.

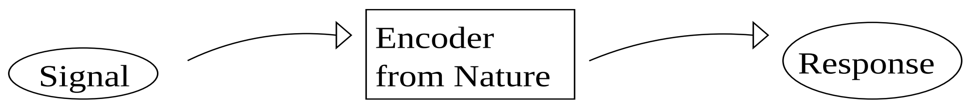
Large Scale Optimization Techniques for Alex's Neural Coding and Decoding Model

Outline

- Mathematical Model: Neural Coding and Decoding
- Optimization Problem
- The Basics: Unconstrained Optimization
    - Line Search Techniques
        - Steepest Descent
        - Newton Method
        - Newton Conjugate Gradient
- Constrained Optimization
    - Projected Gradient
    - Augmented Lagrangian
- Numerical Results
- Future Goals

The Problem

How does neural ensemble activity represent information about sensory stimuli? What was the environmental stimulus that produced a given neural sequence?

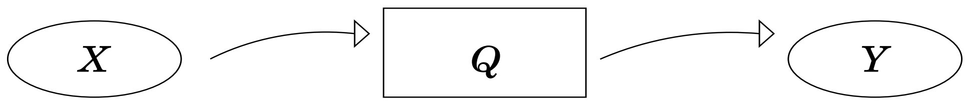


Model Assumptions

- *Typical sequences* in the stimulus and response are known
- The joint probability relating the stimulus and response is known

Information Theoretic Quantities

An **quantizer** or encoder, Q, relates the environmental stimulus, X to the neural response Y through a process called *quantization*. In general, Q is a stochastic map, so that $\sum_y Q(y \mid x) = 1$ for each x.



The **Reproduction** space Y is a quantization of X. This can be repeated: Let Y_N be a reproduction of Y. So there is a quantizer

$$q(y_N \mid y) : Y \to Y_N$$

Mutual Information is a measure of the dependence between two random variables. For X and Y_N

$$I(X, Y_N) = \sum_{x,y,y_N} q(y_N \mid y) p(x,y) \log \left(\frac{\sum_y q(y_N \mid y) p(x,y)}{p(x) \sum_y p(y) q(y_N \mid y)} \right).$$

Conditional Entropy is a measure of the self information of a random variable given another. For Y_N given Y

$$H(Y_N \mid Y) = \sum_{y,y_N} p(y) q(y_N \mid y) \log \left(q(y_N \mid y) \right)$$

The Model for Neural Coding and Decoding

Problem: It would take an inordinate amount of data to determine the coding scheme between X and Y.

Model: Consider the problem of determining the coding scheme between X and Y_N, a quantization of Y, such that: Y_N preserves as much mutual information with X as possible and the entropy of $Y_N|Y$ is maximized.

Justification: *Jayne's maximum entropy principle*, which states that of all the quantizers that satisfy a given set of constraints, choose the one that maximizes the entropy.

Constraints:

- The mutual information $D_{eff} = I(X, Y_N)$ is a measure of how well Y_N represents Y. That is, for a given Y_N, we want a quantizer
$$q(y_N \mid y) : Y \to Y_N$$
that preserves as much mutual information from X as possible.

- q is a quantizer $\Rightarrow$ q is a probability density

Model: We have two maximization problems:

$$\max_{q(y_N|y)} H(Y_N \mid Y) \text{ subject to } D_{eff} \geq I_0 \text{ and } \sum_{y_N} q(y_N \mid y) = 1$$

Reformulated using Lagrange Multipliers:

$$\max_{q(y_N|y)} F(q(y_N \mid y), \beta) \equiv \max_{q(y_N|y)} \left(H(Y_N|Y) + \beta D_{eff}(q(y_N \mid y))\right)$$
$$\text{constrained by } \sum_{y_N} q(y_N \mid y) = 1.$$

The Optimization Problem

We now have two minimization problems:

$$\min_{q(y_N|y)} -H(Y_N \mid Y) \quad \text{constrained by}$$

$$\begin{aligned} & D_{eff} \geq I_0 \\ & \sum_{y_N} q(y_N \mid y) = 1 \quad \forall\ y \in Y \\ & q(y_N \mid y) \geq 0 \quad \forall\ y \in Y \quad \text{and} \quad \forall\ y_N \in Y_N \end{aligned}$$

and

$$\begin{aligned} \min_{q(y_N|y)} -F(q(y_N \mid y), \beta) \;=\; & \min_{q(y_N|y)} \left(-H(Y_N|Y) - \beta D_{eff}(q(y_N \mid y))\right) \\ & \text{constrained by} \\ & \sum_{y_N} q(y_N \mid y) = 1 \quad \forall\ y \in Y \\ & q(y_N \mid y) \geq 0 \quad \forall\ y \in Y \quad \text{and} \end{aligned}$$

We will restrict our attention to $\mathcal{F}(q) \equiv -F(q(y_N \mid y)|\beta)$

Optimization Overview

What? Compute $q^* = \arg\min \mathcal{F}(q)$ subject to the constraints.

Why? To quantize Y into an optimal Y_N.

$$
\begin{array}{lllllll}
q^* = \{ & q(y_{N_1}|y_1) & q(y_{N_1}|y_2) & q(y_{N_1}|y_3) & \ldots & q(y_{N_1}|y_m) \\
& q(y_{N_2}|y_1) & q(y_{N_2}|y_2) & q(y_{N_2}|y_3) & \ldots & q(y_{N_2}|y_m) \\
& . \\
& . \\
& q(y_{N_N}|y_1) & q(y_{N_N}|y_2) & q(y_{N_N}|y_3) & \ldots & q(y_{N_N}|y_m)\}
\end{array}
$$

where $q(y_{Ni}|y_j)$ is a probability, "close" to either zero or one, which determines whether y_j belongs to the class y_{N_i} in the reproduction space Y_N.

How? Use *Optimization Techniques* to build a sequence $\{q_k\}_{k=1}^{\infty}$ to q^* such that

- $\mathcal{F}$ is decreased: $\mathcal{F}_k \geq \mathcal{F}_{k+1}$ for all k
- *global convergence:* $||\nabla \mathcal{F}_k|| \to 0$ as $k \to \infty$
- the constraints are satisfied.

Line Search Techniques can be used to create such a sequence.

Unconstrained Line Search

Goal: Build a sequence $\{q_k\}_{k=1}^{\infty}$ of approximates to q^* such that $\mathcal{F}_k \geq \mathcal{F}_{k+1}$ for all k and $||\nabla \mathcal{F}_k|| \to 0$ as $k \to \infty$.

Idea: At q_k compute q_{k+1} as follows:

1. Compute a **search direction** p_k at q_k.
2. Compute the **step length**
$$\alpha_k \approx \arg\min_{\alpha>0} \mathcal{F}(q_k + \alpha p_k).$$
3. Define $q_{k+1} = q_k + \alpha_k p_k$.

Computing the Step Length α_k

Given the descent direction p_k what conditions should we put on α_k so that we achieve the above goal?

- **Naive Condition:** $\mathcal{F}(q_k + \alpha_k p_k) < \mathcal{F}(q_k)$.

- **The Wolfe Conditions:**

  (W1) $\mathcal{F}(q_k + \alpha_k p_k) \leq \mathcal{F}(q_k) + c_1 \alpha_k \nabla \mathcal{F}(q_k)^T p_k \quad c_1 \in (0, 1)$

  (W2) $\nabla \mathcal{F}(q_k + \alpha_k p_k)^T p_k \geq c_2 \nabla \mathcal{F}(q_k)^T p_k \quad c_2 \in (c_1, 1)$

Zoutendijk's Theorem assures that if $\nabla \mathcal{F}$ is Lipshitz in a neighborhood containing the level set of q_0, then line searches satisfying the Wolfe Conditions meet our goal

Computing a Search Direction p_k

- p_k needs to be a *descent direction*:

$$p_k^T \nabla \mathcal{F}_k < 0.$$

Descent directions and the Associated Methods:

- The direction of steepest descent: $p_k = -\nabla \mathcal{F}_k$.

  The Steepest Descent Method:

  Convergence is linear.

  Cost is low.

- The Newton direction: $p_k = -H_k^{-1} \nabla \mathcal{F}_k$ when H_k is SPD.

  Newton's Method:

  Convergence is quadratic.

  Cost is high.

- The Quasi-Newton direction: $p_k = -B_k^{-1} \nabla \mathcal{F}_k$ when B_k is SPD.

  Quasi-Newton Method:

  A compromise.

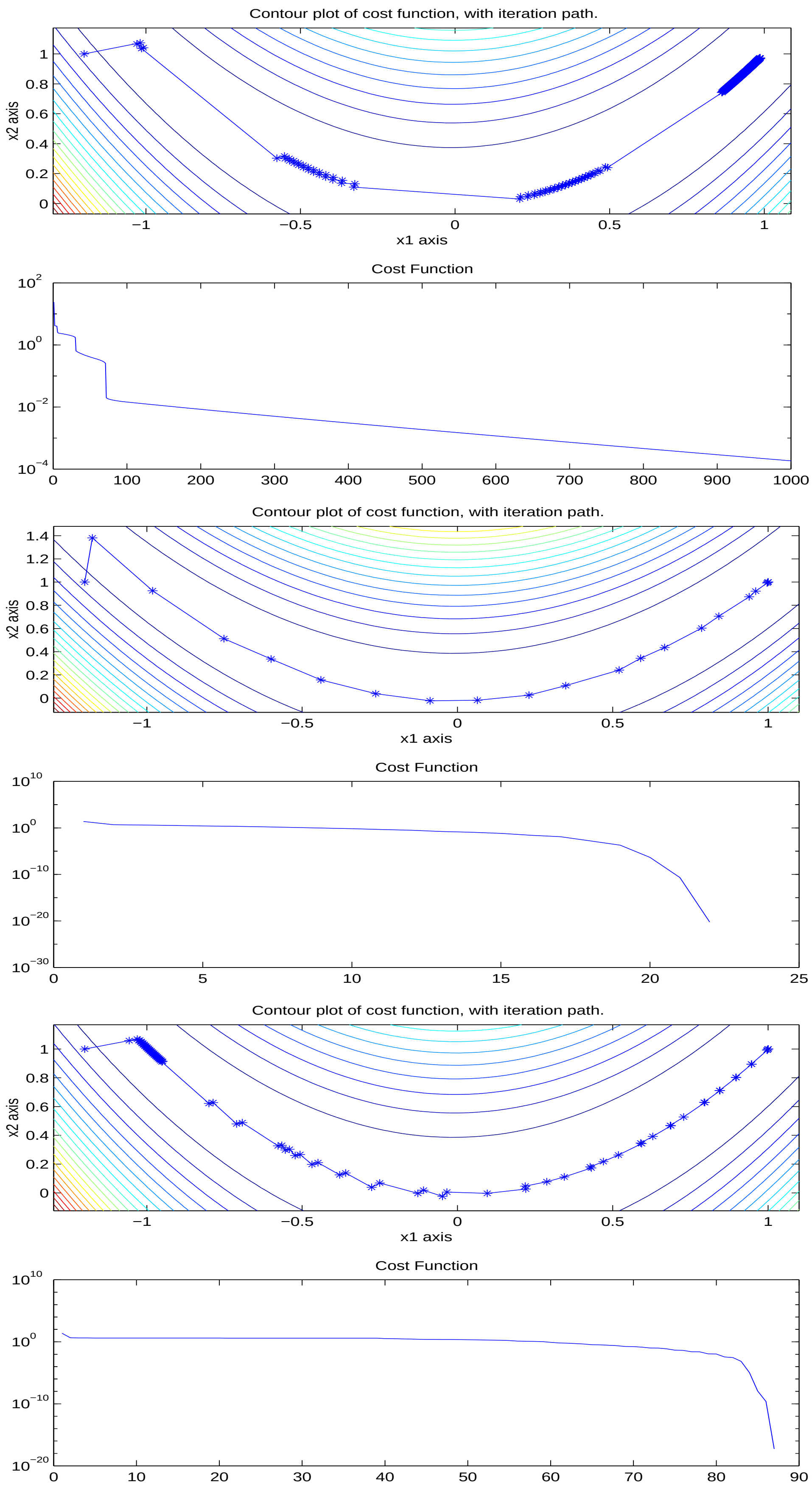


Figure 1: Numerical Performance of (i) Steepest Descent, (ii) Newton's Method (iii) Newton CG applied to the Rosenbrock function for $x_0 = [-1.2, 1]^T$ and $x^* = [1, 1]^T$

Newton Conjugate Gradient

Problem: Solving $H_k p_k = -\nabla \mathcal{F}_k$ can be expensive.

Goal: For H SPD, efficiently solve $Hp = -g$

Idea: Create a sequence $\{p_j\}$ which converges to $p^* = -H^{-1}g$ in finitely many iterations.

- Our goal is equivalent to minimizing $\phi(p) = \frac{1}{2}p^T H p + g^T p$.
- Minimize $\phi(p)$ using a line search:

  Search Direction $d_j = -\nabla \phi_{j-1} + \frac{<\nabla \phi_{j-1}, d_{j-1}>_H}{||d_{j-1}||_H^2} d_{j-1}$

  Step Length $\tau_j = \arg\min_{\tau > 0} \phi(p_j + \tau d_j)$

  So $p_{j+1} = p_j + \tau_j d_j$.

Theorem: For any initial $p_0 \in \Re^n$, $p_j \to p^*$ in at most n steps.

Steihaug's Stopping Criteria: Stop the CG iteration when any of the following occur:

- CG residual $||Hp_j + g|| \leq \epsilon$, where ϵ denotes stopping tolerance.
- Negative curvature detected, i.e., $d_j^T H d_j < 0$ (Newton CG for H not PD).

Preconditioning

Problem: If $\text{cond}(H) \equiv \frac{\lambda_{\max}}{\lambda_{\min}} \gg 1$ or if the eigenvalues of H are not clustered, then it is not economical to use Newton CG to solve $Hp = -g$

Reason: Converegnce of $\{p_j\}$ to p^* is bounded by:

- $||p_j - p^*||_H \leq \left(\frac{\sqrt{\text{cond}(H)}-1}{\sqrt{\text{cond}(H)}+1}\right)^{2j} ||p_0 - p^*||_H$
- $||p_{J+1} - p^*||_H \leq (\lambda_{n-J} - \lambda_1)||p_0 - p^*||_H$
- If eigenvalues occur in r distinct clusters, then Newton CG approximately solves the system in r steps

Goal: Transform $Hp = -g$ to an equivalent system to improve the eigenvalue decomposition of H.

Idea: Set $\hat{p} = Cp$, for nonsingular positive definite C. Then the transformed linear system is

$$C^{-T}HC^{-1}\hat{p} = -C^{-T}g$$

Now, convergence rates depend on the eigenvalues of $C^{-T}HC^{-1}$. So, try to choose a *preconditioning matrix* C such that

- C is positive definite
- $\text{cond}(C^{-T}HC^{-1}) \ll \text{cond(H)}$ *OR* eigenvalues of $C^{-T}HC^{-1}$ are clustered
- C^{-1} is easily calculated

Why? The system $C^{-T}HC^{-1}\hat{p} = -C^{-T}g$ is cheaper to solve.

For $\mathcal{F}$, consider setting $C = \text{Hess}H(Y_N|Y)$, a diagonal matrix.

Constrained/Bent Line Searches

Goal: Build a sequence $\{q_k\}_{k=1}^{\infty}$ of approximates to q^* such that $\mathcal{F}_k \geq \mathcal{F}_{k+1}$ for all k, $||\nabla \mathcal{L}_k|| \to 0$ as $k \to \infty$ ($\implies$ the constraints $\{c_i(q)\}$ are satisfied)

Idea: At q_k, find a search direction p_k, then "bend" (project) it so that q_{k+1} remains feasible. That is,

- p_k must be a descent direction: $\nabla \mathcal{F}_k^T p_k < 0$
- constraints must be satisfied

  $\nabla c_i(q)^T p_k \geq 0$ for inequality constraints

  $\nabla c_i(q)^T p_k = 0$ for equality constraints
- From q^*, there can not exist a direction p that satisfies the above two criteria. That is, for some $\lambda \geq 0$

  $$\nabla \mathcal{F}(q^*) = \lambda \nabla c_i(q^*)$$

  Formally, for p^* (that satisfies the *Linearly Independent Constraint Qualification*) $\exists \lambda$ that satisfies the *Karush-Kuhn-Tucker* or *KKT* conditions.

Problem: The projection can be expensive. So bent line searches work well for simple inequality constraints: $q(y_N \mid y) \geq 0 \quad \forall\ y \in Y$ and $\forall\ y_N \in Y_N$

Projected Gradient Method

Idea: Take steepest descent direction: $p_k = q_k - \max(q_k - \nabla \mathcal{F}_k, \vec{\eta})$

- Deals with the non-negativity constraints
- Convergence is linear.
- Cost is low

How to deal with the constraint $\sum_{y_N} q(y_N \mid y) = 1 \quad \forall\ y \in Y$?

- Rewrite $\mathcal{F}$ and $q(y_N \mid y)$??
- Normalize??

Projected Newton and Quasi Newton Methods

Idea: Let

$$p_k = -{H_{\mathrm{Red}}}_k^{-1} \nabla \mathcal{F}_k$$

where H_{Red} is the ***reduced Hessian***, a semi-positive definite matrix:

$$[H_{Red}]_{ij} = \begin{Bmatrix} \delta_{ij} \quad \text{if either} \quad c_i(q) \quad \text{or} \quad c_j(q) \quad \text{are active} \\ [\mathrm{Hess}\mathcal{F}]_{ij} \quad \text{otherwise} \end{Bmatrix}$$

Why?
Convergence is superlinear

Newton Projection Methods behave like **steepest descent** on the active constraints and like **Newton/Quasi-Newton Methods** on the inactive constraints. Rewrite:

$$q = \begin{bmatrix} q_I \\ q_A \end{bmatrix}, \nabla \mathcal{F}(q) = \begin{bmatrix} \nabla \mathcal{F}_I \\ \nabla \mathcal{F}_A \end{bmatrix}, H_{Red} = \begin{bmatrix} H_I & 0 \\ 0 & I \end{bmatrix}$$

Then

$$\mathrm{p}_k = -{H_{\mathrm{Red}}}_k^{-1} \nabla \mathcal{F}_k = \begin{bmatrix} -{H_I}_k^{-1} \nabla \mathcal{F}_{Ik} \\ -\nabla \mathcal{F}_{Ak} \end{bmatrix}$$

How to deal with the constraint $\sum_{y_N} q(y_N \mid y) = 1 \quad \forall \ y \in Y$?

Augmented Lagrangian

Goal: Want a fast, rigorous Quasi-Newton algorithm which takes into account all the constraints.

Idea: Incorporate the constraint $c_y(q) = 1 - \sum_{y_N} q(y_N \mid y)$ into a new function using penalty terms and explicit Lagrange Multiplier estimates at each optimization step:

- The new cost function to minimize, the Augmented Lagrangian:

  $$\mathcal{L}_A(q, \lambda^l, \mu_l) = \mathcal{F}(q) - \sum_y \lambda_y^l c_y(q) + \frac{1}{2\mu_l} \sum_y c_y(q)^2$$

  deals with $\sum_{y_N} q(y_N \mid y) = 1 \quad \forall\ y \in Y$

- A Projected Newton CG Line Search deals with the non-negativity constraints

- If $q^* = \mathrm{argmin}\mathcal{F}$ subject to the constraints $\{c_i(q)\}$, then $\exists\bar{\mu}$ such that $q^* = \mathrm{argmin}\mathcal{L}_A(q, \lambda^*, \mu)$ if $\mu \in (0, \bar{\mu}]$

- Introduction of Langrange multipliers avoids the ill-conditioning of *quadratic penalty methods* since theory tells us we don't need $\mu_l \to 0$

Implementation: There are three nested iterations:

- The Augmented Lagrangian or outer iteration (l)
- Optimization iteration or inner iteration (k)
- Line Search iteration

Details:

1. $q_l = \text{argmin}\mathcal{L}_A(q, \lambda^l, \mu_l)$

   Use a Projected Line Search with Wolfe Conditions

   CG computes the search direction p_k by solving

   $$H_{Redk} p_k = -\nabla \mathcal{L}_A(q^k, \lambda^l, \mu_l)$$

2. $\lambda_i^{l+1} = \lambda_i^l - c_i(q_l)\mu_l$
3. $\mu_{l+1} = s\mu_l$ such that $\mu_{l+1} < \mu_l$
4. Stop when both of the following occur:

   $||P_{[\eta,\infty)} \nabla \mathcal{L}_A(q^k, \lambda^l, \mu_l)|| \leq \tau_l$

   $||c_y(q)|| < \epsilon_l$

Justification: $\mathcal{L}_A$ is constructed so that it satisfies the KKT conditions:

$$\nabla \mathcal{L}_A = \nabla \mathcal{F} - \left(\lambda^l - \frac{c(q)}{\mu_l} \right) \nabla c^T(q)$$

So

$$\begin{aligned} &\nabla \mathcal{L}_A(q_l) = 0 \\ &\implies \nabla \mathcal{F} = \left(\lambda^l - \frac{c(q)}{\mu_l} \right) \nabla c^T(q) \\ &\implies \lambda^* = \lambda^l - \frac{c(q)}{\mu_l} \end{aligned}$$

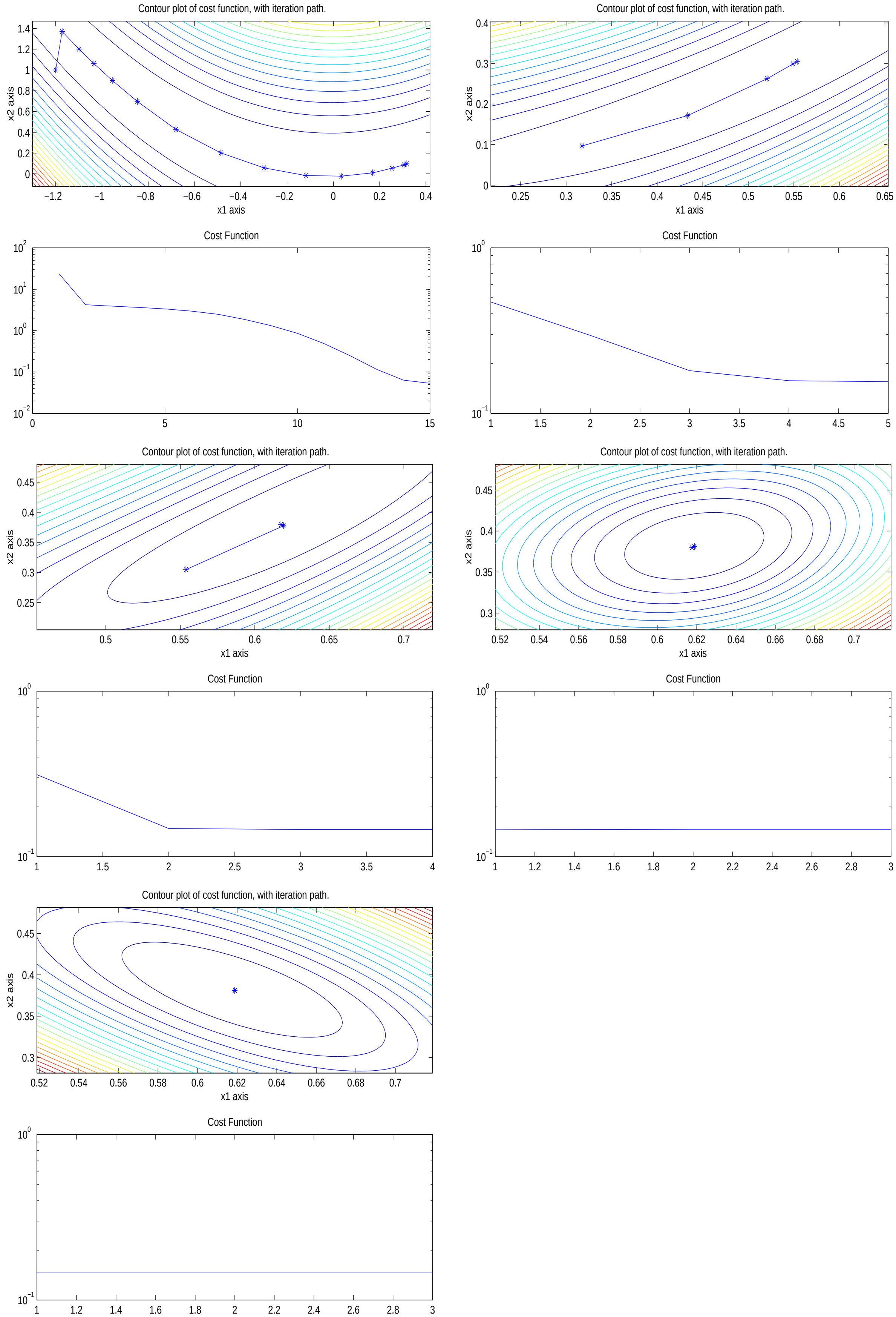


Figure 2: Path of $\{q_k\}$ for the Augmented Lagrangian Method for l=1,2,3, 4 and 5 applied to the Rosenbrock function subject to the constraints that $x_1 + x_2 = 1$

Numerical Results

Problem:

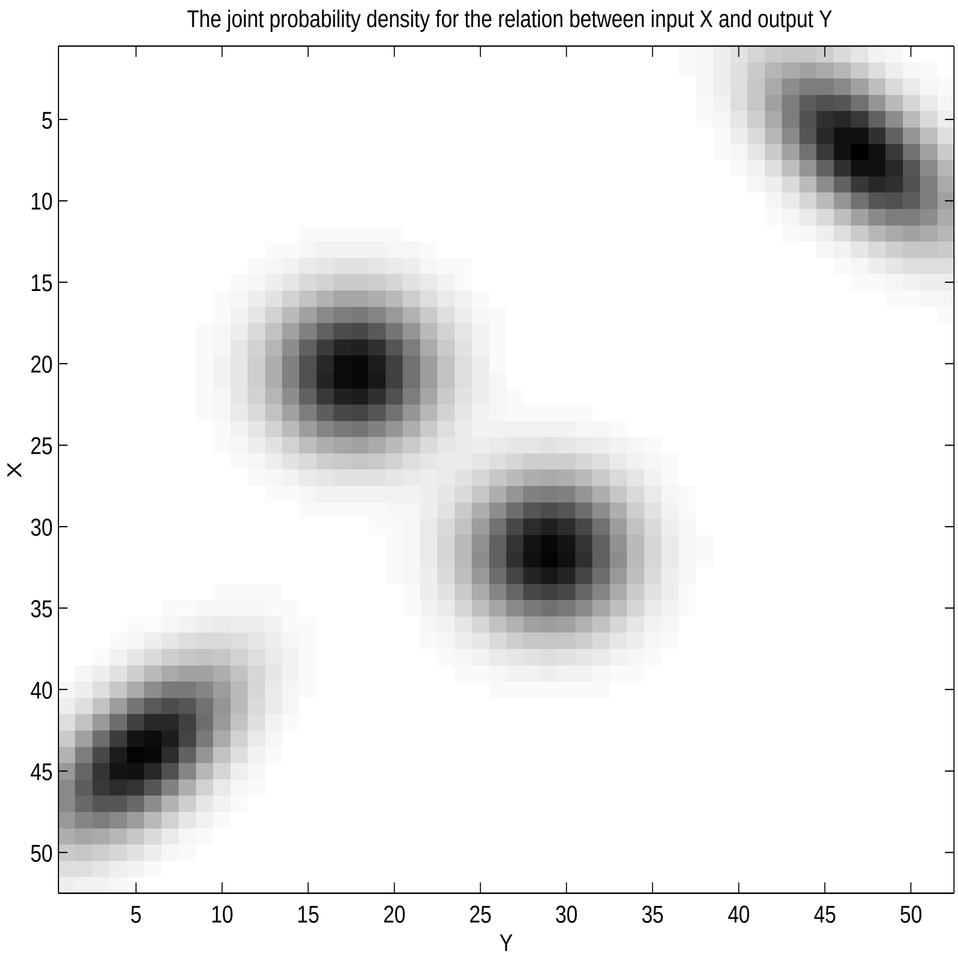


Figure 3: Synthetic Data: The four Blobs

Solution:

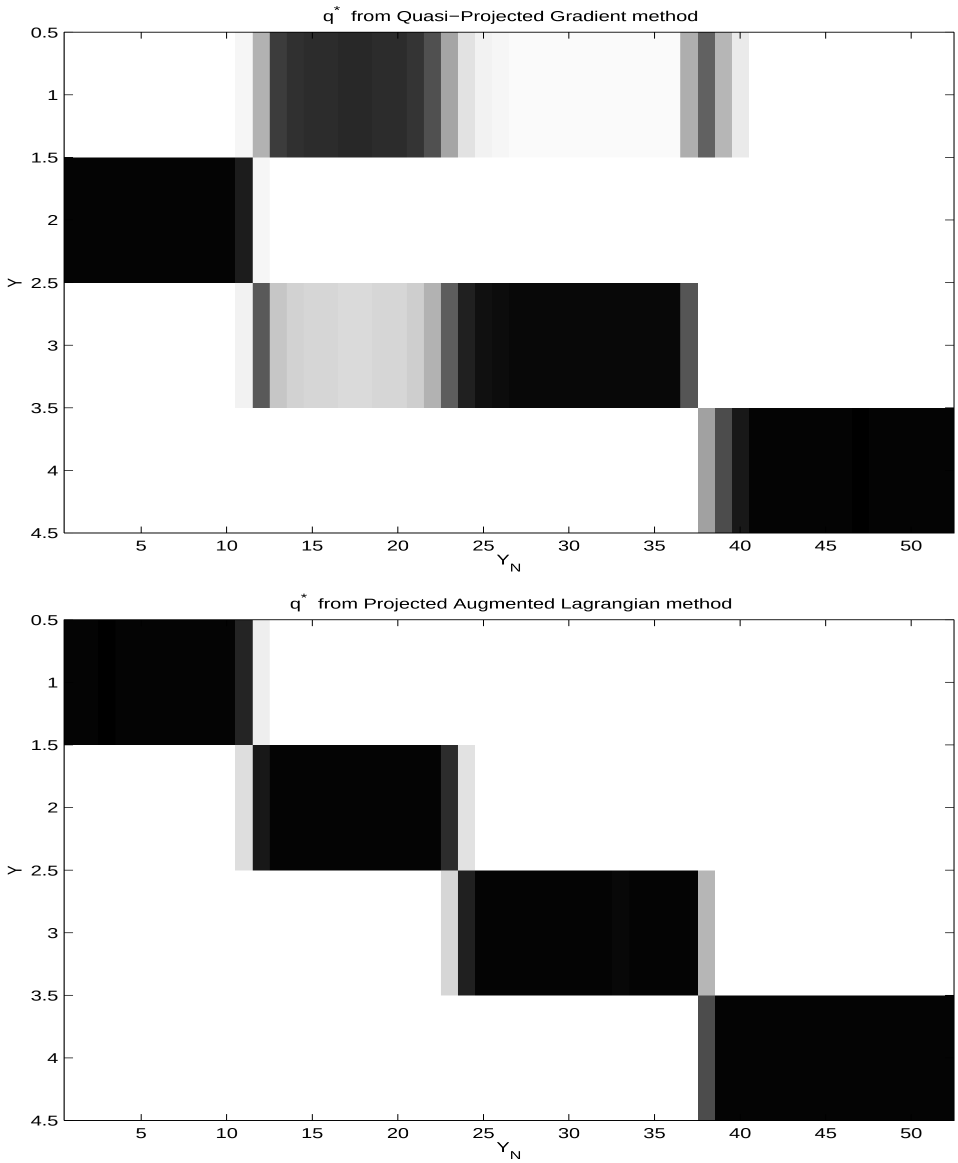


Figure 4: **Symmetric solutions**

COST ANALYSIS:
TOP: 4.8×10^8 flops.
BOTTOM: 5.2×10^{10} flops.
NOTE: Standard MATLAB optimization function *fmincon*: 5×10^{11}

Future Goals

- Preconditioned CG
- Apply optimization techniques to

$$\min_{q(y_N|y)} -H(Y_N \mid Y) \quad \text{constrained by}$$

$$D_{eff} \geq I_0$$

$$\sum_{y_N} q(y_N \mid y) = 1 \quad \forall\ y \in Y$$

$$q(y_N \mid y) \geq 0 \quad \forall\ y \in Y \quad \text{and} \quad \forall\ y_N \in Y_N$$