Fast evaluation of imaginary-time Feynman diagrams and robust bath fitting

Zhen Huang

Department of Mathematics, University of California, Berkeley

hertz@math.berkeley.edu

July 5, 2024
TRIQS Meeting, Collège de France, Paris.

Softwares

Two software tools: **ppsc-soe** and **adapol** .

1. **ppsc-soe**
   - **P**seudo-**p**article **s**elf-consistent **c**alculations via **s**um-**o**f-**e**xponentials.
   - **Fast**, ***deterministic***, and **parallel** impurity solver.
   - Written in C++, with a Python interface, compatible with TRIQS.
2. **adapol**
   - **Ada**ptive **pol**e fitting for Matsubara functions.
   - Applications: bath fitting in DMFT, analytic continuation.
   - Written in Python, with a TRIQS interface.

Acknowledgement

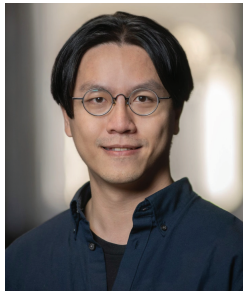
Impurity solver (ppsc-soe)

Jason Kaye
CCQ & CCM

Hugo Strand
Örebro U

Denis Golež
Jozef Stefan

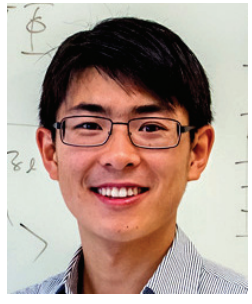
Pole fitting (Adapol)



Chia-nan Yeh
CCQ

Nils Wentzell
CCQ

Jason Kaye
CCQ & CCM



Lin Lin
UC Berkeley

Emanuel Gull
U Michigan

How?

A shared ingredient of both softwares are ***sum-of-exponentials approximation*** of the Matsubara functions, also known as the ***pole fitting*** problem.

Pole fitting of Matsubara functions

We'd like to find an approximation to $G(\mathrm{i}\nu_k)$ in the following form:

$$G(\mathrm{i}\nu_k) \approx \sum_{j=1}^{p} \frac{W_j}{\mathrm{i}\nu_k - \lambda_j}, \quad \forall k \in \mathbb{Z}.$$

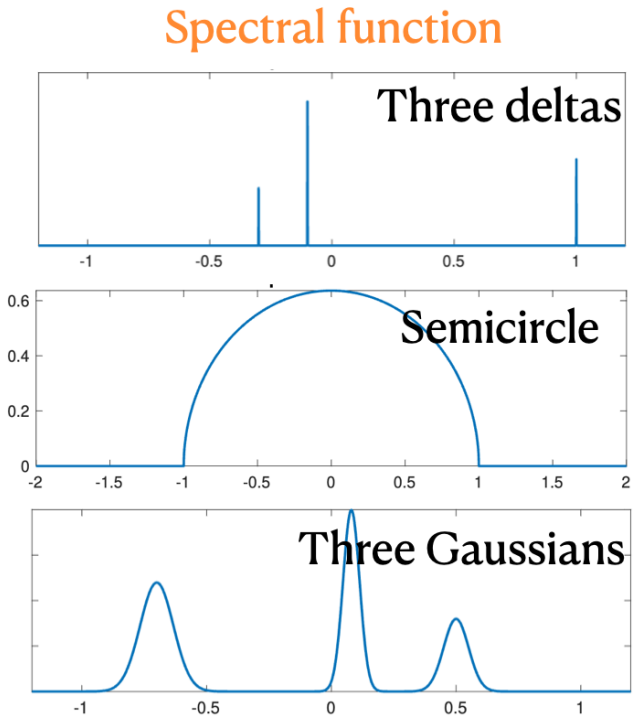
Here λ_j are the poles, and W_j are the weight matrices $(n \times n)$.
This is equivalent to the following approximation to $G(\tau)$:

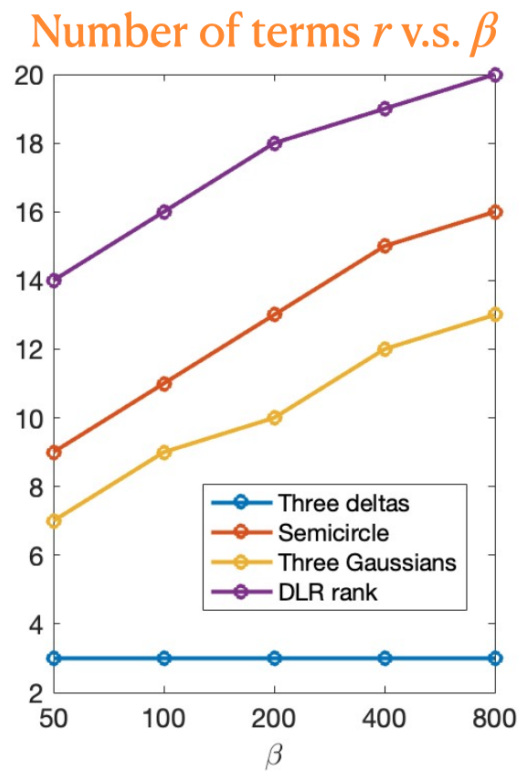
$$G(\tau) \approx -\sum_{j=1}^{p} \widetilde{W}_j \mathrm{e}^{-\lambda_j \tau}, \quad \forall \tau \in [0, \beta].$$

Here $\widetilde{W}_j = \frac{W_j}{1 \mp \mathrm{e}^{-\beta E}}$.
We hope that the number of terms p is ***as small as possible***.

Fewer poles than DLR?

- Let spectral width $\leq \Lambda$. Given $\beta\Lambda$ and accuracy ε, DLR provides a ***basis*** of exponentials for **all** Green's functions.
- Why do we want another pole fitting method?
- If we only need poles for a specific $G(\tau)$, could we find a more compact basis (poles)?
- The answer is YES!





AAA algorithm[1] for rational approximation

Barycentric formula for rational interpolants ($z \in \mathbb{C}$):

$$f(z) = p(z)/q(z) = \left(\sum_{l=1}^{m} \frac{w_l f_l}{z - z_l} \right) / \left(\sum_{l=1}^{m} \frac{w_l}{z - z_l} \right).$$

This satisfies $f(z_j) = f_j$ automatically at support points z_j.
AAA algorithm: an **iterative** procedure that selects the next support point in a ***greedy*** fashion. At the j-th iteration:

- Select the next sample point z_j, at which the error of the current barycentric approximant is maximal.
- Choose w_l's by minimizing $\sum_{z \neq z_j} |f(z)q(z) - p(z)|^2$. This is a linear problem and boils down to compute an SVD.
- Continue until desired accuracy.

The poles of $f(z)$ are the zeros of $q(z)$. This reduces to a polynomial root finding problem.

[1] Nakatsukasa, Yuji, Olivier Sète, and Lloyd N. Trefethen. "The AAA algorithm for rational approximation." SIAM Journal on Scientific Computing 40.3 (2018): A1494-A1522.

Pole fitting procedure

Given $\mathrm{i}\nu_j$ and $G(\mathrm{i}\nu_j)$,

Step 1: Use AAA algorithm to obtain poles $\{\lambda_k\}_{k=1}^{p}$.

Step 2: Obtain weights W_j using linear least square fitting.

$$\min_{W_j} \sum_k \left\| G(\mathrm{i}\nu_k) - \sum_{j=1}^{p} \frac{W_j}{\mathrm{i}\nu_k - \lambda_j} \right\|^2 .$$

Note that this is a linear least squares problem with respect to W_j. (But not linear for the poles λ_j).

Step 3: (optional) postprocessing: further refinement of the poles using bilevel optimizations.

Application: bath fitting

- In (equilibrium) DMFT with Hamitonian-based impurity solvers, bath fitting is a crucial subroutine for the self-consistent iterations.
- Given $\Delta(\mathrm{i}w)$, we want to obtain a fitting in the following form:

$$\Delta(\mathrm{i}w) \approx \sum_{k=1}^{N_b} \frac{V_k V_k^\dagger}{\mathrm{i}w - \omega_k}.$$

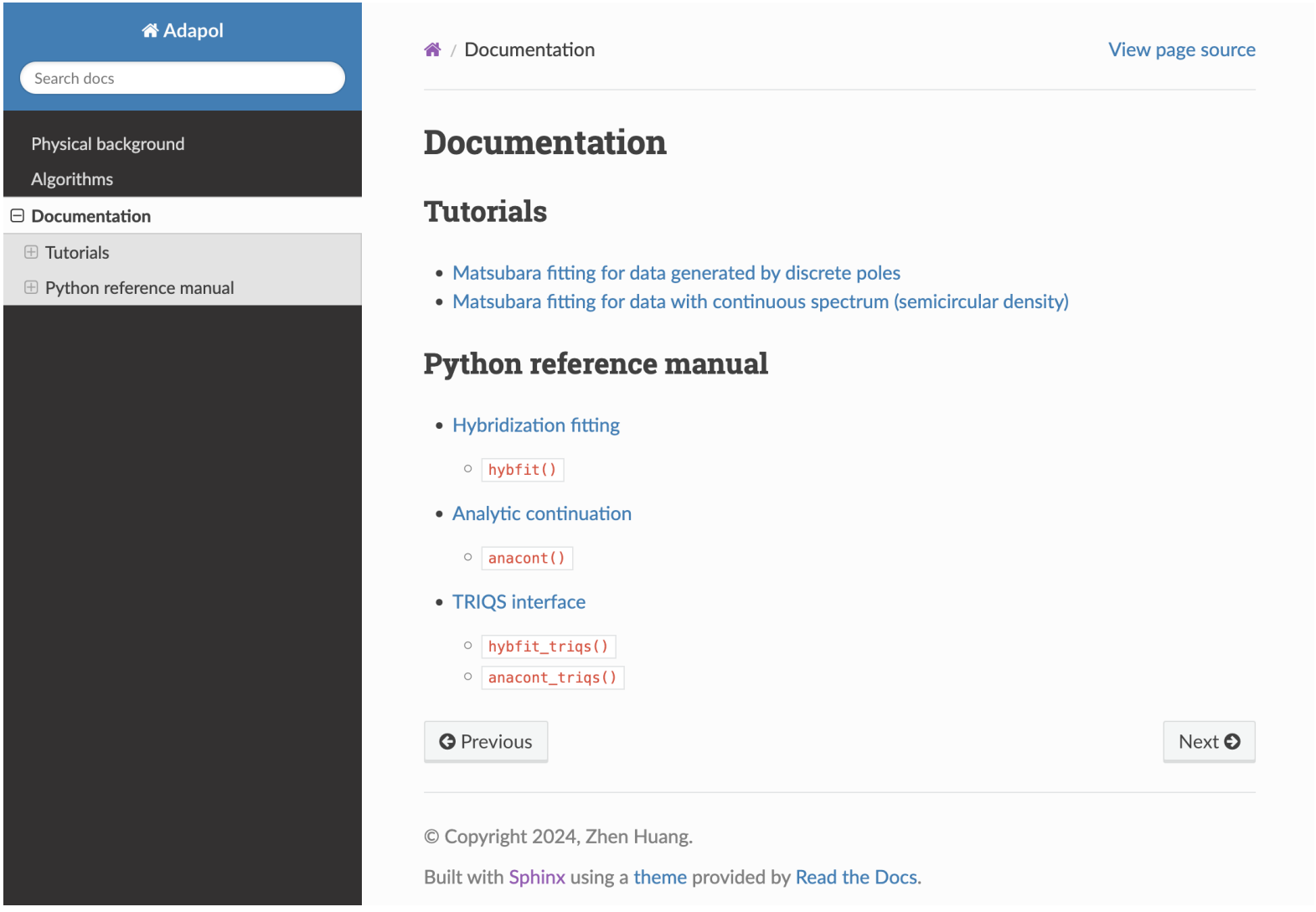
- This could be achieved via **pole fitting** +***semidefinite programming (SDP)*** + ***bilevel optimization***.
- Also applicable to the problem of analytic continuation [2], especially for noisy data.

[2] Z. Huang, E. Gull., and L. Lin. (2023). Robust analytic continuation of Green's functions via projection, pole estimation, and semidefinite relaxation. Physical Review B, 107(7), 075151.

Software for hybridization fitting: Adapol

https://github.com/flatironinstitute/adapol

Documentation: https://flatironinstitute.github.io/adapol/

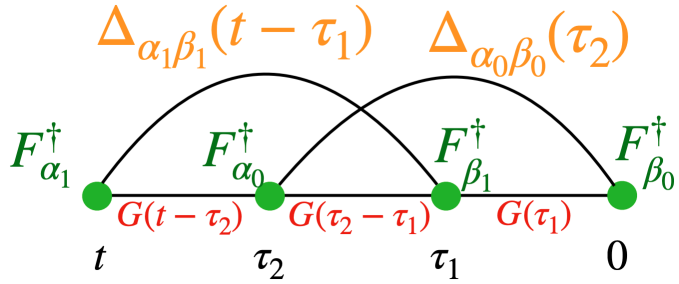


Application: fast diagram evaluation

- Using the same setting as in H. U. R. Strand's previous talk: strong coupling expansion and pseudo-particle representation as an example.
- The most time-consuming part is to calculate the self-energy Feynman diagrams and single-particle Green's functions Feynman diagrams.

Pseudo-particle self energy Feynman diagrams

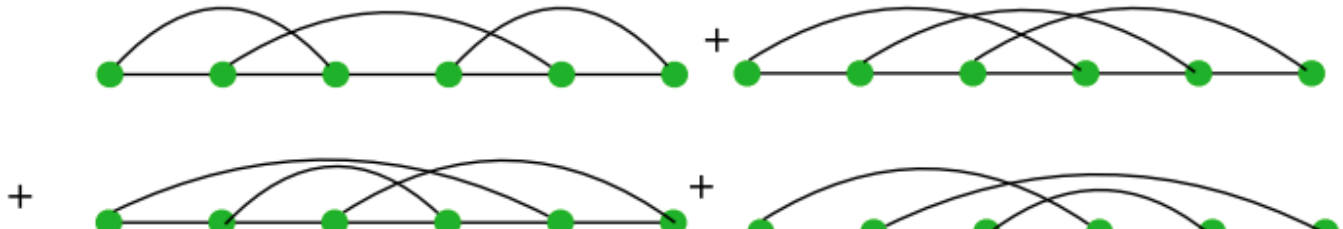
OCA (2nd order) diagrams:



$$= \sum_{\alpha_1,\beta_1=1}^{n}\sum_{\alpha_0,\beta_0=1}^{n} \int_0^t d\tau_2 \int_0^{\tau_2} d\tau_1 \, \Delta_{\alpha_1\beta_1}(t-\tau_1)\, \Delta_{\alpha_0\beta_0}(\tau_2) F^\dagger_{\alpha_1} G(t-\tau_2) F^\dagger_{\alpha_0} G(\tau_2-\tau_1) F_{\beta_1} G(\tau_1) F_{\beta_0}$$

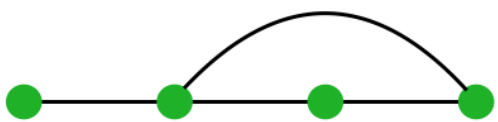
n: impurity size, $\Delta : n \times n$ p.p. hybridization function,
$G : N \times N$ p.p. Green's function,
$F_\alpha, F^\dagger_\beta : N \times N$: ED of impurty operators $\hat{c}_\alpha$, $\hat{c}^\dagger_\beta$. $(\alpha, \beta = 1, \cdots, n.)$

TCA (third order) diagrams:



Why is the calculation of these diagrams difficult: ***crossings of hybridization lines***

- An m-th order diagram is a $(2m-2)$-dimensional integral.
- Direct calculation cost: $O(n^{2m}\beta^{2m-2})$ for an n-th order diagram. (n is the impurity size.)
- If there are no crossing of hybridization lines:



$$= \int_0^t \mathrm{d}t_2 \int_0^{t_2} \mathrm{d}t_1 \Delta(t_2) F^\dagger G(t-t_2) F^\dagger G(t_2-t_1) F G(t_1) F$$

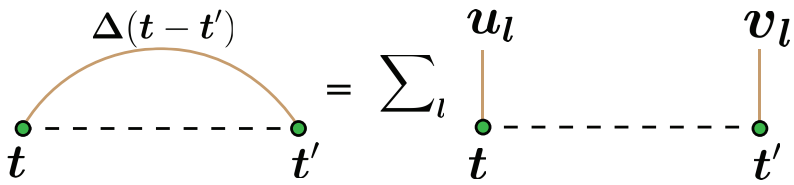
$$= \int_0^t \mathrm{d}t_2 F^\dagger G(t-t_2) F^\dagger \Delta(t_2) F \int_0^{t_2} \mathrm{d}t_1 G(t_2-t_1) F G(t_1) F\,.$$

- The 2D integral becomes two 1D integrals.

Cutting the hybridization

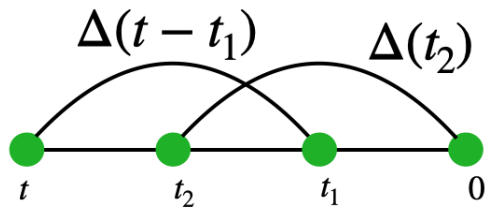
Decomposition of the hybridization function:

$$\Delta(t-t') = \sum_{l=1}^{r} c_l \mathrm{e}^{-\lambda_l t} \mathrm{e}^{\lambda_l t_1} =: \sum_{l=1}^{r} u_l(t) v_l(t_1).$$



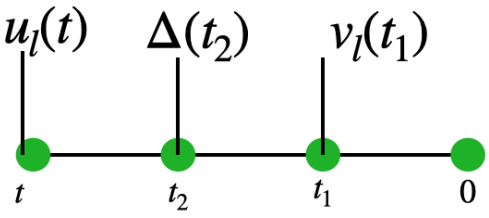
Take OCA diagrams as an example:

Consider $\int_0^t \mathrm{d}t_2 \int_0^{t_2} \mathrm{d}t_1 \Delta(t-t_1)\Delta(t_2)G(t-t_2)G(t_2-t_1)G(t_1)$.

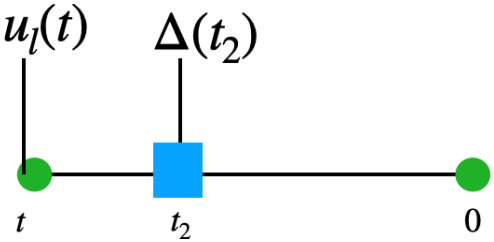


If $\Delta(t-t_1) = \sum_{l=1}^{r} u_l(t)v_l(t_1)$, then the original integral becomes

$$\sum_{l=1}^{r} \int_0^t \mathrm{d}t_2 \int_0^{t_2} \mathrm{d}t_1 u_l(t)v_l(t_1)\Delta(t_2)G(t-t_2)G(t_2-t_1)G(t_1) = \sum_l$$



$$= \sum_{l=1}^{r} u_l(t) \left(\int_0^t \mathrm{d}t_2 \Delta(t_2) G(t-t_2) \left(\int_0^{t_2} \mathrm{d}t_1 v_l(t_1) G(t_2-t_1) G(t_1) \right) \right) = \sum_l$$



Computational complexity

A pole fitting with r poles enables a hybridization of decomposition with nr terms:

$$\Delta(t - t_1) = \sum_{l=1}^{nr} u_l(t) v_l(t_1).$$

Computational cost for m-th order diagram:

$$O((nr)^{m-1} m r_0^2), \quad r_0 \text{ is the DLR rank.}$$

Comparison:

$$n^{2m} \beta^{2m-2} \to (nr)^{m-1} m r_0^2.$$

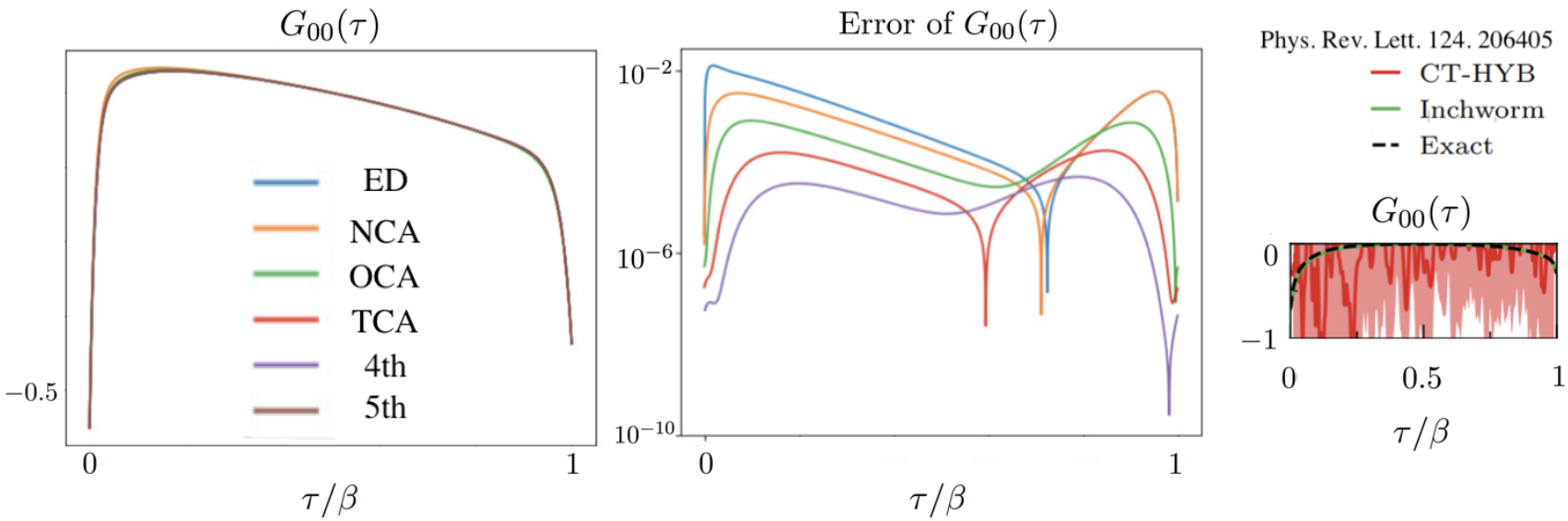
$$r_0 = \log(\beta), \quad r < r_0,$$

Examples

Spinless dimer model: strongly correlated, exact solvable.

$$H = U c_0^\dagger c_0 c_1^\dagger c_1 - v\left(c_0^\dagger c_1 + c_1^\dagger c_0\right) - t\sum_{k=0}^{1}\sum_{i=0}^{1}\left(c_i^\dagger b_{ik} + \text{h.c.}\right) - t'\sum_{k=0}^{1}\left(b_{0k}^\dagger b_{1k} + \text{h.c.}\right),$$

Plot of $G_{00}(\tau)$ [3] . (For $\beta = 16$.)



[3] Z. Huang, J. Kaye, H. Strand, D. Golez, 2024, in preparation.

Runtime [4] [5]

Spinless dimer ($n = 2$):

ppsc-soe (TCA)	ppsc-soe (4th order)	ppsc-soe (5th order)	Inchworm	CT-hyb
7.64 seconds	1 hour	90 hours	500 hours	12000 hours

Anderson two-band model ($n = 4$):

ppsc-soe (TCA)	ppsc-soe (4th order)	Inchworm	CT-hyb
0.2 hours	22 hours	1000 hours	3000 hours

DMFT calculation for toy model of Ca_2RuO_4 ($n = 6$):

TCA: 59.7 core hours (for $\beta = 10$).

It takes 6 iterations to reach accuracy $\varepsilon = 10^{-6}$).

There are $4 \times 2^3 \times 6^6 = 1492992$ 5-dim integrals.

[4] Z. Huang, J. Kaye, H. Strand, D. Golez, in preparation.

[5] Multiorbital Quantum Impurity Solver for General Interactions and Hybridizations, E. Eidelstein, E. Gulll and G. Cohen, Phys. Rev. Lett., 124, 206405.

Implementation

- Code is written in C++, based on cppdlr.
- Parallelizable: the evaluation of $(np)^{m-1}$ terms could be done simultaneously.
- Easy to use: python interface.
- TRIQS compatible.

Conclusion and outlook

Fast impurity solver:

- Robust, high-order accurate diagram evaluation possible beyond typical lowest-order approximations. Free of sign problem.
- Access to low temperature.
- Code ready to be tested on applications.

Bath fitting scheme:

- Python package (Adapol) available.
- TRIQS interface.

Ongoing work:

- Real-time bath fitting, real-time Feynman diagrams, real-time impurity solvers.
- Analytic continuation of continuous spectrum.
- Connections to Tensor Cross Interpolation (TCI).
- More software development ...

Thank you for your attention!