Lectures on Numerical Methods

TCM, Lent 2014

Topics covered:
- Exact diagonalization (Lect 1,2)
- Tensor-Product Methods (Lect 3,4)
- Diagrammatic Monte-Carlo (Lect. 5,6)

Lecture 1: Introduction to exact diagonalization

- Why ED: example problems.
- Basic Ingredients for ED
- Posing the problem in quantum mechanics
- The basic algorithms: Full diagonalization, Lanczos.
- Hilbert-space & Hamiltonian in Brief.

Why ED:

- Want to solve quantum mechanical problems that are 'too large' to be solved fully but provide insight into behaviour of a large system
  ⇒ Interacting Quantum Systems

ex:
- Spins on lattices — Heisenberg $\sum_{\langle i,j\rangle} (\vec{S}_i \cdot \vec{S}_j)$
  mostly relevant to 2D (otho alg. in 1D!)
  Complete later
  $d \sim 10^9$: 42 spins on hex.
  $d \sim 10^{11}$: 20 sites at $n=\frac{1}{2}$ (Hubb).
  $d \sim 10^8$: 32 sites for t-J with 4 holes.
  $d \sim 10^9$: 20 el. depon
- Hubbard / t-J models $\sum_{\langle i,j\rangle} \vec{S}_i \vec{S}_j + P c_i^\dagger c_j P + u \sum n_i(n_i-1)$
- FQ HE: interactions with quenched kinetic energy. (Bosons, fermions)
- few particle systems:
  - quantum-dots
  - few atoms in harmonic trap
  - chemistry (FCI)

Basic Ingredients:

Want to solve Schrödinger's equation

$$H|\psi\rangle = E|\psi\rangle$$

- Choose a finite site system / truncated basis to write $|\psi\rangle$ explicitly

$$|\psi\rangle = \sum_{\mu} \kappa_\mu |\mu\rangle$$

$|\mu\rangle$: microscopic quantum state in configuration-space

$$S.E. \rightarrow \underline{H_{\mu\nu}\kappa_\nu = E\kappa_\mu} \Rightarrow \text{pb. reduced to linear algebra.}$$

Typical micro states: enumerating the Hilbert-space.

* spin configurations in a set of $S = 1/2$ spins

$$|\mu\rangle = |\uparrow\downarrow\uparrow\downarrow\ldots\uparrow\uparrow\rangle$$

N spins $\Rightarrow$ $d = 2^N$ states.

* particles in orbitals ϕ_i, with creation op's $\hat{a}_i^\dagger$
  → single-particle basis $i = 1, \ldots, N_{orb}$.

  states in Fock space

$$|\mu\rangle = \prod_i \left(\hat{a}_i^\dagger\right)^{n_i(\mu)} |\text{vac.}\rangle$$

$$= |n_0\ n_1\ \ldots\ n_{N_{orb}}\rangle$$

typically $\sum n_i = N$ conserved.

dimension (no symmetries) $d = \binom{N_{orb}}{N} = \frac{N_{orb}!}{N!(N_{orb}-N)!}$

Basic crux: $d \sim \exp(N)$.

- need efficient algorithm that scales slowly with d.
- achievable size then limited by memory (states, Hamiltonian)

(a) For small enough matrices $d \lesssim 10^5$ / supercomputer.
$d \sim 10^4$ on workstation.

Full diagonalization: LAPACK - householder rep

time $t \propto d^3$

memory $M \propto d^2$

⇒ both computer-time and memory present challenge
⇒ cannot benefit from sparse Hamiltonian matrix.

Benefit: get entire spectrum + eigenvalues

⇒ can calculate full thermodynamics!

$$Z = \sum_n e^{-\beta \varepsilon_n}$$

known exactly!

(b) Partial diagonalization

calculate only a selected subset of states
using iterative methods such as Lanczos algorithm

time $t \sim d$

memory $M \sim d$

The Lanczos Algorithm

idea: construct a basis, in which the eigenstates (values) can be calculated approximately.

- Krylov-space $|\psi\rangle, H|\psi\rangle, H^2|\psi\rangle, \ldots, H^m|\psi\rangle$
- ensure orthogonality, i.e. Schmidt-reorthogonalize this basis

Lanczos' algorithm: (assume H real symmetric, for now)

- starting vector $|\varphi_0\rangle$, normalized
  - needs to have projection onto EV to be calculated
  - can be random, in absence of further knowledge
- three-vector recursion: $(\beta_0 = 0)$

$$|\phi'\rangle = H|\varphi_n\rangle - \beta_n |\varphi_{n-1}\rangle$$

only place where matrix-vector mult. comes into play!

define $\alpha_n = \langle\varphi_n|\phi'\rangle$

$$|\phi''\rangle = |\phi'\rangle - \alpha_n |\varphi_n\rangle$$

define $\beta_{n+1} = \sqrt{\langle\phi''|\phi''\rangle}$

next vector: $|\varphi_{n+1}\rangle = \frac{1}{\beta_{n+1}} |\phi''\rangle$

proof ←

- this set of instructions re-orthogonalizes the new vector $H|\varphi_n\rangle$ with respect to the prior two vectors (normalization)

$$\beta_{n+1}|\varphi_{n+1}\rangle = H|\varphi_n\rangle - \overbrace{\langle\varphi_n|H|\varphi_n\rangle}^{\alpha_n}|\varphi_n\rangle - \langle\varphi_{n-1}|H|\varphi_n\rangle|\varphi_{n-1}\rangle$$

where we also have $\beta_{n+1} = \langle\varphi_{n+1}|H|\varphi_n\rangle$ (β_n)

(in fact, $\beta_n = \bar{\beta}_n$ real!)

(4a)

proof: $\beta_{n+1} = \| |\varphi''\rangle \|$

$$|\varphi_{n+1}\rangle = \frac{|\varphi''\rangle}{\| |\varphi''\rangle \|}$$

$$\| |\varphi''\rangle \| \; \langle \varphi_{n+1} | \varphi_{n+1} \rangle = \langle \varphi_{n+1} | \varphi'' \rangle$$

$$= \langle \varphi_{n+1} | \left(H|\varphi_n\rangle - \alpha_j |\varphi_n\rangle - \beta_j |\varphi_{n-1}\rangle \right)$$

$$= \langle \varphi_{n+1} | H | \varphi_n \rangle =$$

$$= \langle \varphi_n | H | \varphi_{n+1} \rangle^* = \beta_{n+1}^*$$

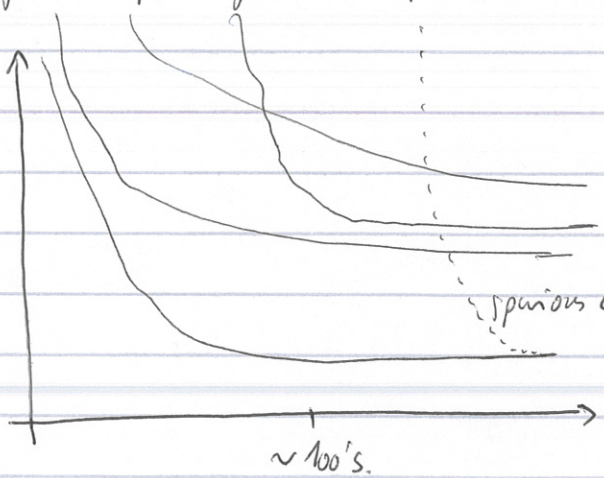
$\Rightarrow$ $\beta_{n+1} = \| |\varphi''\rangle \|$ real.

- it can be shown that a newly constructed vector is orthogonal to all prior ones. Hence, the ~~Hamiltonian~~ takes ~~the~~ Banded form in the Krylov basis:

$$T = \tilde{H} = \begin{pmatrix} \alpha_0 & \beta_1 & & & \\ \beta_1 & \alpha_1 & \beta_2 & & \\ & \beta_2 & \alpha_2 & \beta_3 & \\ & & \beta_3 & \alpha_3 & \ddots \end{pmatrix}$$

[in practice, when computed with finite precision, vectors may become non-orthogonal. In this case, it is possible to re-orthogonalize new vector with respect to the previous ones explicitly.]

Convergence of eigenvalues of T to EV's of H is fast:



- fastest for extremal EV
- EV converged when last entry of corresponding ~~Krylov~~ of T vanishes.
- heuristic: $\hat{T}_2$ test.

- due to non-orthogonality, spurious EV's occur, so multiple degenerate EVals are seen
  ⇒ method cannot be used to determine multiplicities.

Fixes:
- Reorthogonalization
- Banded "Block Lanczos algorithm"

- Convergence is inversely prop. to the spectral gap above the corresponding eigenvalue.

(5a)

- Proof of orthogonality (by induction)

$\langle \varphi_0 | \varphi_0 \rangle = 1$, $\langle v_1^{\perp} | \varphi_0 \rangle = 0$

assume that $\langle \varphi_i | \varphi_j \rangle = \delta_{ij}$ $i, j \leq k$

$$\langle \varphi_j | \varphi_{k+1} \rangle \beta_{k+1} = \langle \varphi_j | A | \varphi_k \rangle - \alpha_k \underbrace{\langle \varphi_j | \varphi_k \rangle}_{\text{By construction } = 0} - \beta_k \underbrace{\langle \varphi_j | \varphi_{k-1} \rangle}_{=0}$$

$$= \left(\beta_{j+1} \langle \varphi_{j+1} | + \alpha_j \langle \varphi_j | + \beta_j \langle \varphi_{j-1} | \right) | \varphi_k \rangle$$

$$= \underbrace{\beta_{j+1} \langle \varphi_{j+1} | \varphi_k \rangle}_{= 0 \text{ if } j \leq k-1 \text{, } = \text{RHS if } j = k.} + \alpha_j \underbrace{\langle \varphi_j | \varphi_k \rangle}_{=0} + \beta_j \underbrace{\langle \varphi_{j-1} | \varphi_k \rangle}_{=0}$$

- Furthermore, T is the projection of H onto the space of Lanczos vectors $V_m = \{ |\varphi_0\rangle, |\varphi_1\rangle, \ldots |\varphi_m\rangle \}$

$$T_m = V_m^{\dagger} H V_m .$$

Extracting eigenvectors: get EV's $\underline{x}^n$ of tri-diagonal T

$$T_{ij}\, \underline{x}_j^n = E_n \underline{x}_i^n \qquad , i = 0, \ldots, m-1.$$

$$|\psi_n\rangle = \sum x_i^n |\varphi_i\rangle \qquad \text{"Ritz-vector"}.$$

$|\varphi_i\rangle$ can be stored on memory / disk and recovered by re-starting the Lanczos run from the same initial vector (for very large calculations).

Algorithmic requirements:

Memory:
- 3 d elements of vector
- further vectors can be stored on disk (optional)
- tri-diagonal form T of H is small.
- Hamiltonian matrix need not be stored; can calculate on the fly.

Time:
- $C(N) \cdot d$ operations for matrix-vector mult. per iteration
- scalar products + additions $\propto d$.

$\Rightarrow$ $M \sim d$ $\qquad$ $t \sim d$

Library routines:

e.g. ARPACK

books:
- Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, H v. d. Vorst "Templates for the Solution of Algebraic EV Problems"
- Jane Cullum & Ralph Willoughby: "Lanczos Algorithms for Large Symmetric Eigenvalue Computations" (1985)

Remaining tasks / ingredients:

① represent / enumerate all configurations / basis states in Hilbert-space.

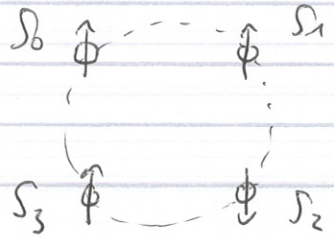
② act with Hamiltonian-matrix on a given vector.

①: Hilbert-space

- use bitwise representation of state

* spin $\frac{1}{2}$ $\quad \uparrow \;(\Leftrightarrow)\; 1$
$\quad \downarrow \;(\Leftrightarrow)\; 0$

map bits of a word to sites on chain / lattice



$\Leftrightarrow$ "1 1 0 1"

* spin 1: ternary, or code on 2 bits / spin

* fermions: occupation number 0, 1 → bit $|\rho\rangle = |n_0 n_1 \dots n_{N_\rho}\rangle$

* bosons: compact notation by 'fermionization' of occupation numbers

e.g. $|2\ 0\ 1\ 3\rangle$

$|1\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 1\rangle$

- create ordered array with all valid configurations
  - not all integers code a valid state due to symmetries
  - simple conserved quantities

S_z, N ⇒ matching words selected; efficiently done recursively.

Basic algorithmic task for Hilbert space:
given an element of the space, find its index

e.g. 3 fermions in four orbitals ← complete later

orbitals	$i=3$	$i=2$	$i=1$	$i=0$	index μ	$m_{Hi} + n_{low}$
n_i:	0	1	1	1	0	0 + 0
	1	0	1	1	1	1 + 0
	1	1	0	1	2	2 + 0
	1	1	1	0	3	2 + 1

} have choice how to distribute the offset!

efficient look-up by bisection:

Hi: word	value	n_{Hi}
0 0	—	0
0 1	0	1
1 0	1	1
1 1	2	2

Low: word	value	n_{Low}
0 0	—	0
0 1	0	1
1 0	1	1
1 1	0	2

⇒ Complete look-up while storing only $2 \cdot 2^{N_{orb}/2}$ entries in memory.

⇒ relies on conserved quantities, here: N = particle number

Alternatives:
- full look-up by hash-table
  ⇒ search $O(1)$
- binary search in ordered table of all entries ⇒ ~~find~~

Hamiltonian:

- once calculated, can be stored for moderate system sizes.
  → sparse-matrix formats
- to evaluate matrix-elements, consider action of terms in Hamiltonian on basis-states.

e.g. NN hopping $H = -t \sum_{\langle i,j \rangle} (c_i^\dagger c_j + \text{h.c.})$

$$|\chi\rangle = H|\varphi\rangle = H \sum_\mu \varphi_\mu |\mu\rangle = \sum_\mu \varphi_\mu H|\mu\rangle$$

includes a term for acting on $\mu = 1$

$$c_2^\dagger c_1 \; |\mu = '1011'\rangle = |1101\rangle$$

- need to search index ν of $|1101\rangle \rightarrow \nu = 2$ (lookup)
- add matrix-element * initial amplitude to target-vector $|\chi\rangle$

Stopped here.

here: $\chi_2 \mathrel{+}= \varphi_1 \cdot (-t)$.

- loop over all terms in Hamiltonian
  and over all { initial basis states.
  alternatively: final basis states!

⇒ easily parallelizable!