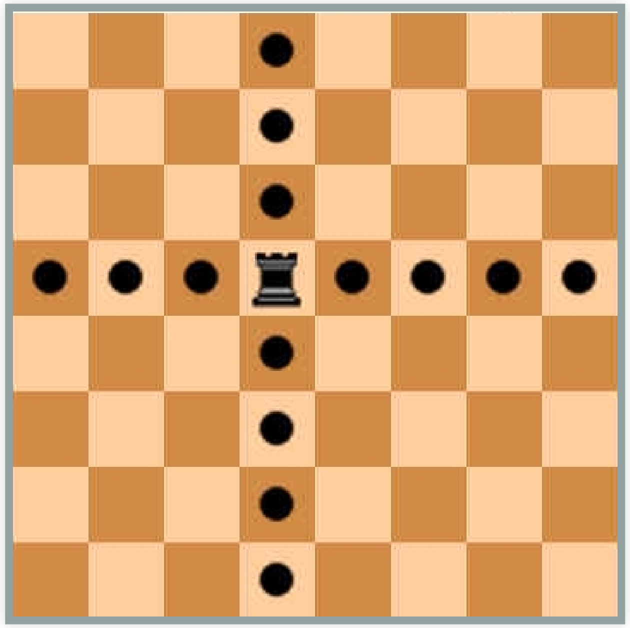
NUMERICAL OPTIMISATION

THE MOST GENERAL PROBLEM

- We have a finite number of variables: collectively denoted as X
- We have a function $F(X)$
- Minimum value of $F(X)$
  for X satisfying some constraints
- i.e. X is restricted to some domain. E.g.
  - $X = (x, y, z)$ and $x = y$
  - positive-definite thermal displacement tensor

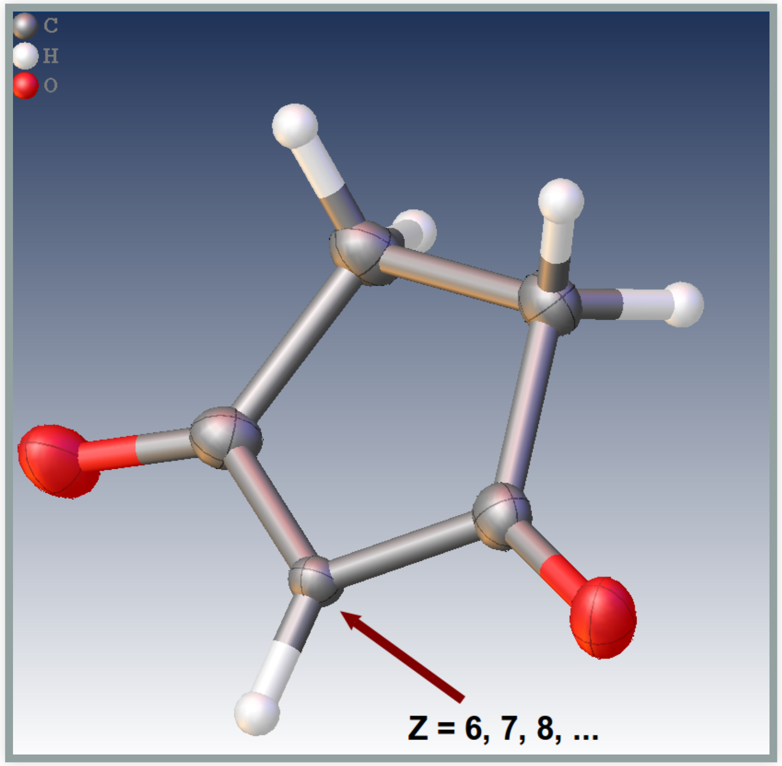
CONTINUOUS VS DISCRETE

- Variables continuously change? Or only by steps?
- Discrete optimisation: all variables are discrete



MIXED OPTIMISATION

- Some variables are continuous, some are discrete
- Very relevant to crystallography: element assignment



HOW TO DEAL WITH DISCRETE VARIABLES?

- Discrete optimisation:
  - Combinatorial methods: from mere enumeration to AI
  - Transform into continuous:
    - example: continuously deform X-ray form factors, C to N to O, with non-integer Z!

$$\sum_i a_i(Z) \exp\left[b_i(Z) \left(\frac{\sin\theta}{\lambda} \right)^2 \right]$$

- I won't talk about this further!

ITERATIVE ALGORITHMS

1. Starting variable X_0
2. Evaluate $F(X_0)$ and perhaps some of its derivatives
3. Guess a small move $X_0 \rightarrow X_1 = X_0 + \delta$
4. Resume at step 1 with that new value unless
    - δ is small
    - 1st order derivatives of F are small enough or $F(X_1) \approx F(X_0)$

ALGORITHMS USED IN CRYSTALLOGRAPHY

- no derivative: evolutionary algorithm and simulated annealing
  - ab-initio solution in powder crystallography (FOX)
- 1st order derivatives (exact) and 2nd order (approximated)
  - quasi-Newton methods:
    - n^2 of them! but many are negligeable (sparse)
      derivative wrt to x_A and x_B very small if A and B are not close
  - Algorithms
    - Phenix: LBFGS (`scitbx.lbfgs`)
    - REFMAC: handmade
    - Small molecule: special approximations permitted by least-squares
      - full matrix
      - CGLS

DATA FITTING (MAXIMUM LIKELIHOOD)

- Mostly interested in particular class of problem: model vs data
- Crystallographic refinement: archetype
  - data: we "measure" Bragg intensities $\Longrightarrow$ HKL file
    - list of reflections: h, k, l, I_o, σ_I
  - model:
    - some parameters X (atom positions, thermal displacements, etc)
    - for each hkl: predict intensity $I_c(X)$
  - search best parameters: use probabilities

PREDICTED INTENSITIES

- Complex structure factors: for each atom

$$F_c = f\left(\frac{\sin\theta}{\lambda}\right) e^{ihr} e^{-hUh^T}$$

- Small molecule: richer!
  - non-periodic crystals: $h, k, l, m, n, \cdots$
  - charge density: fit of anisotropic form factors
  - magnetic structures

MAXIMUM LIKELIHOOD

- model parameters X and measured intensities I_o: random variable
  - a priori probability $p(X)$
  - prob of I_o knowing X: likelihood function $L(X, I_o)$
- Maximise prob of X knowing I_o = $p(X)L(X, I_o)$:
  minimise $-\log$ of that $= -\log p(X) - \log L(X, I_o)$

A PRIORI PROBABILITIES FOR PARAMETERS

- e.g. bond distance $A - B$ shall be $d \pm \sigma$: Gaussian model
$$p \propto e^{-\left(\frac{AB-d}{\sigma}\right)^2}$$
- same for dihedral angles or torsion angles
- Log-likelihood
$$= \frac{1}{\sigma^2}\left(\sqrt{(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2} - d\right)^2$$
  - restraint (soft)

LEAST-SQUARES: A SPECIAL CASE

- Measured intensity: Gaussian distribution about computed one

  $$F(X) = -\log L = \left(\frac{I_o - I_c(X)}{\sigma_I} \right)^2$$

  Numerator $I_o - I_c$ called "residual"
- Intensities for different hkl are independent
  - multiply likelihood, or
  - sum log-likelihood (over all hkl)
- Good assumption for small molecules, not so for proteins
- Inverse of matrix of 2nd order derivatives gives variances and correlations of parameters

LEAST-SQUARES: A SPECIAL CASE

- Newton algorithm with approximate 2nd order derivatives

$$\frac{\partial F_c}{\partial x_i \partial x_j} \approx \frac{\partial F_c}{\partial x_i} \frac{\partial F_c}{\partial x_j}$$

  - Full matrix refinement (Gauss-Newton)
  - very good approximation close to minimum
  - nomenclature:
    - design matrix: row i, col $j \rightarrow \frac{\partial F_c}{\partial x_j}$ for i-th reflection
    - normal matrix: row i, col $j \rightarrow$ right hand side above

LEAST-SQUARES SOLVERS

- `scipy.optimize`
- CCTBX: dedicated module `scitbx.lstbx`
  - unknown scale:

    $$F(X) = \sum_{hkl} \left(\frac{I_o - K I_c(X)}{\sigma_I} \right)^2$$

  - Minimise K, then for other parameters; repeat