

NON-LINEAR ALGORITHMS FOR PARAMETRIC MARKOV PROGRAMMING

Boris S. Verkhovsky, Professor

Computer Science Department, New Jersey Institute of Technology,
Newark, NJ 07102, USA

Yuriy S. Polyakov, Researcher

USPolyResearch, USA

Abstract. *General idea of nonlinear correcting factors that was successfully applied to accelerate and derive algorithms for many linear and nonlinear problems is used to develop several efficient procedures for solving linear systems of equations based on the Jacobi algorithm. Results of the theoretical study and extensive computer experiments for stochastic matrices are presented and analyzed to determine the conditions under which the studied algorithms converge, and the areas of their maximum convergence rate are identified. Applications of the algorithms to Markov decision processes are discussed.*

Many problems in computer science and operations research can be formulated as Markov Decision Processes (MDP) [1]. Examples include routing, optimal stopping, target, replacement, maintenance and repair, and inventory problems as well as optimal control of queues and stochastic scheduling. The utility function of total expected discounted rewards is commonly used in MDPs with finite and action spaces [1]. In this case, the optimality equation takes the following form:

$$V = \max_{\alpha} [R(\alpha) + \omega S(\alpha)V], \quad (1)$$

where V is a value (state) vector, $R(\alpha)$ is a reward vector, ω is a scalar discount factor, $S(\alpha)$ is a transition probability matrix, and α is a policy (control vector). The objective is to find the optimal policy α and corresponding value vector V , which represents the maximum expected discounted sum of future rewards.

The main approaches to solving problem (1) are policy iteration, value iteration, and linear programming [1]. Both policy iteration and value iteration methods strongly depend on the efficiency of algorithms to solve a linear system of equations (LSE). Policy iteration algorithm solves equation

$$V = R(\alpha^{(i)}) + \omega S(\alpha^{(i)})V, \quad (2)$$

for every iteration i of policy α . Value iteration algorithm calculates estimates for next value vector iterates using equation

$$V^{(i+1)} = \max_{\alpha} [R(\alpha) + \omega S(\alpha)V^{(i)}]. \quad (3)$$

The main approaches to accelerate the convergence of algorithm (3) are similar to those for traditional LSE: Gauss-Seidel, Successive Overrelaxation (SOR), etc. [1]. As a result, faster LSE algorithms can help accelerate existing optimization algorithms and broaden the application area of MDPs.

Mathematically a linear system of equations can be formulated as follows:

$$x = Ax + b, \quad (4)$$

where A is a known coefficient matrix, b is a known vector, and x is an unknown solution vector. Two major directions to solving LSE are commonly recognized: direct and iterative [2]. Direct methods [3] often involve factorization, such as Gaussian elimination, and forward and backward substitution on the vector b .

Iterative algorithms are usually divided into the stationary, projection, and multigrid classes [2,4,5]. The objective of this paper is to present, study, and analyze several nonlinear algorithms accelerating one of the standard stationary methods and discuss the potential applications of the algorithms.

LSE algorithm for a matrix with nonnegative components

Delinearization, a technique that applies nonlinear correcting (feedback) factors to iterative methods for solving linear problems, has been thoroughly studied by the first author over the course of the past thirty years [6–14]. A significant part of this extensive research [6-9, 13] dealt with a delinearization algorithm specifically developed for linear systems. The linear iterative method for solving problem (4) that is accelerated by the delinearization algorithm can be presented as:

$$x^{(k+1)} = Ax^{(k)} + b. \quad (5)$$

Everywhere in this paper $\lambda_i(A)$ denotes eigenvalues of matrix A . It is assumed that equation (4) has a unique solution and hence for all i $\lambda_i(A) \neq 1$.

Algorithm in pseudocode

Let $r = 1$. If $\|x^{(n)} - b - Ax^{(n)}\| > \varepsilon$, consider a scalar f such that $f x^{(n)} = b + A f x^{(n)}$ eventually holds.

Then take any m -dimensional row vector $\phi \geq 0$, any $x^{(0)}$ and compute $f = \phi b / [\phi(I - A)x^{(n)}]$.

Iterate
$$x^{(n+1)} = b + \frac{\phi b A x^{(n)}}{\phi(I - A)x^{(n)}}, \quad n = 0, 1, 2, \dots \quad (6)$$

until $\|x^{(n)} - b - Ax^{(n)}\| \leq \varepsilon$. **Stop.**

Analysis of the algorithm

Let $h \equiv \phi(I - A)$, $x^{(0)} > 0$ and $\phi b = 1$. Then the iterative process (6) can be rewritten as

$$x^{(n+1)} = b + Ax^{(n)} / hx^{(n)}, \quad n = 0, 1, 2, \dots \quad (7)$$

or as
$$x^{(n+1)} = Dx^{(n)} / hx^{(n)}, \quad (8)$$

where
$$D = bh + A \quad (9)$$

It is clear that $h \neq 0$, otherwise it would follow that $\phi = \phi A$ and $\lambda_i(A) = 1$. It can be demonstrated by induction that for all n and arbitrary $x^{(0)}$

$$x^{(n)} = D^n x^{(0)} / h D^{n-1} x^{(0)}. \quad (10)$$

Let $|\delta_1| \geq |\delta_2| \geq \dots \geq |\delta_m|$ be eigenvalues of the matrix D with corresponding eigenvectors p_k and w_k :

$$p_k D = \delta_k p_k, \quad D w_k = \delta_k w_k, \quad k = 1, \dots, m \quad (11)$$

It is clear that $|\delta_1| \geq 1$ since $\phi D = \phi$. Let us further define a set

$$\Phi = \{\phi : \phi > 0, \phi b = 1, \phi(I - A) > 0\} \quad (12)$$

Everywhere in this paper $C = \{c_{ij}\} > 0$ means that all $c_{ij} > 0$ for every matrix.

Lemma 1: If $\lim_{n \rightarrow \infty} x^{(n)}$ exists, $A \geq 0$, $b > 0$, $x^{(0)} > 0$, then for every $\phi \in \Phi$, $\lim_{n \rightarrow \infty} x^{(n)} = v$, where v is a solution of (4).

Proof: First, since $h > 0$ and $D > 0$, then $x^{(n)} > 0$ for all $n = 1, 2, \dots$. Let $\lim_{n \rightarrow \infty} x^{(n)} = x^*$. Then $x^* = b + Ax^*/hx^*$ where $x^* \geq b > 0$. If $hx^* = 1$, then $x^* = v$. But from (8) $\phi x^* = \phi Dx^*/hx^* = \phi x^*/hx^*$. Thus $hx^* = 1$ since $\phi x^* \neq 0$, and hence $x^* = v$.

Theorem 1 : If $A \geq 0$ and is a nonsingular matrix, $b > 0$, $\lambda_1(A) < 1$ and $x^{(0)} > 0$, then for every $\phi \in \Phi$, $x^{(n)}$ converges to a solution of equation (1), and $\delta_1(D) = 1$.

Proof: Let us demonstrate that Φ is not an empty set. Indeed, let $\phi(I - A) = \Delta$, where Δ is an arbitrarily-selected and positive vector. Then ϕ exists since $(I - A)^{-1}$ exists and $\phi = \Delta(I - A)^{-1} = \Delta(I + A + A^2 + \dots)^{-1} > 0$. Thus $D \geq 0$, and from the Perron-Frobenius theorems $\delta_1 > |\delta_2|$, $p_1 \geq 0$, and $w_1 \geq 0$ [15, 16]. If the eigenvectors p_1 and w_1 of D are normalized in such a way that $p_1 w_1 = 1$, then it follows that [17]

$$\lim_{n \rightarrow \infty} (D/\delta_1)^n = w_1 p_1 \quad (13)$$

Hence from (8)

$$\lim_{n \rightarrow \infty} x^{(n)} = \lim_{n \rightarrow \infty} \delta_1 \frac{(D/\delta_1)^n x^{(0)}}{h(D/\delta_1)^{n-1} x^{(0)}} = \delta_1 \frac{(w_1 p_1) x^{(0)}}{h(w_1 p_1) x^{(0)}} = \delta_1 \frac{w_1}{h w_1},$$

since $p_1 x^{(0)} \neq 0$, $w_1 p_1 \neq 0$, and $h w_1 \neq 0$. The former two relationships hold because $p_1 \geq 0$, $w_1 \geq 0$, $x^{(0)} > 0$. Let us demonstrate that $h w_1 \neq 0$. Since $\phi = \phi D$, then $\delta_1(D) \geq 1$. If $h w_1 = 0$, then $\delta_1 w_1 = D w_1 = b h w_1 + A w_1 = A w_1$, which contradicts with the assumption that $\lambda_1(A) < 1$.

Finally, by Lemma 1 $\delta_1 w_1 / h w_1 = v$.

However, this is possible if $\delta_1 = 1$. Indeed $\delta w_1 / h w_1 = b + A \delta w_1 / h w_1$ or $\delta_1 w_1 = (b h + \delta_1 A) w_1$. On the other hand $\delta_1 w_1 = D w_1 = (b h + A) w_1$. Hence $\delta_1 = 1$ since $A \neq 0$.

Comment: If $\sum_{j=1}^m a_{ij} \leq 1$ for all $j=1, \dots, m$, then $eA < e$ where $e = (1, \dots, 1)$, $\dim(e) = m$. Thus $e(I - A) > 0$ and $\phi = e/eb \in \Phi$.

Theorem 2 (spectral properties of matrix D) : If D has a unique largest eigenvalue δ_1 , then $\delta_1 = 1$ and $\delta_k = \lambda_k$ for all $k \neq 1$.

Proof: Consider the following system of $(2m + 1)$ equations with $(2m + 1)$ unknowns (ϕ, w_1, γ) :

$$\phi(I - A) = \gamma w_1^T, \quad (m \text{ equations}) \quad (14)$$

$$\begin{aligned} [b\phi(I-A) + A]w_1 &= w_1, \quad (m \text{ equations}) \\ \phi b &= 1. \quad (one \text{ equation}) \end{aligned} \quad (15)$$

Let $h^* = \gamma w_1^T$. Then $h^* w_k = \gamma w_1^T w_k = 0$ and $w_1^T w_k = 0$ for all $\delta_1 \neq \delta_k, k \neq 1$ [16].

Hence $\delta_k w_k = bh^* w_k + Aw_k = Aw_k$, which means that $\delta_k = \lambda_k, k \neq 1$. On the other hand, since $\phi D = \phi$, then one of eigenvalues $\delta_1 = 1$. At the same time, $\lambda_l \neq 1$ for all $l = 1, \dots, m$. Hence $\delta_1 = 1$.

Results of experiments and further analysis

Numerous computer experiments showed that the process (8) has many advantages: it converges fast not only when $\lambda_1(A)$ is very close to unity, (say, $\lambda_1(A) = 0.99999$), but also when $1 < \lambda_1(A) \leq \nu(A)$, where $\nu(A)$ is an increasing function of m . Besides, the larger is m , the faster is the rate of convergence if matrix $A = \nu S$, S is a randomly-generated stochastic matrix, and ν is a scalar. These properties can be explained on the basis of the following facts.

(1) John von Neumann demonstrated in an unpublished paper that if S is a randomly-generated stochastic matrix, then the expected value of its second largest eigenvalue $\beta_2(S, m)$ is a decreasing function of the dimension m [18].

(2) In accordance with Theorem 2 a properly selected h^* transforms a spectrum $(\lambda_1, \lambda_2, \dots, \lambda_m)$ of the matrix A into a spectrum $(\delta_1, \delta_2, \dots, \delta_m)$ of the matrix D , where $\delta_1 = 1$ and $\delta_k = \lambda_k$ for all $k \neq 1$. If $|\lambda_2| < 1$ and all conditions of Theorem 1 hold, then $\lim_{n \rightarrow \infty} x^{(n)} = v$.

(3) $\lambda_k(A) = \beta_k(S)\nu$. Since h^* transforms $(\lambda_1, \lambda_2, \dots, \lambda_m)$ into $(1, \lambda_2, \dots, \lambda_m)$, then $|\delta_2| < 1 = \delta_1$ as long as $|\beta_2(S)|\nu < 1$. The latter explains most of the properties of the process (8) observed during the computer experiments, provided that we understand the nature of the spectrum $(\delta_1, \delta_2, \dots, \delta_k)$ if $h \neq \gamma w_1^T$.

Theorem 3: Let $\delta_1 \neq \delta_k$ for all $k \neq 1$, and let $\tilde{h} = h^* + \varepsilon z$, where z is a restricted vector, and ε is a small scalar. Then \tilde{h} transforms $(\lambda_1, \dots, \lambda_m)$ of matrix A into $(1, \tilde{\delta}_2, \dots, \tilde{\delta}_m)$ of matrix D where $\tilde{\delta}_k = \lambda_k + \varepsilon p_k b z w_k / p_k w_k + o(\varepsilon)$ for all $k > 1$.

Proof: Let $\tilde{\delta}_k = \delta_k + \delta_k^{(1)}\varepsilon + \delta_k^{(2)}\varepsilon^2 + \dots$, $\tilde{w}_k = w_k + w_k^{(1)}\varepsilon + w_k^{(2)}\varepsilon^2 + \dots$, $\tilde{D} = b\tilde{h} + A = D + \varepsilon b z$, where $\tilde{D}\tilde{w}_k = \tilde{\delta}_k \tilde{w}_k$.

Then from $(D + \varepsilon b z)(w_k + w_k^{(1)}\varepsilon + \dots) = (\delta_k + \delta_k^{(1)}\varepsilon + \dots)(w_k + w_k^{(1)}\varepsilon + \dots)$ it follows that

$$b z w_k + D w_k^{(1)} = \delta_k^{(1)} w_k + \delta_k w_k^{(1)}. \quad (16)$$

Multiplying both parts of (16) by vector p_k and taking into account (11), we have

$$\delta_k^{(1)} = p_k b z w_k / p_k w_k, \text{ where } p_k w_k \neq 0 \quad (17)$$

Thus for small ε , $\tilde{\delta}_k = \delta_k + \delta_k^{(1)}\varepsilon + o(\varepsilon) = \lambda_k + \varepsilon p_k b z w_k / p_k w_k + o(\varepsilon)$ if $k \neq 1$.

To generalize (6), let us consider a process

$$x^{(n)} = Dx^{(n-1)} / hx^{(n-1)}, \quad (18)$$

where $D = bh + A$ and h is selected in such a way that $\delta_1(D) = 1$. If $A \geq 0$, $b > 0$, $h > 0$, and $\lambda_1(A) < 1$, then $x^{(n)} \rightarrow v$.

Let us select a vector h that satisfies

$$(bh + A)e \leq e, \quad (19)$$

where $e = (1, \dots, 1)$ and $\dim(e) = m$.

Then $bhe \leq (I - A)e$ or $he \leq \left(1 - \sum_{j=1}^m a_{ij}\right) / b_i$ for all $i = 1, \dots, m$. The equation (19) holds if, for example,

$$he = \min_i \left(1 - \sum_j a_{ij}\right) / b_i. \quad (20)$$

Furthermore $he > 0$ if $\sum_{j=1}^m a_{ij} < 1$ for all $i = 1, \dots, m$. There are many ways to select $h > 0$ and to satisfy (20). One of them is to assign

$$h_j = \min_i \left(1 - \sum_j a_{ij}\right) / mb_i, \text{ for all } j = 1, \dots, m.$$

In general from the above theorems it is not clear whether $x^{(n)} \rightarrow v$ if, say, h is **not** a positive vector, or D is **not** a positive matrix, or $\delta_1(D)$ is **not** unique. On the other hand, more than a thousand computer experiments demonstrated that $x^{(n)} \rightarrow v$ even if (a) $\lambda_1(A) > 1$ and/or (b) $\phi \notin \Phi$. Hence, a further analysis is required to understand the process (7). That possibly can be achieved by more elaborate analysis of equation (10) and/or by other algebraic and nonalgebraic (geometric, topological, operator) methods. Approaches and results described in [17, 19] would probably be helpful for future researchers.

Uniqueness of solution

Consider another normalizing vector ϕ' for which process (8) converges to a solution v' of (1), and $v' \neq v$. Let $v_* \equiv v' - v$. Then $v_* = Av_*$. which means that matrix A has an eigenvalue of unity. The contradiction proves the uniqueness of the solution.

Rate of convergence

Consider $\varepsilon_n \equiv x^{(n)} - v$, where v is the solution of (4).

Theorem 4: In a neighborhood of the solution

$$\varepsilon_{n+1} = (D - vh)\varepsilon_n + o(\varepsilon_n) \quad (21)$$

Proof:

$$\varepsilon_{n+1} = x^{(n+1)} - v = b + \frac{A(\varepsilon_n + v)}{h(\varepsilon_n + v)} - v = \frac{(D - vh)(\varepsilon_n + v)}{h(\varepsilon_n + v)}. \quad (22)$$

It is easy to check that $(D - vh)v = 0$ and $hv = 1$. If vector ε_n is small, then $h\varepsilon_n$ is also small compared with unity. Thus

$$\varepsilon_{n+1} = \frac{(D - vh)\varepsilon_n}{(1 + h\varepsilon_n)} = (D - vh)\varepsilon_n + o(\varepsilon_n)$$

Taking into account that

$$D - vh = A(I - vh) = A[I - (I - A)^{-1}b\phi(I - A)], \quad (23)$$

one obtains

$$\varepsilon_{n+1} = A[I - (I - A)^{-1}b\phi(I - A)]\varepsilon_n = AB\varepsilon_n + o(\varepsilon_n), \quad (24)$$

where $B \equiv [I - (I - A)^{-1}b\phi(I - A)]$.

Lemma 2: Matrices AB and $A(I - b\phi)$ have the same spectrum.

Proof: Indeed, $|AB - \lambda I| = |(I - A)^{-1}| |A[(I - A) - b\phi(I - A)] - \lambda(I - A)| =$
 $= |\text{seconddeterminant}| |(I - A)^{-1}| |A(I - b\phi) - \lambda I| = 0$, since matrices A and
 $(I - A)^{-1}$ are commutable, the determinant of the product of the matrices equals the
product of determinants, and $(I - A)^{-1}$ exists. Hence the rate of convergence of the
process (7) depends on the largest eigenvalue of matrix $A(I - b\phi)$.

Properties of matrices B and $I - b\phi$

Let $P \equiv I - b\phi$. Then 1.) Both B and C are projecting matrices: $B^2 = B, C^2 =$
 C ; 2.) $Bv = 0$; 3.) $Cb = 0$; 4.) $\phi C = 0$; 5.) Spectra of the matrices B and C are equal
to $(1, 1, \dots, 1, 0)$ [20].

Lemma 3: If ϕ is an eigenvector of A , $\phi A = \lambda\phi$, then

$$[AC]^n = A^n C \text{ and } (AB)^n = A^n B.$$

Proof (by induction):

$$(A - Ab\phi)(A - Ab\phi) = A^2 - A^2b\phi - Ab\phi AC = A^2C - Ab\lambda\phi C = A^2C$$

Let $(AC)^k = A^k C$ hold for $k \leq n$.

Then $(AC)^{n+1} = (AC)^n(AC) = A^n CAC = A^{n+1} - A^{n+1}b\phi - A^n b\phi AC = A^{n+1}C$.

The proof that $(AB)^n = A^n B$ is analogous. Indeed, let $Q = (I - A)^{-1}b\phi$.

Easy to observe that

- 1) $B = I - Q(1 - \lambda)$;
- 2) $QA = \lambda Q$;
- 3) $Q^2 = (1 - \lambda)^{-2}Q$;
- 4) $QB = 0$.

Then $(AB)^2 = [A - (1 - \lambda)AQ]AB = A^2B - (1 - \lambda)A\lambda QB = A^2B$.

Let $(AB)^k = A^k B$ for $k \leq n$. Then $(AB)^{n+1} = (AB)(A^n B) = [A - (1 - \lambda)AQ]A^n B =$
 $= A^{n+1}B - (1 - \lambda)A\lambda^n QB = A^{n+1}B$.

Thus from (24)

$$\varepsilon_{n+s} = (AB)^s \varepsilon_n = A^s B \varepsilon_n + o(\varepsilon_n). \quad (25)$$

Theorem 5: If ϕ is the eigenvector of matrix A corresponding to the largest positive eigenvalue λ_1 , i.e. $\phi A = \lambda_1 \phi$, and $\phi \geq 0, A \geq 0, b \neq 0$, and the inverse matrix A^{-1} exists, then every nonzero eigenvalue of matrix AB is at the same time the eigenvalue of matrix A . However λ_1 is not an eigenvalue of matrix AB .

Proof: 1.) Consider matrix $F = A - \lambda_1 b \phi$ and let $Fz = \lambda z, (\lambda \neq 0), (z$ is an eigenvector of $F)$. Then $b \phi Fz = \lambda b \phi z$. On the other hand, $b \phi Fz = b \phi (A - \lambda_1 b \phi)z = b(\lambda_1 \phi - (\phi b) \lambda_1 \phi)z = 0$.

Thus $b \phi z = 0$, since $\lambda \neq 0$.

Then $Fz = Az - \lambda_1 b \phi z = Az = \lambda z$, i.e., z is also an eigenvector of matrix A .

Assume that z_1 is an eigenvector of matrix F corresponding to λ_1 , i.e. $Fz_1 = \lambda_1 z_1$. Then $b \phi z_1 = 0$, and $Az_1 = \lambda_1 z_1$, but $\phi z_1 \neq 0$ (biorthogonal property). Thus the assumption that λ_1 is an eigenvector of F leads to the wrong conclusion that $b \phi z_1 = 0$ since $b \neq 0$ and scalar $\phi z_1 \neq 0$.

2.) Matrices AB and F have the same spectrum if A^{-1} exists, since $|AB - xI| = |A - \lambda_1 b \phi - xI|$ for any x . Indeed

$$|AB - xI| = |I - A|^{-1} |A| |I - b \phi - x A^{-1}| |I - A| = |A - b \phi A - xI| = |A - \lambda_1 b \phi - xI|.$$

QED.

Theorem 6. If ϕ is a nonnegative eigenvector of matrix $A (A \geq 0)$, corresponding to positive eigenvalue λ_1 , then the spectral radius of matrix AB is less than the spectral radius of matrix $A : r(AB) < r(A)$.

Proof: From the Perron-Frobenius theorem, λ_1 is the largest and unique eigenvalue of matrix A . Hence $r(A) = \lambda_1 > \max |\lambda(AB)| = r(AB)$. Thus the rate of convergence for process (6) is faster than the rate of convergence for process (5).

Examples

1. Let $A = kI + ra\phi$, where a is a column vector, $a \neq 0, k$ and r are scalars. It is easy to check that $\lambda_1(A) = k + r, \lambda_2(A) = \dots = \lambda_m(A) = k$. Indeed, $|kI + ra\phi - \lambda I| = |ra\phi - (\lambda - k)I| = |r| |a\phi - \frac{\lambda - k}{r} I| = 0 = |a\phi - \mu I|$. Since $a\phi$ has a rank of 1, then $\mu_1 = \phi a, \mu_2 = \dots = \mu_m = 0$ [20]. Hence $\lambda_2 = \dots = \lambda_m = k, \lambda_1 = r(\phi a) + k, A(I - b\phi) = (kI + ra\phi)(I - b\phi) = k(I - b\phi), \lim_{n \rightarrow \infty} [AB]^n = \lim_{n \rightarrow \infty} k^n (I - b\phi)$, and the algorithm (6) converges with rate $|k|$ if $|k| < 1$.

2. If $A = ra\phi$ (the case where $k = 0$), then the algorithm (6) converges to the solution after only one iteration. Indeed

$$v' = b + \frac{ra\phi v^{(0)}}{(1 - r\phi a)\phi v^{(0)}} = b + \frac{ra}{1 - r\phi a}$$

3. Let $k > 0, r < 0$. Then

$$\lambda_1(A) = \dots = \lambda_{m-1}(A) = k \quad \lambda_m(A) = k - r\phi a \text{ if } \phi a \geq 0$$

Thus $\lambda_1(AB) = \lambda_1(A)$, and the algorithm (6) does not have any advantages compared with process (5).

General case for normalizing vector ϕ

From previous considerations (Theorem 3), if $\phi = u_k (Au_k = \lambda_k u_k)$ and $k \neq 0$, then $r(AB) = r(A)$ and algorithm (6) does not have any advantages compared with process (5). However in most cases it is very difficult to find the eigenvector of matrix A corresponding to eigenvalue λ_0 . Generally if $\phi \neq u_k$, $k = 1, \dots, m$, then $r(AB) \leq r(A)$, and $r(AB)$ significantly depends on the choice of vector ϕ .

Computer experiments

Many hundreds of computer experiments for stochastic square matrices A with different values of m ($m = 2, 5, 10, 20, 30, 50, 60, 75$), various spectra of matrices A , and various normalizing vectors $\phi > 0$ indicate that the necessary number of iterations required to achieve a given accuracy ε ,

$$\|x^{n+1} - x^n\| \leq \varepsilon,$$

is much smaller for the algorithm (6) than for the Jacobi process (5). The algorithm (6) is particularly advantageous where $\phi = e = (1, 1, \dots, 1)$.

Geometric interpretation

If ϕ is an eigenvector of matrix A , then $\phi x^{(n)} = \phi b / (1 - \lambda_k)$ and $\phi v = \phi b / (1 - \lambda_k)$. Hence $\phi \varepsilon_n = \phi(x^{(n)} - v) = 0$, which means that for every n a vector of error ε_n is orthogonal to an eigenvector ϕ of matrix A .

LSE algorithms for a matrix with negative and nonnegative components

Consider situations where A matrix contains both negative and nonnegative components. In this case, A can be presented as a difference of two matrices with nonnegative elements, i.e. $P - N$. In the present discussion, only $r(P) > r(N)$ will be considered. Henceforth, the underlying linear algorithm can be formulated as:

$$x^{(k+1)} = Px^{(k)} - Nx^{(k)} + b. \quad (26)$$

Delin1 algorithm

The first developed algorithm (Delin1) applies two nonlinear accelerators, $g(x^{(k)})$ and $f(x^{(k)})$, to equation (26) resulting in the following expression:

$$x^{(k+1)} = g(x^{(k)})Px^{(k)} - f(x^{(k)})Nx^{(k)} + b. \quad (27)$$

Applying an idea similar to the one used for deriving iterative algorithm (6), one may obtain the following expressions:

$$g(x^{(k)}) = \left(\frac{\phi(b + Px^{(k)})}{\phi[(I + N)x^{(k)}]} \right)^s, \quad (28)$$

$$f(x^{(k)}) = [g(x^{(k)})]^t, \quad (29)$$

where $s > 0$, $t \leq 0$ are scalars. It is noteworthy that $g(x^{(k)})$ and $f(x^{(k)})$ are defined in such a way as to guarantee that if $g(x^{(k)}) > 0$, then $f(x^{(k)}) \leq 1$, and vice versa. It can be shown that if there is a stationary point u for the iterative process (27-29), then

$$g(u) = f(u) = 1. \quad (30)$$

Therefore one obtains equation (26) and u is the solution to (4). To prove the above statement, let us assume that (30) does not hold and instead $g(u) < 1$. Then (29) implies that $f(u) \geq 1$, and from (27) it follows that

$$u < Pu - Nu + b \Rightarrow 1 < \frac{\phi(b + Pu)}{\phi(I + N)u} \Rightarrow g(u) > 1,$$

which is in contradiction with the initial assumption. Analogously, one comes to a contradiction if $g(u) > 1$ is assumed.

Delin2 algorithm

The second delinearization algorithm (Delin2) considers one accelerator and thus can be formulated as:

$$x^{(k+1)} = g(x^{(k)})(P - N)x^{(k)} + b. \quad (31)$$

Again applying an idea similar to the one used in algorithm (6), an alternative formulation of $g(x^{(k)})$ can be developed:

$$g(x^{(k)}) = \left(\frac{\phi(b - Nx^{(k)})}{\phi[(I - P)x^{(k)}]} \right)^s. \quad (32)$$

In this case it can be seen that the numerator and denominator may change signs, which may result in occurrence of special conditions that are impossible for Delin1. At the same time, this algorithm seems to be closer to (6) considering the fact that $r(P) > r(N)$.

Delin3 algorithm

The third algorithm (Delin3) combines features of both Delin1 and Delin2. First, it applies two nonlinear accelerators and thus can be described by expression (27). Second, it uses the same nonlinear accelerator for $g(x^{(k)})$ as Delin2, which is described by formula (32). Expression for $f(x^{(k)})$ is given by the following formula:

$$f(x^{(k)}) = [g(x^{(k)})]^{-\frac{t}{s}}, \quad (33)$$

where t can take any real values.

Complexity of the delinearization algorithms

Before we start analyzing the results of computer experiments, let us briefly take a look at the complexity involved in each iteration of the delinearization algorithms. The costliest computational (time complexity) part is associated with two matrix-vector products for P and N , two inner products for ϕ , and two exponentiations (only one exponentiation in the case of Delin2). The other computational costs, like vector multiplications by a scalar, vector updates, are relatively insignificant. Storage complexity strongly depends on the dimension and sparsity of matrices P and N . In summary, the total complexity of one iteration for the delinearization algorithms does not notably exceed the complexity of one iteration for the linear algorithm (26).

Computer experiments

Detailed analysis of algorithm (6) at the beginning of this paper demonstrated that the major factors affecting the convergence range and rate in the general case are the two largest eigenvalues of A and the dimension of the linear system. Numerous experiments conducted for the three proposed delinearization algorithms showed that the same principle applies in the present case. Therefore, an analysis based on the spectral radii of P and N and the dimension of A is presented.

In all of the experiments, stochastic square matrices P and N were considered and the stopping criterion, the infinity norm between two consecutive iterates, was set to 10^{-10} . The components for both matrices were generated using a uniform random generator that comes with the Mathematica software package and vector $\phi = (1, 1, \dots, 1)$ was applied. The initial solution vector was set to $(10^{-3}, 10^{-3}, \dots, 10^{-3})$. As both P and N were randomly-generated stochastic matrices and the dimension of the linear system was relatively large, it is reasonable to assume in the following discussion that $r(P - N) \approx r(P) - r(N)$. Several thousands of experiments were carried out, a summary of which is provided here. The purpose of this analysis is to indicate the advantageous properties of the delinearization algorithms which could be used to identify specific applications in the future.

Results of experiments with Delin1 can be summarized as follows.

- 1) The algorithm converges only if $r(P - N) < 1$:
- 2) The smaller is $r(P - N)$, the worse is the acceleration of Delin1 because the spectral radius of matrix $(P - N)$ determines the convergence speed of the linear algorithm (26). The spectral radius becomes much less than one resulting in a significantly better convergence for the linear algorithm while Delin1, depending on the second largest eigenvalue, $\lambda_2(P - N)$, which does not experience such a radical absolute reduction, increases its convergence rate at a considerably slower pace.
- 3) Out of the two boosters s and t , t is the one that plays the major role in acceleration by significantly changing its values depending on $r(P - N)$. It can be seen that the optimal value of t , t_{\min} , increases with increase in the value of $r(P - N)$, thus indicating that t is the factor responsible for most of the acceleration provided by Delin1.

4) The larger is $r(P)$ with $r(P-N)$ kept constant, the worse is the acceleration and the smaller is t_{\min} . This behavior can be partially explained using the dependence of Delin1 convergence rate on the second largest eigenvalue of $(P-N)$.

5) Rule for determining the optimal value of s , s_{\min} :

if $0 < r(P-N) \leq 0.9$, $s_{\min} = 1$;

if $0.9 < r(P-N) \leq 0.99$, $s_{\min} = 1.5$;

if $0.99 < r(P-N) < 1$, $s_{\min} = 2$.

6) If $r(P-N)$ is close to one, the acceleration reaches values of 30 – 40 times and t_{\min} , the parameter attributing the most to the acceleration, reaches values of several hundred.

7) Similarly to algorithm (6), the number of iterations for Delin1 decreases with increasing m , the dimension of the system.

Let us now turn our attention to the other two delinearization algorithms: Delin2 and Delin3. If $r(P-N) < 1$, then the Delin2 and Delin3 possess the following properties:

- 1) the larger is $r(P-N)$, the higher is the acceleration (the same way as Delin1);
- 2) number of iterations experiences relatively minor changes with increase in the dimension of $(P-N)$ while Delin1 demonstrates a significant increase in the acceleration;
- 3) both algorithms diverge when $r(P) = 1$;
- 4) Delin2: if $r(P-N) < 0.6$, $s_{\min} = 1$;
if $0.6 < r(P-N) < 1$, $s_{\min} = 0.5$;
- 5) Delin3: $s_{\min} = 0.25 \dots 0.75$. In contrast to Delin1, the value of t_{\min} does not experience major changes, varying from -3 to 2 . For most of the cases, $s = 0.5$ and $t = 0$ can be used.

Both Delin2 and Delin3 also converge when $r(P-N) > 1$. The first major result in this case, the higher is m , the larger is $r(P-N)$ for which the convergence occurs, once again implies the dependence of both algorithms on the second largest eigenvalue. In fact, the algorithms started diverging just when $\lambda_2(P-N)$ exceeded one. It can also be seen that the convergence rate was relatively slow when $r(P-N)$ was close to one, but was gaining speed as $r(P-N)$ was moving away from one. In a sense, the algorithms were operating in some transition area where the divergence effects were still noticeable and the second largest eigenvalue was not yet the dominant factor. Once some minimum was reached (the boundary of the transition area was passed), $\lambda_2(P-N)$ became the driving force of the algorithms.

Comparison with other algorithms

Let us compare the convergence of the delinearization algorithms with the best stationary method, Successive Overrelaxation (SOR). This algorithm and its

symmetric variant SSOR are widely used either as preconditioners for Krylov methods and smoothers for multi-grid methods or, in some specific applications, as standalone algorithms [2]. Consideration is limited only to the cases where SOR converges, i.e. situations where $r(P - N) > 1$ are not discussed. Convergence of Delin2 and Delin3 in the area where $r(P - N)$ is close to one is usually much slower than for Delin1, and in the case where $r(P) \approx 1$ both of those algorithms diverge. Therefore, it is reasonable to take those situations out the consideration and mark the appropriate cells with “x”. As can be seen from Table 1 (each row displays the average of 20 experiments), Delin1 can be much faster than SOR, especially in the area where $r(P - N)$ is close to unity. In addition, Delin1 strongly depends on the value of the second largest eigenvalue which results in the acceleration increase for higher m while SOR appears to be independent of the matrix dimension and the second largest eigenvalue.

Conclusion

In this paper, the general idea of nonlinear correcting factors that was successfully applied to accelerate and derive algorithms for many linear and nonlinear problems is used to develop several efficient procedures for solving LSE based on the Jacobi algorithm. The algorithms are developed not just for the case where the coefficient matrix contains only nonnegative components, but also for the case where the coefficient matrix can be presented as a difference of two matrices with nonnegative components. Results of the theoretical study and extensive computer experiments for stochastic matrices are presented and analyzed to determine the conditions under which the studied algorithms converge, and the areas of their maximum convergence rate are identified. The algorithms developed are shown to outperform the fastest existing stationary iterative algorithms when the spectral radius of the coefficient matrix is less than one, and are even able to converge when the spectral radius exceeds unity. As a result, the algorithms may be used to accelerate optimization algorithms for Markov decision processes.

References

1. Kallenberg, L. (2002). Finite State and Action MDPs. Handbook of Markov Decision Processes: Methods and Applications. Feinberg, Y. & Shwartz A., Eds. Netherlands: Kluwer, 21-88.
2. Saad, Y., & Van der Vorst, H. (2000). Iterative solution of linear systems in the 20th century, J. Comput. Appl. Math., 123, 1-33.
3. Duff, I. S., Erisman, A.M., & Reid, J. K. (1986). Direct methods for sparse matrices. Oxford: Clarendon Press.
4. Saad, Y. (2003). Iterative methods for sparse linear systems (2nd ed.). Society for Industrial & Applied Mathematics.
5. Barrett, R., Berry, M., Chan, T., *et al.* (1994). Templates for the solution of Linear systems: building blocks for iterative methods. Philadelphia: SIAM.
6. Verkhovsky, B. (1976). Feedback algorithm for system of equations. IBM Technical Disclosure Bulletin, 18, No.10.
7. Verkhovsky, B. (1976). Algorithm with controlled feedback for system of equations with stochastic matrix. IBM Technical Disclosure Bulletin, 18, No.10.

8. Verkhovsky, B. (1976). Algorithm for system of equations with stochastic matrix. IBM Technical Disclosure Bulletin, 18, No.10.
9. Verkhovsky, B. (1976). Algorithm with nonlinear acceleration for a system of linear equations. Research Report, No.76-WR-1, Princeton University.
10. Verkhovsky, B. (1977). Smoothing systems design and parametric Markovian programming. In H. C. Tijms & J. Wessels (Eds.), Markov Decision Theory (pp. 105-117). Amsterdam: Mathematische Centrum.
11. Verkhovsky, B. (1978). Delinearization algorithms for elliptic partial differential equations. Research Report, 78-WR-3, Princeton University, 1-12.
12. Veroy (now Verkhovsky), B. (November 1985). Projection algorithm for a stochastic dynamic programming problem. Proc. NEACP Technology & Science Conference, Newton, Massachusetts.
13. Veroy, B. (1987). Convergence and complexity of aggregation-disaggregation algorithm. Research Report, CS-87-04, NJIT.
14. Verkhovsky, B., & Polyakov Yu. (2003). Feedback algorithm for the single-facility minisum problem. Annals of the European Academy of Sciences, 1, 127-136.
15. Perron, O. (1907). Zur Theorie der Matrizen. Mathematische Annalen, LXIV.
16. Gelfand, I. M. (1951). Lectures on Linear Algebra. Moscow: Fizmatgiz.
17. Karlin, S. (1959). Positive Operators. J. Math. Mech., 8, 901-937.
18. Bergmann, V. (1975). Private Communication. Princeton University.
19. Krein, M. G., & Rootman, M. A. (1948). Linear Operators. Uspekhi Mat. Nauk, 3, No. 1.
20. Young, D. M., & Gregory, R. T. (1973). A Survey of Numerical Mathematics, v. 2, 999-1062. Addison-Wesley.
21. Gantmacher, F. R. (1990). The Theory of Matrices. American Mathematical Society.

Table 1 Convergence comparison at $r(N) \approx 0.1$

$r(P)$	Delin1	Delin2	Delin3	Linear	SOR	m
0.65	13.3	15.2	13.4	39.25	14.75	10
0.85	16.85	16.65	15.95	79.75	21.85	10
0.95	21.5	41.32	19.6	140.9	31.6	10
1.05	26.75	x	x	441.2	62.35	10
1.09	84.1	x	x	2243.75	237.9	10
0.65	11	16.05	11.95	39.05	16.05	30
0.85	14.1	16.25	13.85	79.8	23.95	30
0.95	16.9	34.4	17.8	140.2	32.6	30
1.05	17.6	x	x	441.65	65.35	30
1.09	41.8	x	x	2242.4	189.2	30
0.65	10.65	16.1	11.75	39	16.1	50
0.85	13.25	16.4	13.15	79.7	23.6	50
0.95	15.9	35.55	17.85	140.05	31.65	50
1.05	15.1	x	x	440.25	64.45	50
1.09	31.85	x	x	2242.8	191.7	50