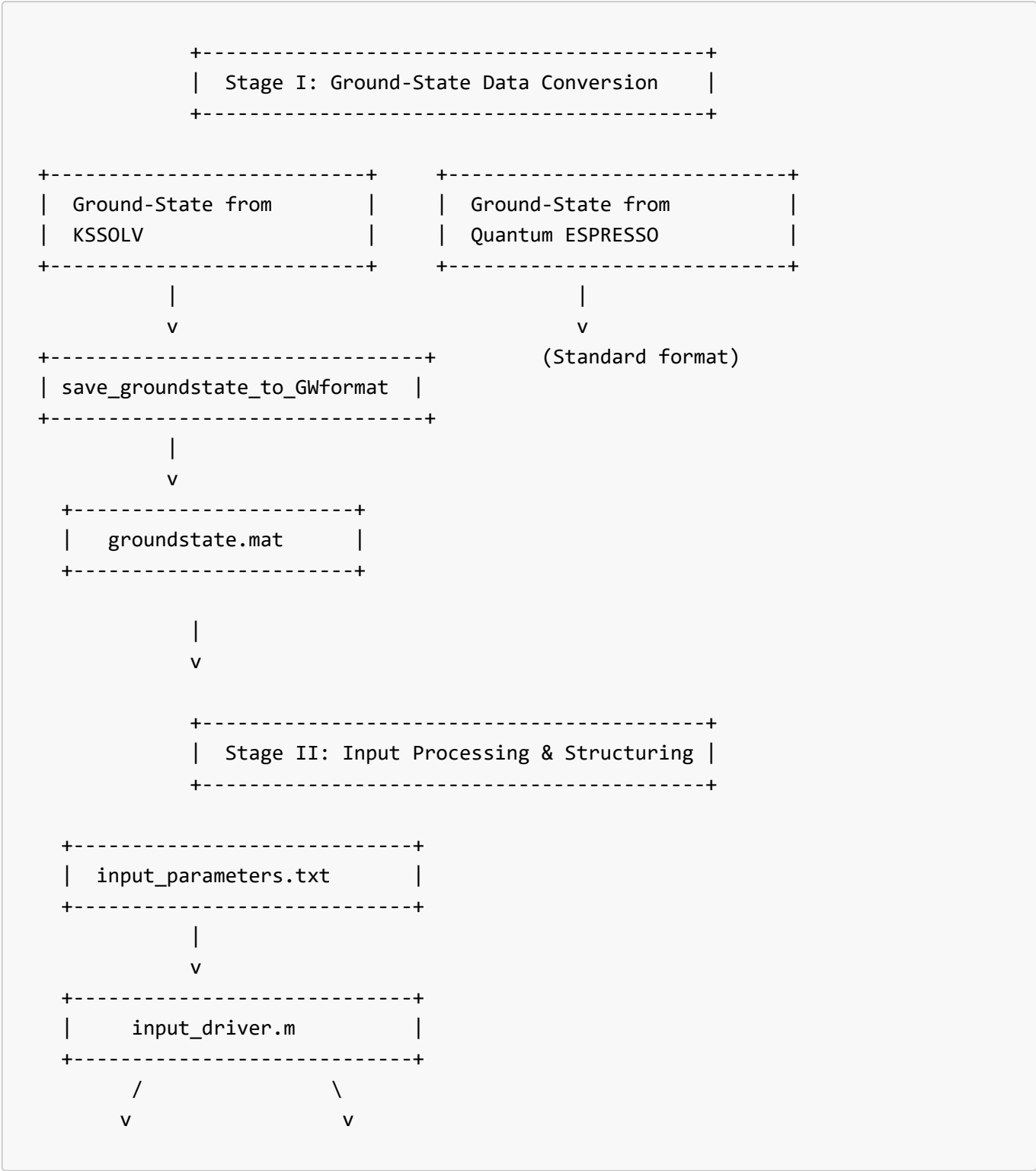


Workflow



+-----+ +-----+ | @GWinfo || Config Struct | +-----+ +-----+
---+





进度：

已完成

- ☐ Input文件夹的重构，以input_driver.m为主函数的从文件到格式化存储的数据
- ☐ 建设./testInput 内置测试demo，测试输入和用户手册

待完成

- ☐ 搁置GWOPTIONS.GWCal和.Groundstate的开发 (觉得原有的结构不甚合理)
- ☐ 需要给input_driver的一个输出

需讨论

- 讨论关于Coulomb里面的选择
- 自行查看minibzaverage.f90中怎么生成integrate的代码，对应vcoul_generator中的dvalue, 用于书写这里的construct_coulG0

注释

- 有些其他代码仍然用的是ha单位 (例如@gvec中的ecut，现在为了减少修改爆炸，所以暂时直接在gvec前将结果/2)

Current framework

GWOPTIONSFramework/ Input/ # Input parsing and validation layer | | | read_input_param.m # Read and parse the input file into a config struct | | | allowed_param_list.m # Define valid blocks and parameter names | | | default_param_values.m # Provide default values for each parameter | | | set_default_values.m # Fill missing fields in config using defaults | | | ... | | | @GWOPTIONS/ # Core class representing parameter configuration | | | GWOPTIONS.m # Class definition with constructor | | | fromConfig.m # Convert struct (config) to GWOPTIONS object | | | display.m # Customized display of essential parameters | | | ... # Additional methods like validation or shortcuts | | | examples/ # Example input files (optional) | | | input_example.in | | | README.md # Project documentation (this file)

Purpose

现在的GW module的输入都是直接提供一个struct options_GW，然后修改。但是实际上这样即使我一个源代码开发者都搞的乱七八糟 现在我期待，重新弄一套规范化输入。

- ☐ 将GW module里面的关键参数分成几大类，每一大类对应着某一个关键功能。
- ☐ 提供一个稳定的、可扩展的参数读取器。
- ☐ 用markdown书写一个面向用户的参数手册。
- ☐ 测试

Transform the Input of other codes, such as QE for the ground-state calculation, or BGW for GW calculation, to the inputs form the code can accept.

Input description