An Interior Point Approach to Electronic Structure Computation

Scott A. Miller

INRIA Rhône-Alpes

Séminaire du Calcul Scientifique du CERMICS

5 June 2002

Outline

1. Electronic structure computation
   [Cancès, Defranceschi, Le Bris and Maday (2002)]
   - approximations
   - linear scaling methods
   - convex program for density matrix
2. Path-following interior point method [Boyd and Vandenberghe (2001)]
   - formulation
   - number of iterations
   - cost of each iteration
3. Automatic differentiation [Griewank (1989), Vavasis (1999)]
4. Automatic CG preconditioning [Morales and Nocedal (2000,2002)]
5. Summary

Quantum Ground State

- Molecular dynamics described by the *Schrödinger equation*:

$$i\frac{\partial}{\partial t}\Psi(t) = H\Psi(t)$$

where

  - $\Psi(t) \in \mathcal{H}$ is a wavefunction in a Hilbert space $\mathcal{H}$, with $\|\Psi(t)\| = 1$ for all t
  - H is the Hamiltonian of the system, a linear self-adjoint operator over $\mathcal{H}$

- Objective: find the *ground state*

$$E_0 := \inf\left\{\langle\psi, H\psi\rangle \;\middle|\; \psi \in \mathcal{H},\ \|\psi\| = 1\right\}$$

- Ignoring spin, $\mathcal{H} \subset L_2(\mathbb{R}^{3(M+N)}, \mathbb{C})$ — huge dimension!

Born-Oppenheimer Approximation

One wavefunction for the nuclei and one for the electrons:

$$E_0 \approx \inf \Big\{ \langle \psi, H\psi \rangle \;\Big|\; \psi = \psi_n \psi_e, \\ \psi_n \in \mathcal{H}_n,\ \|\psi_n\| = 1,\ \psi_e \in \mathcal{H}_e,\ \|\psi_e\| = 1 \Big\}$$

After some simplifications, the problem becomes

$$\inf \Big\{ W(\bar{x}) \;\Big|\; \bar{x} \in \mathbb{R}^{3M} \Big\}$$

where

$$W(\bar{x}) = U(\bar{x}) + \sum_{1 \le i < j \le M} \frac{z_i z_j}{|\bar{x}_i - \bar{x}_j|}$$

$$\boxed{U(\bar{x}) = \inf \Big\{ \langle \psi_e, H_e^{\bar{x}} \psi_e \rangle \;\Big|\; \psi_e \in \mathcal{H}_e,\ \|\psi_e\| = 1 \Big\}}$$

Hartree-Fock Approximation

- Electron Hilbert space is $\mathcal{H}_e = \bigwedge_{i=1}^{N} H^1(\mathbb{R}^3)$, where $H^1(\mathbb{R}^3) \subset L_2(\mathbb{R}^3, \mathbb{R})$. This is too big to discretize directly, combinatorial in N.

- *Hartree-Fock*: approximate $\psi_e \in \mathcal{H}_e$ with

$$\psi_e = \frac{1}{\sqrt{N!}} \det[\phi_i(x_j)]_{i,j} \quad \text{(Slater determinant)}$$

$$\phi_i \in H^1(\mathbb{R}^3) \text{ for } i = 1, \ldots, N, \text{ and } \langle \phi_i, \phi_j \rangle = \delta_{ij}$$

- Positions of N electrons described by N decoupled *molecular orbitals* ϕ_i, so

$$|\psi_e(x_1, \ldots, x_N)|^2 \approx |\phi_1(x_1)|^2 + \cdots + |\phi_N(x_N)|^2$$

Kohn-Sham Approximation

- Density functional:

$$\rho(x) := N \int_{\mathbb{R}^{3(N-1)}} |\psi_e(x, x_2, \ldots, x_N)|^2 \, dx_2 \cdots dx_N \in H^1(\mathbb{R}^3)$$

- Approximate H_e with Hamiltonian corresponding to N noninteracting electrons (Coulomb force suppressed).
- Electronic structure problem $\inf_{\psi_e} \langle \psi_e, H_e \psi_e \rangle$ can be recast in terms of ρ for noninteracting system.
- *Kohn-Sham*: assume there is a minimizing ρ that is the density of a Slater determinant, so

$$\rho(x) = \sum_{i=1}^{N} |\phi_i(x)|^2$$

Formal Equivalence

The two approximations are formally the same:

$$\inf \left\{ \langle \psi_e, H_e \psi_e \rangle \;\middle|\; \psi_e \in \mathcal{H}_e,\ \|\psi_e\| = 1 \right\}$$

$$\approx \inf \left\{ \langle \Phi, F_\Phi \Phi \rangle \;\middle|\; \Phi = [\phi_1, \ldots, \phi_N],\ \phi_i \in H^1(\mathbb{R}^3),\ \langle \Phi, \Phi \rangle = I_N \right\}$$

though F_Φ is different between HF and KS.

Discretization

- Discretize $H^1(\mathbb{R}^3)$: choose a basis $\{\chi_1, \ldots, \chi_{N_b}\}$ for $H^1(\mathbb{R}^3)$ and define $C \in \mathbb{R}^{N_b \times N}$ and $S, F \in \mathcal{S}^{N_b}$ according to

$$\begin{aligned} \phi_j &= \sum_{i=1}^{N_b} C_{ij} \chi_i \quad \text{(coefficient matrix)} \\ S_{ij} &= \langle \chi_i, \chi_j \rangle \qquad \text{(overlap matrix)} \\ F_{ij} &= \langle \chi_i, F_\Phi \chi_j \rangle. \end{aligned}$$

- Note that F depends on C (actually on the *density matrix* $D = CC^T$).

- The discretized electronic structure problem is to find C that solves

$$\min \left\{ C^T F_D C \;\middle|\; C^T S C = I_N,\ D = CC^T \right\} \qquad \text{(MIN-C)}$$

Nonlinear Eigenvalue Problem

- Optimality conditions for (MIN-C) can be written as

$$
\begin{aligned}
F_D C &= SCE \\
C^T SC &= I_N \\
CC^T &= D
\end{aligned}
$$

  where we can assume $E = \mathrm{diag}(\epsilon_1, \ldots, \epsilon_N)$ because of invariance of (MIN-C) to orthogonal change of basis.

- First two equations are a generalized eigenvalue problem.
  Optimal eigenvalues are N lowest eigenvalues of F_D (*aufbau* principle).

- Third equation expresses *Self Consistent Field* (SCF) constraint.

Density Matrix Formulation

- Equivalently (MIN-C) can be written as

$$\min \left\{ \operatorname{Tr} F_D D \;\middle|\; DSD = D,\ \operatorname{Tr} SD = N \right\} \qquad \text{(MIN-D)}$$

- For a fixed $F_D = F$, (MIN-D) can be relaxed to a *convex program*:

$$\boxed{\min \left\{ \operatorname{Tr} FD \;\middle|\; DSD \preceq D,\ \operatorname{Tr} SD \le N \right\}} \qquad \text{(CONV)}$$

  where for $A, B \in \mathcal{S}^n$, $A \succeq B$ means $A - B$ is positive semidefinite.

- Relaxed Constrained Algorithm [Cancès and Le Bris (2000)] uses solutions of (CONV) to find self-consistent solution to (MIN-D) via steepest descent.

Argument for Convex Relaxation

Suppose for simplicity $S = I$. Fix $F_D = F$.

1. Constraints of (MIN-D)
$$D^2 = D, \quad \operatorname{Tr} D = N$$
mean D is a rank-N orthogonal projector QQ^T, and minimizing $\operatorname{Tr} FD = \operatorname{Tr} Q^T FQ$ chooses Q to be a basis for invariant subspace corresponding to N smallest eigenvalues of F.

2. Let $D = V \operatorname{diag}(\delta_1, \dots, \delta_{N_b}) V^T$ be an eigendecomposition of D. Constraints of (CONV) mean that $\delta_i \in [0, 1]$ for all i and at most N eigenvalues are positive.

3. Transform (CONV) by diagonalizing F. We get the same form of problem, but with $F = \operatorname{diag}(f_1, \dots, f_{N_b})$. Assume $f_1 \le f_2 \le \dots \le f_{N_b}$.

4. Transformed (CONV) is equivalent to

$$\min \Big\{ \sum_{i=1}^{N_b} f_i \delta_i \;\Big|\; 0 \le \delta_i \le 1 \text{ for } i = 1, \ldots, N_b,$$

$$\delta_i > 0 \text{ for } i \in \mathcal{I} \subset \{1, \ldots, N_b\},\ |\mathcal{I}| \le N \Big\}$$

Assuming $f_i < 0$ for $i \le N$, the solution is $\delta_i = 1$ for $i \le N$ and $\delta_i = 0$ for $i > N$.

5. Transform back to (CONV) to see that the solution is the same as for (MIN-D).

Linear Scaling Methods

- *Locality* in quantum mechanics: distant electrons do not affect each other much.
- Locality $\Rightarrow$ sparsity of F, S, C and D with the proper choice of basis. Decay properties lead to banded structure after truncation.
- *Linear scaling* methods take advantage of sparsity to solve (MIN-C) or (MIN-D) in $O(N)$ time. This enables solution of large molecular systems. We assume $N_b = mN$ for a fixed m.
- Standard eigenvalue codes for computing *aufbau* solutions to (MIN-C) exhibit $O(N^3)$ scaling; cannot impose sparsity constraints on C or D.

Density Matrix Minimization

- Suppose for simplicity $S = I$, and fix $F_D = F$. DMM replaces the constrained problem
$$\min \left\{ \operatorname{Tr} FD \;\middle|\; D^2 = D,\, \operatorname{Tr} D = N \right\} \qquad \text{(MIN-D')}$$
with the unconstrained problem
$$\min_D \; p(D) := \operatorname{Tr}(F - \mu I)(3D^2 - 2D^3)$$
where μ is the *chemical potential* or *Fermi level* (Lagrange multiplier for $\operatorname{Tr} D = N$ constraint).
- Can show that local minimum of p is unique and solves (MIN-D').
- Solve with conjugate gradient (CG) method; exact linesearch possible because of cubic form; each iteration $O(N)$.

Drawbacks of DMM

- Global minimum of p is $-\infty$. Need good starting point to find desired local minimum.
- CG convergence rate depends on condition number of $\nabla^2 p(D)$. This can be estimated as
$$\kappa = \frac{\epsilon_{N_b} - \epsilon_1}{\epsilon_{N+1} - \epsilon_N}$$
so for systems with small HOMO-LUMO gap (e.g. metals) convergence can be very slow. [Goedecker (1999)]
- Need to identify the chemical potential μ in an outer loop.

Convex Program for Density Minimization

- Let $\mathcal{D} : \mathbb{R}^n \to \mathcal{S}^{N_b}$ be the linear function whose range describes the sparsity pattern of the density matrix D. There are $n = O(N)$ parameters.
- Define constraint functions

$$\begin{aligned} g_1(x) &:= \mathcal{D}(x) - \mathcal{D}(x) S \mathcal{D}(x) \\ g_2(x) &:= N - \operatorname{Tr} S \mathcal{D}(x) \\ g(x) &:= \Big(g_1(x),\ g_2(x) \Big) \end{aligned}$$

  and convex cone $\mathcal{K} = \mathcal{S}_+^{N_b} \times \mathbb{R}_+$.
- Then (CONV) becomes

$$\min \left\{ \operatorname{Tr} F \mathcal{D}(x) \;\middle|\; g(x) \in \mathcal{K} \right\}$$

  We solve this constrained convex program using an *interior-point method*.

Logarithmic Barrier

$$\min \left\{ f(x) \;\middle|\; g(x) \geq 0 \right\} \qquad (*)$$

- Using the *indicator function*

$$\mathcal{I}_g(x) := \begin{cases} 0 & \text{if } g(x) \geq 0 \\ +\infty & \text{otherwise} \end{cases}$$

  we can solve $(*)$ exactly by solving $\min_x f(x) + \mathcal{I}_g(x)$.

- Approximating $\mathcal{I}_g$ by a *logarithmic barrier*

$$\varphi(x) = \begin{cases} -\log g(x) & \text{if } g(x) > 0 \\ +\infty & \text{otherwise} \end{cases}$$

  and solving $\min_x f(x) + \mu\,\varphi(x)$ for some $\mu > 0$, we obtain an approximate solution to $(*)$.

Central Path

- Under mild assumptions

$$x(\mu) := \operatorname*{argmin}_x f(x) + \mu\, \varphi(x)$$

  is well-defined for all $\mu > 0$. This is the *central path*.

- $x(\mu) \to x^*$ as $\mu \to 0$, and x^* solves $(*)$.

- $f(x(\mu)) - f(x^*) \le \theta\mu$ for a known constant θ (the *degree* of the barrier function, e.g. n for barrier for $\mathcal{S}^n_+$)

Path-Following Method

- Given x such that $g(x) > 0$, $\mu_0 > 0$, $\sigma \in (0,1)$, $\varepsilon > 0$:

  Set $\mu \leftarrow \mu_0$.

  repeat

  Compute $x(\mu) := \operatorname{argmin}_x f(x) + \mu\,\varphi(g(x))$.

  Set $x \leftarrow x(\mu)$.

  If $\theta\mu > \varepsilon$, set $\mu \leftarrow \sigma\mu$.

  until $\theta\mu \leq \varepsilon$

- Exits within $\dfrac{\log(\varepsilon/(\theta\mu_0))}{\log\sigma}$ steps with an ε-approximate solution x.
- Total execution time depends on time for centering step.

Self-Concordance

- $\mathbf{f} : \mathbb{R} \to \mathbb{R}$ is *self-concordant* if

$$|\mathbf{f}'''(x)| \leq 2\mathbf{f}''(x)^{3/2}$$

- $\mathbf{f} : V \to \mathbb{R}$ is self-concordant if $\alpha \mapsto \mathbf{f}(x + \alpha v)$ is self-concordant $\forall\, x, v \in V$.

- If $\mathbf{f}$ is self-concordant, can be minimized with Newton's method in $\left(\frac{\mathbf{f}(x_0) - \mathbf{f}_{\min}}{\eta} + c \right)$ steps, for η <u>independent of $\mathbf{f}$</u> and $c \approx 5$.

- Logarithmic barriers are self-concordant.

Total Cost of Path-Following Method

- Let $\mathbf{f}(x) := f(x) + \sigma\mu\,\varphi(g(x))$ from path-following method. Then

$$\mathbf{f}(x(\mu)) - \mathbf{f}(x(\sigma\mu)) \le \theta(\sigma^{-1} - 1 + \log\sigma)$$

and the total number of Newton steps is bounded by

$$\left(\frac{\log(\varepsilon/(\theta\mu_0))}{\log\sigma}\right)\left(\frac{\theta(\sigma^{-1} - 1 + \log\sigma)}{\eta} + c\right).$$

- Choosing $\sigma^{-1} = 1 + 1/\sqrt{\theta}$ and $\mu_0 = \varepsilon_0/\theta$ results in $O\big(\sqrt{\theta}\log(\varepsilon_0/\varepsilon)\big)$ steps to compute ε-optimal solution from ε_0-optimal solution.

Cost of Path-Following for Density Minimization

To solve (CONV) we take

$$\varphi(x) = \begin{cases} -\log\det g_1(x) - \log g_2(x) & \text{if } g(x) \in \operatorname{int}\mathcal{K} \\ +\infty & \text{otherwise} \end{cases}$$

This φ can be shown to be self-concordant with degree $\theta = 1 + N_b$, and so is $\operatorname{Tr} F\mathcal{D}(\cdot) + \mu\varphi(\cdot)$, hence

$$\boxed{\text{total \# Newton iterations is } O\big(\sqrt{N}\big)}$$

But what is the cost of a Newton iteration?

Newton Iteration

- Newton direction d obtained by solving $[\nabla^2 \mathbf{f}(x)]d = -\nabla \mathbf{f}(x)$.

$$\mathbf{f}(x) := \operatorname{Tr} FD - \mu \log\det(D - DSD) - \mu \log(N - \operatorname{Tr} SD)$$

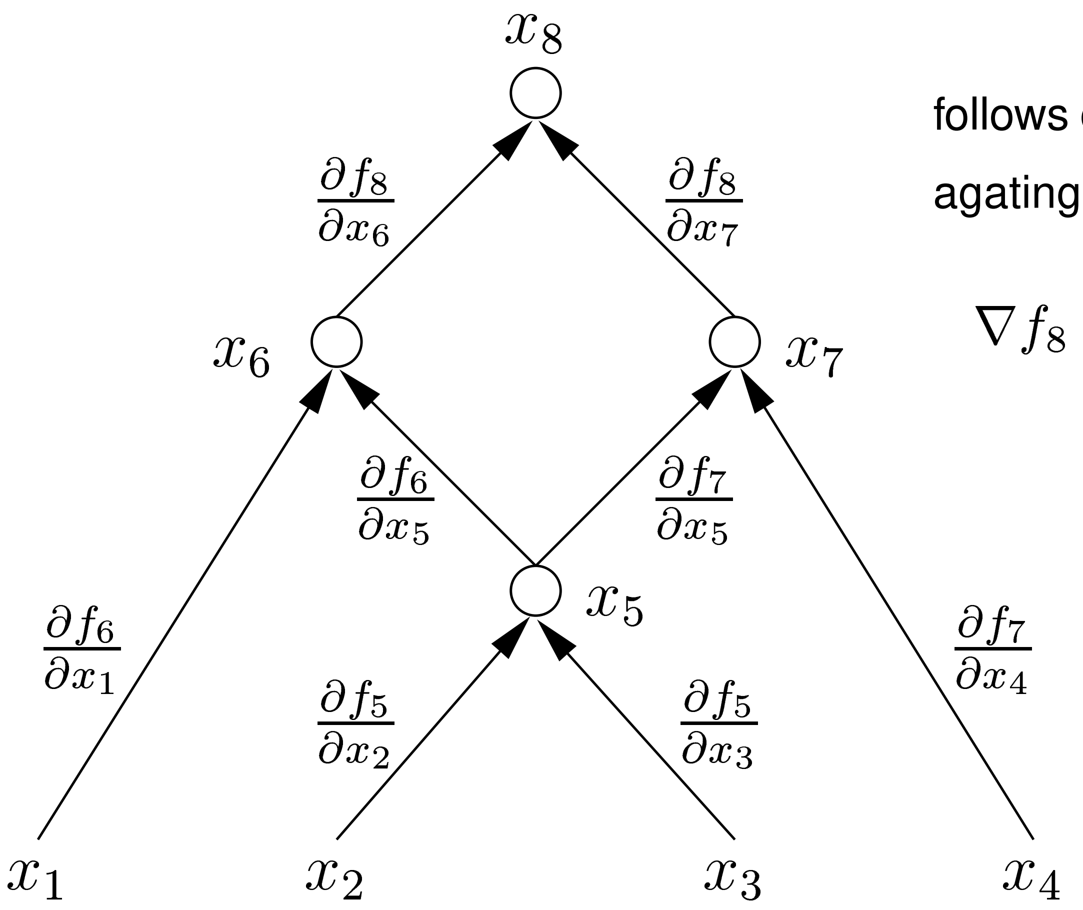
$$\nabla \mathbf{f}(x) = \mathcal{D}^* \left(F - \mu M^{-1} + \mu M^{-1} SD + \mu DSM^{-1} + \frac{\mu}{N - \operatorname{Tr} SD} S \right)$$

$$M := D - DSD, \qquad D := \mathcal{D}(x)$$

$$\nabla^2 \mathbf{f}(x) = ???$$

- Automatic differentiation to the rescue:
  - $\nabla \mathbf{f}(x)$ computable in $O(T_c)$, where T_c is time to compute Cholesky decomposition $M = R^T R$. Expect $T_c = O(N)$.
  - $v \mapsto [\nabla^2 \mathbf{f}(x)]v$ computable in same order of time.
- Apply conjugate gradient method to solve $[\nabla^2 \mathbf{f}(x)]d = -\nabla \mathbf{f}(x)$.

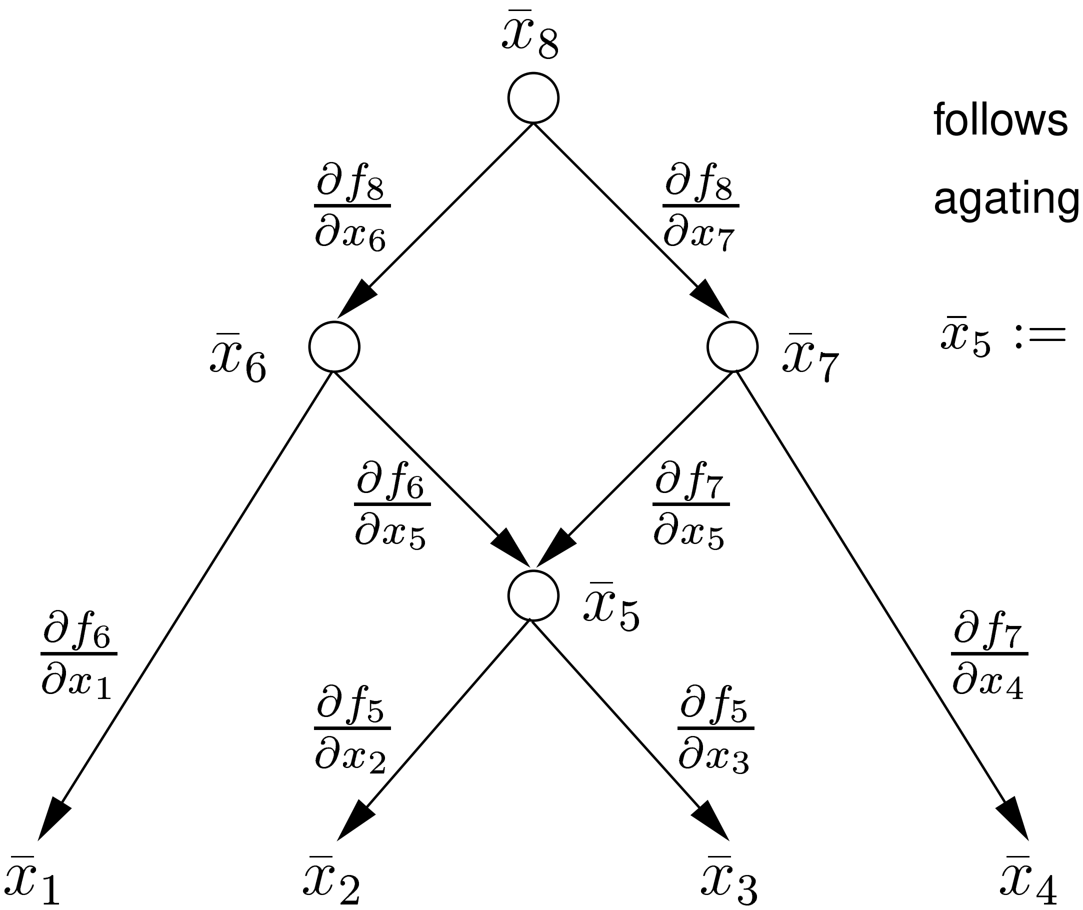
Automatic Differentiation — Forward Accumulation



follows original function graph, propagating *gradients* along each link:

$$\nabla f_8 = \frac{\partial f_8}{\partial x_6}\nabla f_6 + \frac{\partial f_8}{\partial x_7}\nabla f_7$$

Automatic Differentiation — Reverse Accumulation



follows adjoint function graph, propagating *partial derivatives*:

$$\bar{x}_5 := \frac{\partial f_8}{\partial x_5} = \frac{\partial f_6}{\partial x_5}\bar{x}_6 + \frac{\partial f_7}{\partial x_5}\bar{x}_7$$

Cost of Reverse Mode AD

- *Theorem*: $\mathrm{time}(\nabla f(x)) \leq 5\,\mathrm{time}(f(x))$ using reverse mode AD. [Griewank (1989)]
- Drawbacks:
  - Not efficient for vector-valued functions.
  - Need to keep intermediate $\bar{x}_i$'s $\quad \Rightarrow \quad$ storage can be excessive.
- For $f(X) = -\log\det X = -2\sum_{i=1}^{n} \log R_{ii}$ with $X = R^T R$, hand optimization of AD code produces $\mathrm{space}(\nabla f(X)) = O(\mathrm{space}(R))$. [Vavasis (1999)]

Hessian-Vector Products

- Define $g_v(x) := v^T \nabla f(x)$. This is a scalar function which we can compute efficiently using AD.
- Then apply reverse mode AD again to compute $\nabla g_v(x) = [\nabla^2 f(x)]v$ efficiently.
- For $f(X) = -\log\det X$, we get $[\nabla^2 f(x)]v$ in $O(\mathrm{time}(R))$ time and $O(\mathrm{space}(R))$ space.

Conjugate Gradient Convergence

- Worst-case convergence in $O(N)$ iterations, $O(1)$ for well-conditioned problems.
- For condition number $\kappa = \frac{\lambda_{\max}(\nabla^2 \mathbf{f})}{\lambda_{\min}(\nabla^2 \mathbf{f})} \gg 1$, number of iterations $\propto \sqrt{\kappa}$.
- As $\mu \to 0$ and $x(\mu) \to x^*$, Hessian becomes ill-conditioned, so *accurate* solutions to (CONV) may be costly.
- Possible solutions:
  - automatic preconditioning of the CG iteration
  - use of partial primal information as in [Vavasis (2002)] or dual scaling interior point method of [Benson, Ye and Zhang (2000)]

Automatic Preconditioning of CG

- Basic idea [Morales and Nocedal (2000)]:
  - at Newton iteration k, construct $M_k \approx [\nabla^2 \mathbf{f}_k(x_k)]^{-1}$ from CG iterates, using a limited memory BFGS (L-BFGS) technique
  - use M_k as CG preconditioner at Newton iteration $k+1$
- Enhancement [Morales and Nocedal (2002)]:
  - alternate some number of iterations of Hessian-free Newton (using preconditioned CG) and L-BFGS, using each to enhance the other
  - L-BFGS is cheaper than HFN but does not converge as well when Hessian is ill-conditioned

Summary of Program Structure

density minimization via RCA

path-following IP method

Newton method for centering

AD to compute $\nabla \mathbf{f}(x)$

CG to compute Newton direction

AD to compute $v \mapsto [\nabla^2 \mathbf{f}(x)]v$

L-BFGS to compute next preconditioner

Advantages of IP Method

- Global convergence.
- Quick computation of approximate solutions — this may be all that RCA requires most of the time.
- When high accuracy required, CG can be automatically preconditioned, taking advantage of *sequence* of slowly changing Newton subproblems.
- Small HOMO-LUMO gap should pose no particular problems, except convergence to extremely accurate (and therefore ill-conditioned) solution can be slow.
- No need to separately identify chemical potential μ.

Total Cost of IP Method

- Assume $O(N)$ to compute $\det(D - DSD)$
  $\Rightarrow \; O(N)$ for each CG iteration.

- Then total cost is $O(N n_{\mathrm{N}} n_{\mathrm{CG}})$, where

$$n_{\mathrm{N}} = \begin{cases} O(\sqrt{N}) & \text{worst case} \\ O(1) & \text{in practice} \end{cases}$$

$$n_{\mathsf{CG}} = \begin{cases} O(N) & \text{worst case} \\ O(1) & \text{if low accuracy or good preconditioner} \end{cases}$$

- So $O(N)$ method for (CONV) is possible.

References

- S. J. Benson and Y. Ye and X. Zhang. *Solving large-scale sparse semidefinite programs for combinatorial optimization.* SIAM J. Opt., 10(2):443–461, 2000.
- S. Boyd and L. Vandenberghe. *Convex Optimization.* Manuscript, December 2001.
- E. Cancès, M. Defranceschi, C. Le Bris and Y. Maday. *Computational Quantum Chemistry: A Primer.* Preprint submitted to Elsevier Science, April 2002.
- S. Goedecker. *Linear scaling electronic structure methods.* Rev. Modern Physics, 71(4):1085–1123, 1999.
- A. Griewank. On automatic differentiation. Pages 83–108 in *Mathematical Programming: Recent Developments and Applications*, M. Iri and K. Tanabe, eds. Kluwer, 1989.
- J. L. Morales and J. Nocedal. *Automatic preconditioning by limited memory quasi-Newton updating.* SIAM J. Optim., 10(4):1079–1096, 2000.
- J. L. Morales and J. Nocedal. *Enriched methods for large-scale unconstrained optimization.* Comp. Opt. Appl., 21(2):143–154, 2002.
- S. A. Vavasis. *A note on efficient computation of the gradient in semidefinite programming.* Manuscript, 24 Sept. 1999.
- S. A. Vavasis. *Sparse SDP via automatic differentiation and chordal graphs.* Presented at SIAM Conf. on Opt., May 2002.