A Linearly Convergent Double Stochastic Gauss-Seidel Algorithm for Linear Systems

Zhi-Quan Luo

Shenzhen Research Institute of Big Data
The Chinese University of Hong Kong, Shenzhen, China

Joint work with Mingyi Hong, Meisam Razaviyayn, Navid Reyhanian

2018 MOA, Beijing

Outline

Review of Gauss-Seidel Algorithm for Linear System

- Consider solving a $n \times n$ linear system $Ax = b$.
- The classical G-S algorithm uses one equation to update one variable at a time:

$$x_i^+ = \frac{b_i - \sum_{j \neq i} a_{ij} x_j}{a_{ii}}. \tag{1}$$

Using a step-size $\alpha > 0$, this strategy leads to the following successive over-relaxed (SOR) update rule:

$$x_i^{r+1} = (1 - \alpha) x_i^r + \alpha \frac{b_i - \sum_{j \neq i} a_{ij} x_j^r}{a_{ii}}, \tag{2}$$

where a_{ij} is the (i, j)-th element of A; b_i is the i-th element of b.

An Illustrative Example

Consider the following example.

$$A = \begin{bmatrix} 1 & -\tau \\ -\tau & 1 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

where $\tau > 1$ is some given constant.

$$x_1^{r+1} = (1-\alpha)x_1^r + \alpha\tau x_2^r, \quad \text{when } x_1 \text{ is updated before } x_2 \tag{3}$$

$$x_2^{r+1} = (1-\alpha)x_2^r + \alpha\tau x_1^r, \quad \text{when } x_2 \text{ is updated before } x_1. \tag{4}$$

Assume $x_1^0 = x_2^0 > 0$.

Variants of the G-S Algorithms

Let us consider the following five different update rules which are all of the G-S type.

1. **Cyclic Successive Over Relaxation (SOR):** At iteration $r+1$, we perform

$$x_1^{r+1} = (1-\alpha)x_1^r + \alpha\tau x_2^r, \quad x_2^{r+1} = (1-\alpha)x_2^r + \alpha\tau x_1^{r+1}. \tag{5}$$

2. **Symmetric SOR:** At iteration $r+1$, the variables are updated using a forward-sweep G-S step followed by a back-sweep G-S step:

$$\begin{aligned} x_1^{r+1/2} &= (1-\alpha)x_1^r + \alpha\tau x_2^r, \quad x_2^{r+1/2} = (1-\alpha)x_2^r + \alpha\tau x_1^{r+1/2}. \\ x_2^{r+1} &= (1-\alpha)x_2^{r+1/2} + \alpha\tau x_1^{r+1/2}, \quad x_1^{r+1} = (1-\alpha)x_1^{r+1/2} + \alpha\tau x_2^{r+1}. \end{aligned}$$

3. **Uniformly Randomized (UR) SOR:** At iteration $r+1$, uniformly randomly pick one variable from x_1 and x_2. Update according to (3) or (4) based on which variables are selected, while fixing the remaining variable at its previous value.

Variants of the G-S Algorithms

4. **Non-Uniformly Randomized (NUR) SOR:** At iteration $r+1$, let $p_1^{r+1} > 0$ and $p_2^{r+1} > 0$ satisfy $p_1^{r+1} + p_2^{r+1} = 1$; randomly pick x_i according to p_i^{r+1}. Update according to (3) or (4) based on which variables are selected, while fixing the remaining variable at its previous value.
5. **Random Permutation (RP) SOR:** At iteration $r+1$, randomly select a permutation π of the index set $\{1, 2\}$; The variables are updated according to

$$x_{\pi(1)}^{r+1} = (1-\alpha)x_{\pi(1)}^r + \alpha\tau x_{\pi(2)}^r, \quad x_{\pi(2)}^{r+1} = (1-\alpha)x_{\pi(2)}^r + \alpha\tau x_{\pi(1)}^{r+1}. \tag{6}$$

This method is referred to as the *shuffled* SOR in [Oswald15b].

It is easily seen that for any update order listed above, the resulting algorithm have the following property:

$$\min\{x_1^r, x_2^r\} > \min\{x_1^0, x_2^0\} > 0, \quad \forall r,\ \forall \alpha > 0.$$

On the other hand, the solution of the system of linear equation is $x_1^* = x_2^* = 0$; hence none of these algorithms will find the solution. ■

Convergence of the G-S Algorithms

- For $0 < \alpha < 2$, the G-S algorithm converges linearly for the special cases
  - A is symmetric and positive definite (coordinate descent for the potential function $\frac{1}{2}x^T Ax + b^T x$)
  - A is diagonally dominant (the distance function $\|x - x^*\|^2$ is contracting)

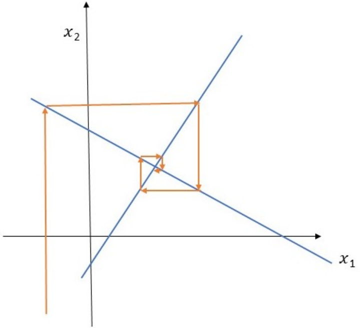


Figure: Gauss-Seidel Algorithm in 2-D

Convergence of the G-S Algorithms

For general A (possibly non-square), the G-S algorithm may diverge

- lack of a potential function when A is asymmetric
- the distance to x^* may diverge without diagonal dominance

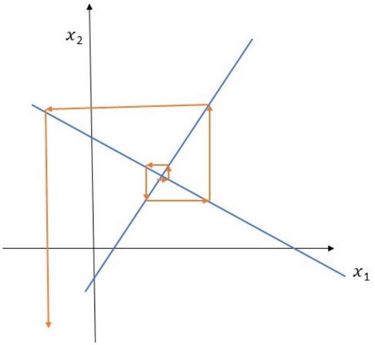
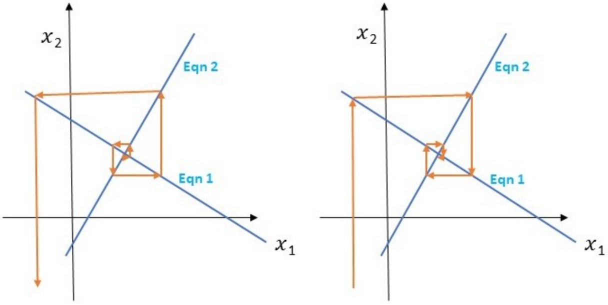


Figure: Divergence of G-S algorithm in 2-D

Fixing G-S algorithm for a general linear system $Ax = b$?

How to design a variable-equation association and the updating order to ensure G-S convergence?

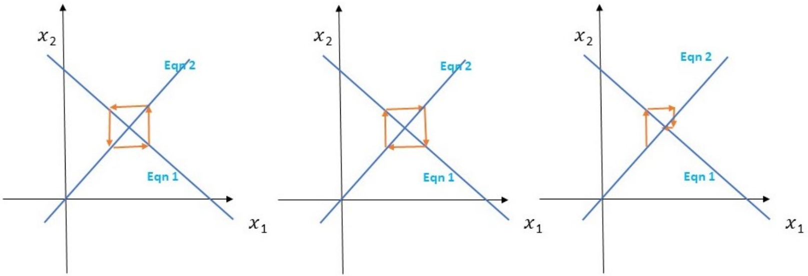


Reversing the variable-equation updating order can induce convergence

In the G-S schemes mentioned earlier, the indices of the variables and equations follow a fixed (ad hoc) association. That is, if variable i is chosen, then equation i has to be chosen as well.

Fixing the G-S algorithm for a general linear system $Ax = b$?

Will also need to adjust the stepsize α to ensure G-S convergence!



Stepsize control is necessary to ensure convergence

Is Linear Convergence Possible?

For general $m \times n$ linear system $Ax = b$, consider the linear squared error residual

$$f(x) = \sum_{i=1}^{m} \frac{1}{2} |a_i^T x - b_i|^2 \tag{7}$$

Then the G-S algorithm can be viewed as the coordinate-wise incremental minimization algorithm where

- each error term $\frac{1}{2}|a_i^T x - b_i|^2$ is associated with one variable x_i
- each iteration minimizes $\frac{1}{2}|a_i^T x - b_i|^2$ (i.e., setting $a_i^T x - b_i = 0$) by adjusting variable x_i

It is well-known that

- the *incremental gradient descent* algorithm with diminishing stepsizes converges to an minimizer of the LS function
- but the convergence is not linear
- for constant stepsize, convergence to the neighborhood of x^* only (may get away from x^*)

Other Related Work

The existing literature has

- Random Reordering in SOR-Type Methods [Oswald-Zhou16]
- Randomized Block Kaczmarz Method (RK) [Strohmer-Vershynin09]
- Semi-Stochastic Gradient Descent Methods [Konecny-Richtarik15]
- SVRG [Johnson-Zhang13], Semi-Stochastic Coordinate Descent [Konecny-Qu-Richtarik15]

Linear convergence requires

- full gradient at each iteration, or
- at each epoch (SVRG, S2CD)

Outline

Simultaneously Selecting Equations and Variables

Will *unlocking* the paring between the variables and the equations ensure convergence of G-S type algorithm for arbitrary A?

- select a pair (i, j) at each iteration where i is an index for an equation while j is an index for the variable to be updated
- after picking the pair (i, j), one can update the variable j by

$$x_j^{r+1} = (1-\alpha)x_j^r + \alpha \frac{b_i - \sum_{k \neq j} a_{ik} x_k^r}{a_{ij}}. \tag{8}$$

Divergence!

Consider the same (A, b) as given in Example 1.

$$A = \begin{bmatrix} 1 & -\tau \\ -\tau & 1 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

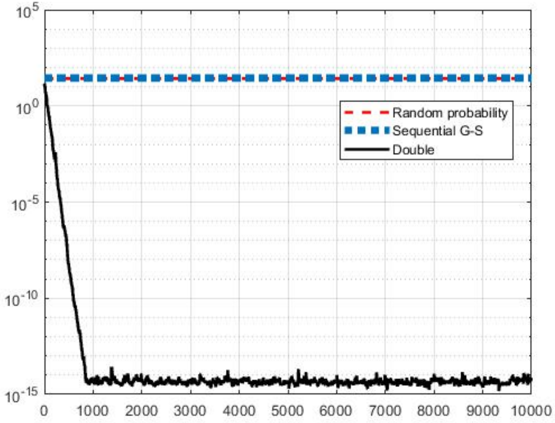
After selecting the pair (i, j), we can use one of the following four update rules:

1. **Case 1.** $i = 1, j = 1 \to x_1^{r+1} = (1-\alpha)x_1^r + \alpha\tau x_2^r$;
2. **Case 2.** $i = 1, j = 2 \to x_2^{r+1} = (1-\alpha)x_2^r + \frac{\alpha}{\tau} x_1^r$;
3. **Case 3.** $i = 2, j = 1 \to x_1^{r+1} = (1-\alpha)x_1^r + \frac{\alpha}{\tau} x_2^r$;
4. **Case 4.** $i = 1, j = 2 \to x_2^{r+1} = (1-\alpha)x_2^r + \alpha\tau x_1^r$.

- However, we can show that for any fixed updating order, the resulting algorithm will diverge for almost all initialization, and for any $0 < \alpha < 1$.
- The uniform randomized update rule, and the randomly permuted rule will also diverge for almost all initialization and for any $0 < \alpha < 1$.

Numerical Behaviors

For a randomly generated linear system



A Double Stochastic Algorithm

Algorithm 1. The double stochastic G-S (DSGS) algorithm.

At iteration 0, randomly generate x^0.
At iteration $r+1$, randomly pick the index pair (i,j) with probability

$$p_{ij} = \frac{a_{ij}^2}{\sum_{ij} a_{ij}^2}.$$

Update x_j by the following:

$$x_j^{r+1} = (1-\alpha)x_j + \alpha\left(\frac{b_i - \sum_{k\neq j} a_{ik}x_k}{a_{ij}}\right) = x_j + \alpha\left(\frac{b_i - \sum_{k=1} a_{ik}x_k}{a_{ij}}\right). \quad (9)$$

Linear Convergence: A is full column rank

Let us define

$$\Delta := Ax - b \in \mathbb{R}^m, \quad \text{and} \quad \Delta^+ = Ax^+ - b. \tag{10}$$

We have the following result.

Claim

Suppose that A has full column rank, and $a_{ii} \neq 0$ for all $i \in [n]$. Let us choose $\alpha = 1/n$. Then the double stochastic G-S algorithm achieves the following convergence rate

$$\mathbb{E}[\|\Delta^+\|^2 \mid x] \leq \left(1 - \frac{1}{n\kappa^2(A)}\right)\|\Delta\|^2. \tag{11}$$

Remarks

1. The rate of the randomized CD method [Leventhal-Lewis10] with A being symmetric and PD) is given by

$$\mathbb{E}[\|\Delta^+\|^2] \le \left(1 - \frac{1}{\|A^{-1}\|_2 \mathrm{Tr}[A]}\right)\|\Delta\|^2 \le \left(1 - \frac{1}{\sqrt{n}\kappa(A)}\right)\|\Delta\|^2. \tag{12}$$

Our rate is proportional to $\left(1 - \left(\frac{1}{\sqrt{n}\kappa(A)}\right)^2\right)$, which is worse. This is reasonable due to the lack of symmetry or positive definiteness of A.

2. The rate of Randomized Kacmartz (RK) method [Strohmer-Vershynin08] with A being full column rank is given by

$$\mathbb{E}[\|\Delta^+\|^2 \mid x] \le \left(1 - \frac{2\alpha - \alpha^2}{\kappa^2(A)}\right)\|\Delta\|^2 \overset{\alpha=1}{=} \left(1 - \frac{1}{\kappa^2(A)}\right)\|\Delta\|^2. \tag{13}$$

Note that at each iteration of RK, n variables are updated instead of just one in the doubly stochastic method. When n is large and $\kappa(A)$ is large, the two rates are comparable since

$$\left(1 - \frac{1}{n\kappa^2(A)}\right)^n \approx \exp(-1/\kappa^2(A)) \approx 1 - \frac{1}{\kappa^2(A)}$$

Linear Convergence: general case

The following claims shows that the quantity $\|\Delta\|^2$ converges linearly to zero in expectation.

Claim

Consider a consistent system $Ax = b$ with arbitrary A. Let us pick

$$\alpha = \frac{1}{\|A\|_F^2}\underline{\lambda}_{\min}(AA^T).$$

Then the double stochastic G-S algorithm achieves the following convergence rate

$$\mathbb{E}[\|\Delta^+\|^2 \mid x] \leq \|\Delta\|^2 \left(1 - \left(\frac{1}{\|A\|_F^2}\underline{\lambda}_{\min}(AA^T)\right)^2\right) \tag{14}$$

where $\underline{\lambda}_{\min}(AA^T)$ denotes the smallest positive eigenvalue of AA^T.

Outline

Linear Inequality System

- Consider the problem of finding a point in the intersection of multiple polyhedral sets.
- Unlike the classical alternating projection algorithm, we perform the projection in a "coordinate descent" manner.

Specifically, we consider the following problem:

$$\text{Find } x \qquad \text{s.t.} \quad A_{i:}x \leq b_i, \quad i = 1, \cdots, m. \tag{15}$$

We will assume in the rest of this section that the system $Ax \leq b$ is feasible.

A Double Stochastic Alternating Projection Algorithm

The proposed algorithm is closely related to Algorithm 1, except that we only update those equations that violate the constraint.

Algorithm 2. The double stochastic alternating projection algorithm.

At iteration 0, randomly generate x^0.
At iteration $r+1$, randomly pick the index pair (i,j) with probability

$$p_{ij} = \frac{a_{ij}^2}{\sum_{ij} a_{ij}^2}.$$

Update x_j by the following

$$x_j^{r+1} = \begin{cases} x_j^r, & \text{if } A_{i:}x^r \le b_i \\ x_j + \alpha\left(\frac{b_i - \sum_{k=1} a_{ik}x_k}{a_{ij}}\right), & \text{otherwise.} \end{cases} \tag{16}$$

Convergence Result

Let us define the following function

$$f(x) := \sum_{i=1}^{m} f_i(x) = \frac{1}{2}\sum_{i=1}^{m}(a_i^T x - b_i)_+^2. \tag{17}$$

We note that any feasible solution of (15) will imply $f(x) = 0$. Further, each function f_i is differentiable, and its gradient is

$$\nabla f_i(x) = \begin{cases} A_{i:}^T(A_{i:}x - b_i) & \text{if } A_{i:}x - b_i \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Full Column Rank Case

Suppose A is full column rank.

$$\lambda_{\min}(AA^T) > 0. \tag{18}$$

Claim

Suppose A has full row rank, and α is chosen as

$$\alpha < \frac{\lambda_{\min}(AA^T)}{\|A\|_F}. \tag{19}$$

Then we have

$$\mathbb{E}[f(x^+) \mid x] \leq \left(1 - \left(\frac{\lambda_{\min}(A^TA)}{2\|A\|_F^2}\right)^2\right) f(x) \tag{20}$$

This implies global linear convergence of the feasibility error residual in expectation.

General Rank Case: Hoffman's Error Bound

W need to use the well-known Hoffman's error bound.

Theorem

Let $\mathcal{S}$ denote the solution set for the linear system in the constraint (15). *Then there exists a constant $\tau > 0$ independent of b, with the following property*

$$x \in \mathbb{R}^n,\ S \neq \emptyset \quad \Longrightarrow \quad \textit{dist}(x, S) \leq \tau \|(Ax - b)_+\|. \tag{21}$$

where we have defined

$$(Ax - b)_+ = \max\{0, Ax - b\}, \quad \textit{dist}(x, S) := \inf_{y \in S} \|x - y\|. \tag{22}$$

Convergence Result: General Rank Case

Assume that the system (15) is feasible, and let S denote its solution set and let $x^* \in S$. Clearly we have $f(x^*) := \sum_{i=1}^{m} f_i(x^*) = 0$. We have the following claim.

Claim

Consider a consistent system $Ax \le b$ with arbitrary A. Let us pick

$$\alpha = \frac{1}{n}.$$

Then Algorithm 2 achieves the following convergence rate

$$\mathbb{E}[\mathit{dist}^2(x^+, S) \mid x] \le \left(1 - \frac{1}{n\tau^2 \sum_{ij} a_{ij}^2}\right) \mathit{dist}^2(x, S). \tag{23}$$

Again, global linear convergence!

A Coordinate Gradient Descent Method for Linear Feasibility

Let us consider the following algorithm:

Algorithm 3. A double stochastic coordinate descent algorithm.

At iteration 0, randomly generate x^0.
At iteration $r+1$, randomly pick the index pair (i,j) with probability $p_{ij} = p = \frac{1}{mn}$.
Update x_j by the following

$$x_j^{r+1} = \begin{cases} x_j^r, & \text{if } a_i^T x^r \le b_i \\ x_j + \alpha a_{ij}\left(b_i - \sum_{k=1} a_{ik} x_k\right), & \text{otherwise.} \end{cases} \tag{24}$$

A Coordinate Gradient Descent Method for Nonlinear Feasibility

Let us consider the convex feasibility problem:

Find $x \in S := \{x \mid g_i(x) \le 0, \quad i = 1, \dots, m\}, \quad g_i$ is convex and differentiable

or equivalently

$$\min_x f(x) := \sum_{i=1}^{m} (g_i(x))_+^2 = \sum_{i=1}^{m} f_i(x), \quad \text{where } f_i(x) := \max\{0, g_i(x)\}.$$

Then f_i remains convex and differentiable. Consider the following algorithm:

Algorithm 4. A double stochastic coordinate descent algorithm.

At iteration 0, randomly generate x^0.
At iteration $r+1$, randomly pick the index pair (i,j) with probability $p_{ij} = p = \frac{1}{mn}$.
Update x_j by the following

$$x_j^{r+1} = \begin{cases} x_j^r, & \text{if } g_i(x^r) \le 0 \\ x_j + \alpha \nabla_j g_i(x^r) g_i(x^r), & \text{otherwise.} \end{cases} \tag{25}$$

Error Bound

Assume the following global error bound holds: there exists some $\tau > 0$ such that

$$\mathsf{dist}(x, S) \leq \tau \| \max_{i=1,\ldots,m} (g_i(x))_+ \|, \quad \forall x \in \Re^n. \tag{26}$$

The the error bound (26) is known to hold

- linear case (Hoffman'52) and
- convex quadratic systems satisfying Slater condition (Luo-Luo'94).

Convergence Result

Moreover, S has a nonempty interior.

Claim

Consider a consistent system $S = \{x \mid g_i(x) \leq 0,\ i = 1, ..., m\}$ with an nonempty interior, where each g_i is convex and continuously differentiable. Suppose the error bound holds. Then for sufficiently small α, Then Algorithm 4 achieves a linear rate of convergence

$$\mathbb{E}[\textit{dist}^2(x^+, S) \mid x] \leq \rho\,\textit{dist}^2(x, S), \quad \textit{for some } \rho > 0. \tag{27}$$

Outline

Proof Sketch

Recall that from the update rule we have

$$x^+ = x + \alpha\left(\frac{b_i - \sum_{k=1} a_{ik}x_k}{a_{ij}}\right)e_j.$$

Let us define $x^*(x) := \arg\min_{y\in S}\|x-y\|$. We have the following

$$\begin{aligned}\mathbb{E}[\text{dist}^2(x^+,S)\mid x] &= \mathbb{E}[\|x^+ - x^*(x^+)\|^2 \mid x] \le \mathbb{E}[\|x^+ - x^*(x)\|^2 \mid x] \\ &= \underbrace{\mathbb{E}[\|x^+ - x\|^2 \mid x]}_{\text{term 1}} + \underbrace{\mathbb{E}\left[\langle x^+ - x, x - x^*(x)\rangle \mid x\right]}_{\text{term 2}} + \|x - x^*(x)\|^2.\end{aligned} \tag{28}$$

Proof Sketch: bounding term 1

Let us bound the above equality term by term. First we have

$$
\begin{aligned}
\mathbb{E}[\|x^+ - x\|^2 \mid x] &= \sum_{(i,j):i\in\mathcal{I}} p_{ij}\alpha^2 \left\| \left(\frac{b_i - \sum_{k=1} a_{ik}x_k}{a_{ij}} \right) e_j \right\|^2 \\
&= \sum_{(i,j):i\in\mathcal{I}} \frac{1}{\sum_{i,j} a_{ij}} \alpha^2 \left\| a_{ij} \left(b_i - \sum_{k=1} a_{ik}x_k \right) e_j \right\|^2 \\
&= \sum_{i:i\in\mathcal{I}} \frac{1}{\sum_{i,j} a_{ij}} \alpha^2 n \left\| a_{ij} \left(b_i - \sum_{k=1} a_{ik}x_k \right) \right\|^2 \\
&= \frac{\alpha^2 n}{\sum_{i,j} a_{ij}} \sum_{(i):i\in\mathcal{I}} \|A_{\mathcal{I}}x - b_{\mathcal{I}}\|^2 .
\end{aligned}
$$

where $\mathcal{I}$ denotes the set of violated constraints at current iteration.

Proof Sketch: bounding term 2

The second term in (29) is given by

$$
\begin{aligned}
\mathbb{E}\left[\langle x^+ - x, x - x^*(x)\rangle \mid x\right] &= 2\alpha \sum_{(i,j):i\in\mathcal{I}} \left\langle p_{ij}\alpha\left(\frac{b_i - \sum_{k=1} a_{ik}x_k}{a_{ij}}\right) e_j, x - x^*(x)\right\rangle \\
&= 2\alpha\frac{1}{\sum_{ij} a_{ij}} \sum_{(i,j):i\in\mathcal{I}} \left\langle \alpha\left(a_{ij}(b_i - \sum_{k=1} a_{ik}x_k)\right) e_j, x - x^*(x)\right\rangle \\
&= 2\alpha\frac{1}{\sum_{ij} a_{ij}} \left\langle A_{\mathcal{I}}^T(b_{\mathcal{I}} - A_{\mathcal{I}}x), x - x^*(x)\right\rangle \\
&= -2\alpha\frac{1}{\sum_{ij} a_{ij}} \langle \nabla f(x), x - x^*(x)\rangle \\
&\le -2\alpha\frac{1}{\sum_{ij} a_{ij}} (f(x) - f(x^*(x))) \\
&= -2\alpha\frac{1}{\sum_{ij} a_{ij}} \|A_{\mathcal{I}}x - b_{\mathcal{I}}\|^2.
\end{aligned}
$$

where the first step is because $x - x^*(x)$ is deterministic when conditioned on x; the first inequality is due to the convexity of f; and the last step is because $f(x^*(x)) = 0$.

Proof Sketch

Therefore, overall we have

$$\mathbb{E}[\text{dist}^2(x^+, S)] \leq \frac{n\alpha^2 - 2\alpha}{\sum_{ij} a_{ij}^2} \|A_{\mathcal{I}}x - b_{\mathcal{I}}\|^2 + \|x - x^*(x)\|^2.$$

Therefore if $\alpha \leq 2/n$, we can apply the Hoffman error bound (21)

$$\begin{aligned}
\mathbb{E}[\text{dist}^2(x^+, S)] &\leq \frac{n\alpha^2 - 2\alpha}{\tau^2 \sum_{ij} a_{ij}^2} \text{dist}^2(x, S) + \|x - x^*(x)\|^2 \\
&= \left(1 - \frac{n\alpha^2 - 2\alpha}{\tau^2 \sum_{ij} a_{ij}^2}\right) \text{dist}^2(x, S) \\
&\overset{\alpha = \frac{1}{n}}{\leq} \left(1 - \frac{1}{n\tau^2 \sum_{ij} a_{ij}^2}\right) \text{dist}^2(x, S).
\end{aligned} \tag{29}$$

Therefore we conclude that the algorithm converges linearly in expectation.

Concluding Remarks

We considered G-S algorithms for linear systems. Many big data applications involve a minimization problem of the form

$$\min_{x\in\mathbb{R}^n} f(x) = \sum_{i=1}^{m} f_i(x)$$

- When m is large and n moderate, we use incremental/stochastic gradient descent
  - needs full gradient to ensure linear convergence
- When m is moderate and n is large, we use block coordinate descent/minimization
  - needs partial gradient of f to ensure linear convergence
- When m and n are both large, we need to use incremental/stochastic block coordinate descent/minimization method
  - needs proper randomization to ensure linear convergence

Thank You!