An Improved Self-Adaptive Projection Method for Solving Variational Inequalities

Hongjin He\textsuperscript{1}, Hongchao Zhang\textsuperscript{1}, and Deren Han\textsuperscript{1}

Abstract

In this paper, we propose an improved projection method, where the profitable direction and the step-size are constructed from those in Han and Lo (2002). Thus, it is an “improvement” of the method in Han and Lo (2002). To enhance the numerical efficiency of the algorithm, the self-adaptive strategy for choosing the parameter is adopted. Under mild assumptions, we prove the global convergence of the proposed algorithm. Moreover, some preliminary numerical results are reported, demonstrating that the new algorithm is efficient and reliable.

Key words: Variational inequality problems, projection methods, profitable directions, self-adaptive.

1 Introduction

In this paper, we consider the classical variational inequality problem, denoted by $\text{VIP}(F, \Omega)$, which is to find a vector $u^* \in \Omega$, such that

$$F(u^*)^T(v - u^*) \geq 0, \quad \forall v \in \Omega,$$

where $\Omega$ is a nonempty closed convex subset of $\mathbb{R}^n$, and $F$ is a continuous mapping from $\mathbb{R}^n$ into $\mathbb{R}^n$. Variational inequality problems arise from many important applications in network economics, transportation equilibrium problems, and engineering sciences, etc., see [1, 2, 4, 17]. In past decades, many novel iterative numerical methods, such as projection methods, Newton-type methods, alternating direction methods and proximal point algorithms, have been proposed; see for example, [4–7, 10, 11, 16], and the references therein.

Among all the iterative methods, one of the simplest methods is projection-type method. This type of method is attractive because of its little storage requirement and its easy implementation, especially when the feasible sets are

\textsuperscript{1}School of Mathematical Sciences, Nanjing Normal University; (b) Jiangsu Key Laboratory for NSLSCS, Nanjing 210046, P.R. China.
E-mail: handeren@njnu.edu.cn.

AMO - Advanced Modeling and Optimization. ISSN: 1841-4311
simple, such as balls, boxes and nonnegative orthant. On the other hand, projection methods can readily exploit any separable structure in the corresponding mapping or the constrained set of the problem, i.e., they can perform in a parallel way. Generally, in projection methods, the new iterate $u_{k+1}$ is generated from an arbitrary starting point $u_0 \in \mathbb{R}^n$ via the following procedure:

$$u_{k+1} = P_{\Omega} [u_k - \beta_k \tilde{g}(u_k)], \quad (2)$$

where $P_{\Omega}[\cdot]$ denotes the orthogonal projection from $\mathbb{R}^n$ onto $\Omega$, $\tilde{g}(u_k)$ is a profitable direction, i.e., it satisfies

$$\tilde{g}(u_k)^T (u_k - u^*) \geq \varphi(u_k) \geq 0,$$

and $\varphi(u_k) \geq 0$ is called a measure function, satisfying

$$\varphi(u_k) = 0 \iff u_k \text{ is a solution of VIP}(F, \Omega).$$

By constructing different profitable directions and measure functions, various projection-type methods were proposed. For example, for variational inequality problems with strongly monotone and Lipschitz continuous mappings, Goldstein [3], Levitin and Polyak [15] adopted the mapping $g = F$ as the profitable direction, and proves that if the positive step size $\beta_k$ is judiciously chosen, the generated sequence $\{u_k\}$ converges globally. The strong monotonicity and Lipschitz continuity are quite strict assumptions, which precludes many applications of the methods of Goldstein, and Levitin and Polyak. To relax these strong conditions, Korpelevich [14] first proposed the following extra-gradient method:

$$u_{k+\frac{1}{2}} = P_{\Omega} [u_k - \beta F(u_k)],
\quad u_{k+1} = P_{\Omega} [u_k - \beta F(u_{k+\frac{1}{2}})].$$

When $F$ is monotone and Lipschitz continuous, and $0 < \beta < 1/L_F$ (where $L_F > 0$ is the Lipschitz constant of $F$), the method converges globally. Many variant forms of the extra-gradient method were introduced, for example [9, 12, 18, 19].

In [6], Han and Lo proposed a new self-adaptive projection method for solving variational inequality problems with the following recursion:

$$u_{k+1} = P_{\Omega} [u_k - \gamma \bar{\rho}(u_k, \beta_k) \bar{d}(u_k, \beta_k)], \quad (3)$$

where

$$\bar{d}(u, \beta) = \alpha e(u, \beta) + \beta F(u - \alpha e(u, \beta)),
\quad \bar{\rho}(x, \beta) = \alpha e(u, \beta)^T \{e(u, \beta) - \beta [F(u) - F(u - \alpha e(u, \beta))]/||d(u, \beta)||^2\},$$

and

$$e(u, \beta) = u - P_{\Omega} [u - \beta F(u)] \quad (4)$$
is the residual function. Under the mild conditions, that the underlying mapping $F$ is continuous and monotone, their method is globally convergent for suitable parameter $\beta_k$. In addition, Han and Lo’s algorithm reduced amount of time to compute the projection $P_{\Omega}[\cdot]$ at each line search procedure, and the reported numerical results demonstrated the new algorithm is efficient for solving variational inequality.

Adopted the similar self-adaptive strategy, in this paper we construct a new search direction of VIP$(F, \Omega)$. Based on the new direction, many profitable properties can be obtained and the global convergence is established under some mild assumptions. Furthermore, our preliminary computational experiments show that the new algorithm is efficient and reliable for variational inequality problems.

Our paper is divided into 5 sections. In the next section, we give some useful preliminaries, which play the central roles in the convergence analysis. In section 3, we describe the improved self-adaptive projection algorithm formally and the global convergence is established. Some preliminary compared numerical results are reported in section 4. Finally, we give some conclusion remarks to complete our paper.

## 2 Preliminaries

In this section, we summarize some basic properties, which play significant roles in the following analysis.

Throughout this paper, the projection operator $P_{\Omega}[\cdot]$ from $\mathbb{R}^n$ onto $\Omega$ is defined by

$$P_{\Omega}[u] := \arg \min \{ \| v - u \| : v \in \Omega \},$$

where $\| \cdot \|$ denotes the Euclidean norm. For any closed convex set $\Omega \subseteq \mathbb{R}^n$, the projection operator $P_{\Omega}[\cdot]$ has the following well-known properties; see for example [2], and the references therein.

**Lemma 2.1.** Let $\Omega \subseteq \mathbb{R}^n$ be a closed convex set, Then

$$\langle v - P_{\Omega}[v], w - P_{\Omega}[v] \rangle \leq 0, \quad \forall u \in \mathbb{R}^n, \forall w \in \Omega,$$

consequently, we obtain,

$$\| P_{\Omega}[u] - P_{\Omega}[v] \| \leq \| u - v \|, \quad \forall u, v \in \mathbb{R}^n$$

and

$$\| P_{\Omega}[v] - u \|^2 \leq \| v - u \|^2, \quad \forall u \in \Omega. \quad \square$$

The following lemma states that $\| e(u, \beta) \|$ defined by (4) is nondecreasing and $\| e(u, \beta) \| / \beta$ is nonincreasing with respect to $\beta$. It will play an important role in the following convergence analysis.
Lemma 2.2. For any $u \in \Omega$ and $\beta_2 \geq \beta_1 > 0$, the following two inequalities hold:
\[
\|e(u, \beta_2)\| \geq \|e(u, \beta_1)\|,
\]
and
\[
\frac{\|e(u, \beta_2)\|}{\beta_2} \leq \frac{\|e(u, \beta_1)\|}{\beta_1}.
\]

Proof. See a simple proof in [21].

Lemma 2.3. ([2]) Let $\beta > 0$, then $u^*$ is a solution of the VIP$(F, \Omega)$ if and only if
\[
u^* = P_{\Omega}[u^* - \beta F(u^*)],
\]
i.e.,
\[e(u^*, \beta) = 0.\]

The lemma shows that solving variational inequality VIP$(F, \Omega)$ is equivalent to finding a zero point of the residual function $e(u, \beta)$, and it also provides us a stopping criterion in designing a solution method.

In the following analysis, we assume that:

(a) The underlying mapping $F$ is monotone on $\Omega$, i.e.,
\[(u - v)^T[F(u) - F(v)] \geq 0, \quad \forall u, v \in \Omega;\]

(b) The solution set of VIP$(F, \Omega)$, denoted by $\Omega^*$, is nonempty.

3 The algorithm and convergence analysis

In this section, we first describe our improved self-adaptive algorithm formally. Then some related properties are presented. Finally, we prove the global convergence of the proposed algorithm.


Step0. Given $l \in (0, 1), \eta \in (0, 1), \gamma \in (0, 2), \theta_1 > 1, \theta_2 > 1, \alpha_{-1} = 1$ and $\varepsilon > 0$.

Choose an arbitrarily starting point $x_0 \in \Omega$. Set $k := 0$.

Step1. Set $\beta_k = \min\{1, \theta_1 \alpha_{k-1}\}$ and compute $\|e(u_k, \beta_k)\|$ via (4). If $\|e(u_k, \beta_k)\| \leq \varepsilon$ then stop;

Otherwise, go to Step2.
**Step 2.** Find the smallest nonnegative integer \(m_k\), such that \(\alpha_k = \beta_k l^m_k\) satisfying

\[
\beta_k \|F(u_k) - F(u_k - \alpha_k e(u_k, \beta_k))\| \leq \eta \|e(u_k, \beta_k)\|.  
\] (7)

**Step 3.** Update the iterate via

\[
u_{k+1} = P_{\Omega} [u_k - \gamma \rho(u_k, \beta_k) d(u_k, \beta_k)],
\] (8)

where \(\rho(u_k, \beta_k)\) is given by

\[
\rho(u_k, \beta_k) := \frac{e(u_k, \beta_k)^T g(u_k, \beta_k)}{\|d(u_k, \beta_k)\|^2}.
\] (9)

and

\[
d(u_k, \beta_k) := \alpha_k [e(u_k, \beta_k) - \beta_k F(u_k)] + \beta_k F(u_k - \alpha_k e(u_k, \beta_k)),
\] (10)

\[
g(u_k, \beta_k) := \alpha_k \{e(u_k, \beta_k) - \beta_k [F(u_k) - F(u_k - \alpha_k e(u_k, \beta_k))]\}.
\] (11)

**Step 4.** If

\[
\beta_k \|F(u_k) - F(u_k - \alpha_k e(u_k, \beta_k))\| \leq 0.3 \|e(u_k, \beta_k)\|,
\]

\(\alpha_k = \theta_2 \alpha_k\); else \(\alpha_k = \alpha_k\). Set \(k := k + 1\), and go to Step 1.

**Remark 3.1.** Note that if the self-adaptive parameter \(\alpha_k \equiv 1\), then the ascent direction \(d(u_k, \beta_k)\) defined by (10) is reduced to the profitable direction proposed in \([9, 19, 20]\). The more details are referred to \([9, 19, 20]\) and references cited therein.

**Remark 3.2.** The main purpose of introducing two different parameters \(\theta_1\) and \(\theta_2\) is to accelerate the convergence at each iteration with a larger initial parameter \(\beta_k\). Certainly, \(\theta_1\) and \(\theta_2\) can be equivalent, but the computational experience in \([6]\) demonstrates that the self-adaptive methods perform well with different \(\theta_1\) and \(\theta_2\), and we preserve this strategy in this paper.

**Lemma 3.1.** (\([6, \text{Lemma 3.2}]\)) If \(\|e(x, 1)\| \neq 0\), then there exist \(0 < \eta < 1\) and \(\hat{\alpha} > 0\), such that for all \(0 < \alpha < \hat{\alpha}\)

\[
\beta \|F(u) - F(u - \alpha e(u, \beta))\| \leq \eta \|e(u, \beta)\|.  
\] (12)

As pointed out in \([6]\), the sequence \(\{\alpha_k\}\) generated by Algorithm 3.1 is bounded away from zero; that is

\[
\alpha_k \geq \alpha_{\min} := \min\{\alpha_{-1}, l \hat{\alpha}\} > 0.
\] (13)

The following theorem means that for any \(\alpha\) satisfying (12), \(d(u, \beta)\) defined by (10) is a profitable direction of \(\frac{1}{2} \|u - w^*\|^2\), where \(w^* \in \Omega^*\).
Theorem 3.2. If the parameter $\alpha$ satisfies (12) and $\beta > 0$, then for any $u^* \in \Omega^*$ and $u \neq u^*$,

$$(u - u^*)^T d(u, \beta) \geq e(u, \beta)^T g(u, \beta) \geq \alpha (1 - \eta) ||e(u, \beta)||^2 > 0,$$

and

$$||d(u, \beta)|| \neq 0.$$

Proof. From the definition of variational inequality, for any $v \in \Omega$, it follows that

$$F(u^*)^T (v - u^*) \geq 0.$$

Combining the above inequality and the monotonicity of $F$, we have

$$F(v)^T (v - u^*) \geq 0. \tag{14}$$

Setting $v := u - \alpha e(u, \beta)$ in (14), we obtain

$$\beta F(u - \alpha e(u, \beta))^T (u - u^* - \alpha e(u, \beta)) \geq 0. \tag{15}$$

On the other hand, setting $v := u - \beta F(u)$ and $w := u^*$ in (5), we get the following inequality

$$\alpha (e(u, \beta) - \beta F(u))^T (u - u^* - e(u, \beta)) \geq 0. \tag{16}$$

Adding (15) and (16), we get

$$(u - u^*)^T d(u, \beta) \geq e(u, \beta)^T g(u, \beta). \tag{17}$$

By using (12) and (17), we finally conclude that

$$(u - u^*)^T d(u, \beta) \geq e(u, \beta)^T g(u, \beta) \geq \alpha (1 - \eta) ||e(u, \beta)||^2 > 0.$$

According to the assumption and Cauchy-Schwarz inequality, $||d(u, \beta)|| \neq 0$ can be obtained directly. The proof is completed. \hfill \Box

Based on the above properties, we analyze the convergence of Algorithm 3.1 in the rest of this section. First, we present the bounded property of the generated sequence $\{u_k\}$ in the following theorem.

Theorem 3.3. Let $\{u_k\} \subset \mathbb{R}^n$ be the sequence generated by Algorithm 3.1, then

$$||u_{k+1} - u^*||^2 \leq ||u_k - u^*||^2 - \alpha^2 (1 - \eta)^2 \gamma (2 - \gamma) ||e(u_k, \beta_k)||^4 / ||d(u_k, \beta_k)||^2 \tag{18}$$

and the sequence $\{u_k\}$ is bounded.
Since the underlying mapping is continuous, so is \( d(u_k, \beta_k) \). Therefore, it follows from the boundedness of \( \{u_k\} \) and (21) that \( \|d(u_k, \beta_k)\| \) is also bounded. Then it follows from (20) that
\[
\lim_{k \to +\infty} \|e(u_k, \beta_k)\| = 0. 
\]
From (13), we get \( \beta_k \geq \beta_{\min} \equiv \min\{1, \theta_1 \alpha_{\min}\} > 0, \forall k > 0 \). It follows from Lemma 2.2 that
\[
\lim_{k \to +\infty} \|e(u_k, \beta_{\min})\| = 0. 
\]
Since \( \{u_k\} \) is bounded, it has at least one cluster point. Let \( \bar{u} \) be a cluster point of \( \{u_k\} \) and \( \{u_k\}_{k_j \in \mathcal{N}} \) be the corresponding subsequence converging to \( \bar{u} \), where \( \mathcal{N} \subseteq \{0, 1, \cdots\} \). Then,
\[
\|e(\bar{u}, \beta_{\min})\| = \lim_{k_j \to +\infty} \|e(u_{k_j}, \beta_{\min})\| = 0. 
\]
That is \( \bar{u} \) is a solution of VIP(F, \( \Omega \)). Setting \( u^* = \bar{u} \) in (19), we get \( \{u_k\} \) converges to a solution of VIP(F, \( \Omega \)). This completes the proof. \( \square \)
4 Numerical Example

In this section, we report some numerical experiments of two examples and present comparisons between the proposed algorithm and Han and Lo’s Algorithm 3.2 in [6], denoted by SPVI and HLM for short respectively. All the codes were written in MATLAB and run on a HP personal computer with Intel Pentium Dual-Core processor 2.6GHz, 2GB memory.

Example 4.1. The first experimental problem that we considered is a nonlinear complementarity problem as follows:

\[ u \geq 0, \quad F(u) \geq 0, \quad u^T F(u) = 0, \]

where

\[ F(u) = Mu + D(u) + q, \]

\( Mu+q \) and \( D(u) \) are the linear part and the nonlinear part of \( F(u) \), respectively. We construct the test problems as similar as in He et al. [11]. The matrix \( M \) of the linear part is \( M = A^T A + B \), where \( A \) is an \( n \times n \) matrix whose entries are randomly generated in the interval \((-5, 5)\) and the skew-symmetric matrix \( B \) is generated in the same way. The vector \( q \) is generated from a uniform distribution in the interval \((-500, 0)\). In practice, the case of \( q \in (-500, 500) \) is easier to solve than the above case. The components of nonlinear part \( D(u) \) are \( D_j(u) = d_j \times \arctan(u_j - 2) \), where \( d_j \) is a random variable in \((0, 1)\). We test the problems with dimension \( n = 10, 50, 100, 200, 500 \) and 800, and the corresponding numerical results are reported in Table 1.

Example 4.2. This example is a modification of the Example 6.1 discussed in [20]. The problem is a linear complementarity problem, i.e.,

\[ F(u) = Mu + q, \]

where \( M \) is a tridiagonal matrix as follows:

\[ M = \begin{pmatrix} 4 & -2 \\ 1 & 4 & \ddots \\ & \ddots & \ddots & -2 \\ & & 1 & 4 \end{pmatrix}, \]

and the vector \( q \) is randomly generated in the interval \([-1, 0]\). We test the problems with dimension \( n = 50, 100, 200, 500, 1000 \) and 2000, and the corresponding numerical results are reported in Table 2.

In all our tests, we set the stopping criterion utilized in the test as

\[ \|e(u_k, \beta_k)\| \leq 10^{-6}. \]

The values of some parameters in the Han and Lo’s method are specified as \( \mu = 0.5, \ L = 0.8, \ \theta_1 = 2.9 \) and \( \theta_2 = 2.0 \), while these parameters are specified
as $\eta = 0.5$, $l = 0.9$, $\theta_1 = 3.1$ and $\theta_2 = 2.5$ in the proposed algorithm. And the rest parameters in the two algorithms are set as $\gamma = 1.9$, and $\alpha_{-1} = 1$.

Because of the test problems are generated randomly, all the number of iteration and CPU time are the average of 20 trials. In Table 1, we present four groups of numerical results for different starting points. Subtable (a) and (b) are corresponding to the starting point $u_0 = (0, 0, \cdots, 0)^T$ and $u_0 = (1, 1, \cdots, 1)^T$ respectively, and (c) and (d) are corresponding to two different starting vectors, which are randomly generated in the interval $(0, 1)$, respectively. Differ with Table 1(c) and (d), Table 2(b) is obtained by different initial points randomly generated in each trial, that is 20 trials with 20 different starting points. Note that in our implementation of the algorithms, if the $i$-th component of the vector $u_k$ and $d(u_k, \beta_k)$ satisfies $|u_k|_i < 10^{-6}$ and $|d(u_k, \beta_k)|_i > 10^{-4}$ respectively, we set $|d(u_k, \beta_k)|_i = 0$. In addition, we can see that the proposed direction is the combination of $-F(u_k)$ and the direction of Han and Lo’s method. So in our practical computation, if the cosine between Han and Lo’s direction and $-F(u_k)$ is greater than 0.999999, we shrink the stepsize in the direction $-F(u_k)$, i.e. $\alpha_{k+1} = 0.7\alpha_k$, otherwise we set the proposed direction as Han and Lo’s direction.

From the Table 1 and Table 2, we can see that the average iteration and CPU time of the proposed algorithm is less than Han and Lo’s method. The numerical results demonstrate the new method is efficient and reliable.

5 Conclusions

In this paper, we construct a new search direction and then present an improved self-adaptive projection method for solving variational inequalities. Furthermore, we analyze the global convergence under the mild conditions that the underlying mapping $F$ is continuous and monotone. Some preliminary numerical results demonstrate the new method is efficient and reliable.

Acknowledgement: The authors thank the referee for the constructive comments on the first version of the paper, which led to great improvements of the paper. The research was supported by the NSFC grants 11071122, 10871098, and NSF of Jiangsu Province at Grant No. BK2009397.

References

An Improved Self-Adaptive Projection Method for Solving VIs


Table 1: Comparison of (HLM) and (SPVI) for Example 4.1.

(a) Starting point $u_0 = (0, 0, \cdots, 0)^T$.

<table>
<thead>
<tr>
<th>Dim.</th>
<th>HLM</th>
<th>SPVI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iter.</td>
<td>CPU(s)</td>
</tr>
<tr>
<td>n=10</td>
<td>257.40</td>
<td>0.0062</td>
</tr>
<tr>
<td>n=50</td>
<td>291.05</td>
<td>0.0117</td>
</tr>
<tr>
<td>n=100</td>
<td>265.20</td>
<td>0.0182</td>
</tr>
<tr>
<td>n=200</td>
<td>277.15</td>
<td>0.0413</td>
</tr>
<tr>
<td>n=500</td>
<td>308.60</td>
<td>0.3370</td>
</tr>
<tr>
<td>n=800</td>
<td>355.20</td>
<td>1.8850</td>
</tr>
</tbody>
</table>

(b) Starting point $u_0 = (1, 1, \cdots, 1)^T$.

<table>
<thead>
<tr>
<th>Dim.</th>
<th>HLM</th>
<th>SPVI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iter.</td>
<td>CPU(s)</td>
</tr>
<tr>
<td>n=10</td>
<td>223.95</td>
<td>0.0055</td>
</tr>
<tr>
<td>n=50</td>
<td>317.70</td>
<td>0.0129</td>
</tr>
<tr>
<td>n=100</td>
<td>273.30</td>
<td>0.0190</td>
</tr>
<tr>
<td>n=200</td>
<td>304.80</td>
<td>0.0454</td>
</tr>
<tr>
<td>n=500</td>
<td>504.25</td>
<td>0.5490</td>
</tr>
<tr>
<td>n=800</td>
<td>777.65</td>
<td>4.1805</td>
</tr>
</tbody>
</table>

(c) Starting point $u_0 = \text{rand}(n,1)$.

<table>
<thead>
<tr>
<th>Dim.</th>
<th>HLM</th>
<th>SPVI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iter.</td>
<td>CPU(s)</td>
</tr>
<tr>
<td>n=10</td>
<td>246.30</td>
<td>0.0060</td>
</tr>
<tr>
<td>n=50</td>
<td>265.40</td>
<td>0.0108</td>
</tr>
<tr>
<td>n=100</td>
<td>264.15</td>
<td>0.0185</td>
</tr>
<tr>
<td>n=200</td>
<td>313.10</td>
<td>0.0471</td>
</tr>
<tr>
<td>n=500</td>
<td>306.15</td>
<td>0.3471</td>
</tr>
<tr>
<td>n=800</td>
<td>340.60</td>
<td>1.8972</td>
</tr>
</tbody>
</table>

(d) Starting point $u_0 = \text{rand}(n,1)$.

<table>
<thead>
<tr>
<th>Dim.</th>
<th>HLM</th>
<th>SPVI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iter.</td>
<td>CPU(s)</td>
</tr>
<tr>
<td>n=10</td>
<td>203.25</td>
<td>0.0051</td>
</tr>
<tr>
<td>n=50</td>
<td>237.55</td>
<td>0.0098</td>
</tr>
<tr>
<td>n=100</td>
<td>279.85</td>
<td>0.0196</td>
</tr>
<tr>
<td>n=200</td>
<td>297.70</td>
<td>0.0450</td>
</tr>
<tr>
<td>n=500</td>
<td>297.20</td>
<td>0.3393</td>
</tr>
<tr>
<td>n=800</td>
<td>342.40</td>
<td>1.8826</td>
</tr>
</tbody>
</table>
Table 2: Comparison of (HLM) and (SPVI) for Example 4.2.

(a) Starting point $u_0 = (1, 1, \ldots, 1)^T$.

<table>
<thead>
<tr>
<th>Dim.</th>
<th>HLM</th>
<th>SPVI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iter.</td>
<td>CPU(s)</td>
</tr>
<tr>
<td>n= 50</td>
<td>24.60</td>
<td>0.0010</td>
</tr>
<tr>
<td>n= 100</td>
<td>25.40</td>
<td>0.0016</td>
</tr>
<tr>
<td>n= 200</td>
<td>25.60</td>
<td>0.0037</td>
</tr>
<tr>
<td>n= 500</td>
<td>26.40</td>
<td>0.0335</td>
</tr>
<tr>
<td>n= 1000</td>
<td>26.75</td>
<td>0.2285</td>
</tr>
<tr>
<td>n= 2000</td>
<td>27.80</td>
<td>0.9024</td>
</tr>
</tbody>
</table>

(b) Starting point $u_0^{\text{th}} = \text{rand}(n, 1)^T$.

<table>
<thead>
<tr>
<th>Dim.</th>
<th>HLM</th>
<th>SPVI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iter.</td>
<td>CPU(s)</td>
</tr>
<tr>
<td>n= 50</td>
<td>21.80</td>
<td>0.0009</td>
</tr>
<tr>
<td>n= 100</td>
<td>21.90</td>
<td>0.0015</td>
</tr>
<tr>
<td>n= 200</td>
<td>23.20</td>
<td>0.0035</td>
</tr>
<tr>
<td>n= 500</td>
<td>24.50</td>
<td>0.0342</td>
</tr>
<tr>
<td>n= 1000</td>
<td>24.40</td>
<td>0.2274</td>
</tr>
<tr>
<td>n= 2000</td>
<td>24.75</td>
<td>0.8734</td>
</tr>
</tbody>
</table>