Computational Mathematics for Novices

Curt Vogel, Montana State University

CfAO 2002 Fall Retreat

Outline

- Simple Model Problem: Image Deblurring
  - Yields large stuctured systems of equations
- “Exact” vs. “Approximate” Solution Methods
- Computational Complexity
- Iterative Solution Methods for Linear Systems
  - Conjugate Gradient (CG) Iteration
  - Preconditioning
  - Illustration: CG with FFT-based Preconditioning for image deblurring
- Harder Example: Wavefront Reconstruction
  - CG with multigrid preconditioning for Ex-AO wavefront reconstruction

Model for Image Formation

A very general continuous linear model for image formation is

$$g(x,y) = \underbrace{\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s(x,y,x',y')\, f(x',y')\, dx'dy'}_{(\mathcal{S}f)(x,y)}$$

- f is the object.
- s is the point spread function (PSF).
- g is the (continuous) blurred image.
- (x',y') denotes source location.
- (x,y) denotes sensor location.

Atmospheric Optics Example

The PSF is

$$s(x, y, x', y') = \left| \int \int e^{-i2\pi(xp+yq)} \, A(p, q) e^{i\phi(p,q;x',y')} dp \, dq \right|^2 .$$

- (p, q) denotes position in the aperture plane.
- $A(p, q)$ is the aperture mask.
- $\phi(p, q; x', y')$ is the aperture-plane phase, or wavefront aberration, in direction (x', y').

With incoherent light sources,

- Object f represents a photon density.
- Image $g = \mathcal{S}f$ also represents a photon density.

Discrete Model

CCD array measures photon count of image g over subregions Ω_{ij} (pixels) in the image plane. Gives

$$d_{ij} = \int\int_{\Omega_{ij}} \underbrace{(\mathcal{S}f)(x,y)}_{g(x,y)}\, dx\, dy + \tilde{\eta}_{ij},$$

where $\tilde{\eta}_{ij}$ is noise due to image photon counting process (Poisson), background photon count (Poisson), dark current (Gaussian).
Apply numerical quadrature (e.g., trapaziod rule) to obtain fully discrete model, $\mathbf{d} = S\mathbf{f} + \boldsymbol{\eta}$, with component form

$$d_{ij} = \sum_{i'=1}^{n'_x} \sum_{j'=1}^{n'_y} s_{iji'j'} f_{i'j'} + \eta_{ij}, \quad 1 \le i \le n_x,\ 1 \le j \le n_y.$$

Linear Object Estimation

Apply maximum a posterior (MAP) estimation for the object, given data $\mathbf{d} = S\mathbf{f} +$ noise,

$$\hat{\mathbf{f}} = \arg\max_{\mathbf{f}} \{ \underbrace{L(S\mathbf{f}|\mathbf{d})}_{\text{log likelihood}} + \underbrace{P(\mathbf{f})}_{\text{prior}} \}$$

Assumption of iid (white) Gaussian noise and iid Gaussian prior give penalized linear least squares problem

$$\begin{aligned} \hat{\mathbf{f}} &= \arg\min_{\mathbf{f}} \{||S\mathbf{f} - \mathbf{d}||^2 + \alpha||\mathbf{f}||^2\} \\ &= (\underbrace{S^T S + \alpha I}_{A})^{-1} \underbrace{S^T \mathbf{d}}_{\mathbf{b}}. \end{aligned}$$

Need to solve linear system $A\hat{\mathbf{f}} = \mathbf{b}$.

Some Details

- $\arg\max$ (or $\arg\min$) denotes "the value which maximizes (or minimizes)".
- $||\cdot||^2$ denotes squared Euclidean norm of an array,

$$||\mathbf{f}||^2 = \sum_{i'=1}^{n'_x} \sum_{j'=1}^{n'_y} f_{i'j'}^2$$

- The log likelihood function $L(S\mathbf{f}|\mathbf{d})$ is a measure of "fit-to-data".
- The prior $P(f)$ is a measure of "prior information".
- α is a trade-off parameter between fit-to-data and prior information. In the iid Gaussian case, α is the reciprocal of the squared signal-to-noise ratio.

Nonlinear Object Estimation

Again apply MAP estimation, but assume Poisson noise, with ij^{th} Poisson parameter $\lambda_{ij} = [S\mathbf{f}]_{ij} + b$, and iid Gaussian prior.

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \underbrace{\{\sum_{i=1}^{n_x}\sum_{j=1}^{n_y}[S\mathbf{f}]_{ij} + b - \sum_{i=1}^{n_x}\sum_{j=1}^{n_y} d_{ij}\log([S\mathbf{f}]_{ij} + b) + \alpha||\mathbf{f}||^2\}}_{J(\mathbf{f})}$$

Can apply iterative methods like Newton's method. Requires solution of sequence of linear systems. Limit of the sequence of iterates is the minimizer of J.

“Exact” vs. “Approximate” Solutions

- Any model is “inexact”, so all solutions are “approximate”.
- The Gaussian MAP solution is easier to compute.
- Using Newton iteration, each Poisson MAP iterate costs as much to compute as one Gaussian MAP solution.
- For low photon counts and negligible read noise, the Poisson MAP solution is “more realistic”. Gives better reconstructions, provided enough iterations are taken.

Need an “error budget” for iterative methods.

- Trade off accuracy vs. computational cost, both of which increase with iteration count.

Computational Complexity

Measure of asymptotic cost of a computational method.

Example: Solution of $N \times N$ linear system $A\mathbf{x} = \mathbf{b}$ using Gaussian elimination algorithm. Cost in terms of floating point multiplications is

$$\frac{1}{3}N^3 + \mathcal{O}(N^2).$$

- Measure is independent of computer architecture. To get estimate of "total clock time", compute

  complexity $\times$ clock time per floating point multiplication

Complexity of Image Deblurring

Assume a CCD array with $N = 1024^2 \approx 10^6$ pixels. Applying penalized least squares (MAP with iid Gaussian statistics), need to solve $N \times N$ linear system $A\mathbf{f} = \mathbf{b}$. Using the Gaussian elimination algorithm,

- Complexity (measured in floating point multiplications) is $\approx \frac{1}{3}10^{18}$.
- Storage is $\approx 10^{12}$.

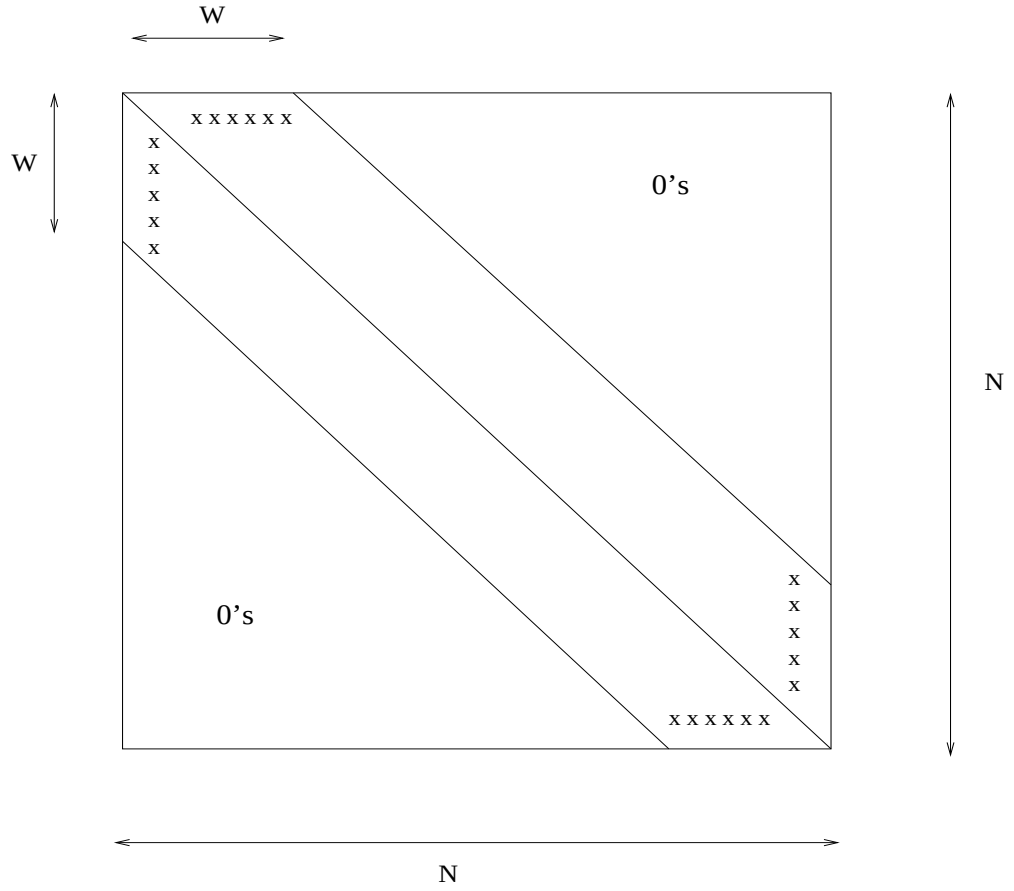
Assuming a computer with 10^9 floating point multiplications per second, total clock time to solve system is $\approx 10^9/3$ seconds, or ≈ 11 years!

- Parallel computers won't help, because Gaussian elimination is inherently sequential.

Sparse, Banded Structure

Matrix A is called sparse & banded with bandwidwidth w if

$$a_{ij} = 0 \text{ whenever } |i - j| > w.$$



Sparse, Banded Structure, Continued

- If bandwidth $w \ll N$, can modify Gaussian elimination so that
  - Storage is $N \times w$.
  - Complexity is $N \times w + \mathcal{O}(N)$.

Application: Solution of Laplace's equation on $n \times n$ grid. System size $N = n^2$, band width $w = n$, and complexity is $N^{3/2}$, in contrast with $\frac{1}{3}N^3$ for unstructured Gaussian elimination.

CfAO Application: Astronomical AO reconstructor computations. Ref: Ellerbroek, JOSA-A, September 2002.

Shift Invariance

A is 1-D shift invariant, or Toeplitz, if it is const along diags.

$$A = \begin{bmatrix} a_0 & a_{-1} & \cdots & a_{2-N} & a_{1-N} \\ a_1 & a_0 & a_{-1} & \ddots & a_{2-N} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{N-2} & \ddots & a_1 & a_0 & a_{-1} \\ a_{N-1} & a_{N-2} & \cdots & a_1 & a_0 \end{bmatrix}.$$

Matrix is called 2-D shift invariant, or block Toeplitz, if (i) it has a block decomposition of the above form; and (ii) each of the blocks is Toeplitz.

CfAO Applications: Deblurring when phase is independent of source direction, covariance computations in (MC)AO.

Shift Invariance with Wrap-Around

A is called 1-D shift invariant with wrap-around, or circulant, if it is Toeplitz and its rows are circular right shifts of the elements of the preceding row:

$$A = \begin{bmatrix} a_0 & a_{N-1} & \cdots & a_2 & a_1 \\ a_1 & a_0 & a_{N-1} & \cdots & a_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{N-2} & \ddots & a_1 & a_0 & a_{N-1} \\ a_{N-1} & a_{N-2} & \cdots & a_1 & a_0 \end{bmatrix}.$$

Matrix is 2-D shift invariant with wrap-around, or block circulant, if (i) it is block Toeplitz; (ii) the block rows are circular right shifts of the block elements of the preceding row; and (iii) each block is a circulant matrix.

FFT Implementation

Let A be circulant (shift-invariant with wrap-around); let FFT$(\cdot)$ and IFFT$(\cdot)$ denote the discrete Fourier transform and its inverse, implemented via the FFT algorithm; and take 1st row $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1})$ of A. Given an N-vector $\mathbf{v}$,

$$A\mathbf{v} = \text{IFFT}(\underbrace{\text{FFT}(\mathbf{a}) .* \text{FFT}(\mathbf{v})}_{\text{component-wise product}})$$

$$A^{-1}\mathbf{v} = \text{IFFT}(\underbrace{\text{FFT}(\mathbf{v}) ./ \text{FFT}(\mathbf{a})}_{\text{component-wise quotient}})$$

Complexity of FFT$(\mathbf{v})$ is $\frac{1}{2} N \log_2(N) + \mathcal{O}(N)$.

- Can solve $A\mathbf{x} = \mathbf{b}$ with complexity $\frac{3}{2} N \log_2(N)$.

FFT-Based Image Deblurring

Assuming phase ϕ independent of source direction and appropriate object discretization (e.g., 2-D midpoint quadrature), the discrete PSF is $s_{iji'j'} = s_{i-i',j-j'}$, and the blurring matrix S is block Toeplitz (2-D shift invariant). If S is block circulant (add wrap-around), can compute penalized least squares solution (MAP estimate with Gaussian stats)

$$\mathbf{f} = (S^T S + \alpha I)^{-1} S^T \mathbf{d}$$

using 2-D FFTs:

$$f = \mathsf{IFFT2}(\mathsf{conj}(\mathsf{FFT2}(s)) .* \mathsf{FFT2}(d) ./ (|\mathsf{FFT2}(s)|^2 + \alpha))$$

Complexity for $N \approx 10^6$ is $\frac{3}{2} N \log_2(N) \approx 3 \times 10^7$, in contrast to $\frac{1}{3} N^3 \approx 3 \times 10^{17}$ for Gaussian elimination!

Problems with "Direct FFT" Approach

S is not block circulant. We don't have wrap-around (discrete PSF is not doubly periodic).

- Direct FFT approach can yield "periodic artifacts". These can seriously degrade features near the boundary of the computational domain.
  - These artifacts can be mitigated, but not eliminated, by "zero padding", but this increases the storage and the complexity.

Good news: With zero padding (ignoring round-off error), we can exactly compute BTTB matrix-vector products using FFTs, with $N \log_2(N)$ complexity.

Question: Can this be used to get better approximations?

Iterative Methods for Linear Systems

- Large size of A makes solution by direct decomposition methods (e.g., Gaussian elimination) impractical.
  - "Brute force" parallelization doesn't help.
- In certain cases, approximate solution methods (e.g., FFT-based approx soln to shift-invariant systems) may be used.
  - Are these stable? Are these accurate enough?
- An alternative is iterative solution methods.
  - These methods are exact in the limit, i.e., they can be made arbitrarily accurate by taking arbitrarily many iterations.
  - Can be combined with approximate solution methods to speed up convergence by preconditioning.

Conjugate Gradient (CG) Algorithm

Standard iterative method for $N \times N$ symmetric positive definite (SPD) linear systems $A\mathbf{x} = \mathbf{b}$. SPD matrices commonly arise in optimization and control.

- Gives sequence of iterates $\mathbf{x}_k \approx A^{-1}\mathbf{b},\ k = 0, 1, \ldots$.
- Complexity = cost per iteration $\times$ number of iterations.
- Cost per iteration = cost of one matrix-vector multiplication + $\mathcal{O}(N)$.
  - If A is sparse with $\mathcal{O}(N)$ nonzero entries, then total cost per iteration is $\mathcal{O}(N)$.
  - If A is Toeplitz or BTTB (1- or 2-D shift invariant), then total cost per iter using FFTs is $\mathcal{O}(N \log_2 N)$.
  - Can often easily exploit parallelism.
- Method is optimal in certain weighted least squares sense.

The Gory Details

Goal: Given SPD matrix A and vector $\mathbf{b}$, compute

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \left[\frac{1}{2} \mathbf{x}^T A \mathbf{x} - \mathbf{x}^T \mathbf{b} \right] = A^{-1} \mathbf{b}.$$

CG yields sequence of approximations $\mathbf{x}_k$ to $\mathbf{x}^*$.

- k^{th} CG iterate has form (when $\mathbf{x}_0 = 0$) $\mathbf{x}_k = \hat{p}_k(A)\mathbf{b}$ $= \sum_{j=0}^{k} c_j A^j \mathbf{b}$, where $\hat{p}_k$ is a polynomial of degree k.
- $\mathbf{x}_k$ is optimal in that the weighted least squares norm

$$||\mathbf{x}_k - \mathbf{x}^*||_A^2 \stackrel{\text{def}}{=} (\mathbf{x}_k - \mathbf{x}^*)^T A (\mathbf{x}_k - \mathbf{x}^*)$$

  is minimized over all polynomials p_k of degree $k-1$.
- Rate of convergence depends on relative spread of eigenvalues of A.

Preconditioned CG (PCG)

In principle, apply CG to transformed problem

$$\underbrace{M^{-1/2}AM^{-1/2}}_{\tilde{M}}\underbrace{M^{1/2}x}_{\tilde{x}} = \underbrace{M^{-1/2}b}_{\tilde{b}}$$

where preconditioning matrix M is also SPD.

- PCG algorithm is formulated so each iteration requires
  - One matrix-vector product $A\mathbf{v}_k$, for some $\mathbf{v}_k$.
  - One matrix-vector product $M^{-1}\mathbf{w}_k$.
- If relative spread of eigenvalues of $M^{-1}A$ is small, then PCG convergence is fast.

Fast Convergence + Cheap M^{-1} $\Rightarrow$ Efficient PCG

Examples of Preconditioners

- Sparse Choleski matrix factorizations for sparse linear systems. Works well for MCAO fitting step. Key idea: Take $A \approx LL^T = M$, where L is lower triangular and sparse. Implement by forward elimination followed by back substitution. Complexity of M^{-1} is $\mathcal{O}(N)$.
- Sparse approximate inverse. Key idea: Construct sparse approximation to A^{-1} (possibly with low-rank terms added on). Resembles what Dekany et al are doing in wavefront reconstruction, but it is more cost-effective to use this as a preconditioner than as an approximate solution method.
- (Block) circulant preconditioners for (block) Toeplitz (i.e., 1- or 2-D shift invariant) systems. Work well for image deblurring, covariance matrix inversion. Complexity is $\mathcal{O}(N \log_2(N))$.

Basics of AO Wavefront Reconstruction

Recall that the PSF can be expressed as

$$s = |\mathcal{F}\{Ae^{i\phi}\}|^2,$$

where $\mathcal{F}$ denotes 2-D Fourier transform and ϕ denotes the aperture-plane phase. In ordinary AO (but not MCAO), this is assumed to be independent of light source direction. The "flatter" ϕ is, the more "delta function-like", the PSF is.

- Basic Idea for AO: Put a deformable mirror (DM) in the aperture plane to "flatten" ϕ.
- In multiconjugate AO (MCAO), need several deformable mirrors, each conjugate to a different height, to compensate for directional dependence of ϕ on source direction.

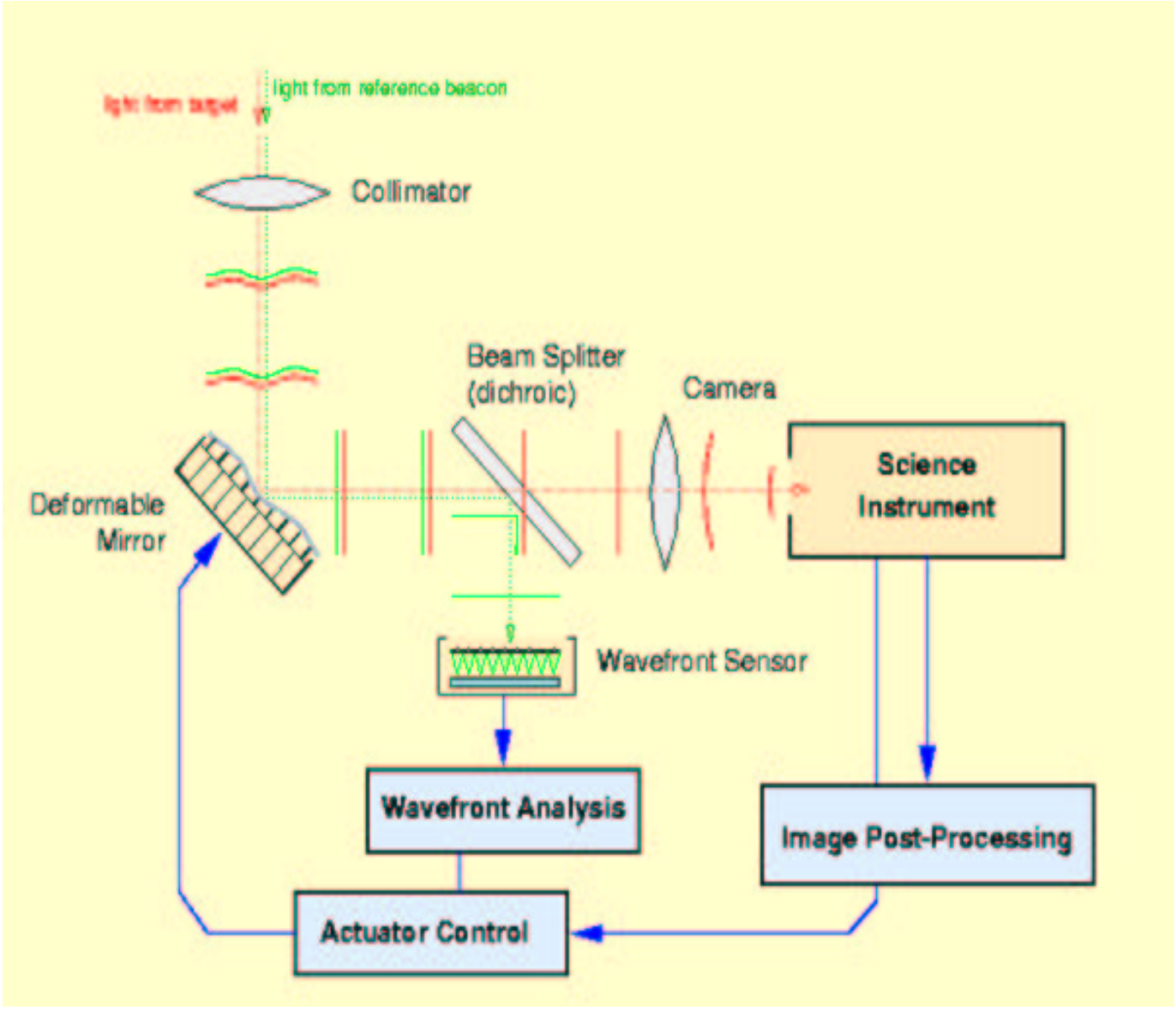
Reconstruction Basics, Continued

We can measure the slope of the ϕ at discrete points using a device called a wavefront sensor (WFS).

- For closed loop control, we actually measure the "compensated" phase, after it has been flattened by the DM.
- For MCAO, measure phase in several different directions.

Goal of AO Wavefront Reconstruction: Given WFS measurements s, compute actuator deformations a which determine the shape of the DM to flatten aperture-plane phase ϕ.

Schematic Diagram of AO System



Linear, Open-Loop Reconstruction

Assumes a linear relationship between the actuator commands a (which determine the mirror deformations) and the WFS meaurements s,

$$a = Rs.$$

R is called the wavefront reconstructor matrix.

- Let $\hat{\phi} = Qa$ represent the actuator influence. This is the aperture-plane phase compensation resulting from an actuator command vector a.
- Let $\phi = Px$ denote represent the influence of the phase due to an atmospheric turbulence profile x. In the MCAO case P is a projection (line integration) operator. In the AO case, P is the identity.

The corrected phase is "flat" if $\hat{\phi} - \phi$ is "small".

Minimum Variance Reconstruction

- Model the atmospheric turbulence profile (and hence, ϕ) as a "wide-sense stationary" stochastic process (i.e., if you subtract off the "slowly-varying mean", it has a covariance which is translation invariant).
- WFS measurements s also have a mean and covariance, which depend on stats of x (ϕ) and on stats of WFS noise.
- Let $\langle \cdot, \cdot \rangle$ denote expected value ("average").

Choose reconstruction matrix R to minimize

$$\begin{aligned} J(R) &\stackrel{\text{def}}{=} \langle ||\hat{\phi} - \phi||^2 \rangle \\ &= \langle ||Qa - Px||^2 \rangle \\ &= \langle ||QRs - Px||^2 \rangle \end{aligned}$$

Min Variance Reconstruction, Continued

Taking the derivative of J w.r.t. R and setting it equal to zero gives

$$QR \left\langle ss^T \right\rangle = P \left\langle xs^T \right\rangle.$$

Solving for R gives the optimal (in expected least squares sense) open loop reconstructor,

$$\hat{R} = \underbrace{Q^\dagger P}_{F} \underbrace{\left\langle xs^T \right\rangle \left\langle ss^T \right\rangle^{-1}}_{E}$$

- E is called the estimation matrix.
- F is called the fitting matrix.
- $Q^\dagger$ denotes pseudoinverse, or generalized inverse, of Q.

Estimation Matrix

Assume linear, additive noise model for WFS measurements,

$$s = \Gamma \underbrace{Px}_{\phi} + n.$$

Γ denotes discrete gradient (wavefront slope) operator; n denotes WFS noise.

Assume x, n are independent, zero mean random vectors with covariances C_x, C_n. Then the estimation matrix is

$$\begin{aligned} E &= \left\langle x s^T \right\rangle \left\langle s s^T \right\rangle^{-1} \\ &= C_x P^T \Gamma^T \left(\Gamma P C_x P^T \Gamma^T + C_n \right)^{-1} \\ &= \left(P^T \Gamma^T C_n^{-1} \Gamma P + C_x^{-1} \right)^{-1} P^T \Gamma^T C_n^{-1} \end{aligned}$$

Reconstructor Matrix Computation

Given WFS measurement s, mirror actuator command is

$$a = Rs = F \underbrace{E\, s}_{x}$$

- Vector $x = Es$ is solution to "estimation equation"

$$(P^T \Gamma^T C_n^{-1} \Gamma P + C_x^{-1}) x = P^T \Gamma^T C_n^{-1} s.$$

- Vector a is solution to "regularized fitting equation"

$$(Q^T Q + \alpha I) a = Q^T P x.$$

(Note $Q^\dagger = (Q^T Q)^{-1} Q^T$.)

Linear Algebraic Structure

Need to solve $Au = b$, where

$A = P^T \Gamma^T C_n^{-1} \Gamma P + C_x^{-1}$ in estimation step ($u = x$).

$A = Q^T Q + \alpha I$ in fitting step ($u = a$).

- In both steps, A is symmetric and positive definite (SPD). Can apply CG iteration.
- A is sparse (most entries are zero) in fitting step.
- In estimation step, projection operator P is sparse; discrete gradient Γ is sparse; atmospheric covariance C_x is block Toeplitz (2-D shift invariant); with natural guide stars, C_n is diagonal.

In MCAO case, all matrices have additional block structure, due to layered structure of atmosphere.

Estimation Step for Ex-AO

Assume only one layer (phase screen), so $P = I$, $x = \phi$. Also assume noise covariance $C_n = \sigma_n^2 I$, so we need to invert

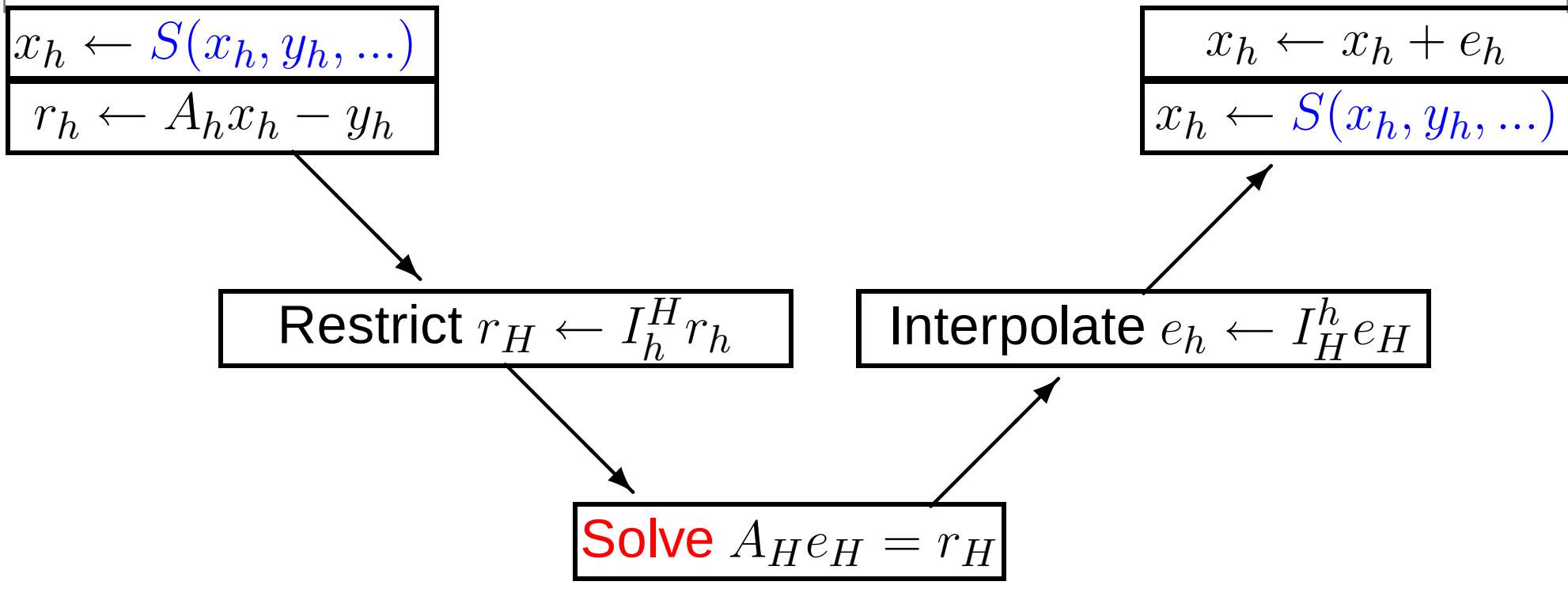
$$A = \Gamma^T \Gamma + \sigma_n^2 C_x^{-1}$$

- $\Gamma^T \Gamma$ acts like a discrete Laplacian operator.
- C_x^{-1} behaves like a discretize biharmonic operator (squared Laplacian).
- Both the Laplacian and biharmonic can be efficiently inverted using multigrid methods.

Multigrid (MG) Methods

- Can sometimes be used as “stand-alone” system solvers.
- Can be used as preconditioners.
- Rely on multiple scales (grid sizes) inherent in certain problems.
  - Need “smoother” which damps out high-frequency components of error on fine grids.
    - Classical Jacobi iteration and Gauss-Seidel iteration both work well as smoothers for Laplace’s equation.
  - Remaining low frequency error is well-represented on coarser grids.
- Are recursive versions of the following 2-grid scheme.

2-Grid Scheme



- $S(v, w, ...)$ denotes application of smoother to solve $Ax = w$ with initial guess $x = v$.
- To obtain MG V-cycle, apply 2-grid scheme recursively. Carry out Solve step with (e_H, r_H) in place of (x_h, y_h).

Multigrid, Continued

- Inter-grid transfers (restriction, or up-binning, and interpolation, or down-binning) are cheap.
- Cost is typically dominated by smoother application on finest grid.
- Choice of smoother is problem-dependent.
  - Block (i.e., layer-oriented) symmetric Gauss-Seidel (B-SGS) works well for MCAO estimation step.
  - FFT-based modified Richardson iteration works well for Ex-AO estimation.

Modified Richardson for EX-AO

Based on "splitting"

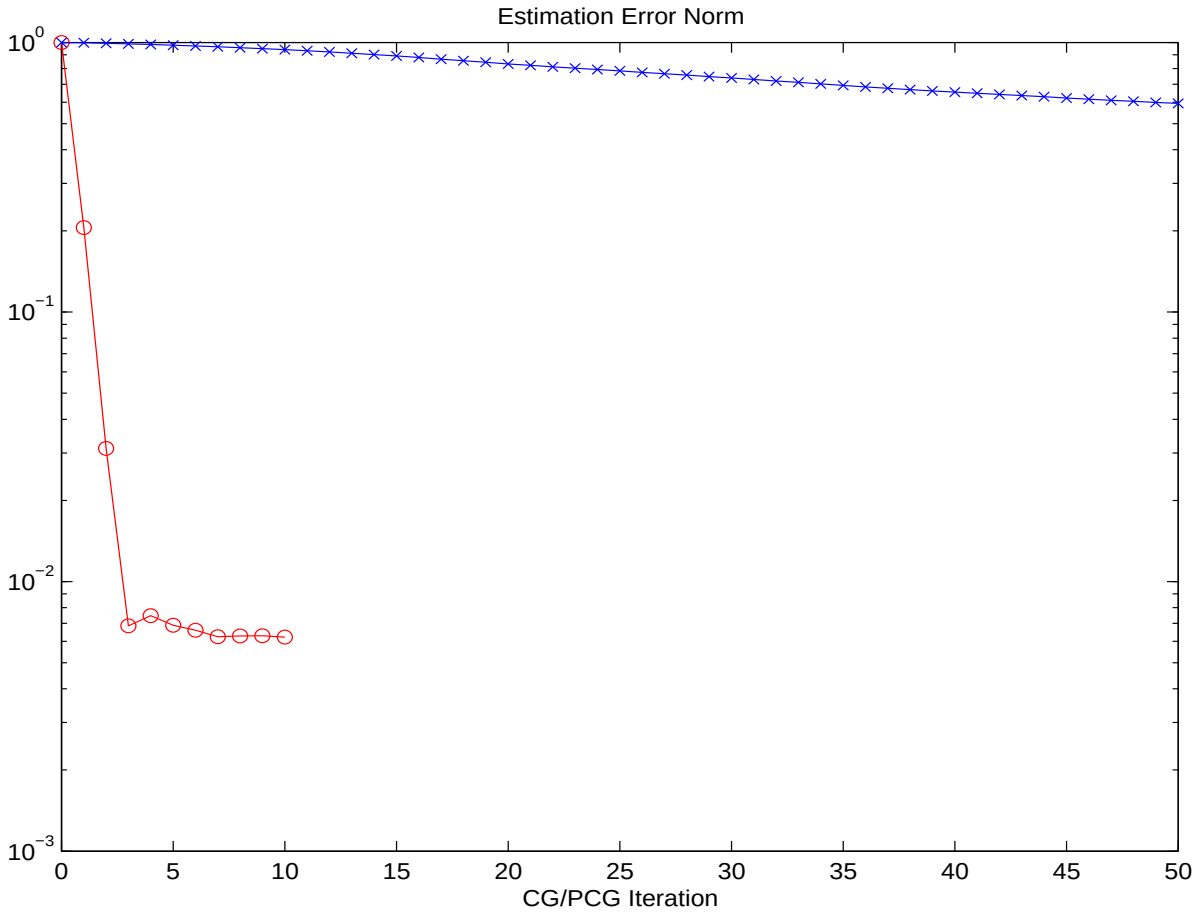
$$A = (\Gamma^T\Gamma - \omega I) + (\omega I + \sigma_n^2 C_x^{-1})$$

Solve $A\mathbf{x} = \mathbf{b}$ using following iteration:

$$(\omega I + \sigma_n^2 C_x^{-1})\mathbf{x}_{k+1} = \mathbf{b} + \omega\mathbf{x}_k - \Gamma^T\Gamma\mathbf{x}_k, \quad k = 0, 1, \ldots.$$

- Invert $\omega I + \sigma_n^2 C_x^{-1}$ on left hand side using FFTs.
- $\Gamma^T\Gamma$ on right hand side is sparse.
- Cost per iteration is $\mathcal{O}(N \log_2(N))$.
- Pick ω to be largest (positive) eigenvalue of $\Gamma^T\Gamma$.
- Only need 1 modified Jacobi iter to damp out high frequencies. Hence, cost of multigrid iteration is $\mathcal{O}(N \log_2(N))$.

Results for Ex-AO Estimation



Blue x's represent phase estimation error norm for CG with no preconditioning. Red o's correspond to PCG with MG preconditioning (SNR = 10, no. d.f. = 12556).

References

1. B. L. Ellerbroek, “Efficient computation of minimum variance wavefront reconstructors using sparse matrix techniques,” JOSA-A, **19** (2002).
2. L. Gilles, C. R. Vogel, and B. L. Ellerbroek, “A multigrid preconditioned conjugate gradient method for large scale wavefront reconstruction”, JOSA-A, **19** (2002), pp. 1817-1822.
3. Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996.
4. C. Vogel, *Computational Methods for Inverse Problems*, SIAM, 2002.
5. U. Trottenberg, C. W. Oosterlee and A. Schüller, *Multigrid*, Academic Press, London, 2001.