ビット表現

- スピン状態 ($\uparrow$ / $\downarrow$) や量子ビット状態 ($|0\rangle$ / $|1\rangle$) を表現するには二進数を考えるのが便利
  - スピン数 (量子ビット数): N
  - 状態数: 2^N
  - 整数を二進数表現したときの下から i 番目 ($i = 0, 1, \ldots, N-1$) の数字 (0 / 1) を i 番目のスピン状態 ($\uparrow$ / $\downarrow$) あるいは i 番目の量子ビット状態 ($|0\rangle$ / $|1\rangle$) に対応させる
- $N = 3$ の例
  - $0 \to 000$: $\uparrow\uparrow\uparrow$ $|000\rangle$
  - $1 \to 001$: $\uparrow\uparrow\downarrow$ $|001\rangle$
  - $2 \to 010$: $\uparrow\downarrow\uparrow$ $|010\rangle$
  - $3 \to 011$: $\uparrow\downarrow\downarrow$ $|011\rangle$
  - $4 \to 100$: $\downarrow\uparrow\uparrow$ $|100\rangle$
  - $5 \to 101$: $\downarrow\uparrow\downarrow$ $|101\rangle$
  - $6 \to 110$: $\downarrow\downarrow\uparrow$ $|110\rangle$
  - $7 \to 111$: $\downarrow\downarrow\downarrow$ $|111\rangle$

ビット表現

- 特定のビットの取り出し
  - シフト演算 (>>) と AND 演算 (&) を利用　例)
    3 番目 ($i = 3$) のビット (0 / 1) を取り出す: `(s>>3)&1`
    3 番目 ($i = 3$) のスピン状態 ($\sigma_3 = \pm 1$) を取り出す: `1-2*((s>>3)&1)`
    $\sigma_i \sigma_j$ の計算: `(1-2*((s>>i)&1)) * (1-2*((s>>j)&1))`
  - i は 0 から数えることに注意 ($i = 0, 1, \ldots, N-1$)
  - ビット AND (&) と論理 AND (&&) との違いに注意
- 特定のビットのフリップ (反転)
  - シフト演算 (<<) と XOR(排他的論理和) 演算 (^) を利用　例)
    3 番目 ($i = 3$) のビットを反転: `s^(1<<3)`
- N ビット全てが 1 の状態を作る
  - `(1<<N)-1`
- N 重の for ループを書く代わりに、状態を 1 つの N ビットの整数 ($s = 0, \cdots, 2^N - 1$) で表し、ひとつのループに

疎行列

- 疎行列: 非零の要素の数が非常に少ない行列
  - 全ての要素 (N^2) を保存しておくのはメモリの無駄
  - 行列ベクトル積の計算量 (通常 $O(N^2)$) も削減の余地あり

- 疎行列の格納
  - 三重対角行列 (一次元ラプラシアンなど): 例) `tridiagonal.c`
    対角成分＋副対角成分を 3 本のベクトルに保存しておけばよい
    対称 (エルミート) 行列の場合は副対角はどちらか 1 本だけでよい
    LAPACK の三重対角行列用のソルバー (`DSTEV`, `DGTTRF` など) にはこの形式の行列を渡す
  - 一般の疎行列: 例) `sparse.c`
    各行で非零の要素の場所 (列) とその値をベクトルに保存する
    CRS (Compressed Row Storage) 形式とも呼ばれる
  - matfree 形式: 例) `matfree.c`
    要素は保存せず、その場で非零の場所と要素を計算する
    FTCS 法を行列形式を使わずに素直に実装するのと同じ
    横磁場イジング模型のハミルトニアンの掛け算などでも使える