Electromagnetic properties of nuclei: from few- to many-body systems

Lecture 5

Integral Transforms

- Continued -

Sonia Bacca

November 22nd, 2017

Lecture series for SFB 1245
TU Darmstadt

Best kernel

A **“good” Kernel** has to satisfy **two** requirements

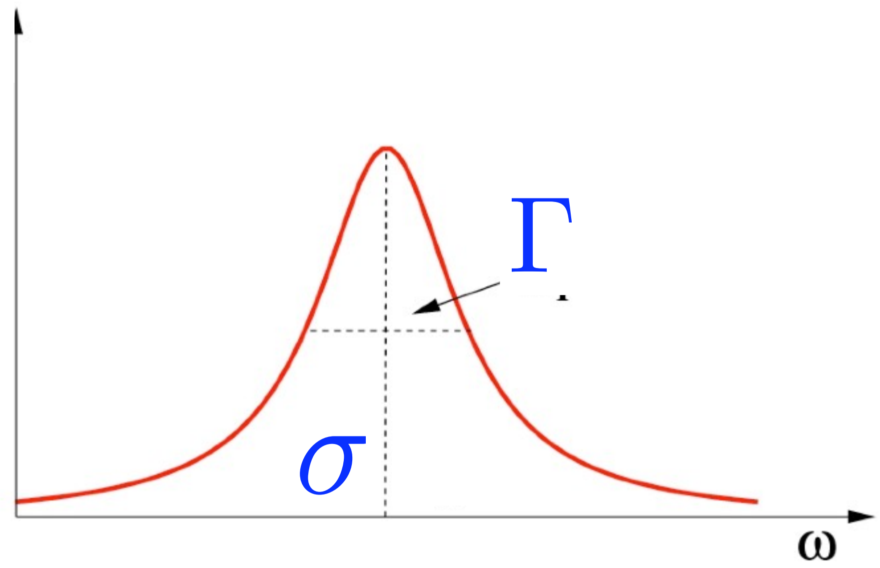
1) one must be able to calculate the integral transform

2) one must be able to invert the transform minimizing uncertainties

Which is the best kernel?

… a representation of the δ-function

Lorentzian kernel



$$K(\omega, \sigma, \Gamma) = \Gamma/\pi \left[(\omega - \sigma)^2 + \Gamma^2 \right]^{-1}$$

It is a representation of the δ-function

$$L(\sigma, \Gamma) = \frac{\Gamma}{\pi} \int d\omega \frac{R(\omega)}{(\omega - \sigma)^2 + \Gamma^2}$$

Lorentz Integral Transform (LIT) Efros, *et al.*, JPG.: Nucl.Part.Phys. **34** (2007) R459

Illustration of requirement N.1:
One can calculate the integral transform

Lorentz Integral Transform

$$\boxed{L(\sigma,\Gamma) = \langle \psi_0 | \Theta^\dagger K(H - E_0, \sigma, \Gamma) \Theta | \psi_0 \rangle}$$

$$K(\omega,\sigma,\Gamma) = \frac{\Gamma}{\pi} \frac{1}{(\omega-\sigma)^2 + \Gamma^2}$$

$$K(\omega,\sigma,\Gamma) = \frac{\Gamma}{\pi} \frac{1}{(\omega-\sigma-i\Gamma)(\omega-\sigma+i\Gamma)}$$

$$L(\sigma,\Gamma) = \left\langle \psi_0 | \Theta^\dagger \frac{1}{H - E_0 - \sigma - i\Gamma} \underbrace{\boxed{\frac{1}{H - E_0 - \sigma + i\Gamma} \Theta | \psi_0}}_{= \left|\tilde\psi\right\rangle} \right\rangle \frac{\Gamma}{\pi}$$

$$= \langle \tilde\psi | \tilde\psi \rangle \frac{\Gamma}{\pi}$$

Lorentz Integral Transform

main point of the LIT :

Schrödinger-like equation with a source

$$\boxed{(H - E_0 - \sigma + i\Gamma)|\tilde{\Psi}\rangle = \Theta\, |\Psi_0\rangle}$$

- Due to imaginary part Γ the solution $|\tilde{\psi}\rangle$ is unique
- Since rhs is finite, $|\tilde{\psi}\rangle$ has bound state asymptotic behaviour

⇓

Can solve it with bound state methods

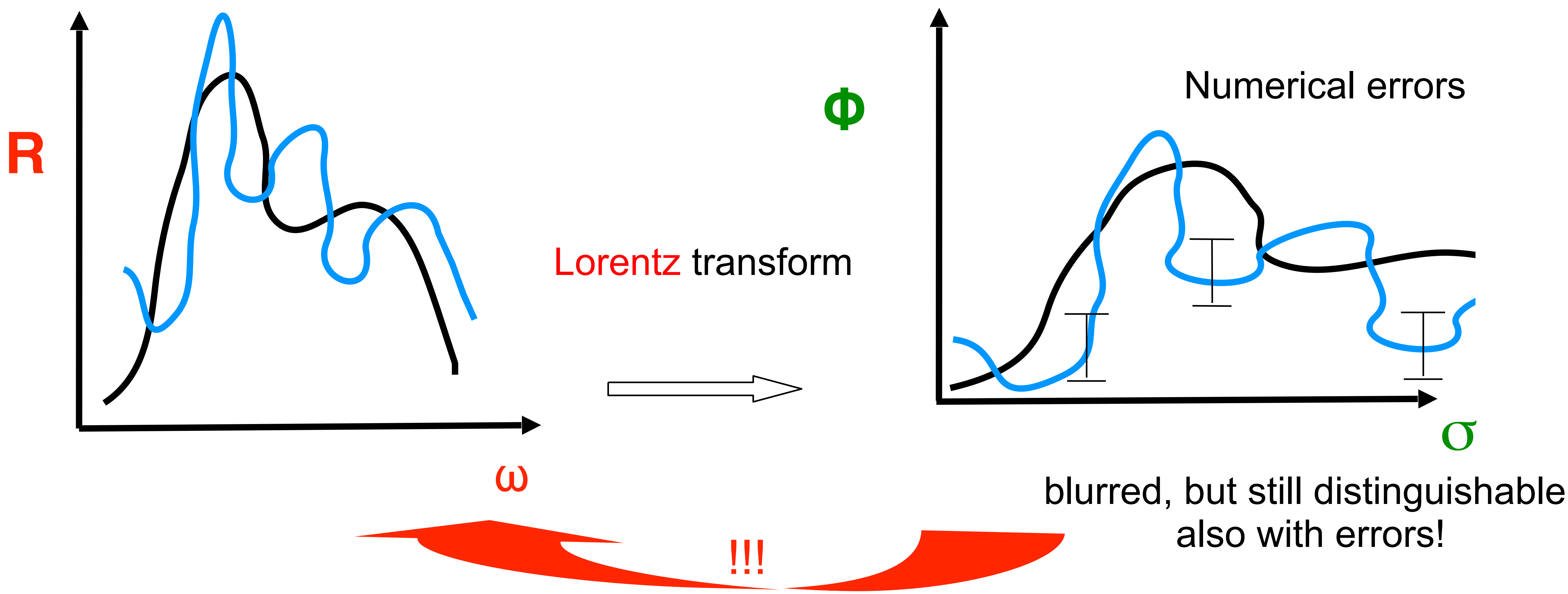
Efros, *et al.*, JPG.: Nucl.Part.Phys. **34** (2007) R459

Illustration of requirement N.2: One can invert the integral transform minimizing uncertainties

Inversion

How can one easily understand why the inversion is
much less problematic?

Inversion: e.g. “regularization method” at fixed width



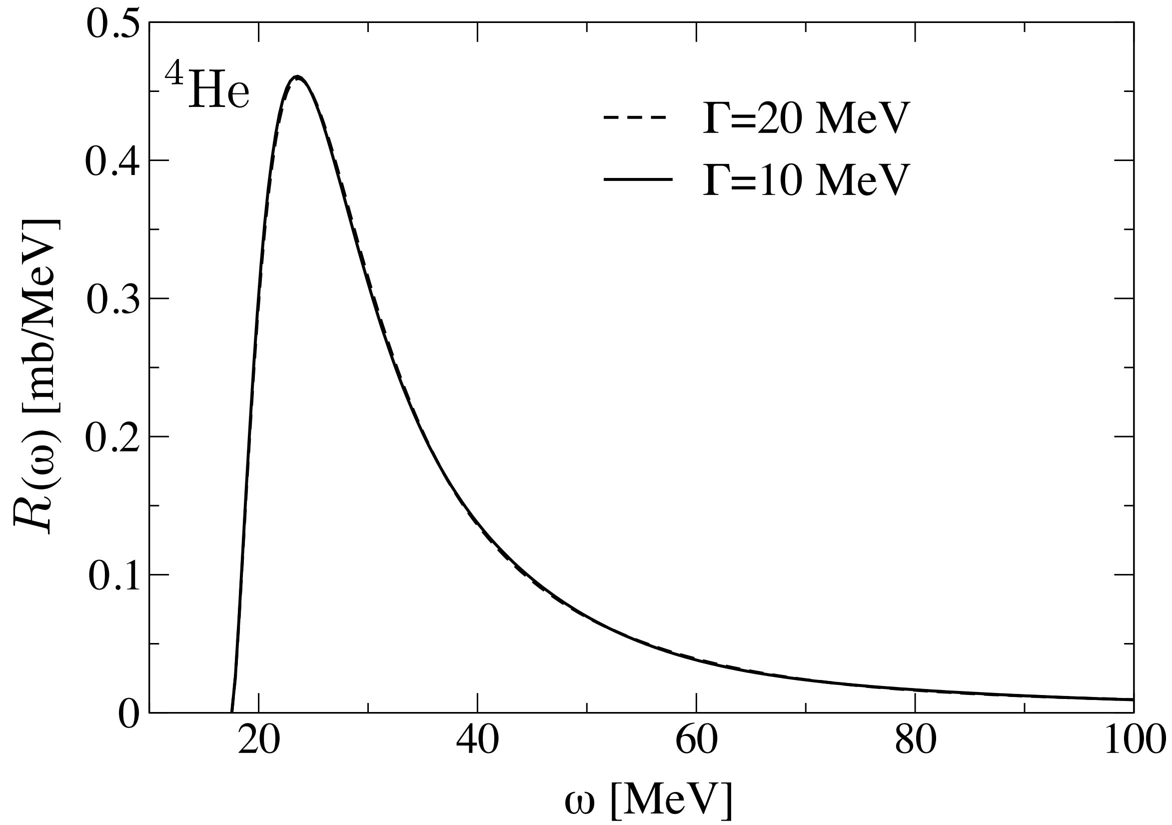
Regularization method

(from A.I N.Tikhonov, "Solutions of ill posed problems", Scripta series in mathematics (Winston,1977).

$$R(\omega) = \sum_i^{I_{\max}} c_i \chi_i(\omega, \alpha) \quad \rightarrow \quad L(\sigma, \Gamma) = \sum_i^{I_{\max}} c_i \mathcal{L}[\chi_i(\omega, \alpha)]$$

$$\chi_i(\omega, \alpha) = \omega^{3/2} \exp\left(\alpha_{em} Z_1 Z_2 \sqrt{\frac{2\mu}{\omega}}\right) \cdot e^{-\frac{\omega}{\alpha i}}$$

Least square fit of the coefficients c_i to reconstruct the response function



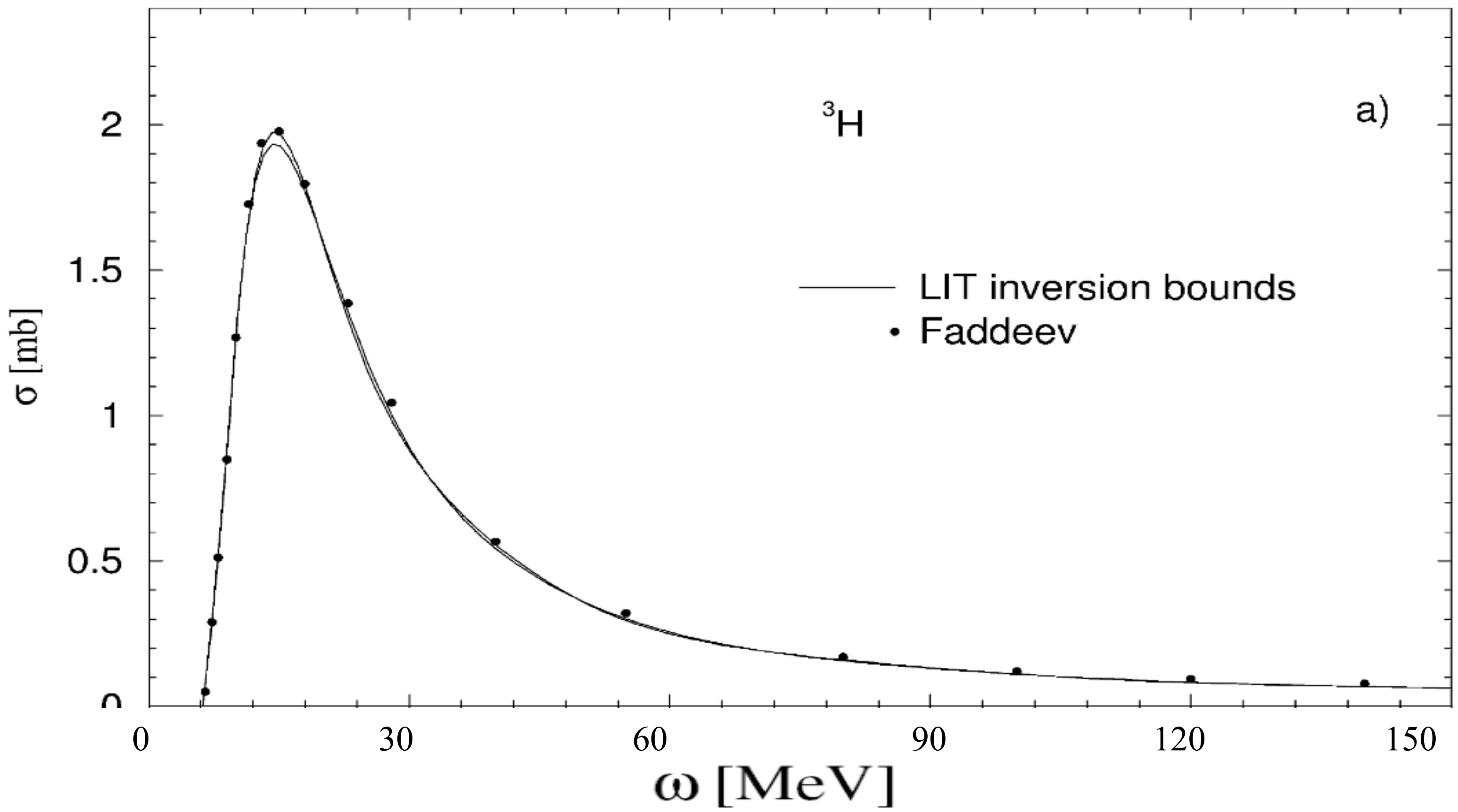
Other methods, see Mirko Miorelli's talk

Benchmarks

The LIT method has been benchmarked with other few-body methods where $|\psi_f\rangle$ is calculated directly using same dynamical ingredients

With Fadeev approach

Nucl.Phys. A707 365 (2002)

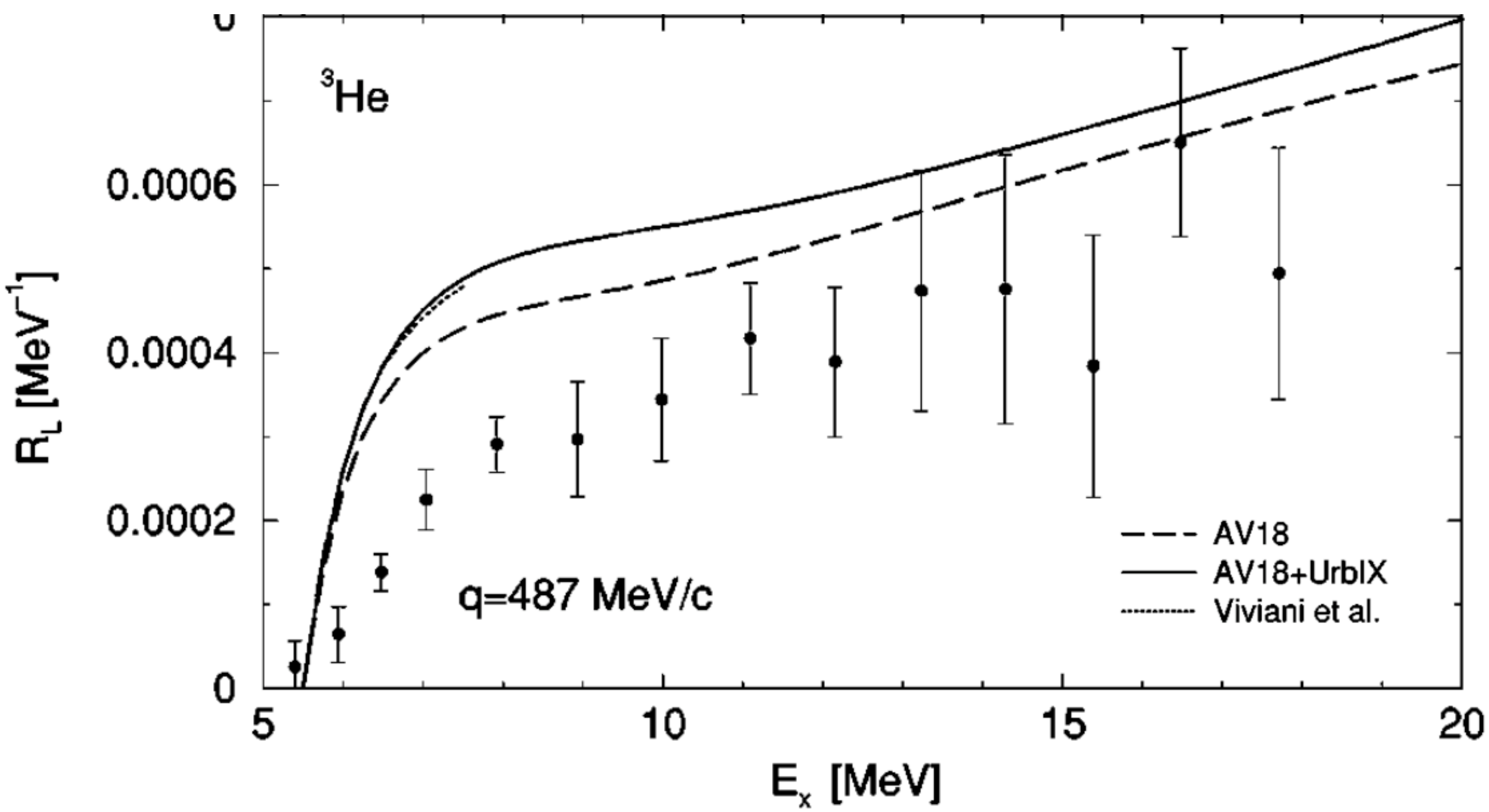


Benchmarks

The LIT method has been benchmarked with other few-body methods where $|\psi_f\rangle$ is calculated directly using same dynamical ingredients

With variational approach

Phys.Rev.C 69 (2004)



Other remarks on the LIT

Rewriting the response function

$$R(\omega)=\sum_f |\langle f|\Theta|0\rangle|^2 \boxed{\delta(\omega-E_f-E_0)}$$

NB: often interchange
Notation for g.s. and final states

Sokhotski formula

$$\frac{1}{x+i\epsilon}=\mathcal{P}\int dx\frac{1}{x}-i\delta(x)\pi \qquad\qquad \epsilon\rightarrow 0$$

Taking the imaginary part only

$$\mathrm{Im}\frac{1}{x+i\epsilon}=-\delta(x)\pi \qquad \Rightarrow \qquad \delta(x)=-\frac{1}{\pi}\mathrm{Im}\frac{1}{x+i\epsilon}$$

$$R(\omega)=\boxed{-\frac{1}{\pi}\mathrm{Im}}\left[\sum_f |\langle f|\Theta|0\rangle|^2 \boxed{\frac{1}{\omega-E_f-E_0+i\epsilon}}\right]$$

Rewriting the response function

$$R(\omega)=-1/\pi\ \mathrm{Im}\left[\Sigma_f\langle 0|\Theta^+|f\rangle\langle f|\,\Theta\,|0\rangle\ (\omega-E_f+E_0+\imath\varepsilon)^{-1}\right]$$

$$=-1/\pi\ \mathrm{Im}\left[\Sigma_f\langle 0|\,\Theta^+(\omega-E_f+E_0+\imath\varepsilon)^{-1}|f\rangle\langle f|\Theta\,|0\rangle\right]$$

$H|f\rangle=E_f|f\rangle$

$$=-1/\pi\ \mathrm{Im}\left[\Sigma_f\langle 0|\,\Theta^+(\omega-H+E_0+\imath\varepsilon)^{-1}|f\rangle\langle f|\Theta\,|0\rangle\right]$$

change sign

$$=1/\pi\ \mathrm{Im}\left[\Sigma_f\langle 0|\,\Theta^+(H-\omega-E_0-\imath\varepsilon)^{-1}|f\rangle\langle f|\Theta\,|0\rangle\right]$$

$\Sigma_f|f\rangle\langle f|=1$ and change sign

$$=-1/\pi\ \mathrm{Im}\left[\langle 0|\,\Theta^+(H-\omega-E_0+\imath\varepsilon)^{-1}\,\Theta\,|0\rangle\right]$$

Like a Green's function with poles on the real axis

Rewriting the LIT

$$L(\sigma, \Gamma) = \Gamma/\pi \int [(\omega - \sigma_R)^2 + \Gamma^2]^{-1} R(\omega)\, d\omega$$

$$= \Gamma/\pi \int d\omega \; [(\omega - \sigma_R)^2 + \Gamma^2]^{-1} \sum_f |\langle f| \Theta |0\rangle|^2 \delta(\omega - E_f + E_0)$$

Integrate delta and use $H|f\rangle = E_f|f\rangle$

Completeness

$$= \Gamma/\pi \sum_f \langle 0 | \Theta^+ [(H - E_0 - \sigma_R)^2 + \Gamma^2]^{-1} |f\rangle \langle f| \Theta | 0 \rangle$$

$$= \Gamma/\pi \langle 0 | \Theta^+ [(H - E_0 - \sigma_R)^2 + \Gamma^2]^{-1} \Theta | 0 \rangle$$

$-\mathrm{Im}\, [(H - E_0 + \sigma_R + i\Gamma)^{-1}] =$

$-\mathrm{Im}[(H - E_0 - \sigma_R + i\Gamma)^{-1} (H - E_0 - \sigma_R - i\Gamma)^{-1} (H - E_0 - \sigma_{Ro} - i\Gamma)] =$

$= \Gamma\, [(H - E_0 - \sigma_R)^2 + \Gamma^2]^{-1}$

Finite, not infinitesimal

$$= -1/\pi\, \mathrm{Im}\, [\langle 0 | \Theta^+ (H - E_0 - \sigma_R + i\Gamma)^{-1} \Theta | 0 \rangle]$$

Summarizing

$$R(\omega) = -1/\pi \ \mathrm{Im}\left[\langle 0|\, \Theta^{+}(H - \omega - E_0 + i\varepsilon)^{-1}\, \Theta\, |0\rangle\right]$$

ε infinitesimal

$$L(\sigma, \Gamma) = -1/\pi \ \mathrm{Im}\left[\langle 0 |\, \Theta^{+}\, (H - E_0 - \sigma_R + i\,\Gamma)^{-1}\, \Theta\, | 0\rangle\right]$$

Γ finite, not infinitesimal

Of course, when $\varepsilon = \Gamma$ then $R(\omega) = L(\sigma, \Gamma)$
That is indeed the case where the Kernel is the delta function

However, due to the fact that Γ is finite and $L(\sigma, \Gamma)$
is finite, one is allowed to use bound -state techniques to calculate it

Lanczos Algorithm

Algorithm used to tri-diagonalize matrices $H \longrightarrow H_{tr}$

1) Choose first Lanczos vector $|\phi_0\rangle$

2) Use recursive definition to find the other Lanczos vectors

$$b_{n+1} |\phi_{n+1}\rangle = H |\phi_n\rangle - a_n |\phi_n\rangle - b_n |\phi_{n-1}\rangle$$

With $a_n = \langle\phi_n| H |\phi_n\rangle$

$$b_n = \| b_n |\phi_n\rangle \|$$

3) Matrix represented on the Lanczos vectors is tridiagonal

$$H_{tr} = \begin{pmatrix} a_0 & b_1 & 0 & 0 & \dots \\ b_1 & a_1 & b_2 & 0 & \dots \\ 0 & b_2 & a_2 & b_3 & \dots \\ 0 & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \end{pmatrix}.$$

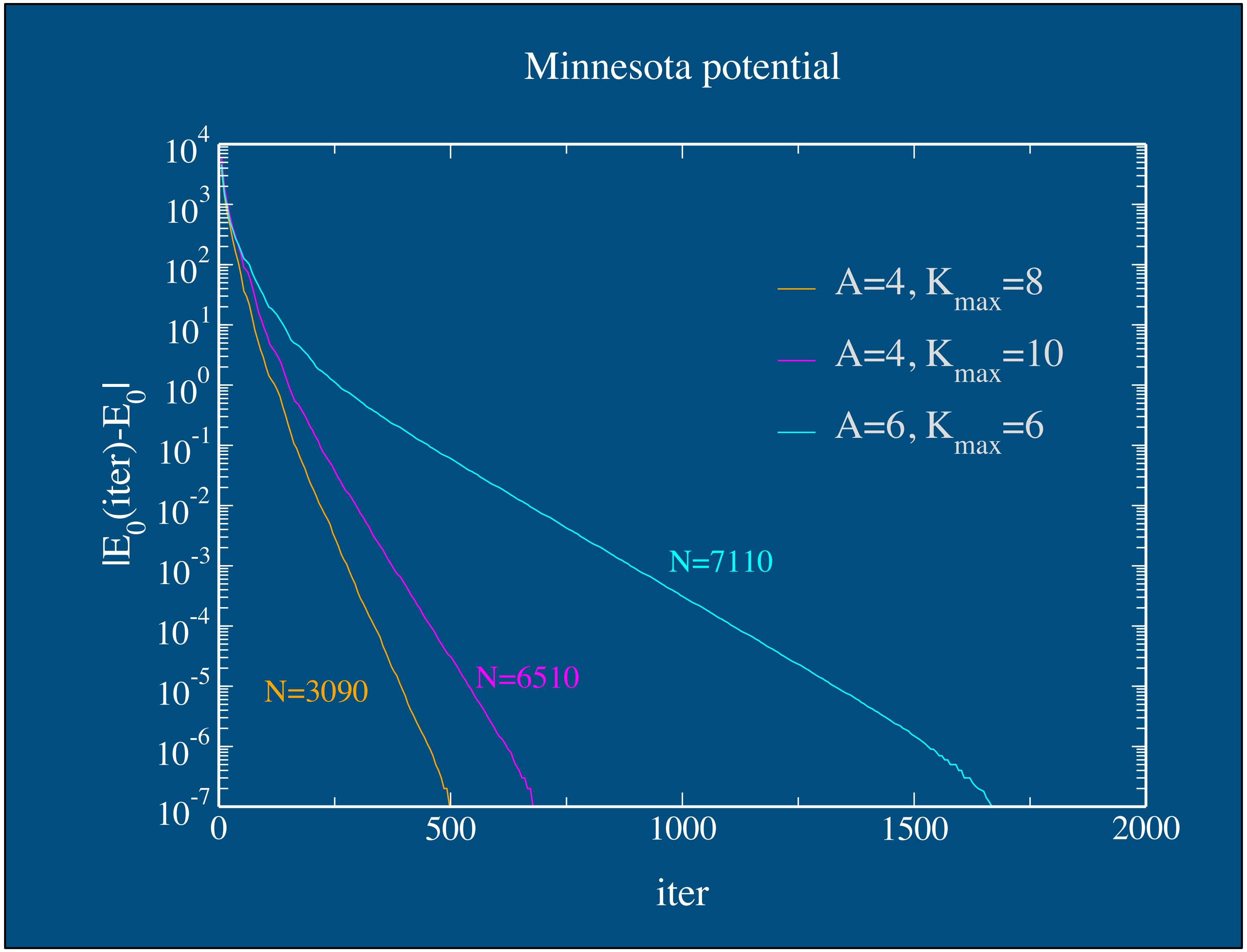
Can diagonalize it using Numerical Recipes routine, e.g. TQLI

Lanczos Algorithm

For large scale eigenvalue problems

Generally, diagonalizing a matrix is a N^3 operation

With the Lanczos algorithm you can reduce it to nN^2 with $n = \max(\text{iter}) < N$



Lanczos Algorithm

The algorithm can be also used calculate the LIT

$$L(\sigma, \Gamma) = -1/\pi \, \mathrm{Im} \left[\langle 0 | \Theta^+ \, (H - \underline{E_0 - \sigma_R + i\Gamma})^{-1} \, \Theta \, | 0 \rangle \right]$$

Using the Lanczos algorithm one can represent $(H - E_0 - \sigma_R + i\sigma_I)^{-1}$ as a continuum fraction of the Lanczos coefficients

1) Choose first Lanczos vector $$|\phi_0\rangle = \frac{\Theta|0\rangle}{\sqrt{\langle 0|\Theta^\dagger \Theta|0\rangle}}$$

2) After applying the recursive definition you obtain

$$L(\sigma) = -\frac{1}{\pi} \langle 0| \Theta^\dagger \Theta |0\rangle \, Im \left\{ \frac{1}{(z - a_0) - \frac{b_1^2}{(z-a_1) - \frac{b_2^2}{(z-a_2) - b_3^2 \ldots}}} \right\}$$

Efros, *et al.*, JPG.: Nucl.Part.Phys. **34** (2007) R459

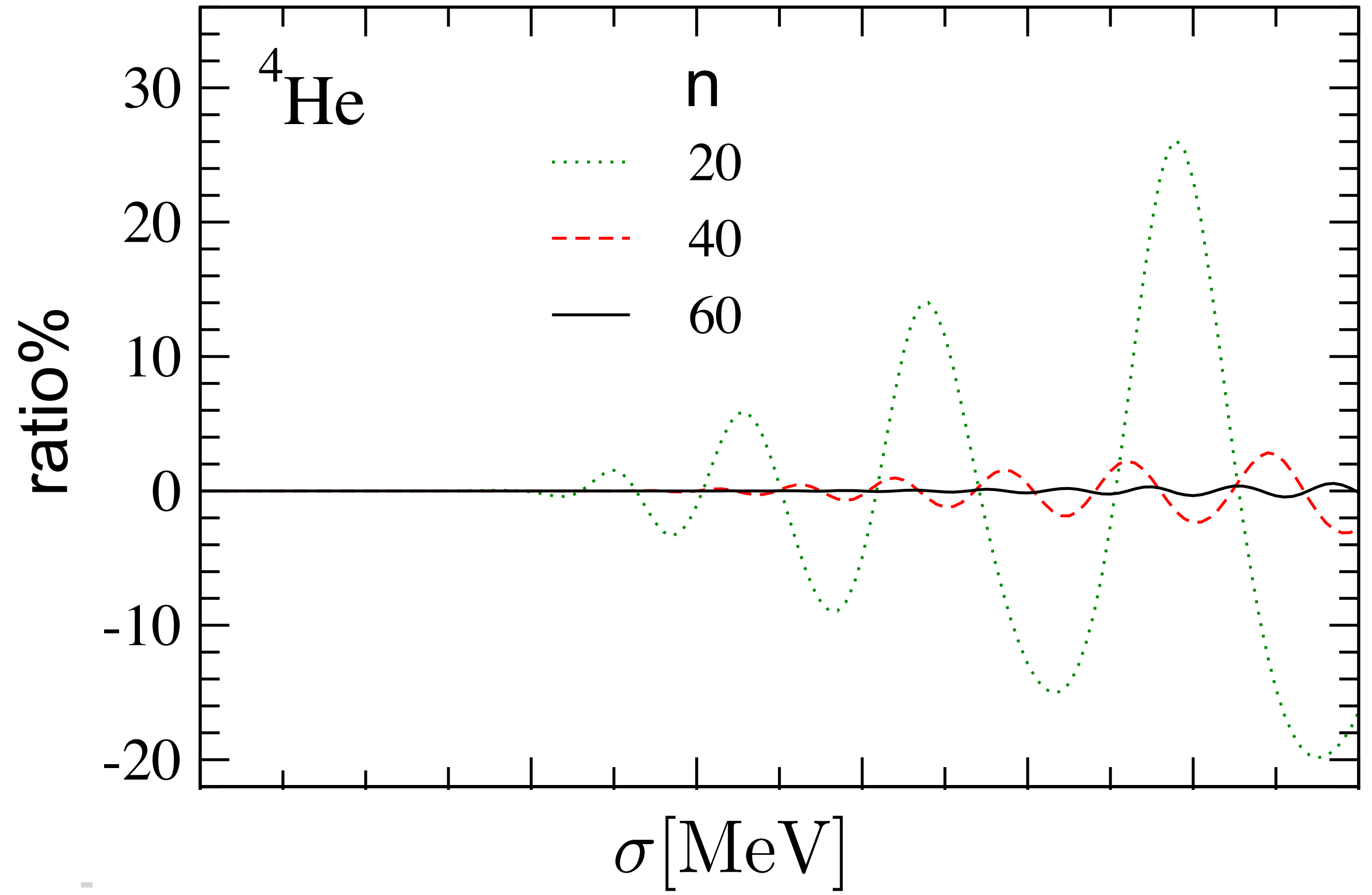
Lanczos Algorithm

Advantages

The Lanczos algorithm involves just a matrix-vector multiplication (N^2)

Continues fractions converge fast

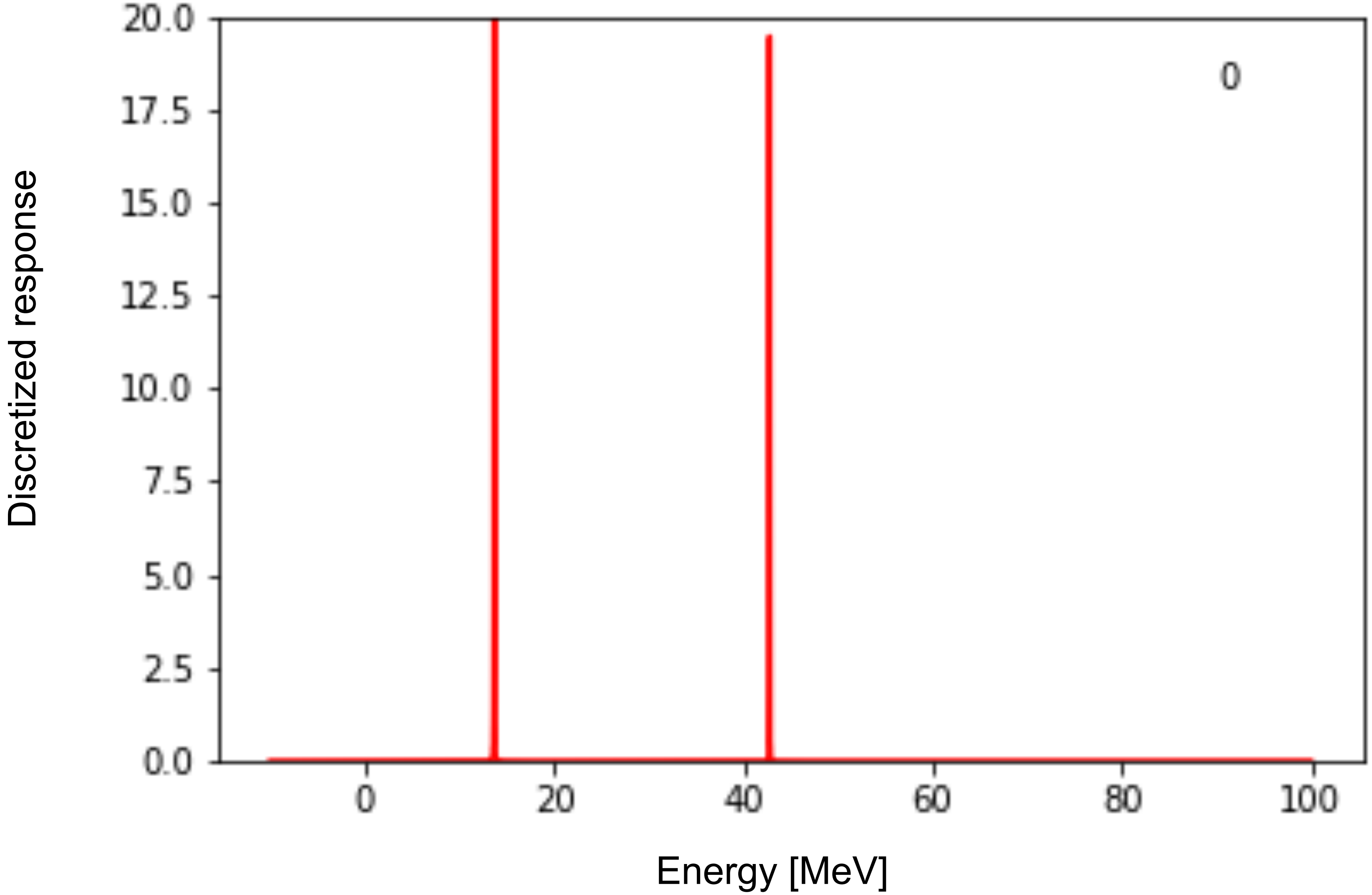
Again, with the Lanczos algorithm the computational load is becoming nN^2 with n = max(iter)< N



Lanczos Algorithm

Strength building up from Lanczos vectors

Movie from M.Miorelli



Other computational aspects

Parallel algorithm

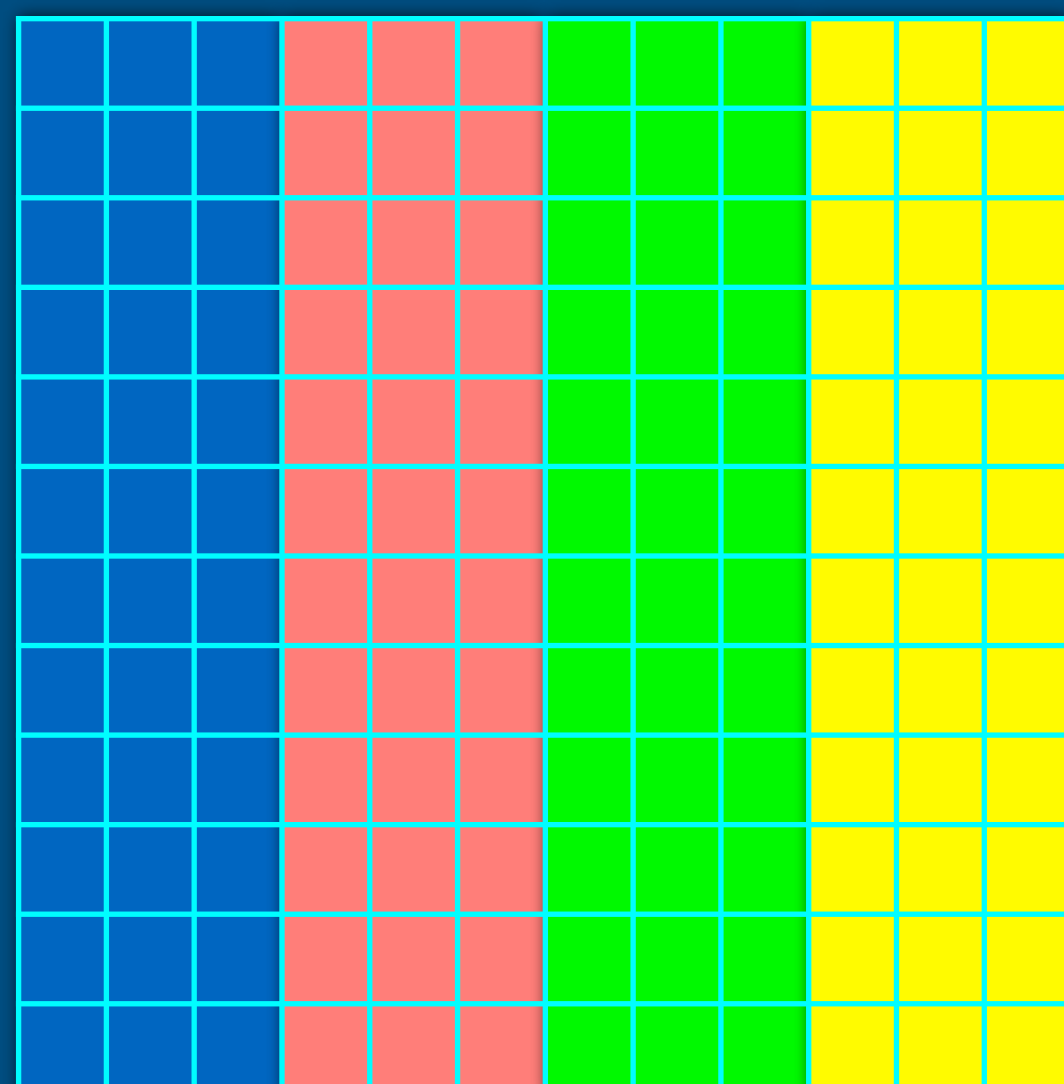
50 to 90% of the CPU times is spent in the Lanczos algorithm +
matrices become too large to be loaded in the memory of a single core

It is wise to distribute the load (memory and computation) on different cores

Example

PE1
PE2
PE3
PE4

H =



This can lead to unbalance among the threads

Other computational aspects

Parallel algorithm

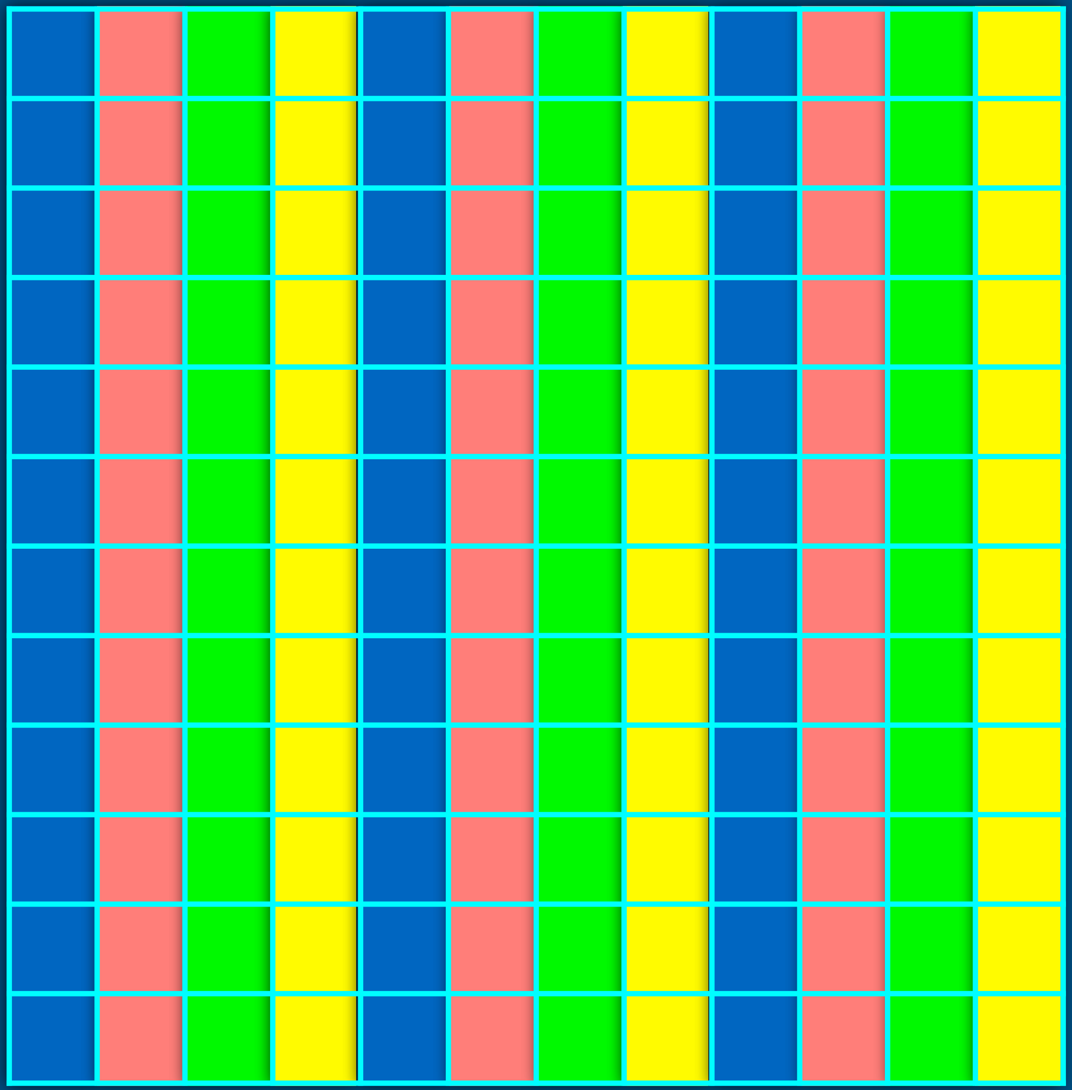
50 to 90% of the CPU times is spent in the Lanczos algorithm +
matrices become too large to be loaded in the memory of a single core

It is wise to distribute the load (memory and computation) on different cores

Example

PE1
PE2
PE3
PE4

H =



More balanced distribution

Other computational aspects

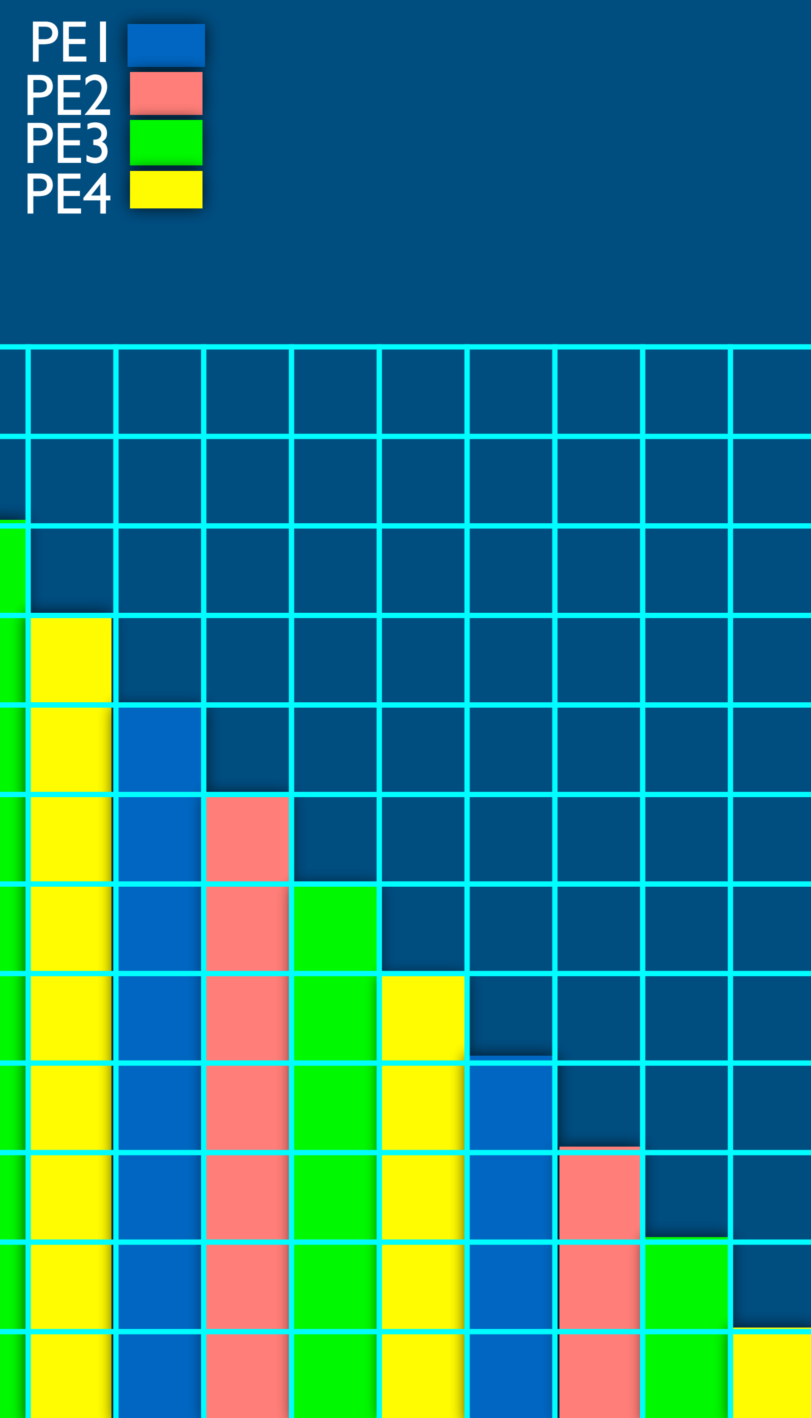
Parallel algorithm

50 to 90% of the CPU times is spent in the Lanczos algorithm +
matrices become too large to be loaded in the memory of a single core

It is wise to distribute the load (memory and computation) on different cores

Example

H =



Possible Improvement:
Can save a factor of 2
using hermiticity of **H**

Other computational aspects

Parallel algorithm

"Scaling" of the problem

Scaling means that at a constant problem size the parallel speedup increases linearly with the number of used cores

Speedup $S_c = \frac{T_1}{T_c}$

T_1: Time needed for a sequential algorithm

T_c: Time needed for a parallel algorithm with c cores

Ideal situation $T_c = \frac{T_1}{c}$ $\qquad S_c = c$ linear speedup

limited by algorithm and by communications among threads

Matrix dimension ~40000