Flow Equation Approach for Bosonic Impurity Problems

Jan-Philipp Christ

LMU München

17. Juli 2023

Überblick

Überblick

Problemstellung

Der Flow Equation Approach

Flow Equations im rein quadratischen Fall

Implementierung des rein quadratischen Falls

Der physikalische Rahmen
- Das 1D Bose Polaron
- Die LLP- und die Displacement Transformation

Ergebnisse- was haben wir gelernt?

Was kommt jetzt? - Ausblick

Problemstellung

Wie bosonischen Hamiltonian der Form

$$\hat{\mathcal{H}} = \underbrace{\sum_k \omega_k \hat{a}_k^\dagger \hat{a}_k}_{=:\hat{\mathcal{H}}_0} + \underbrace{\sum_{q \neq q'} V_{q,q'} \hat{a}_q^\dagger \hat{a}_{q'} + \sum_{p,p'} \left(W_{p,p'} \hat{a}_p^\dagger \hat{a}_{p'}^\dagger + W_{p,p'}^* \hat{a}_p \hat{a}_{p'} \right)}_{=:\hat{\mathcal{H}}_{\mathrm{int}}}$$

diagonalisieren?

- bei skalaren Matrix-Elementen eigentlich uninteressant, da auch exakt diagonalisierbar (später)
- spannender: Matrix-Elemente sind Funktionale der (bosonischen) Besetzungszahlen $\{\hat{n}_k\}_k$

Überblick

Der Flow Equation Approach

- Hamiltonian $\hat{\mathcal{H}} = \hat{\mathcal{H}}_0 + \hat{\mathcal{H}}_{\text{int}}$
- Kochrezept zum Herleiten der Gleichungen:
  1. Berechne kanonischen Generator $\hat{\eta} = \left[\hat{\mathcal{H}}_0, \hat{\mathcal{H}}_{\text{int}}\right]$
  2. Berechne $\left[\hat{\eta}, \hat{\mathcal{H}}\right]$

     $\rightarrow$ Terme der Form $\left(\prod_{k_i} \hat{a}^\dagger_{k_i}\right) \cdot \left(\prod_{\kappa_i} \hat{a}_{\kappa_i}\right)$ werden mit $\left(\sum_{k_i} \omega_{k_i} - \sum_{\kappa_i} \omega_{\kappa_i}\right)^2$ multipliziert
  3. Schließe auf Form des Flow Hamiltonians. Wenn Flow Hamiltonian nicht von gleicher Form wie ursprünglicher Hamiltonian, gehe zu 2.
  4. Flow Equations durch Koeffizientenvergleich in

$$\frac{\mathrm{d}\hat{\mathcal{H}}(\lambda)}{\mathrm{d}\lambda} = \left[\hat{\eta}(\lambda), \hat{\mathcal{H}}(\lambda)\right]$$

Überblick

Flow Equations im rein quadratischen Fall

$$\partial_\lambda \omega_k = \sum_{\tilde q} 2V_{\tilde q,k}V_{k,\tilde q}(\omega_k - \omega_{\tilde q}) - 2\sum_{\tilde p}(W_{k,\tilde p} + W_{\tilde p,k})(\omega_k + \omega_{\tilde p})(W^*_{\tilde p,k} + W^*_{k,\tilde p})$$

$$\partial_\lambda V_{q,q'} = -V_{q,q'}(\omega_q - \omega_{q'})^2 - \sum_{\tilde p}(W_{q,\tilde p} + W_{\tilde p,q})(\omega_q + \omega_{q'} + 2\omega_{\tilde p})(W^*_{\tilde p,q'} + W^*_{q',\tilde p}) + \sum_{\tilde q} V_{\tilde q,q'}V_{q,\tilde q}(\omega_q + \omega_{q'} - 2\omega_{\tilde q})$$

$$\partial_\lambda W_{p,p'} = -W_{p,p'}(\omega_p + \omega_{p'})^2 - \sum_q V_{p,q}(\omega_q + \omega_{p'})(W_{p',q} + W_{q,p'}) + \sum_q V_{p,q}(\omega_p - \omega_q)(W_{q,p'} + W_{p',q})$$

$$\partial_\lambda W^*_{p,p'} = -W^*_{p,p'}(\omega_p + \omega_{p'})^2 - \sum_q V_{q,p}(\omega_q + \omega_{p'})(W^*_{p',q} + W^*_{q,p'}) + \sum_q V_{q,p}(\omega_p - \omega_q)(W^*_{q,p'} + W^*_{p',q})$$

$$\partial_\lambda \varepsilon = -2\sum_{p,p'}(W_{p,p'} + W_{p',p})(\omega_p + \omega_{p'})W^*_{p,p'}$$

Überblick

Main

```
flat = npzread(PATH*name)
om0,V0,W0,eps0 = unpack_arr(flat,N)
om0 = om0 + diag(V0)
V0 = V0 - Diagonal(diag(V0))

u0 = flat_arr(om0,V0,W0,eps0)

tspan = (0.0,30)

prob = ODEProblem(f,u0,tspan,[N])
sol = solve(prob, Tsit5(),saveat=LinRange(tspan[1],tspan[2],n), reltol = 1e-8,
↪  abstol = 1e-8)

npzwrite(SAVEPATH*FILENAME,sol)
npzwrite(SAVEPATH*"sol_full_t"*name,sol.t)
```

$\rightarrow$ sehr langsam :-(

ABER… einige Verbesserungen denkbar und möglich

Hilfsfunktionen

```
using NPZ, DifferentialEquations, LinearAlgebra

function flat_arr(om,V,W,eps)
    return vcat(vec(om),vec(V),vec(W),eps)
end

function unpack_arr(flat,N)
    N = Int(N)
    om0 = flat[1:N]
    V0 = flat[N+1:N+Int(N^2)]
    W0 = flat[N+Int(N^2)+1:N+Int(2*N^2)]
    eps = flat[end]
    V = reshape(V0,(N,N))
    W = reshape(W0,(N,N))
    return om0,V,W,eps
end
```

Definition der ODE

```
function f(u,p,t)
    N = Int(p[1]); om,V,W,eps = unpack_arr(u,N); Wdag = conj(W)

    om_ret = [sum([2*V[q,k]*V[k,q]*(om[k]-om[q])
    ↪   -2*(W[k,q]+W[q,k])*(om[k]+om[q])*(Wdag[q,k]+Wdag[k,q]) for q in 1:N])
    ↪   for k in 1:N]

    V_ret = reshape([-V[q,q_]*(om[q]-om[q_])^2
    ↪   +sum([-(W[(q),p]+W[p,(q)])*(Wdag[p,(q_)]+Wdag[(q_),p])*(om[(q)]+om[(q_)]
    ↪   +2*om[p])+V[p,(q_)]*V[(q),p]*(om[(q)]+om[(q_)] -2*om[p]) for p in 1:N])
    ↪   for q_ in 1:N for q in 1:N],(N,N)).*(ones(N,N)-Diagonal(ones(N)))

    W_ret = reshape([-W[p,p_]*(om[p]+om[p_])^2
    ↪   +sum([-V[p,q]*(om[q]+om[p_])*(W[p_,q]+W[q,p_])
    ↪   +V[p,q]*(om[p]-om[q])*(W[q,p_]+W[p_,q]) for q in 1:N]) for p_ in 1:N for
    ↪   p in 1:N],(N,N))

    eps_ret = -2*sum([(W[p,p_]+W[p_,p])*(om[p]+om[p_])*Wdag[p,p_] for p in 1:N
    ↪   for p_ in 1:N])

    return flat_arr(om_ret,V_ret,W_ret,eps_ret)
end
```

Überblick

Das 1D Bose Polaron I

Fröhlich Hamiltonian:

$$\hat{\mathcal{H}}_F = g_{IB} n_0 + \frac{\hat{p}^2}{2M} + \int \mathrm{d}k \omega_k \hat{a}_k^\dagger \hat{a}_k + \sqrt{\frac{n_0}{2\pi}} g_{IB} \int \mathrm{d}k W_k e^{ik\hat{x}} \left(\hat{a}_k + \hat{a}_{-k}^\dagger \right)$$

Zwei-Phonon Streuung:

$$\hat{\mathcal{H}}_{2\mathrm{ph}} = \frac{g_{IB}}{2\pi} \int \mathrm{d}k \mathrm{d}k' \left(c_k \hat{a}_k^\dagger - s_k \hat{a}_{-k} \right) \left(c_{k'} \hat{a}_{k'} - s_{k'} \hat{a}_{-k'}^\dagger \right) e^{i(k-k')x}$$

Gesamthamiltonian

$$\hat{\mathcal{H}} = \hat{\mathcal{H}}_F + \hat{\mathcal{H}}_{2\mathrm{ph}}$$

ist nicht in unserer rein quadratischen Form von vorhin.

Die LLP- und die Displacement Transformation I

LLP-Transformation: Nutze Erhaltung des *Gesamt*impulses, um in mit der Impurity bewegtes Bezugssystem zu transformieren

$$\hat{U}_{\mathrm{LLP}} := \exp\left(i\hat{x}\cdot\hat{p}_b\right)$$

Hamiltonian ist dann:

$$\begin{aligned}\hat{U}^\dagger_{\mathrm{LLP}}\hat{\mathcal{H}}\hat{U}_{\mathrm{LLP}} := \hat{\mathcal{H}}_{\mathrm{LLP}}(p) &= g_{IB}n_0 + \frac{1}{2M}\left(p - \int \mathrm{d}k k\hat{a}^\dagger_k\hat{a}_k\right)^2 \\ &+ \int \mathrm{d}k\omega_k\hat{a}^\dagger_k\hat{a}_k + \sqrt{\frac{n_0}{2\pi}}g_{IB}\int \mathrm{d}kW_k\left(\hat{a}_k + \hat{a}^\dagger_{-k}\right) \\ &+ \frac{g_{IB}}{2\pi}\int \mathrm{d}k\mathrm{d}k'\left(c_k\hat{a}^\dagger_k - s_k\hat{a}_{-k}\right)\left(c_{k'}\hat{a}_{k'} - s_{k'}\hat{a}^\dagger_{-k'}\right)\end{aligned}$$

Die LLP- und die Displacement Transformation II

Displacement Transformation:

$$\hat{D}(\underline{\alpha}) := \exp\left(\sum_k \alpha_k \hat{a}_k^\dagger - \text{h.c.}\right) = \exp\left(-\underline{\alpha}^\dagger \underline{\underline{\Omega}}\;\underline{\hat{a}}\right)$$

Wirkung auf Erzeugungs- und Vernichtungsoperatoren:

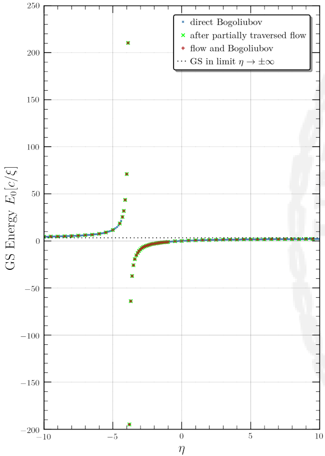
$$\hat{D}^\dagger(\underline{\alpha})\hat{a}_{k_i}\hat{D}(\underline{\alpha}) = \hat{a}_{k_i} + \alpha_{k_i}$$
$$\hat{D}^\dagger(\underline{\alpha})\hat{a}_{k_i}^\dagger\hat{D}(\underline{\alpha}) = \hat{a}_{k_i}^\dagger + \alpha_{k_i}^*$$

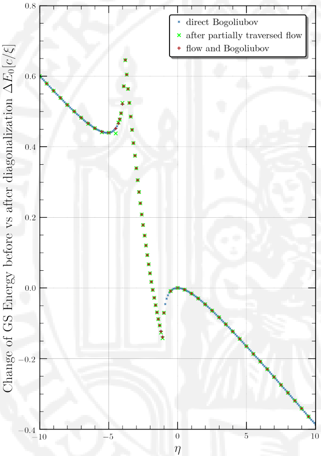
$\rightarrow$ transformiere lineare Terme weg durch geeignete Wahl von $\underline{\alpha}$

Überblick

Ergebnisse- was haben wir gelernt? I

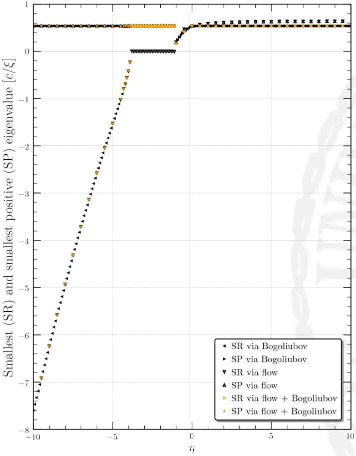
- anfängliche Symmetrie $k \leftrightarrow -k$ des Hamiltonians verhindert Konvergenz der Flow Equations zum exakten Spektrum
  $\rightarrow$ führe twisted boundary conditions ein

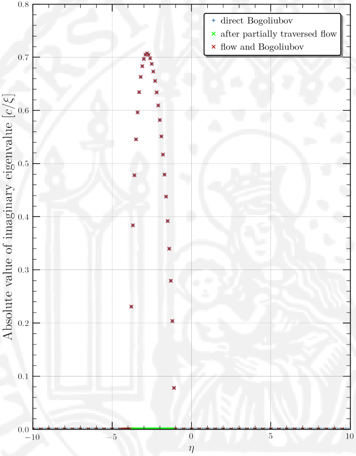




Ergebnisse- was haben wir gelernt? II

- Verschiebung des $k-$grids bricht Symmetrie, Konvergenz zum exakten Spektrum des Hamiltonian wird erreicht





Überblick

Problemstellung

Der Flow Equation Approach

Flow Equations im rein quadratischen Fall

Implementierung des rein quadratischen Falls

Der physikalische Rahmen
- Das 1D Bose Polaron
- Die LLP- und die Displacement Transformation

Ergebnisse- was haben wir gelernt?

Was kommt jetzt? - Ausblick

Was kommt jetzt? - Ausblick

a

Und nun her mit den Fragen :-)