubara frequency
36
37 32        ! number of matsubara frequency for the two-particle green's function
38 8         ! number of bosonic frequency for the two-particle green's function
39 128       ! maximum number of matsubara frequency sampling by quantum impurity solver
40 1024      ! number of time slice
41 20000     ! flip period for spin up and spin down states
42 200000    ! maximum number of thermalization steps
43 200000000 ! maximum number of quantum Monte Carlo sampling steps
44 20000000  ! output period
45 100000    ! clean update period
46 100       ! how often to sampling the gmat and nmat
47 100       ! how often to sampling the gtau and prob
48 ~~~~~

```

具体的输入格式如表4.2所示。

表 4.2: *Gardenia* 组件 solver.ctqmc.in 文件的输入格式

变量名称	所在行号	数据类型	作用参考
isscf	4	integer	第6.1节

issun	5	integer	第6.2节
isspn	6	integer	第6.3节
isbin	7	integer	第6.4节
isort	8	integer	第6.5节
isvrt	9	integer	第6.6节
nband	11	integer	第6.8节
nspin	12	integer	第6.9节
norbs	13	integer	第6.10节
ncfgs	14	integer	第6.11节
niter	15	integer	第6.16节
U	17	real(dp)	第6.17节
Uc	18	real(dp)	第6.18节
Uv	19	real(dp)	第6.19节
Jz	20	real(dp)	第6.20节
Js	21	real(dp)	第6.21节
Jp	22	real(dp)	第6.22节
mune	24	real(dp)	第6.25节
beta	25	real(dp)	第6.26节
part	26	real(dp)	第6.27节
alpha	27	real(dp)	第6.28节
lemax	29	integer	第6.29节
legrd	30	integer	第6.30节
chmax	31	integer	第6.31节

chgrd	32	integer	第6.32节
mkink	34	integer	第6.33节
mfreq	35	integer	第6.34节
nffrq	37	integer	第6.35节
nbfrq	38	integer	第6.36节
nfreq	39	integer	第6.37节
ntime	40	integer	第6.38节
nflip	41	integer	第6.41节
ntherm	42	integer	第6.42节
nsweep	43	integer	第6.43节
nwrite	44	integer	第6.44节
nclean	45	integer	第6.45节
nmonte	46	integer	第6.46节
ncarlo	47	integer	第6.47节

4.1.3 \mathcal{N} arcissus //solver.ctqmc.in

\mathcal{N} arcissus 组件的标准 solver.ctqmc.in 文件如下:

```

1  =====
2  NARCISSUS: continuous time quantum Monte Carlo quantum impurity solver
3  =====
4  2          ! non-self-consistent (1) or self-consistent mode (2)
5  2          ! without symmetry (1) or with symmetry mode (2)
6  1          ! spin projection, PM (1) or AFM mode (2)
7  2          ! without binning (1) or with binning mode (2)
8  2          ! normal measurement (1) or legendre polynomial (2) or chebyshev polynomial (3)
9  1          ! without vertex (1) or with vertex function (2)
10 1          ! normal (1) or holstein-hubbard (2) or plasmon pole (3) or ohmic model (4)
11 =====

```

```

12 1          ! number of bands
13 2          ! number of spin projection
14 2          ! number of orbitals (= nband * nspin)
15 4          ! number of atomic states
16 20         ! maximum number of DMFT + CTQMC self-consistent iterations
17
18 4.00       ! U : average Coulomb interaction
19 4.00       ! Uc: intraorbital Coulomb interaction
20 4.00       ! Uv: interorbital Coulomb interaction , Uv = Uc-2*Jz for t2g system
21 0.00       ! Jz: Hund's exchange interaction in z axis (Jz = Js = Jp = J)
22 0.00       ! Js: spin-flip term
23 0.00       ! Jp: pair-hopping term
24 0.00       ! lc: strength of screening effect
25 0.00       ! wc: screening frequency
26
27 2.00       ! chemical potential or fermi level
28 8.00       ! inversion of temperature
29 0.50       ! coupling parameter t for Hubbard model
30 0.70       ! mixing parameter for self-consistent engine
31 ~~~~~
32 24         ! maximum order for legendre polynomial
33 20001      ! number of mesh points for legendre polynomial
34 24         ! maximum order for chebyshev polynomial
35 20001      ! number of mesh points for chebyshev polynomial
36
37 1024       ! maximum perturbation expansions order
38 8193       ! maximum number of matsubara frequency
39
40 32         ! number of matsubara frequency for the two-particle green's function
41 8          ! number of bosonic frequency for the two-particle green's function
42 128        ! maximum number of matsubara frequency sampling by quantum impurity solver
43 1024       ! number of time slice
44 20000      ! flip period for spin up and spin down states
45 200000     ! maximum number of thermalization steps
46 200000000  ! maximum number of quantum Monte Carlo sampling steps
47 20000000   ! output period
48 100000     ! clean update period
49 100        ! how often to sampling the gmat and nmat

```



```

50 100          ! how often to sampling the gtau and prob
51 ~~~~~

```

具体的输入格式如表4.3所示。

表 4.3: *Narcissus* 组件 solver.ctqmc.in 文件的输入格式

变量名称	所在行号	数据类型	作用参考
isscf	4	integer	第6.1节
issun	5	integer	第6.2节
isspn	6	integer	第6.3节
isbin	7	integer	第6.4节
isort	8	integer	第6.5节
isvrt	9	integer	第6.6节
isscr	10	integer	第6.7节
nband	12	integer	第6.8节
nspin	13	integer	第6.9节
norbs	14	integer	第6.10节
ncfgs	15	integer	第6.11节
niter	16	integer	第6.16节
U	18	real(dp)	第6.17节
Uc	19	real(dp)	第6.18节
Uv	20	real(dp)	第6.19节
Jz	21	real(dp)	第6.20节
Js	22	real(dp)	第6.21节
Jp	23	real(dp)	第6.22节

lc	24	real(dp)	第6.23节
wc	25	real(dp)	第6.24节
mune	27	real(dp)	第6.25节
beta	28	real(dp)	第6.26节
part	29	real(dp)	第6.27节
alpha	30	real(dp)	第6.28节
lemax	32	integer	第6.29节
legrd	33	integer	第6.30节
chmax	34	integer	第6.31节
chgrd	35	integer	第6.32节
mkink	37	integer	第6.33节
mfreq	38	integer	第6.34节
nffrq	40	integer	第6.35节
nbfrq	41	integer	第6.36节
nfreq	42	integer	第6.37节
ntime	43	integer	第6.38节
nflip	44	integer	第6.41节
ntherm	45	integer	第6.42节
nsweep	46	integer	第6.43节
nwrite	47	integer	第6.44节
nclean	48	integer	第6.45节
nmonte	49	integer	第6.46节
ncarlo	50	integer	第6.47节

4.1.4 Begonia //solver.ctqmc.in

Begonia 组件的标准 solver.ctqmc.in 文件如下:

```

1
2  setup continuous time quantum Monte Carlo quantum impurity solver
3
4  2          ! non-self-consistent (1) or self-consistent mode (2)
5  2          ! without symmetry (1) or with symmetry mode (2)
6  1          ! spin projection, PM (1) or AFM mode (2)
7  2          ! without binning (1) or with binning mode (2)
8
9  1          ! number of bands
10 2          ! number of spin projection
11 2          ! number of orbitals (= nband * nspin)
12 4          ! number of atomic states
13 1024       ! maximum number of non-zero elements in sparse matrix style
14 20         ! maximum number of DMFT + CTQMC self-consistent iterations
15
16 4.00       ! U : average Coulomb interaction
17 4.00       ! Uc: intraorbital Coulomb interaction
18 3.00       ! Uv: interorbital Coulomb interaction, Uv = Uc-2*Jz for t2g system
19 0.50       ! Jz: Hund's exchange interaction in z axis (Jz = Js = Jp = J)
20 0.50       ! Js: spin-flip term
21 0.50       ! Jp: pair-hopping term
22
23 5.00       ! chemical potential or fermi level
24 10.0       ! inversion of temperature
25 0.50       ! coupling parameter t for Hubbard model
26 0.70       ! mixing parameter for self-consistent engine
27 ~~~~~
28 1024       ! maximum perturbation expansions order
29 8193       ! maximum number of matsubara frequency
30
31 128        ! maximum number of matsubara frequency sampling by quantum impurity solver
32 1024       ! number of time slice
33 4          ! number of parts that the imaginary time axis is split
34 20000      ! flip period for spin up and spin down states
35 200000     ! maximum number of thermalization steps

```

```

36 20000000      ! maximum number of quantum Monte Carlo sampling steps
37 2000000      ! output period
38 10000        ! clean update period
39 100          ! how often to sampling the gmat and nmat
40 100          ! how often to sampling the gtau and prob
41 ~~~~~

```

具体的输入格式如表4.4所示。

表 4.4: *Begonia* 组件 solver.ctqmc.in 文件的输入格式

变量名称	所在行号	数据类型	作用参考
isscf	4	integer	第6.1节
issun	5	integer	第6.2节
isspn	6	integer	第6.3节
isbin	7	integer	第6.4节
nband	9	integer	第6.8节
nspin	10	integer	第6.9节
norbs	11	integer	第6.10节
ncfgs	12	integer	第6.11节
nzero	13	integer	第6.12节
niter	14	integer	第6.16节
U	16	real(dp)	第6.17节
Uc	17	real(dp)	第6.18节
Uv	18	real(dp)	第6.19节
Jz	19	real(dp)	第6.20节
Js	20	real(dp)	第6.21节

Jp	21	real(dp)	第6.22节
mune	23	real(dp)	第6.25节
beta	24	real(dp)	第6.26节
part	25	real(dp)	第6.27节
alpha	26	real(dp)	第6.28节
mkink	28	integer	第6.33节
mfreq	29	integer	第6.34节
nfreq	31	integer	第6.37节
ntime	32	integer	第6.38节
npart	33	integer	第6.40节
nflip	34	integer	第6.41节
ntherm	35	integer	第6.42节
nsweep	36	integer	第6.43节
nwrite	37	integer	第6.44节
nclean	38	integer	第6.45节
nmonte	39	integer	第6.46节
ncarlo	40	integer	第6.47节

4.1.5 **Lavender** //solver.ctqmc.in

Lavender 组件的标准 solver.ctqmc.in 文件如下:

```

1  =====
2  setup continuous time quantum Monte Carlo quantum impurity solver
3  =====
4  2          ! non-self-consistent (1) or self-consistent mode (2)
5  2          ! without symmetry (1) or with symmetry mode (2)
6  1          ! spin projection, PM (1) or AFM mode (2)

```

```

7 2          ! without binning      (1) or with binning      mode (2)
8 1          ! normal measurement  (1) or legendre polynomial (2) or chebyshev polynomial (3)
9 1          ! without vertex      (1) or with vertex function (2)
10
11 1          ! number of bands
12 2          ! number of spin projection
13 2          ! number of orbitals (= nband * nspin)
14 4          ! number of atomic states
15 1024       ! maximum number of non-zero elements in sparse matrix style
16 20         ! maximum number of DMFT + CTQMC self-consistent iterations
17
18 2.00       ! U : average Coulomb interaction
19 2.00       ! Uc: intraorbital Coulomb interaction
20 1.50       ! Uv: interorbital Coulomb interaction , Uv = Uc-2*Jz for t2g system
21 0.25       ! Jz: Hund's exchange interaction in z axis (Jz = Js = Jp = J)
22 0.25       ! Js: spin-flip term
23 0.25       ! Jp: pair-hopping term
24
25 8.0        ! chemical potential or fermi level
26 80.0       ! inversion of temperature
27 0.50       ! coupling parameter t for Hubbard model
28 0.70       ! mixing parameter for self-consistent engine
29 ~~~~~
30 24         ! maximum order for legendre polynomial
31 20001      ! number of mesh points for legendre polynomial
32 24         ! maximum order for chebyshev polynomial
33 20001      ! number of mesh points for chebyshev polynomial
34
35 1024       ! maximum perturbation expansions order
36 8193       ! maximum number of matsubara frequency
37
38 128        ! maximum number of matsubara frequency sampling by quantum impurity solver
39 1024       ! number of time slice
40 4          ! number of parts that the imaginary time axis is split
41 20000      ! flip period for spin up and spin down states
42 200000     ! maximum number of thermalization steps
43 20000000   ! maximum number of quantum Monte Carlo sampling steps
44 20000000   ! output period

```

```

45 100000      ! clean update period
46 100         ! how often to sampling the gmat and nmat
47 100         ! how often to sampling the gtau and prob
48 ~~~~~

```

具体的输入格式如表4.5所示。

表 4.5: *Lavender* 组件 solver.ctqmc.in 文件的输入格式

变量名称	所在行号	数据类型	作用参考
isscf	4	integer	第6.1节
issun	5	integer	第6.2节
isspn	6	integer	第6.3节
isbin	7	integer	第6.4节
isort	8	integer	第6.5节
isvrt	9	integer	第6.6节
nband	11	integer	第6.8节
nspin	12	integer	第6.9节
norbs	13	integer	第6.10节
ncfgs	14	integer	第6.11节
nzero	15	integer	第6.12节
niter	16	integer	第6.16节
U	18	real(dp)	第6.17节
Uc	19	real(dp)	第6.18节
Uv	20	real(dp)	第6.19节
Jz	21	real(dp)	第6.20节
Js	22	real(dp)	第6.21节

Jp	23	real(dp)	第6.22节
mune	25	real(dp)	第6.25节
beta	26	real(dp)	第6.26节
part	27	real(dp)	第6.27节
alpha	28	real(dp)	第6.28节
lemax	30	integer	第6.29节
legrd	31	integer	第6.30节
chmax	32	integer	第6.31节
chgrd	33	integer	第6.32节
mkink	35	integer	第6.33节
mfreq	36	integer	第6.34节
nfreq	38	integer	第6.37节
ntime	39	integer	第6.38节
npart	40	integer	第6.40节
nflip	41	integer	第6.41节
ntherm	42	integer	第6.42节
nsweep	43	integer	第6.43节
nwrite	44	integer	第6.44节
nclean	45	integer	第6.45节
nmonte	46	integer	第6.46节
ncarlo	47	integer	第6.47节

4.2 solver.eimp.in

solver.eimp.in 文件是 *iQIST* 量子杂质求解器组件的可选输入文件，它可以设置杂质能级 E_{imp} 和轨道的对称性。如果要求解的量子杂质系统具有以下特性之一，那么该文件是至关重要的。

- 晶体场劈裂不为 0
- 需要考虑自旋 – 轨道耦合效应
- 系统具有磁序

如果用户不提供 solver.eimp.in 文件，量子杂质求解器组件也能正常执行，但是此时所有杂质轨道的能级都将被置为 0.0 eV，并且量子杂质求解器会根据默认设置进行对称化处理。如果在计算目录下存在 solver.eimp.in 文件，那么量子杂质求解器就会根据该文件设定杂质能级 E_{imp} ，并且根据对称指标 *symm* 对杂质轨道进行对称化处理。

对于不同的量子杂质求解器组件，该文件格式完全一样。下面给出该文件的格式和说明。以三带杂质模型为例，典型的 solver.eimp.in 文件如下所示：

```

1  1  +0.500    1
2  2  -0.500    2
3  3  -0.500    2
4  4  +0.500    3
5  5  -0.500    4
6  6  -0.500    4

```

文件详细说明如下：

- 第 1 列，轨道指标 i ，自 1 变化到 *norbs*，数据类型：integer
- 第 2 列，杂质能级 E_{imp} ，数据类型：real(dp)
- 第 3 列，对称指标 *symm*，数据类型：integer

轨道排列是按照先 spin up，后 spin down 的规则进行的，因此在该文件中前三行描述自旋向上的轨道，后三行描述自旋朝下的轨道¹。在上述的 solver.eimp.in 文件中，能带 1(包含轨道 1 和轨道 4) 的杂质能级为 +0.5 eV，而能带 2(包含轨道 2 和轨道 5) 和能带 3(包含轨道 3 和轨道 6) 的杂质能级为 -0.5 eV，即能带 1 在上，能带 2 和能带 3 在下。对称指标 *symm* 用于标定哪些轨道之间是简并的，它的取值范围是 [1,norbs]。如果强制对称化功能被激活(参见第6.3与6.2节)，那么量子杂质求解器组件将会认为对称指标相同的那些轨道之间是简并的，并对它们采取强制对称化。在上面给出的 solver.eimp.in 文件中，轨

¹亦即轨道排列顺序为 $1\uparrow 2\uparrow 3\uparrow 1\downarrow 2\downarrow 3\downarrow$ 。

道 2 和轨道 3 之间是简并的，轨道 5 和轨道 6 之间是简并的，如果 $\text{issun} = 2$ ，那么量子杂质求解器组件将会对它们进行强制对称化。

注意: 用户可以用任意字处理软件编辑 `solver.eimp.in` 文件，也可以通过 `pysci.py` 程序产生标准的 `solver.eimp.in` 文件，具体步骤请参阅第 7.6 节。

4.3 solver.hyb.in

`solver.hyb.in` 文件是 *iQIST* 组件的可选输入文件，它可以设置初始杂化函数 $\Delta(i\omega_n)$ 。该文件一般不手动生成，而是将量子杂质求解器的输出文件 `solver.hyb.dat` 直接重命名为 `solver.hyb.in`，或者是将 LDA+DMFT 接口程序生成的杂化函数文件直接重命名为 `solver.hyb.in` 即可。如果用户不提供 `solver.hyb.in`，那么量子杂质求解器组件内部将会使用无相互作用系统的杂化函数作为初始输入，如果用户提供了 `solver.hyb.in`，那么从 `solver.hyb.in` 中读取的数据将覆盖掉缺省值，作为初始输入。

对于所有的量子杂质求解器组件，`solver.hyb.in` 文件的格式都是一样的，其格式如下：

1	1	0.07853982	-0.72761369	-0.53948595	-0.72761369	-0.53948595
2	1	0.23561945	-0.61858636	-0.48305610	-0.61858636	-0.48305610
3	1	0.39269908	-0.53404592	-0.44801239	-0.53404592	-0.44801239
4	1	0.54977871	-0.46753732	-0.42301926	-0.46753732	-0.42301926
5	1	0.70685835	-0.41300349	-0.40288646	-0.41300349	-0.40288646
6	1	0.86393798	-0.36815701	-0.38507214	-0.36815701	-0.38507214
7	1	1.02101761	-0.33059377	-0.36941834	-0.33059377	-0.36941834
8	1	1.17809725	-0.29807311	-0.35534588	-0.29807311	-0.35534588
9	1	1.33517688	-0.27024493	-0.34189638	-0.27024493	-0.34189638
10					

其文件共有 `nband` 个 block，每个 block 之间间隔两个空行，最后一个 block 后面有两个空行，每个 block 的格式如下：

- 第 1 列，能带指标 i ，自 1 变化至 `nband`，数据类型：integer
- 第 2 列，松原频率点 $\omega_n = (2n + 1)\pi/\beta$ ， n 自 0 变化至 `mfreq-1`，数据类型：real(dp)
- 第 3 列，实部，自旋朝上， $\Re\Delta_{i\uparrow}(i\omega_n)$ ，数据类型：real(dp)
- 第 4 列，虚部，自旋朝上， $\Im\Delta_{i\uparrow}(i\omega_n)$ ，数据类型：real(dp)
- 第 5 列，实部，自旋朝下， $\Re\Delta_{i\downarrow}(i\omega_n)$ ，数据类型：real(dp)
- 第 6 列，虚部，自旋朝下， $\Im\Delta_{i\downarrow}(i\omega_n)$ ，数据类型：real(dp)

第 1 个 block 存放第 1 条能带的数据，第 2 个 block 存放第 2 条能带的数据，然后依此类推。非对角项的数据均没有考虑在内。

4.4 solver.anydos.in

solver.anydos.in 文件存储的是晶格模型的电子态密度 $N(\epsilon)$ ，它是量子杂质求解器组件的可选输入文件，仅在特殊情况下才会被用到。当晶格模型定义在 bethe 晶格上时，*iQIST* 的量子杂质求解器组件内置了自洽方程，可以简捷了当地实现自洽迭代循环。但是当晶格模型不是定义在 bethe 晶格上时，此时又该当如何呢？解决方案是将晶格模型的电子态密度存储在 solver.anydos.in 中，由量子杂质模型求解器组件将其读入，再通过 Hilbert 变换实现自洽方程。

对于所有的量子杂质求解器组件，solver.anydos.in 文件的格式都是一样的，示例文件如下：

1	-4.00000000	0.00020898
2	-3.99000000	0.00021008
3	-3.98000000	0.00021119
4	-3.97000000	0.00021231
5	-3.96000000	0.00021344
6	-3.95000000	0.00021458
7	
8	3.95000000	0.00021458
9	3.96000000	0.00021344
10	3.97000000	0.00021231
11	3.98000000	0.00021119
12	3.99000000	0.00021008
13	4.00000000	0.00020898

solver.anydos.in 文件的格式如下：

- 第 1 列，实频率点 ϵ ，共 801 个点，数据类型:real(dp)
- 第 2 列， $N(\epsilon)$ ，数据类型:real(dp)

在量子杂质求解器组件中，假定不同轨道的 $N(\epsilon)$ 都是简并的，因此只需提供一套 $N(\epsilon)$ 数据即可。假定实频率点围绕 0 点对称分布，左侧 (< 0) 有 400 个点，右侧 (> 0) 有 400 个点，包含 0 点后一共 801 个点。默认原始的 $N(\epsilon)$ 数据未经归一化处理²。

²如果用户提供的 $N(\epsilon)$ 数据不止 801 个点，或者是不同轨道的 $N(\epsilon)$ 之间并不简并，那么需要修改量子杂质求解器组件的源程序，详见 ctqmc_dmft.f90 文件中的 ctqmc_dmft_anydos() 子程序。

注意: 一般不手动编辑 solver.anydos.in 文件, 用户可以使用 *Hibiscus* 组件的 makedos 程序针对不同的晶格模型产生电子态密度文件, 将其输出文件重命名为 solver.anydos.in, 并拷贝至当前计算目录下即可, 具体步骤请参阅第7.5节。

4.5 solver.ktau.in

对于 *Narcissus* 组件而言, 如果需要考虑动态屏蔽效应时 ($\text{isscr} = 99$, 参见第6.7节), 双积分推迟作用函数 $\mathcal{K}(\tau)$ 是一个很重要的输入数据。solver.ktau.in 文件的作用就是存储 $\mathcal{K}(\tau)$ 数据, 它对于 *Narcissus* 组件来说也是一个可选的输入文件, 仅当 $\text{isscr} = 99$ 时, *Narcissus* 组件才需要读取它, 如果 isscr 取其它的值, 那么 *Narcissus* 组件会完全忽略它。除了 *Narcissus* 组件以外, 其余的量子杂质求解器组件完全不需要读取 solver.ktau.in 文件。

典型的 solver.ktau.in 文件如下所示:

1	# u shift:	2.0	mu shift:	1.0
2	0.00000000	0.00000000	0.00000000	
3	0.00977517	0.02397222	0.02397222	
4	0.01955034	0.04705862	0.04705862	
5	0.02932551	0.06932213	0.06932213	
6	0.03910068	0.09081919	0.09081919	
7			
8	9.96089932	0.09081919	0.09081919	
9	9.97067449	0.06932213	0.06932213	
10	9.98044966	0.04705862	0.04705862	
11	9.99022483	0.02397222	0.02397222	
12	10.00000000	0.00000000	0.00000000	

solver.ktau.in 文件的格式如下:

- 第 1 行, 指定关于 U 以及 μ 的 shift
- 第 2 行至第 $\text{ntime} + 1$ 行, 包含 $\mathcal{K}(\tau)$ 数据
- 第 1 列, 虚时间点 τ_i , 从 0 变化至 β , 数据类型:real(dp)
- 第 2 列, $\mathcal{K}(\tau_i)$ 数据, 数据类型:real(dp)
- 第 3 列, 参考数据, 用途暂时不确定, 数据类型:real(dp)

注意: 一般不手动编辑 solver.ktau.in 文件, 用户可以使用 *Hibiscus* 组件的 makescr 程序计算 $\mathcal{K}(\tau)$, 将输出文件重命名为 solver.ktau.in, 并拷贝至当前计算目录下即可, 具体步骤请参阅第7.5节。

4.6 atom.cix

atom.cix 文件是广义相互作用版本的量子杂质模型求解器³必需的输入文件。该文件包含了对角化原子问题后所获得的本征值/本征态, \mathcal{F} 矩阵等信息, 由原子问题程序自动生成。下面给出该文件的例子和详细说明。

如下是一个 2 带系统的 atom.cix 文件, 该文件可以分为两部分。

```

1 # atom.cix, style v1, begonia and lavender compatible
2 # it is generated by jasmine code
3 # eigenvalues: index | energy | occupy | spin
4 1      0.00000000    0.00000000    0.00000000
5 2      0.00000000    1.00000000    0.50000000
6 3      0.00000000    1.00000000    0.50000000
7 4      0.00000000    1.00000000   -0.50000000
8 5      0.00000000    1.00000000   -0.50000000
9 6      1.00000000    2.00000000    1.00000000
10 7      1.00000000    2.00000000    0.00000000
11 8      1.00000000    2.00000000   -1.00000000
12 9      3.00000000    2.00000000    0.00000000
13 10     3.00000000    2.00000000    0.00000000
14 11     5.00000000    2.00000000    0.00000000
15 12     7.00000000    3.00000000    0.50000000
16 13     7.00000000    3.00000000    0.50000000
17 14     7.00000000    3.00000000   -0.50000000
18 15     7.00000000    3.00000000   -0.50000000
19 16    14.00000000    4.00000000    0.00000000
20 # f matrix element: alpha | beta | orbital | fmat
21 1      1      1      0.00000000
22 2      1      1      0.00000000
23 3      1      1      0.00000000

```

³在 *iQIST* 中, *Begonia*、*Lavender*、*Camellia*、*Epiphyllum*、*Pansy*、*Manjushaka* 等组件属于广义相互作用版本的量子杂质模型求解器。这些量子杂质模型求解器又可以细分为三类, 分别是:

- *Begonia*、*Lavender*
- *Camellia*、*Epiphyllum*
- *Pansy*、*Manjushaka*

它们的实现机制有较大的差异, 因此所需要的 atom.cix 文件的格式也各不相同。由于在最新版本的 *iQIST* 软件包中, 我们只公开发布 *Begonia* 与 *Lavender* 组件, 因此此处仅就它们所需要的 atom.cix 文件的格式稍作说明。至于 atom.cix 文件的其它格式, 待相关量子杂质求解器组件发布后, 本文档再一并更新。

```

24  4      1      1      0.00000000
25  5      1      1      0.00000000
26  6      1      1      0.00000000
27  7      1      1      0.00000000
28  8      1      1      0.00000000
29  9      1      1      0.00000000
30 10      1      1      0.00000000
31 11      1      1      0.00000000
32 12      1      1      0.00000000
33 13      1      1      0.00000000
34 14      1      1      0.00000000
35 15      1      1      0.00000000
36 16      1      1      0.00000000
37  . . . . .

```

atom.cix 文件的格式如下:

- 第 1~2 行, 注释部分, 注明此 atom.cix 文件的格式代码以及适用组件。
- 第 1 部分, 原子问题本征值、占据数、自旋。以第 3 行的描述文字开头。
 - 第 1 列, 原子本征态指标 i , 自 1 变化到 ncfgs(此处为 16), 数据类型: integer
 - 第 2 列, 原子问题本征值 E_i , 数据类型: real(dp)
 - 第 3 列, 原子本征态的占据数 N_i , 数据类型: real(dp)
 - 第 4 列, 原子本征态的自旋值 S_i , 数据类型: real(dp)
- 第 2 部分, \mathcal{F} 矩阵 $\langle \alpha | d | \beta \rangle$, 以第 20 行的描述文字开头。
 - 第 1 列, 原子本征态指标 α , 自 1 变化到 ncfgs(此处为 16), 数据类型: integer
 - 第 2 列, 原子本征态指标 β , 自 1 变化到 ncfgs(此处为 16), 数据类型: integer
 - 第 3 列, 轨道指标 i , 自 1 变化到 norbs(此处为 4), 数据类型: integer
 - 第 4 列, \mathcal{F} 矩阵元 $\langle \alpha | d_i | \beta \rangle$, d_i 为第 i 条轨道上的消灭算符, 数据类型: real(dp)

有两种方法可以生成 atom.cix 文件。方法一: 应用 *iQIST* 软件包中内含的 *Jasmine* 组件, 这是我们的首选程序, 与 *iQIST* 软件包的其余组件配合得最好。关于 *Jasmine* 组件的具体用法, 请参阅第 7.1 节。方法二: 应用杜亮博士开发的修改版的 rambutan 程序。rambutan 程序不包含在 *iQIST* 中, 想要获得它请与杜亮博士联系 (<mailto:duliang@iphy.ac.cn>)。在生成 atom.cix 文件之后, 如非十分必要, 请勿对其进行修改, 任何微小的改动都有可能導致计算失败。

第五章

*i*QIST 的标准输出文件

在第4章中，我们已经描述了 *i*QIST 的量子杂质求解器组件所需要的输入文件，那么在本章我们将详细描述 *i*QIST 的量子杂质求解器组件所产生的标准输出文件。

5.1 终端输出

5.1.1 out.dat

*i*QIST 的量子杂质求解器组件在运行过程中会输出大量运行时刻信息到标准终端中，这些信息对于后续的模拟结果分析是十分重要的，因此在调用量子杂质求解器组件时，通常将程序输出重定向至名为 out.dat 的文件中去。这个 out.dat 就是所谓的终端输出文件。

对于不同的量子杂质求解器组件而言其终端输出都是十分类似的。下面以 *Azalea* 组件为例，简要描述 out.dat 文件的结构。典型的由 *Azalea* 组件输出的 out.dat 文件如下所示:

```
1  AZALEA
2  >>> A DMFT Engine With Continuous Time Quantum Monte Carlo Impurity Solver
3
4  version: 2012.08.20T (built at 10:39:32 Nov 06 2012)
5  develop: by li huang, CAEP & IOP
6  support: huangli712@yahoo.com.cn
7  license: GPL2 and later versions
8
9  AZALEA >>> start running at 10:39:39 Nov 6 2012
10 AZALEA >>> parallelism: Yes >>> processors: 1
```

```
11
12 AZALEA >>> parameters list :
13     isscf :          2     isbin :          2
14     issun :          2     isspn :          1
15     mkink :       1024     mfreq :       8193
16     nband :          1     nspin :          2
17     norbs :          2     ncfgs :          4
18     niter :         20     nfreq :         128
19     ntime :       1024     nflip :      20000
20     ntherm:    200000     nsweep: 20000000
21     nclean:   100000     nwrite: 20000000
22     nmonte:        10     ncarlo:         10
23     U       :   4.00000   Uc       :   4.00000
24     Js      :   0.00000   Uv      :   4.00000
25     Jp      :   0.00000   Jz      :   0.00000
26     mune    :   2.00000   beta   :   8.00000
27     part    :   0.50000   temp   :1450.56313
28
29 AZALEA >>> DMFT iter:  1 <<< SELFING
30 AZALEA >>> CTQMC quantum impurity solver running
31     nband :          1     Uc       :   4.00000
32     nspin :          2     Jz       :   0.00000
33
34     quantum impurity solver initializing
35     seed:   747988718
36     time:    0.790s
37
38     quantum impurity solver retrieving
39     time:    0.000s
40
41     quantum impurity solver warmming
42     time:    0.200s
43
44     quantum impurity solver sampling
45
46 AZALEA >>> iter:  1 sweep:  2000000 of  20000000
47     auxiliary system observables:
48     etot :  -0.02928     epot :   0.13050
```



```

49     ekin :   -0.15978    <Sz> :   -0.07174
50     insert kink statistics:
51     count:    899142    98635    800507
52     ratio:    1.00000    0.10970    0.89030
53     remove kink statistics:
54     count:    900978    98635    802343
55     ratio:    1.00000    0.10948    0.89052
56     lshift kink statistics:
57     count:     99871    10329    89542
58     ratio:    1.00000    0.10342    0.89658
59     rshift kink statistics:
60     count:    100009    10354    89655
61     ratio:    1.00000    0.10353    0.89647
62     global swap statistics:
63     count:         -1         0         0
64     ratio:    1.00000    0.00000    0.00000
65     global flip statistics:
66     count:         100         100         0
67     ratio:    1.00000    1.00000    0.00000
68     >>> quantum impurity solver status: normal
69     >>> used time:  2.27 s in this iteration.
70     >>> used time:  2.27 s in total iteration.
71     .....
72     AZALEA >>> CTQMC quantum impurity solver shutdown
73
74     AZALEA >>> DMFT hybridization function is updated
75
76     AZALEA >>> cur_iter: 20  min_iter: 16  max_iter: 20
77     AZALEA >>> sig_curr:0.9306E-06  eps_curr:0.1000E-07
78     AZALEA >>> self-consistent iteration convergence is F
79
80     AZALEA >>> DMFT iter:999 <<< BINNING
81     AZALEA >>> CTQMC quantum impurity solver running
82     nband :         1    Uc    :    4.00000
83     nspin :         2    Jz    :    0.00000
84
85     quantum impurity solver initializing
86     seed:  710026923

```

```

87     time:      0.780s
88
89     quantum impurity solver retrieving
90     time:      0.000s
91
92     quantum impurity solver warmming
93     time:      0.100s
94
95     quantum impurity solver sampling
96
97     AZALEA >>> iter:999 sweep: 20000000 of 200000000
98     auxiliary system observables:
99     etot :  -0.03153      epot :   0.03284
100    ekin :  -0.06437      <Sz> :  -0.00056
101    insert kink statistics:
102    count:   9001306      466067      8535239
103    ratio:    1.00000      0.05178      0.94822
104    remove kink statistics:
105    count:   8997343      466067      8531276
106    ratio:    1.00000      0.05180      0.94820
107    lshift kink statistics:
108    count:    1001183       20190      980993
109    ratio:    1.00000      0.02017      0.97983
110    rshift kink statistics:
111    count:    1000168       19878      980290
112    ratio:    1.00000      0.01987      0.98013
113    global swap statistics:
114    count:         -1          0          0
115    ratio:    1.00000      0.00000      0.00000
116    global flip statistics:
117    count:         1000         1000          0
118    ratio:    1.00000      1.00000      0.00000
119    >>> quantum impurity solver status: binned
120    >>> quantum impurity solver status: normal
121    >>> used time: 11.61 s in this iteration.
122    >>> used time: 11.61 s in total iteration.
123    .....
124    AZALEA >>> CTQMC quantum impurity solver shutdown

```

```

125
126 AZALEA >>> total time spent:      414.10s
127
128 AZALEA >>> I am tired and want to go to bed. Bye!
129 AZALEA >>> happy ending at 11:44:57 Nov  6 2012

```

此 out.dat 文件的格式如下:

- 第 1 行, 组件名称 *Azalea*。
- 第 4 行, 版本号以及此可执行程序的生成时间。版本号为日期与大写字母的组合, D 表示开发版, T 表示测试版, S 表示稳定版。程序稳定程度为 $S > T > D$ 。括号内的时间为此可执行程序的编译生成时间。
- 第 9 行, 程序的启动时间。
- 第 10 行, 程序是否支持并行, 当前一共使用多少个计算核心。
- 第 12 ~ 27 行, 输出当前设置的全部计算参数。
- 第 29 行, 输出 DMFT 迭代计算的信息, 当前是第几步。SELFING 表示处于自治计算模式, BINNING 表示处于 data binning 模式。
- 第 31 ~ 32 行, 再次输出关键的相互作用常数以及能带数目信息。
- 第 35 行, 随机数种子。这个数字很关键, 当程序出错时, 利用此随机数种子可以重复全部的计算结果, 便于我们找出问题的所在。
- 第 46 行, 输出量子杂质求解器组件的计算进程, 当前的 DMFT 循环进行到第几步, 总共 nsweep 次 Monte Carlo sampling, 目前已经进行了多少次。
- 第 47 ~ 49 行, 输出关键的观测量, 如 E_{tot} 、 E_{kin} 、 E_{pot} 和 $\langle S_z \rangle$ 。
- 第 50 ~ 52 行, 输出 INSERT KINK 操作的统计信息。第 51 行给出的三个数字依次分别是: 尝试次数, 接受次数, 拒绝次数, 第 52 行给出的三个数字分别是各自的概率。
- 第 53 ~ 55 行, 输出 REMOVE KINK 操作的统计信息。
- 第 56 ~ 58 行, 输出 LSHIFT KINK 操作的统计信息。
- 第 59 ~ 61 行, 输出 RSHIFT KINK 操作的统计信息。
- 第 62 ~ 64 行, 输出 GLOBAL SWAP 操作的统计信息。第 63 行的第 1 个数字 -1 表示此操作没有被启用。

- 第 62 ~ 64 行，输出 GLOBAL FLIP 操作的统计信息。
- 第 68 行，判断量子杂质求解器组件的运行状态是否正常，normal 表示一切正常。
- 第 69 ~ 70 行，当前运行时间小计。
- 第 76 ~ 78 行，输出自洽迭代过程中电子自能函数的收敛情况。在缺省情况下，量子杂质求解器组件内置的 DMFT 引擎最少迭代计算 16 次，最多迭代计算 20 次。电子自能函数的收敛标准为 0.1×10^{-7} 。
- 第 80 行，程序结束迭代循环计算模式，进入到 data binning 模式。
- 第 126 行，程序总耗时。
- 第 128 ~ 129 行，程序终止，输出终止时间。

5.2 文件输出

5.2.1 solver.green.dat

solver.green.dat 文件包含了虚时杂质格林函数 $G(\tau)$ ，所有量子杂质求解器组件在运行时刻均会输出该文件¹。其文件共有 nband 个 block，每个 block 之间间隔两个空行，最后一个 block 后面有两个空行，每个 block 的格式如下：

- 第 1 列，能带指标 i ，自 1 变化至 nband，格式 i5
- 第 2 列，时间片指标 j ，自 1 变化至 ntime，格式 i5
- 第 3 列，虚时间点， τ_j ，格式 f12.6
- 第 4 列，自旋朝上， $G_{i\uparrow}(\tau_j)$ ，格式 f12.6
- 第 5 列，自旋朝下， $G_{i\downarrow}(\tau_j)$ ，格式 f12.6

第 1 个 block 存放第 1 条能带的数据，第 2 个 block 存放第 2 条能带的数据，然后依此类推。不包括非对角项的数据。如果需要使用 gnuplot 绘图，可用如下的命令：

```
gnuplot> plot 'solver.green.dat' u 2:4 w lp
```

或者是

```
gnuplot> plot 'solver.green.dat' u 3:4 w lp
```

¹每隔 nwrite 次 Monte Carlo sampling，量子杂质求解器组件会输出一当前时的 $G(\tau)$ 到 solver.green.dat 文件中。

5.2.2 solver.green.bin

solver.green.bin 文件包含了虚时杂质格林函数 $G(\tau)$ ，仅当 data binning 模式被激活时 (isbin = 2，参见第6.4节) 量子杂质求解器组件才会输出此文件²。其文件共有 nband 个 block，每个 block 之间间隔两个空行，最后一个 block 后面有两个空行，每个 block 的格式如下：

- 第 1 列，能带指标 i ，自 1 变化至 nband，格式 i5
- 第 2 列，时间片指标 j ，自 1 变化至 ntime，格式 i5
- 第 3 列，虚时间点， τ_j ，格式 f12.6
- 第 4 列，自旋朝上， $G_{i\uparrow}(\tau_j)$ ，格式 f12.6
- 第 5 列，自旋朝下， $G_{i\downarrow}(\tau_j)$ ，格式 f12.6

第 1 个 block 存放第 1 条能带的数据，第 2 个 block 存放第 2 条能带的数据，然后依此类推。不包括非对角项的数据。如果需要使用 gnuplot 绘图，可用如下的命令：

```
gnuplot> plot 'solver.green.bin.1' u 2:4 w lp
```

或者是

```
gnuplot> plot 'solver.green.bin.1' u 3:4 w lp
```

5.2.3 solver.weiss.dat

solver.weiss.dat 文件包含了虚时 bath 函数 $\mathcal{G}(\tau)$ ，仅当自洽计算模式被激活时 (isscf = 2，参见第6.1节) 量子杂质求解器组件才会输出此文件³。其文件共有 nband 个 block，每个 block 之间间隔两个空行，最后一个 block 后面有两个空行，每个 block 的格式如下：

- 第 1 列，能带指标 i ，自 1 变化至 nband，格式 i5
- 第 2 列，时间片指标 j ，自 1 变化至 ntime，格式 i5
- 第 3 列，虚时间点， τ_j ，格式 f12.6
- 第 4 列，自旋朝上， $\mathcal{G}_{i\uparrow}(\tau_j)$ ，格式 f12.6
- 第 5 列，自旋朝下， $\mathcal{G}_{i\downarrow}(\tau_j)$ ，格式 f12.6

第 1 个 block 存放第 1 条能带的数据，第 2 个 block 存放第 2 条能带的数据，然后依此类推。不包括非对角项的数据。如果需要使用 gnuplot 绘图，可用如下的命令：

²事实上，正确的文件名应该为 solver.green.bin.*，* 代表当前 data bin 的编号，data bin 的总数等于 nsweep/nwrite。

³solver.weiss.dat 文件在每次 DMFT 迭代计算后才会被更新一次。

```
gnuplot> plot 'solver.weiss.dat' u 2:4 w lp
```

或者是

```
gnuplot> plot 'solver.weiss.dat' u 3:4 w lp
```

5.2.4 solver.hybri.dat

solver.hybri.dat 文件包含了虚时杂化函数 $\Delta(\tau)$ 。所有量子杂质求解器组件在运行时刻均会输出该文件⁴。其文件共有 nband 个 block，每个 block 之间间隔两个空行，最后一个 block 后面有两个空行，每个 block 的格式如下：

- 第 1 列，能带指标 i ，自 1 变化至 nband，格式 i5
- 第 2 列，时间片指标 j ，自 1 变化至 ntime，格式 i5
- 第 3 列，虚时间点， τ_j ，格式 f12.6
- 第 4 列，自旋朝上， $\Delta_{i\uparrow}(\tau_j)$ ，格式 f12.6
- 第 5 列，自旋朝下， $\Delta_{i\downarrow}(\tau_j)$ ，格式 f12.6

第 1 个 block 存放第 1 条能带的数据，第 2 个 block 存放第 2 条能带的数据，然后依此类推。不包括非对角项的数据。如果需要使用 gnuplot 绘图，可用如下的命令：

```
gnuplot> plot 'solver.hybri.dat' u 2:4 w lp
```

或者是

```
gnuplot> plot 'solver.hybri.dat' u 3:4 w lp
```

5.2.5 solver.grn.dat

solver.grn.dat 文件包含了虚频杂质格林函数 $G(i\omega)$ 。所有量子杂质求解器组件在运行时刻均会输出该文件⁵。其文件共有 nband 个 block，每个 block 之间间隔两个空行，最后一个 block 后面有两个空行，每个 block 的格式如下：

- 第 1 列，能带指标 i ，自 1 变化至 nband，格式 i5
- 第 2 列，松原频率点 $\omega_n = (2n + 1)\pi/\beta$ ， n 自 0 变化至 mfreq-1，格式 f16.8
- 第 3 列，实部，自旋朝上， $\Re G_{i\uparrow}(i\omega_n)$ ，格式 f16.8
- 第 4 列，虚部，自旋朝上， $\Im G_{i\uparrow}(i\omega_n)$ ，格式 f16.8

⁴solver.hybri.dat 文件在每次 DMFT 迭代计算后才会被更新一次。

⁵solver.grn.dat 文件在每次 DMFT 迭代计算后才会被更新一次。

- 第 5 列, 实部, 自旋朝下, $\Re G_{i\downarrow}(i\omega_n)$, 格式 f16.8
- 第 6 列, 虚部, 自旋朝下, $\Im G_{i\downarrow}(i\omega_n)$, 格式 f16.8

第 1 个 block 存放第 1 条能带的数据, 第 2 个 block 存放第 2 条能带的数据, 然后依此类推。不包括非对角项的数据。如果需要使用 gnuplot 绘图, 例如画出虚部, 可用如下的命令:

```
gnuplot> plot 'solver.grn.dat' u 2:4 w lp
```

或者画出低频的部分, 例如前 50 个频率点, 可用如下的命令:

```
gnuplot> plot [0:50] 'solver.grn.dat' u 2:4 w lp
```

5.2.6 solver.wss.dat

solver.wss.dat 文件包含了虚频 bath 函数 $\mathcal{G}(i\omega)$ 。仅当自洽计算模式被激活时 (isscf = 2, 参见第 6.1 节) 量子杂质求解器组件才会输出此文件⁶。其文件共有 nband 个 block, 每个 block 之间间隔两个空行, 最后一个 block 后面有两个空行, 每个 block 的格式如下:

- 第 1 列, 能带指标 i , 自 1 变化至 nband, 格式 i5
- 第 2 列, 松原频率点 $\omega_n = (2n + 1)\pi/\beta$, n 自 0 变化至 mfreq-1, 格式 f16.8
- 第 3 列, 实部, 自旋朝上, $\Re \mathcal{G}_{i\uparrow}(i\omega_n)$, 格式 f16.8
- 第 4 列, 虚部, 自旋朝上, $\Im \mathcal{G}_{i\uparrow}(i\omega_n)$, 格式 f16.8
- 第 5 列, 实部, 自旋朝下, $\Re \mathcal{G}_{i\downarrow}(i\omega_n)$, 格式 f16.8
- 第 6 列, 虚部, 自旋朝下, $\Im \mathcal{G}_{i\downarrow}(i\omega_n)$, 格式 f16.8

第 1 个 block 存放第 1 条能带的数据, 第 2 个 block 存放第 2 条能带的数据, 然后依此类推。不包括非对角项的数据。如果需要使用 gnuplot 绘图, 例如画出虚部, 可用如下的命令:

```
gnuplot> plot 'solver.wss.dat' u 2:4 w lp
```

或者画出低频的部分, 例如前 50 个频率点, 可用如下的命令:

```
gnuplot> plot [0:50] 'solver.wss.dat' u 2:4 w lp
```

5.2.7 solver.hyb.dat

solver.hyb.dat 文件包含了虚频杂化函数 $\Delta(i\omega)$ 。所有量子杂质求解器组件在运行时刻均会输出该文件⁷。其文件共有 nband 个 block, 每个 block 之间间隔两个空行, 最后一个 block 后面有两个空行, 每个 block 的格式如下:

⁶solver.wss.dat 文件在每次 DMFT 迭代计算后才会被更新一次。

⁷solver.hyb.dat 文件在每次 DMFT 迭代计算后才会被更新一次。

- 第 1 列, 能带指标 i , 自 1 变化至 n_{band} , 格式 i5
- 第 2 列, 松原频率点 $\omega_n = (2n + 1)\pi/\beta$, n 自 0 变化至 $\text{mfreq}-1$, 格式 f16.8
- 第 3 列, 实部, 自旋朝上, $\Re\Delta_{i\uparrow}(i\omega_n)$, 格式 f16.8
- 第 4 列, 虚部, 自旋朝上, $\Im\Delta_{i\uparrow}(i\omega_n)$, 格式 f16.8
- 第 5 列, 实部, 自旋朝下, $\Re\Delta_{i\downarrow}(i\omega_n)$, 格式 f16.8
- 第 6 列, 虚部, 自旋朝下, $\Im\Delta_{i\downarrow}(i\omega_n)$, 格式 f16.8

第 1 个 block 存放第 1 条能带的数据, 第 2 个 block 存放第 2 条能带的数据, 然后依此类推。不包括非对角项的数据。如果需要使用 gnuplot 绘图, 例如画出虚部, 可用如下的命令:

```
gnuplot> plot 'solver.hyb.dat' u 2:4 w lp
```

或者画出低频的部分, 例如前 50 个频率点, 可用如下的命令:

```
gnuplot> plot [0:50] 'solver.hyb.dat' u 2:4 w lp
```

5.2.8 solver.sgm.dat

solver.sgm.dat 文件包含了虚频自能函数 $\Sigma(i\omega)$ 。所有量子杂质求解器组件在运行时刻均会输出该文件⁸。其文件共有 n_{band} 个 block, 每个 block 之间间隔两个空行, 最后一个 block 后面有两个空行, 每个 block 的格式如下:

- 第 1 列, 能带指标 i , 自 1 变化至 n_{band} , 格式 i5
- 第 2 列, 松原频率点 $\omega_n = (2n + 1)\pi/\beta$, n 自 0 变化至 $\text{mfreq}-1$, 格式 f16.8
- 第 3 列, 实部, 自旋朝上, $\Re\Sigma_{i\uparrow}(i\omega_n)$, 格式 f16.8
- 第 4 列, 虚部, 自旋朝上, $\Im\Sigma_{i\uparrow}(i\omega_n)$, 格式 f16.8
- 第 5 列, 实部, 自旋朝下, $\Re\Sigma_{i\downarrow}(i\omega_n)$, 格式 f16.8
- 第 6 列, 虚部, 自旋朝下, $\Im\Sigma_{i\downarrow}(i\omega_n)$, 格式 f16.8

第 1 个 block 存放第 1 条能带的数据, 第 2 个 block 存放第 2 条能带的数据, 然后依此类推。不包括非对角项的数据。如果需要使用 gnuplot 绘图, 例如画出虚部, 可用如下的命令:

```
gnuplot> plot 'solver.sgm.dat' u 2:4 w lp
```

或者画出低频的部分, 例如前 50 个频率点, 可用如下的命令:

```
gnuplot> plot [0:50] 'solver.sgm.dat' u 2:4 w lp
```

⁸solver.sgm.dat 文件在每次 DMFT 迭代计算后才会被更新一次。

5.2.9 solver.hub.dat

solver.hub.dat 文件包含了原子极限下的虚频杂质格林函数 $G_{\text{atom}}(i\omega)$ 以及自能函数 $\Sigma_{\text{atom}}(i\omega)$ 。所有量子杂质求解器组件在运行时刻均会输出该文件⁹。其文件共有 nband 个 block，每个 block 之间间隔两个空行，最后一个 block 后面有两个空行，每个 block 的格式如下：

- 第 1 列，能带指标 i ，自 1 变化至 nband，格式 i5
- 第 2 列，虚频点 $\omega_n = (2n + 1)\pi/\beta$ ， n 自 0 变化至 mfreq-1，格式 f16.8
- 第 3 列，实部，自旋朝上， $\Re G_{\text{atom}}(i\omega_n)$ ，格式 f16.8
- 第 4 列，虚部，自旋朝上， $\Im G_{\text{atom}}(i\omega_n)$ ，格式 f16.8
- 第 5 列，实部，自旋朝上， $\Re \Sigma_{\text{atom}}(i\omega_n)$ ，格式 f16.8
- 第 6 列，虚部，自旋朝上， $\Im \Sigma_{\text{atom}}(i\omega_n)$ ，格式 f16.8

第 1 个 block 存放第 1 条能带的数据，第 2 个 block 存放第 2 条能带的数据，然后依此类推。不包括非对角项的数据。如果需要使用 gnuplot 绘图，例如画出 G_{atom} 虚部，可用如下的命令：

```
gnuplot> plot 'solver.hub.dat' u 2:4 w lp
```

或者画出低频的部分，例如前 50 个频率点，可用如下的命令：

```
gnuplot> plot [0:50] 'solver.hub.dat' u 2:4 w lp
```

5.2.10 solver.nmat.dat

solver.nmat.dat 文件包含了占据数 $\langle n_i \rangle$ 与双占据数 $\langle n_i n_j \rangle$ 。所有量子杂质求解器组件在运行时刻均会输出该文件¹⁰。其文件格式如下：

- 第 1 部分，占据数 $\langle n_i \rangle$ ，该部分以描述文字“< n_i > data: “开头
 - 接下来 norbs 行，轨道占据数 $\langle n_i \rangle$ 。
 - * 第 1 列，轨道指标 i ，自 1 变化至 norbs，格式 i5；
 - * 第 2 列，占据数 $\langle n_i \rangle$ ，格式 f12.6
 - 接下来 2 行，不同自旋的占据数
 - * 第 1 列，不同自旋的占据数标识，sup 和 sdn，格式 a
 - * 第 2 列，不同自旋的占据数，格式 f12.6
 - 接下来 1 行，总占据数

⁹solver.hub.dat 文件在每次 DMFT 迭代计算后才会被更新一次。

¹⁰solver.nmat.dat 文件在每次 DMFT 迭代计算后才会被更新一次。

- * 第 1 列, 总占据数标识, sum, 格式 a
- * 第 2 列, 总占据数, 格式 f12.6
- 第 2 部分, 双占据数 $\langle n_i n_j \rangle$, 该部分以描述文字" <n_i n_j > data: "开头
 - 接下来 norbs*norbs 行, 双占据数 $\langle n_i n_j \rangle$
 - * 第 1 列, 轨道指标 i , 自 1 变化至 norbs, 格式 i5
 - * 第 2 列, 轨道指标 j , 自 1 变化至 norbs, 格式 i5
 - * 第 3 列, 双占据数 $\langle n_i n_j \rangle$, 格式 f12.6

5.2.11 solver.schi.dat

solver.schi.dat 文件包含了自旋 – 自旋关联函数 $\langle S_z(0)S_z(\tau) \rangle$ 。仅适用于 *Gardenia* 和 *Narcissus* 组件, 当计算模式被设为 (isvrt = 2, 参见第6.6节) 时才会输出该文件¹¹。其文件共有 nband + 2 个 block, 每个 block 之间间隔两个空行, 最后一个 block 后面有两个空行。每个 block 的格式如下:

- 第 1 列, 虚时点 τ_i , 一共 ntime 个点, 自 0 变化至 β , 格式 f12.6
- 第 2 列, $\langle S_z(0)S_z(\tau_i) \rangle$, 格式 f12.6

前 nband 个 block 是轨道分辨的自旋 – 自旋关联函数 $\langle S_z^i(0)S_z^i(\tau) \rangle$, 轨道指标 i 自 1 变化至 nband, 这些 block 以描述文字" #flvr: i "开头; 第 nband + 1 个 block 是平均的自旋 – 自旋关联函数 $\langle S_z(0)S_z(\tau) \rangle / \text{nband}$, 该 block 以描述文字" #flvr: 8888 "开头; 第 nband + 2 个 block 也是平均的自旋 – 自旋关联函数, 但是采用了另外一种平均方法 $(\sum_i \langle S_z^i(0)S_z^i(\tau) \rangle) / \text{nband}$, 该 block 以描述文字" #flvr: 9999 "开头。如果需要使用 gnuplot 绘图, 可用如下的命令:

```
gnuplot> plot 'solver.schi.dat' u 1:2 w lp
```

5.2.12 solver.ochi.dat

solver.ochi.dat 文件包含了轨道 – 轨道关联函数 $\langle N(0)N(\tau) \rangle$ 。仅适用于 *Gardenia* 和 *Narcissus* 组件, 当计算模式被设为 (isvrt = 3, 参见第6.6节) 时才会输出该文件¹²。其文件共有 norbs + 2 个 block, 每个 block 之间间隔两个空行, 最后一个 block 后面有两个空行。每个 block 的格式如下:

- 第 1 列, 虚时点 τ_i , 一共 ntime 个点, 自 0 变化至 β , 格式 f12.6
- 第 2 列, $\langle N(0)N(\tau_i) \rangle$, 格式 f12.6

¹¹solver.schi.dat 文件在每次 DMFT 迭代计算后才会被更新一次。

¹²solver.ochi.dat 文件在每次 DMFT 迭代计算后才会被更新一次。

前 norbs 个 block 是轨道分辨的轨道 - 轨道关联函数 $\langle N^i(0)N^i(\tau) \rangle$, 轨道指标 i 自 1 变化至 norbs , 这些 block 以描述文字" # flvr: i "开头; 第 $\text{norbs} + 1$ 个 block 是平均的轨道 - 轨道关联函数 $\langle N(0)N(\tau) \rangle / \text{nband}$, 该 block 以描述文字" # flvr: 8888 "开头; 第 $\text{norbs} + 2$ 个 block 也是平均的轨道 - 轨道关联函数, 但是采用了另外一种平均方法 $(\sum_i \langle N^i(0)N^i(\tau) \rangle) / \text{nband}$, 该 block 以描述文字" # flvr: 9999 "开头。如果需要使用 gnuplot 绘图, 可用如下的命令:

```
gnuplot> plot 'solver.ochi.dat' u 1:2 w lp
```

5.2.13 solver.twop.dat

solver.twop.dat 文件包含了双粒子格林函数以及顶角函数。仅适用于 *Gardenia* 和 *Narcissus* 组件, 当计算模式被设为 ($\text{isvrt} = 4$, 参见第6.6节) 时才会输出该文件¹³。其文件一共有 $\text{norbs} \times \text{norbs} \times \text{nbfrq}$ 个 block, 每个 block 之间间隔两个空行, 最后一个 block 后面有两个空行。每个 block 的格式如下:

- 第 1 行, 描述性文字, 轨道指标 m , 自 1 变化至 norbs , 格式 i5
- 第 2 行, 描述性文字, 轨道指标 n , 自 1 变化至 norbs , 格式 i5
- 第 3 行, 描述性文字, 玻色频率 ν 的指标, 自 1 变化至 nbfrq , 格式 i5
- 接下来 $\text{nffrq} \times \text{nffrq}$ 行, 存储相关的数据
 - 第 1 列, 费米频率 ω' 的指标, 从 1 - nffrq 变化至 $\text{nffrq} - 1$ (步长为 2), 格式 i5
 - 第 2 列, 费米频率 ω 的指标, 从 1 - nffrq 变化至 $\text{nffrq} - 1$ (步长为 2), 格式 i5
 - 第 3 列, 实部, 总的双粒子格林函数, $\Re \chi_{\text{tot}}^{mn}(\omega, \omega', \nu)$, 格式 f16.8
 - 第 4 列, 虚部, 总的双粒子格林函数, $\Im \chi_{\text{tot}}^{mn}(\omega, \omega', \nu)$, 格式 f16.8
 - 第 5 列, 实部, 可约双粒子格林函数, $\Re \chi_0^{mn}(\omega, \omega', \nu)$, 格式 f16.8
 - 第 6 列, 虚部, 可约双粒子格林函数, $\Im \chi_0^{mn}(\omega, \omega', \nu)$, 格式 f16.8
 - 第 7 列, 实部, 不可约双粒子格林函数, $\Re \chi_{\text{irr}}^{mn}(\omega, \omega', \nu)$, 格式 f16.8¹⁴
 - 第 8 列, 虚部, 不可约双粒子格林函数, $\Im \chi_{\text{irr}}^{mn}(\omega, \omega', \nu)$, 格式 f16.8
 - 第 9 列, 实部, 全顶角函数 $\Re \mathcal{F}^{mn}(\omega, \omega', \nu)$, 格式 f16.8
 - 第 10 列, 虚部, 全顶角函数 $\Im \mathcal{F}^{mn}(\omega, \omega', \nu)$, 格式 f16.8

在许多情况下, 双粒子格林函数以及顶角函数的数值比较大, 用格式 f16.8 已经无法满足数据的输出要求, 后果就是在 solver.twop.dat 文件中出现很多无法识别的"*" 符号。如果出现此类情况, 请用户关注 ctqmc_dump.f90 文件中的 ctqmc_dump_twop() 子程序, 修改其中的数据输出格式, 例如将 f16.8 替换为 f24.8 或者是 f16.4, 然后重新编译程序, 重新进行计算。如果情况还未得到改善, 请重复上述步骤, 直至 solver.twop.dat 文件中的数据显示正常。

¹³solver.twop.dat 文件在每次 DMFT 迭代计算后才会被更新一次。

¹⁴ $\chi_{\text{irr}} = \chi_{\text{tot}} - \chi_0$ 。

5.2.14 solver.vrtx.dat

solver.vrtx.dat 文件包含了双粒子格林函数以及顶角函数。仅适用于 *Gardenia* 和 *Narcissus* 组件，当计算模式被设为 (isvrt = 5，参见第6.6节) 时才会输出该文件¹⁵。其文件一共有 $\text{norbs} \times \text{norbs} \times \text{nbfrq}$ 个 block，每个 block 之间间隔两个空行，最后一个 block 后面有两个空行。每个 block 的格式如下：

- 第 1 行，描述性文字，轨道指标 m ，自 1 变化至 norbs ，格式 i5
- 第 2 行，描述性文字，轨道指标 n ，自 1 变化至 norbs ，格式 i5
- 第 3 行，描述性文字，玻色频率 ν 的指标，自 1 变化至 nbfrq ，格式 i5
- 接下来 $\text{nffrq} \times \text{nffrq}$ 行，存储相关的数据
 - 第 1 列，费米频率 ω' 的指标，从 1 - nffrq 变化至 $\text{nffrq} - 1$ (步长为 2)，格式 i5
 - 第 2 列，费米频率 ω 的指标，从 1 - nffrq 变化至 $\text{nffrq} - 1$ (步长为 2)，格式 i5
 - 第 3 列，实部，总的双粒子格林函数， $\Re\chi_{\text{tot}}^{mn}(\omega, \omega', \nu)$ ，格式 f16.8
 - 第 4 列，虚部，总的双粒子格林函数， $\Im\chi_{\text{tot}}^{mn}(\omega, \omega', \nu)$ ，格式 f16.8
 - 第 5 列，实部，可约双粒子格林函数， $\Re\chi_0^{mn}(\omega, \omega', \nu)$ ，格式 f16.8
 - 第 6 列，虚部，可约双粒子格林函数， $\Im\chi_0^{mn}(\omega, \omega', \nu)$ ，格式 f16.8
 - 第 7 列，实部，不可约双粒子格林函数， $\Re\chi_{\text{irr}}^{mn}(\omega, \omega', \nu)$ ，格式 f16.8
 - 第 8 列，虚部，不可约双粒子格林函数， $\Im\chi_{\text{irr}}^{mn}(\omega, \omega', \nu)$ ，格式 f16.8
 - 第 9 列，实部，全顶角函数 $\Re\mathcal{F}^{mn}(\omega, \omega', \nu)$ ，格式 f16.8
 - 第 10 列，虚部，全顶角函数 $\Im\mathcal{F}^{mn}(\omega, \omega', \nu)$ ，格式 f16.8

在许多情况下，双粒子格林函数以及顶角函数的数值比较大，用格式 f16.8 已经无法满足数据的输出要求，后果就是在 solver.vrtx.dat 文件中出现很多无法识别的“*”符号。如果出现此类情况，请用户关注 ctqmc_dump.f90 文件中的 ctqmc_dump_vrtx() 子程序，修改其中的数据输出格式，例如将 f16.8 替换为 f24.8 或者是 f16.4，然后重新编译程序，重新进行计算。如果情况还未得到改善，请重复上述步骤，直至 solver.vrtx.dat 文件中的数据显示正常。

5.2.15 solver.hist.dat

solver.hist.dat 文件包含了连续时间量子杂质求解器微扰展开项的阶数分布 P_H 。所有量子杂质求解器组件在运行时刻均会输出该文件¹⁶。其文件格式如下：

¹⁵solver.vrtx.dat 文件在每次 DMFT 迭代计算后才会被更新一次。

¹⁶每隔 nwrite 次 Monte Carlo sampling，量子杂质求解器组件会输出一当前次的 P_H 到 solver.hist.dat 文件中。

- 第 1 行, 描述文字: “ # histogram: order | count | percent ”, 格式 a
- 接下来 mkink 行, 存储微扰展开项的阶数分布
 - 第 1 列, 阶数指标 i , 自 1 变化至 mkink, 格式 i5
 - 第 2 列, 阶数的统计数目, count, 格式 i12
 - 第 3 列, 阶数的统计百分比, percent, 格式 f12.6

请注意 0 阶的数据保存在文件的最后一行。如果需要使用 gnuplot 绘图, 可用如下的命令¹⁷:

```
gnuplot> plot [1:100]'solver.hist.dat' u 1:3 w lp
```

5.2.16 solver.prob.dat

solver.prob.dat 文件包含了原子组态的出现概率 P_{Γ} 。所有量子杂质求解器组件在运行时刻均会输出该文件¹⁸。该文件包含 3 部分, 其文件格式如下:

- 第 1 部分, 原子组态概率分布, 该部分以描述文字 “ # state probability: index | prob | occupy | spin ” 开头
 - 第 1 列, 原子组态指标 i , 自 1 变化至 ncfgs, 格式 i5
 - 第 2 列, 第 i 个原子组态的概率 P_{Γ} , 格式 f12.6
 - 第 3 列, 第 i 个原子组态的占据数 N_{Γ} , 格式 f12.6
 - 第 4 列, 第 i 个原子组态的自旋值 $S_{z,\Gamma}$, 格式 f12.6
- 第 2 部分, 具有不同占据数的原子组态的概率分布, 该部分以描述文字 “ # orbital probability: index | occupy | prob ” 开头
 - 第 1 列, 占据数指标 i , i 的变化范围视研究的具体系统而定, 格式 i5
 - 第 2 列, 占据数 N , 格式 f12.6
 - 第 3 列, 占据数为 N 的原子组态的概率分布 P_N , 格式 f12.6
- 第 3 部分, 具有不同自旋值的原子组态的概率分布, 该部分以描述文字 “ # spin probability: index | spin | prob ” 开头
 - 第 1 列, 自旋指标 i , i 的变化范围视研究的具体系统而定, 格式 i5
 - 第 2 列, 自旋 S_z , 格式 f12.6
 - 第 3 列, 自旋为 S_z 的原子组态的概率分布 P_{S_z} , 格式 f12.6

¹⁷一般来说, 微扰展开阶数大于 100 的情况十分少见, 因此只需绘制出前 100 阶的分布即可。

¹⁸solver.prob.dat 文件在每次 DMFT 迭代计算后才会被更新一次。

5.2.17 solver.kernel.dat

solver.kernel.dat 文件包含了双积分推迟作用函数 $\mathcal{K}(\tau)$ 的信息，仅仅 *Narcissus* 组件可能会输出它。*Narcissus* 量子杂质求解器组件在初始化时，会输出 $\mathcal{K}(\tau)$ 到 solver.kernel.dat 文件中。其文件格式如下：

- 第 1 列，虚时点的指标 i ，从 1 变化至 ntime，格式 i5
- 第 2 列，虚时间点 τ_i ，从 0 变化至 β ，格式 f12.6
- 第 3 列， $\mathcal{K}(\tau_i)$ ，格式 f12.6

注意：实际上 solver.kernel.dat 文件中的数据就是 solver.ktau.in 文件中数据的一个拷贝，因此如果 solver.ktau.in 文件不存在的话，那么 solver.kernel.dat 文件中的数据将全部为 0。关于 solver.ktau.in 文件的信息，请参阅第 4.5 节。

5.2.18 solver.status.dat

solver.status.dat 文件包含了量子杂质求解器的图形配置信息。当量子杂质求解器组件关闭时，它会输出当前的图形配置信息到 solver.status.dat 文件中。当下次量子杂质求解器组件重新启动时，它会在当前目录中搜索 solver.status.dat 文件。如果 solver.status.dat 文件存在，那么量子杂质求解器组件会读取该文件，并利用该文件中包含的信息进行初始化。借助 solver.status.dat 文件可以帮助我们调试程序，也可以促使量子杂质求解器快速达到平衡状态。

由于 solver.status.dat 文件关系到量子杂质求解器组件的内部实现细节，因此此处暂不公布其内在格式。我们强烈建议用户千万不要手动修改 solver.status.dat 文件，即使用户明确地知道自己在干什么。如有疑问，请与开发者联系。

第六章

iQIST 配置文件详解

此前在第4章的第4.1节已经介绍了 `solver.ctqmc.in` 配置文件的标准格式，但是并没有解释该文件所包含内容的具体含义。本章将详细描述 `solver.ctqmc.in` 配置文件中的各个控制参数。由于量子杂质求解器组件仍在不断的更新，控制参数的具体定义与具体作用也在不断变化当中，因此本章的内容仅作为用户的参考。关于各个控制参数的权威定义请参考相应量子杂质求解器的 `ctqmc_control.f90` 文件以及 `ctqmc_stream.f90` 文件。

6.1 isscf

含义: 控制量子杂质求解器组件的运行模式，该参数仅有如下两种可能的取值。

- `isscf = 1`，单步运行模式，量子杂质求解器仅仅被调用一次即中断。
- `isscf = 2`，迭代运行模式，利用内置 DMFT 自洽条件反复调用量子杂质求解器进行迭代求解 bethe 晶格上的 Hubbard 模型。

类型: integer

缺省: 1

组件: 全部

注解: `isscf = 1` 此运行模式通常在 LDA+DMFT 计算中使用。此外，如果需要进行 data binning 或者是精确计算观测量¹也可以应用此运行模式。在 `isscf = 1` 运行模式下，用户通常需要提供正确的 `solver.hyb.in` 文件 (参阅第4.3节) 与 `solver.eimp.in` 文件 (参阅第4.2节)。如果用户不提供这两个文件，那么量子杂质求解器也能够正常运行，只是将会采用缺省的杂化函数并且将杂质轨道的能级通通置为 0。

¹例如精确计算自旋 - 自旋关联函数与虚时格林函数。

6.2 issun

含义: 是否采用强制对称化, 该参数仅有如下两种可能的取值。

- $\text{issun} = 1$, 不采用强制对称化。
- $\text{issun} = 2$, 采用强制对称化, 利用对称矢量所定义的对称性对各观测量进行对称化。

类型: integer

缺省: 1

组件: 全部

注解: 如果 $\text{issun} = 2$, 那么对称矢量 $\text{symm}(\text{norbs})$ 十分关键, 它定义了不同轨道的对称性, 亦即那些轨道是简并的, 那些轨道是不简并的。如果 $\text{symm}(i) = \text{symm}(j)$, 那么认为第 i 条轨道与第 j 条轨道是简并的, 否则是非简并的。 $1 \leq \text{symm}(i) \leq \text{norbs}$ 。对称矢量可由文件 `solver.eimp.in` 中读入, 缺省值为 1。换言之, 如果在当前目录中不存在 `solver.eimp.in` 文件, 并且 $\text{issun} = 2$, 那么量子杂质求解器组件将会认为所有轨道都是简并的, 并依此对物理观测量进行强制对称化。

如果您需要求解的量子杂质模型包含了自旋 – 轨道耦合项, 那么我们强烈建议您采用如下配置: $\text{issun} = 2$, $\text{isspn} = 2$, 并且在 `solver.eimp.in` 文件中手动指定各个轨道之间的对称性。

如果您需要求解的量子杂质模型包含了磁性态, 那么我们强烈建议您采用如下配置: $\text{issun} = 2$, $\text{isspn} = 2$, 并且在 `solver.eimp.in` 文件中手动指定各个轨道之间的对称性。

实际上, issun 仅仅对 n_α , $G_\alpha(\tau)$, $\Delta_\alpha(\tau)$, $G_\alpha(i\omega)$ 与 $\Sigma_\alpha(i\omega)$ 等观测量有直接的影响, 如果量子杂质求解器是迭代执行的, 那么其余的观测量也会相应地受到影响。

关于 `solver.eimp.in` 文件的细节, 请参阅第 4.2 节。

6.3 isspn

含义: 是否对自旋自由度进行强制对称化, 该参数仅有如下两种可能的取值。

- $\text{isspn} = 1$, 非磁或者是顺磁计算, 自旋朝上的物理量与自旋朝下的物理量之间强制进行对称化。
- $\text{isspn} = 2$, 铁磁或者是反铁磁计算, 自旋朝上的物理量与自旋朝下的物理量之间不简并。

类型: integer

缺省: 1

组件: 全部

注解: 此控制参数通常与 issun 联用。从原则上说, isspn 可以恒定为 2, 令 $\text{issun} = 2$, 并且通过设定合适的 $\text{symm}(\text{norbs})$ (通过 `solver.eimp.in` 文件), 也可以对自旋自由度进行强制对称化。

isspn 与 issun 是两个独立的控制参量，量子杂质求解器首先依据 issun 的取值决定是否进行轨道对称化，然后再根据 isspn 的取值决定是否进行自旋对称化。前一种对称操作可能会被后一种对称操作所覆盖。例如考虑单带 Hubbard 模型，如果 issun = 2, 在 solver.eimp.in 文件中指定 $\text{symm}(1) = 1$, $\text{symm}(2) = 2$ ，或者直接设定 issun = 1，那么程序将不会对轨道 1 和轨道 2 进行对称化。接下来考虑 isspn，如果 isspn = 2，那么同样不进行自旋对称化，轨道 1 与轨道 2 之间不简并。如果 isspn = 1，那么将会在轨道 1 和轨道 2 之间进行对称化。

如果您需要求解的量子杂质模型包含了自旋 - 轨道耦合项，或者是包含了磁性态，那么 isspn 必须设为 2，否则计算结果是不准确的。

6.4 isbin

含义: 是否进入到 data binning 模式，该参数仅有如下两种可能的取值。

- isbin = 1, 正常计算模式。
- isbin = 2, data binning 模式。

类型: integer

缺省: 1

组件: 全部

注解: 当量子杂质求解器组件进入到 data binning 模式时，迭代数 (iter) 会固定为 999，nsweep 以及 nwrite 会自动变为输入值的 10 倍，虚时格林函数被周期性输出到 solver.green.bin.X 文件中，其中 X 代表 data bin 的编号，从 1 至 nsweep/nwrite，后处理程序²将利用这些 data bins 对虚时格林函数进行解析延拓。

isbin 与 isscf 组合，将可以得到如下四种不同的运行模式:

- A: isscf = 1, isbin = 1，量子杂质求解器组件执行一次，不涉及 DMFT 自治迭代，采用正常计算模式，不进行 data binning。
- B: isscf = 2, isbin = 1，量子杂质求解器组件执行多次，利用 DMFT 方程进行自治迭代，在迭代过程中采用正常计算模式，迭代完毕后即停止，不进行 data binning。
- C: isscf = 1, isbin = 2，量子杂质求解器组件执行一次，不涉及 DMFT 自治迭代，直接进入到 data binning 模式。
- D: isscf = 2, isbin = 2，量子杂质求解器组件执行多次，利用 DMFT 方程进行自治迭代，在迭代过程中采用正常计算模式，在迭代完毕后，量子杂质求解器组件进入到 data binning 模式中执行一次。

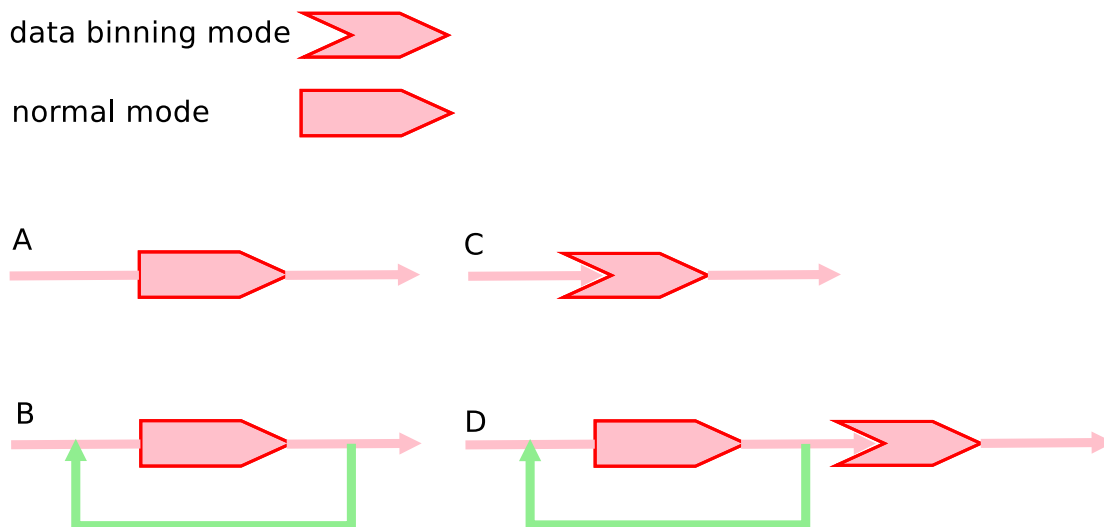


图 6.1: 量子杂质模型求解器组件的四种运行模式

关于上述四种运行模式的解释可以参考图6.1。如果用户需要进行 LDA + DMFT 计算，那么可以利用运行模式 A。模式 B 通常用于迭代求解一般的模型哈密顿量，粗略判断计算结果是否合理。如果用户已经迭代求解 DMFT 方程完毕，需要获得更为精确的结果，或者是为后处理步骤准备必须的数据，那么可以利用运行模式 C。运行模式 D 则相当于模式 B 与模式 C 的组合，一般很少使用。

6.5 isort

含义: 是否应用正交多项式模式改善物理量的测量精度，该参数仅有如下六种可能的取值。

- isort = 1, 采用正常方法测量 $G(\tau)$ 。
- isort = 2, 采用 Legendre 正交多项式方法测量 $G(\tau)$ 。
- isort = 3, 采用 Chebyshev 正交多项式方法测量 $G(\tau)$ 。
- isort = 4, 采用正常方法测量 $G(\tau)$ 与 $F(\tau)$ 。
- isort = 5, 采用 Legendre 正交多项式方法测量 $G(\tau)$ 与 $F(\tau)$ 。
- isort = 6, 采用 Chebyshev 正交多项式方法测量 $G(\tau)$ 与 $F(\tau)$ 。

$G(\tau)$ 为虚时杂质格林函数， $F(\tau)$ 为附加关联函数，利用它可以解析地计算出 $\Sigma(i\omega)$ ，提高自洽计算的稳定性与可靠性。

类型: integer

²例如 Hibiscus 组件中的 maketau 程序，请参阅第7.5节。

缺省:1

组件: 仅 *Gardenia*、*Narcissus*、*Lavender* 组件

注解: *Gardenia* 与 *Narcissus* 组件支持 $\text{isort} = 1 \sim 6$, 但是 *Lavender* 组件仅仅支持 $\text{isort} = 1 \sim 3$ 。

当 $\text{isort} = 1$ 时, 采用正常方法测量 $G(\tau)$ 与 $G(i\omega)$ 的低频部分。然后在 `ctqmc_make_hub1()` 子程序中, 利用 Dyson 方程计算低频部分的自能函数 $\Sigma(i\omega)$, 将其与高频部分的原子自能函数 $\Sigma_{\text{atom}}(i\omega)$ 拼接起来, 最后再利用 Dyson 方程计算最终的 $G(i\omega)$ 。

当 $\text{isort} = 2$ 时, 采用 Legendre 正交多项式方法测量 $G(\tau)$, 采用正常方法测量 $G(i\omega)$ 的低频部分。然后在 `ctqmc_make_hub1()` 子程序中, 利用 Dyson 方程计算低频部分的自能函数 $\Sigma(i\omega)$, 将其与高频部分的原子自能函数 $\Sigma_{\text{atom}}(i\omega)$ 拼接起来, 最后再利用 Dyson 方程计算最终的 $G(i\omega)$ 。

当 $\text{isort} = 3$ 时, 采用 Chebyshev 正交多项式方法测量 $G(\tau)$, 采用正常方法测量 $G(i\omega)$ 的低频部分。然后在 `ctqmc_make_hub1()` 子程序中, 利用 Dyson 方程计算低频部分的自能函数 $\Sigma(i\omega)$, 将其与高频部分的原子自能函数 $\Sigma_{\text{atom}}(i\omega)$ 拼接起来, 最后再利用 Dyson 方程计算最终的 $G(i\omega)$ 。

当 $\text{isort} = 4$ 时, 首先通过正常方法测量获得 $F(\tau)$ 与 $G(\tau)$, 然后利用 Fourier 变换, 获得 $F(i\omega)$ 与 $G(i\omega)$, 最后利用解析表达式直接给出 $\Sigma(i\omega)$ 。

当 $\text{isort} = 5$ 时, 首先通过 Legendre 正交多项式方法测量获得 $F(\tau)$ 与 $G(\tau)$, 然后利用解析表达式直接获得 $F(i\omega)$ 与 $G(i\omega)$, 最后利用解析表达式直接给出 $\Sigma(i\omega)$ 。

当 $\text{isort} = 6$ 时, 首先通过 Chebyshev 正交多项式方法测量获得 $F(\tau)$ 与 $G(\tau)$, 然后利用 Fourier 变换, 获得 $F(i\omega)$ 与 $G(i\omega)$, 最后利用解析表达式直接给出 $\Sigma(i\omega)$ 。

总体而言, 采用正交多项式方法能够有效地改善 $G(\tau)$ 与 $F(\tau)$ 的测量精度。尤其是在金属区域, 改善幅度十分显著, 但是在绝缘体区域, 在 $G(\tau)$ 的底部可能会诱发 Gibbs 振荡。解决办法是采用内核多项式方法, 具体措施请参考附录 A.8。

采用正交多项式方法还会轻微地降低计算效率。通过解析方法 ($\text{isort} = 4 \sim 6$) 计算 $\Sigma(i\omega)$ 的效果非常好, 推荐采用, 但是由于要额外测量关联函数 $F(\tau)$, 因此计算效率还会进一步降低, 同时广义相互作用版本的量子杂质模型求解器暂不支持此功能。

采用 Chebyshev 或者是 Legendre 正交多项式的效果没有显著的差别, 但是如果选用 Legendre 正交多项式, 那么有解析的表达式可以直接计算 $F(i\omega)$ 与 $G(i\omega)$, 避免了 Fourier 变换步骤, 因此可能更为优越。

关于正交多项式方法在 CT-QMC 量子杂质求解器中的应用, 请参阅 L. Boehnke 等人的工作⁶⁶ 以及 H. Hafermann 等人的工作⁶⁹。

6.6 isvrt

含义: 是否计算高阶关联函数, 该参数仅有如下五种可能的取值。

- $\text{isvrt} = 1$, 不计算任何高阶关联函数。
- $\text{isvrt} = 2$, 计算自旋 - 自旋关联函数 $\langle S_z(0)S_z(\tau) \rangle^3$ 。
- $\text{isvrt} = 3$, 计算轨道 - 轨道关联函数 $\langle N(0)N(\tau) \rangle^4$ 。
- $\text{isvrt} = 4$, 计算双粒子格林函数与顶角函数。
- $\text{isvrt} = 5$, 计算双粒子格林函数与顶角函数。

类型:integer

缺省:1

组件: 仅 *Gardenia* 、 *Narcissus* 组件

注解: 当 $\text{isvrt} = 2$ 时, 将会产生输出文件 `solver.schi.dat` ; 当 $\text{isvrt} = 3$ 时, 将会产生输出文件 `solver.ochi.dat` ; 当 $\text{isvrt} = 4$ 时, 将会产生输出文件 `solver.twop.dat` ; 当 $\text{isvrt} = 5$ 时, 将会产生输出文件 `solver.vrtx.dat`。

取 $\text{isvrt} = 4$ 与 $\text{isvrt} = 5$ 时均会输出双粒子格林函数以及顶角函数, 输出文件的格式完全一样。唯一的区别在于所用的算法不同。 $\text{isvrt} = 5$ 所用算法更好一些, 计算量也要大一些。

如果要计算高阶关联函数, 那么计算量会很大, 需要耗费较多的计算资源, 因此在迭代计算过程中 (即 $\text{isscf} = 2$) 不建议打开此参数。待 DMFT 自洽迭代计算完毕后, 再利用 $\text{isscf} = 1$ 单步计算模式并结合 isvrt 的设置来对高阶关联函数进行测量。关于 isscf 参数的详情, 请参阅第6.1节。

采用正交多项式算法 ($\text{isort} = 2, 3, 5, 6$) 对双粒子格林函数以及顶角函数 ($\text{isvrt} = 4, 5$) 的测量精度有正面的效果。关于 isort 参数的详情, 请参阅第6.5节。

6.7 isscr

含义: 是否在计算中支持动态屏蔽效应或者是 Hubbard-Holstein 模型, 该参数仅有如下五种可能的取值。

- $\text{isscr} = 1$, 正常计算模式。
- $\text{isscr} = 2$, 考虑 Hubbard-Holstein 模型。
- $\text{isscr} = 3$, 考虑动态屏蔽效应, palsmon pole 模型。
- $\text{isscr} = 4$, 考虑动态屏蔽效应, ohmic 模型。
- $\text{isscr} = 99$, 考虑动态屏蔽效应, 真实材料计算。

³包括轨道分辨与平均两种类型。

⁴包括轨道分辨与平均两种类型。

类型:integer

缺省:1

组件: 仅 *Narcissus* 组件

注解: 当 $\text{isscr} > 1$ 时, 需要通过 lc 参数 (请参阅第6.23节) 与 wc 参数 (请参阅第6.24节) 来定义相关的模型。当 $\text{isscr} = 99$ 时, 用户必须在当前计算目录下提供 `solver.ktau.in` 文件作为输入, 该文件包含了关键的双推迟作用函数 $\mathcal{K}(\tau)$ 。如果该文件不存在, 则程序会报错退出。关于 `solver.ktau.in` 文件的细节, 请参阅第4.5节与第7.5节。

关于动态屏蔽效应的细节, 请参阅 P. Werner 等人的工作²⁹, 关于 CT-HYB 量子杂质求解器在 Hubbard-Holstein 模型中的应用, 请参阅 P. Werner 等人的工作²⁸。

6.8 nband

含义: 量子杂质模型中能带的数目, 对于单带 Hubbard 模型而言 $\text{nband} = 1$, 对于三带 Hubbard 模型而言 $\text{nband} = 3$, 其余可以依此类推。

类型:integer

缺省:1

组件: 全部

注解: 显而易见, nband 的取值范围为 $7 \geq \text{nband} \geq 1$ 。

6.9 nspin

含义: 自旋的两个投影方向。

类型:integer

缺省:2

组件: 全部

注解: 请勿修改 nspin 的预置值 2。

6.10 norbs

含义: 量子杂质模型中轨道的数目。 $\text{norbs} = \text{nband} \times \text{nspin}$ 。

类型:integer

缺省:2

组件: 全部

注解: 请保证 `norbs` 的值与 `nband` 的值自治, 程序本身不做检查。

在 `iQIST` 软件包所有计算组件的设计与实现中, 无论是输入/输出, 还是在程序内部, 都是先安排自旋朝上的轨道, 再排自旋朝下的轨道, 亦即 $1 \uparrow 2 \uparrow 3 \uparrow \dots 1 \downarrow 2 \downarrow 3 \downarrow$, 其中 $1, 2, 3$ 为能带指标。

6.11 `ncfgs`

含义: 原子组态的数目。

类型: integer

缺省: 4

组件: 全部

注解: 对于 `Azalea`、`Gardenia`、`Narcissus` 等基于 `segment` 表象算法的组件而言, $\text{ncfgs} = 2^{\text{norbs}}$, 对于 `Begonia`、`Lavender`、`Camellia`、`Epiphyllum`、`Pansy`、`Manjushaka` 等基于广义矩阵表象算法的组件而言, 由于可以采用原子组态截断近似来加速, 因此 $1 \leq \text{ncfgs} \leq 2^{\text{norbs}}$ 。请保证 `ncfgs` 的值与 `norbs` 的值自治, 程序本身不做检查。

6.12 `nzero`

含义: 在普通的广义相互作用版本的量子杂质求解器中, 采用稀疏矩阵格式 (CSR) 存储 \mathcal{F} 矩阵, `nzero` 表示非零矩阵元的最大数目。

类型: integer

缺省: 128

组件: 仅 `Begonia` 与 `Lavender` 组件

注解: 如果 `nzero` 过小, 那么程序会报错退出, 如果 `nzero` 过大, 那么计算效率会较低。 `nzero` 最好设置为 2 的幂次方。

6.13 `nvect`

含义: 在基于 Krylov 子空间迭代算法或者是 Newton-Leja 多项式插值算法的量子杂质模型求解器中 (亦即 `Camellia` 与 `Epiphyllum` 组件), 可以通过截断近似来提高计算效率。 `nvect` 表示保留下来的本征态数目。

类型: integer

缺省: 4

组件: 仅 `Camellia` 与 `Epiphyllum` 组件

注解: 在截断近似中, 一般而言仅仅只保留属于基态的那几个本征态, 这称之为 $O(1)$ 近似; 如果还保留了第一激发态, 那么称之为 $O(2)$ 近似。

关于 Krylov 子空间迭代算法在 CT-HYB 量子杂质求解器中的应用, 请参阅 P. Werner 等人的工作³⁰。

6.14 nhmat

含义: 在基于 Krylov 子空间迭代算法或者是 Newton-Leja 多项式插值算法的量子杂质模型求解器中 (亦即 *Camellia* 与 *Epiphyllum* 组件), 采用稀疏矩阵格式 (CSR) 存储局域哈密顿量 \mathcal{H} 矩阵, nhmat 表示非零矩阵元的最大数目。

类型: integer

缺省: 128

组件: 仅 *Camellia* 与 *Epiphyllum* 组件

注解: 如果 nhmat 过小, 那么程序会报错退出, 如果 nhmat 过大, 那么计算效率会较低。nhmat 最好设置为 2 的幂次方。

关于 Krylov 子空间迭代算法在 CT-HYB 量子杂质求解器中的应用, 请参阅 P. Werner 等人的工作³⁰。

6.15 nfmat

含义: 在基于 Krylov 子空间迭代算法或者是 Newton-Leja 多项式插值算法的量子杂质模型求解器中 (亦即 *Camellia* 与 *Epiphyllum* 组件), 采用稀疏矩阵格式 (CSR) 存储 \mathcal{F} 矩阵, nfmat 表示非零矩阵元的最大数目。

类型: integer

缺省: 128

组件: 仅 *Camellia* 与 *Epiphyllum* 组件

注解: 如果 nfmat 过小, 那么程序会报错退出, 如果 nfmat 过大, 那么计算效率会较低。nfmat 最好设置为 2 的幂次方。nfmat 参数的含义与 nzero 参数的含义类似, 只不过是用在不同的组件当中。关于 nzero 参数的详情, 请参阅第 6.12 节。

关于 Krylov 子空间迭代算法在 CT-HYB 量子杂质求解器中的应用, 请参阅 P. Werner 等人的工作³⁰。

6.16 niter

含义:DMFT 自治迭代的最大次数。

类型:integer

缺省:20

组件: 全部

注解: 仅当 $\text{isscf} = 2$ 时有意义。DMFT 自治迭代的最小次数为 16 次, 在 `ctqmc_dmft_conver()` 子程序内部指定。如果需要改动它, 那么必须重新编译程序。关于 `isscf` 参数的详情, 请参阅第6.1节。

6.17 U

含义: 平均 Coulomb 相互作用。

类型:real(dp)

缺省:4.0

组件: 全部

注解: 实际上在程序中没有用到它。习惯上将 U 与 U_c 的值设为一致。

6.18 U_c

含义: 轨道内的 Coulomb 相互作用。

类型:real(dp)

缺省:4.0

组件: 全部

注解: 对于 *Begonia* 与 *Lavender* 组件而言无意义, 但是对 *Azalea*, *Gardenia* 与 *Narcissus* 组件而言有意义。满足 $U_c = U_v - 2J_z$ 关系。关于 U_v 参数的详情, 请参阅第6.19节。

6.19 U_v

含义: 轨道间的 Coulomb 相互作用。

类型:real(dp)

缺省:4.0

组件: 全部

注解: 对于 *Begonia* 与 *Lavender* 组件而言无意义, 但是对 *Azalea*, *Gardenia* 与 *Narcissus* 组件而言有意义。满足 $U_c = U_v - 2J_z$ 关系。关于 U_c 参数的详情, 请参阅第6.18节。

6.20 J_z

含义: Hund 交换常数。

类型: real(dp)

缺省: 0.0

组件: 全部

注解: 对于 *Begonia* 与 *Lavender* 组件而言无意义, 但是对 *Azalea*, *Gardenia* 与 *Narcissus* 组件而言有意义。对于单带模型, $J_z = J_s = J_p = 0$ 。满足 $U_c = U_v - 2J_z$ 关系。

6.21 J_s

含义: 自旋翻转项 (spin-flip) 常数。

类型: real(dp)

缺省: 0.0

组件: 全部

注解: 在程序中实际上没有使用。

6.22 J_p

含义: 对跃迁项 (pair-hopping) 常数。

类型: real(dp)

缺省: 0.0

组件: 全部

注解: 在程序中实际上没有使用。

6.23 lc

含义: 用于定义 Hubbard-Holstein 模型或者是动态屏蔽效应的模型参数。lc 的具体含义与作用与 isscr 参数密切相关。

- isscr = 1, 无意义
 - isscr = 2, Hubbard-Holstein 模型, λ 参数
 - isscr = 3, 动态屏蔽效应, plasmon pole 模型, λ 参数
-

- $\text{isscr} = 4$, 动态屏蔽效应, ohmic 模型, α 参数
- $\text{isscr} = 99$, 动态屏蔽效应, 实际材料, 取为 [Hibiscus](#) /hibiscus-toolbox/makescr 程序输出值

类型:real(dp)

缺省:2.50

组件: 仅 [Narcissus](#) 组件

注解: 仅当 $\text{isscr} > 1$ 时才有意义。关于 isscr 参数的详情, 请参阅第6.7节。关于 Hubbard-Holstein 模型的细节, 请参阅 P. Werner 的 PRL 论文²⁸。关于动态屏蔽效应, plasmon pole 模型与 ohmic 模型的细节, 请参阅 P. Werner 的另一篇 PRL 论文²⁹。

6.24 wc

含义: 用于定义 Hubbard-Holstein 模型或者是动态屏蔽效应的模型参数。wc 的具体含义与作用与 isscr 参数密切相关。

- $\text{isscr} = 1$, 无意义
- $\text{isscr} = 2$, Hubbard-Holstein 模型, ω_0 参数
- $\text{isscr} = 3$, 动态屏蔽效应, plasmon pole 模型, ω' 参数
- $\text{isscr} = 4$, 动态屏蔽效应, ohmic 模型, ω_c 参数
- $\text{isscr} = 99$, 动态屏蔽效应, 实际材料, 无意义

类型:real(dp)

缺省:3.00

组件: 仅 [Narcissus](#) 组件

注解: 仅当 $\text{isscr} > 1$ 时才有意义。关于 isscr 参数的详情, 请参阅第6.7节。关于 Hubbard-Holstein 模型的细节, 请参阅 P. Werner 的 PRL 论文²⁸。关于动态屏蔽效应, plasmon pole 模型与 ohmic 模型的细节, 请参阅 P. Werner 的另一篇 PRL 论文²⁹。

6.25 mune

含义: 化学势 μ 。

类型:real(dp)

缺省:2.0

组件: 全部

注解: 量子杂质求解器软件在缺省状态下是固定化学势求占据数。如果用户需要固定占据数搜索化学势, 那么需要对程序进行小的修改, 请参阅附录A.9。

6.26 beta

含义: 反系统温度, $\beta = \frac{1}{T}$ 。

类型: real(dp)

缺省: 10.0

组件: 全部

注解: 单位为 eV^{-1} 。beta 的取值范围为 $0.5 \leq \beta \leq 400$, 但是仅仅 *Azalea* 与 *Gardenia* 对 $\beta < 5$ 有较好的支持。

当 $\beta < 5$ 时, *Azalea* 与 *Gardenia* 组件会自动启用内置的 GLOBAL SWAP 算法, 该算法对于高温体系十分适用, 其它的量子杂质求解器暂不具备此功能, 详情请参阅第1.4节。

6.27 part

含义: Hubbard 模型中的 hopping 参数。

类型: real(dp)

缺省: 0.5

组件: 全部

注解: part 参数主要用在构造初始的虚频杂化函数 $\Delta(i\omega)$ 以及 DMFT 自洽迭代条件中, 仅当 isscf = 2 时会对计算结果产生影响。关于 isscf 参数的详情, 请参阅第6.1节。

6.28 alpha

含义: 线性混合因子, 在 DMFT 自洽迭代计算时用于混合新旧虚频杂化函数。

$$\Delta_{\text{new}} = (1 - \alpha)\Delta_{\text{old}} + \alpha\Delta_{\text{cur}} \quad (6.1)$$

类型: real(dp)

缺省: 0.70

组件: 全部

注解: 仅当 isscf = 2 时才有意义, 一般无须改动, 但是较小的 alpha 值可能更适合难收敛的系统。关于 isscf 参数的详情, 请参阅第6.1节。

6.29 lemax

含义: Legendre 正交多项式最大所允许的展开阶数，必须大于 2。

类型: integer

缺省: 32

组件: 仅 *Gardenia* , *Narcissus* , 与 *Cavender* 组件

注解: 仅当 $\text{isort} = 2$ 或 5 时有意义。lemax 取值太小计算结果不够精确，取值太大则易诱发 Gibbs 振荡。32, 48 都是比较好的选择。特别需要注意的是由于算法的限制，lemax 不可大于 50! 关于 isort 参数的详情，请参阅第6.5节。

关于正交多项式方法在 CT-QMC 量子杂质求解器中的应用，请参阅 L. Boehnke 等人的工作⁶⁶ 以及 H. Hafermann 等人的工作⁶⁹。

6.30 legrd

含义: 在 $[-1,1]$ 间建立一个线性网格，Legendre 正交多项式即定义在此线性网格上，legrd 为网格点的数目。

类型: integer

缺省: 20001

组件: 仅 *Gardenia* , *Narcissus* , 与 *Cavender* 组件

注解: 仅当 $\text{isort} = 2$ 或 5 时有意义。一般无须改动。关于 isort 参数的详情，请参阅第6.5节。

关于正交多项式方法在 CT-QMC 量子杂质求解器中的应用，请参阅 L. Boehnke 等人的工作⁶⁶ 以及 H. Hafermann 等人的工作⁶⁹。

6.31 chmax

含义: Chebyshev 正交多项式 (第二类) 最大所允许的展开阶数，必须大于 2。

类型: integer

缺省: 32

组件: 仅 *Gardenia* , *Narcissus* , 与 *Cavender* 组件

注解: 仅当 $\text{isort} = 3$ 或 6 时有意义。chmax 取值太小计算结果不够精确，取值太大则易诱发 Gibbs 振荡。32, 48 都是比较好的选择，无上限限制。关于 isort 参数的详情，请参阅第6.5节。

关于正交多项式方法在 CT-QMC 量子杂质求解器中的应用，请参阅 L. Boehnke 等人的工作⁶⁶ 以及 H. Hafermann 等人的工作⁶⁹。

6.32 chgrd

含义: 在 $[-1,1]$ 间建立一个线性网格, Chebyshev 正交多项式 (第二类) 即定义在此线性网格上, chgrd 为网格点的数目。

类型: integer

缺省: 20001

组件: 仅 *Gardenia* , *Narcissus* , 与 *Lavender* 组件

注解: 仅当 `isort = 3` 或 `6` 时有意义。一般无须改动。关于 `isort` 参数的详情, 请参阅第6.5节。

关于正交多项式方法在 CT-QMC 量子杂质求解器中的应用, 请参阅 L. Boehnke 等人的工作⁶⁶ 以及 H. Hafermann 等人的工作⁶⁹。

6.33 mkink

含义: 图形微扰展开最大允许的阶数。

类型: integer

缺省: 1024

组件: 全部

注解: mkink 一般取为 2 的幂次方, 如 256, 512, 1024 等等。减少 mkink 的值可以略微的提高计算效率。1024 是一个经典的取值, 一般无须改动。

6.34 mfreq

含义: 松原频率点的数目。

类型: integer

缺省: 8193 ($= 2^{13} + 1$)

组件: 全部

注解: $G(i\omega)$, $\Sigma(i\omega)$, $\Delta(i\omega)$ 都定义在此松原频率网格上。8193 为经典设置, 一般无须改动。

6.35 nffrq

含义: 双粒子格林函数 $\chi(i\omega, i\omega', \nu)$ 与顶角函数 $\mathcal{F}(i\omega, i\omega', \nu)$ 有三个频率指标。其中 ω 与 ω' 为费米频率, 而 ν 为玻色频率。nffrq 为费米频率点的数目。

$$\nu_n = \frac{2n\pi}{\beta}$$

$$\omega_m = \frac{(2m+1)\pi}{\beta}$$

类型:integer

缺省:32

组件: 仅 *Gardenia* 与 *Narcissus* 组件

注解: nffrq 通常取为 2 的幂次方, 如 32, 64 等。双粒子格林函数以及顶角函数的计算十分耗时, 因此 nffrq 不宜设得过大。玻色频率点的数目由 nbfrq 参数指定, 详情请参阅第6.36节。

6.36 nbfrq

含义: 双粒子格林函数 $\chi(i\omega, i\omega', \nu)$ 与顶角函数 $\mathcal{F}(i\omega, i\omega', \nu)$ 有三个频率指标。其中 ω 与 ω' 为费米频率, 而 ν 为玻色频率。nbfrq 为玻色频率点的数目。

$$\nu_n = \frac{2n\pi}{\beta}$$

$$\omega_m = \frac{(2m+1)\pi}{\beta}$$

类型:integer

缺省:8

组件: 仅 *Gardenia* 与 *Narcissus* 组件

注解: nbfrq 通常较小, 一般设为 2 的幂次方。双粒子格林函数以及顶角函数的计算十分耗时, 因此 nbfrq 不宜设得过大。费米频率点的数目由 nffrq 参数指定, 详情请参阅第6.35节。

6.37 nfreq

含义: 直接进行测量的松原频率点的数目。我们可以直接在 Monte Carlo 抽样过程中测量 $G(i\omega)$, 但是这涉及到复数计算, 因此计算效率并不是很高。因此我们通常仅仅只测量处于低频的 nfreq 个点, 至于剩余高频点上的数据则用原子极限时的数据补齐。

类型:integer

缺省:128

组件: 全部

注解: 一般不修改该参数。

6.38 ntime

含义: 虚时间轴 $[0, \beta)$ 划分为时间片的数目。

类型:integer

缺省:1024

组件: 全部

注解: β 越大, 那么 ntime 就要越大, 以保证插值的精度。根据我们的经验, 当 $\beta < 40$, ntime 取 1024 为宜。当 $\beta > 40$, ntime 取 2048 为宜。

6.39 nleja

含义: Leja 点的数目。如果 nleja < 0 那么表明 Krylov 子空间迭代方法启用, 反之则是 Newton-Leja 多项式插值方法启用。

类型: integer

缺省: 64

组件: 仅 Camellia 与 Epiphyllum 组件

注解: 一般无须改动。

关于 Krylov 子空间迭代算法在 CT-HYB 量子杂质求解器中的应用, 请参阅 P. Werner 等人的工作³⁰。

6.40 npart

含义: 在广义相互作用版本的量子杂质模型求解器中, 为了提高计算效率, 通常将虚时间轴 $[0, \beta)$ 划分为 npart 个部份, 分别计算每个部份的贡献。在随机产生新的图形时, 首先判断是那个部份上的算符发生了变化, 然后再更新该部份的贡献。这就是所谓的“分而治之”的策略 (参见示意图6.2)。

类型: integer

缺省: 16

组件: 仅 Begonia 与 Cavender 组件

注解: npart 的取值十分讲究, 如果取值过小, 那么起不到加速的效果; 如果取值过大, 那么计算开销也会随之增大。假定当前模型的能带数目为 nband, 平均的微扰展开阶数为 $\langle k \rangle$, 那么 npart 的最优取值应符合如下公式:

$$2\sqrt{3\langle k \rangle \text{nband}} < \text{npart} < 4\sqrt{3\langle k \rangle \text{nband}} \quad (6.2)$$

6.41 nflip

含义: 控制量子杂质模型求解器进行 GLOBAL FLIP 操作的频率。亦即每隔 nflip 个 Monte Carlo sampling 进行一次 GLOBAL FLIP 操作。为了避免量子杂质模型求解器在运行过程中陷入到某种非物

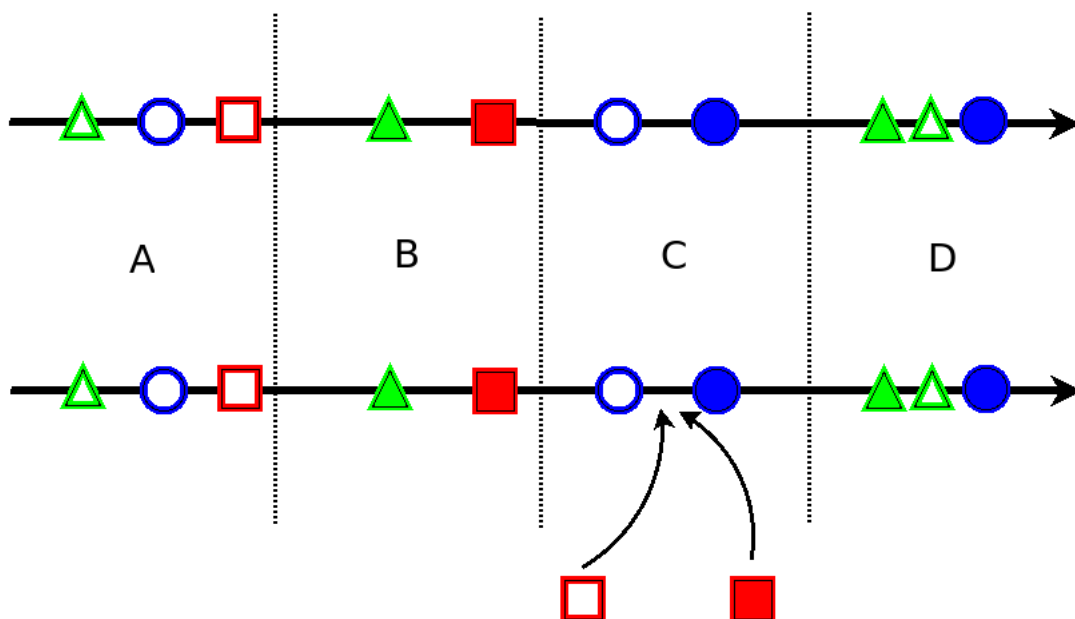


图 6.2: 分而治之的计算策略示意图。实心符号表示产生算符，空心符号表示表示消灭算符。不同的颜色与形状表示不同的轨道。图中展示的是往左起第三部分插入一对算符的情况。

理的有序态中，定期进行 GLOBAL FLIP 操作是十分有必要的。在 *iQIST* 中，我们定义了三种基本的 GLOBAL FLIP 操作，用内部整型变量 `cflip` 标示：

- `cflip = 1`，任意挑选两条对称性相同的轨道，对其上的量进行交换。
- `cflip = 2`，依次选择每条能带，交换其 (名义上的)spin up 和 spin down 的部分。
- `cflip = 3`，同时选择所有能带，各自交换其 (名义上的)spin up 和 spin down 的部分。

`nflip` 也有三种取值可能：

- `nflip = 0`，GLOBAL FLIP 的周期无限长，也就是不进行 GLOBAL FLIP。
- `nflip > 0`，组合 `cflip = 2` 和 `cflip = 3` 这两种 GLOBAL FLIP 模式，前者占 80%，后者占 20%。
- `nflip < 0`，组合 `cflip = 1` 和 `cflip = 3` 这两种 GLOBAL FLIP 模式，前者占 80%，后者占 20%。

类型:integer

缺省:2000

组件: 全部

注解: 如果需要考虑含自旋 - 轨道耦合的系统或者是系统具有磁序，建议选择 `nflip = 0`。

6.42 ntherm

含义: 在正式开始测量之间, 需要进行 thermalization, 以使得系统达到平衡。ntherm 表示在达到平衡之间, 需要进行的 Monte Carlo sampling 次数。

类型: integer

缺省: 200000

组件: 全部

注解: ntherm 一般不要超过 nsweep 的 1/10。

6.43 nsweep

含义: 总的 Monte Carlo sampling 次数。

类型: integer

缺省: 20000000

组件: 全部

注解: nsweep 不能太小, 太小无法保证测量精度。但是也不能太大, 最大不宜超过 200000000。在 data binning 计算模式 ($isbin = 2$, 参见第6.4节) 中, nsweep 和 nwrite 会自动增加为原来的十倍。如果 nsweep 的原始值太大, 那么最终的值会超出 integer 的数值范围, 导致运行错误。

在并行运行模式中, 假定采用 nprocs 个进程, 那么每个进程都同样执行 nsweep 个 Monte Carlo sampling, 而不是每个进程执行 $nsweep/nprocs$ 个 Monte Carlo sampling。因此, 如果用户采用大量处理器进行并行计算, 那么可以适当减少 nsweep 的数值。但是 nsweep 的最小取值不能小于 ntherm, 也不能小于 nwrite, nmonte, 与 ncarlo 中的任意一个值。我们强烈建议 nsweep 的最小取值不要小于缺省值。

6.44 nwrite

含义: 输出中间结果的频率。亦即每间隔 nwrite 次 Monte Carlo sampling, 量子杂质求解器会输出一些中间结果到文件中供用户分析, 用户可依此判断计算结果是否正确合理, 如果不正确, 那么可以及时终止任务, 节约计算时间。目前主要输出的中间结果主要有: E_{kin} , E_{pot} , E_{tot} , $\langle S_z \rangle$, $G(\tau)$ 与 histogram(亦即 P_H)。

类型: integer

缺省: 2000000

组件: 全部

注解: nwrite 不宜太小, 如果 nwrite 太小, 这意味着频繁输出中间结果, 这样有两个坏处:(1) 加重了 IO 的负担; (2) 每次输出之前, 各个并行进程之间要同步, 通信交换数据, 会降低执行效率。

此外，在 data binning 模式中 ($\text{isbin} = 2$)，还利用 nsweep 与 nwrite 的比值来决定 data bin 的数目。关于 isbin 参数的详情，请参阅第6.4节，关于 nsweep 参数的详情，请参阅第6.44节。

6.45 nclean

含义: 量子杂质模型求解器进行 clean update 的频率。亦即每间隔 nclean 次 Monte Carlo sampling 进行一次 clean update，以保证合理的数值精度。

类型: integer

缺省: 100000

组件: 全部

注解: 每间隔 nclean 步，量子杂质模型求解器会根据当前的图形配置从头计算 $G(i\omega)$ ，以消除长期累积的数值误差。此参数一般无须改动。

6.46 nmonte

含义: 用于控制观测量的测量间隔。受影响的物理量有 $G(i\omega)$ ， E_{tot} ， E_{kin} ， E_{pot} ， $\langle S_z \rangle$ ， $\langle n_i \rangle$ ， $\langle n_i n_j \rangle$ ， $\langle S_z(0) S_z(\tau) \rangle$ ， $\langle N(0) N(\tau) \rangle$ ， $\chi(i\omega, i\omega', \nu)$ 与 $\mathcal{F}(i\omega, i\omega', \nu)$ 。

类型: integer

缺省: 10

组件: 全部

注解: 为了避免相邻两次测量之间产生关联，这两次测量之间需要间隔 nmonte 个 Monte Carlo sampling。 nmonte 不宜过大，过大会浪费抽样时间，也不宜过小，过小则易产生关联。恰当的 nmonte 取值应满足以下关系：

$$\text{nmonte} * P_{\text{insert}} \geq 1$$

其中 P_{insert} 为 INSERT KINK 操作的接受概率，详情请参阅第5.1节。如果用户的计算对上述物理量的精度要求不高，那么可以把 nmonte 设为一个很大的数值，这样可以提高计算效率。

6.47 ncarlo

含义: 用于控制观测量的测量间隔。受影响的物理量有 $G(\tau)$ ， $F(\tau)$ 与 $P(\Gamma_i)$ 。

类型: integer

缺省: 10

组件: 全部

注解: 为了避免相邻两次测量之间产生关联, 这两次测量之间需要间隔 ncarlo 个 Monte Carlo sampling。ncarlo 不宜过大, 过大会浪费抽样时间, 也不宜过小, 过小则易产生关联。恰当的 ncarlo 取值应满足以下关系:

$$\text{ncarlo} * P_{\text{insert}} \geq 1$$

其中 P_{insert} 为 INSERT KINK 操作的接受概率, 详情请参阅第5.1 节。如果用户的计算对 $G(\tau)$, $F(\tau)$ 与 $P(\Gamma_i)$ 的精度要求不高, 那么可以把 ncarlo 设为一个很大的数值, 这样可以提高计算效率。

第七章

iQIST 附属工具程序

除了量子杂质模型求解器以外，iQIST 软件包中还包含了许多工具程序，可以完成众多的前处理与后处理工作。本章将对这些工具程序的用法做全面的介绍。

7.1 Jasmine 组件

Jasmine 组件属于原子问题程序，它的作用在于对角化局域哈密顿量，得到相应的本征值与本征态，产生 \mathcal{F} 矩阵，为广义相互作用版本的连续时间量子蒙特卡洛杂质求解器组件提供 atom.cix 输入文件。Jasmine 组件有三种运行模式，分别对应于以下三类量子杂质求解器组件：

- Begonia、Lavender
- Camellia、Epiphyllum
- Pansy、Manjushaka

这三类量子杂质求解器组件的主体框架是类似的，区别在于计算 trace 部分所使用的算法与技巧。Begonia 与 Lavender 组件应用了稀疏矩阵技术与分而治之的策略。Camellia 与 Epiphyllum 组件应用了 Krylov 子空间迭代算法以及 Newton-Leja 多项式插值算法。Pansy 与 Manjushaka 组件应用了好量子数技术。这些量子杂质求解器组件所需要的输入数据是大不相同的，自然地，相应的 atom.cix 文件的格式也是不大相同的。

Jasmine 组件的可执行程序名称为 atom，所需要的输入文件为 atom.in。典型的 atom.in 文件如下所示，这是一个 4 带模型的配置。

```

2 >>>      atom.in: configuration file for atomic eigenvalues problem      <<<
3
4 2          ! imode: running mode, 1 = eigen basis mode; 2 = occupation number basis mode
5 1          ! ifock: source of fock space, 1 = internal mode; 2 = external mode
6 -1         ! isoce: with or without SOC, -1 = without; 0 = s; 1 = p; 2 = d; 3 = f; 99 = debug mode
7
8 4          ! nband: number of correlated bands
9 2          ! nspin: number of spin projection
10 8          ! norbs: number of correlated orbitals (= nband * nspin)
11 256        ! ncfgs: number of atomic states (= 2**norbs)
12
13 6.00       ! U      : average Coulomb interaction
14 6.00       ! Uc     : intraorbital Coulomb interaction
15 3.00       ! Uv     : interorbital Coulomb interaction, Uv = Uc - 2 * Jz for t2g system
16 1.50       ! Jz     : Hund's exchange interaction in z axis (Jz = Js = Jp = J)
17 1.50       ! Js     : spin-flip term
18 1.50       ! Jp     : pair-hopping term
19 0.00       ! lsoc   : spin orbital coupling strength
20
21 -9.75000 ! eimp(01) orbital, impurity level
22 -9.75000 ! eimp(02) orbital
23 -9.75000 ! eimp(03) orbital
24 -9.75000 ! eimp(04) orbital
25 -9.75000 ! eimp(05) orbital
26 -9.75000 ! eimp(06) orbital
27 -9.75000 ! eimp(07) orbital
28 -9.75000 ! eimp(08) orbital
29 0.00000 ! eimp(09) orbital
30 0.00000 ! eimp(10) orbital
31 0.00000 ! eimp(11) orbital
32 0.00000 ! eimp(12) orbital
33 0.00000 ! eimp(13) orbital
34 0.00000 ! eimp(14) orbital

```

各输入参数的简明解释如下，至于详细权威的解释请参阅 *Jasmine* /atom_control.f90 文件中的注释。由于我们还需要对 *Jasmine* 组件进行深度的重构，因此这些参数的名称以及具体含义可能会发生较大的变化，用户需要特别注意。

- imode

设定 Jasmine 的运行模式。

imode = 1 时，所产生的 atom.cix 文件适合于 Begonia 与 Lavender 组件。

imode = 2 时，所产生的 atom.cix 文件适合于 Camellia 与 Epiphyllum 组件。

imode = 3 时，所产生的 atom.cix 文件适合于 Pansy 与 Manjushaka 组件，此选项暂不可用。

- ifock

设定 Fock 态的来源。

ifock = 1 时，由 Jasmine 自身产生 Fock 态。

ifock = 2 时，从外部文件 fock.in 中读取 Fock 态，此选项暂时不可用。

- isoce

是否考虑自旋 – 轨道耦合项 (SOC)。

isoce = -1 时，不支持 SOC。

isoce = 0 时，考虑 $l = 0$, s 轨道的情况。

isoce = 1 时，考虑 $l = 1$, p 轨道的情况。

isoce = 2 时，考虑 $l = 2$, d 轨道的情况。

isoce = 3 时，考虑 $l = 3$, f 轨道的情况。

isoce = 99 时，进入 debug 模式。

由于当前版本的 Jasmine 组件不支持 SOC，因此 isoce 必须设为 -1，否则所产生的 atom.cix 文件是不正确的。

- nband

能带的数目。

此参数雷同于 solver.ctqmc.in 文件中的 nband 参数，请参阅第6.8节。

- nspin

自旋取向的数目，恒定为 2。

此参数雷同于 solver.ctqmc.in 文件中的 nspin 参数，请参阅第6.9节。

- norbs

轨道的数目，等于 $nband \times nspin$ 。

此参数雷同于 solver.ctqmc.in 文件中的 norbs 参数，请参阅第6.10节。

- **ncfgs**

原子组态的数目，等于 2^{norbs} 。

此参数雷同于 solver.ctqmc.in 文件中的 ncfgs 参数，请参阅第6.11节。

- **U**

平均 Coulomb 相互作用。

此参数在程序中实际上不起作用，通常设为 $U = U_c$ 。此参数雷同于 solver.ctqmc.in 文件中的 U 参数，请参阅第6.17节。

- **U_c**

轨道内的 Coulomb 相互作用。

此参数雷同于 solver.ctqmc.in 文件中的 U_c 参数，满足 $U_c = U_v - 2 * J_z$ 关系，请参阅第6.18节。

- **U_v**

轨道间的 Coulomb 相互作用。

此参数雷同于 solver.ctqmc.in 文件中的 U_v 参数，满足 $U_c = U_v - 2 * J_z$ 关系，请参阅第6.19节。

- **J_z**

Hund 交换常数。

此参数雷同于 solver.ctqmc.in 文件中的 J_z 参数，满足 $U_c = U_v - 2 * J_z$ 关系，请参阅第6.20节。

- **J_s**

自旋翻转项 (spin-flip) 常数，通常设为 $J_z = J_s = J_p$ 。

此参数雷同于 solver.ctqmc.in 文件中的 J_s 参数，请参阅第6.21节。

- **J_p**

对跃迁项 (pair-hopping) 常数，通常设为 $J_z = J_s = J_p$ 。

此参数雷同于 solver.ctqmc.in 文件中的 J_p 参数，请参阅第6.22节。

- **lsoc**

自旋 - 耦合作用强度。

此参数仅当 isocce 不等于 -1 时起作用。

- **eimp**

杂质轨道能级 $E_{\text{imp}}(i)$ 。

此参数实际上是一个一维数组，先安排自旋朝上的轨道，后安排自旋朝下的轨道，没有用到的轨道全部都置为 0。

Jasmine 组件在运行后，会输出以下文件：

- atom.cix，最重要的输出文件，量子杂质求解器组件就需要它，文件格式请参阅第4.6节。
- atom.eimp.dat，包含杂质轨道能级信息。
- atom.eval.dat，包含原子问题的本征值。
- atom.evec.dat，包含原子问题的本征态。
- atom.fmat.dat，包含 \mathcal{F} 矩阵的数据。
- atom.fock.dat，包含 Fock 态的信息。
- atom.hist.dat，统计每个本征态的简并度。
- atom.hmat.dat，包含局域哈密顿量的非零矩阵元。
- atom.smat.dat，包括自旋 - 轨道耦合矩阵。
- atom.umat.dat，包括 Coulomb 相互作用矩阵的非零矩阵元。

除了 *Jasmine* 组件以外，利用特别定制的 rambutan 程序也能生成合格的 atom.cix 文件，计算结果是一致的。如果系统中包含了自旋 - 轨道耦合作用，那么 rambutan 程序是唯一的选择。关于 rambutan 程序的详情请与杜亮博士联系。

7.2 *Hibiscus* /hibiscus-entropy 组件

Hibiscus 组件包含了四个子组件，分别为：

- entropy
- stochastic
- swing
- toolbox

其中前三个子组件主要瞄准观测量（如虚时格林函数与虚频电子自能函数）的解析延拓过程，最后的 toolbox 子组件主要用于前处理与后处理任务。从本节开始，我们将依次详细介绍上述四个子组件。

hibiscus-entropy 组件的作用是对虚时格林函数 $G(\tau)$ 进行解析延拓，获得电子谱函数 $A(\omega)$ 。hibiscus-entropy 组件的实现基于最大熵方法，这是解析延拓领域的经典算法，应用十分广泛⁶⁷。

hibiscus-entropy 组件的可执行程序为 entropy，所需要的主要输入文件为 entropy.in。entropy.in 文件包含了 hibiscus-entropy 组件的全部控制参数，典型的 entropy.in 文件如下所示：

```

1
2 >>> entropy.in: config parameters for maximum entropy method code
3
4 129      ! ntime : number of imaginary time slice
5 200      ! nwmax : number of frequency point on half axis
6 20       ! niter : number of cycles for maximum entropy method
7 16       ! ntune : number of smooth runs for maximum entropy method
8 3000     ! nstep : number of annealing steps per maximum entropy method cycle
9 1        ! nband : number of bands
10 2       ! norbs : number of orbitals
11 1       ! ntype : type of default model, if ntype = 0, gaussian type, if ntype = 1, flat type
12
13 1200.    ! ainit : initial alpha parameter
14 0.001    ! devia : it is the deviation from the green's function
15 8.000    ! beta  : inversion of real temperature
16 1.600    ! sigma : gauss broadening parameter
17 0.025    ! wstep : frequency step, used to build the frequency mesh
18 ~~~~~

```

各输入参数的简明解释如下，至于详细权威的解释请参阅 `hibiscus-entropy/entropy_control.f90` 文件中的注释。

- **ntime**

虚时间轴 $[0, \beta)$ 上所划分的时间片段的数目。

请注意，如果量子杂质求解器采用的是 **Daisy** 组件，那么此处的 `ntime` 应该等于 `solver.hfqmc.in` 文件中的 `ntime` 的值加 1。如果量子杂质求解器基于 CT-HYB 算法，那么此处的 `ntime` 就直接等于 `solver.ctqmc.in` 文件中的 `ntime`。关于量子杂质求解器组件的 `ntime` 参数的详细信息，请参阅第 6.38 节。

- **nwmax**

定义实频率点的数目。

在 `hibiscus-entropy` 组件中，默认实频数据点是围绕 0 点对称分布的。亦即在正半轴有 `nwmax` 个数据点，在负半轴也有 `nwmax` 个数据点，一共 $2 \times \text{nwmax} + 1$ 个数据点。数据点之间的间隔为 `wstep`，总共覆盖的频率区间为 $[-\text{nwmax} \times \text{wstep}, +\text{nwmax} \times \text{wstep}]$ 。

- **niter**

最大熵方法的最大迭代计算次数。

- **ntune**

谱函数 $A(\omega)$ 微调次数。

最大熵方法迭代计算完毕后，还需对获得的谱函数进行细微调整以及平滑处理，ntune 表示进行后期调整的次数。

- **nstep**

Monte Carlo sampling 次数。

在最大熵算法的每次迭代计算中，所需要进行的退火步数，可以理解为 Monte Carlo 抽样次数。

- **nband**

能带的数目。

此参数雷同于 solver.ctqmc.in 文件中的 nband 参数。关于量子杂质求解器组件的 nband 参数的详细信息，请参阅第6.8节。

- **norbs**

轨道的数目， $\text{norbs} = \text{nband} \times 2$ 。

此参数雷同于 solver.ctqmc.in 文件中的 norbs 参数。关于量子杂质求解器组件的 norbs 参数的详细信息，请参阅第6.10节。

- **ntype**

指定缺省模型 $m(\omega)$ 的类型。

ntype = 0 时，采用 Gauss 模型；

ntype = 1 时，采用常数模型。

- **ainit**

初始的 α 参数。

ainit 参数通常设得比较大，一般大于 1000。

- **devia**

$G(\tau)$ 的数值偏差程度。

该参数用来描述虚时格林函数数据的涨落，取 0.001 即可，通常无须改动。

- **beta**

反温度 $\beta = 1/T$ 。

此参数雷同于 solver.ctqmc.in 文件中的 beta 参数。关于量子杂质求解器组件的 beta 参数的详细信息，请参阅第6.26节。

- **sigma**

Gauss 模型中的展宽参数。

该参数仅当 $n_{\text{type}} = 0$ 时起作用。

- **wstep**

实轴频率数据点之间的间隔。

该参数也就是俗称的步长，由 `wstep` 参数与 `nwmax` 参数可以定义整个实轴网格。

除了 `entropy.in` 文件外，`hibiscus-entropy` 组件还需要的输入文件为 `tau.grn.dat`。该文件包含了虚时格林函数 $G(\tau)$ 的数据，可由 `hibiscus-toolbox` 组件的 `maketau` 程序产生 (请参阅第7.5节)。

`hibiscus-entropy` 组件的输出文件比较少，仅有 `mem.dos.dat` 与 `mem.sum.dat`，前者包含谱函数 $A(\omega)$ 的数据，后者包含了 $A(\omega)$ 的 `sum rules` 数据，普通用户不必关心。用户还可以利用 `hibiscus-toolbox` 组件中的 `makekra` 程序以及 `makeups` 程序对 `mem.dos.dat` 文件进行进一步的处理，计算出更多的物理量，详情请参阅第7.5节。

7.3 Hibiscus /hibiscus-stochastic 组件

`hibiscus-stochastic` 组件的作用是对虚时格林函数 $G(\tau)$ 进行解析延拓，获得电子谱函数 $A(\omega)$ 。也就是说 `hibiscus-stochastic` 组件的作用与 `hibiscus-entropy` 组件的作用是相同的，只不过前者基于随机解析延拓方法⁶⁸，而后者基于最大熵方法⁶⁷。从原理上说最大熵方法仅仅是随机解析延拓方法的一个特例，随机解析延拓方法更为先进，解析延拓的结果更为可靠。但是随机解析延拓方法的计算效率很低，需要长时间的计算才能得到比较平滑的计算结果，因此在日常的应用中我们建议用户还是选择 `hibiscus-entropy` 组件进行解析延拓，而采用 `hibiscus-stochastic` 组件的计算结果作为对照。

`hibiscus-stochastic` 组件的可执行文件为 `sai`，它所需要的主要输入文件为 `sai.in`。`sai.in` 包含了全部的控制参数，典型的 `sai.in` 文件示例如下：

```

1
2 >>> sai.in: config parameters for stochastic analytic continuation code
3
4 1024      ! ntime : number of imaginary time slice
5 400      ! nwmax : number of frequency point on half axis
6 10001    ! ngrid : number of slice of x in [0,1]
7 1024     ! ngamm : number of r_{\gamma} and a_{\gamma}
8 1        ! nalph : number of alpha parameter used in parallel tempering
9 2000     ! nwarm : maximum number of thermalization steps
10 200000   ! nstep : maximum number of quantum Monte Carlo sampling steps

```

```

11 2000      ! ndump : output period for stochastic analytic continuation code
12 2         ! ltype : measurement scheme
13 64        ! lemax : maximum number of legendre polynomial
14 20001     ! legrd : number of mesh points for legendre polynomial
15
16 1.000     ! ainit : initial alpha parameter
17 2.000     ! ratio : \alpha_(p+1) / \alpha_p = R
18 8.000     ! beta  : inversion of real temperature
19 0.005     ! eta1  : lorentz broadening parameter \eta_1
20 0.000     ! sigma : gauss broadening parameter
21 0.020     ! wstep : frequency step, used to build the real frequency mesh
22 ~~~~~

```

那么 sai.in 文件中各个输入参数的简明解释如下。至于详细权威的解释请参阅 hibiscus-stochastic/sai_control.f90 文件中的注释。

- **ntime**

虚时间轴 $[0, \beta)$ 上所划分的时间片段的数目。

虚时格林函数 $G(\tau)$ 定义在 $[0, \beta)$ 上，这一区间被划分为 ntime 个时间片，以便于处理。该参数须等于 solver.ctqmc.in 文件中设置的 ntime。关于量子杂质求解器组件的 ntime 参数的详细信息，请参阅第6.38节。

- **nwmax**

定义实频率点的数目。

在 hibiscus-stochastic 组件中，默认实频数据点是围绕 0 点对称分布的。亦即在正半轴有 nwmax 个数据点，在负半轴也有 nwmax 个数据点，一共 $2 \cdot \text{nwmax} + 1$ 个数据点。数据点之间的间隔为 wstep，总共覆盖的频率区间为 $[-\text{nwmax} \cdot \text{wstep}, +\text{nwmax} \cdot \text{wstep}]$ 。

- **ngrid**

用于定义稠密的 $[0, 1]$ 区间内的线性网格。

hibiscus-stochastic 组件内部使用一个平滑映射 ϕ 将谱函数 $A(\omega)$ 存在的频率区间 $[-\infty, +\infty]$ 映射至 $[0, 1]$ 区间:

$$\phi(\omega) = x \quad (7.1)$$

并且 $x \in [0, 1]$ 。再将 $[0, 1]$ 区间均分为 ngrid 个间隔。经过简单的数学变换后，谱函数 $A(\omega)$ 可以用无量纲场 $n(x)$ 表示出来。随机解析延拓方法实质上是使用 Monte Carlo 方法抽样求解 $\langle n(x) \rangle$ ，然后在程序的最后才将其转换为 $A(\omega)$ 。

- **ngamm**

δ 函数的数目。

在随机解析延拓算法中，使用大量 δ 函数的叠加来刻画无量纲场 $n(x)$:

$$n_C(x) = \sum_{\gamma} r_{\gamma} \delta(x - a_{\gamma}). \quad (7.2)$$

r_{γ} 为 δ 函数的权重，满足 $r_{\gamma} > 0$ ，而且 $\sum_{\gamma} r_{\gamma} = 1$ 。 a_{γ} 为 δ 函数的位置，满足 $0 \leq a_{\gamma} \leq 1$ 。 γ 为 δ 函数的索引，一般 2^{10} 个 δ 函数即可取得令人满意的结果。 $C = \{r_{\gamma}, a_{\gamma}\}$ 表示某时刻无量纲场的构型，一旦 C 确定下来，那么 $n(x)$ 也就确定了。我们使用经典蒙特卡洛方法来计算 $\langle n(x) \rangle$ ，应用 Metropolis 算法，随机选择 C 中任意一个 δ 函数，尝试修改其位置与权重： $a_{\gamma} \mapsto a'_{\gamma}$ 以及 $r_{\gamma} \mapsto r'_{\gamma}$ ，场构型也会发生相应的变化： $C \mapsto C'$ 。**ngamm** 参数即为 δ 函数的数目，从原则上说，**ngamm** 越大，那么计算精度越高，但是计算效率越低。

- **nalph**

α 参数的数目。

如果 **nalph** 大于 1，那么 parallel tempering 算法自动启用。

- **nwarm**

Monte Carlo 抽样算法的热平衡步数。

此参数雷同于 solver.ctqmc.in 文件中的 **ntherm** 参数。关于量子杂质求解器组件的 **ntherm** 参数的详细信息，请参阅第6.42节。

- **nstep**

Metropolis 算法的 Monte Carlo 抽样次数。

此参数雷同于 solver.ctqmc.in 文件中的 **nsweep** 参数。关于量子杂质求解器组件的 **nsweep** 参数的详细信息，请参阅第6.43节。

- **ndump**

Monte Carlo 抽样算法的输出间隔。

亦即每间隔 **ndump** 次抽样，hibiscus-stochastic 组件会输出一当前次的计算结果。此参数雷同于 solver.ctqmc.in 文件中的 **nwrite** 参数。关于量子杂质求解器组件的 **nwrite** 参数的详细信息，请参阅第6.44节。

- **ltype**

谱函数的测量方式。

ltype = 1 时，采用标准测量方式；

ltype = 2 时，那么采用 Legendre 正交多项式算法改善测量精度。

- **lemax**

Legendre 正交多项式的展开阶数。

此参数仅当 $ltype = 2$ 时有意义。此参数雷同于 `solver.ctqmc.in` 文件中的 `lemax` 参数。关于量子杂质求解器组件的 `lemax` 参数的详细信息，请参阅第6.29节。

- **legrd**

用于定义稠密的 $[-1, 1]$ 区间内的线性网格。

在 $[-1, 1]$ 间建立一个线性网格，Legendre 正交多项式即定义在此线性网格上，`legrd` 为网格点的数目。此参数雷同于 `solver.ctqmc.in` 文件中的 `legrd` 参数。关于量子杂质求解器组件的 `legrd` 参数的详细信息，请参阅第6.30节。

- **ainit**

第一个 α 参数的值。

- **ratio**

前后两个 α 参数的比值 R ，仅当 `nalph` 参数大于 1 时才有意义。

$$\frac{\alpha_{i+1}}{\alpha_i} = R \quad (7.3)$$

当 `hibiscus-stochastic` 组件启动时，即会根据 `nalph`、`ainit` 与 `ratio` 的值创建 α 参数的列表 α_i 。

- **beta**

反温度 $\beta = 1/T$ 。

此参数雷同于 `solver.ctqmc.in` 文件中的 `beta` 参数。关于量子杂质求解器组件的 `beta` 参数的详细信息，请参阅第6.26节。

- **etal**

η 参数，用于定义 $\delta(x)$ 函数：

$$\delta(x) = \frac{\eta}{x^2 + \eta^2} \quad (7.4)$$

- **sigma**

Gauss 展宽参数 σ ，用于设定缺省模型 $m(\omega)$ 。

如果 $\sigma > 0$ ，那么采用 Gauss 模型为缺省模型，

$$m(\omega) = \exp\left(-\frac{\omega^2}{2\sigma^2}\right) \quad (7.5)$$

如果 $\sigma \leq 0$ ，那么采用常数模型为缺省模型：

$$m(\omega) = \text{constants} \quad (7.6)$$

- **wstep**

实轴频率数据点之间的间隔。

该参数也就是俗称的步长，由 `wstep` 参数与 `nwmax` 参数可以定义整个实轴网格。

除了 `sai.in` 文件外，`hibiscus-stochastic` 组件还需要的输入文件为 `tau.grn.dat`。该文件包含了虚时格林函数 $G(\tau)$ 的数据，可由 `hibiscus-toolbox` 组件的 `maketau` 程序产生 (请参阅第7.5节)。值得注意的是 `hibiscus-stochastic` 组件每次仅能处理单个轨道的数据。因此如果用户面临的是多带系统，那么需要将 `tau.grn.dat` 文件按轨道拆分为多个同名文件，然后用 `hibiscus-stochastic` 组件依次进行处理。这也是 `hibiscus-stochastic` 组件不如 `hibiscus-entropy` 组件的地方之一。

`hibiscus-stochastic` 组件运行时刻输出的文件比较多，具体如下所示。

- `sai.image.dat`，存储谱函数 $A_\alpha(\omega)$ 。每个 α 参数都对应着一个谱函数，因此 `sai.image.dat` 文件中有很多个 block，每个 block 前有 α 参数的数值，然后是 $A_\alpha(\omega)$ ，block 与 block 之间间隔两个空行。
- `sai.imsum.dat`，存储谱函数 $A(\omega)$ ，它实际上是 $A_\alpha(\omega)$ 的算术平均值。这就是我们所要求的谱函数。用户可将 `sai.imsum.dat` 文件重命名为 `mem.dos.dat`，然后使用 `hibiscus-toolbox` 组件中的 `makekra` 和 `makeups` 程序对其进行后处理，详情可参阅第7.5节。
- `sai.move.dat`，存储 MOVE UPDATE 的 Monte Carlo 抽样统计信息，也就是该操作的接受概率。
- `sai.swap.dat`，存储 SWAP UPDATE 的 Monte Carlo 抽样统计信息，也就是该操作的接受概率。
- `sai.ppleg.dat`，存储 Legendre 正交多项式展开系数，仅当 `ltype = 2` 时才有意义，普通用户对此无须关心。

7.4 Hibiscus /hibiscus-swing 组件

`hibiscus-swing` 组件的作用是对虚频电子自能函数 $\Sigma(i\omega)$ 进行解析延拓，得到实频电子自能函数 $\Sigma(\omega)$ 。`hibiscus-swing` 组件来源于 K. Haule 开发的解析延拓程序⁷⁰，后经我们重新封装改进。它的主要原理是使用 Gaussian 多项式拟合 $\Sigma(i\omega)$ ，关于具体的细节，请参考原始文献。

`hibiscus-swing` 组件的主体是使用 Python 语言开发，部分计算模块采用 Fortran 90 语言编写。在编译安装 `hibiscus-swing` 组件之前，请确保系统中已经正确安装了 `f2py`。`hibiscus-swing` 组件所需要的 Python 运行环境为 2.6+，但是暂不支持 3.0+。如要正常执行 `hibiscus-swing` 组件，还需要 `scipy`、`numpy` 等 Python 数值库的支持。

在应用 `hibiscus-swing` 组件之前，请先预判系统是处于金属态还是绝缘态。`hibiscus-swing` 组件内含两个启动脚本，`metal.sh` 与 `insulator.sh`，分别对应上述两种情况。换言之，如果系统属于金属态，那么

执行脚本 metal.sh 以启动 hibiscus-swing 的主程序，反之则使用 insulator.sh 脚本。metal.sh 脚本的示例如下：

```
1 exec = ../../../../hibiscus-swing/swing_main.py
2 para='sig std.sgm.dat -nom 128 -beta 10.0 -wexp 1.15 -Ng 60 -FL True ...'
3 python $exec $para
```

insulator.sh 脚本的示例如下：

```
1 exec = ../../../../hibiscus-swing/swing_main.py
2 para='sig std.sgm.dat -nom 128 -beta 40.0 -wexp 1.05 -Ng 120 -FL False ...'
3 python $exec $para
```

metal.sh 与 insulator.sh 脚本的解释如下：

第 1 行是设定 hibiscus-swing 组件的主程序所在的位置。主程序名称为 swing_main.py。

第 2 行设定 hibiscus-swing 组件的启动参数，-sig 选项用来指定包含虚频电子自能函数的文件，-nom 选项用来指定一共考虑多少个松原频率点，-beta 选项用来指定反温度，-FL 选项用来指定 Fermi liquid 行为是否需保持。至于其余选项的具体含义，请参阅 swing_main.py 文件，此处不再赘述。

第 3 行的作用是启动 hibiscus-swing 组件的主程序。

hibiscus-swing 组件的输入文件通常是 std.sgm.dat，该文件由 hibiscus-toolbox 组件中的 makestd 程序产生，详情请参阅第 7.5 节。需要特别提醒的是：目前 hibiscus-swing 组件每次只能处理一个轨道的数据。对于多带系统而言，如果轨道之间是简并的，那么只运行一次 hibiscus-swing 组件即可。如果轨道之间不是简并的，那么需要对 std.sgm.dat 文件进行预先拆分，拆分后的文件仍然命名为 std.sgm.dat，只不过每个 std.sgm.dat 文件中包含的自能数据都互不相同而已。接下来需要用 hibiscus-swing 组件依次处理所有的 std.sgm.dat 文件。我们建议用户针对不同的轨道（亦即不同的 std.sgm.dat 文件）建立不同的文件夹，各自进行处理，以免产生混淆。

hibiscus-swing 组件的输出文件比较多，主要有如下几类：

- gaus.*，* 代表迭代计算的次数，下同。gaus.* 文件主要存放与 Gaussian 多项式展开系数有关的数据，普通用户对此不必关心。
- siom.*，此文件主要存放（利用当前的 Gaussian 多项式展开系数计算得到的）虚频电子自能函数 $\Sigma_{\text{fit}}(i\omega)$ 。用户可以比较原始自能数据 $\Sigma(i\omega)$ 与此拟合数据之间的差异。
- sres.*，此文件主要存放迭代计算过程中所获得的实频电子自能函数 $\Sigma_{\text{iter}}(\omega)$ 。普通用户不必关心。
- sigr_linear.out，此文件为最终的计算结果，存放 $\Sigma_{\text{fit}}(\omega)$ 。注意：sigr_linear.out 文件中仅仅包含一个轨道的电子自能数据。如果用户面对的是多带系统，那么除了拆分 std.sgm.dat 文件之外，还有一件麻烦事要做，那就是把若干 sigr_linear.out 文件合并起来。

hibiscus-swing 组件在运行过程中不需要用户去干预。通常需要 $O(10)$ 量级的迭代次数才能收敛，如果虚频电子自能函数的数据质量比较差（换言之，曲线不够平滑），那么程序将很难收敛。

除了 hibiscus-swing 组件以外，利用 hibiscus-toolbox 组件中的 makesig 程序也能对虚频电子自能函数 $\Sigma(i\omega)$ 进行解析延拓，获得 $\Sigma(\omega)$ 。只不过 makesig 程序利用了 Páde 展开方法，在技术路线上与 swing 稍有不同，详情请参阅第7.5节。

7.5 Hibiscus /hibiscus-toolbox 组件

hibiscus-toolbox 组件内包含了许多小工具程序，它们的主要作用是为量子杂质求解器组件提供前处理与后处理，下面将分门别类地介绍它们的功能与具体用法。

7.5.1 makechi

makechi 程序的作用是由自旋 - 自旋关联函数计算有效局域磁矩 M_e 。具体计算公式如下：

$$M_e = \sqrt{T\chi_{loc}}, \quad (7.7)$$

其中磁化系数 χ_{loc} 的定义式为：

$$\chi_{loc} = \int_0^\beta d\tau \chi_{loc}(\tau) = \int_0^\beta d\tau \langle S_z(0) S_z(\tau) \rangle, \quad (7.8)$$

其中 $\chi_{loc}(\tau) = \langle S_z(0) S_z(\tau) \rangle$ 就是自旋 - 自旋关联函数的表达式。

makechi 程序需要的输入文件为 solver.schi.dat，无输出文件。在执行 makechi 程序时，用户只需按照屏幕输出的提示一步一步操作即可。在程序结束时，屏幕上会输出 χ_{loc} 与 M_e 的数值。

使用 Gardenia 与 Narcissus 量子杂质求解器组件可以计算自旋 - 自旋关联函数，相应的控制参数为 isvrt = 2，详情请参阅第6.6节。

7.5.2 makedos

makedos 程序的作用是计算几种典型的晶格模型的电子态密度。目前最新版本的 makedos 程序支持以下几种情况：

- 无穷维立方晶格，Gaussian 型态密度，输出文件:dos.gauss.dat
- 简立方晶格，输出文件:dos.cubic.dat
- bethe 晶格，半圆态密度:dos.bethe.dat
- Lorentzian 型态密度，输出文件:dos.loren.dat

`makedos` 程序无须任何输入文件，对于不同的态密度，输出文件的名称也不尽相同，具体如上所示。在执行 `makedos` 程序时，用户只需按照屏幕输出的提示一步一步操作即可。用户可将 `dos*.dat` 文件重命名为 `solver.anydos.in` 文件，然后提供给量子杂质求解器组件作为输入。关于 `solver.anydos.in` 文件的详情，用户可以参阅第4.4节。

7.5.3 makekra

`makekra` 程序的作用是由电子谱函数 $A(\omega)$ 计算出实频杂质格林函数的虚部 $\Im G(\omega)$ ，

$$A(\omega) = -\frac{1}{\pi} \Im G(\omega). \quad (7.9)$$

再利用 Kramers-Krönig 变换关系：

$$\Re G(\omega) = -\frac{1}{\pi} \int_{-\infty}^{\infty} d\omega' \frac{\Im G(\omega')}{\omega - \omega'}, \quad (7.10)$$

进一步求出实部 $\Re G(\omega)$ 。

目前 `makekra` 程序与 `hibiscus-entropy` 程序相互接口，它需要的输入文件为 `mem.dos.dat`，输出文件为 `kra.grn.dat`。在执行 `makekra` 程序时，用户只需按照屏幕输出的提示一步一步操作即可。

7.5.4 makescr

`makescr` 程序的作用是由屏蔽谱函数 $W(\omega)$ 计算出双积分推迟作用函数 $\mathcal{K}(\tau)$ ，以及相互作用强度 U 与化学势 μ 的调整量。

$\mathcal{K}(\tau)$ 的定义如下²⁹：

$$\mathcal{K}(\tau) = \int_0^{\infty} \frac{d\omega'}{\pi} \frac{\Im W(\omega')}{\omega'^2} [\mathcal{B}(\omega', \tau) - \mathcal{B}(\omega', 0)]. \quad (7.11)$$

相互作用强度 U 的调整量由下式计算，调整量即为下式右端第二项：

$$U_{scr} = U + 2 \int_0^{\infty} \frac{d\omega'}{\pi} \frac{\Im W(\omega')}{\omega'}. \quad (7.12)$$

化学势 μ 的调整量由下式计算，调整量即为下式右端第二项：

$$\mu_{scr} = \mu + \int_0^{\infty} \frac{d\omega'}{\pi} \frac{\Im W(\omega')}{\omega'}, \quad (7.13)$$

在执行 `makescr` 程序时，用户只需按照屏幕输出的提示一步一步操作即可。`makescr` 程序所需要的输入文件为 `scr.frq.dat`¹，其中包含了屏蔽谱函数 $W(\omega)$ 的原始数据。输出文件为 `scr.tau.dat`，其中包含了 $\mathcal{K}(\tau)$ 的数据。`makescr` 程序的实际作用是为 *Narcissus* 组件准备输入文件。在通常情况下，我们需要将 `scr.tau.dat` 重命名为 `solver.ktau.in` (请参阅第4.5节)，以便与 *Narcissus* 组件配合使用。相关的控制参数为 `isvrt = 99`，详情请参阅第6.6节。

¹`scr.frq.dat` 文件中的数据通常由文献中获取。

7.5.5 makesig

`makesig` 程序的作用是对量子杂质求解器组件输出的虚频电子自能函数 $\Sigma(i\omega)$ 数据进行解析延拓，得到实频上的电子自能函数 $\Sigma(\omega)$ 。此程序十分简单，所采用的算法为 Páde 算法。如果原始电子自能函数的数据质量不佳，那么此程序给出的结果不一定是合理的，用户需要自行甄别。

`makesig` 程序需要的输入文件为 `solver.sgm.dat`，输出文件为 `sig.sgm.dat`。在执行 `makesig` 程序时，用户只需按照屏幕输出的提示一步一步操作即可。

`hibiscus-swing` 组件的功能也是对虚频电子自能函数进行解析延拓，这两个程序的计算结果可用于相互比较，详情请参阅第7.4节。

7.5.6 makestd

`makestd` 程序的作用是对量子杂质求解器组件输出的虚频电子自能函数 $\Sigma(i\omega)$ 数据进行预处理，为自能解析延拓程序（亦即 `hibiscus-swing` 组件，详情请参阅第7.4节）提供合适的输入文件。

`makestd` 程序需要的输入文件为 `solver.sgm.dat.*`，* 表示自 1 起的连续数字，没有上限，和 `solver.green.bin.*` 的情况类似²。`makestd` 程序的输出文件为 `std.sgm.dat`。在执行 `makestd` 程序时，用户只需按照屏幕输出的提示一步一步操作即可。

7.5.7 maketau

`maketau` 程序的作用是对量子杂质求解器组件输出的虚时格林函数 $G(\tau)$ 数据进行预处理，为解析延拓程序（`hibiscus-entropy` 组件与 `hibiscus-stochastic` 组件，请参阅第7.2节与第7.3节）提供合适的输入文件。`maketau` 程序支持以下四种情况。

- CT-QMC 组件（除 `Daisy` 组件外的其余量子杂质求解器组件，下同）输出的 `solver.green.dat` 文件
- HF-QMC 组件（仅 `Daisy` 组件，下同）输出的 `solver.green.dat` 文件
- CT-QMC 组件输出的 `solver.green.bin.*` 文件
- HF-QMC 组件输出的 `solver.green.bin.*` 文件

其中前两种情况的处理结果适合 `hibiscus-entropy` 组件，而后两种情况的处理结果适合 `hibiscus-stochastic` 组件。`maketau` 程序所需的输入文件为 `solver.green.dat` 或者是 `solver.green.bin.*`，输出文件为 `tau.grn.dat`。在执行 `maketau` 程序时，用户只需按照屏幕输出的提示一步一步操作即可。

²很遗憾，目前 `iQIST` 并没有提供一个类似于 data binning 的机制，自动生成 `solver.sgm.dat.*`。因此，为了得到一系列的 `solver.sgm.dat.*` 文件，用户必须重复多次以单步模式执行量子杂质求解器组件，并手动重命名所获得的 `solver.sgm.dat` 文件。

7.5.8 makeups

makeups 程序的作用是由谱函数 $A(\omega)$ 计算紫外光电子能谱以及 X 射线吸收谱。

紫外光电子能谱的计算: 杂质谱函数 $A(\omega)$ 乘以 Fermi-Dirac 分布函数 $f(\omega)$ 即可获得理论紫外光电子能谱, 如下式所示:

$$A_{\text{PES}}(\omega) = A(\omega)f_T(\omega) = \frac{A(\omega)}{1 + e^{\frac{\omega - \mu}{k_B T}}}. \quad (7.14)$$

由于实验仪器的分辨率是有限的, 为了能够与实验数据相比较, 须得考虑对理论曲线进行 Gaussian 展宽, 具体计算公式如下所示:

$$\tilde{A}_{\text{PES}}(\omega) = \int d\omega' \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(\omega - \omega')^2}{2\sigma^2}} A_{\text{PES}}(\omega'), \quad (7.15)$$

其中 σ 为展宽因子, 对于高分辨率的紫外光电子能谱实验, σ 可以取为 0.1, 至于其它情况 σ 的取值范围在 0.2 ~ 0.36 之间, 大致上与实验分辨率保持一致。

X 射线吸收谱的计算: 杂质谱函数 $A(\omega)$ 乘以反 Fermi-Dirac 分布函数 $f(-\omega)$ 即可获得理论 X 射线吸收谱, 如下式所示:

$$A_{\text{XAS}}(\omega) = \frac{A(\omega)}{1 + e^{\frac{\mu - \omega}{k_B T}}}. \quad (7.16)$$

为了能够与实验谱线相比较, 同样须得对上述理论曲线进行 Gaussian 展宽操作, 具体计算公式与公式 (7.15) 完全一致, 展宽参数 σ 的取值范围在 0.2 ~ 0.4 之间。

makeups 程序的输入文件为 mem.dos.dat, 该文件由 hibiscus-entropy 程序产生, 内含谱函数 $A(\omega)$ 的数据, 详情请参阅第 7.2 节。输出文件为 ups.pes.dat 与 ups.xas.dat, 分别包含紫外光电子能谱数据 $A_{\text{PES}}(\omega)$ 以及 X 射线吸收谱数据 $A_{\text{XAS}}(\omega)$ 。在执行 makeup 程序时, 用户只需按照屏幕输出的提示一步一步操作即可。

7.6 pysci.py 程序

iQIST 的量子杂质求解器组件对于输入文件的格式有十分严格的要求, 一个额外的空行或者是字符都有可能造成严重的错误 (请参阅第 4.1 节)。为了解决这个问题, 在 *iQIST* 软件包中提供了一个名为 pysci.py 的 Python 脚本, 利用此脚本, 我们可以快速地产生标准的 solver.ctqmc.in 输入文件。

如要运行 pysci.py 脚本, 系统中首先需要安装 Python 3.0+。

第一步:

在执行 pysci.py 之后, 脚本会提示用户选择量子杂质求解器组件。目前 pysci.py 脚本仅仅支持 *Azalea*、*Gardenia* 和 *Narcissus* 组件。用户只需要输入量子杂质求解器组件名称的首字母 (大小写均可), 并按下回车键, 即可选定一种组件。如果用户不提供任何输入, 只是直接按下回车键, 那么脚本默认用户选择了 *Azalea* 组件。

第二步:

根据用户选定的组件名称，脚本程序会自动产生一系列的参数。对于每个参数，脚本程序都会显示参数的内部名称、数据类型、缺省值以及详细具体的解释。用户需要输入合适的数值，并按下回车键。然后脚本程序会提示下一个参数。这样直至所有必须的参数都被设定完毕。如果用户不输入任何数值，只是简单的按下回车键，那么脚本程序将把缺省值赋给当前的参数。此外，脚本程序对于用户所输入数据的合理性以及有效性不做任何检测。我们将决定权都留给用户。

第三步:

根据用户的输入，脚本程序将产生标准的 `solver.ctqmc.in` 文件，输出到当前目录下。`pysci.py` 程序只保证 `solver.ctqmc.in` 文件格式的准确性，而不保证文件内容的合理性。用户需要根据自身的需求，仔细检查 `solver.ctqmc.in` 文件中的设定，并做出相应的调整。

在后续版本的 `pysci.py` 程序中，我们将添加更多的组件支持，并引入用户输入数据检测功能。

第八章

iQIST 典型应用场景

此前的各章已经详述了 iQIST 软件包的方方面面，但是理论终须联系实际。本章将通过若干个典型的案例引导用户逐步掌握 iQIST 软件包的基本用法。在此之前，我们首先假定用户已经成功地在自己的系统上安装了 iQIST 软件包，安装位置为 `/opt/iqist1`，并且已经将 `/opt/iqist/bin` 目录加入到环境变量 `PATH` 中。

```
$ export PATH=/opt/iqist/bin:$PATH
```

如果用户是初次接触 iQIST，那么请按照下面的教程一步一步进行操作，当下述例子都完成后，相信你已经成为一位 iQIST 专家了。如果用户对 iQIST 已经有了一定的了解，那么就不必从头开始，可以挑选自己感兴趣的案例直接进行学习。

8.1 基本应用

8.1.1 初识 iQIST

我们首先使用 iQIST 软件包内含的 Azalea 组件来自洽求解 bethe 晶格上的单带半满 Hubbard 模型。由于单带 Hubbard 模型具有密度 – 密度相互作用形式，因此可以采用基于段表示算法的 Azalea 组件来进行计算。下面开始具体的计算流程。

步骤 1: 创建自己的工作目录/工作文件夹

在用户自己的目录下创建专用的工作文件夹，`iqist_test` 应该是一个不错的选择。

```
$ mkdir iqist_test
```

进入到 `iqist_test` 文件夹中:

¹具体的安装步骤请参阅第2章。

```
$ cd iqist_test
```

创建新的文件夹 t811²:

```
$ mkdir t811
```

再进入到 t811 文件夹中:

```
$ cd t811
```

那么当前所在的位置就是 /iqist_test/t811。

步骤 2:准备 solver.ctqmc.in

在 /opt/iqist/tutor/t811 文件夹中, 我们已经准备好了 *Azalea* 组件的输入文件 solver.ctqmc.in, 请将它复制到当前目录下。

```
$ cp /opt/iqist/tutor/t811/solver.ctqmc.in .
```

关于 solver.ctqmc.in 文件的具体格式, 在第4.1节中已经有十分详细的介绍, 并且各个控制参数的具体含义, 在第6章中也有权威的描述, 此处不再赘述。在此 solver.ctqmc.in 文件中, 一些关键的控制参数设置如下:

- 采用自洽计算模式, $isscf = 2$
- 强制进行轨道对称化处理, $issun = 2$
- 强制进行自旋对称化处理, $isspn = 1$
- 启用 data binning 模式, $isbin = 2$
- 该模型只有 1 条能带, $nband = 1$
- 自旋投影个数, $nspin = 2$
- 轨道数目, $norbs = 2$
- 原子组态数目, $ncfgs = 4$
- 最大自洽迭代次数, $niter = 20$
- 库仑相互作用强度, $U = U_c = U_v = 2.0$
- Hund 交换作用强度, $J_z = J_s = J_p = 0.0$
- 化学势, $\mu = 1.0$
- 反温度, $\beta = 40.0$
- 最近邻跃迁系数, $part = 0.50$

²在缺省状态下, 我们采用章节的名称作为工作文件夹的名字。

- 最大图形展开阶数, $mkink = 1024$
- 虚时点数目, $ntime = 1024$
- 自旋翻转操作的周期, $nflip = 2000$
- 蒙特卡洛抽样总次数, $nsweep = 20,000,000$
- 程序输出周期, $nwrite = 2,000,000$
- clean 操作的间隔周期, $nclean = 100,000$
- 物理量测量的间隔周期, $nmonte = ncarlo = 100$

步骤 3: 启动 *Azalea* 组件进行实际计算

在此案例中, 我们使用 8 个计算核心运行此 *ctqmc* 程序, 用户可以根据现有的计算资源做出适当调整, 当然, 用户也可以用 1 个计算核心以串行模式运行此 *ctqmc* 程序³。

使用 8 个计算核心的命令示例:

```
$ nohup mpiexec -n 8 ctqmc < /dev/null > out.dat &
```

使用 1 个计算核心的命令示例:

```
$ nohup ctqmc < /dev/null > out.dat &
```

对于蒙特卡洛抽样次数, 我们想特别强调的一点就是: 在输入文件 *solver.ctqmc.in* 中设定的抽样次数 *nsweep* (请参阅第 6.43 节) 是每个计算核心上执行的抽样次数, 总的抽样次数是每个计算核心的抽样次数乘以计算核心的数目。因此, 在总的抽样次数保持不变的情况下, 用户可以通过增加每个计算核心的抽样次数来减少所使用的计算核心数目, 或者是通过增加计算核心的数目来减少每个计算核心上的抽样次数, 以提高计算效率。但是请注意, 为了保证蒙特卡洛抽样算法的稳定可靠, 请保证每个计算核心上的抽样次数不低于 20,000,000 次。在本案例中, 每次迭代计算共需进行 $8 \times 20,000,000$ 次蒙特卡洛抽样, 如果迭代计算的次数为 20 次, 再加上 data binning 模式所需要的抽样次数⁴, 那么总共需要进行 $30 \times 8 \times 20,000,000$ 次蒙特卡洛抽样。这是一个十分巨大的数字, 幸好 *Azalea* 是一个非常强大的量子杂质模型求解器组件, 它的计算效率非常非常的高, 马上你就能感受到它的威力。

步骤 4: 监测 *Azalea* 组件的运行状态

在 *ctqmc* 程序运行期间, 用户须不时观察程序的运行状态。请用户注意监测 *out.dat* 文件 (请参阅第 5.1 节), 该文件记录了 *ctqmc* 程序的全部终端输出。用户可以从该文件中了解到当前 DMFT 自治计算的迭代次数 (以 "iter:" 标记); 蒙特卡洛抽样进行的次数 (以 "sweep:" 标记); 一些物理量的即时统计平均值; 蒙特卡洛抽样的统计信息⁵等等。另外, 用户还可以知道当前程序运行的时间。

³此处为了简便起见, 我们假定 *Azalea* 组件的可执行程序就是 *ctqmc*。在实际应用中, 为了防止程序名的冲突, 用户可能会将它重命名为其它的名字, 存放在 */opt/iqist/bin* 目录中。

⁴相当于进行十次迭代计算。

⁵例如插入、删除、左移、右移等更新操作被接受/拒绝的次数和概率。

ctqmc 程序每间隔 `nwrite` 次蒙特卡洛抽样就会输出一中间结果，同时每次 DMFT 迭代计算结束，量子杂质求解器被 shutdown 后，相应的计算结果全部会被输出到文件中（请参阅第5.2节）。用户可以随时检测这些中间的输出结果。例如：

- 使用 `vim` 打开 `solver.nmat.dat` 文件来查看占据数是否已经收敛到了我们需要的值。

```
$ vim solver.nmat.dat
```

- 使用 `gnuplot` 观察虚时格林函数 $G(\tau)$ 。

```
$ gnuplot> plot "solver.green.dat" u 2:4 w lp
```

- 使用 `gnuplot` 观察虚频格林函数 $G(i\omega)$ 的虚部。

```
$ gnuplot> plot [0:50] "solver.grn.dat" u 2:4 w lp
```

- 使用 `gnuplot` 观察电子自能函数 $\Sigma(i\omega)$ 的虚部。

```
$ gnuplot> plot [0:50] "solver.sgm.dat" u 2:4 w lp
```

在现代主流计算机上完成本计算案例大约需要 25~30 分钟。当 `ctqmc` 程序运行结束后，在 `out.dat` 文件末尾，用户应该看到如下类似的输出：

```
1  AZALEA >>> I am tired and want to go to bed. Bye!
2  AZALEA >>> happy ending at 10:42:21 Oct 18 2012
```

这表示用户的 `ctqmc` 程序成功运行完毕。恭喜你！你已经会求解强关联电子问题了。

步骤 5:分析计算结果

A: 分析 `out.dat` 文件

`out.dat` 文件中包含着很多有用的信息。采用 `grep` 命令可以从 `out.dat` 文件中提取出 E_{tot} 、 E_{kin} 、 E_{pot} 与 $\langle S_z \rangle$ 等物理量。用户还须快速浏览 `out.dat` 文件以查看 `ctqmc` 程序运行中是否出现异常，主要是蒙特卡洛抽样的统计信息是否合理，如果出现很诡异的数值，请检查 `solver.ctqmc.in` 输入文件并重新进行计算。对于 DMFT 自洽计算模式 (`isscf = 2`，请参阅第6.1节)，用户需要关注电子自能函数是否已经收敛，请使用如下命令检查电子自能函数的收敛情况：

```
$ grep sig_curr out.dat
```

用户应该得到类似如下的输出：

```
1  .....
2  AZALEA >>> sig_curr:0.2768E-07  eps_curr:0.1000E-07
3  AZALEA >>> sig_curr:0.1962E-07  eps_curr:0.1000E-07
4  AZALEA >>> sig_curr:0.1676E-07  eps_curr:0.1000E-07
5  AZALEA >>> sig_curr:0.1697E-07  eps_curr:0.1000E-07
```

```

6  AZALEA >>> sig_curr:0.1044E-07  eps_curr:0.1000E-07
7  AZALEA >>> sig_curr:0.2538E-07  eps_curr:0.1000E-07

```

其中 sig_curr 是当前电子自能函数的收敛精度， eps_curr 是程序预设要达到的精度⁶，在本次计算中，电子自能函数的收敛程度已经是很不错了。如果用户仔细地分析 sig_curr 的变化情况就会发现：在迭代计算开始时， sig_curr 的值比较大，随着迭代次数的增加， sig_curr 振荡减少，当 sig_curr 减小到达某个数量级时，就会停止下降，在此数量级附近涨落。这种涨落现象是 Monte Carlo 算法的内禀性质，简单地增加 DMFT 自洽迭代计算步数 (niter 参数，请参阅第6.16节) 或者是 Monte Carlo 抽样次数 (nsweep 参数，请参阅第6.43节) 并不能完全消除这种现象。根据经验，当 sig_curr 与 eps_curr 处于同一数量级时 (至少连续两次)，我们就可以认为 DMFT 自洽迭代计算已经收敛了。

B: 分析 solver.nmat.dat 文件

`solver.nmat.dat` 文件中储存着电子占据数的相关信息。在此案例中，因为考虑的是单带半满的 Hubbard 模型，因此每个轨道的占据数必须为 0.5，总的占据数为 1.0。小的数值误差是允许的，如果用户的计算结果偏离这个理论值，请查看 `solver.ctqmc.in` 输入文件中化学势 μ 是否被设为 $U/2$ 。

电子双占据数可以用来判断系统处在什么相中，这也是一个十分有用的物理量。在此计算中用户应该可以找到如下类似的输出，

```

1  < n_i n_j > data:
2      1      1      0.000000
3      1      2      0.078905
4      2      1      0.078905
5      2      2      0.000000

```

其中第 3, 4 行表示双占据数为 0.078905，这意味着系统还有有限的双占据概率，系统可能正处于金属相。

C: 分析 $G(\tau)$ 、 $G(i\omega)$ 、 $\Sigma(i\omega)$ 数据

$G(\tau)$ 、 $G(i\omega)$ 和 $\Sigma(i\omega)$ 是量子杂质求解器组件最重要的输出结果之一。我们可以用 `gnuplot` 快速绘制出虚时格林函数 $G(\tau)$ 、虚频格林函数的虚部 $\Im G(i\omega)$ 与虚频电子自能函数的虚部 $\Im \Sigma(i\omega)$ ，对计算结果做出初步的分析。用户也应该能得到类似的结果。

图8.1中给出的是虚时格林函数 $G(\tau)$ ，用户首先要检查该函数曲线是否足够光滑，尤其是底部靠近 0 的位置是否光滑。有一个经验是，如果底部非常贴近 0⁷，那么该系统是绝缘相。图8.1清楚地表明系统属于典型的金属相。除此之外，还需确认所有的 $G(\tau)$ 数据点均小于 0，如果大于 0 则是不合理的。

图8.2中给出的是虚频格林函数的虚部 $\Im G(i\omega)$ ，仅仅画出了低频部分的数据点。用户也需要检查该函数曲线是否足够光滑，尤其是随着频率增大 $\omega \rightarrow \infty$ ，曲线应该有趋近于 0 的渐进行为。 $\Im G(i\omega)$ 的低频

⁶在程序内部指定，不属于用户可调参数。

⁷所谓的底部指的是 $\tau = \beta/2$ 附近两侧的地方。

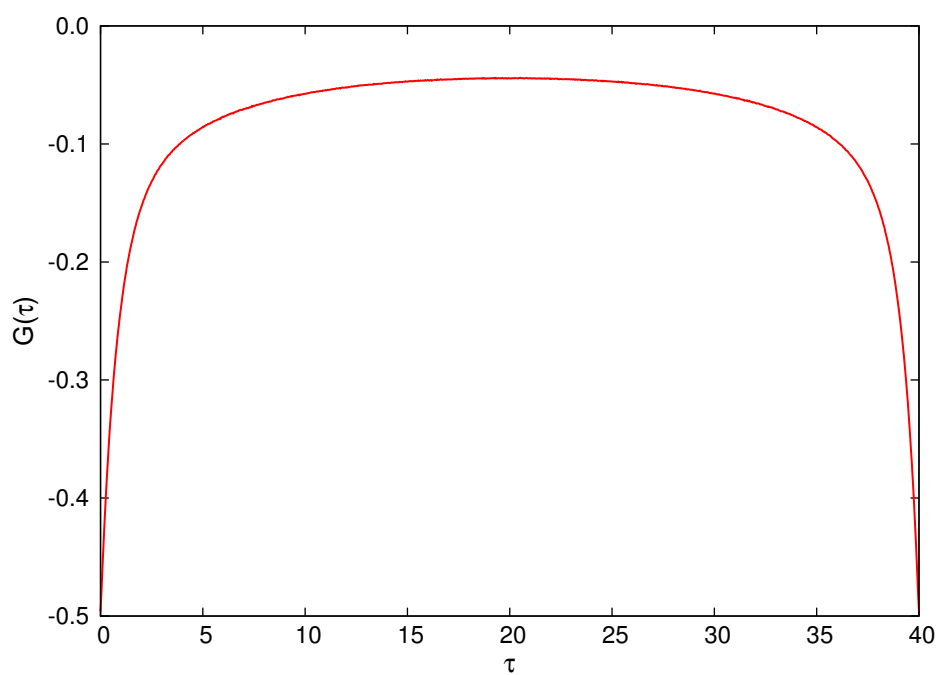


图 8.1: 单带半满 Hubbard 模型的虚时格林函数 $G(\tau)$

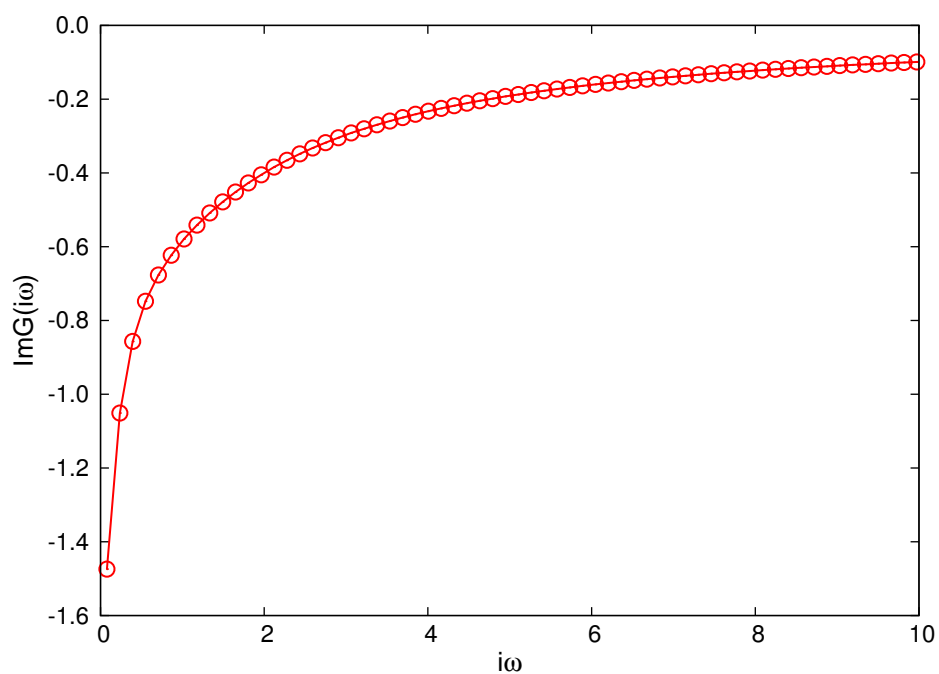


图 8.2: 单带半满 Hubbard 模型的虚频格林函数的虚部 $\Im G(i\omega)$

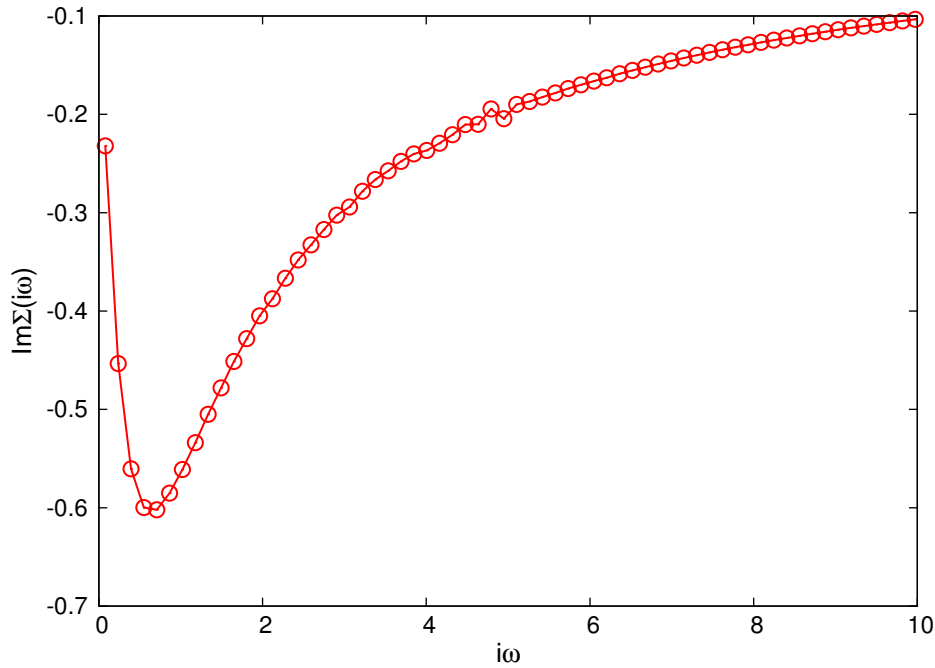


图 8.3: 单带半满 Hubbard 模型的电子自能函数的虚部 $\Im\Sigma(i\omega)$

部分的弯曲情况可以用来粗略判断系统处在什么相。如果是一个勾 ($\sqrt{\quad}$) 的形状, 那么该系统应该处于绝缘相, 反之, 则属于金属相。图8.2明显展示的是金属相的特征, 与上文的推测相符。

图8.3中给出的是虚频电子自能函数的虚部 $\Im\Sigma(i\omega)$, 仅仅画出了低频部分的数据点。用户也需要检查该函数曲线是否足够光滑, 尤其是随着频率增大 $\omega \rightarrow \infty$, 曲线应该有趋近于 0 的渐进行为。

D: 分析 P_H 、 P_T 数据

图形微扰展开序列的阶数分布 P_H 以及原子组态的概率分布 P_T 能提供十分重要的信息。

使用如下命令可以绘制出 P_H 数据:

```
$ gnuplot> plot [0:20] "solver.hist.dat" u 1:3 w lp
```

请注意, 0 阶的数据是位于文件最后的一行, 请在绘图之前将最后一行剪切粘贴到 1 阶那一行之前, 并把阶数由 1024 改为 0(参见第5.2.15小节)。图8.4给出的是本次计算所获得的图型微扰展开阶数的分布情况。不难发现, 图中曲线是以 5 为中心的类 Gaussian 分布。由此, 我们可以得到一个直观的认识: 那就是在应用连续时间量子蒙特卡罗算法进行抽样时, 高阶项和低阶项的出现概率是非常低的, 因此计算结果不会出现发散。

另外用户也可以用 vim 查看 solver.prob.dat 文件, 了解原子组态概率的分布情况 P_T 。在本例中, 应该是单占据态 ($N = 1$) 的出现概率占据了绝大部分, 空占据态 ($N = 0$) 以及双占据态 ($N = 2$) 的出现概率比较小。

OK! 到目前为止你已经完成了第一个例子: 使用 *iQIST* 软件包的 *Azalea* 量子杂质求解器组件在动力

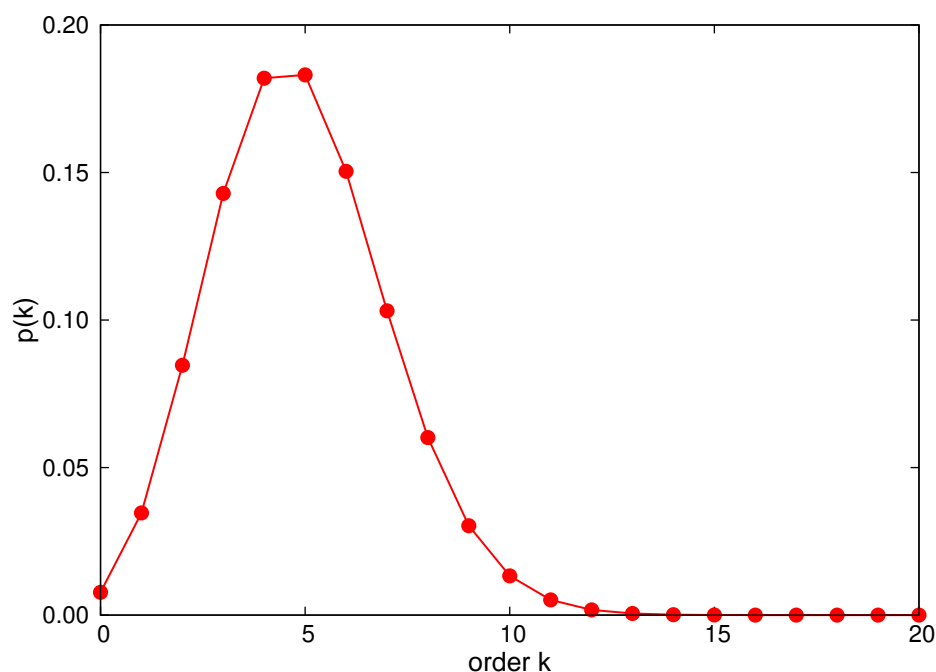


图 8.4: 图形微扰展开序列的阶数分布 P_H

学平均场理论的框架内求解了一个单带半满 Hubbard 模型。这个例子十分简单，但是却十分的重要，由此学到的方法和技巧将会帮助你挑战更困难的问题。

8.1.2 Mott 金属 – 绝缘体相变

在本案例中，主要展示如何应用 iQIST 软件包来研究 Mott 金属 – 绝缘体相变。研究对象还是最简单的 bethe 晶格上的单带半满 Hubbard 模型，此模型已经包含了 Mott 金属 – 绝缘体相变现象的绝大部分物理在内。众所周知，Mott 相变的主要驱动力是 Coulomb 相互作用 U ，因此我们只需在上一个案例的基础上，调整 solver.ctqmc.in 输入文件中 U 参数的值，再重复进行几次 DMFT 自治计算即可。事不宜迟，下面让我们按部就班开始计算吧。

步骤 1: 创建自己的工作目录/工作文件夹

首先创建 iqist_test 文件夹，该文件夹可作为用户的练习目录。如果用户在练习上一个案例时已经创建了此文件夹，那么可以略过此操作。

在 iqist_test 文件夹下创建子文件夹 t812:

```
$ mkdir t812
```

再进入到 t812 文件夹中:

```
$ cd t812
```

那么当前所在的位置就是 /iqist_test/t812。

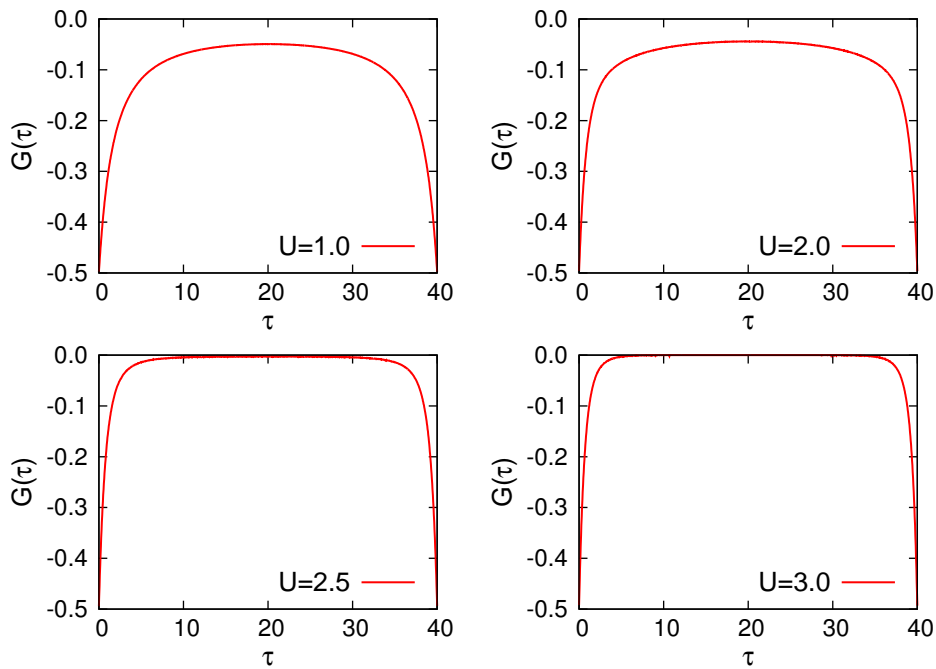


图 8.5: 虚时格林函数 $G(\tau)$ 随着 U 的变化

我们选择 9 组不同的 U 值进行计算，分别为:1.0、1.5、2.0、2.5、3.0、3.5、4.0、5.0、6.0 eV。请为不同的 U 值建立相应的文件夹，分别为 u10、u15、u20、u25、u30、u35、u40、u50、u60。创建文件夹的命令示例如下:

```
$ mkdir u10 u15 u20 u25 u30 u35 u40 u50 u60
```

步骤 2:准备 solver.ctqmc.in

在/opt/iqist/tutor/t812 文件夹中，我们已经准备好了 **Azalea** 组件的输入文件 solver.ctqmc.in，请将它复制到当前目录下对应的文件夹中，例如:

```
$ cp /opt/iqist/tutor/t812/u10/solver.ctqmc.in ./u10
```

u10/solver.ctqmc.in 文件与 u20/solver.ctqmc.in 文件基本上完全一致，差别仅在于 U 、 U_c 、 U_v 与 mune 参数的设置。用户可以使用 diff/colordiff 命令比较这两个文件的异同。

```
$ diff u10/solver.ctqmc.in u20/solver.ctqmc.in
```

用户可进一步两两比较其它文件夹中的 solver.ctqmc.in 文件，亦能得到类似的发现。

步骤 3:启动 **Azalea** 组件进行实际计算

依次进入到各个 u^* 文件夹中，按照上一小节介绍的方法，启动 **Azalea** 组件的 ctqmc 程序进行计算。完成全部计算过程估计需要一段时间，用户可以利用这段时间来阅读该文档的其它部分内容。

步骤 4:数据的分析和处理

当所有 U 值的 DMFT 计算都进行完毕之后，我们需要对计算结果进行细致的分析，以判明是否发生

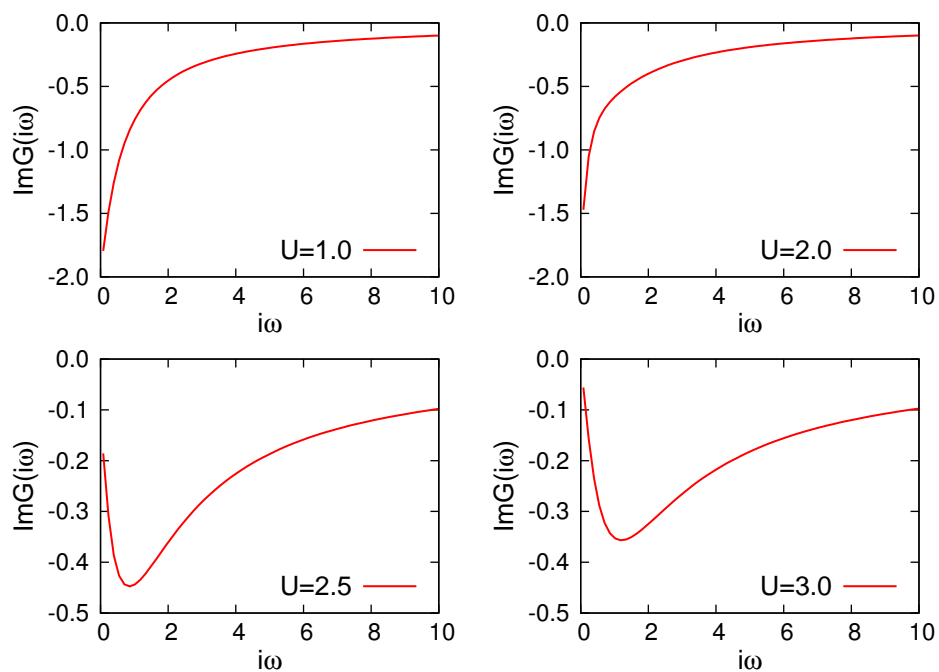


图 8.6: 虚频格林函数的虚部 $\Im G(i\omega)$ 随着 U 的变化

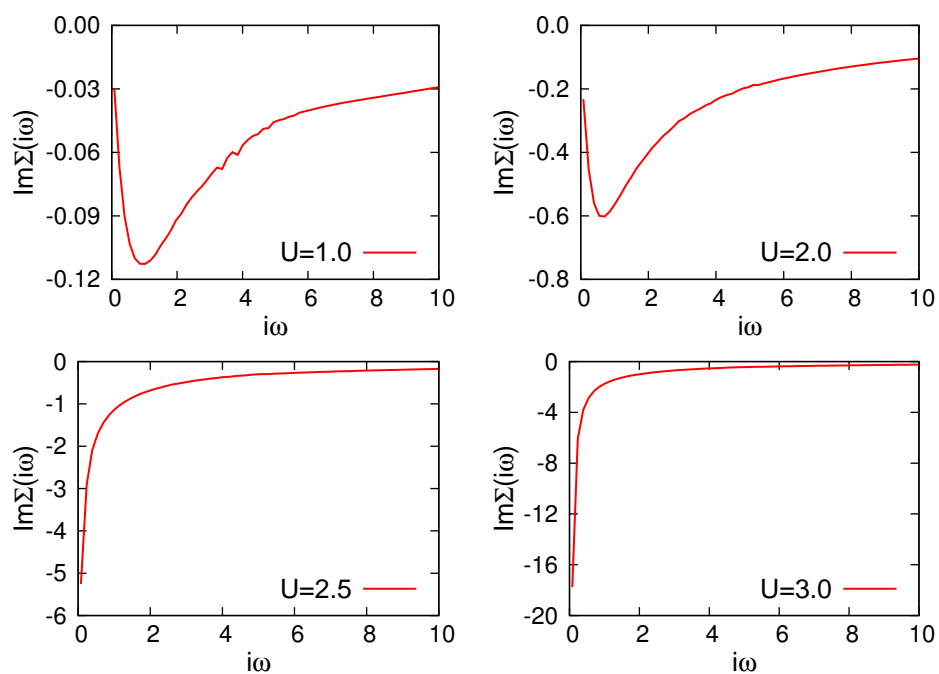


图 8.7: 电子自能函数的虚部 $\Im \Sigma(i\omega)$ 随着 U 的变化

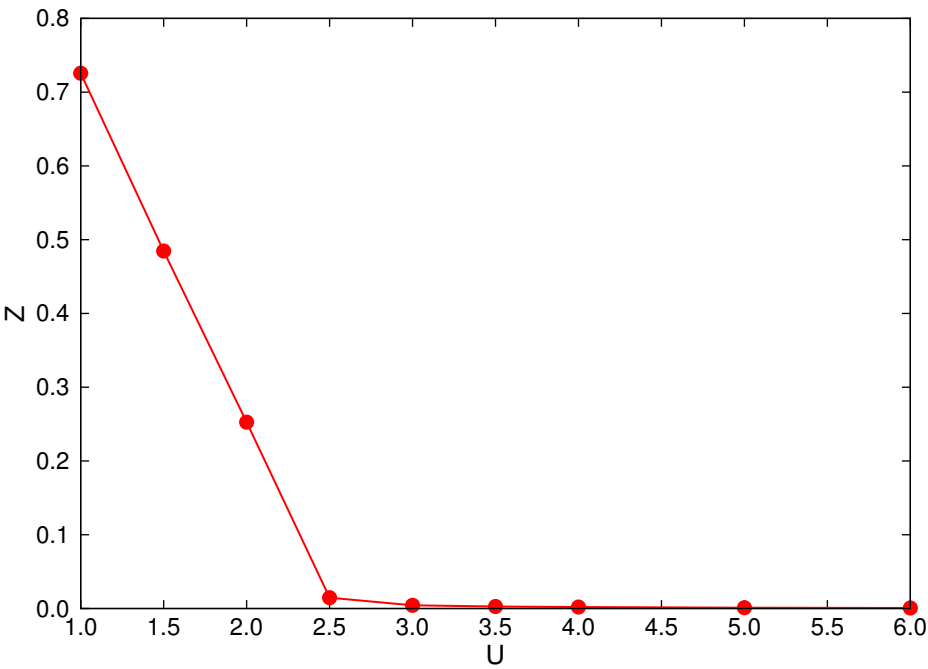


图 8.8: 准粒子权重 Z 随着 U 的变化

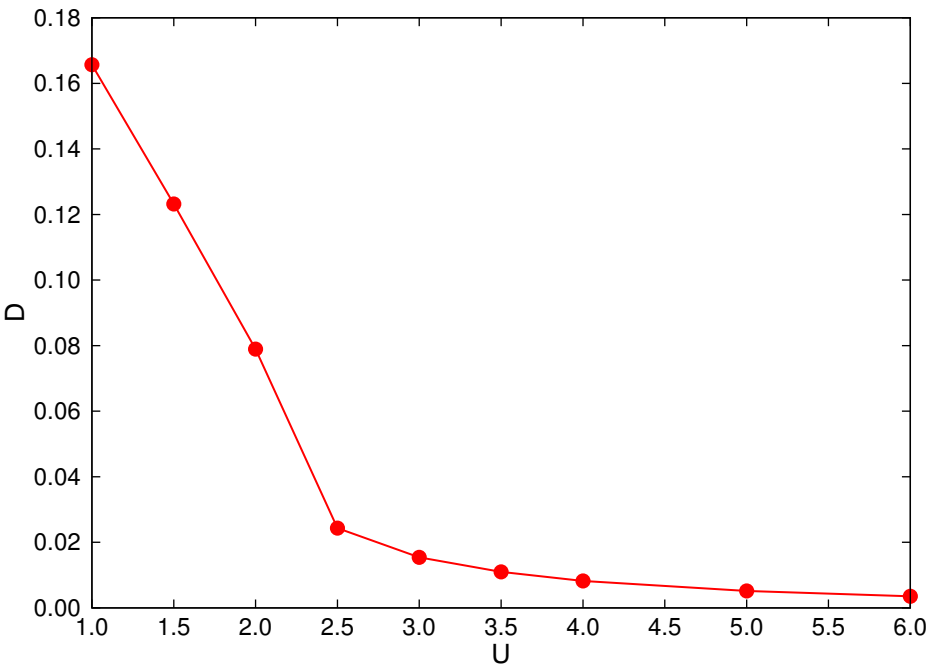


图 8.9: 双占据数 D 随着 U 的变化

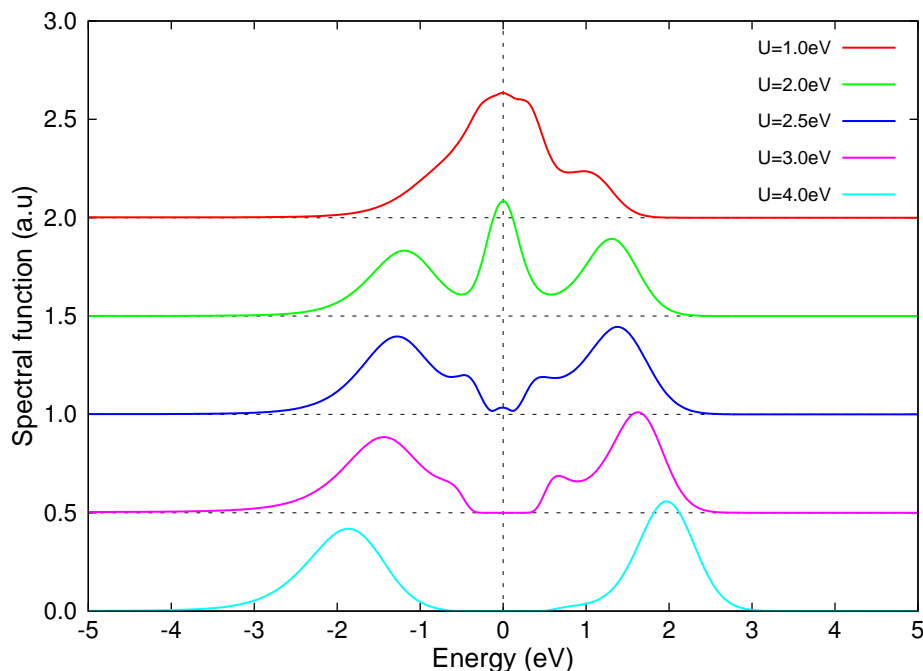


图 8.10: 单粒子谱函数 $A(\omega)$ 随着 U 的变化

了 Mott 金属 – 绝缘体相变。请首先检查每个 u^* 文件夹的 $out.dat$ 文件，确认该次计算没有异常出现。接下来先选取几个典型的 U 值，用 $gnuplot$ 画出相应的虚时格林函数 $G(\tau)$ (参见图8.5)、虚频格林函数的虚部 $\Im G(i\omega)$ (参见图8.6) 与虚频电子自能函数的虚部 $\Im \Sigma(i\omega)$ (参见图8.7)。通过细致比较这几个图，用户可以看到随着 U 的逐渐增大，体系是如何从金属相转变到 Mott 绝缘相的。很明显，当 $U = 2.0$ eV 时，系统还处于金属相，但是当 $U = 2.5$ eV 时，系统已经转化为绝缘相。

对于 Mott 金属 – 绝缘体转变，准粒子权重 Z 最能准确地反应相变的进程。 Z 为大于 0 的有限值意味着金属相， Z 接近于 0 意味着 Mott 绝缘相。我们可以利用一个近似公式来计算准粒子权重 Z ，该公式如下所示：

$$Z \approx \frac{1}{1 - \frac{\Im \Sigma(i\omega_0)}{\omega_0}} \quad (8.1)$$

其中， $\Im \Sigma(i\omega_0)$ 是虚频电子自能函数的虚部在第一个频率点上的值， ω_0 是第一个松原频率点。该公式在温度 $T = 0$ K 时严格成立，当温度很低时仍是一个很好的近似。如图8.8所示，随着 Coulomb 相互作用 U 的增大，准粒子权重 Z 不断下降，最后趋近于 0，体系从金属变为 Mott 绝缘体。此相变的相变点大致位于 $U \sim 2 - 2.5$ eV。另外一个反映相变过程的关键物理量为双占据数。在上一个案例中我们已经介绍了如何得到双占据数，此刻将双占据数对 U 作图，如图8.9 所示。不难发现，双占据数和准粒子权重具有类似的变化趋势，这反应了电子在逐渐增强的 Coulomb 相互作用下，从巡游态变为局域态。

最后，我们可以将虚时格林函数 $G(\tau)$ 进行解析延拓 (请参阅第8.4.1小节)，得到电子谱函数 $A(\omega)$ 。如图8.10所示，增大 U 值时，电子谱函数从近自由电子形状演化到具有三峰结构。继续增大 U ，谱函数

将打开能隙，表明体系进入到 Mott 绝缘体相。

8.2 进阶应用 I: 新的物理

在上一节，我们所介绍的案例均是基于半满的单带 Hubbard 模型，其所包含的物理比较简单。在这一节，我们将以多带模型为例，进一步向用户展示 *iQIST* 软件包的强大功能。多带模型蕴涵了许多新的物理，这是 *iQIST* 软件包的主要战场之所在。

8.2.1 广义相互作用项

利用 *Azalea* 组件可对具有密度 - 密度相互作用的 Hubbard 模型进行了研究，但是对于具有广义相互作用项的多带 Hubbard 模型，基于段表示算法的 *Azalea* 组件就不再适用了。在本小节，我们将展示如何使用基于广义矩阵算法的 *Begonia* 组件来研究 bethe 晶格上的两带半满 Hubbard 模型。

该两带 Hubbard 模型的局域哈密顿量如下：

$$\begin{aligned}
 H_{loc} = & - \sum_{a\sigma} (\mu - \Delta_a) n_{a\sigma} + \sum_a U n_{a\uparrow} n_{a\downarrow} \\
 & + \sum_{a>b, \sigma} [U' n_{a\sigma} n_{b\bar{\sigma}} + (U' - J) n_{a\sigma} n_{b\sigma}] \\
 & - \sum_{a<b} J (d_{a\downarrow}^\dagger d_{b\uparrow}^\dagger d_{b\downarrow} d_{a\uparrow} + d_{b\uparrow}^\dagger d_{b\downarrow}^\dagger d_{a\uparrow} d_{a\downarrow} + h.c.).
 \end{aligned} \tag{8.2}$$

其中， a, b 为能带指标 ($a, b = 1 \sim 2$)， σ 为自旋指标， $n_{a\sigma} = c_{a\sigma}^\dagger c_{a\sigma}$ 为占据数算符， μ 为化学势， $U(U')$ 分别为轨道间 (轨道内) Coulomb 相互作用， Δ 为杂质能级⁸。上式右端的第三项为自旋 - 翻转项与对跃迁项，如果哈密顿量包含了这一项，我们就称此相互作用是广义的，具有旋转不变性，如果没有这一项，那么此相互作用具有密度 - 密度形式。在本案例中，假定 $J = J_z = J_s = J_p$ ，也就是说，此两带模型具有广义形式的相互作用项。对于此类型的哈密顿量，必须使用基于广义矩阵算法的量子杂质求解器组件，例如 *Begonia* 组件或者是 *Lavender* 组件来进行求解。下面我们展示如何使用最简单的 *Begonia* 组件来求解此两带模型，具体计算流程如下。

步骤 1: 创建自己的工作目录/工作文件夹

首先创建 `iqist_test` 文件夹，该文件夹可作为用户的练习目录。如果用户在练习上一个案例时已经创建了此文件夹，那么可以略过此操作。

在 `iqist_test` 文件夹下创建子文件夹 `t821`:

```
$ mkdir t821
```

再进入到 `t821` 文件夹中:

```
$ cd t821
```

⁸在本案例中 $\Delta = 0$ 。

那么当前所在的位置就是/iqist_test/t821。

步骤 2:准备 solver.ctqmc.in

在/opt/iqist/tutor/t821 文件夹中,我们已经准备好了 Begonia 组件的输入文件 solver.ctqmc.in,请将它复制到当前目录下,例如:

```
$ cp /opt/iqist/tutor/t821/solver.ctqmc.in .
```

关于 Begonia 组件的 solver.ctqmc.in 输入文件,用户要注意以下几点:

- 此时 U 、 U_c 、 U_v 、 J_z 、 J_s 与 J_p 等参数是无效的,因为相关参数已经在原子问题程序 Jasmine 组件 (请参阅第7.1节) 里设定了,所以从原则上说这些值设不设都无所谓。但是为了方便事后复查计算过程,我们建议用户还是正确设定上述参数。
- 化学势 μ 设为 3.954869,此时占据数为半满。
- n_{part} 参数的最佳取值不好确定,建议设为 1,这是最稳妥的数值,但是计算效率未必是最佳的。关于 n_{part} 参数的详情,请参阅第6.27节。

步骤 3:准备 atom.in

在/opt/iqist/tutor/t821 文件夹中,我们已经准备好了 Jasmine 组件的输入文件 atom.in,请将它复制到当前目录下,例如:

```
$ cp /opt/iqist/tutor/t821/atom.in .
```

在该文件中,主要设定了如下参数

- $U = U_c = 4.0$
- $J_z = J_s = J_p = 0.25U = 1.0$
- $U_v = U_c - 2J_z = 2.0$
- $\text{isoce} = -1$
- $\text{lsoc} = 0.0$
- $\text{eimp} = 0.0$

至于 atom.in 输入文件的其它参数项,请查看输入文件中的注释以及第7.1节对于输入参数的解释。

步骤 4:运行 Jasmine 组件程序

利用 Jasmine 组件来对角化原子问题,直接执行如下命令:

```
$ atom
```

该程序很快执行完毕,如果没有输出任何错误信息的话,这表示用户的计算已经成功了。在当前目录中将会出现很多以 atom 打头的数据文件,其中唯有 atom.cix 是最重要的,它是基于广义矩阵算法的量子杂质求解器组件必须的输入文件 (请参阅第4.6节),缺失该文件将导致 Begonia 程序不能运行。

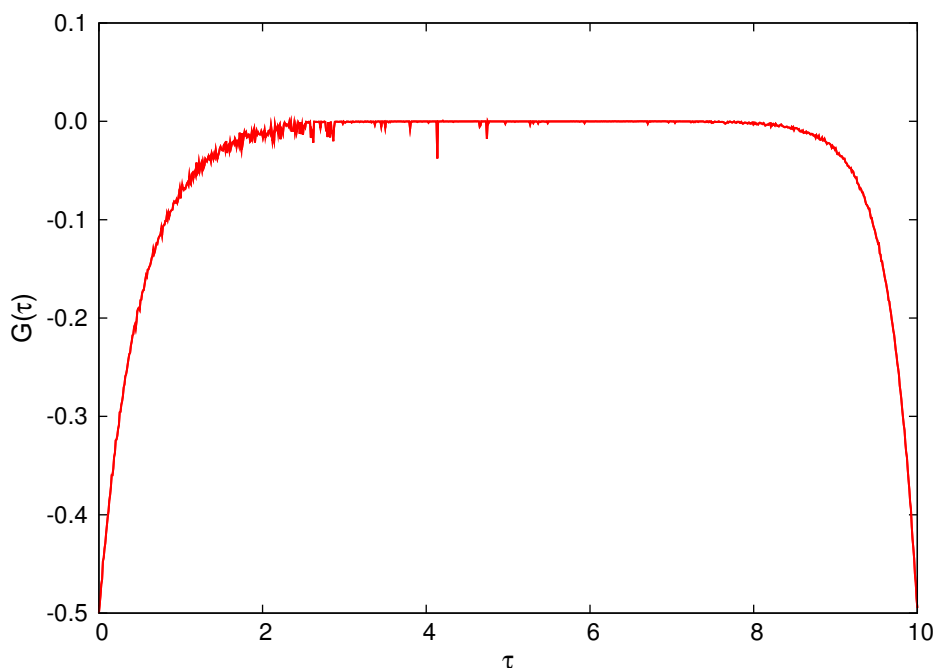


图 8.11: 广义相互作用的两带半满 Hubbard 模型的虚时格林函数 $G(\tau)$

步骤 5: 启动 *Begonia* 组件进行实际计算

在此案例中, 我们使用 8 个计算核心运行 *Begonia* 组件的 *ctqmc* 程序, 用户可以根据现有的计算资源做出适当调整。具体命令如下:

```
$ nohup mpiexec -n 8 ctqmc < /dev/null > out.dat &
```

Begonia 组件程序所需的运行时间较长, 请用户耐心等待。

步骤 6: 监测 *Begonia* 组件的运行状态

密切观察 *ctqmc* 程序的运行情况, 在第一时间对异常情况做出反应十分必要。用户需要不时查看输出文件 *out.dat*, 关注的信息除了前面章节所提到的以外, 用户还需要额外关注的是 Monte Carlo 抽样算法符号问题的统计信息, 如下所示⁹:

```
1  negative sign counter:      0
2  averaged sign sampler:    1.00000
```

其中第 1 行是抽样时负符号出现的次数, 第 2 行是正符号和负符号的平均值。如果该平均值的绝对值趋于 0, 那么表示正号和负号各占一半, 符号问题最严重了, 此时 Monte Carlo 抽样结果完全是不可信的。在我们研究的这个模型中, 几乎没有负符号出现, 因此计算是可信的。

步骤 7: 数据的分析和处理

⁹请注意: *Azalea*、*Gardenia* 与 *Narcissus* 组件是不会输出如下统计信息的, 因为在这些组件中不存在负符号问题⁷¹。

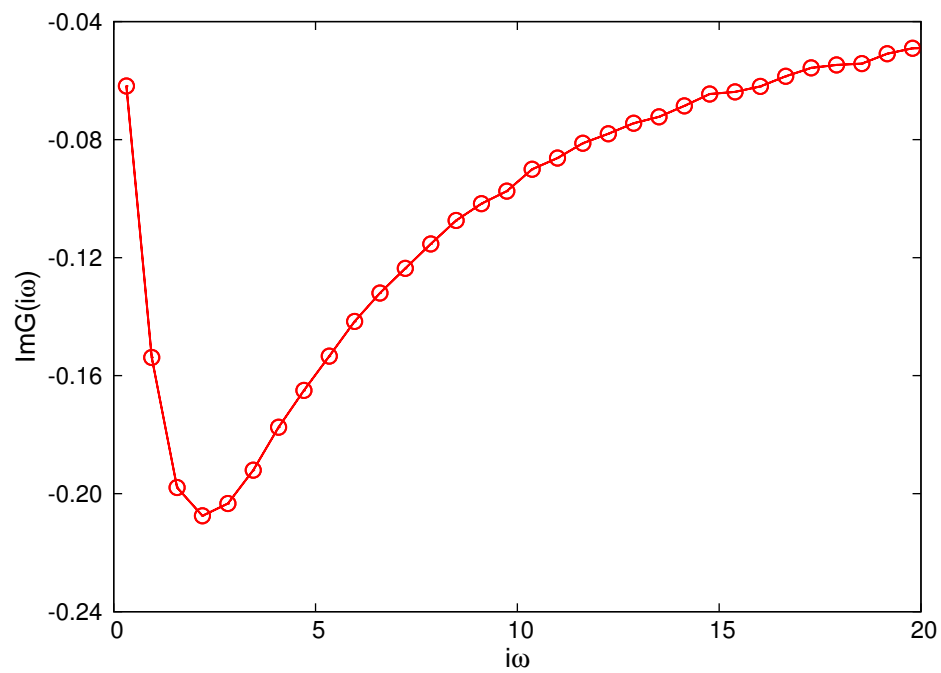


图 8.12: 广义相互作用的两带半满 Hubbard 模型的虚频格林函数的虚部 $\Im G(i\omega)$

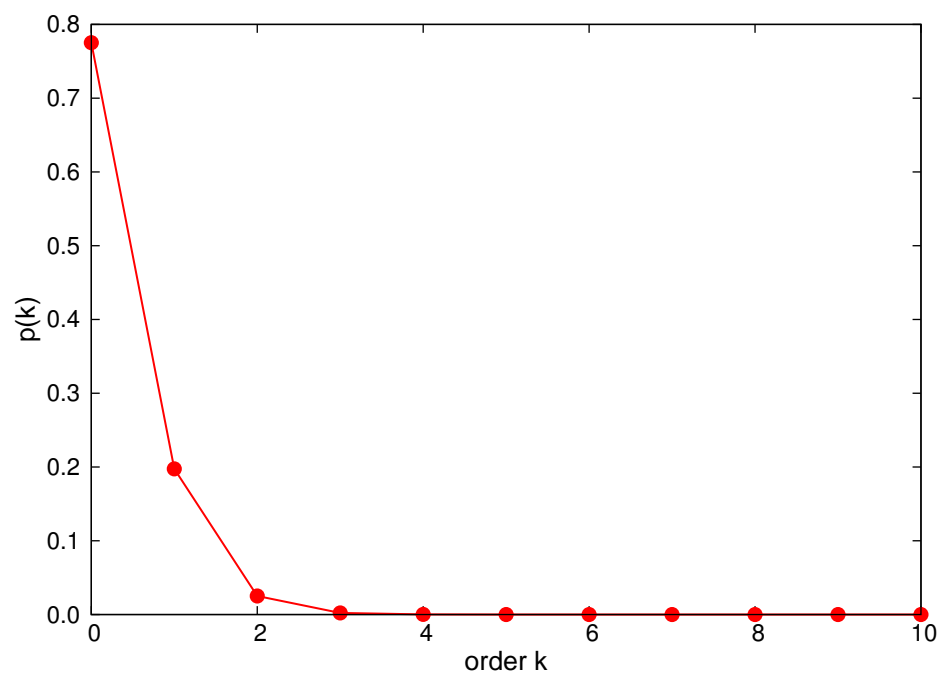


图 8.13: 广义相互作用的两带半满 Hubbard 模型的图形微扰展开序列的阶数分布 P_H

确认计算完成后，用户可以按照上一节中介绍的方法分析计算结果。在这里分析任务就交给用户自己去完成了。我们计算结果如下：图8.11 展示的是虚时格林函数 $G(\tau)$ ，图8.12展示的是虚频格林函数的虚部 $\Im G(i\omega)$ ，图8.13展示的是图形微扰展开序列的阶数分布 P_H 。原子组态概率如下所示：

#	state	probability:	index	prob	occupy	spin
1	1	0.000017	0.000000	0.000000		
2	2	0.002528	1.000000	0.500000		
3	3	0.002514	1.000000	0.500000		
4	4	0.002538	1.000000	-0.500000		
5	5	0.002557	1.000000	-0.500000		
6	6	0.326522	2.000000	1.000000		
7	7	0.326595	2.000000	0.000000		
8	8	0.326408	2.000000	-1.000000		
9	9	0.000057	2.000000	0.000000		
10	10	0.000054	2.000000	0.000000		
11	11	0.000035	2.000000	0.000000		
12	12	0.002546	3.000000	0.500000		
13	13	0.002559	3.000000	0.500000		
14	14	0.002537	3.000000	-0.500000		
15	15	0.002510	3.000000	-0.500000		
16	16	0.000022	4.000000	0.000000		

8.2.2 自旋 – 轨道耦合作用

在本小节，我们将引导用户使用 *iQIST* 软件包来研究含自旋 – 轨道耦合项的系统。对于重元素体系，尤其是 $4f/5f$ 电子系统而言，自旋 – 轨道耦合效应起着十分重要的作用，它与 Coulomb 相互作用之间存在着相互竞争的关系，这也是最近凝聚态物理领域十分热门的研究课题。此次我们的研究对象是一个定义在 bethe 晶格上的三带 Hubbard 模型，该模型不但具有广义相互作用项 (参见公式 (8.2))，并且还具有自旋 – 轨道耦合项。自旋 – 轨道耦合项的哈密顿量如下所示：

$$H_{soc} = \sum_{a\sigma, b\sigma'} \zeta \langle a\sigma | l_x s_x + l_y s_y + l_z s_z | b\sigma' \rangle d_{a\sigma}^\dagger d_{b\sigma'} \quad (8.3)$$

其中 ζ 为自旋 – 轨道耦合作用强度。在 *iQIST* 软件包中，唯有定制的 *Lavender* 组件¹⁰可以求解此模型哈密顿量。具体的计算步骤如下。

步骤 1: 创建自己的工作目录/工作文件夹

¹⁰标准版的 *Lavender* 组件不可用于此类计算。

首先创建 `iqist_test` 文件夹，该文件夹可作为用户的练习目录。如果用户在练习上一个案例时已经创建了此文件夹，那么可以略过此操作。

在 `iqist_test` 文件夹下创建子文件夹 `t822`:

```
$ mkdir t822
```

再进入到 `t822` 文件夹中:

```
$ cd t822
```

那么当前所在的位置就是 `/iqist_test/t822`。

步骤 2: 准备 `solver.ctqmc.in`

在 `/opt/iqist/tutor/t822` 文件夹中，我们已经准备好了 `Lavender` 组件的输入文件 `solver.ctqmc.in`，请将它复制到当前目录下，例如:

```
$ cp /opt/iqist/tutor/t822/solver.ctqmc.in .
```

关于 `Lavender` 组件的 `solver.ctqmc.in` 输入文件，用户要注意以下几点：

- 由于有自旋 - 轨道耦合项， S_z 不再是好量子数，需要设置 `isspn = 2`，`issun = 2`。
- 使用 Legendre 正交多项式方法测量虚时格林函数，设置 `isort = 2`，这样可以得到比较平滑的虚时格林函数 $G(\tau)$ 。
- 此时 U 、 U_c 、 U_v 、 J_z 、 J_s 与 J_p 参数是无效的，因为相关参数已经在原子问题程序里设定好了，所以从原则上说这些值设不设都无所谓。但是为了方便事后复查计算过程，我们建议用户还是正确设定上述参数。
- 化学势 `mune` 设为 8.360310，此时总占据数恰好为 4.0。
- 设置 Legendre 正交多项式的最大展开阶数 `lemax = 24`。
- 设置 Legendre 正交多项式线性网格点数目 `legrd = 20001`。
- 关闭 GLOBAL FLIP 更新操作，设置 `nflip = 0`。

步骤 3: 准备 `solver.eimp.in`

在 `/opt/iqist/tutor/t822` 文件夹中，我们已经准备好了 `Lavender` 组件的输入文件 `solver.eimp.in`，请将它复制到当前目录下，例如:

```
$ cp /opt/iqist/tutor/t822/solver.eimp.in .
```

在该文件中，我们设定杂质能级都为 0.0，前 4 个轨道兼并，对应 $j = 3/2$ 态，后 2 个轨道兼并，对应 $j = 1/2$ 态。`Lavender` 组件将利用这些设置对轨道进行强制对称化 (`issun = 2`)。

步骤 4: 准备 `dft.atom.in`

由于 *Jasmine* 组件目前尚不支持自旋 – 轨道耦合作用，因此我们无法再利用它来对角化原子问题，以产生 `atom.cix` 文件，详情请参阅第7.1节。十分幸运的是，杜亮博士为他的原子问题程序 *rambutan* 针对 *Lavender* 组件开发了专用的接口，我们可以使用他的程序来产生 `atom.cix` 文件。*rambutan* 程序所需要的输入文件为 `dft.atom.in`。在 `/opt/iquist/tutor/t822` 文件夹中，我们已经准备好了 `dft.atom.in` 文件，请将它复制到当前目录下，例如：

```
$ cp /opt/iquist/tutor/t822/dft.atom.in .
```

在 `dft.atom.in` 文件中，主要设定了以下参数：

- $U = U_c = 4.0$
- $J_z = J_s = J_p = 0.25U = 1.0$
- $U_v = U_c - 2J_z = 2.0$
- $\zeta = -0.20^{11}$

至于 `dft.atom.in` 输入文件的其它设置参数，请用户仔细查看输入文件中的注释或者是直接向杜亮博士咨询 (<mailto:duliang@iphy.ac.cn>)。

步骤 5: 运行 *rambutan* 程序

运行 *rambutan* 程序来对角化原子问题，*rambutan* 的可执行程序名为 `atomic`，使用如下命令：

```
$ atomic
```

该程序很快执行完毕，如果没有输出任何错误信息的话，表示用户的计算很成功，*rambutan* 程序产生了标准的 `atom.cix` 文件。此文件是基于广义矩阵算法的量子杂质模型求解器所必须的输入文件（请参阅第4.6节），缺失该文件将导致 *Lavender* 组件的 `ctqmc` 程序不能运行。

步骤 6: 启动 *Lavender* 组件进行实际计算

在此案例中，我们使用 8 个计算核心运行 *Lavender* 组件的 `ctqmc` 程序，用户可以根据现有的计算资源做出适当调整。具体命令如下：

```
$ nohup mpiexec -n 8 ctqmc < /dev/null > out.dat &
```

Lavender 组件程序所须的运行时间较长，请用户耐心等待。

步骤 7: 监测 *Lavender* 组件的运行状态

密切观察 `ctqmc` 程序的运行情况，在第一时间对异常情况做出反应十分必要。用户需要不时查看输出文件 `out.dat`，关注的信息除了前面章节所提到的以外，用户还需要额外关注的是 Monte Carlo 抽样算法符号问题的统计信息，如下所示¹²：

¹¹ 此为 *rambutan* 程序中定义的自旋 – 轨道耦合强度，请注意它采用了负号规范。

¹² *Begonia*、*Lavender*、*Camellia*、*Epiphyllum*、*Pansy* 与 *Manjushaka* 等基于广义矩阵算法的量子杂质求解器组件均会输出与负符号问题有关的统计信息。

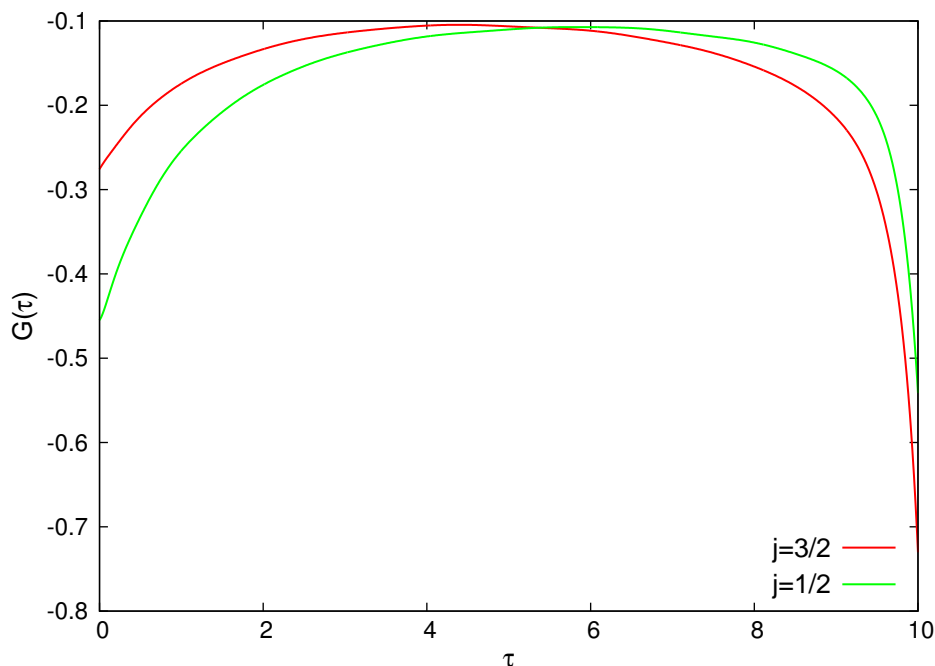


图 8.14: 包含自旋 – 轨道耦合的三带 Hubbard 模型的虚时格林函数 $G(\tau)$

```

1   negative sign counter:    3802766
2   averaged sign sampler:   0.61972

```

其中第 1 行是抽样时负符号出现的次数，第 2 行是正符号和负符号的平均值。如果该平均值的绝对值趋于 0，那么表示正号和负号各占一半，符号问题最严重了，此时 Monte Carlo 抽样结果完全是不可信的。在我们研究的这个模型中，有负符号出现，但是负符号问题还不是十分严重，计算结果还是基本可信的。

步骤 8: 数据的分析和处理

在漫长的等待过后，计算终于成功结束了。在确认计算完成后，用户可以按照第 8.1 节中介绍的方法分析计算结果。在这里分析任务就交给用户自己去完成了。我们计算结果如下：图 8.14 展示的是虚时格林函数 $G(\tau)$ ，由于采用了 Legendre 正交多项式方法测量 $G(\tau)$ ，因此 $G(\tau)$ 函数的曲线显得尤为平滑；图 8.15 展示的是虚频格林函数的虚部 $\Im G(i\omega)$ ，显而易见，自旋 – 轨道耦合效应导致原来简并的三条能带发生了劈裂；图 8.16 展示的是图形微扰展开序列的阶数分布 P_H 。占据数情况如下：

- $j = 3/2$ 轨道，平均占据数为 0.728540
- $j = 1/2$ 轨道，平均占据数为 0.542734

用户还可以变换 Coulomb 相互作用强度 U 以及自旋 – 轨道耦合作用强度 ζ ，重复进行上述计算，深入探讨自旋 – 轨道耦合作用对于关联电子系统电子结构的影响¹³。

¹³相关的研究结果可以参阅 [arXiv:1211.2055](https://arxiv.org/abs/1211.2055)。

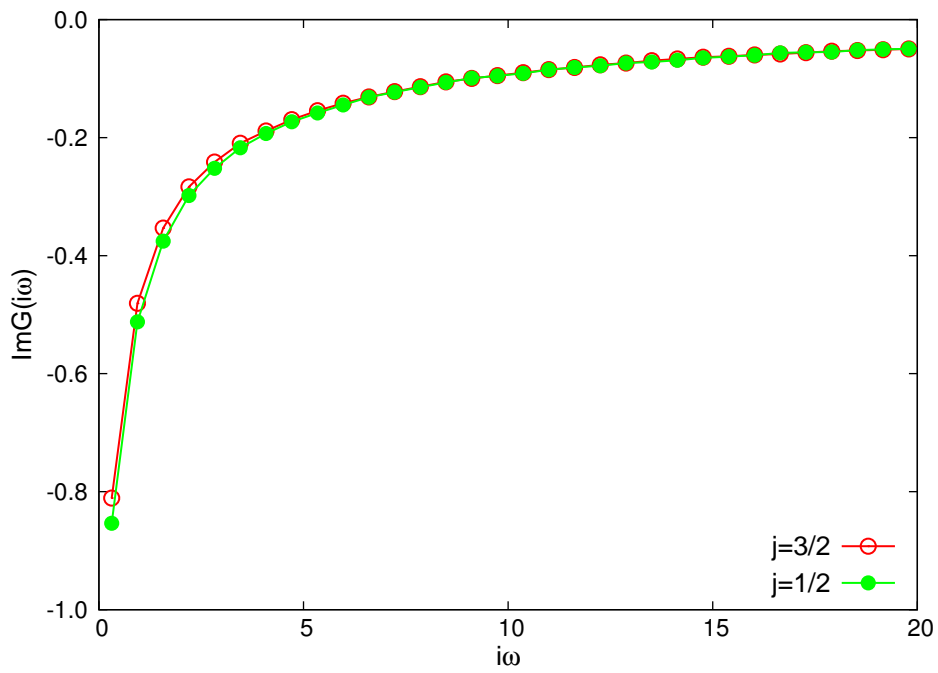


图 8.15: 包含自旋 - 轨道耦合的三带 Hubbard 模型的虚频格林函数的虚部 $\Im G(i\omega)$

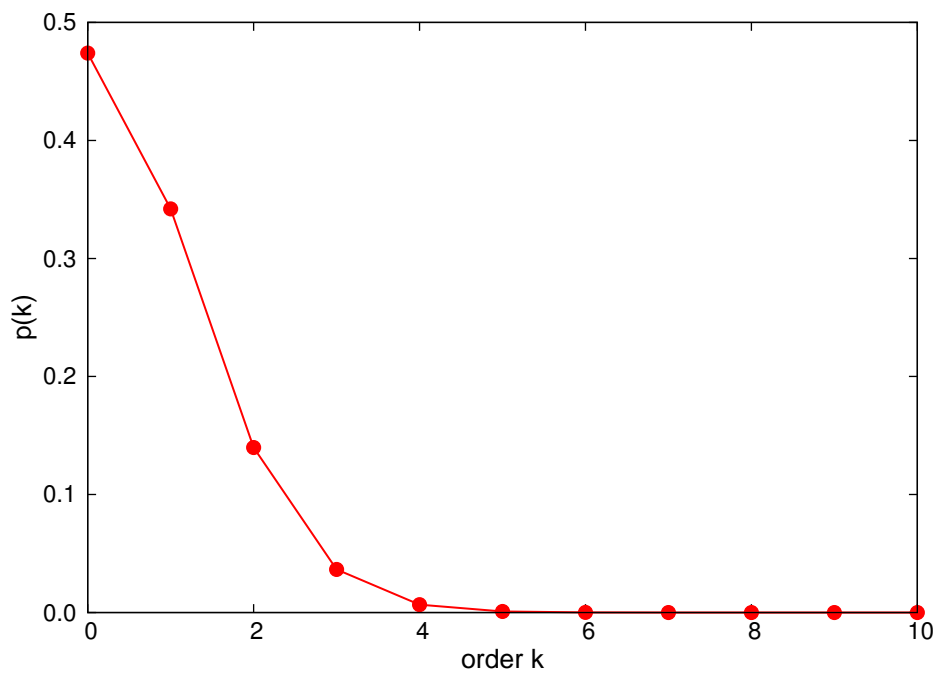


图 8.16: 包含自旋 - 轨道耦合的三带 Hubbard 模型的图形微扰展开序列的阶数分布 P_H

8.2.3 晶体场劈裂效应

在本小节中我们将利用 iQIST 软件包来研究多带 Hubbard 模型中的晶体场劈裂效应。我们的研究对象是 bethe 晶格上的两带 Hubbard 模型，晶体场将使两个能带发生劈裂。为了简化问题，此处暂不考虑广义相互作用，仅仅考虑密度 - 密度形式的相互作用。这样我们就可以利用高效的 Azalea 组件来解决问题。

步骤 1: 创建自己的工作目录/工作文件夹

首先创建 iqist_test 文件夹，该文件夹可作为用户的练习目录。如果用户在练习上一个案例时已经创建了此文件夹，那么可以略过此操作。

在 iqist_test 文件夹下创建子文件夹 t823:

```
$ mkdir t823
```

再进入到 t823 文件夹中:

```
$ cd t823
```

那么当前所在的位置就是 /iqist_test/t823。

步骤 2: 准备 solver.ctqmc.in

在 /opt/iqist/tutor/t823 文件夹中，我们已经准备好了 Azalea 组件的输入文件 solver.ctqmc.in，请将它复制到当前目录下，例如:

```
$ cp /opt/iqist/tutor/t823/solver.ctqmc.in .
```

此 solver.ctqmc.in 文件的关键设置如下：

- 采用自洽计算模式，isscf = 2
- 强制进行轨道对称化处理，issun = 2
- 强制进行自旋对称化处理，isspn = 1
- 该模型有 2 条能带，nband = 2
- 自旋投影方向，nspin = 2
- 轨道数目，norbs = 4
- 原子组态数目，ncfgs = 16
- 最大 DMFT 自洽迭代次数，niter = 20
- 库仑相互作用强度， $U = U_c = 4.0$ ， $U_v = U_c - 2J_z = 2.0$
- Hund 交换作用强度， $J_z = 0.25U = 1.0$ ， $J_s = J_p = 0.0$

- 化学势, $\mu = 2.0$
- 反温度, $\beta = 10.0$
- 最近邻跃迁系数, $t = 0.50$

上述设置与不存在晶体场劈裂时的情况基本一致, 没有什么特别之处。

步骤 3: 准备 solver.eimp.in

各轨道的杂质能级必须在 solver.eimp.in 文件中设定。在默认情况下, 杂质能级都是 0.0, 所有轨道完全简并, 因此可以不提供 solver.eimp.in 文件给量子杂质求解器组件作为输入。但是在本案例中, 各轨道的杂质能级不再为 0.0, 轨道之间存在着晶体场劈裂, 因此 solver.eimp.in 文件是必须的 (关于 solver.eimp.in 文件的细节, 请参阅第4.2节)。

在 /opt/iqist/tutor/t823 文件夹中, 我们已经准备好了 solver.eimp.in, 请将它复制到当前目录下, 可以使用如下命令:

```
$ cp /opt/iqist/tutor/t823/solver.eimp.in .
```

在该文件中, 设定轨道 1 和 3 (亦即能带 1) 的杂质能级为 +1.0, 轨道 2 和 4 (亦即能带 2) 的杂质能级为 -1.0, 从而晶体场劈裂的大小为 2.0。在 solver.eimp.in 文件中, 还对各轨道的对称性做出了定义: 轨道 1 与轨道 3 对称, 轨道 2 和轨道 4 对称。由于 issun 参数被设为 2, 因此量子杂质求解器组件将依此对各个轨道进行强制对称化。

步骤 4: 启动 Azalea 组件进行实际计算

在此案例中, 我们使用 8 个计算核心运行 Azalea 组件的 ctqmc 程序, 用户可以根据现有的计算资源做出适当调整。具体命令如下:

```
$ nohup mpiexec -n 8 ctqmc < /dev/null > out.dat &
```

步骤 5: 监测 Azalea 组件的运行状态

请用户按照之前讲述的方法监测程序运行状态。

步骤 6: 数据的分析和处理

确认计算完成后, 用户可以按照第8.1节中介绍的方法分析计算结果。在这里分析任务就交给用户自己去完成了。我们计算结果如下: 图8.17 展示的是虚时格林函数 $G(\tau)$, 图8.18展示的是虚频格林函数的虚部 $\Im G(i\omega)$ 。占据数情况如下:

- 能量高的轨道 (upper orbital) 的平均占据数为 0.257216
- 能量低的轨道 (lower orbital) 的平均占据数为 0.500051

用户可以调整能带数目、相互作用强度、晶体场劈裂强度等参数, 研究更为复杂的情况。晶体场劈裂与 Hund 交换作用 J 相互耦合, 可以诱发出具有轨道选择性的 Mott 金属 - 绝缘体相变现象¹⁴。

¹⁴相关的研究结果可以参阅arXiv:1203.5159。

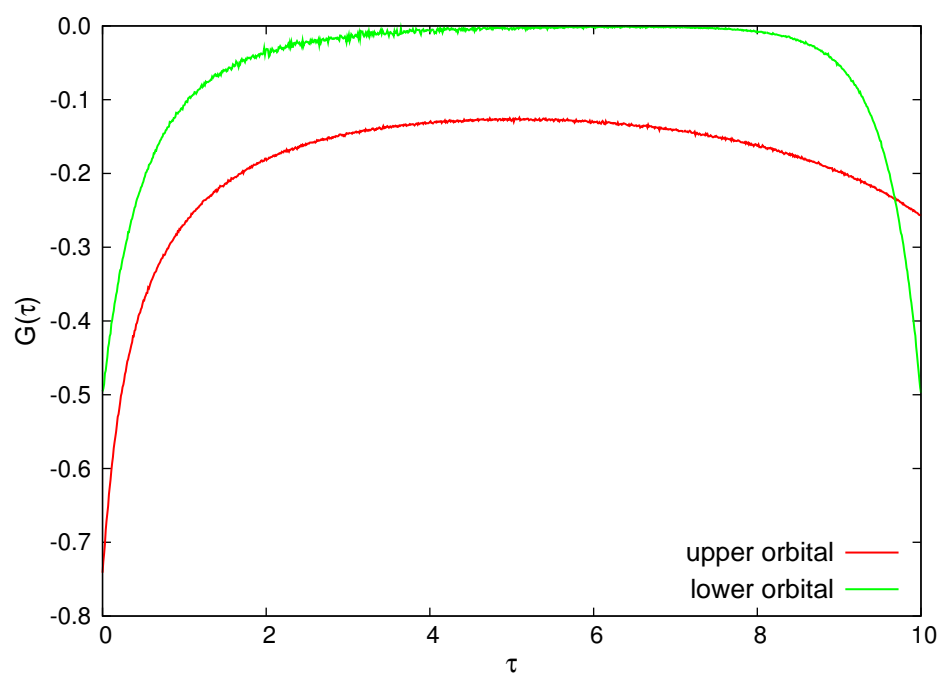


图 8.17: 包含晶体场劈裂的两带 Hubbard 模型的虚时格林函数 $G(\tau)$

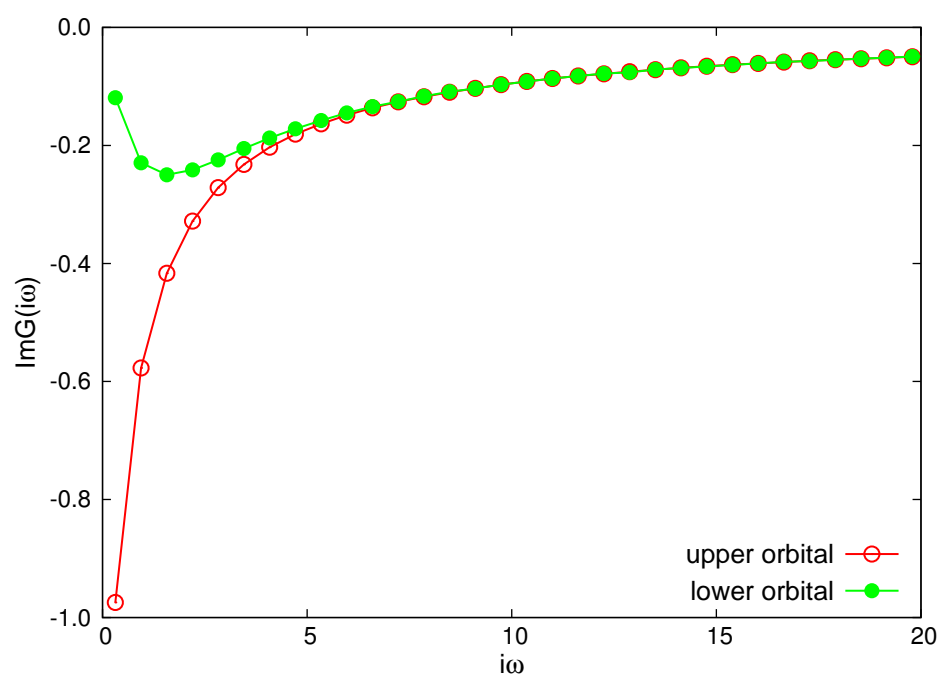


图 8.18: 包含晶体场劈裂的两带 Hubbard 模型的虚频格林函数的虚部 $\Im G(i\omega)$

8.2.4 动态屏蔽效应

此前我们所研究的 Hubbard 模型中, Coulomb 相互作用强度 U 是一个恒定值, 换言之 Coulomb 相互作用是一个恒定的相互作用, 但是实际情况并非如此。Coulomb 相互作用具有频率依赖性¹⁵, 也就是说:

$$U = U(\omega) \quad (8.4)$$

以金属 Gd 为例, 当频率 ω 由 0 变化至 3 eV 时, $U(\omega)$ 由 6.5 eV 变化至 17 eV。再例如金属 Ce, 当频率由 0 变化至 4 eV 时, $U(\omega)$ 由 3.5 eV 变化至 7 eV。我们定义 $U_0 = U(\omega = 0)$, 在传统的 LDA + U 计算或者是 LDA + DMFT 计算中, 通常忽略掉 Coulomb 相互作用的频率依赖性, 而仅仅考虑 U_0 的效果。P. Werner 在段表示算法的基础上发展了一套全新的算法, 可以将 U 的频率依赖性严格考虑进去²⁹。我们在 [Narcissus](#) 组件中也实现了该算法, 因此在本小节, 我们将演示如何使用 [Narcissus](#) 组件研究 U 随频率变化的单带半满 Hubbard 模型。

我们分别采用如下两种模型来描述 U 的频率依赖性:

plasmon pole 模型

$$\text{Re}U(\omega) = U + \frac{2\lambda^2\omega'}{\omega^2 - \omega'^2} \quad (8.5)$$

ohmic 模型

$$\text{Re}U(\omega) = U + \alpha\omega \ln \left| \frac{\omega_c + \omega}{\omega_c - \omega} \right| - 2\alpha\omega_c \quad (8.6)$$

在 [Narcissus](#) 组件中, 我们设计了 `isscr` 参数 (请参阅第6.7节) 来控制是否激活动态屏蔽效应计算功能, 并且通过 `lc` 参数 (请参阅第6.23节) 与 `wc` 参数 (请参阅第6.24节) 来设定具体的 plasmon pole 模型或者是 ohmic 模型。在本案例中, 我们考虑如下的单带半满模型: $U = 6.0$ eV, $\lambda = \omega' = \alpha = \omega_c = 1.0$, 相应的 plasmon pole 模型与 ohmic 模型如下图8.19所示。

具体的计算流程如下所示。

步骤 1: 创建自己的工作目录/工作文件夹

首先创建 `iqist_test` 文件夹, 该文件夹可作为用户的练习目录。如果用户在练习上一个案例时已经创建了此文件夹, 那么可以略过此操作。

在 `iqist_test` 文件夹下创建子文件夹 `t824`:

```
$ mkdir t824
```

再进入到 `t824` 文件夹中:

```
$ cd t824
```

那么当前所在的位置就是 `/iqist_test/t824`。

步骤 2: 准备 `solver.ctqmc.in`

¹⁵此即所谓的动态屏蔽效应。

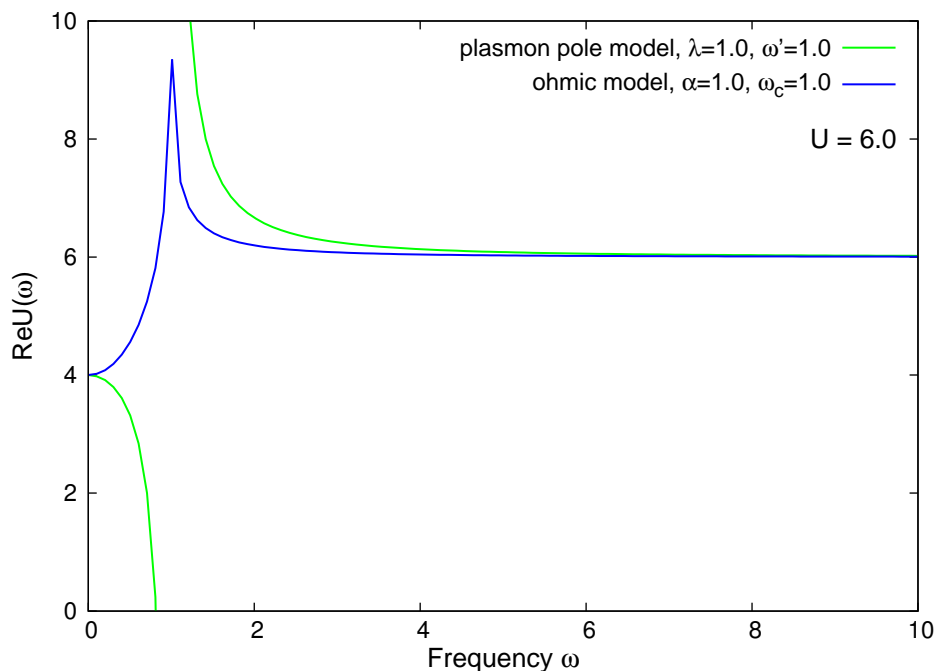


图 8.19: plasmon pole 模型与 ohmic 模型示意图

在 `/opt/iqist/tutor/t824` 文件夹中，我们已经准备好了 `Narcissus` 组件的输入文件 `solver.ctqmc.in`，请将它复制到当前目录下，例如：

```
$ cp /opt/iqist/tutor/t824/solver.ctqmc.in .
```

此 `solver.ctqmc.in` 文件的关键设置如下：

- 采用自洽计算模式，`isscf = 2`
- 不计算任何高阶关联函数，`isvrt = 1`
- 不激活动态屏蔽效应计算，`isscr = 1`
- 该模型有 1 条能带，`nband = 1`
- 自旋投影方向，`nspin = 2`
- 轨道数目，`norbs = 2`
- 原子组态数目，`ncfgs = 4`
- 最大 DMFT 自洽迭代次数，`niter = 20`
- 库仑相互作用强度， $U = U_c = U_v = 6.0$
- Hund 交换作用强度， $J_z = J_s = J_p = 0.0$

- $U(\omega)$ 的模型参数, $lc = wc = 0.0$
- 化学势, $\mu = 3.0$
- 反温度, $\beta = 10.0$
- 最近邻跃迁系数, $t = 0.50$

在本案例中, 我们分别计算以下三种情况:

- 静态 Coulomb 相互作用, $U = U_0 = 6.0$ eV
- 动态屏蔽效应 $U(\omega)$, plasmon pole 模型, $\lambda = \omega' = 1.0$
- 动态屏蔽效应 $U(\omega)$, ohmic 模型, $\alpha = \omega_c = 1.0$

此处首先考虑第一种情况, 因此将 `isscr` 设为 1, 而将 `lc` 与 `wc` 参数全部置为 0.0。

步骤 3: 启动 `Narcissus` 组件进行实际计算

在此案例中, 我们使用 8 个计算核心运行 `Narcissus` 组件的 `ctqmc` 程序, 用户可以根据现有的计算资源做出适当调整。具体命令如下:

```
$ nohup mpiexec -n 8 ctqmc < /dev/null > out.dat &
```

步骤 4: 监测 `Narcissus` 组件的运行状态

请用户按照之前讲述的方法监测程序运行状态。

步骤 5: plasmon pole 模型计算

当静态模型的计算完毕后, 注意保存计算结果。然后使用 `vim` 编辑器打开 `solver.ctqmc.in` 文件, 将 `isscr` 参数的值修改为 3, `lc` 参数¹⁶的值修改为 1.0, `wc` 参数¹⁷的值修改为 1.0, 然后保存退出。按照步骤 3 与步骤 4 的做法启动 `Narcissus` 组件进行计算。此时, `Narcissus` 组件将求解一个具有动态 Coulomb 相互作用的单带半满 Hubbard 模型, 而 U 的频率依赖性由 plasmon pole 模型近似描述。

步骤 6: ohmic 模型计算

当 plasmon pole 模型的计算完毕后, 注意保存计算结果。然后使用 `vim` 编辑器打开 `solver.ctqmc.in` 文件, 将 `isscr` 参数的值修改为 4, `lc` 参数¹⁸的值修改为 1.0, `wc` 参数¹⁹的值修改为 1.0, 然后保存退出。按照步骤 3 与步骤 4 的做法启动 `Narcissus` 组件进行计算。此时, `Narcissus` 组件将求解一个具有动态 Coulomb 相互作用的单带半满 Hubbard 模型, 而 U 的频率依赖性由 ohmic 模型近似描述。

步骤 7: 数据的分析和处理

在漫长的等待过后, 计算终于成功结束了。在确认计算完成后, 用户可以按照第 8.1 节中介绍的方法分析计算结果。在这里分析任务就交给用户自己去完成了。我们计算结果如下: 图 8.20 展示的是虚频格林

¹⁶亦即公式 (8.5) 中的 λ 参数。

¹⁷亦即公式 (8.5) 中的 ω' 参数。

¹⁸亦即公式 (8.6) 中的 α 参数。

¹⁹亦即公式 (8.6) 中的 ω_c 参数。

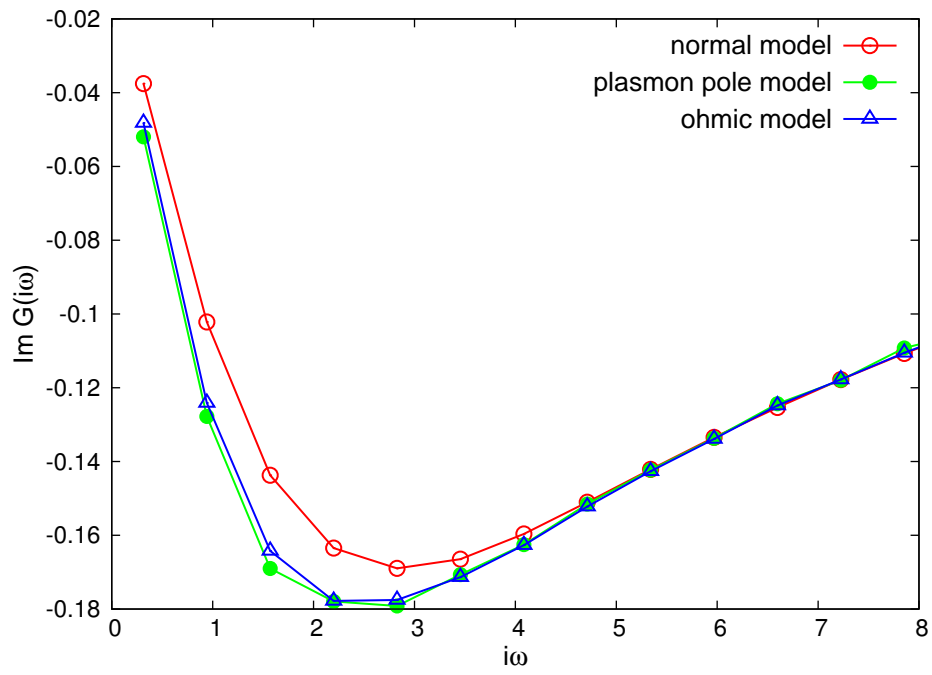


图 8.20: 包含动态屏蔽效应的单带半满 Hubbard 模型的虚频格林函数的虚部 $\Im G(i\omega)$

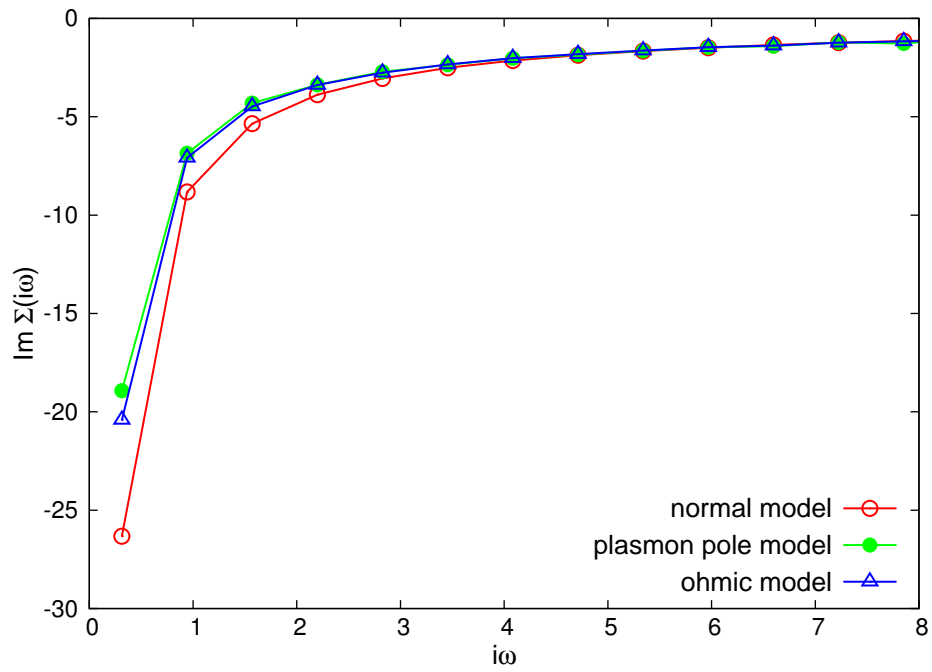


图 8.21: 包含动态屏蔽效应的单带半满 Hubbard 模型的电子自能函数的虚部 $\Im \Sigma(i\omega)$

函数的虚部 $\Im G(i\omega)$ ；图8.21展示的是电子自能函数的虚部 $\Im \Sigma(i\omega)$ 。显而易见，动态屏蔽效应对 $G(i\omega)$ 与 $\Sigma(i\omega)$ 均有十分重大的影响。

用户还可以变换 isscf 、 U_c 、 U_v 、 lc 与 wc 等参数的值，重复进行上述计算，深入探讨动态屏蔽效应对关联电子系统电子结构的影响。

8.3 进阶应用 II: 物理量的精确测量

本节将介绍如何使用量子杂质求解器组件精确测定各种关键的物理量。

8.3.1 单步执行模式

在本小节里，我们简单的说明一下如何启用量子杂质求解器的单步执行模式。所谓的单步执行模式，指的是量子杂质求解器不做 DMFT 迭代计算，仅仅只求解量子杂质模型一次。单步执行模式主要有两种用途：

- 在 LDA + DMFT 计算模式中，量子杂质模型求解器必须以单步模式运行
- 在自洽迭代已经完成后，精确测定特定的物理量

进入单步计算模式十分简单，用户只需要在 `solver.ctqmc.in` 输入文件里设置控制参数 `isscf = 1` 即可（请参阅第6.1节）。如果用户希望用已经得到的杂化函数²⁰作为输入，那么用户只需要将包含杂化函数的数据文件 `solver.hyb.dat` 重命名为 `solver.hyb.in` 即可。特别注意：如果用户使用的是其它程序产生的杂化函数，请预先转化为 *iQIST* 杂化函数的特有格式（请参阅第4.3节）。

8.3.2 data binning 模式

所谓的 data binning 模式，指的是在 Monte Carlo 抽样统计过程中，每间隔固定的时间，将观测量打包输出。当程序结束后，我们将获得一个个数据包，这些数据包通常称之为 bin。那么 bin 的数目取决于抽样时间以及间隔时间的长短。我们在量子杂质求解器组件中设计这么一种 data binning 模式，主要目的是精确测定虚时格林函数 $G(\tau)$ 或者是其它的物理量。在本小节中，我们将展示如何进入 data binning 计算模式，为基于最大熵方法的解析延拓程序²¹积累虚时格林函数的抽样样本。解析延拓程序将从这些样本数据中抽取出重要的统计信息，例如协方差。

data binning 模式产生的 data bin 文件以 `solver.green.bin.*` 为名，其中 * 号代表 bin 的编号。要进行 data binning 计算很简单，其实就是在做完 DMFT 自洽计算之后，以最后收敛得到的虚频杂化函数 $\Delta(i\omega)$ 为输入再做一次单步计算。此前，我们已经在第8.1.1小节中自洽求解了一个单带半满 Hubbard 模

²⁰该杂化函数由量子杂质求解器组件生成。

²¹亦即 *Hibiscus* 组件的 `entropy` 程序，请参阅第7.2节。

型，获得了收敛后的虚频杂化函数。现在我们将以此为基础来进行 data binning 计算，我们的目标是要产生 100 个 data bins。具体的计算流程如下。

步骤 1: 创建自己的工作目录/工作文件夹

首先创建 iqist_test 文件夹，该文件夹可作为用户的练习目录。如果用户在练习上一个案例时已经创建了此文件夹，那么可以略过此操作。

在 iqist_test 文件夹下创建子文件夹 t832:

```
$ mkdir t832
```

再进入到 t832 文件夹中:

```
$ cd t832
```

那么当前所在的位置就是 /iqist_test/t832。

步骤 2: 准备 solver.ctqmc.in

在 /opt/iqist/tutor/t832 文件夹中，我们已经准备好了 *Azalea* 组件的输入文件 solver.ctqmc.in，请将它复制到当前目录下，例如:

```
$ cp /opt/iqist/tutor/t832/solver.ctqmc.in .
```

在 solver.ctqmc.in 文件中，有如下几个地方需要特别注意:

- 设置单步计算模式，`isscf = 1`
- 设置 data binning 计算模式，`isbin = 2`
- `nwrite = nsweep / 100`，这样就能产生 100 个 data bins

步骤 3: 准备 solver.hyb.in

除了 solver.ctqmc.in 文件以外，solver.hyb.in 也是必须的输入文件，它提供了初始化的杂化函数 $\Delta(i\omega)$ 。在 /opt/iqist/tutor/t832 文件夹中，我们已经准备好了 *Azalea* 组件的输入文件 solver.hyb.in，请将它复制到当前目录下，例如:

```
$ cp /opt/iqist/tutor/t832/solver.hyb.in .
```

实际上，用户完全可以不使用我们所提供的 solver.hyb.in 文件。用户在完成例子 t811 后 (参阅第 8.1.1 小节)，可以将最后得到杂化函数文件 solver.hyb.dat 重命名为 solver.hyb.in，然后将其拷贝到当前目录下即可，结果应该是完全一样的。

步骤 4: 启动 *Azalea* 组件进行实际计算

在此案例中，我们使用 8 个计算核心运行 *Azalea* 组件的 ctqmc 程序，用户可以根据现有的计算资源做出适当调整。具体命令如下:

```
$ nohup mpiexec -n 8 ctqmc < /dev/null > out.dat &
```

此案例所须的运行时间较长，请用户耐心等待。

步骤 5: 数据的分析和处理

计算结束后, 用户就能得到 100 个 data bins 文件。在讲述后处理的章节时, 我们将要用这些数据来进行最大熵解析延拓 (请参阅第8.4.1小节)。

最后需要提醒的一点是, 如果用户希望获得更多的 data bins, 例如 1000 个, 请不要简单的将 nsweep 参数扩大 100 倍, 因为这有可能导致 integer 类型的 nsweep 参数溢出。目前最好的方法是建立 10 个 imp_bin* 文件夹, 分别进行上述计算。

8.3.3 虚时格林函数

虚时格林函数 $G(\tau)$ 是量子杂质求解器组件所输出的最重要的观测量之一。那么为了获取精确的 $G(\tau)$ 数据, 用户可以遵循以下的规则。

- 采用并行模式运行量子杂质求解器 (请参阅第3.3节)。iQIST 的量子杂质求解器组件对于计算核心的数目没有任何限制。一般而言, 在其它条件相同的情况下, 计算核心的数目越多, 所获得的计算结果就越精确。
- 增加 Monte Carlo 抽样的次数 (nsweep 参数, 请参阅第6.43节)。在其它条件相同的情况下, 抽样次数越多, 所获得的计算结果就越精确。
- 适当减少抽样间隔 (ncarlo 参数, 请参阅第6.47节)。在自关联程度得到控制的前提下, 减少抽样间隔就相当于增加抽样次数, 有利于获得更精确的计算结果。
- 采用 data binning 计算模式 (isbin 参数, 请参阅第6.4节)。采用 data binning 计算模式就相当于增加抽样次数十倍, 有利于获得更精确的计算结果。
- 采用正交多项式方法测量 $G(\tau)$ (isort 参数, 请参阅第6.5 节)²²。对于具有金属性质的系统而言, 应用正交多项式方法能够获得十分平滑的虚时格林函数。对于具有绝缘体性质的系统而言, 正交多项式方法的计算结果不佳, 但是可以采用内核多项式方法对其进行矫正, 详情请参阅附录A.8。
- 选择合适的虚时点数目 (ntime 参数, 请参阅第6.38节)。一般而言, ntime 参数越大, 抽样结果越精确, 但是会大大降低计算效率, 因此我们需要做出折衷的选择。此外 ntime 的取值与 beta 参数也有密切的联系。
- 适当进行强制对称化 (isspn 参数与 issun 参数, 请参阅第6.3节与第6.2节)。如果系统的不同轨道之间存在着某种对称性, 那么对它们进行强制对称化有助于提高计算精度。

²²注意: 此方法不是对所有量子杂质求解器组件都适用。使用此方法还需要仔细调整 lemax 参数 (请参阅第6.29节) 或者是 chmax 参数 (请参阅第6.31节), 否则会弄巧成拙。

8.3.4 虚频格林函数与电子自能函数

虚频格林函数 $G(i\omega)$ 与电子自能函数 $\Sigma(i\omega)$ 是十分重要的物理量，应用 iQIST 软件包中的量子杂质求解器组件同样可以直接测量它们的值。那么如何获得精确的测量值呢？请用户遵循下面的规则²³。

- 采用并行模式运行量子杂质求解器（请参阅第3.3节）。iQIST 的量子杂质求解器组件对于计算核心的数目没有任何限制。一般而言，在其它条件相同的情况下，计算核心的数目越多，所获得的计算结果就越精确。
- 增加 Monte Carlo 抽样的次数 (nsweep 参数，请参阅第6.43节)。在其它条件相同的情况下，抽样次数越多，所获得的计算结果就越精确。
- 适当减少抽样间隔 (nmonte 参数，请参阅第6.46节)。在自关联程度得到控制的前提下，减少抽样间隔就相当于增加抽样次数，有利于获得更精确的计算结果。
- 采用 data binning 计算模式 (isbin 参数，请参阅第6.4节)。采用 data binning 计算模式就相当于增加抽样次数十倍，有利于获得更精确的计算结果。
- 采用正交多项式方法测量 $G(i\omega)$ 与 $\Sigma(i\omega)$ (isort 参数，请参阅第6.5节)²⁴。对于具有金属性质的系统而言，应用正交多项式方法能够获得十分平滑的结果。对于具有绝缘体性质的系统而言，正交多项式方法的计算结果不佳，但是可以采用内核多项式方法对其进行矫正，详情请参阅附录A.8。
- 选择合适的虚时点数目 (ntime 参数，请参阅第6.38节)。一般而言，ntime 参数越大，抽样结果越精确，但是会大大降低计算效率，因此我们需要做出折衷的选择。此外 ntime 的取值与 beta 参数也有密切的联系。
- 适当进行强制对称化 (isspn 参数与 issun 参数，请参阅第6.3节与第6.2节)。如果系统的不同轨道之间存在着某种对称性，那么对它们进行强制对称化有助于提高计算精度。

8.3.5 自旋 – 自旋关联函数与轨道 – 轨道关联函数

在本小节中，我们将展示如何使用 Gardenia 组件来计算自旋 – 自旋关联函数以及轨道 – 轨道关联函数。从原理上说，在 DMFT 自洽计算过程中可以一并测量这些关联函数，但是这样做无疑会加重计算负担，并且所获得的关联函数并不是正确值。因此通常的做法是在 DMFT 自洽迭代计算完成后，进行一次单步计算（请参阅第8.3.1小节），专注于测量相关的物理量，这其中自然也包括了关联函数。此前，我们已经在第8.1.1小节中自洽求解了一个单带半满 Hubbard 模型，获得了收敛后的虚频杂化函数。那么

²³下述规则与第8.3.3小节中精确测量 $G(\tau)$ 的规则基本雷同。

²⁴注意：此方法不是对所有量子杂质求解器组件都适用，要求 isort 参数取 4 ~ 6 (仅 Gardenia 组件与 Narcissus 组件支持)。使用此方法还需要仔细调整 lemax 参数（请参阅第6.29节）或者是 chmax 参数（请参阅第6.31节），否则会弄巧成拙。

现在我们将以此为基础进行单步计算，精确测量自旋 – 自旋关联函数与轨道 – 关联函数。具体计算流程如下。

步骤 1: 创建自己的工作目录/工作文件夹

首先创建 `iqist_test` 文件夹，该文件夹可作为用户的练习目录。如果用户在练习上一个案例时已经创建了此文件夹，那么可以略过此操作。

在 `iqist_test` 文件夹下创建子文件夹 `t835`:

```
$ mkdir t835
```

再进入到 `t835` 文件夹中:

```
$ cd t835
```

那么当前所在的位置就是 `/iqist_test/t835`。

步骤 2: 准备 `solver.ctqmc.in`

在进行普通的 DMFT 自治计算时，我们可以选用 `Azalea` 组件。但是由于 `Azalea` 组件不具备计算自旋 – 自旋关联函数以及轨道 – 轨道关联函数的功能，我们必须弃用它而选用具有此功能的 `Gardenia` 组件或者是 `Narcissus` 组件。

在 `/opt/iqist/tutor/t835` 文件夹中，我们已经准备好了 `Gardenia` 组件的输入文件 `solver.ctqmc.in`，请将它复制到当前目录下，例如:

```
$ cp /opt/iqist/tutor/t835/solver.ctqmc.in .
```

在此 `solver.ctqmc.in` 文件中，有如下几个地方需要特别注意:

- 设定单步计算模式，`isscf = 1`
- 无需采用 data binning 模式，`isbin = 1`
- 采用 Legendre 正交多项式方法测量虚时格林函数，`isort = 2`
- 计算自旋 – 自旋关联函数，`isvrt = 2`
- Legendre 正交多项式的最大展开阶数，`lemax = 24`
- Legendre 正交多项式线性网格点数目，`legrd = 20001`
- `nffrq`，`nbfrq` 参数不起作用，可任意设置

其它的输入参数雷同于第8.1.1小节中的设置。

步骤 3: 准备 `solver.hyb.in`

除了 `solver.ctqmc.in` 文件以外，`solver.hyb.in` 也是必须的输入文件，它提供了初始化的杂化函数 $\Delta(i\omega)$ 。在 `/opt/iqist/tutor/t835` 文件夹中，我们已经准备好了 `Gardenia` 组件的输入文件 `solver.hyb.in`，请将它复制到当前目录下，例如:

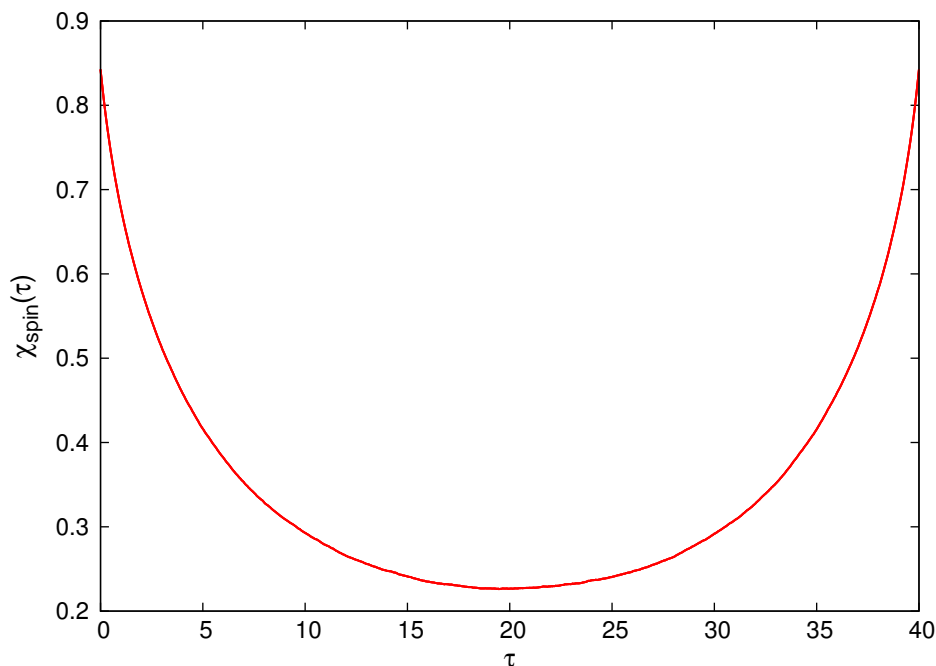


图 8.22: 单带半满 Hubbard 模型的自旋 – 自旋关联函数 $\chi_{\text{spin}}(\tau)$

```
$ cp /opt/iqist/tutor/t835/solver.hyb.in .
```

实际上，用户完全可以不使用我们所提供的 solver.hyb.in 文件。用户在完成例子 t811 后（参阅第8.1.1小节），可以将最后得到杂化函数文件 solver.hyb.dat 重命名为 solver.hyb.in，然后将其拷贝到当前目录下即可，结果应该是完全一样的。虽然此 solver.hyb.in 文件是由 *Azalea* 组件输出的，但是 *Gardenia* 组件可以正确读取它。

步骤 4: 启动 *Gardenia* 组件进行实际计算

在此案例中，我们使用 8 个计算核心运行 *Gardenia* 组件的 ctqmc 程序，用户可以根据现有的计算资源做出适当调整。具体命令如下：

```
$ nohup mpiexec -n 8 ctqmc < /dev/null > out.dat &
```

该计算在大约两分钟之内就能完成。

步骤 5: 数据的分析和处理

计算完成后，首先还是要照例检查 out.dat 文件，确认该次计算有效。然后用 gnuplot 画图比较该次计算得到的虚时格林函数和第8.1.1小节中计算得到的虚时格林函数。它们应该非常吻合，否则就是出现错误了。

接下来我们用 gnuplot 画出自旋 – 自旋关联函数，该函数的数据存储在 solver.schi.dat 文件中，如图8.22所示，用户可以自行比较自己的计算结果。用户还可以利用 *Hibiscus* 组件中的 makechi 程序对 solver.schi.dat 文件进行后处理，通过计算得到有效局域磁矩 M_e ，详情请参阅第7.5节。

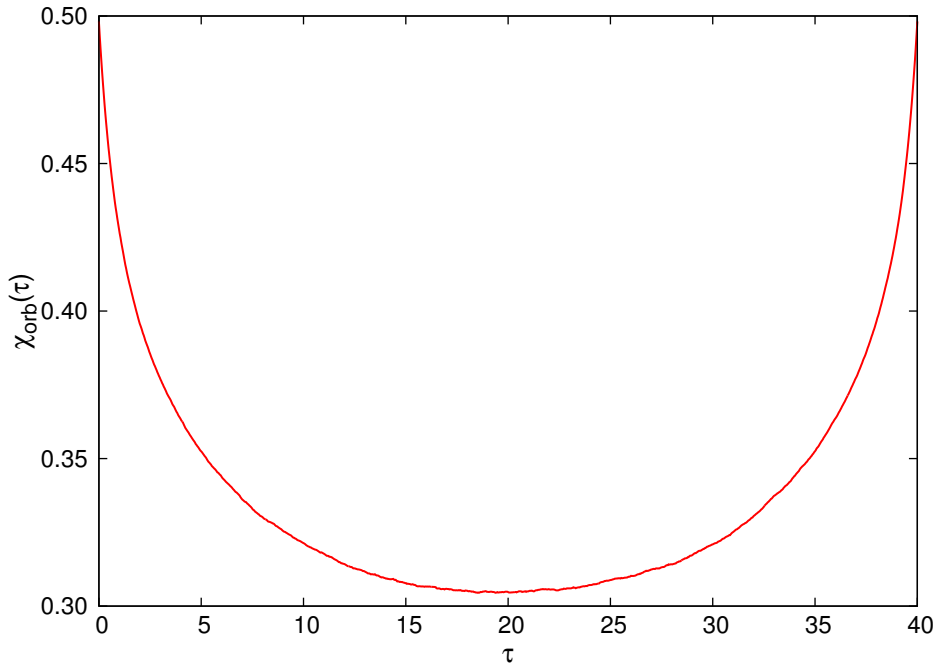


图 8.23: 单带半满 Hubbard 模型的轨道 – 轨道关联函数 $\chi_{\text{orb}}(\tau)$

步骤 6: 轨道 – 轨道关联函数的计算

在上述步骤中只是计算了自旋 – 自旋关联函数，我们也可以通过类似的步骤计算轨道 – 轨道关联函数。请修改上述 `solver.ctqmc.in` 文件，将 `isvrt` 参数的取值设为 3，重新进行计算即可获得轨道 – 轨道关联函数。相关计算结果存储在 `solver.ochi.dat` 文件中，我们可以用 `gnuplot` 软件对其进行绘图，结果如图 8.23 所示。

8.3.6 双粒子格林函数与顶角函数

在本小节中，我们将展示如何使用 `Gardenia` 组件来计算双粒子格林函数 $\chi(\omega, \omega', \nu)$ 以及顶角函数 $\mathcal{F}(\omega, \omega', \nu)$ ，这两种函数在 DMFT 理论的非局域扩展中有着十分重要的应用^{50–52}。双粒子格林函数以及顶角函数的计算十分费时，我们通常是在 DMFT 自治计算结束后，再开启单步计算模式（请参阅第 8.3.1 小节），对它们进行详细的计算。此前，我们已经在第 8.1.1 小节中自治求解了一个单带半满 Hubbard 模型，获得了收敛后的虚频杂化函数。那么现在我们将以此为基础进行单步计算，精确测量双粒子格林函数与顶角函数。具体计算流程如下²⁵。

步骤 1: 创建自己的工作目录/工作文件夹

首先创建 `iqist_test` 文件夹，该文件夹可作为用户的练习目录。如果用户在练习上一个案例时已经创建了此文件夹，那么可以略过此操作。

²⁵ 此计算流程与第 8.3.5 小节中介绍的自旋 – 自旋关联函数以及轨道 – 轨道关联函数的计算流程十分类似。

在 `iqist_test` 文件夹下创建子文件夹 `t836`:

```
$ mkdir t836
```

再进入到 `t836` 文件夹中:

```
$ cd t836
```

那么当前所在的位置就是 `/iqist_test/t836`。

步骤 2: 准备 `solver.ctqmc.in`

在进行普通的 DMFT 自洽计算时, 我们可以选用 *Azalea* 组件。但是由于 *Azalea* 组件不具备计算双粒子格林函数以及顶角函数的功能, 我们必须弃用它而选用具有此功能的 *Gardenia* 组件或者是 *Narcissus* 组件。在本案例中, 我们选用的是 *Gardenia* 组件, 如果需要考虑动态屏蔽效应对双粒子格林函数以及顶角函数的影响, 那么必须使用 *Narcissus* 组件。

在 `/opt/iqist/tutor/t836` 文件夹中, 我们已经准备好了 *Gardenia* 组件的输入文件 `solver.ctqmc.in`, 请将它复制到当前目录下, 例如:

```
$ cp /opt/iqist/tutor/t836/solver.ctqmc.in .
```

在此 `solver.ctqmc.in` 文件中, 有如下几个地方需要特别注意:

- 设定单步计算模式, `isscf = 1`
- 无需采用 data binning 模式, `isbin = 1`
- 采用 Legendre 正交多项式方法测量虚时格林函数, `isort = 2`
- 计算双粒子格林函数以及顶角函数, `isvrt = 4`
- Legendre 正交多项式的最大展开阶数, `lemax = 24`
- Legendre 正交多项式线性网格点数目, `legrd = 20001`
- 设置费米频率点的数目, `nffrq = 32`
- 设置玻色频率点的数目, `nbfrq = 2`

其中, `isvrt` 参数可以设置为 4, 也可以设置为 5, 这两种设置都能够得到双粒子格林函数以及顶角函数。只不过前者的输出数据存储在 `solver.twop.dat` 文件中, 而后者的输出数据存储在 `solver.vrtx.dat` 文件中。后者所用的算法更为精确, 计算量也更大。关于 `isvrt` 参数的详情, 请参阅第6.6节。此外, 利用正交多项式方法可以有效地提升双粒子格林函数以及顶角函数的测量精度, 因此在本案例中, 我们激活了 Legendre 正交多项式方法 (`isort` 参数, 请参阅第6.5节)。为了提升计算效率, 减少计算时间, 费米频率点的数目 (`nffrq` 参数, 请参阅第6.35节) 以及玻色频率点的数目 (`nbfrq` 参数, 请参阅第6.36节) 都设得比较小。如果要获得更精确的结果, 请用户酌情增加这两个参数的值。其它的输入参数雷同于第8.3.5小节中的设置。

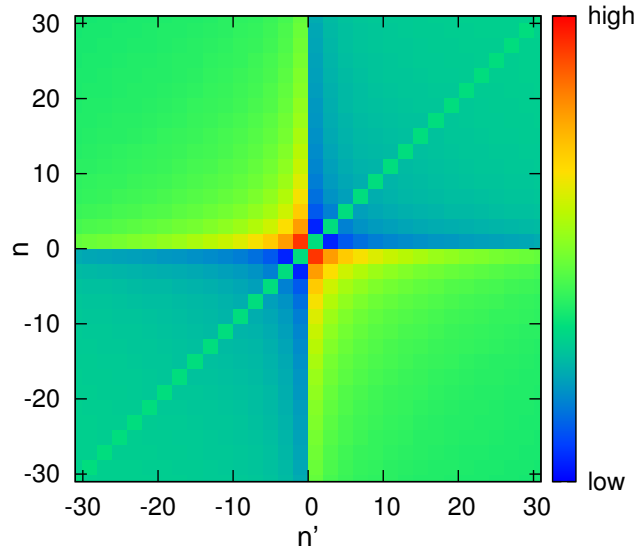


图 8.24: 单带半满 Hubbard 模型的总的双粒子格林函数 $\chi_{\text{tot}}^{mn}(\omega, \omega', \nu)$ 。high = 120, low = -60。

步骤 3: 准备 solver.hyb.in

除了 solver.ctqmc.in 文件以外, solver.hyb.in 也是必须的输入文件, 它提供了初始化的杂化函数 $\Delta(i\omega)$ 。在 /opt/iqist/tutor/t836 文件夹中, 我们已经准备好了 *Gardenia* 组件的输入文件 solver.hyb.in, 请将它复制到当前目录下, 例如:

```
$ cp /opt/iqist/tutor/t836/solver.hyb.in .
```

实际上, 用户完全可以不使用我们所提供的 solver.hyb.in 文件。用户在完成例子 t811 后 (参阅第 8.1.1 小节), 可以将最后得到杂化函数文件 solver.hyb.dat 重命名为 solver.hyb.in, 然后将其拷贝到当前目录下即可, 结果应该是完全一样的。虽然此 solver.hyb.in 文件是由 *Azalea* 组件输出的, 但是 *Gardenia* 组件可以正确读取它。

步骤 4: 启动 *Gardenia* 组件进行实际计算

在此案例中, 我们使用 8 个计算核心运行 *Gardenia* 组件的 ctqmc 程序, 用户可以根据现有的计算资源做出适当调整。具体命令如下:

```
$ nohup mpiexec -n 8 ctqmc < /dev/null > out.dat &
```

由于 nfrq 参数以及 nbfrq 参数取得比较小, 计算量不大, 该计算在大约两分钟之内就能够完成。

步骤 5: 数据的分析和处理

计算完成后, 首先还是要照例检查 out.dat 文件, 确认该次计算有效。然后用 gnuplot 画图比较该次计算得到的虚时格林函数和第 8.1.1 小节中计算得到的虚时格林函数 (请参阅图 8.1)。它们应该非常吻合,

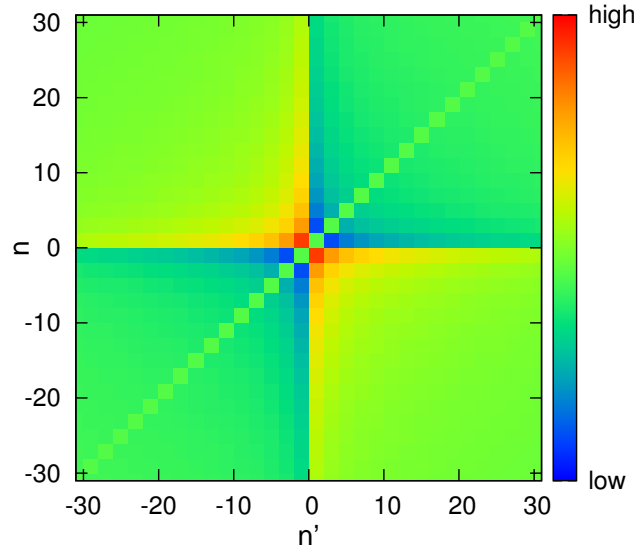


图 8.25: 单带半满 Hubbard 模型的可约双粒子格林函数 $\chi_0^{mn}(\omega, \omega', \nu)$ 。high = 100, low = -80。

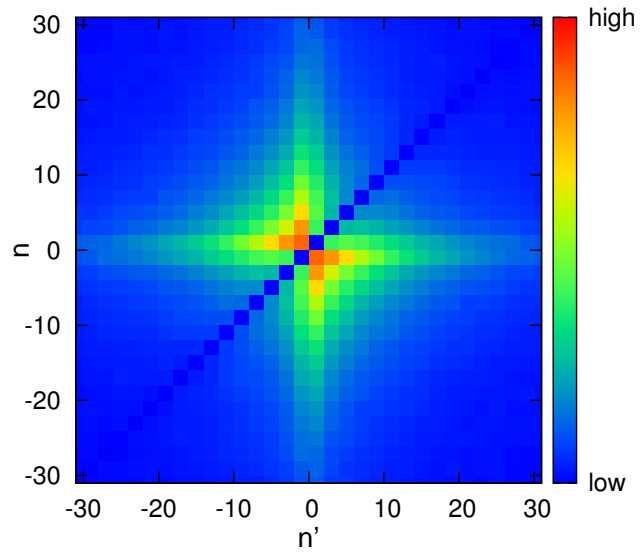


图 8.26: 单带半满 Hubbard 模型的不可约双粒子格林函数 $\chi_{\text{irr}}^{mn}(\omega, \omega', \nu)$ 。high = 25, low = 0。

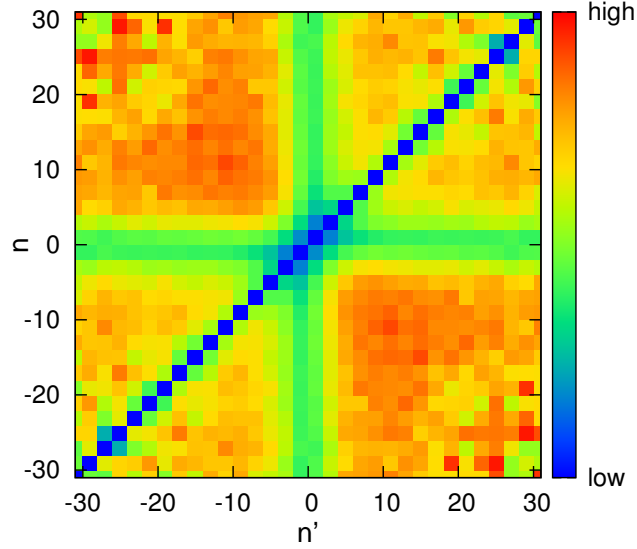


图 8.27: 单带半满 Hubbard 模型的全顶角函数 $\mathcal{F}^{mn}(\omega, \omega', \nu)$ 。high = 25, low = 0。

否则就是出现错误了。

接下来我们关注双粒子格林函数以及顶角函数的计算结果。由于 isvrt 参数设置为 4，那么相应的输出文件为 solver.twop.dat。如果 isvrt 参数设置为 5，那么输出文件为 solver.vrtx.dat。这两个文件的数据格式是完全一样的，详情请参阅第 5.2 节。在 solver.twop.dat (solver.vrtx.dat) 文件中，有很多数据 block，每个玻色频率点对应一个 block。在本案例中，由于 nbfrq = 2，因此文件中一共有 2 个 block。在绘图时，我们首先需要将 block 分离，然后对每一个 block 单独绘图。下面的图 8.24-图 8.27 分别绘制了总的双粒子格林函数 $\chi_{\text{tot}}^{mn}(\omega, \omega', \nu)$ 、可约双粒子格林函数 $\chi_0^{mn}(\omega, \omega', \nu)$ 、不可约双粒子格林函数 $\chi_{\text{irr}}^{mn}(\omega, \omega', \nu)$ 以及全顶角函数 $\mathcal{F}^{mn}(\omega, \omega', \nu)$ 。其中 $\nu = 0$ ，亦即图中的数据为第一个 block 中的数据。用户可以自行绘图，以检验计算结果是否准确。

最后，用户可以将 isvrt 参数设为 5，重新进行上述计算，并比较前后两次的计算结果，判明孰优孰劣。此外，用户还可以增加 nffrq、nbfrq 等参数的值，看看计算结果是否有明显的改变。特别值得一提的是，isort 参数的取值对于相关计算的精度有十分显著的影响，这一切就留待用户去自行摸索了，此处不再赘述。

8.4 进阶应用 III: 后处理步骤

本节专注于后处理步骤，介绍如何使用 *Hibiscus* 组件对 $G(\tau)$ 和 $\Sigma(i\omega)$ 进行解析延拓，分别获取电子谱函数 $A(\omega)$ 和实频电子自能 $\Sigma(\omega)$ 。

8.4.1 虚时格林函数的解析延拓

在本小节中，我们将演示在第8.3.2小节的基础上，如何使用 *Hibiscus* /hibiscus-entropy 组件对虚时格林函数 $G(\tau)$ 进行解析延拓，得到谱函数 $A(\omega)$ 。

步骤 1: 创建自己的工作目录/工作文件夹

首先创建 `iqist_test` 文件夹，该文件夹可作为用户的练习目录。如果用户在练习上一个案例时已经创建了此文件夹，那么可以略过此操作。

在 `iqist_test` 文件夹下创建子文件夹 `t841`:

```
$ mkdir t841
```

步骤 2: 准备 tau.grn.dat

`hibiscus-entropy` 组件需要的数据输入文件为 `tau.grn.dat`，该文件包含了处理过的 $G(\tau)$ 数据。我们首先使用 *Hibiscus* /hibiscus-toolbox 组件下的 `maketau` 程序产生 `tau.grn.dat` 文件。请进入在第8.3.2小节中所建立的 `t832` 文件夹:

```
$ cd t832
```

请判明在 `t832` 目录下是否存在着 100 个 data bins(亦即 100 个 `solver.green.bin.*` 文件)，如果答案是否定的，那么请按照第8.3.2小节中的介绍首先执行例子 `t832`；如果答案是肯定的，那么可以调用 `maketau` 程序进行处理。`maketau` 程序的可执行文件为 `mtau.x`(详情请参阅第7.5节)，运行 `mtau.x` 的命令如下:

```
$ mtau.x
```

用户只要根据程序提示一步一步输入控制参数就可以了。下面是一个运行例子:

```
1 MIAU
2 making tau-dependent imaginary time green's function
3 version: 2011.08.18T
4
5 >>> number of bands (default = 1):
6 >>> 1
7
8 >>> number of time slice (default = 129 or 1024):
9 >>> 1024
10
```

```

11 >>> number of data bins (default = 1):
12 >>> 100
13
14 >>> file type generated by quantum impurity solver (default = 1):
15 ctqmc: 1
16 hfqmc: 2
17 ctqmc: 3 (bin mode)
18 hfqmc: 4 (bin mode)
19 >>> 3
20
21 >>> inversion of temperature (default = 10.0):
22 >>> 40
23
24 >>> reading solver.green.bin ... 1
25 >>> status: OK
26
27 >>> reading solver.green.bin ... 2
28 >>> status: OK
29
30 .....

```

对于 maketau 程序的主要控制参数，我们简单介绍如下：

- 第 6 行，总的能带数，本例中只有 1 个能带。
- 第 9 行，总的虚时片数，本例中为 1024
- 第 12 行，solver.green.bin* 文件数目，本例中为 100
- 第 19 行，杂质求解器类型，本例中为 3
- 第 22 行，反温度 β ，本例中为 40

当 maketau 程序运行完毕后，如果一切正常，那么即可获得 tau.grn.dat 文件。请不要手动修改 tau.grn.dat 文件，而是把它直接拷贝到 t841 文件夹：

```
$ cp tau.grn.dat ../t841
```

然后，我们进入到 t841 文件夹：

```
$ cd ../t841
```

那么当前所在目录就是 t841。

步骤 3: 准备 entropy.in

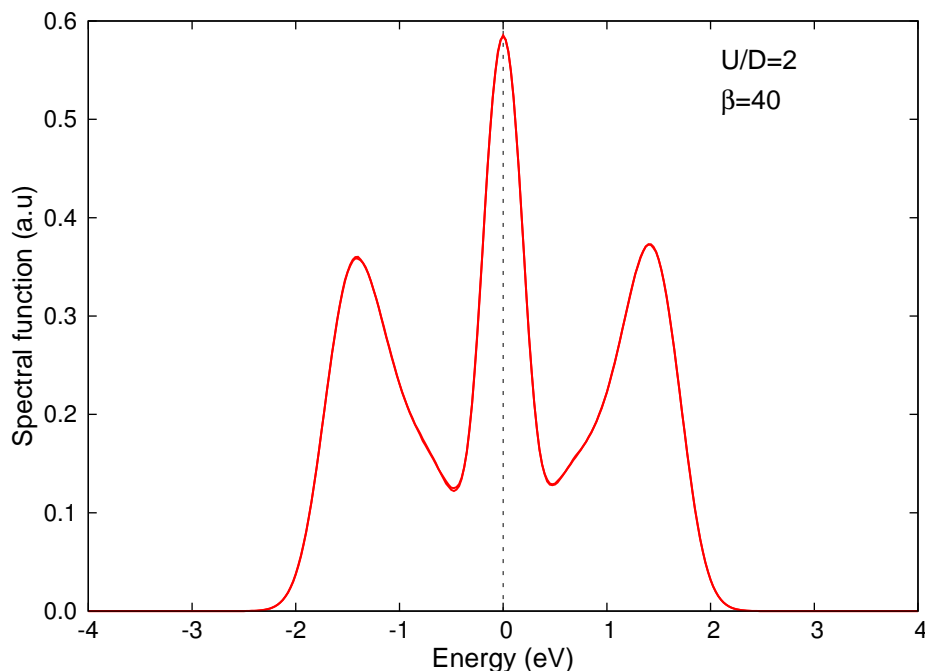


图 8.28: 单带半满 Hubbard 模型的谱函数 $A(\omega)$

hibiscus-entropy 组件所需要的控制文件为 entropy.in。如果没有 entropy.in，那么 hibiscus-entropy 组件将无法正常运行。在 /opt/iqist/tutor/t841 文件夹中，我们已经准备好了 hibiscus-entropy 组件的输入文件 entropy.in，请将它复制到当前目录下，例如：

```
$ cp /opt/iqist/tutor/t841/entropy.in .
```

关于该输入文件的内容，请用户参考第7.2节中关于该文件的详细介绍，此处不再赘述。

步骤 4: 启动 hibiscus-entropy 组件进行实际计算

hibiscus-entropy 组件的可执行程序名为 entropy，这是一个串行程序，不支持 MPI 并行，那么执行它的具体命令如下：

```
$ nohup entropy </dev/null> out.dat &
```

程序很快运行完毕，耗时约 1 分钟。

步骤 5: 数据分析和处理

entropy 程序运行完毕后，所产生的谱函数 $A(\omega)$ 储存于 mem.dos.dat 文件中，用户可以使用 gnuplot 软件画出谱函数：

```
$ gnuplot> plot "mem.dos.dat" u 1:2 w lp
```

具体结果如图8.28所示。在本案例中，Coulomb 相互作用强度还比较小，因而谱函数显示出典型的三峰结构。上下 Hubbard 带十分对称，这意味着系统具有粒子 - 空穴对称性，处于半填满状态。

hibiscus-entropy 组件基于最大熵方法⁶⁷，在 iQIST 软件包中，我们还提供基于随机解析延拓方法⁶⁸

的 `hibiscus-stochastic` 组件。利用 `hibiscus-stochastic` 组件也可以对 $G(\tau)$ 进行解析延拓，得到电子谱函数 $A(\omega)$ (详情请参阅第7.3节)，用户不妨自行一试。

8.4.2 虚频电子自能函数的解析延拓

在本小节中，我们将演示在第8.1.1小节的基础上，如何使用 `Hibiscus` /`hibiscus-swing` 组件对虚频电子自能函数 $\Sigma(i\omega)$ 进行解析延拓，得到实频率上的自能信息 $\Sigma(\omega)$ 。 $\Sigma(\omega)$ 对于 LDA + DMFT 计算而言十分有用，利用它可以进一步计算 $A(k, \omega)$ 、光电导等物理量。

为了获得 `Hibiscus` /`hibiscus-swing` 组件所需要的数据输入文件 `std.sgm.dat`，我们首先需要得到虚频自能函数的多个样本，然后再利用 `Hibiscus` /`hibiscus-toolbox` 组件里的 `makestd` 程序处理这些样本，求得平均值和方差值等信息，生成 `std.sgm.dat` 文件，最后才用 `Hibiscus` /`hibiscus-swing` 组件对其进行处理，获得最终的实频电子自能。下面是详细的计算流程。

步骤 1: 创建自己的工作目录/工作文件夹

首先创建 `iqist_test` 文件夹，该文件夹可作为用户的练习目录。如果用户在练习上一个案例时已经创建了此文件夹，那么可以略过此操作。

在 `iqist_test` 文件夹下创建子文件夹 `t842`:

```
$ mkdir t842
```

进入到 `t842` 文件夹中:

```
$ cd t842
```

那么当前所在的目录就是 `t842`。在该文件夹中建立子文件夹 `sigm1`，然后拷贝我们预先准备好的输入文件 `solver.ctqmc.in` 到 `sigm1` 中:

```
$ cp /opt/iqist/tutor/t842/solver.ctqmc.in sigm1/
```

然后拷贝 `t811` 文件夹中的 `solver.hyb.dat` 文件到 `sigm1` 文件夹中，并将其重命名为 `solver.hyb.in`:

```
$ cp ../t811/solver.hyb.dat sigm1/solver.hyb.in
```

如果在 `t811` 文件夹中不存在 `solver.hyb.dat` 文件，这说明用户还没有完成 `t811` 这个案例。请用户按照第8.1.1小节中的指示，一步一步完成案例 `t811`，产生正确的 `solver.hyb.dat` 文件，再执行上述拷贝/重命名操作。

接下来请产生 `sigm1` 文件夹的 9 个拷贝，命名为 `sigm*`，* 从 2 变化到 10，例如：

```
$ cp -r sigm1 sigm2
```

在 `t842` 文件夹下建立文件夹 `sigm_std`，用作预处理程序 `makestd` 的工作目录:

```
$ mkdir sigm_std
```

在 `t842` 文件夹下建立文件夹 `sigm_swing`，作为 `Hibiscus` /`hibiscus-swing` 组件的工作目录:

```
$ mkdir sigm_swing
```

步骤 2:启动Azalea 组件进行计算

依次进入每个 sigm^* 目录，启动Azalea 组件进行计算，例如：

```
$ nohup mpiexec -n 8 ctqmc < /dev/null > out.dat &
```

实际上，此刻我们所做的即是第8.3.1小节里所讲的单步计算模式。由于Azalea 组件的高效率，10 次计算很快就能执行完毕。

步骤 3:使用 makestd 程序对自能函数进行预处理

分别将 sigm^* 文件夹中的 solver.sgm.dat 文件依次拷贝到 sigm_std 文件夹，并将其重命名为 solver.sgm.dat.* 文件，例如：

```
$ cp sigm1/solver.sgm.dat sigm_std/solver.sgm.dat.1
```

然后进入到 sigm_std 文件夹中，启动 makestd 程序进行自能预处理。makestd 程序的可执行程序名为 mstd.x，执行命令如下：

```
$ mstd.x
```

用户只需根据程序提示输入控制参数就可以了，具体如下所示：

```

1  MSTD
2  making average and standard deviation for self-energy function
3  version: 2011.08.18T
4
5  >>> number of bands (default = 1):
6  >>> 1
7
8  >>> number of matsubara frequency points (default = 8193):
9  >>> 8193
10
11 >>> number of data bins (default = 1):
12 >>> 10
13
14 >>> reading solver.sgm.dat ...    1
15 >>> status: OK
16
17 >>> reading solver.sgm.dat ...    2
18 >>> status: OK
19
20 .....
```

对于 makestd 程序的主要控制参数，我们简单介绍如下：

- 第 6 行，总的能带数，本例中只有 1 个能带

- 第 9 行，虚频率点数，本例中为 8193
- 第 12 行，solver.sgm.dat.* 文件数目，本例中为 10

当 makestd 程序运行完毕后，如果一切正常，那么即可获得 std.sgm.dat 文件。请不要手动修改 std.sgm.dat 文件，而是把它直接拷贝到 sigm_swing 文件夹中：

```
$ cp std.sgm.dat ../sigm_swing
```

步骤 4: 使用 hibiscus-swing 组件进行自能解析延拓

进入 sigm_swing 文件夹，将我们预先准备好的 metal.sh 脚本文件拷贝到当前目录：

```
$ cp /opt/iqist/tutor/t842/metal.sh .
```

该文件是 *Hibiscus* /hibiscus-swing 组件的启动脚本。由于在本例中，该体系明显处于金属相，因此我们使用 metal.sh 脚本来启动 hibiscus-swing 组件，反之则应该使用 insulator.sh 脚本。对于该脚本，我们需要注意以下几点：

- 请用户调整 exec 变量，指向 hibiscus-swing 组件所在目录。
- para 变量中设置的参数请保持不变。

更多的关于 hibiscus-swing 组件的信息，请参考第 7.4 节中的介绍，此处不再赘述。

下面请运行脚本 metal.sh 进行解析延拓操作，命令示例如下：

```
$ ./metal.sh
```

该程序很快就会运行完毕，并且输出很多数据文件。

步骤 5: 数据分析和处理

Hibiscus /hibiscus-swing 组件在运行过程中将会产生许多输出文件，具体信息请参考第 7.4 节中的介绍。此刻我们只须关注 sigr_linear.out 文件，使用 gnuplot 软件画出实频电子自能的实部：

```
$ gnuplot> plot [-1:1] "sigr_linear.out" u 2:3 w lp
```

如图 8.29 所示，实频自能的实部在费米能级附近具有很好的线性行为，这表示该系统具有很好的费米液体行为。该系统的谱函数即是第 8.4.1 小节中所展示的图 8.28，结合谱函数更能明显看出该系统符合费米液体理论的描述。

再使用 gnuplot 软件画出实频电子自能的虚部：

```
$ gnuplot> plot [-1:1] "sigr_linear.out" u 2:4 w lp
```

如图 8.30 所示，实频自能的虚部在费米能级附近趋于 0，这也是费米液体金属的典型行为。

8.5 *iQIST* 实战应用

在前面的章节中我们已经详细讲述了 *iQIST* 软件包里各个组件的使用方法，如果用户是按照教程一步步操作下来的话，那么相信用户已经熟练掌握了 *iQIST* 软件包的各种应用技能。本章的前几节展示的

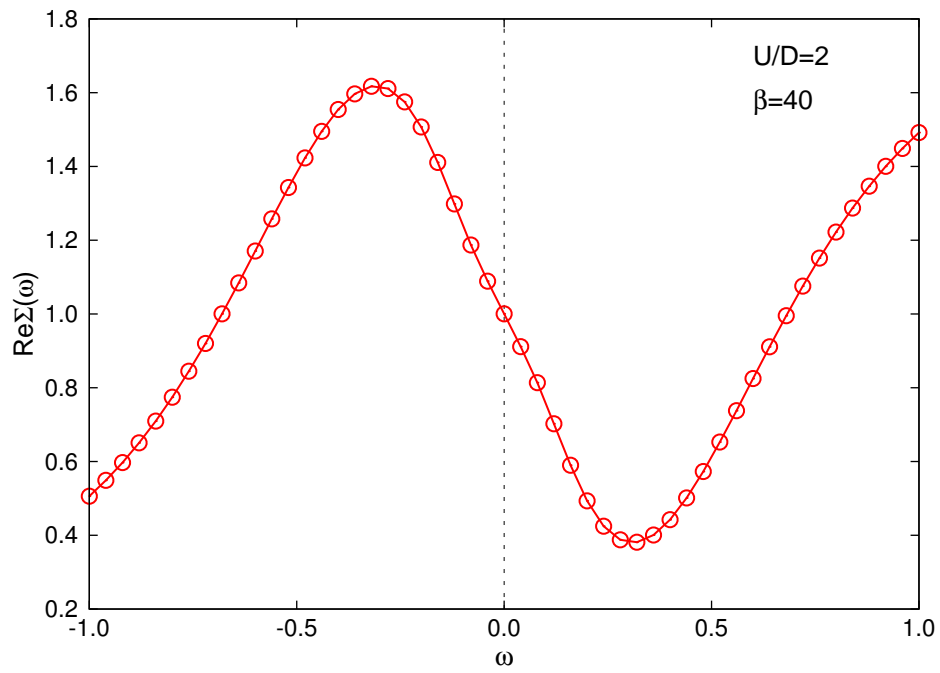


图 8.29: 单带半满 Hubbard 模型实频电子自能的实部 $\Re\Sigma(\omega)$

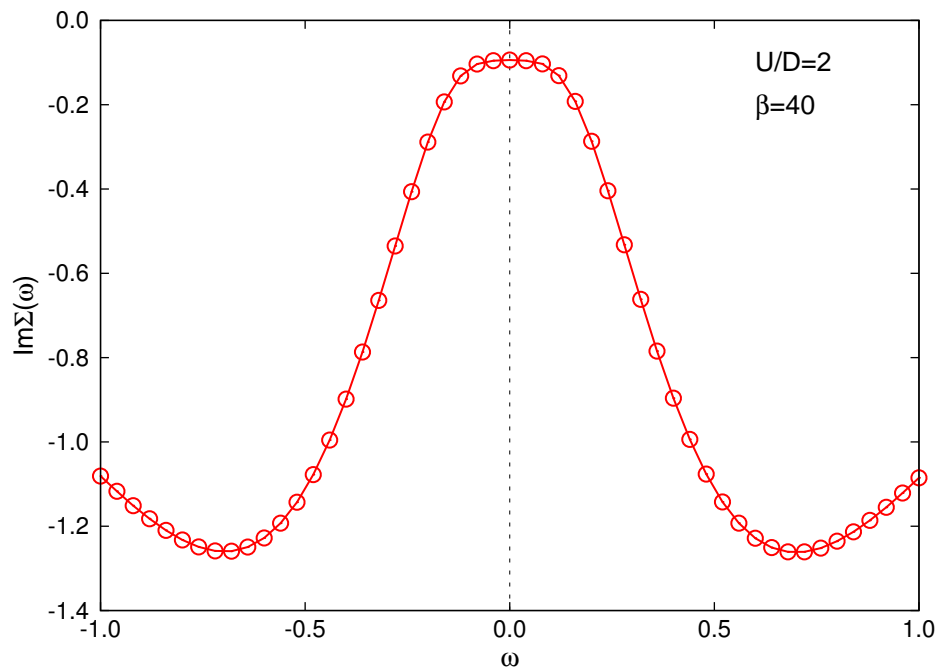


图 8.30: 单带半满 Hubbard 模型实频电子自能的虚部 $\Im\Sigma(\omega)$

都是相对比较简单案例，那么在这一节中，我们将列举两个 *iQIST* 软件包实战应用的综合案例。第一个例子是两带 Hubbard 模型中掺杂和晶体场劈裂相互作用导致的具有轨道选择性的 Mott 相变，第二个例子是三带 Anderson 杂质模型中的轨道 Kondo 效应和自旋 Kondo 效应。在本节中，我们首先简单叙述物理模型和计算细节，然后展示计算结果。至于详细的计算过程 and 数据分析、作图等步骤，请用户独立完成，以此来检验用户是否熟练掌握了前面教程的内容。下面我们就开始 *iQIST* 软件包的实战应用。

8.5.1 两带 Hubbard 模型中掺杂与晶体场相互作用导致的轨道选择性 Mott 相变

为了解释 $\text{Ca}_{2-x}\text{Sr}_x\text{RuO}_4$ 材料中导带电子的混合性质，人们提出了所谓的轨道选择性 Mott 相变的概念⁷²。在该材料中，一部分导带电子表现出巡游性质（具有金属性的电导），而另外一部分导带电子表现出局域性质（具有居里 - 外斯特特性的磁化系数）。经过长期的研究，人们提出了许多导致轨道选择性 Mott 相变的机制^{37,73-85}，请用户参阅相关文献。

在本小节中，我们将研究在两带 Hubbard 模型中，在晶体场劈裂存在的情况下，由于掺杂所导致的轨道选择性 Mott 相变。我们将主要计算该系统随着晶体场劈裂和电子掺杂变化的相图。在初始的时候，该系统总的平均电子占据数为 1，通过调节 Coulomb 相互作用的强度使其处于金属 - 绝缘体相变的临界区域，但是依然处在金属相区域。然后，我们对该系统进行电子掺杂，并且调节晶体场劈裂的大小，我们发现当电子掺杂大于一定量之后，在一定晶体场范围内，系统进入了具有轨道选择性的 Mott 相 (OSMP) 区域，即能级较低的那个轨道（半满）从金属相变为 Mott 绝缘相，而能级较高的那个轨道依然保持为金属相。通过分析轨道占据数和局域磁矩，我们得出这样的结论：在 Hund 交换作用存在的情况下，掺杂电子促使系统形成较大的局域磁矩，而局域磁矩的存在会散射电子，从而导致有效的电子相互作用强度增加，促使能级较低的那个半满轨道发生了从金属到 Mott 绝缘体的相变。

1. 模型哈密顿量

我们研究的两带 Hubbard 模型的哈密顿量如下：

$$H = -t \sum_{\langle i,j \rangle, a\sigma} d_{ia\sigma}^\dagger d_{ja\sigma} + \sum_{i,a\sigma} (-\mu + \Delta_a) n_{a\sigma} + U \sum_{i,a} n_{a\uparrow} n_{a\downarrow} + \sum_{i,a>b,\sigma\sigma'} (U' - \delta_{\sigma\sigma'} J) n_{a\sigma} n_{b\sigma'} \quad (8.7)$$

其中， $d^\dagger(d)$ 为 d 电子产生（消灭）算符， a, b 为能带指标 ($a, b = 1 \sim 2$)， σ, σ' 为自旋指标， $n_{a\sigma} = d_{a\sigma}^\dagger d_{a\sigma}$ 为占据数算符， μ 为化学势， Δ 为杂质能级。 $U(U')$ 分别为轨道间（轨道内）Coulomb 相互作用， J 为 Hund 交换相互作用，我们采取如下限制 $U = U' + 2J$ 。

2. 计算细节

由于上述哈密顿量具有密度 - 密度相互作用的形式（不包括自旋翻转和对跃迁项，请参阅公式 (8.2)），我们将使用 *iQIST* 软件包的 *Azalea* 组件计算其相图。我们还会使用 *Gardenia* 组件计算系统的自旋 - 自旋关联函数，从而得到局域磁矩。

计算要点如下：

- 使用半圆态密度，半能带宽度设为 $D = 2t = 1.0$ eV，保持两个能带宽度一致²⁶
- 反温度设为 $\beta = 40$ ，这个温度相当于室温
- 通过改变杂质能级 Δ_α 来设置晶体场劈裂的大小，例如： $\Delta_1 = +1.0$ eV， $\Delta_2 = -1.0$ eV，则晶体场劈裂的大小为 $\Delta_1 - \Delta_2 = 2.0$ eV，请在文件 solver.eimp.in 中设置杂质能级。在本案例中，我们约定杂质能级较高的那个能带为”能带 1”，杂质能级较低的那个能带为”能带 2”
- 通过调节化学势的办法来调节电子掺杂量，具体方法请参考附录A.9
- 在计算局域磁矩时，请参考第8.3.5小节所介绍的办法。首先计算自旋 - 自旋关联函数，然后使用 *Hibiscus* /hibiscus-toolbox 组件里的 makechi 程序计算局域磁矩

3. 计算结果及分析

下面简单展示我们的计算结果。

图8.31是关于总占据数 N 与晶体场劈裂 Δ 的相图，上图展示的是参数为 $U = 4.90$ eV， $J = 0.25U$ 的结果，下图展示的是参数为 $U = 4.90$ eV， $J = 0.20U$ 的结果。从图中可以看出：当掺杂较小的时候，两个能带都保持为金属相，即图中的粉色区域；当掺杂大于某个临界值时，能带 2 发生金属 - Mott 绝缘体转变，而能带 1 依然保持为金属相，亦即此系统发生了具有轨道选择性的金属 - Mott 绝缘体转变。如图中的橙色区域所示，我们称此区域为具有轨道选择性的 Mott 相 (OSMP)，掺杂越大 OSMP 区域将变得越大。对比上下两图可以发现，较大的 Hund 交换作用 J 有利于形成 OSMP。

图8.32是固定总占据数 N 时各个能带占据数随着晶体场劈裂的变化情况。从图中可以发现，进入 OSMP 区域后能带 2 的占据数为半满，剩下的电子占据能带 1。

图8.33是总占据数 N 固定时局域磁矩随着晶体场劈裂的变化情况。从图中我们可以看出，较大的掺杂导致系统形成了较大的局域磁矩，而较大的局域磁矩的存在是导致体系发生具有轨道选择性的 Mott 转变的根源。

8.5.2 三带 Anderson 杂质模型中的轨道 Kondo 效应和自旋 Kondo 效应

对于同时具有轨道磁矩和自旋磁矩的磁性杂质，多带 Anderson 杂质模型是一个很好的模型。人们发现轨道 Kondo 效应和自旋 Kondo 效应在此模型中能同时存在^{38,86}。在本小节中，我们将简单研究三带 Anderson 杂质模型中的轨道 Kondo 效应和自旋 Kondo 效应。在该模型中，当平均电子数为 $N = 2$ 或者 4 时，系统具有 $L = S = 1$ 的总轨道和自旋角动量。当温度降低时，轨道和自旋磁矩将渐渐地被 bath 里的自由电子屏蔽。我们使用 *Begonia* 组件系统计算了该模型杂质部分的轨道磁化系数和自旋磁化系数。根据这些磁化系数，我们发现轨道磁化和自旋磁化在温度降低时趋于饱和，发生 Kondo 转变，并且轨道的 Kondo 温度比自旋的 Kondo 温度要高一个数量级。

1. 模型哈密顿量

²⁶亦即采用 bethe 晶格，part 参数取为 0.5。

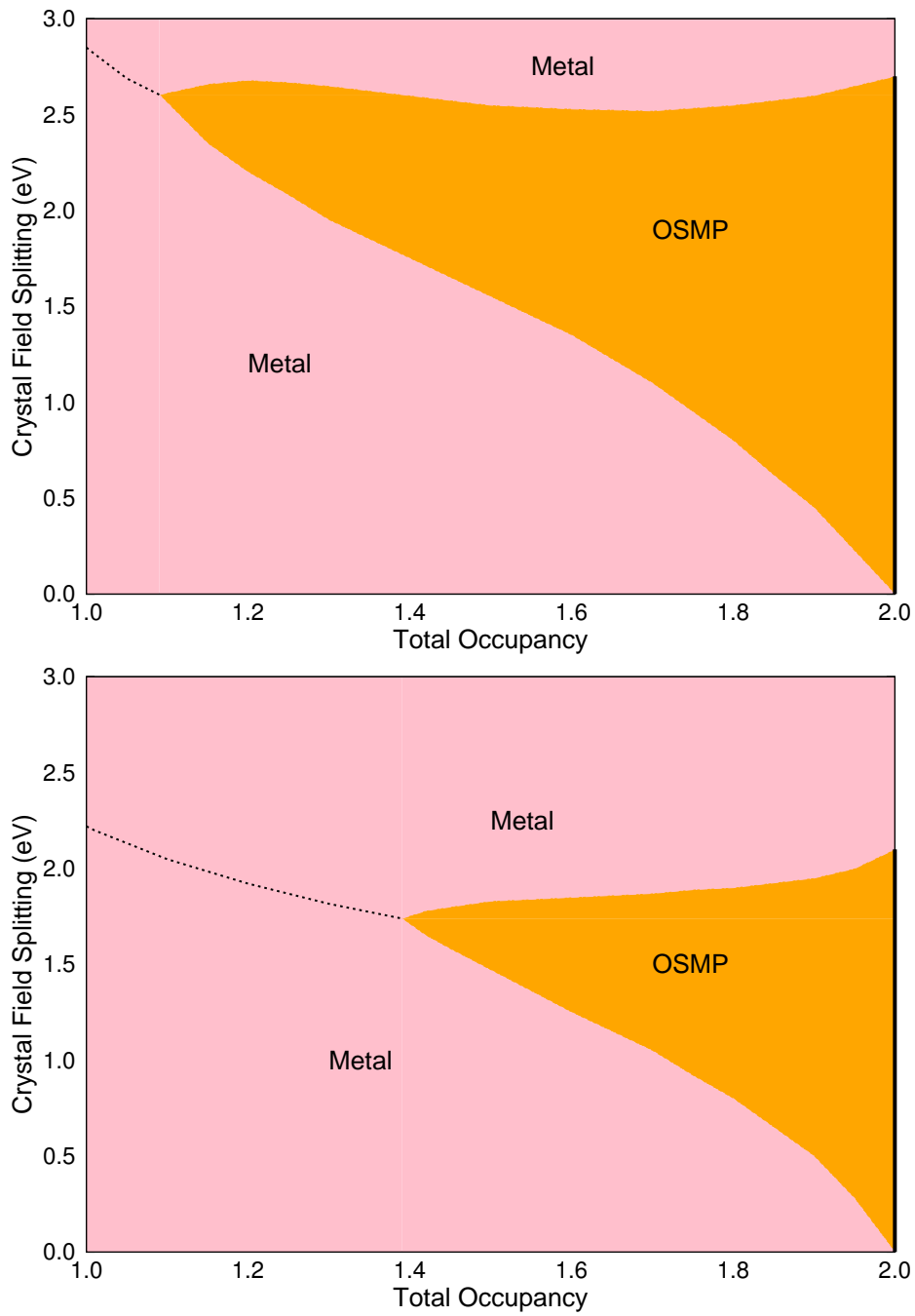


图 8.31: 总占据数 N 与晶体场 Δ 的相图。上图: $U = 4.90 \text{ eV}$, $J = 0.25U$ 。下图: $U = 4.90 \text{ eV}$, $J = 0.20U$ 。在粉色区域里两个能带都是金属相; 在橙色区域里能带 2 半满, 处于 Mott 绝缘相, 而能带 1 处于金属相。虚线代表能带 2 处于半满占据状态。

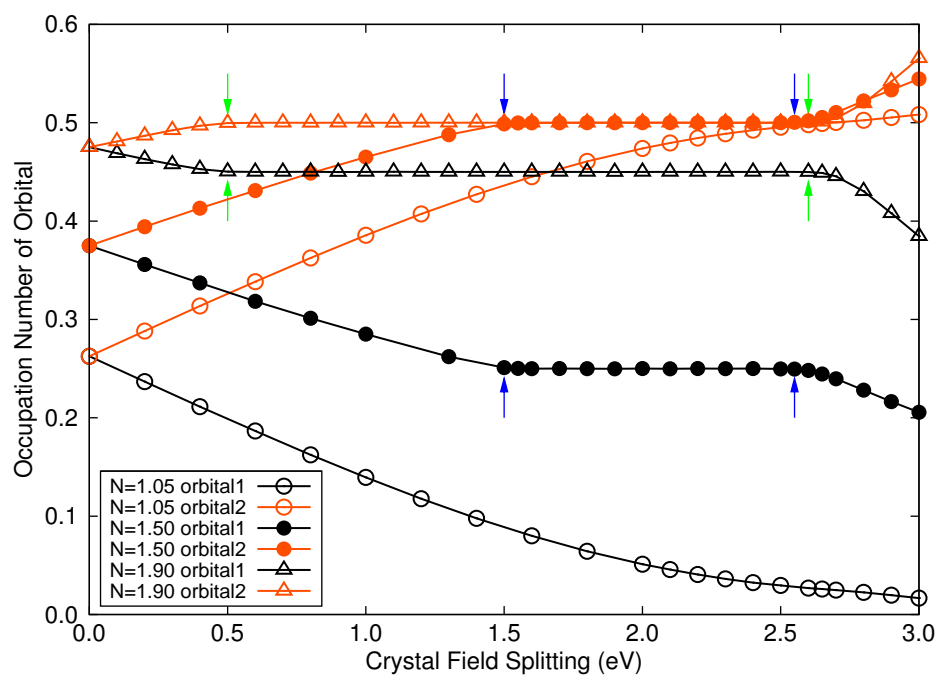


图 8.32: 能带的占据数随着晶体场变化的情况

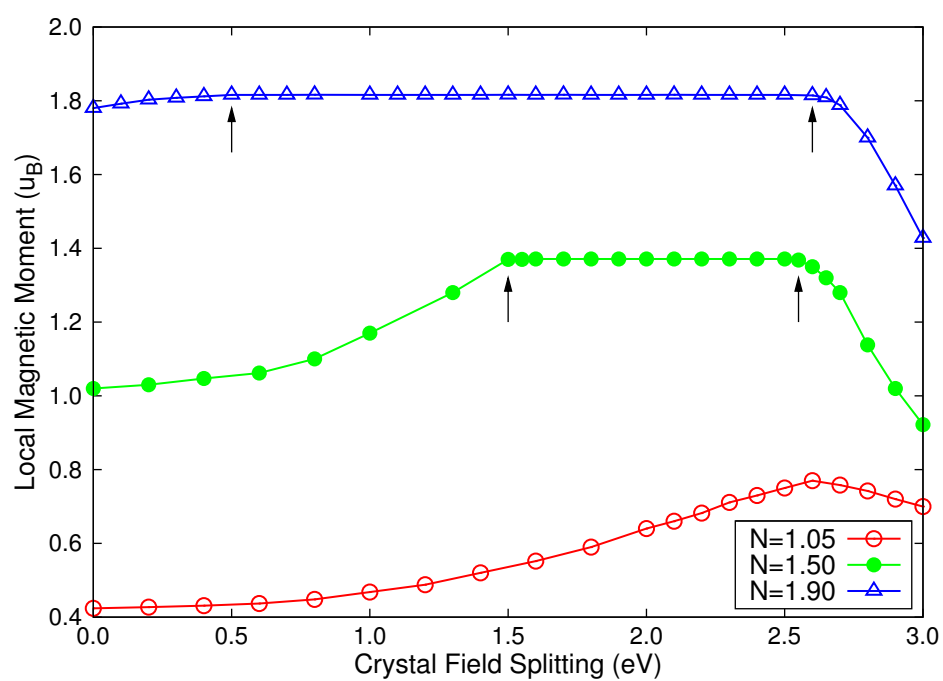


图 8.33: 局域磁矩随着晶体场变化的情况

该三带 Anderson 杂质模型的局域哈密顿量如下所示:

$$\begin{aligned}
 H_{loc} = & - \sum_{a\sigma} (\mu - \Delta_a) n_{a\sigma} + \sum_a U n_{a\uparrow} n_{a\downarrow} \\
 & + \sum_{a>b,\sigma} [U' n_{a\sigma} n_{b\sigma} + (U' - J) n_{a\sigma} n_{b\sigma}] \\
 & - \sum_{a<b} J (d_{a\downarrow}^\dagger d_{b\uparrow}^\dagger d_{b\downarrow} d_{a\uparrow} + d_{b\uparrow}^\dagger d_{b\downarrow}^\dagger d_{a\uparrow} d_{a\downarrow} + h.c.).
 \end{aligned} \tag{8.8}$$

其中, a, b 为能带指标 ($a, b = 1 \sim 3$), σ 为自旋指标, $n_{a\sigma} = c_{a\sigma}^\dagger c_{a\sigma}$ 为占据数算符, μ 为化学势, Δ 为杂质能级²⁷。 $U(U')$ 分别为轨道间 (轨道内) Coulomb 相互作用, J 为 Hund 交换相互作用, 我们采取如下限制 $U = U' + 2J$ 。实际上, 该局域哈密顿量与公式 (8.2) 中所描述的哈密顿量基本一致, 只是能带数目有所不同。

2. 计算细节

我们采用线性响应的方法来计算杂质部分的轨道磁化系数和自旋磁化系数, 计算要点如下:

- 半能带宽度 $D = 2t = 1.0$ eV, 三个能带完全简并
- Coulomb 相互作用 $U = 5.0$ eV, Hund 交换相互作用 $J = U/3$ 或者 $U/6$
- 在局域部分哈密顿量加上 Zeeman 项: $H_Z = \mu_B H_z (\hat{L}_z + 2\hat{S}_z)$
- 对角化原子问题, 对于每个原子本征态 Γ , 给出总轨道角动量的 z 分量的平均值 $L_\Gamma^{z,tot}$ 和总自旋角动量的 z 分量的平均值 $S_\Gamma^{z,tot}$
- 使用 **Begonia** 组件进行单步计算, 杂化函数由 ctqmc 程序初始化产生, 不需要用户提供任何 solver.hyb.in 文件。计算得到原子组态概率 P_Γ , 则杂质部分的总轨道角动量的 z 分量的平均值为 $L_{tot}^z = \sum_\Gamma P_\Gamma L_\Gamma^{z,tot}$, 总自旋角动量的 z 分量的平均值为 $S_{tot}^z = \sum_\Gamma P_\Gamma S_\Gamma^{z,tot}$
- 根据线性响应理论, 轨道磁化系数为 $\chi_{orb} = L_{tot}^z / H_z$, 自旋磁化系数为 $\chi_{spin} = S_{tot}^z / H_z$

3. 计算结果及分析

下面简单展示我们的计算结果。图8.34是杂质的轨道磁化系数 χ_{orb}^{-1} 随着温度的变化, 图8.35是杂质的自旋磁化系数 χ_{spin}^{-1} 随着温度的变化。从图中可以看出: 在温度较高时, 磁化系数显示出居里 - 外斯的特征, 表示系统具有自由的局域磁矩存在; 当温度降低时磁化趋于饱和, 局域磁矩被屏蔽。Kondo 温度定义为磁矩被完全屏蔽的转变点, 图8.34显示轨道部分的 Kondo 温度大约为 0.1 eV, 图8.35显示自旋部分的 Kondo 温度大约为 0.01 eV, 可见轨道 Kondo 温度比自旋 Kondo 温度约高一个数量级。

²⁷在本案例中 $\Delta = 0$ 。

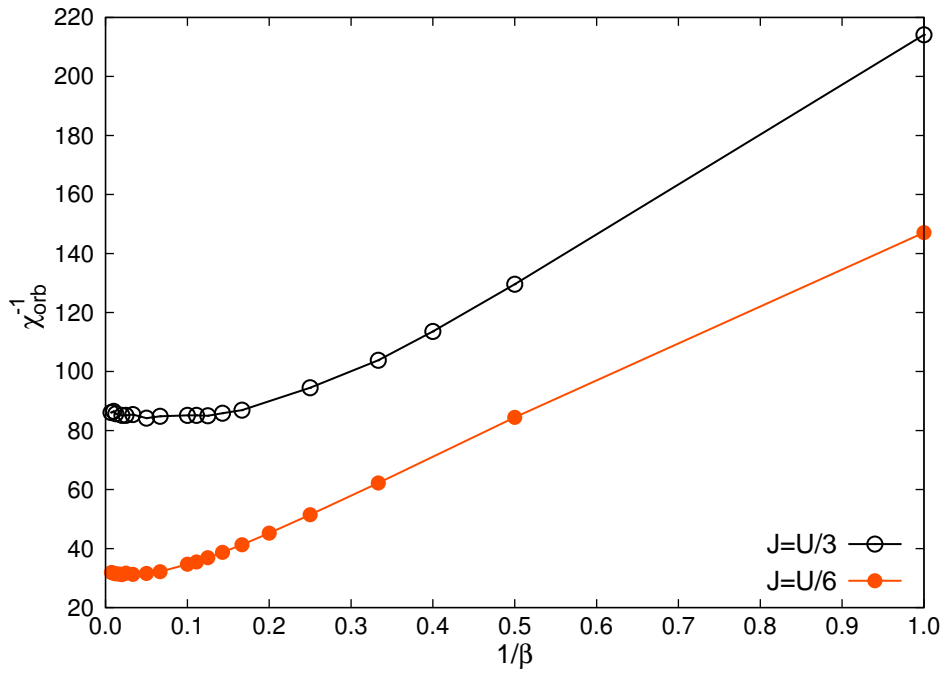


图 8.34: 三带 Anderson 杂质模型中杂质的轨道磁化系数 χ_{orb}^{-1} 随着温度的变化

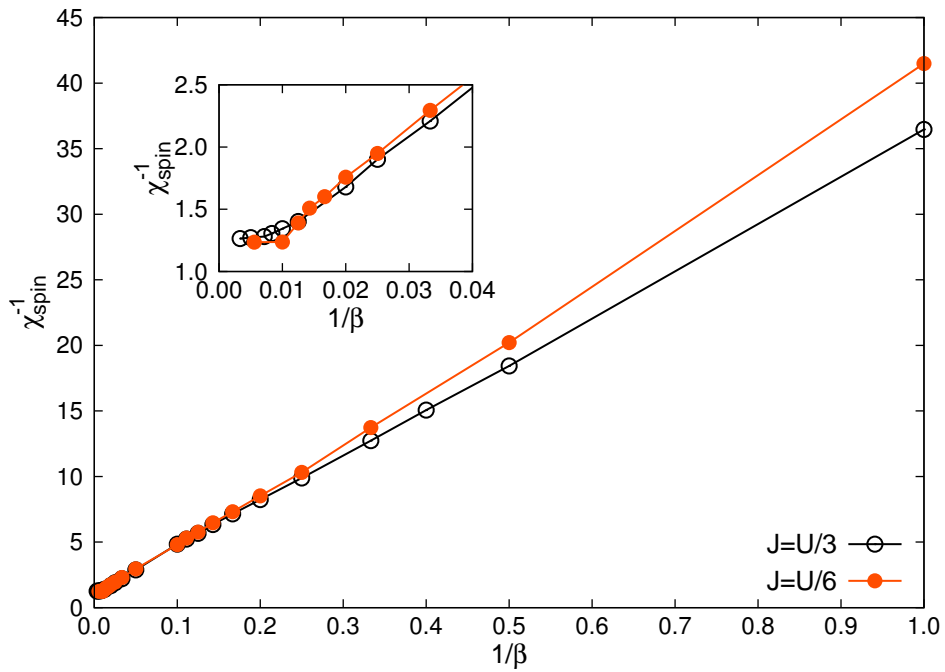


图 8.35: 三带 Anderson 杂质模型中杂质的自旋磁化系数 χ_{spin}^{-1} 随着温度的变化

附录

A.1 如何获得 *iQIST* 的最新版本？

请发送电子邮件至 huangli712@gmail.com 或者是 huangli712@yahoo.com.cn 索取，请在邮件标题上注明 *iQIST* 软件包字样并在邮件正文中留下您的工作单位以及详细的联系方式。经确认后，我们会在 1 ~ 3 个工作日内以附件的形式向您发送 *iQIST* 最新版本的安装包。

A.2 如何获得本文档的最新版本？

请发送电子邮件至 huangli712@gmail.com 或者是 huangli712@yahoo.com.cn 索取，请在邮件标题上注明 *iQIST* 用户手册字样并在邮件正文中留下您的工作单位以及详细的联系方式。经确认后，我们会在 1 ~ 3 个工作日内以附件的形式向您发送 *iQIST* 用户手册的最新版本。

A.3 *iQIST* 的开发者是谁？

到目前为止 (March 26, 2013)，*iQIST* 的唯一开发者为本文作者黄理 (四川材料与工艺研究所)。如果您对 *iQIST* 的开发有兴趣，可以与我联系 ([mailto: huangli712@gmail.com](mailto:huangli712@gmail.com) 或者 huangli712@yahoo.com.cn)。

A.4 *iQIST* 的维护者是谁？

到目前为止 (March 26, 2013)，*iQIST* 的唯一维护者为本文作者黄理 (四川材料与工艺研究所)。如果您对 *iQIST* 的维护有兴趣，可以与我联系 ([mailto: huangli712@gmail.com](mailto:huangli712@gmail.com) 或者 huangli712@yahoo.com.cn)。

A.5 如何获得技术支持？

很遗憾，由于时间精力等方面的原因，对于用户在使用 *iQIST* 过程中所遇到的技术问题，我们目前一般不提供无偿的技术支持。

A.6 如何正确选取 *iQIST* 的组件？

在 *iQIST* 软件包中，提供了许多量子杂质求解器组件，那么究竟使用那一个组件呢？

首先请判断系统的相互作用类型，是否具有密度 – 密度相互作用形式亦或者为广义相互作用。如果是前者，那么用户可以选择 *Azalea*、*Gardenia*、*Narcissus* 等组件；如果是后者，那么可以选择 *Begonia*、*Lavender*、*Camellia*、*Epiphyllum*、*Pansy* 和 *Manjushaka* 等组件。

其次请确定需要计算那些物理量，因为不同的量子杂质求解器组件，它能够测量的物理量是不完全一样的。以双粒子格林函数为例，目前仅有 *Gardenia*、*Narcissus* 和 *Lavender* 组件能够测量。至于动态屏蔽效应以及 Hubbard-Holstein 模型，目前仅有 *Narcissus* 组件能够处理。关于量子杂质求解器组件的功能，请参阅第1.4节。

再次请依照效率优先原则，优先选用效率高的量子杂质求解器组件。例如，在基于段表示算法的量子杂质求解器组件中，*Azalea* 组件的效率最高，*Gardenia* 组件次之，*Narcissus* 组件最差。如果没有特殊的要求，那么 *Azalea* 组件是最优的选择。

A.7 什么时候可以使用正交多项式方法？

当用户需要更为精确的测量结果，同时对计算速度的要求不是太高的时候，建议激活正交多项式方法。详情请参阅第6.5节、第8.3.3小节以及第8.3.4小节。

请注意，不是所有的量子杂质求解器组件均支持正交多项式功能，请参阅第 1.4节的描述。

A.8 如何激活内核多项式功能？

请打开 `ctqmc_record.f90` 文件，找到 `cat_make_gtau()` 子程序，再找到 `damp` 参数。`damp` 参数用来指定不同的内核多项式。在缺省状态下，`damp = 0`，这意味着不采用内核多项式方法。将 `damp` 参数置为 1，这意味着采用 Jackson 内核，经过我们的测试，该内核是最优的选择。当然了，用户也可以尝试其它的选择。修改 `damp` 参数完毕后，请保存退出，重新编译程序产生 `ctqmc` 可执行文件。此后用户的 `ctqmc` 程序将激活内核多项式功能。

请注意，不是所有的量子杂质求解器组件均支持内核多项式功能，请参阅第 1.4节的描述。

A.9 如何固定占据数，搜索费米面？

第一步：修改 `ctqmc_main.f90` 文件，在 `call ctqmc_dmft_selfer()` 语句后面加上对 `ctqmc_dmft_fermi()` 子程序的调用。

第二步：修改 `ctqmc_dmft.f90` 文件，添加 `ctqmc_dmft_fermi()` 子程序。

第三步：`ctqmc_dmft_fermi()` 子程序的核心算法如下：

- 定义目标占据数 N_{final}
- 计算当前占据数 N_{curr}
- 获取当前的化学势 μ_{curr}
- 计算新的化学势 $\mu_{\text{new}} = \mu_{\text{curr}} + (N_{\text{final}} - N_{\text{curr}})$
- 如有必要，更新相关的物理量

第四步：保存所有修改，重新编译程序。

此算法的精度不是很高，实际占据数与目标占据数之间的偏差大约在 ± 0.01 左右。并且如果体系属于绝缘相，那么可能需要迭代计算很多次才能得到所需要的结果 ($\text{niter} \sim 60$)。

A.10 如何计算高阶关联函数？

请使用 [Gardenia](#) 组件或者是 [Narcissus](#) 组件，并查阅第6.6小节关于 `isvrt` 参数的叙述。

A.11 如何考虑不同的晶格态密度？

第一步：产生合适的晶格态密度。用户既可以使用 [Hibiscus](#) /`hibiscus-toolbox` 组件中的 `makedos` 程序来产生晶格态密度，亦可以自行构造晶格态密度。晶格态密度数据必须存储在 `solver.anydos.in` 文件中，关于 `solver.anydos.in` 文件的详细格式，请参阅第4.4节。

第二步：修改 `ctqmc_dmft.f90` 文件中的 `ctqmc_dmft_selfer()` 子程序，注释掉对 `ctqmc_dmft_bethe()` 子程序的调用，改为对 `ctqmc_dmft_anydos()` 子程序的调用即可。

第三步：保存修改后的 `ctqmc_dmft.f90` 文件，重新进行编译产生 `ctqmc` 程序。

第四步：正常进行 DMFT 自治计算。

A.12 出现 segment fault 错误时应该怎么办？

出现此错误多半是因为 `ctqmc` 程序所使用的堆栈空间超出了操作系统的限制。解决办法是修改 `make.sys` 文件，在 `LEVEL` 选项后面添加 `-heap-arrays 8192`，然后重新编译，重新运行程序即可。如果问题还没有解决，请将相关的资料（包括程序源代码、输入文件、用户的系统配置、编译环境配置和当前的随机数种子等等）发送给程序开发者。

A.13 如何使用 gfortran 编译 *iQIST* 组件程序？

使用 `gfortran` 编译 *iQIST* 组件程序很简单，只需要修改 `make.sys` 文件中的以下几点：

- 将预处理选项 `FPP=-fpp` 改为 `FPP=-cpp`
- 注释掉机器优化选项 `CHECK`、`CDUMP`、`MTUNE` 的所有内容
- `LEVEL` 选项只保留 `-O3` 选项，其余的全部注释掉
- `LAPACK` 和 `BLAS` 也必须使用 `gfortran` 编译的版本

保存修改后的 `make.sys` 文件，然后重新编译即可。

参考文献

- [1] Antoine Georges, Gabriel Kotliar, Werner Krauth, and Marcelo J. Rozenberg. Dynamical mean-field theory of strongly correlated fermion systems and the limit of infinite dimensions. *Rev. Mod. Phys.*, 68:13, 1996.
- [2] G. Kotliar, S. Y. Savrasov, K. Haule, V. S. Oudovenko, O. Parcollet, and C. A. Marianetti. Electronic structure calculations with dynamical mean-field theory. *Rev. Mod. Phys.*, 78:865, 2006.
- [3] K. Held. Electronic structure calculations using dynamical mean field theory. *Adv. Phys.*, 56:829–926, 2007.
- [4] J. Hubbard. Electron Correlations in Narrow Energy Bands. II. The Degenerate Band Case. *Proc. R. Soc. A*, 277:237, 1964.
- [5] N. E. Bickers. Review of techniques in the large- N expansion for dilute magnetic alloys. *Rev. Mod. Phys.*, 59:845, 1987.
- [6] T. Pruschke and N. Grewe. The Anderson model with finite Coulomb repulsion. *Z. Phys. B*, 74:439, 1989.
- [7] K. Haule, S. Kirchner, J. Kroha, and P. Wolfle. Anderson impurity model at finite Coulomb interaction U : Generalized noncrossing approximation. *Phys. Rev. B*, 64:155111, 2001.
- [8] K. G. Wilson. The renormalization group: Critical phenomena and the Kondo problem. *Rev. Mod. Phys.*, 47:773, 1975.
- [9] S. R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69:2863, 1992.
- [10] U. Schollwöck. The density-matrix renormalization group. *Rev. Mod. Phys.*, 77:259, 2005.
- [11] M. Caffarel and W. Krauth. Exact diagonalization approach to correlated fermions in infinite dimensions: Mott transition and superconductivity. *Phys. Rev. Lett.*, 72:1545, 1994.

- [12] J. E. Hirsch and R. M. Fye. Monte Carlo Method for Magnetic Impurities in Metals. *Phys. Rev. Lett.*, 56:2521, 1986.
- [13] M. Jarrell. Hubbard model in infinite dimensions: A quantum Monte Carlo study. *Phys. Rev. Lett.*, 69:168–171, 1992.
- [14] Shiro Sakai, Ryotaro Arita, Karsten Held, and Hideo Aoki. Quantum Monte Carlo study for multi-orbital systems with preserved spin and orbital rotational symmetries. *Phys. Rev. B*, 74:155102, 2006.
- [15] D. Handscomb. The Monte Carlo method in quantum statistical mechanics. *Proc. Cambridge Philos. Soc.*, 58:594, 1962.
- [16] D. Handscomb. A Monte Carlo method applied to the Heisenberg ferromagnet. *Proc. Cambridge Philos. Soc.*, 60:115, 1964.
- [17] A. W. Sandvik and J. Kurkijarvi. Quantum Monte Carlo simulation method for spin systems. *Phys. Rev. B*, 43:5950, 1991.
- [18] N. V. Prokof'ev, B. V. Svistunov, and I. S. Tupitsyn. Exact quantum Monte Carlo process for the statistics of discrete systems. *JETP Lett.*, 64:911, 1996.
- [19] B. B. Beard and U. J. Wiese. Simulations of Discrete Quantum Systems in Continuous Euclidean Time. *Phys. Rev. Lett.*, 77:5130, 1996.
- [20] N. V. Prokof'ev and B. V. Svistunov. Polaron Problem by Diagrammatic Quantum Monte Carlo. *Phys. Rev. Lett.*, 81:2514, 1998.
- [21] N. V. Prokof'ev, B. V. Svistunov, and I. S. Tupitsyn. Exact, Complete, and Universal Continuous-Time Worldline Monte Carlo Approach to the Statistics of Discrete Quantum Systems. *Sov. Phys. JETP*, 87:310, 1998.
- [22] S. M. A. Rombouts, K. Heyde, and N. Jachowicz. Quantum Monte Carlo Method for Fermions, Free of Discretization Errors. *Phys. Rev. Lett.*, 82:4155–4159, 1999.
- [23] Emanuel Gull, Andrew J. Millis, Alexander I. Lichtenstein, Alexey N. Rubtsov, Matthias Troyer, and Philipp Werner. Continuous-time Monte Carlo methods for quantum impurity models. *Rev. Mod. Phys.*, 83:349–404, 2011.
- [24] A. N. Rubtsov, V. V. Savkin, and A. I. Lichtenstein. Continuous-time quantum Monte Carlo method for fermions. *Phys. Rev. B*, 72:035122, 2005.

- [25] F. F. Assaad and T. C. Lang. Diagrammatic determinantal quantum Monte Carlo methods: Projective schemes and applications to the Hubbard-Holstein model. *Phys. Rev. B*, 76:035116, 2007.
- [26] Philipp Werner, Armin Comanac, Luca de' Medici, Matthias Troyer, and Andrew J. Millis. Continuous-Time Solver for Quantum Impurity Models. *Phys. Rev. Lett.*, 97:076405, 2006.
- [27] Philipp Werner and Andrew J. Millis. Hybridization expansion impurity solver: General formulation and application to Kondo lattice and two-orbital models. *Phys. Rev. B*, 74:155107, 2006.
- [28] Philipp Werner and Andrew J. Millis. Efficient Dynamical Mean Field Simulation of the Holstein-Hubbard Model. *Phys. Rev. Lett.*, 99:146404, 2007.
- [29] Philipp Werner and Andrew J. Millis. Dynamical Screening in Correlated Electron Materials. *Phys. Rev. Lett.*, 104:146401, 2010.
- [30] Andreas M. Lauchli and Philipp Werner. Krylov implementation of the hybridization expansion impurity solver and application to 5-orbital models. *Phys. Rev. B*, 80:235117, 2009.
- [31] Kristjan Haule. Quantum Monte Carlo impurity solver for cluster dynamical mean-field theory and electronic structure calculations with adjustable cluster base. *Phys. Rev. B*, 75:155113, 2007.
- [32] E. Gull, P. Werner, O. Parcollet, and M. Troyer. Continuous-time auxiliary-field Monte Carlo for quantum impurity models. *Europhys. Lett.*, 82:57003, 2008.
- [33] Junya Otsuki, Hiroaki Kusunose, Philipp Werner, and Yoshio Kuramoto. Continuous-time quantum monte carlo method for the coqblin-schrieffer model. *J. Phys. Soc. Japan*, 76:114707, 2007.
- [34] Emanuel Gull, Philipp Werner, Andrew Millis, and Matthias Troyer. Performance analysis of continuous-time solvers for quantum impurity models. *Phys. Rev. B*, 76:235123, 2007.
- [35] N. Blumer. Efficiency of quantum Monte Carlo impurity solvers for the dynamical mean-field theory. *Phys. Rev. B*, 76:205120, 2007.
- [36] Munehisa Matsumoto, Myung Joon Han, Junya Otsuki, and Sergey Y. Savrasov. First-Principles Simulations of Heavy Fermion Cerium Compounds Based on the Kondo Lattice. *Phys. Rev. Lett.*, 103:096403, 2009.
- [37] Philipp Werner and Andrew J. Millis. High-Spin to Low-Spin and Orbital Polarization Transitions in Multiorbital Mott Systems. *Phys. Rev. Lett.*, 99:126405, 2007.
- [38] Philipp Werner, Emanuel Gull, Matthias Troyer, and Andrew J. Millis. Spin Freezing Transition and Non-Fermi-Liquid Self-Energy in a Three-Orbital Model. *Phys. Rev. Lett.*, 101:166405, 2008.

- [39] Philipp Werner, Emanuel Gull, and Andrew J. Millis. Metal-insulator phase diagram and orbital selectivity in three-orbital models with rotationally invariant Hund coupling. *Phys. Rev. B*, 79:115119, 2009.
- [40] Ching-Kit Chan, Philipp Werner, and Andrew J. Millis. Magnetism and orbital ordering in an interacting three-band model: A dynamical mean-field study. *Phys. Rev. B*, 80:235114, 2009.
- [41] Kristjan Haule and Gabriel Kotliar. Strongly correlated superconductivity: A plaquette dynamical mean-field theory study. *Phys. Rev. B*, 76:104509, 2007.
- [42] E. Gull, P. Werner, X. Wang, M. Troyer, and A. J. Millis. Local order and the gapped phase of the Hubbard model: A plaquette dynamical mean-field investigation. *Europhys. Lett.*, 84:37009, 2008.
- [43] H. Park, K. Haule, and G. Kotliar. Cluster Dynamical Mean Field Theory of the Mott Transition. *Phys. Rev. Lett.*, 101:186403, 2008.
- [44] Michel Ferrero, Pablo S. Cornaglia, Lorenzo De Leo, Olivier Parcollet, Gabriel Kotliar, and Antoine Georges. Pseudogap opening and formation of Fermi arcs as an orbital-selective Mott transition in momentum space. *Phys. Rev. B*, 80:064501, 2009.
- [45] E. Gull, O. Parcollet, P. Werner, and A. J. Millis. Momentum-sector-selective metal-insulator transition in the eight-site dynamical mean-field approximation to the Hubbard model in two dimensions. *Phys. Rev. B*, 80:245102, 2009.
- [46] K. Mielsonson, E. Khatami, D. Galanakis, A. Macridin, J. Moreno, and M. Jarrell. Thermodynamics of the quantum critical point at finite doping in the two-dimensional Hubbard model studied via the dynamical cluster approximation. *Phys. Rev. B*, 80:140505, 2009.
- [47] Philipp Werner, Emanuel Gull, Olivier Parcollet, and Andrew J. Millis. Momentum-selective metal-insulator transition in the two-dimensional Hubbard model: An 8-site dynamical cluster approximation study. *Phys. Rev. B*, 80:045120, 2009.
- [48] G. Sordi, K. Haule, and A.-M. S. Tremblay. Finite Doping Signatures of the Mott Transition in the Two-Dimensional Hubbard Model. *Phys. Rev. Lett.*, 104:226402, 2010.
- [49] C. A. Marianetti, K. Haule, and O. Parcollet. Quasiparticle Dispersion and Heat Capacity of $\text{Na}_{0.3}\text{CoO}_2$: A Dynamical Mean-Field Theory Study. *Phys. Rev. Lett.*, 99:246404, 2007.
- [50] C. Slezak, M. Jarrell, T. Maier, and J. Deisz. Multi-scale extensions to quantum cluster methods for strongly correlated electron systems. *J. Phys.: Condens. Matter*, 21:435604, 2009.

- [51] A. N. Rubtsov, M. I. Katsnelson, and A. I. Lichtenstein. Dual fermion approach to nonlocal correlations in the Hubbard model. *Phys. Rev. B*, 77:033101, 2008.
- [52] A. Toschi, A. A. Katanin, and K. Held. Dynamical vertex approximation: A step beyond dynamical mean-field theory. *Phys. Rev. B*, 75:045118, 2007.
- [53] V. V. Savkin, A. N. Rubtsov, M. I. Katsnelson, and A. I. Lichtenstein. Correlated Adatom Trimer on a Metal Surface: A Continuous-Time Quantum Monte Carlo Study. *Phys. Rev. Lett.*, 94:026402, 2005.
- [54] Kristjan Haule and Gabriel Kotliar. Avoided criticality in near-optimally doped high-temperature superconductors. *Phys. Rev. B*, 76:092503, 2007.
- [55] J. H. Shim, K. Haule, and G. Kotliar. Modeling the Localized-to-Itinerant Electronic Transition in the Heavy Fermion System CeIrIn₅. *Science*, 318:1615–1617, 2007.
- [56] J. H. Shim, K. Haule, and G. Kotliar. Fluctuating valence in a correlated solid and the anomalous properties of δ -plutonium. *Nature*, 446:513–516, 2007.
- [57] Hyowon Park, K. Haule, C. A. Marianetti, and G. Kotliar. Dynamical mean-field theory study of Nagaoka ferromagnetism. *Phys. Rev. B*, 77:035107, 2008.
- [58] L. De Leo, C. Kollath, A. Georges, M. Ferrero, and O. Parcollet. Trapping and Cooling Fermionic Atoms into Mott and Néel States. *Phys. Rev. Lett.*, 101:210403, 2008.
- [59] T. L. Dao, M. Ferrero, A. Georges, M. Capone, and O. Parcollet. Polarized Superfluidity in the Attractive Hubbard Model with Population Imbalance. *Phys. Rev. Lett.*, 101:236405, 2008.
- [60] Philipp Werner, Takashi Oka, Martin Eckstein, and Andrew J. Millis. Weak-coupling quantum Monte Carlo calculations on the Keldysh contour: Theory and application to the current-voltage characteristics of the Anderson model. *Phys. Rev. B*, 81:035108, 2010.
- [61] Philipp Werner, Takashi Oka, and Andrew J. Millis. Diagrammatic Monte Carlo simulation of nonequilibrium systems. *Phys. Rev. B*, 79:035320, 2009.
- [62] Lothar Muhlbacher and Eran Rabani. Real-Time Path Integral Approach to Nonequilibrium Many-Body Quantum Systems. *Phys. Rev. Lett.*, 100:176403, 2008.
- [63] Marco Schiró. Real-time dynamics in quantum impurity models with diagrammatic Monte Carlo. *Phys. Rev. B*, 81:085126, 2010.
- [64] Marco Schiro and Michele Fabrizio. Real-time diagrammatic Monte Carlo for nonequilibrium quantum transport. *Phys. Rev. B*, 79:153302, 2009.

- [65] T. L. Schmidt, P. Werner, L. Muhlbacher, and A. Komnik. Transient dynamics of the Anderson impurity model out of equilibrium. *Phys. Rev. B*, 78:235110, 2008.
- [66] Lewin Boehnke, Hartmut Hafermann, Michel Ferrero, Frank Lechermann, and Olivier Parcollet. Orthogonal polynomial representation of imaginary-time Green’s functions. *Phys. Rev. B*, 84:075145, 2011.
- [67] Mark Jarrell and J.E. Gubernatis. Bayesian inference and the analytic continuation of imaginary-time quantum Monte Carlo data. *Phys. Rep.*, 269:133–195, 1996.
- [68] K. S. D. Beach. Identifying the maximum entropy method as a special limit of stochastic analytic continuation. 2004.
- [69] Hartmut Hafermann, Kelly R. Patton, and Philipp Werner. Improved Estimators for the Self-Energy and Vertex Function in Hybridization Expansion Continuous-Time Quantum Monte Carlo Simulations. 2011.
- [70] Kristjan Haule, Chuck-Hou Yee, and Kyoo Kim. Dynamical mean-field theory within the full-potential methods: Electronic structure of CeIrIn₅, CeCoIn₅, and CeRhIn₅. *Phys. Rev. B*, 81:195107, 2010.
- [71] Jaebeom Yoo, Shailesh Chandrasekharan, Ribhu K Kaul, Denis Ullmo, and Harold U Baranger. On the sign problem in the hirsch–fye algorithm for impurity problems. *J. Phys. A: Math. Gene.*, 38:10307, 2005.
- [72] V. I. Anisimov, I. A. Nekrasov, D. E. Kondakov, T. M. Rice, and M. Sigrist. Orbital-selective mott-insulator transition in $\text{Ca}_{2-x}\text{Sr}_x\text{RuO}_4$. *The European Physical Journal B*, 25(2):191–201, 2002.
- [73] Akihisa Koga, Norio Kawakami, T. M. Rice, and Manfred Sigrist. Orbital-selective mott transitions in the degenerate hubbard model. *Phys. Rev. Lett.*, 92(21):216402, 2004.
- [74] Akihisa Koga, Norio Kawakami, T. M. Rice, and Manfred Sigrist. Spin, charge, and orbital fluctuations in a multiorbital mott insulator. *Phys. Rev. B*, 72(4):045128, 2005.
- [75] L. de’Medici, A. Georges, and S. Biermann. Orbital-selective mott transition in multiband systems: Slave-spin representation and dynamical mean-field theory. *Phys. Rev. B*, 72(20):205124, 2005.
- [76] Michel Ferrero, Federico Becca, Michele Fabrizio, and Massimo Capone. Dynamical behavior across the mott transition of two bands with different bandwidths. *Phys. Rev. B*, 72(20):205126, 2005.
- [77] R. Arita and K. Held. Orbital-selective mott-hubbard transition in the two-band hubbard model. *Phys. Rev. B*, 72(20):201102(R), 2005.

- [78] C. Knecht, N. Blümer, and P. G. J. van Dongen. Orbital-selective mott transitions in the anisotropic two-band hubbard model at finite temperatures. *Phys. Rev. B*, 72(8):081103(R), 2005.
- [79] A. Liebsch. Mott Transitions in Multiorbital Systems. *Phys. Rev. Lett.*, 91:226401, 2003.
- [80] Akihisa Koga, Norio Kawakami, T.M. Rice, and Manfred Sigrist. Mott transitions in the multi-orbital systems. *Physica B*, 1366(16):359–361, 2005.
- [81] Kensuke Inaba, Akihisa Koga, Sei Suga, and Norio Kawakami. Phase diagram of orbital-selective mott transitions at finite temperatures. *J. Phys. Soc. Jpn.*, 74(9):2393, 2005.
- [82] Luca de’Medici, S.R. Hassan, Massimo Capone, and Xi Dai. Orbital-selective mott transition out of band degeneracy lifting. *Phys. Rev. Lett.*, 102(16):126401, 2009.
- [83] Tomoko Kita, Takuma Ohashi, and Norio Kawakami. Mott transition in three-orbital hubbard model with orbital splitting. *Phys. Rev. B*, 84(16):195130, 2011.
- [84] Akihisa Koga and Kensuke Inaba. Metal-insulator transition in the two-orbital hubbard model at fractional band fillings: Self-energy functional approach. *J. Phys. Soc. Jpn.*, 76(9):094712, 2007.
- [85] Eberhard Jakobi, Nils Blümer, and Peter Dogen. Orbital-selective mott transitions in a doped two-band hubbard model. *Phys. Rev. B*, 80(11):115109, 2009.
- [86] Z.P. Yin, K. Haule, and G. Kotliar. Fractional power-law behavior and its origin in iron-chalcogenide and ruthenate superconductors: Insights from first-principles calculations. 2012.