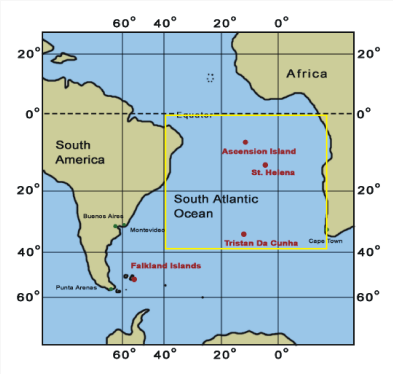
Polynomial accelerated MCMC ...
and other sampling algorithms inspired by computational optimization

Colin Fox fox@physics.otago.ac.nz
Al Parker, John Bardsley

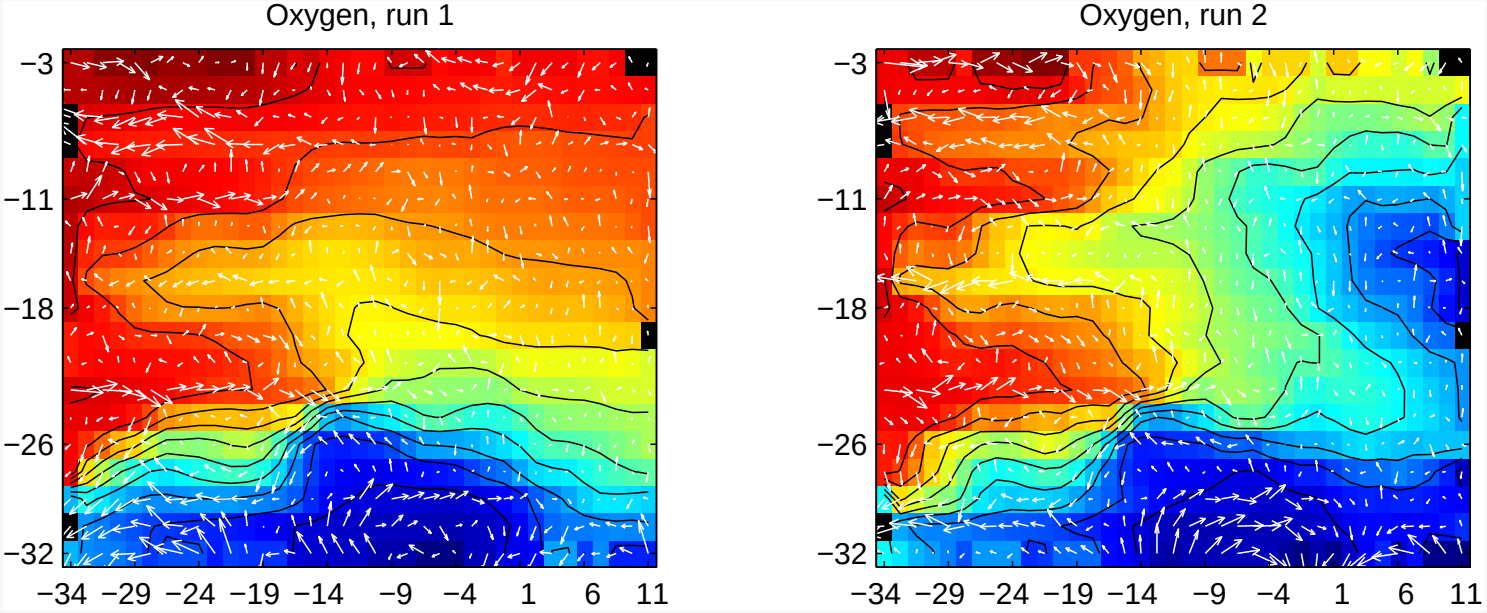
Fore-words

- Motivation: an inverse oceanography problem
- Cartoon of polynomial acceleration of MCMC
- Equivalence of Gibbs & linear solvers
- Polynomial acceleration of Gibbs sampling
- Some other sampling algorithms from CSE

Ocean circulation :: 2 samples from the posterior



Data on traces, assert physics and observational models, infer abyssal advection



McKeague Nicholls Speer Herbei 2005 *Statistical Inversion of South Atlantic Circulation in an Abyssal Neutral Density Layer*

Accelerating MCMC (cartoon)

MH-MCMC repeatedly simulates fixed transition kernel $\mathcal{P}$ with

$$\pi\mathcal{P} = \pi$$

via detailed balance (self-adjoint in metric π)

n-step distribution: $\pi^{(n)} = \pi^{(n-1)}\mathcal{P} = \pi^{(0)}\mathcal{P}^n$

n-step distribution error: $\pi^{(n)} - \pi = \left(\pi^{(0)} - \pi\right)\mathcal{P}^n$

Accelerating MCMC (cartoon)

MH-MCMC repeatedly simulates fixed transition kernel $\mathcal{P}$ with

$$\pi\mathcal{P} = \pi$$

via detailed balance (self-adjoint in metric π)

n-step distribution: $\pi^{(n)} = \pi^{(n-1)}\mathcal{P} = \pi^{(0)}\mathcal{P}^n$

n-step distribution error: $\pi^{(n)} - \pi = \left(\pi^{(0)} - \pi\right)\mathcal{P}^n$

Initial error is multiplied by n-th order polynomial P_n of $(I - \mathcal{P})$ called the the error polynomial

Accelerating MCMC (cartoon)

MH-MCMC repeatedly simulates fixed transition kernel $\mathcal{P}$ with

$$\pi\mathcal{P} = \pi$$

via detailed balance (self-adjoint in metric π)

n-step distribution: $\pi^{(n)} = \pi^{(n-1)}\mathcal{P} = \pi^{(0)}\mathcal{P}^n$

n-step distribution error: $\pi^{(n)} - \pi = \left(\pi^{(0)} - \pi\right)\mathcal{P}^n$

error polynomial: $P_n(I - \mathcal{P}) = \mathcal{P}^n = (I - (I - \mathcal{P}))^n$ or $P_n(\lambda) = (1 - \lambda)^n$

Hence convergence is geometric for this iteration

Accelerating MCMC (cartoon)

MH-MCMC repeatedly simulates fixed transition kernel $\mathcal{P}$ with

$$\pi\mathcal{P} = \pi$$

via detailed balance (self-adjoint in metric π)

n-step distribution: $\pi^{(n)} = \pi^{(n-1)}\mathcal{P} = \pi^{(0)}\mathcal{P}^n$

n-step distribution error: $\pi^{(n)} - \pi = \left(\pi^{(0)} - \pi\right)\mathcal{P}^n$

error polynomial: $P_n(I - \mathcal{P}) = \mathcal{P}^n = (I - (I - \mathcal{P}))^n$ or $P_n(\lambda) = (1-\lambda)^n$

Hence convergence is geometric for this iteration

Polynomial acceleration: modify iteration to give better error polynomial e.g.

$$(\text{choose } x^{(i)} \text{ w.p. } \alpha_i) \sim \pi^{(0)} \sum_{i=1}^{n} \alpha_i \mathcal{P}^i \qquad \text{error poly: } Q_n = \sum_{i=1}^{n} \alpha_i (1-\lambda)^i$$

We choose α_i so can choose Q_n

Can build a *perfect* sampler when π over finite state space, or Gaussian(!)

Gibbs sampling from normal distributions

Gibbs sampling[a] repeatedly samples from (block) conditional distributions

Normal distributions

$$\pi(\mathbf{x}) = \sqrt{\frac{\det(\mathbf{A})}{2\pi^n}} \exp\left\{-\frac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} + \mathbf{b}^\mathsf{T}\mathbf{x}\right\}$$

precision matrix $\mathbf{A}$, covariance matrix $\Sigma = \mathbf{A}^{-1}$ (both SPD)

Mean $\bar{\mathbf{x}}$ satisfies

$$\mathbf{A}\bar{\mathbf{x}} = \mathbf{b}$$

Particularly interested in case where $\mathbf{A}$ is sparse (GMRF) and n large

When is $\pi^{(0)}$ is also normal, then so is the n-step distribution

$$\mathbf{A}^{(n)} \to \mathbf{A} \qquad \Sigma^{(n)} \to \Sigma$$

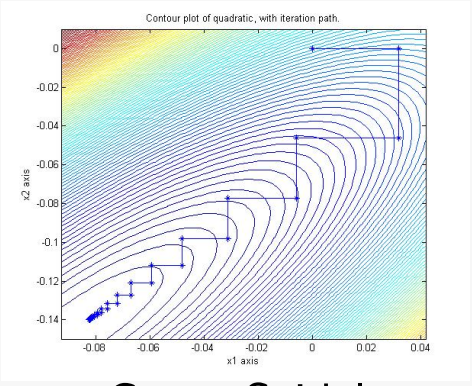
In what sense is "stochastic relaxation" related to "relaxation"?

What decomposition of $\mathbf{A}$ is this performing?

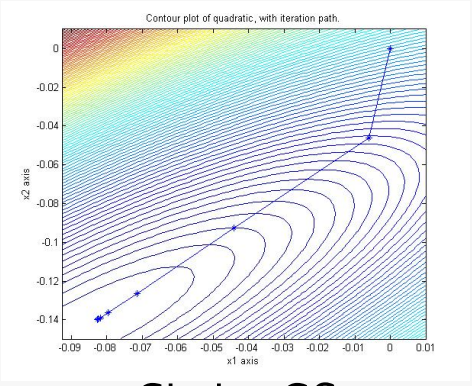
[a] Glauber 1963 (heat-bath algorithm), Turcin 1971, Geman and Geman 1984

Gibbs samplers and equivalent linear solvers

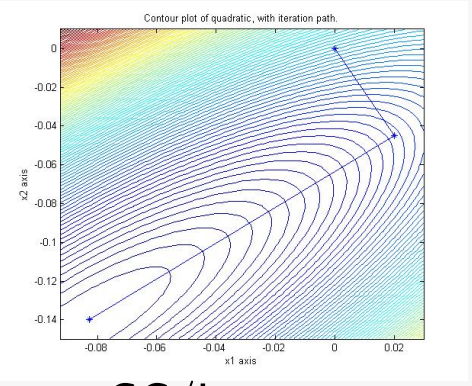
Optimization ...



Gauss-Seidel

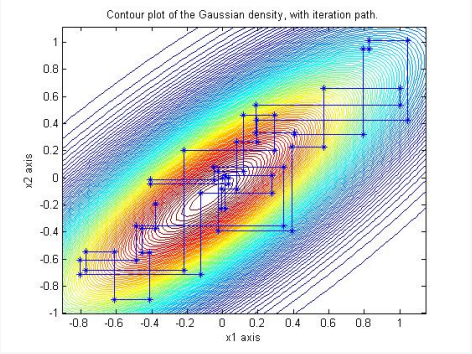


Cheby-GS

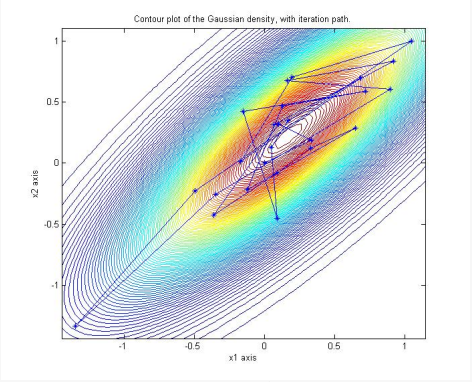


CG/Lanczos

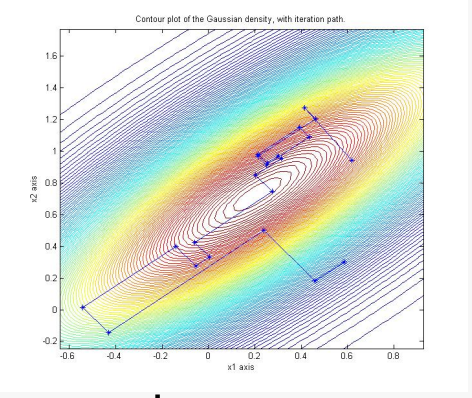
Sampling ...



Gibbs



Cheby-Gibbs



Lanczos

Matrix splitting form of stationary iterative methods

Want to solve

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

The *splitting* $\mathbf{A} = \mathbf{M} - \mathbf{N}$ converts $\mathbf{A}\mathbf{x} = \mathbf{b}$ to $\mathbf{M}\mathbf{x} = \mathbf{N}\mathbf{x} + \mathbf{b}$

If $\mathbf{M}$ is nonsingular

$$\mathbf{x} = \mathbf{M}^{-1}\mathbf{N}\mathbf{x} + \mathbf{M}^{-1}\mathbf{b}$$

Iterative methods compute successively better approximations by

$$\begin{aligned}\mathbf{x}^{(k+1)} &= \mathbf{M}^{-1}\mathbf{N}\mathbf{x}^{(k)} + \mathbf{M}^{-1}\mathbf{b} \\ &= \mathbf{G}\mathbf{x}^{(k)} + \mathbf{g}\end{aligned}$$

Many splittings use terms in $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$. Gauss-Seidel sets $\mathbf{M} = \mathbf{L} + \mathbf{D}$

$$\mathbf{x}^{(k+1)} = -\mathbf{D}^{-1}\mathbf{L}\mathbf{x}^{(k+1)} - \mathbf{D}^{-1}\mathbf{L}^{\mathsf{T}}\mathbf{x}^{(k)} + \mathbf{D}^{-1}\mathbf{b}$$

Matrix formulation of Gibbs sampling from $\mathrm{N}(0, \mathbf{A}^{-1})$

Let $\mathbf{y} = (y_1, y_2, ..., y_n)^T$

Component-wise Gibbs updates each component in sequence from the (normal) conditional distributions

One 'sweep' over all n components can be written

$$\mathbf{y}^{(k+1)} = -\mathbf{D}^{-1}\mathbf{L}\mathbf{y}^{(k+1)} - \mathbf{D}^{-1}\mathbf{L}^T\mathbf{y}^{(k)} + \mathbf{D}^{-1/2}\mathbf{z}^{(k)}$$

where: $\mathbf{D} = \mathrm{diag}(\mathbf{A})$, $\mathbf{L}$ is the strictly lower triangular part of $\mathbf{A}$, $\mathbf{z}^{(k-1)} \sim \mathrm{N}(\mathbf{0}, \mathbf{I})$

$$\mathbf{y}^{(k+1)} = \mathbf{G}\mathbf{y}^{(k)} + \mathbf{c}^{(k)}$$

$\mathbf{c}^{(k)}$ is iid 'noise' with zero mean, finite covariance

---

Goodman & Sokal 1989

Matrix formulation of Gibbs sampling from $\mathrm{N}(0, \mathbf{A}^{-1})$

Let $\mathbf{y} = (y_1, y_2, ..., y_n)^T$

Component-wise Gibbs updates each component in sequence from the (normal) conditional distributions

One 'sweep' over all n components can be written

$$\mathbf{y}^{(k+1)} = -\mathbf{D}^{-1}\mathbf{L}\mathbf{y}^{(k+1)} - \mathbf{D}^{-1}\mathbf{L}^T\mathbf{y}^{(k)} + \mathbf{D}^{-1/2}\mathbf{z}^{(k)}$$

where: $\mathbf{D} = \mathrm{diag}(\mathbf{A})$, $\mathbf{L}$ is the strictly lower triangular part of $\mathbf{A}$, $\mathbf{z}^{(k-1)} \sim \mathrm{N}(\mathbf{0}, \mathbf{I})$

$$\mathbf{y}^{(k+1)} = \mathbf{G}\mathbf{y}^{(k)} + \mathbf{c}^{(k)}$$

$\mathbf{c}^{(k)}$ is iid 'noise' with zero mean, finite covariance

Spot the similarity to Gauss-Seidel iteration for solving $\mathbf{A}\mathbf{x} = \mathbf{b}$

$$\mathbf{x}^{(k+1)} = -\mathbf{D}^{-1}\mathbf{L}\mathbf{x}^{(k+1)} - \mathbf{D}^{-1}\mathbf{L}^{\mathsf{T}}\mathbf{x}^{(k)} + \mathbf{D}^{-1}\mathbf{b}$$

Goodman & Sokal 1989; Amit & Grenander 1991

Gibbs converges $\iff$ solver converges

Theorem 1 *Let* $\mathbf{A} = \mathbf{M} - \mathbf{N}$, $\mathbf{M}$ *invertible. The stationary linear solver*

$$\begin{aligned}\mathbf{x}^{(k+1)} &= \mathbf{M}^{-1}\mathbf{N}\mathbf{x}^{(k)} + \mathbf{M}^{-1}\mathbf{b} \\ &= \mathbf{G}\mathbf{x}^{(k)} + \mathbf{M}^{-1}\mathbf{b}\end{aligned}$$

converges, if and only if the random iteration

$$\begin{aligned}\mathbf{y}^{(k+1)} &= \mathbf{M}^{-1}\mathbf{N}\mathbf{y}^{(k)} + \mathbf{M}^{-1}\mathbf{c}^{(k)} \\ &= \mathbf{G}\mathbf{y}^{(k)} + \mathbf{M}^{-1}\mathbf{c}^{(k)}\end{aligned}$$

converges in distribution. Here $\mathbf{c}^{(k)} \overset{iid}{\sim} \pi_n$ *has zero mean and finite variance*

Proof. Both converge iff $\varrho(\mathbf{G}) < 1$ $\square$

Convergent splittings generate convergent (generalized) Gibbs samplers

Mean converges with asymptotic convergence factor $\varrho(\mathbf{G})$, covariance with $\varrho(\mathbf{G})^2$

Young 1971 Thm 3-5.1, Duflo 1997 Thm 2.3.18-4, Goodman & Sokal, 1989, Galli & Gao 2001
F Parker 2012

Average contractive iterated random functions

Diaconis Freedman SIAM Review 1999, Stenflo JDEA $\geq$2012

Some not so common Gibbs samplers for $\mathrm{N}(0, \mathbf{A}^{-1})$

splitting/sampler	$\mathbf{M}$	$\mathbf{Var}\left(\mathbf{c}^{(k)}\right) = \mathbf{M}^T + \mathbf{N}$	converge if
Richardson	$\frac{1}{\omega}\mathbf{I}$	$\frac{2}{\omega}\mathbf{I} - \mathbf{A}$	$0 < \omega < \frac{2}{\varrho(\mathbf{A})}$
Jacobi	$\mathbf{D}$	$2\mathbf{D} - \mathbf{A}$	$\mathbf{A}$ SDD
GS/Gibbs	$\mathbf{D} + \mathbf{L}$	$\mathbf{D}$	always
SOR/B&F	$\frac{1}{\omega}\mathbf{D} + \mathbf{L}$	$\frac{2-\omega}{\omega}\mathbf{D}$	$0 < \omega < 2$
SSOR/REGS	$\frac{\omega}{2-\omega}\mathbf{M}_{\mathsf{SOR}}\mathbf{D}^{-1}\mathbf{M}_{\mathsf{SOR}}^{\mathsf{T}}$	$\frac{\omega}{2-\omega}\left(\mathbf{M}_{\mathsf{SOR}}\mathbf{D}^{-1}\mathbf{M}_{\mathsf{SOR}}^{T} + \mathbf{N}_{\mathsf{SOR}}^{T}\mathbf{D}^{-1}\mathbf{N}_{\mathsf{SOR}}\right)$	$0 < \omega < 2$

Good choice has: convenient to solve $\mathbf{Mu} = \mathbf{r}$ and sample from $\mathrm{N}(0, \mathbf{M}^T + \mathbf{N})$

Relaxation parameter ω can accelerate Gibbs

SOR: Adler 1981; Barone & Frigessi 1990, Amit & Grenander 1991, SSOR: Roberts & Sahu 1997

Some not so common Gibbs samplers for $\mathrm{N}(0, \mathbf{A}^{-1})$

splitting/sampler	$\mathbf{M}$	$\mathbf{Var}\left(\mathbf{c}^{(k)}\right) = \mathbf{M}^T + \mathbf{N}$	converge if
Richardson	$\frac{1}{\omega}\mathbf{I}$	$\frac{2}{\omega}\mathbf{I} - \mathbf{A}$	$0 < \omega < \frac{2}{\varrho(\mathbf{A})}$
Jacobi	$\mathbf{D}$	$2\mathbf{D} - \mathbf{A}$	$\mathbf{A}$ SDD
GS/Gibbs	$\mathbf{D} + \mathbf{L}$	$\mathbf{D}$	always
SOR/B&F	$\frac{1}{\omega}\mathbf{D} + \mathbf{L}$	$\frac{2-\omega}{\omega}\mathbf{D}$	$0 < \omega < 2$
SSOR/REGS	$\frac{\omega}{2-\omega}\mathbf{M}_{\mathsf{SOR}}\mathbf{D}^{-1}\mathbf{M}_{\mathsf{SOR}}^{\mathsf{T}}$	$\frac{\omega}{2-\omega}\left(\mathbf{M}_{\mathsf{SOR}}\mathbf{D}^{-1}\mathbf{M}_{\mathsf{SOR}}^{T} + \mathbf{N}_{\mathsf{SOR}}^{T}\mathbf{D}^{-1}\mathbf{N}_{\mathsf{SOR}}\right)$	$0 < \omega < 2$

Good choice has: convenient to solve $\mathbf{Mu} = \mathbf{r}$ and sample from $\mathrm{N}(0, \mathbf{M}^T + \mathbf{N})$

Relaxation parameter ω can accelerate Gibbs

SSOR is a forwards and backwards sweep of SOR to give a *symmetric* splitting

SOR: Adler 1981; Barone & Frigessi 1990, Amit & Grenander 1991, SSOR: Roberts & Sahu 1997

A closer look at convergence

To sample from $\mathrm{N}(\mu, \mathbf{A}^{-1})$ where $\mathbf{A}\mu = \mathbf{b}$

Split $\mathbf{A} = \mathbf{M} - \mathbf{N}$, $\mathbf{M}$ invertible. $\mathbf{G} = \mathbf{M}^{-1}\mathbf{N}$, and $\mathbf{c}^{(k)} \overset{\text{iid}}{\sim} \mathrm{N}(0, \mathbf{M}^T + \mathbf{N})$

The iteration

$$\mathbf{y}^{(k+1)} = \mathbf{G}\mathbf{y}^{(k)} + \mathbf{M}^{-1}\left((\mathbf{c}^{(k)} + \mathbf{b}\right)$$

implies

$$\mathrm{E}\left(\mathbf{y}^{(n)}\right) - \mu = \mathbf{G}^n\left[\mathrm{E}\left(\mathbf{y}^{(0)}\right) - \mu\right]$$

and

$$\mathrm{Var}\left(\mathbf{y}^{(n)}\right) - \mathbf{A}^{-1} = \mathbf{G}^n\left[\mathrm{Var}\left(\mathbf{y}^{(0)}\right) - \mathbf{A}^{-1}\right]\mathbf{G}^n$$

(Hence asymptotic average convergence factors $\varrho(\mathbf{G})$ and $\varrho(\mathbf{G})^2$)

Errors go down as the polynomial

$$P_n(\lambda) = (1 - \lambda)^n$$

A closer look at convergence

To sample from $\mathrm{N}(\mu, \mathbf{A}^{-1})$ where $\mathbf{A}\mu = \mathbf{b}$

Split $\mathbf{A} = \mathbf{M} - \mathbf{N}$, $\mathbf{M}$ invertible. $\mathbf{G} = \mathbf{M}^{-1}\mathbf{N}$, and $\mathbf{c}^{(k)} \overset{\text{iid}}{\sim} \mathrm{N}(0, \mathbf{M}^T + \mathbf{N})$

The iteration

$$\mathbf{y}^{(k+1)} = \mathbf{G}\mathbf{y}^{(k)} + \mathbf{M}^{-1}\left((\mathbf{c}^{(k)} + \mathbf{b}\right)$$

implies

$$\mathrm{E}\left(\mathbf{y}^{(n)}\right) - \mu = \mathbf{G}^n\left[\mathrm{E}\left(\mathbf{y}^{(0)}\right) - \mu\right]$$

and

$$\mathrm{Var}\left(\mathbf{y}^{(n)}\right) - \mathbf{A}^{-1} = \mathbf{G}^n\left[\mathrm{Var}\left(\mathbf{y}^{(0)}\right) - \mathbf{A}^{-1}\right]\mathbf{G}^n$$

(Hence asymptotic average convergence factors $\varrho(\mathbf{G})$ and $\varrho(\mathbf{G})^2$)

Errors go down as the polynomial

$$P_n(\lambda) = (1 - \lambda)^n$$

solver and sampler have exactly the same error polynomial

Controlling the error polynomial

The splitting

$$\mathbf{A} = \frac{1}{\tau}\mathbf{M} + \left(1 - \frac{1}{\tau}\right)\mathbf{M} - \mathbf{N}$$

gives the iteration operator

$$\mathbf{G}_\tau = \left(\mathbf{I} - \tau\mathbf{M}^{-1}\mathbf{A}\right)$$

and error polynomial $Q_n(\lambda) = (1 - \tau\lambda)^n$

The *sequence* of parameters $\tau_1, \tau_2, \ldots, \tau_n$ gives the error polynomial

$$Q_n(\lambda) = \prod_{l=1}^{m} (1 - \tau_l \lambda)$$

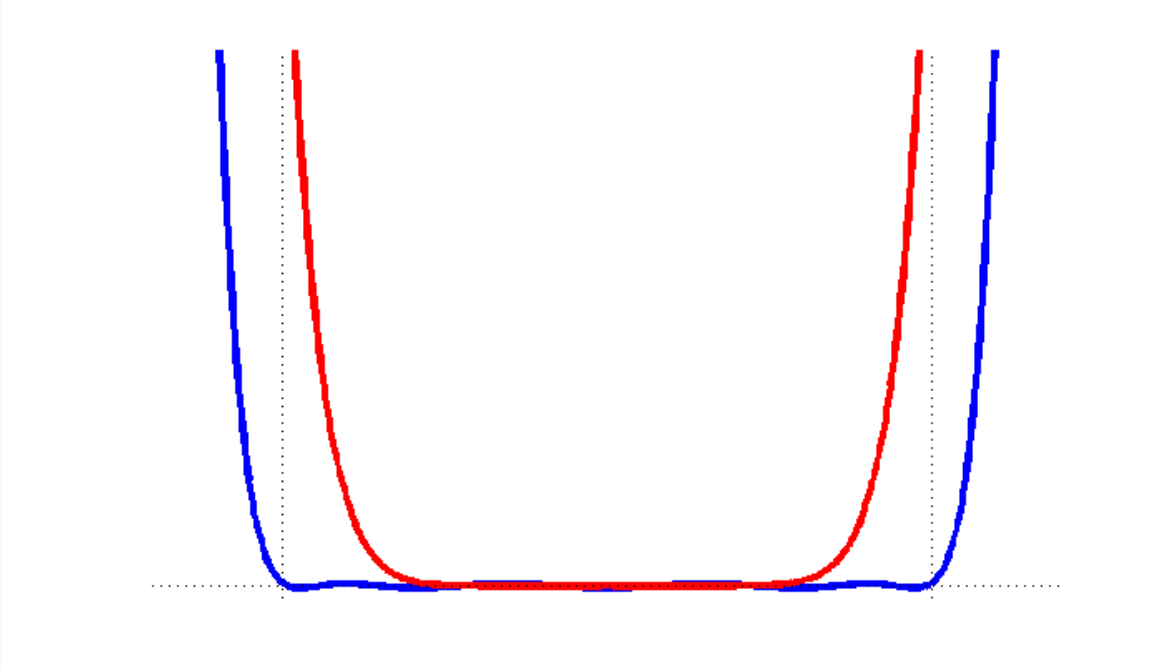
... so we can choose the zeros of Q_n

This gives a non-stationary solver $\equiv$ non-homogeneous Markov chain

---

Golub & Varga 1961, Golub & van Loan 1989, Axelsson 1996, Saad 2003, F & Parker 2012

The best (Chebyshev) polynomial



10 iterations, factor of 300 improvement

Choose

$$\frac{1}{\tau_l} = \frac{\lambda_n + \lambda_1}{2} + \frac{\lambda_n - \lambda_1}{2} \cos\left(\pi \frac{2l+1}{2p}\right) \qquad l = 0, 1, 2, \ldots, p-1$$

where λ_1 λ_n are extreme eigenvalues of $\mathbf{M}^{-1}\mathbf{A}$

Second-order accelerated sampler

First-order accelerated iteration turns out to be unstable

Numerical stability, and optimality at each step, is given by the second-order iteration

$$\mathbf{y}^{(k+1)} = (1-\alpha_k)\mathbf{y}^{(k-1)} + \alpha_k \mathbf{y}^{(k)} + \alpha_k \tau_k \mathbf{M}^{-1}(\mathbf{c}^{(k)} - \mathbf{A}\mathbf{y}^{(k)})$$

with α_k and τ_k chosen so error polynomial satisfies Chebyshev recursion.

Theorem 2 *2^{nd}-order solver converges $\Rightarrow$ 2^{nd}-order sampler converges (given correct noise distribution)*
Error polynomial is optimal, at each step, for both mean and covariance

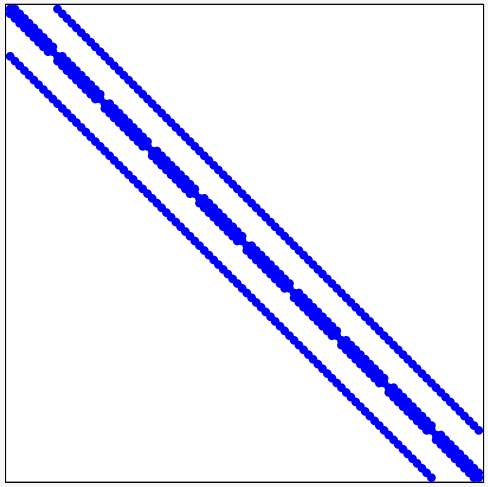
Asymptotic average reduction factor (Axelsson 1996) is

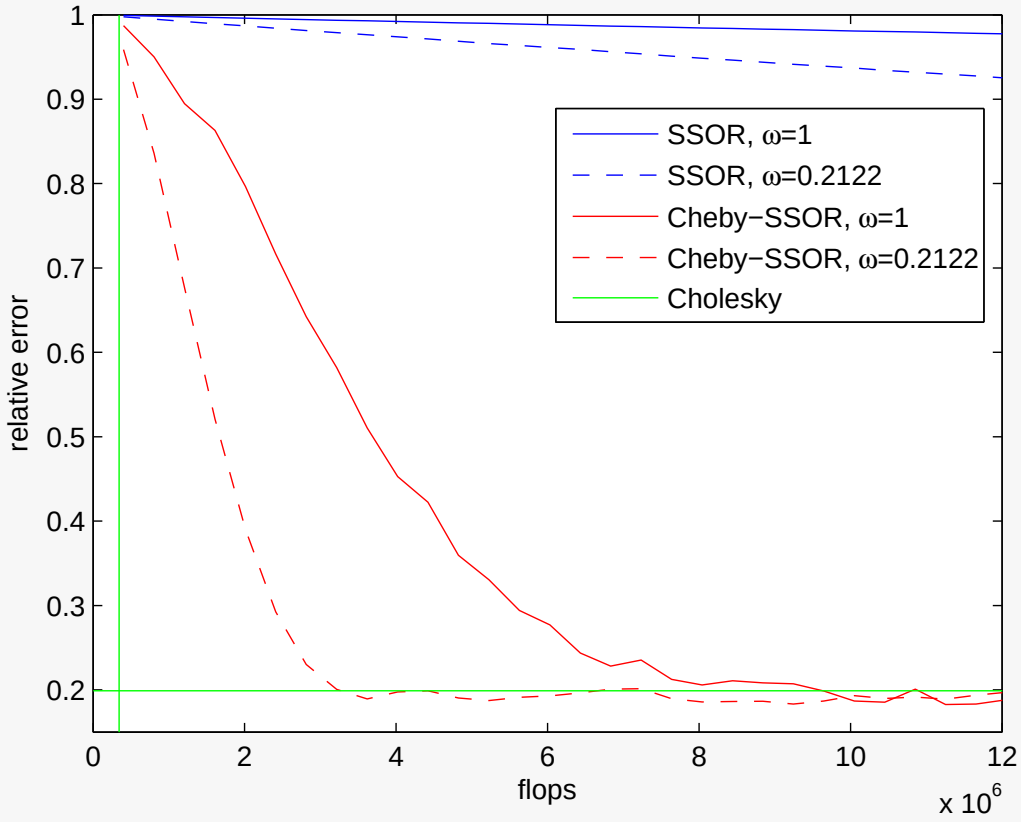
$$\sigma = \frac{1 - \sqrt{\lambda_1 / \lambda_n}}{1 + \sqrt{\lambda_1 / \lambda_n}}$$

Axelsson 1996, F & Parker 2012

10×10 lattice ($d = 100$) sparse precision matrix

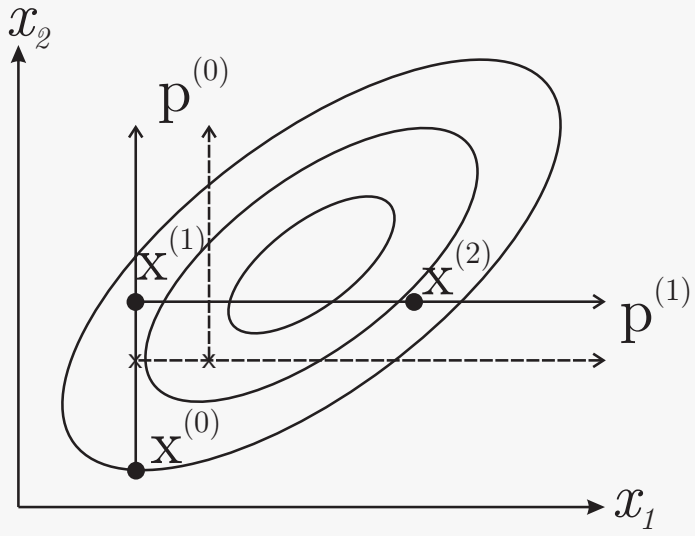
$$[\mathbf{A}]_{ij} = 10^{-4}\delta_{ij} + \begin{cases} n_i & \text{if} \quad i = j \\ -1 & \text{if} \quad i \neq j \quad \text{and} \quad ||s_i - s_j||_2 \leq 1 \\ 0 & \text{otherwise} \end{cases} .$$



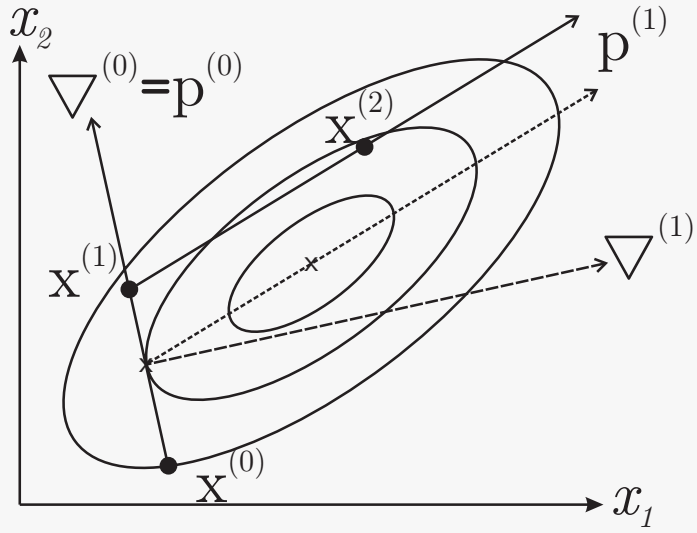


$\approx 10^4$ times faster

CG (Krylov space) sampling



Gibbs sampling & Gauss Seidel



CG sampling & optimization

Mutually conjugate vectors (wrt $\mathbf{A}$) are independent directions

$$\mathbf{V}^{\mathsf{T}}\mathbf{A}\mathbf{V} = \mathbf{D} \Rightarrow \mathbf{A}^{-1} = \mathbf{V}\mathbf{D}^{-1}\mathbf{V}^{\mathsf{T}}$$

so if $z \sim \mathbf{N}(0, I)$ then $x = \mathbf{V}\sqrt{\mathbf{D}^{-1}}z \sim \mathbf{N}(0, \mathbf{A}^{-1})$

Schneider & Willsky 2003, F 2008, Parker & F SISC 2012

CG sampling algorithm

ALGORITHM 1 (CG sampler from $N(0, A^{-1})$). *Given b and x^0, let $r^0 = b^0 - Ax^0$, $p^0 = r^0$, $d_0 = p^{(0)T} A p^0$, $y^0 = x^0$, and $k := 1$. Specify some stopping tolerance ϵ. Iterate:*

1. $\gamma_{k-1} = \frac{r^{(k-1)T} r^{k-1}}{d_{k-1}}$ *is the 1-D minimzer of ϕ in the direction* $x^{k-1} + \gamma p^{k-1}$
2. $x^k = x^{k-1} + \gamma_{k-1} p^{k-1}$
3. *Sample* $z \sim N(0,1)$, *and set* $y^k = y^{k-1} + \frac{z}{\sqrt{d_{k-1}}} p^{k-1}$
4. $r^k = -\nabla_x \phi(x^k) = r^{k-1} - \gamma_{k-1} A p^{k-1}$ *is the residual*
5. $\beta_k = -\frac{r^{kT} r^k}{r^{(k-1)T} r^{k-1}}$
6. $p^k = r^k - \beta_k p^{k-1}$ *is the next conjugate search direction.*
7. $d_k = p^{(k)T} A p^k$
8. *Quit if* $||r^k||_2 < \epsilon$. *Else set* $k := k + 1$ *and go to step 1.*

- Apart from step 3, this is exactly (linear) CG optimization
- $\mathrm{Var}(y^k)$ is the CG polynomial

CG sampling example

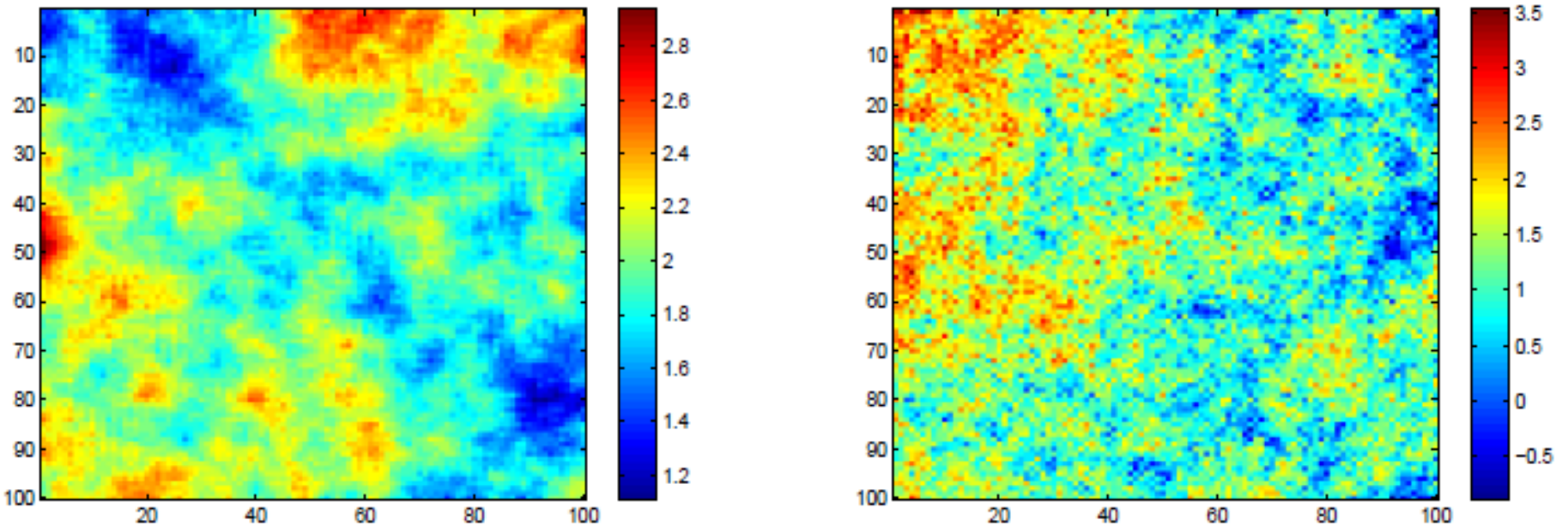


FIG. 4.2. *In the left panel is a CG sample* $y^k \dot{\sim} N(0, A^{-1})$ *from a* 10^4*-dimensional Gaussian over a 2D domain with a second order locally linear precision matrix. The realized variance* $\mathrm{Var}(y^k)$ *accounts for* 80% *of the variability in* A^{-1}*. A Cholesky sample is shown in the right panel.*

So CG (by itself) over-smooths : initialize Gibbs with CG

Parker & F SISC 2012, Schneider & Willsky 2003

Evaluating $f(\mathbf{A}) = \mathbf{A}^{-1/2}$

If $z \sim N(0, I)$ then $x = \mathbf{A}^{-1/2} z \sim N(0, A^{-1})$

Compute a rational approximation

$$x = \mathbf{A}^{-1/2} z \approx \sum_{j=1} N \alpha_j \left(\mathbf{A} - \sigma_j I\right)^{-1} x$$

on 'spectrum' $\{\sigma_j\}$ of $\mathbf{A}$, based on

- Lanczos approximation: $x = x_0 + \beta_0 \mathbf{V} \mathbf{T}^{1/2} \mathbf{e}_1$
- Contour integration
- Continuous deformation ODE

all using CG

Simpson Turner & Pettitt 2008, Aune Eidsvik & Pokern STCO 2012

Sampling by BFGS

BFGS optimization over $\log \pi$ builds an approximation to the inverse Hessian

$$\mathbf{A}^{-1} \approx \mathbf{A}_0^{-1} + \sum_{i=1}^{N} \rho_i s_i s_i^{\mathsf{T}}$$

so if $z_0 \sim N(0, \mathbf{A}_0^{-1})$ and $z_i \sim N(0,1)$ then

$$z_0 + \sum_{i=1}^{N} \sqrt{\rho_i} s_i z_i \sim N(0, \mathbf{A}^{-1})$$

Can propagate sample covariance in large-scale EnKF ...

though CG sampler does a better job (don't know why)

and seeding proposal in AM

Veersé 1999, Auvinen Bardsley Haario & Kauranne 2010, Bardsley Solonen Parker Haario & Howard IJUQ 2011

Non-negativity constrained sampling

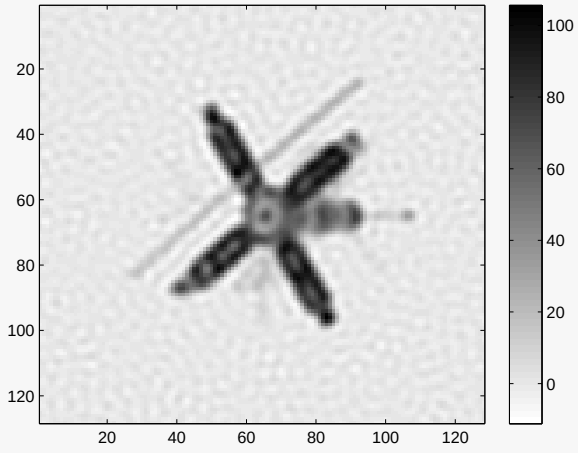
Linear inverse problem $\mathbf{b} = \mathbf{A}\mathbf{x} + \boldsymbol{\eta} \qquad \boldsymbol{\eta} \sim \mathbf{N}(0, \sigma^2 I)$

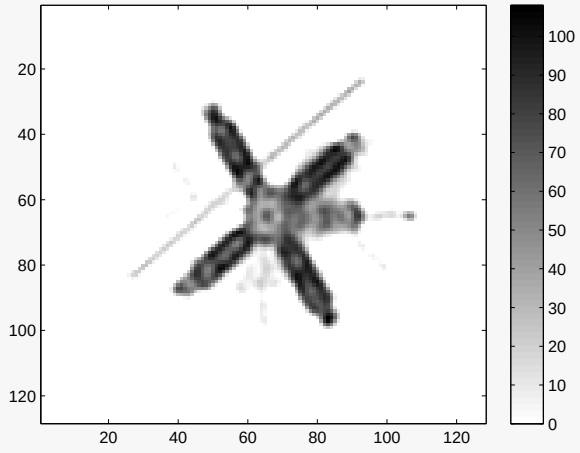
$$p(\mathbf{x}, \lambda, \delta | \mathbf{b}) \propto \lambda^{n/2+\alpha_\lambda - 1} \delta^{n/2+\alpha_\delta - 1} \exp\left(-\frac{\lambda}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 - \frac{\delta}{2} \mathbf{x}^T \mathbf{C} \mathbf{x} - \beta_\lambda \lambda - \beta_\delta \delta \right)$$

with probability mass projected to constraint boundary in metric of Hessian of $\log \pi$

$$\mathbf{x}^k = \arg\min_{\mathbf{x} \geq \mathbf{0}} \left\{ \frac{1}{2} \mathbf{x}^T (\lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{C}) \mathbf{x} - \mathbf{x}^T (\lambda_k \mathbf{A}^T \mathbf{b} + \mathbf{w}) \right\}$$

where $\mathbf{w} \sim N(\mathbf{0}, \lambda_k \mathbf{A}^T \mathbf{A} + \delta_k \mathbf{C})$. Compute by projected-gradient CG.





Some observations

In the Gaussian setting

- GS $\equiv$ GS
- *stochastic relaxation* is fundamentally equivalent to classical *relaxation*
- existing fast solvers make fast samplers (mutligrid, fast multipole, parallel tools)

More generally

- acceleration of convergence in mean and covariance not limited to Gaussian targets
- ... can be applied to inverse problems
- ... but no results for rates of convergence
- A sequence of non-optimal kernels can do *much* better than repeated application of the 'optimal' kernel

Summary