(To appear in *Journal of Computational Finance*)

An MPEC Approach to Inverse Pricing of American Options: The Case of an Implied Volatility Surface

Jacqueline Huang and Jong-Shi Pang
Department of Mathematical Sciences
The Johns Hopkins University
Baltimore, Maryland 21218-2682, U.S.A.
Emails: jhuang@mts.jhu.edu and jsp@vicp1.mts.jhu.edu.

October 13, 1999

Abstract

This paper presents a novel approach to deal with the computation of an implied volatility surface of American options written on a risky asset. The approach is based on the simple observation that this computational problem is the inverse of the forward pricing problem of American options. As detailed in [17], the latter forward problem can be modeled by a discretized partial differential linear complementarity system. As such, the inverse problem, i.e., the implied volatility problem, becomes an instance of a **M**athematical **P**rogram with **E**quilibrium **C**onstraints, which is a class of constrained optimization problem with a finite-dimensional parametric linear complementarity system as part of its constraints. Two methods for solving an MPEC are described and applied to the problem of computing an implied volatility surface of American options. Some computational results on experimental data are reported.

1 Introduction

The Black-Scholes analysis of option pricing is well known. An important input to this analysis is the volatility parameter of the underlying risky asset on which options of various kinds are written. In their pioneering work [5], Black and Scholes assumed this parameter to be a constant and derived their famous formula for the theoretical price of a vanilla European call option:

$$C(S,t) \;=\; SN(d_1) - Ee^{-r(T-t)}N(d_2), \tag{1}$$

where the notation is fairly standard; see [29] for instance. The two constants d_1 and d_2 contain the volatility parameter σ. As it is well known, this parameter is in general not a constant; indeed it is a highly complicated function of several deterministic and random factors. Previous approaches for dealing with this difficult problem of unknown volatility are plenty and include (i) statistical estimation methods based on historical data (see e.g. [18, Section 10.4] and [16, Section 1.E]), (ii) mathematical models of stochastic volatilities (such as those in [19, 21, 31]), and (iii) implied volatilities based on observed option prices (suggested originally by Latané and Rendleman [24] and empirically tested by Beckers [4]).

In the case of American options, that is, options with an early exercise feature, the above formula (1) and its analogs are no longer valid. In fact, as shown in [20], a rigorous mathematical model for pricing an American option is an infinite-dimensional free boundary problem. As such, there is in general no explicit formula or finite procedure for computing the exact price of an American option. Upon a suitable discretization of the partial differential operator defining the free boundary problem, we obtain a finite-dimensional variational inequality that is in turn equivalent to a linear complementarity problem (LCP). With the discretization parameters properly chosen, the solution of the LCP can provide an arbitrarily close approximation to the exact American option prices. Expanding the preliminary work done by various authors including [29, 11, 12], the paper [17] presents an in-depth treatment of pricing American options of various kinds by the LCP approach and reports extensive computational results that support the effectiveness of this approach. A related paper is [28].

The problem of computing an implied volatility surface of (European or American) options is an instance of an *inverse* problem that is the counterpart of the *forward* problem of pricing these options; see [7]. Specifically, in the forward option pricing problem, a constant volatility parameter (along with other constants, such as the interest rate and asset dividend) is taken as an input to a mathematical model that produces the (theoretical) option prices. Ideally, it is most desirable for the computed option prices to agree with the option prices observed on the trading floors. Unfortunately, this is hardly the case in practice. One possible factor that causes the discrepancy in the computed and observed option prices is that the volatility constant is being used incorrectly. Borrowing a well-known idea from applied mathematics, we attempt to build an inverse model to infer the volatility parameter that takes into account the observed values of the unknown option prices. Thus, in the inverse model, the asset volatility and the option prices are both unknowns to be determined.

The main objective of this paper is to present the MPEC approach to the implied volatility problem of American options, where MPEC stands for **M**athematical **P**rogram with **E**quilibrium **C**onstraints [25]. A brief general discussion of this class of constrained optimization problem is presented in Section 3. For now, we attempt to motivate the need for such an advanced optimization methodology by briefly reviewing a common approach for computing implied volatility of European options. Basically, in the latter approach, one equates the theoretical option value predicted by the Black-Scholes formula with an observed value and numerically solves for the unknown volatility parameter in the resulting equation, with the remaining parameters fixed. This simplistic approach has several drawbacks and deficiencies. From a mathematical point of view, the approach lacks a sound basis due to the following reasons.

A. Although the Black-Scholes partial differential equation remains valid when the volatility is a function of asset price and time, the explicit formula (1) for the (call) option value is derived under the assumption that volatility is a fixed constant. Indeed, as detailed in Exercise 5 in Chapter 5 of [29], when the volatility is a known function of time, the actual formula for the option value is much more complicated than (1). Thus using (1) as the basis for the computation of volatilities does not seem mathematically sound unless one accepts the premise that volatility is a constant independent of asset price and option duration. Unfortunately, studies have shown that this premise is not consistent with observations. Indeed, volatility is highly dependent on the maturity and the strike of the option (known as the volatility smile)[15].

B. Even if one accepts an explicit formula such as (1) for an option price, there are still several practical deficiencies associated with solving the equation

$$V_{\mathrm{BS}}(S,t) \;=\; V_{\mathrm{obs}} \tag{2}$$

to compute the implied volatility, where the left-hand quantity is the theoretically computed option price and the right-hand quantity is the observed option value corresponding to the asset price S and time t. First, this equation is not guaranteed to have a solution in the unknown volatility σ. We refer the reader to the paper [7] where a uniqueness result for a local volatility function was established using the theory of inverse problems in partial differential equations. Second, even if an (approximate) solution of the equation can be obtained (usually by an iterative scheme such as the Newton-Raphson method), the computed value of σ may not be useful for predictive purposes because it is not practically meaningful. Third, solving one single equation will produce only one value of the volatility $\sigma(S,t)$. In order to obtain the entire *volatility surface*, we need to solve such an equation for all pairs (S,t) of interest. Fourth, the numerical solution of (2), which is a high complex nonlinear equation, could be very time consuming for the calibration of the entire volatility surface, especially if many equations of this kind have to be solved repeatedly that correspond to a portfolio of different observed option values.

The equation approach to computing implied volatility lacks the flexibility in handling restrictions that one may wish to impose on this quantity. For instance, in addition to (approximately) satisfying (2), one may want the volatility parameter (perhaps after a suitable aggregation or weighing) to lie within a certain range. Restrictions like this one often appear in the form of inequalities that are not captured by the single equation (2). The advantage of imposing additional restrictions is clear: for one thing, they allow us to exclude quantities that are deemed not suitable for practical use. These considerations are important factors for the use of an optimization approach to computing implied volatilities.

As mentioned above and noted by various authors (see references cited below), the implied volatility problem is essentially an inverse problem. We adopt an optimization framework by introducing an objective function that captures the goal of the inverse problem, which is to compute the unknown volatilities and option prices according to a prescribed minimization principle. The optimization approach has the additional advantage in that constraints on the unknowns can easily be included and dealt with. The idea of using an optimization approach to deal with uncertain volatilities is not new. Within the context of the binomial tree method, Rubinstein [27] proposed an optimization problem as a way to calculate implied posterior risk-neutral probabilities. Avellaneda, Friedman, Holmes and Sampeir [2] and Avellaneda [1] introduced a relative entropy minimization problem to calibrate volatility surfaces for European options. Lagnado and Osher [22, 23] used a gradient descent method to solve a nonlinear optimization problem in which the partial differential equation governing the European option price is part of the constraint. Most recently, Coleman, Li, and Verma [9] used a 2-D spline approximation coupled with finite-dimensional constrained nonlinear optimization to reconstruct a smooth local volatility surface. All these prior works deal with volatilities implied by European options. Expanding on the work of Rubinstein [27] and Derman and Kani [14], two papers [3, 8] discuss heuristic tree methods for evaluating implied volatilities of European and American options. No optimization is attempted in the latter two papers. The recent paper [6] discusses an optimization-based inverse problem for the nonparametric estimation of an implied volatility surface of European options.

Unlike the previous works, we proposed an optimization approach to directly deal with the problem of computing an implied volatility surface by constraining the computed option values to be of the American type. In turn, the basis of this approach is a discretized model for American options; namely, the LCP. We do not treat European options in this paper because the resulting (discretized) model is a straightforward optimization problem that can be solved by well-known methods. In contrast, the resulting optimization problem derived from the inverse problem of American options is an unusual optimization problem that needs advanced treatment, as we will see subsequently.

2 The Optimization Formulation

This section presents the formulation of the computational problem of an implied volatility surface of American options as an MPEC. We begin with the review of the forward pricing problem of an American option formulated as a discretized linear complementarity problem (LCP). Although this formulation is well explained in several references (e.g. [17]), it is useful for us to repeat the derivation so that we can fix the notation and set up the basic framework. Moreover, the review will facilitate the discussion of multiple options and the promised presentation of the MPEC. Throughout this paper, we treat only vanilla American options for simplicity. The same general approach can be applied to computing volatilities implied by exotic American options and/or American options with transaction costs as well as the inverse pricing of other financial derivatives.

2.1 Forward option pricing

The basic framework is that of Black and Scholes [5]; that is, the price of a risky asset is assumed to satisfy the following stochastic differential equation:

$$dS \;=\; (\mu - D_0)\, S\, dt + \sigma(S,t)\, S\, dW$$

where S denotes the asset price that is a function of the time $t \in [0,T]$, with $T > 0$ being the duration of a vanilla European option written on the asset, μ is the drift of the stochastic price process of the asset, D_0 is the constant dividend rate of the asset, W is a standard Wiener process, and $\sigma(S,t)$ is the volatility that we take to be an unknown function of the pair (S,t), which defines the volatility surface. (In Black-Scholes original analysis, this function is a constant.) Our goal is compute a discretization of the volatility surface so that a prescribed objective function is minimized subject to given constraints on the unknown volatilities and option prices. Examples of such an objective function and constraints will be presented later.

It is known that the value $V(S,t)$ of an American option must satisfy the following partial differential linear complementarity problem (PDLCP): for t in $[0,T)$ and S in $[0,\infty)$,

$$\begin{aligned} 0 &\leq V(S,t) - \Lambda(S,t), \\ 0 &\geq \mathcal{L}_{\rm BS}(V), \\ 0 &= [\, V(S,t) - \Lambda(S,t)\,]\, \mathcal{L}_{\rm BS}(V), \end{aligned} \tag{3}$$

where

$$\mathcal{L}_{\rm BS} \;\equiv\; \frac{\partial}{\partial t} + \tfrac{1}{2}\,\sigma^2\, S^2\, \frac{\partial^2}{\partial S^2} + (\, r - D_0\,)\, S \frac{\partial}{\partial S} - r,$$

is the Black-Scholes partial differential operator, with r being the constant interest rate of a risk-free asset and $\Lambda(S,t)$ being the payoff function when early exercise occurs. To complete the description of the above model, there must be given boundary values of $V(S,t)$ at $S = 0$ and $S = \infty$ and terminal values of $V(S,t)$ at $t = T$, the expiration time.

Since no explicit formula exists for the solution to the above PDLCP, we resort to a finite difference scheme for approximating the partial differential equation (3). Specifically, we truncate the state variable S to a finite range $[0, N\delta S]$, where N is a positive integer and $\delta S > 0$ is the step size that will be used for discretizing the partial differentiation with respect to S. Similarly, we choose a time step $\delta t > 0$ for which $M \equiv T/\delta t$ is an integer for discretizing the partial differentiation with respect to t. The following approximations are employed:

$$\begin{aligned}
\frac{\partial V}{\partial t}(S,t) &\approx \frac{V(S,t+\delta t) - V(S,t)}{\delta t}, \\
\frac{\partial^2 V}{\partial S^2}(S,t) &\approx \theta\, \frac{V(S+\delta S,t) - 2\,V(S,t) + V(S-\delta S,t)}{(\delta S)^2} + \\
&\qquad (\,1-\theta\,)\, \frac{V(S+\delta S,t+\delta t) - 2\,V(S,t+\delta t) + V(S-\delta S,t+\delta t)}{(\delta S)^2}, \\
\frac{\partial V}{\partial S}(S,t) &\approx \theta\, \frac{V(S+\delta S,t) - V(S-\delta S,t)}{2\,\delta S} + \\
&\qquad (\,1-\theta\,)\, \frac{V(S+\delta S,t+\delta t) - V(S-\delta S,t+\delta t)}{2\,\delta S},
\end{aligned}$$

where $\theta \in [0,1]$ is a given parameter whose specializations yield the explicit approximation ($\theta = 0$), the implicit approximation ($\theta = 1$), and the Crank-Nicolson approximation ($\theta = 1/2$). Based on the above finite difference scheme and the initial and boundary values of $V(S,t)$, we wish to compute the unknown option and volatility values at the grid points $(n\delta S, m\delta t)$, for $n = 1, 2, \ldots, N-1$ and $m = 0, 1, 2, \ldots M-1$, in the (state, time)-product space. Let

$$V_{mn} \equiv V(n\delta S, m\delta t) \quad \text{and} \quad \sigma_{mn} \equiv \sigma(n\delta S, m\delta t)$$

denote these unknown values. The boundary values

$$V_{m0} \equiv V(0, m\delta t), \quad V_{mN} \equiv V(N\delta S, m\delta t), \quad V_{Mn} \equiv V(n\delta S, T),$$

for $m = 0, \ldots, M-1$ and $n = 1, \ldots, N-1$ are all given. We also let

$$\Lambda_{mn} \equiv \Lambda(n\delta S, m\delta t)$$

denote the payoff function evaluated at the grid point $(n\delta S, m\delta t)$. For each m, let $\boldsymbol{V}_m$, $\boldsymbol{\sigma}_m$ and $\boldsymbol{\Lambda}_m$ denote, respectively, the $(N-1)$-vectors $(V_{mn})_{n=1}^{N-1}$, $(\sigma_{mn})_{n=1}^{N-1}$ and $(\Lambda_{mn})_{n=1}^{N-1}$. Along with suitable boundary conditions, the PDLCP (3) is approximated by the following $(M-1)$ LCPs, each of order $(N-1)$: at each time $t = m\delta t$ for $m = M-1, M-2, \ldots, 1, 0$,

$$0 \le \boldsymbol{V}_m - \boldsymbol{\Lambda}_m \;\perp\; \boldsymbol{b}_m(\boldsymbol{\sigma}_m) + \boldsymbol{Q}(\boldsymbol{\sigma}_m)\,\boldsymbol{V}_m + \boldsymbol{N}(\boldsymbol{\sigma}_m)\,\boldsymbol{V}_{m+1} \ge 0 \tag{4}$$

where $\perp$ is the notation for "perpendicular to" and for an arbitrary $(N-1)$-vector $\boldsymbol{\omega}$,

$$\boldsymbol{Q}(\boldsymbol{\omega}) \equiv (\,\delta t^{-1} + r\,)\,\boldsymbol{I}_{N-1} + \boldsymbol{L}_{\theta}(\boldsymbol{\omega}), \quad \boldsymbol{N}(\boldsymbol{\sigma}) \equiv -\delta t^{-1}\,\boldsymbol{I}_{N-1} + \boldsymbol{L}_{1-\theta}(\boldsymbol{\omega}),$$

$$\boldsymbol{b}_m(\boldsymbol{\omega}) \equiv \tfrac{1}{2} \begin{pmatrix} \tfrac{1}{2}\,[\,\theta\,V_{m0} + (\,1-\theta\,)V_{(m+1)0}\,]\,(\,r - D_0 - \omega_1\,)\,\omega_1 \\ 0 \\ \vdots \\ 0 \\ -\tfrac{1}{2}\,[\,\theta\,V_{mN} + (\,1-\theta\,)V_{(m+1)N}\,]\,(\,r - D_0 + \omega_{N-1}\,)\,\omega_{N-1}, \end{pmatrix}$$

with $\boldsymbol{I}_{N-1}$ being the identity matrix of order $N-1$ and $\boldsymbol{L}_{\alpha}(\boldsymbol{\omega})$ being the $(N-1)\times(N-1)$ tridiagonal matrix whose entries are given by: for $i,j = 1,2,\ldots,N-1$,

$$(\,\boldsymbol{L}_{\alpha}(\boldsymbol{\omega})\,)_{ij} \equiv \begin{cases} -\dfrac{\alpha}{2}\,i^2\,\omega_i^2 + \dfrac{\alpha}{2}\,i\,(\,r - D_0\,) & \text{if } j = i-1 \\ \alpha\,i^2\,\omega_i^2 & \text{if } j = i \\ -\dfrac{\alpha}{2}\,i^2\,\omega_i^2 - \dfrac{\alpha}{2}\,i\,(\,r - D_0\,) & \text{if } j = i+1 \\ 0 & \text{otherwise.} \end{cases}$$

Provided that

$$\frac{1}{\delta t} + r \;\geq\; N\,|\,r - D_0\,|, \tag{5}$$

the matrix $\boldsymbol{L}_{\alpha}(\boldsymbol{\omega})$, and hence $\boldsymbol{Q}(\boldsymbol{\sigma}_m)$, is strictly diagonally dominant, thus positive definite, for all vectors $\boldsymbol{\omega}$ and scalars $\alpha \in [0,1]$. Throughout the rest of this paper, we assume the condition (5) on δt is satisfied.

To solve the forward pricing problem where the volatilities σ_{mn} are all given, we time step the LCPs (4), starting with $m = M-1$; since $\boldsymbol{V}_M$ is known, by solving the LCP at time $t = (M-1)\delta t$, we can obtain a unique solution $\boldsymbol{V}_{M-1}$. Proceeding backward in time, we can compute a set of discrete option prices V_{mn} that depends on the input volatilities σ_{mn}. In order to understand the details of this dependence, it would be convenient to write the M LCPs (4) as an aggregate LCP of size $M(N-1)$. Indeed, define the $M(N-1)\times M(N-1)$ matrix:

$$\boldsymbol{A}(\boldsymbol{\sigma}) \equiv \begin{bmatrix} \boldsymbol{Q}(\boldsymbol{\sigma}_0) & \boldsymbol{N}(\boldsymbol{\sigma}_0) & & & \\ & \boldsymbol{Q}(\boldsymbol{\sigma}_1) & \boldsymbol{N}(\boldsymbol{\sigma}_1) & & \\ & & \ddots & \ddots & \\ & & & \boldsymbol{Q}(\boldsymbol{\sigma}_{M-2}) & \boldsymbol{N}(\boldsymbol{\sigma}_{M-2}) \\ & & & & \boldsymbol{Q}(\boldsymbol{\sigma}_{M-1}) \end{bmatrix}$$

and the $M(N-1)$-vector

$$\boldsymbol{b}(\boldsymbol{\sigma}) \equiv \begin{bmatrix} \boldsymbol{b}_1(\boldsymbol{\sigma}_1) \\ \vdots \\ \boldsymbol{b}_M(\boldsymbol{\sigma}_M) \end{bmatrix}$$

where $\boldsymbol{\sigma}$ is the $M(N-1)$-vector whose entries are the discretized volatilities σ_{mn} for $m = 0, \ldots, M-1$ and $n = 1, \ldots, N-1$:

$$\boldsymbol{\sigma} \equiv \begin{pmatrix} \boldsymbol{\sigma}_0 \\ \vdots \\ \boldsymbol{\sigma}_{M-1} \end{pmatrix}.$$

Similarly, we define the aggregate $M(N-1)$-vectors of option prices and payoffs:

$$\boldsymbol{V} \equiv \begin{pmatrix} \boldsymbol{V}_0 \\ \vdots \\ \boldsymbol{V}_{M-1} \end{pmatrix} \quad \text{and} \quad \boldsymbol{\Lambda} \equiv \begin{pmatrix} \boldsymbol{\Lambda}_0 \\ \vdots \\ \boldsymbol{\Lambda}_{M-1} \end{pmatrix}.$$

The above time-stepping scheme for computing the forward prices of a vanilla American option can be summarized as the following LCP of size $M(N-1)$:

$$0 \leq \boldsymbol{V} - \boldsymbol{\Lambda} \perp \boldsymbol{b}(\boldsymbol{\sigma}) + \boldsymbol{A}(\boldsymbol{\sigma})\,\boldsymbol{V} \geq 0. \tag{6}$$

Under the condition (5), $\boldsymbol{A}(\boldsymbol{\sigma})$, being a block upper triangular matrix with positive definitive diagonal blocks, is a P-matrix, albeit $\boldsymbol{A}(\boldsymbol{\sigma})$ is not symmetric or positive definite. For a comprehensive treatment of P-matrices and the LCP, we refer the reader to the text [10]. Observe that $\boldsymbol{A}(\boldsymbol{\sigma})$ depends on the finite-difference scheme being used to discretize the Black-Scholes partial differential operator $\mathcal{L}_{\mathrm{BS}}$ and is independent of the option–i.e. the payoff function and the expiration. Presumably, if we use a different discretization of this operator, we obtain a different matrix. The general methodology presented below applies to other discretization schemes as well, provided that the resulting $\boldsymbol{A}(\boldsymbol{\sigma})$ is a P-matrix. In contrast to $\boldsymbol{A}(\boldsymbol{\sigma})$, the vectors $\boldsymbol{\Lambda}$ and $\boldsymbol{b}(\boldsymbol{\sigma})$ are both dependent on the option.

Multiple options

In the case where there are multiple options written on the same underlying asset, the LCP (6) can easily be embedded in a larger system. Specifically, suppose that there are K American options each being characterized by its payoff function $\Lambda^k(S,t)$ at exercise. The prices of these options are all calculated under the same volatility function $\sigma(S,t)$ of the asset. Thus for each $k = 1, \ldots, K$, we solve the LCP:

$$0 \leq \boldsymbol{V}^k - \boldsymbol{\Lambda}^k \perp \boldsymbol{b}^k(\boldsymbol{\sigma}) + \boldsymbol{A}(\boldsymbol{\sigma})\,\boldsymbol{V}^k \geq 0,$$

to obtain the discretized option prices V^k_{mn} of type k. Notice that all these LCPs are defined by the same matrix $\boldsymbol{A}(\boldsymbol{\sigma})$. Concatenating these K LCPs, we arrive at the final LCP formulation for the forward pricing problem of multiple American options on a single asset:

$$0 \leq \boldsymbol{x} - \boldsymbol{p} \perp \boldsymbol{q}(\boldsymbol{\sigma}) + \boldsymbol{M}(\boldsymbol{\sigma})\,\boldsymbol{x} \geq 0, \tag{7}$$

where

$$\boldsymbol{x} \equiv (\boldsymbol{V}^k)_{k=1}^K, \quad \boldsymbol{p} \equiv (\boldsymbol{\Lambda}^k)_{k=1}^K,$$

are, respectively, the $KM(N-1)$-dimensional vectors of unknown option prices and known payoffs at the discretized grid points,

$$\boldsymbol{q}(\boldsymbol{\sigma}) \equiv (\boldsymbol{b}^k(\boldsymbol{\sigma}))_{k=1}^K,$$

is the $KM(N-1)$-dimensional vector that contains the given initial and boundary values of the options, and $\boldsymbol{M}(\boldsymbol{\sigma})$ is the $KM(N-1) \times KM(N-1)$ block diagonal matrix all of whose K diagonal blocks are equal to the $M(N-1) \times M(N-1)$ matrix $\boldsymbol{A}(\boldsymbol{\sigma})$.

In summary, we have formulated the forward option pricing problem as the LCP (7), which we call the *forward option LCP*. This LCP requires the discretized volatility matrix $\boldsymbol{\sigma}$ as an input; the output from this LCP yields the discrete option prices $V^k(n\delta S, m\delta t)$ of K options. By the P-property of the matrix $\boldsymbol{M}(\boldsymbol{\sigma})$, the forward option LCP has a unique solution $\boldsymbol{x}(\boldsymbol{\sigma})$ that depends on the input volatilities. This option function $\boldsymbol{x}(\boldsymbol{\sigma})$ is only implicitly known; for any given $\boldsymbol{\sigma}$, $\boldsymbol{x}(\boldsymbol{\sigma})$ can be evaluated by solving the forward option LCP. We will address some basic properties of this option function subsequently; see Theorem 1.

2.2 The implied volatility problem

We may now introduce the optimization problem for the joint computation of the volatility matrix $\boldsymbol{\sigma}$ and the option price vector $\boldsymbol{x}(\boldsymbol{\sigma})$, under a set of prescribed criteria that is the result of some practical market considerations. The constraints on the unknown volatilities are modeled by the set $\boldsymbol{\Gamma} \subseteq \Re^{M(N-1)}$. Examples of these constraints include upper and lower bounds on $\boldsymbol{\sigma}$ and other practical restrictions that are deemed necessary. The objective is described by the function $\theta(\boldsymbol{\sigma}, \boldsymbol{x})$. Examples of this function include a standard least-squares deviation from observed option prices and historical volatilities. More precisely, suppose that a subset of option prices $V_{mn}^{\mathrm{obs},k}$ with (k,m,n) belonging to a subset $\mathcal{O}$ of $\{1,\ldots,K\} \times \{0,\ldots,M-1\} \times \{1,\ldots,N-1\}$ and a subset of volatilities $\sigma_{mn}^{\mathrm{his}}$ with (m,n) belonging to a subset $\mathcal{S}$ of $\{0,\ldots,M-1\} \times \{1,\ldots,N-1\}$ are given (e.g. these could be the observed market prices and historical volatilities). An objective would be to seek $(\boldsymbol{\sigma}, \boldsymbol{x})$ so that the computed values would be least deviated from these given values. Mathematically, the objective function is then:

$$\theta(\boldsymbol{\sigma}, \boldsymbol{x}) \equiv \sum_{(k,m,n)\in\mathcal{O}} (V_{mn}^k - V_{mn}^{\mathrm{obs},k})^2 + \sum_{(m,n)\in\mathcal{S}} (\sigma_{mn} - \sigma_{mn}^{\mathrm{his}})^2,$$

where V_{mn}^k is the theoretical option price (collectively, these prices are the components of $\boldsymbol{x}$). Another example of θ would be a function similar to the one used in [9] that includes a measure of the smoothness of the discretized volatility surface. Variations of these functions and/or other realistic objective functions are amenable to the same general methodology.

In general, with the set $\boldsymbol{\Gamma}$ and the objective function θ given, the implied volatility problem of American options is defined as the following constrained optimization problem: compute $(\boldsymbol{\sigma}, \boldsymbol{x})$ to

$$\begin{array}{ll} \text{minimize} & \theta(\boldsymbol{\sigma}, \boldsymbol{x}) \\ \text{subject to} & \boldsymbol{\sigma} \in \boldsymbol{\Gamma} \\ \text{and} & 0 \le \boldsymbol{x} - \boldsymbol{p} \perp \boldsymbol{q}(\boldsymbol{\sigma}) + \boldsymbol{M}(\boldsymbol{\sigma})\,\boldsymbol{x} \ge 0. \end{array} \tag{8}$$

In terms of the option function $\boldsymbol{x}(\boldsymbol{\sigma})$, we may rewrite this optimization problem as an *implicit program* in the variable $\boldsymbol{\sigma}$ alone:

$$\begin{array}{ll} \text{minimize} & \varphi(\boldsymbol{\sigma}) \equiv \theta(\boldsymbol{x}(\boldsymbol{\sigma}), \boldsymbol{\sigma}) \\ \text{subject to} & \boldsymbol{\sigma} \in \boldsymbol{\Gamma}. \end{array} \tag{9}$$

The rest of the paper is devoted to the numerical solution of these two equivalent optimization problems.

3 Mathematical Programs with Equilibrium Constraints

The optimization problem (8) is an instance of an MPEC. A comprehensive study of this class of constrained optimization problems, including a historical account and an extensive bibliography, is documented in the monograph [25]. There are various major reasons why an MPEC cannot be treated as a standard optimization problem. We briefly mention several of these reasons and refer the reader to the cited reference for details. Although the constraints of an MPEC are in the form of equations and inequalities (just like those in a nonlinear program), the presence of the complementarity constraint (cf. the second constraint in (8)) invalidates the well-known Karush-Kuhn-Tucker (KKT) optimality theory in traditional nonlinear programming (NLP). In fact, KKT multipliers do not generally exist for an MPEC. The complementarity condition introduces a disjunction into the constraints of an MPEC. This is another important feature of the MPEC that distinguishes it from being a smooth nonlinear optimization problem. As we shall see from the results in the next subsection, the implicit option function $\boldsymbol{x}(\boldsymbol{\sigma})$ is a nonsmooth function of its argument; hence so is the objective function $\varphi(\boldsymbol{\sigma})$ in the equivalent formulation (9). The disjunctive and nonsmooth nature of the MPEC necessitates the development of a new theory and methodology for treating this class of optimization problems. Such a development is the contribution of the monograph [25].

3.1 Properties of the implicit option function

Clearly, understanding the properties of the implicit option function $\boldsymbol{x}(\boldsymbol{\sigma})$ is essential for the design of efficient numerical methods for solving the implied volatility problem. In essence, $\boldsymbol{x}(\boldsymbol{\sigma})$ is the solution function of a family of LCPs parameterized by the discretized volatilities $\boldsymbol{\sigma}$. As such, known results from parametric LCP theory can be applied; see [10, Chapter 7]. In order to present the key result for our use, we first introduce some notation.

To begin, it would be useful to consider $\boldsymbol{\sigma}$ and $\boldsymbol{x}$ as vectors in the Euclidean spaces $\Re^{M(N-1)}$ and $\Re^{KM(N-1)}$, respectively. Let

$$\boldsymbol{F}(\boldsymbol{\sigma}, \boldsymbol{x}) \equiv \boldsymbol{q}(\boldsymbol{\sigma}) + \boldsymbol{M}(\boldsymbol{\sigma})\boldsymbol{x}$$

be the function defining the option LCP (7). The partial Jacobian matrix of $\boldsymbol{F}(\boldsymbol{\sigma}, \boldsymbol{x})$ with respect to $\boldsymbol{\sigma}$ is denoted $J_\sigma \boldsymbol{F}(\boldsymbol{\sigma}, \boldsymbol{x})$; since $\boldsymbol{F}(\boldsymbol{\sigma}, \boldsymbol{x})$ is linear in $\boldsymbol{x}$, the partial Jacobian matrix of $\boldsymbol{F}(\boldsymbol{\sigma}, \boldsymbol{x})$ with respect to $\boldsymbol{x}$ is equal to the matrix $\boldsymbol{M}(\boldsymbol{\sigma})$. For an arbitrary $\boldsymbol{\sigma}$, we define three index sets associated with the solution $\boldsymbol{x}(\boldsymbol{\sigma})$:

$$\begin{aligned} \alpha(\boldsymbol{\sigma}) &\equiv \{\, i : (\, \boldsymbol{x}(\boldsymbol{\sigma}) - \boldsymbol{p} \,)_i > 0 = (\, \boldsymbol{q}(\boldsymbol{\sigma}) + \boldsymbol{M}(\boldsymbol{\sigma})\boldsymbol{x} \,)_i \,\} \\ \beta(\boldsymbol{\sigma}) &\equiv \{\, i : (\, \boldsymbol{x}(\boldsymbol{\sigma}) - \boldsymbol{p} \,)_i = 0 = (\, \boldsymbol{q}(\boldsymbol{\sigma}) + \boldsymbol{M}(\boldsymbol{\sigma})\boldsymbol{x} \,)_i \,\} \\ \gamma(\boldsymbol{\sigma}) &\equiv \{\, i : (\, \boldsymbol{x}(\boldsymbol{\sigma}) - \boldsymbol{p} \,)_i = 0 < (\, \boldsymbol{q}(\boldsymbol{\sigma}) + \boldsymbol{M}(\boldsymbol{\sigma})\boldsymbol{x} \,)_i \,\}. \end{aligned}$$

These index sets play an important role in the local properties of $\boldsymbol{x}(\boldsymbol{\sigma})$ when $\boldsymbol{\sigma}$ undergoes small perturbations. The particular case where $\beta(\boldsymbol{\sigma})$, called the *degenerate set*, is the empty set is particularly noteworthy. This case corresponds to the solution $\boldsymbol{x}(\boldsymbol{\sigma})$ being *nondegenerate*. As we see from the theorem below, $\boldsymbol{x}(\cdot)$ is then locally smooth around this value $\boldsymbol{\sigma}$. Similar to the "big O" notation, we write $f(x) = o(g(x))$ if

$$\lim_{g(x)\to 0} \frac{f(x)}{g(x)} = 0.$$

The following result summarizes the key properties of the option function $\boldsymbol{x}(\boldsymbol{\sigma})$. See [10, Section 7.4] for a detailed discussion. In a nutshell, the validity of this result is due to the "P-property" of the matrix $\boldsymbol{M}(\boldsymbol{\sigma})$, which is ensured by condition (5). This result belongs to the domain of sensitivity analysis of parametric complementarity problems, a subject that has been well researched in mathematical programming.

Theorem 1 *Suppose that the condition (5) holds. The option function $\boldsymbol{x}(\boldsymbol{\sigma})$ is Lipschitz continuous and directionally differentiable in its argument $\boldsymbol{\sigma} \in \Re^{M(N-1)}$. The directional derivative of $\boldsymbol{x}(\boldsymbol{\sigma})$ at any $\boldsymbol{\sigma} \in \Re^{M(N-1)}$ along any direction $d\boldsymbol{\sigma} \in \Re^{M(N-1)}$, denoted $\boldsymbol{x}'(\boldsymbol{\sigma}; d\boldsymbol{\sigma})$, is the unique solution $d\boldsymbol{x} \in \Re^{KM(N-1)}$ to the following mixed LCP:*

$$\left\{ \begin{array}{c} (\, J_\sigma \boldsymbol{F}(\boldsymbol{\sigma}, \boldsymbol{x}(\boldsymbol{\sigma})) d\boldsymbol{\sigma} + \boldsymbol{M}(\boldsymbol{\sigma}) d\boldsymbol{x} \,)_i = 0, \quad \forall\, i \in \alpha(\boldsymbol{\sigma}), \\ 0 \le (\, d\boldsymbol{x} \,)_i \perp (\, J_\sigma \boldsymbol{F}(\boldsymbol{\sigma}, \boldsymbol{x}(\boldsymbol{\sigma})) d\boldsymbol{\sigma} + \boldsymbol{M}(\boldsymbol{\sigma}) d\boldsymbol{x} \,)_i \ge 0, \quad \forall\, i \in \beta(\boldsymbol{\sigma}), \\ (\, d\boldsymbol{x} \,)_i = 0, \quad \forall\, i \in \gamma(\boldsymbol{\sigma}). \end{array} \right\} \tag{10}$$

Furthermore, it holds that

$$\boldsymbol{x}(\boldsymbol{\sigma} + d\boldsymbol{\sigma}) = \boldsymbol{x}(\boldsymbol{\sigma}) + \boldsymbol{x}'(\boldsymbol{\sigma}; d\boldsymbol{\sigma}) + o(\| d\boldsymbol{\sigma} \|). \tag{11}$$

Finally, if $\beta(\boldsymbol{\sigma})$ is empty, then $\boldsymbol{x}$ is Fréchet differentiable at $\boldsymbol{\sigma}$.

The theorem has several important consequences. The first consequence is that the implied volatility problem as formulated as either (8) or (9) always has a solution if the objective function $\theta(\boldsymbol{\sigma}, \boldsymbol{x})$ is continuous and the feasible volatility region $\boldsymbol{\Gamma}$ is compact.

Corollary 1 *Suppose that the objective function $\theta(\boldsymbol{\sigma}, \boldsymbol{x})$ is continuous. If $\boldsymbol{\Gamma}$ is compact, then an optimal solution to (8) exists.*

Another consequence of Theorem 1 is that it provides a computationally effective way of approximately updating a set of option values when the volatilities undergo small perturbations, without recomputing the exact values from scratch. Indeed, the formula (11) yields

$$\boldsymbol{x}(\boldsymbol{\sigma}') \approx \boldsymbol{x}(\boldsymbol{\sigma}) + \boldsymbol{x}'(\boldsymbol{\sigma}; d\boldsymbol{\sigma}), \quad \text{where} \quad d\boldsymbol{\sigma} \equiv \boldsymbol{\sigma}' - \boldsymbol{\sigma},$$

provided that $\boldsymbol{\sigma}'$ is a sufficiently small perturbation of $\boldsymbol{\sigma}$. Thus, we can compute an approximation to $\boldsymbol{x}(\boldsymbol{\sigma}')$ very easily from $\boldsymbol{x}(\boldsymbol{\sigma})$ by simply computing the directional derivative $\boldsymbol{x}'(\boldsymbol{\sigma}, d\boldsymbol{\sigma})$. In general, the computation of this derivative involves the solution of a mixed LCP of reduced size. The reason why the latter problem is of reduced size is because the last equation in (10) allows

the γ-components of $d\boldsymbol{x}$ to be fixed at zero. A noteworthy instance of (10) is when $\boldsymbol{x}(\boldsymbol{\sigma})$ is a nondegenerate solution, that is, when $\beta(\boldsymbol{\sigma})$ is empty. In this case, the mixed LCP (10) reduces to a single system of linear equations involving only the variables $d\boldsymbol{x}_i$ for $i \in \alpha(\boldsymbol{\sigma})$. In the general case, the mixed LCP (10) can either be solved as stated or be converted into a standard LCP involving only the variables $d\boldsymbol{x}_i$ for $i \in \beta(\boldsymbol{\sigma})$, by using the first equation in (10) to eliminate the variables $d\boldsymbol{x}_i$ for all $i \in \alpha(\boldsymbol{\sigma})$. See [10, Section 1.5] for details.

Since the option function $\boldsymbol{x}(\boldsymbol{\sigma})$ is not a smooth function, (9) is a nonsmooth optimization problem in general. Nevertheless, if the original objective function $\theta(\boldsymbol{\sigma}, \boldsymbol{x})$ is smooth, then the composite objective function $\varphi(\boldsymbol{\sigma})$ is "B(ouligand)-smooth", meaning that it is locally Lipschitz continuous and directionally differentiable. As such, we can attempt to apply a descent method for solving the problem (9). Such an approach was first discussed in [26] and later expanded in [25, Section 6.3]. Details are given in the next section. Furthermore, we can use the "B-derivative" of φ to describe the stationarity conditions of the program (9). We refer the reader to [25] for a comprehensive theory of the MPEC, which includes a full coverage of such stationarity conditions and many references on this class of optimization problem.

4 Two Solution Algorithms

We present two solution algorithms for solving the implied volatility problem. We call them, respectively, **IM**plicit **P**rogramming **A**lgorithm (IMPA) and **P**enalty **I**nterior **P**oint **A**lgorithm (PIPA). IMPA solves the implied volatility problem based on the optimization formulation (9); PIPA is based on the former formulation (8). Details of these algorithms and their convergence for general MPECs can be found in [25]. Here, we present the algorithms and briefly discuss some simplifications in the linear algebraic calculations when the algorithms are specialized to the implied volatility problem. For practical purposes, the feasible volatility region $\boldsymbol{\Gamma}$ is assumed to be a relatively simple set such as a polyhedron; as a result, the directional subprograms in both IMPA and PIPA (see descriptions in the respective subsections) can be easily solved.

4.1 An implicit programming algorithm

If not for the non-smoothness of $\boldsymbol{x}(\boldsymbol{\sigma})$, (9) is a fairly standard constrained optimization problem. With $\varphi(\boldsymbol{\sigma})$ being directionally differentiable, we may apply an iterative descent algorithm of the sequential quadratic programming kind to minimize this function; the resulting procedure is the essence of IMPA. Before describing the details of this iterative algorithm, we motivate a key step therein. Let $\boldsymbol{\sigma}^\nu \in \boldsymbol{\Gamma}$ be a given volatility vector that is not stationary for (9). Write $\boldsymbol{x}^\nu \equiv \boldsymbol{x}(\boldsymbol{\sigma}^\nu)$ and

$$\alpha_\nu \equiv \alpha(\boldsymbol{\sigma}^\nu), \quad \beta_\nu \equiv \beta(\boldsymbol{\sigma}^\nu), \quad \text{and} \quad \gamma_\nu \equiv \gamma(\boldsymbol{\sigma}^\nu).$$

We can generate a descent direction of the objective function $\varphi(\boldsymbol{\sigma})$ at $\boldsymbol{\sigma}^\nu$ by solving the following minimization problem: for any symmetric positive definite matrix $Q_\nu \in \Re^{M(N-1)\times M(N-1)}$,

$$\begin{aligned} &\text{minimize} && \varphi'(\boldsymbol{\sigma}^\nu; d\boldsymbol{\sigma}) + \tfrac{1}{2}\, d\boldsymbol{\sigma}^T Q_\nu d\boldsymbol{\sigma} \\ &\text{subject to} && \boldsymbol{\sigma}^\nu + d\boldsymbol{\sigma} \in \boldsymbol{\Gamma}. \end{aligned}$$

Since

$$\varphi'(\boldsymbol{\sigma}^\nu; d\boldsymbol{\sigma}) = \nabla_\sigma \varphi(\boldsymbol{\sigma}^\nu, \boldsymbol{x}^\nu)^T d\boldsymbol{\sigma} + \nabla_x \varphi(\boldsymbol{\sigma}^\nu, \boldsymbol{x}^\nu)^T \boldsymbol{x}'(\boldsymbol{\sigma}^\nu; d\boldsymbol{\sigma}),$$

the above directional minimization problem is equivalent to:

$$\begin{aligned}
\text{minimize} \quad & \nabla_\sigma \varphi(\boldsymbol{\sigma}^\nu, \boldsymbol{x}^\nu)^T d\boldsymbol{\sigma} + \nabla_x \varphi(\boldsymbol{\sigma}^\nu, \boldsymbol{x}^\nu)^T d\boldsymbol{x} + \tfrac{1}{2}\, d\boldsymbol{\sigma}^T Q_\nu d\boldsymbol{\sigma} \\
\text{subject to} \quad & \boldsymbol{\sigma}^\nu + d\boldsymbol{\sigma} \in \boldsymbol{\Gamma} \\
& (\, J_\sigma \boldsymbol{F}(\boldsymbol{\sigma}^\nu, \boldsymbol{x}^\nu) d\boldsymbol{\sigma} + \boldsymbol{M}(\boldsymbol{\sigma}^\nu) d\boldsymbol{x}\,)_i = 0, \quad \forall\, i \in \alpha_\nu, \\
& 0 \le (\, d\boldsymbol{x}\,)_i \perp (\, J_\sigma \boldsymbol{F}(\boldsymbol{\sigma}^\nu, \boldsymbol{x}^\nu) d\boldsymbol{\sigma} + \boldsymbol{M}(\boldsymbol{\sigma}^\nu) d\boldsymbol{x}\,)_i \ge 0, \quad \forall\, i \in \beta_\nu, \\
& (\, d\boldsymbol{x}\,)_i = 0, \quad \forall\, i \in \gamma_\nu,
\end{aligned} \tag{12}$$

where we have used Theorem 1 to substitute for the directional derivative $\boldsymbol{x}'(\boldsymbol{\sigma}^\nu; d\boldsymbol{\sigma})$. The latter optimization problem turns out to be another MPEC whose solution is not trivial. There are various strategies that one can use to modify (12) in order to obtain a practically efficient method. One such modification, whose goal is to simplify the directional computation, is included in the IMPA described below. In presenting this algorithm, we use the following additional notation:

$$\varphi_\nu \equiv \varphi(\boldsymbol{\sigma}^\nu, \boldsymbol{x}^\nu), \quad d\varphi_\sigma^\nu \equiv \nabla_\sigma \varphi(\boldsymbol{\sigma}^\nu, \boldsymbol{x}^\nu), \quad d\varphi_x^\nu \equiv \nabla_x \varphi(\boldsymbol{\sigma}^\nu, \boldsymbol{x}^\nu),$$

and $J_\sigma \boldsymbol{F}^\nu \equiv J_\sigma \boldsymbol{F}(\boldsymbol{\sigma}^\nu, \boldsymbol{x}^\nu)$.

IMPA for implied volatilities

Step 0. (Initialization) Let $\rho,\ \gamma \in (0, 1)$ be given scalars and let Q_0 be a given symmetric positive definite matrix of order $M(N-1)$. Let $\boldsymbol{\sigma}^0 \in \boldsymbol{\Gamma}$ be given. Set $\nu = 0$.

Step 1. (Direction generation) Let β' be an arbitrary subset of β_ν. Solve the convex quadratic program

$$\begin{aligned}
\text{minimize} \quad & (\, d\varphi_\sigma^\nu\,)^T d\boldsymbol{\sigma} + (\, d\varphi_x^\nu\,)^T d\boldsymbol{x} + \tfrac{1}{2}\, d\boldsymbol{\sigma}^T Q_\nu d\boldsymbol{\sigma} \\
\text{subject to} \quad & \boldsymbol{\sigma}^\nu + d\boldsymbol{\sigma} \in \boldsymbol{\Gamma} \\
& (\, J_\sigma \boldsymbol{F}^\nu d\boldsymbol{\sigma} + \boldsymbol{M}^\nu d\boldsymbol{x}\,)_i = 0, \quad \forall\, i \in \alpha_\nu \cup \beta', \\
& (\, d\boldsymbol{x}\,)_i = 0, \quad \forall\, i \in \gamma_\nu \cup (\, \beta_\nu \setminus \beta'\,),
\end{aligned} \tag{13}$$

to obtain the search direction $d\boldsymbol{\sigma}$ and the auxiliary vector $d\boldsymbol{x}$, both of which must necessarily be unique.

Step 2. (Termination test) If

$$\| d\boldsymbol{\sigma} \| \le \text{ prescribed tolerance},$$

stop; we take the pair $(\boldsymbol{\sigma}^\nu, \boldsymbol{x}^\nu)$ to be a desired approximate solution of (9).

Step 3. (Step size determination) With

$$\boldsymbol{\sigma}^\nu(\tau) \equiv \boldsymbol{\sigma}^\nu + \tau\, d\boldsymbol{\sigma} \quad \text{and} \quad \boldsymbol{x}^\nu(\tau) \equiv \boldsymbol{x}(\boldsymbol{\sigma}^\nu(\tau)), \quad \forall\, \tau \in [0, 1],$$

set

$$\tau_\nu \equiv \max(\, 0.01,\ \rho^{\ell_\nu}\,) \tag{14}$$

where ℓ_ν is the smallest nonnegative integer ℓ such that with $\tau \equiv \rho^\ell$,

$$\varphi(\boldsymbol{\sigma}^\nu(\tau), \boldsymbol{x}^\nu(\tau)) - \varphi_\nu \;\leq\; \gamma\,\tau\,[\,(\,d\varphi_\sigma^\nu\,)^T d\boldsymbol{\sigma} + (\,d\varphi_x^\nu\,)^T d\boldsymbol{x}\,].$$

Set $\boldsymbol{\sigma}^{\nu+1} \equiv \boldsymbol{\sigma}^\nu(\tau_\nu)$. Choose a symmetric positive definite matrix $Q_{\nu+1}$ and replace ν by $\nu+1$. Return to Step 1.

The choice of the matrices $\{Q_\nu\}$ could presumably affect the practical performance of the algorithm. Ideally, this should be chosen to reflect some second-order information of the objective function $\theta(\boldsymbol{\sigma}, \boldsymbol{x})$. Nevertheless, this issue is not well studied in the MPEC literature, due to the nonconvexity of such problems. In our experiments, the choice of the identity matrix for each Q_ν yields fairly satisfactory results. The rule (14) imposes a lower bound of 0.01 on the step size. The rationale for this is that with the modification of the direction search as described in Step 1, the search direction $d\boldsymbol{\sigma}$ obtained from this step is no longer guaranteed to be a descent direction for the function $\varphi(\boldsymbol{\sigma})$ at $\boldsymbol{\sigma}^\nu$. Thus we use the lower bound to prevent the iteration to take too small a step. Other than this precaution, the step size determination is the well-known Armijo inexact line search rule in standard unconstrained optimization algorithms. A convergence analysis of the algorithm without the modification of the directional step can be found in [25]. At this time, there is no convergence proof of the algorithm as described above (with the directional modification). Nevertheless, the computational results reported in the next section suggests that the algorithm performs quite well on the test problems.

4.2 A penalty interior point algorithm

Unlike IMPA that originates from an iterative descent algorithm for solving smooth nonlinear programs, PIPA originates from a penalty method coupled with an interior point routine to deal with complementarity constraint in the other formulation (8) of the implied volatility problem. For background on such an interior point method, the reader is referred to [30] and the many references therein.

PIPA operates on the following obvious reformulation of (8):

$$\begin{aligned}
\text{minimize} \quad & \theta(\boldsymbol{\sigma}, \boldsymbol{x}) \\
\text{subject to} \quad & \boldsymbol{\sigma} \;\in\; \boldsymbol{\Gamma} \\
& \boldsymbol{w} - \boldsymbol{q}(\boldsymbol{\sigma}) - \boldsymbol{M}(\boldsymbol{\sigma})\,\boldsymbol{x} \;=\; 0 \\
& \boldsymbol{w} \circ (\,\boldsymbol{x} - \boldsymbol{p}\,) \;=\; 0 \\
& (\,\boldsymbol{w},\, \boldsymbol{x} - \boldsymbol{p}\,) \;\geq\; 0
\end{aligned}$$

where $\circ$ denotes the Hadamard operator on two vectors; that is, $\boldsymbol{w} \circ \boldsymbol{x}$ is the vector whose i-th component is equal to the product of the i-th components of $\boldsymbol{w}$ and $\boldsymbol{x}$. Before presenting the details of PIPA, we introduce some notation and explain the key ideas behind the algorithm. Let

$$\boldsymbol{G}(\boldsymbol{\sigma}, \boldsymbol{x}, \boldsymbol{w}) \;\equiv\; \boldsymbol{w} - \boldsymbol{q}(\boldsymbol{\sigma}) - \boldsymbol{M}(\boldsymbol{\sigma})\,\boldsymbol{x} \;=\; \boldsymbol{w} - \boldsymbol{F}(\boldsymbol{\sigma}, \boldsymbol{x})$$

and define the residual function of the complementarity conditions:

$$r(\boldsymbol{\sigma}, \boldsymbol{x}, \boldsymbol{w}) \;\equiv\; (\,\boldsymbol{G}(\boldsymbol{\sigma}, \boldsymbol{x}, \boldsymbol{w})\,)^T \boldsymbol{G}(\boldsymbol{\sigma}, \boldsymbol{x}, \boldsymbol{w}) + (\,\boldsymbol{x} - \boldsymbol{p}\,)^T \boldsymbol{w}.$$

Also define a penalized objective function

$$P_c(\boldsymbol{\sigma}, \boldsymbol{x}, \boldsymbol{w}) \equiv \theta(\boldsymbol{\sigma}, \boldsymbol{x}) + c\, r(\boldsymbol{\sigma}, \boldsymbol{x}, \boldsymbol{w}),$$

where $c > 0$ is a penalty scalar to be adjusted. For a given vector $\boldsymbol{x}$, let $\mathrm{diag}(\boldsymbol{x})$ be the diagonal matrix whose diagonal entries are the components of this vector.

PIPA is an iterative algorithm that generates a sequence of iterates $\{(\boldsymbol{\sigma}^\nu, \boldsymbol{x}^\nu, \boldsymbol{w}^\nu)\}$ satisfying the following conditions for all ν:

a. (feasible volatilities) $\boldsymbol{\sigma}^\nu \in \boldsymbol{\Gamma}$;

b. (positivity of state variables) $\boldsymbol{x}^\nu - \boldsymbol{p} > 0$ and $\boldsymbol{w}^\nu > 0$; and

c. (centrality condition) $(\boldsymbol{x}^\nu - \boldsymbol{p}) \circ \boldsymbol{w}^\nu \geq \eta g_\nu \mathbf{1}_{M(N-1)}$, where $\eta \in (0, 1)$ is a given scalar, $\mathbf{1}_{M(N-1)}$ is the $M(N-1)$-vector of all ones, and

$$g_\nu \equiv \frac{(\boldsymbol{x}^\nu - \boldsymbol{p})^T (\boldsymbol{w}^\nu)}{M(N-1)}$$

is the average "complementary gap" between $\boldsymbol{x}^\nu$ and $\boldsymbol{w}^\nu$.

While maintaining these conditions, PIPA attempts to decrease the objective function $\theta(\boldsymbol{\sigma}, \boldsymbol{x})$ by reducing the penalty function $P_c(\boldsymbol{\sigma}, \boldsymbol{x}, \boldsymbol{w})$ via an Armijo inexact line search using a properly defined search direction and a proper choice of the penalty scalar c. The ultimate goal of PIPA is twofold: one, to drive the residual $r(\boldsymbol{\sigma}, \boldsymbol{x}, \boldsymbol{w})$ toward zero, thereby achieving feasibility to the MPEC, and two, to obtain a satisfactory "minimum" value of the objective $\theta(\boldsymbol{\sigma}, \boldsymbol{x})$. For the detailed convergence theory of this algorithm, we refer the reader to the text [25].

At the beginning of each iteration ν, a triple $(\boldsymbol{\sigma}^\nu, \boldsymbol{x}^\nu, \boldsymbol{w}^\nu)$ satisfying the above conditions is given. Consistent with the notation used so far, we write

$$\boldsymbol{G}^\nu \equiv \boldsymbol{G}(\boldsymbol{\sigma}^\nu, \boldsymbol{x}^\nu, \boldsymbol{w}^\nu) \quad \text{and} \quad r_\nu \equiv r(\boldsymbol{\sigma}^\nu, \boldsymbol{x}^\nu, \boldsymbol{w}^\nu).$$

At this triple, a Newton linearization step is applied to the equation:

$$\boldsymbol{w} - \boldsymbol{q}(\boldsymbol{\sigma}) - \boldsymbol{M}(\boldsymbol{\sigma})\, \boldsymbol{x} = 0$$

and also to the following *perturbed* complementarity equation:

$$\boldsymbol{w} \circ (\boldsymbol{x} - \boldsymbol{p}) = \kappa\, g_\nu\, \mathbf{1}_{M(N-1)},$$

where κ is a given constant in the interval $(0, 1)$. The resulting linear equations are used to define a directional quadratic program whose solution yields a search direction along which the aforementioned line search is carried out. The following is a detailed step-by-step description of PIPA.

PIPA for implied volatilities

Step 0. (Initialization) Let ρ, γ, κ_0, and η be given scalars in the interval $(0, 1)$. Let $c_{-1} \geq 1$ and $\zeta > 0$ be given scalars. Let H_0 be a given symmetric positive (semi)definite matrix of order $2M(N-1)$. Let $(\boldsymbol{\sigma}^0, \boldsymbol{x}^0, \boldsymbol{w}^0)$ be a given triple satisfying the three conditions (a, b, c). In particular,

for a given $\boldsymbol{\sigma}^0 \in \boldsymbol{\Gamma}$, x^0 is the set of American option prices calculated using $\boldsymbol{\sigma}^0$. $\boldsymbol{w}^0$ is any arbitrary positive vector. Set $\nu = 0$.

Step 1. (Direction generation) Solve the convex quadratic program

$$\begin{aligned} \text{minimize} \quad & (\, d\varphi_\sigma^\nu \,)^T d\boldsymbol{\sigma} + (\, d\varphi_x^\nu \,)^T d\boldsymbol{x} + \tfrac{1}{2} \begin{pmatrix} d\boldsymbol{\sigma} \\ d\boldsymbol{x} \end{pmatrix}^T H_\nu \begin{pmatrix} d\boldsymbol{\sigma} \\ d\boldsymbol{x} \end{pmatrix} \\ \text{subject to} \quad & \boldsymbol{\sigma}^\nu + d\boldsymbol{\sigma} \in \boldsymbol{\Gamma} \\ & \boldsymbol{G}^\nu + d\boldsymbol{w} - J_\sigma \boldsymbol{F}^\nu d\boldsymbol{\sigma} - \boldsymbol{M}^\nu d\boldsymbol{x} = 0 \\ & (\, \boldsymbol{x}^\nu - \boldsymbol{p} \,) \circ \boldsymbol{w}^\nu + \mathrm{diag}(\boldsymbol{x}^\nu - \boldsymbol{p}) d\boldsymbol{w} + \mathrm{diag}(\boldsymbol{w}^\nu) d\boldsymbol{x} = \kappa_\nu \, g_\nu \, \mathbf{1}_{M(N-1)} \\ & \| \, d\boldsymbol{\sigma} \, \|_\infty \le \zeta \, r_\nu \end{aligned} \tag{15}$$

to obtain the search triple $(d\boldsymbol{\sigma}, d\boldsymbol{x}, d\boldsymbol{w})$.

Step 2. (Termination test) If

$$\| \, (\, d\boldsymbol{\sigma}, \, d\boldsymbol{x}, \, d\boldsymbol{w} \,) \, \| + r_\nu \le \text{ prescribed tolerance},$$

stop; we take $(\boldsymbol{\sigma}^\nu, \boldsymbol{x}^\nu)$ to be a desired approximate solution of (8).

Step 3. (Penalty update) Let $s_\nu \ge 1$ be the smallest integer $s \ge 1$ such that

$$(\, d\varphi_\sigma^\nu \,)^T d\boldsymbol{\sigma} + (\, d\varphi_x^\nu \,)^T d\boldsymbol{x} - c_{\nu-1}^s \left[2 \, (\, \boldsymbol{G}^\nu \,)^T \boldsymbol{G}^\nu + (\, 1 - \kappa_\nu \,) (\, \boldsymbol{x}^\nu \,)^T \boldsymbol{w}^\nu \right] < -r_\nu.$$

Set $c_\nu \equiv c_{\nu-1}^{s_\nu}$.

Step 4. (Step size determination: centrality) With

$$\begin{pmatrix} \boldsymbol{\sigma}^\nu(\tau) \\ \boldsymbol{x}^\nu(\tau) \\ \boldsymbol{w}^\nu(\tau) \end{pmatrix} \equiv \begin{pmatrix} \boldsymbol{\sigma}^\nu \\ \boldsymbol{x}^\nu \\ \boldsymbol{w}^\nu \end{pmatrix} + \tau \begin{pmatrix} d\boldsymbol{\sigma} \\ d\boldsymbol{x} \\ d\boldsymbol{w} \end{pmatrix},$$

determine the largest $\tau \in (0, 1]$ such that

$$\boldsymbol{x}^\nu(\tau) \circ \boldsymbol{w}^\nu(\tau) \ge \eta \, g_\nu(\tau) \, \mathbf{1}_{M(N-1)},$$

where

$$g_\nu(\tau) = \frac{(\, \boldsymbol{x}^\nu(\tau) - \boldsymbol{p} \,)^T (\, \boldsymbol{w}^\nu(\tau) \,)}{M(N-1)}.$$

Set $\tau_\nu' \equiv 0.9999\tau$.

Step 5. (Step size determination: Armijo line search) Set

$$\tau_\nu \equiv \tau_\nu' \, \rho^{\ell_\nu}$$

where ℓ_ν is the smallest nonnegative integer ℓ such that with $\tau = \tau'_\nu \rho^\ell$,

$$P_{c_\nu}(\boldsymbol{\sigma}^\nu(\tau), \boldsymbol{x}^\nu(\tau), \boldsymbol{w}^\nu(\tau)) - P_{c_\nu}(\boldsymbol{\sigma}^\nu, \boldsymbol{x}^\nu, \boldsymbol{w}^\nu) <$$

$$\gamma\,\tau\,\left[\,(\,d\varphi^\nu_\sigma\,)^T d\boldsymbol{\sigma} + (\,d\varphi^\nu_x\,)^T d\boldsymbol{x} - c^s_{\nu-1}\,\left(\,2\,(\,\boldsymbol{G}^\nu\,)^T\boldsymbol{G}^\nu + (\,1-\kappa_\nu\,)(\,\boldsymbol{x}^\nu\,)^T\boldsymbol{w}^\nu\,\right)\right].$$

Set $(\boldsymbol{\sigma}^{\nu+1}, \boldsymbol{x}^{\nu+1}, \boldsymbol{w}^{\nu+1}) \equiv (\boldsymbol{\sigma}^\nu(\tau_\nu), \boldsymbol{x}^\nu(\tau_\nu), \boldsymbol{w}^\nu(\tau_\nu))$. Choose a scalar $\kappa_{\nu+1} \in (0, \kappa_\nu]$ and a symmetric positive (semi)definite matrix $H_{\nu+1}$. Return to Step 1 with ν replaced by $\nu+1$.

IMPA and PIPA share much resemblance in the computational steps; they also have substantial differences in the underlying design philosophy, numerical implementation, and convergence analysis. There is also much fine tuning that one can apply to the algorithms. We have presented the essence of both algorithms. The implementation reported in the next section has adopted some of the most basic numerical techniques in dealing with the linear algebraic calculations; the computer codes that we have written for the experimentation are far from being a final product of commercial quality.

5 A Numerical Study

We have written two experimental computer codes in MATLAB to evaluate the numerical performance of IMPA and PIPA. We consider the inverse pricing of $K \in \{2, 11, 14\}$ American put options in order to determine the discretized volatilities implied by certain given option values. Thus for $k = 1, \ldots, K$, the payoff function of option k is given by

$$\Lambda^k(S,t) \;\equiv\; \max(S - E_k, 0)$$

where E_k is the strike price of the option at its expiration date T_k. For each option k, a set of discretized American option prices $V_n^{\mathrm{obs},k} \equiv V^{\mathrm{obs},k}(n\delta S, 0)$, representing the set of currently observable option prices, is generated by solving the single-option LCP (4) using a constant volatility of $\boldsymbol{\sigma}_0 = 0.4$ and the following parameters: $N = 16$, $\delta S = 1$, $T = T_k$, $\delta t = 0.125$, $r = 0.05$, and $D_0 = 0.02$.

The constant $\boldsymbol{\sigma}_0$ is not known to IMPA or PIPA; instead, we use the values $V_n^{\mathrm{obs},k}$ to define the following least-squares objective function:

$$\theta(\boldsymbol{x}) \;\equiv\; \frac{1}{2}\sum_{k=1}^{K}\sum_{n\in\mathcal{N}_k}\,(\,V^k(n\delta S, 0) - V_n^{\mathrm{obs},k}\,)^2.$$

The index set $\mathcal{N}_k$ is a singleton in all the runs; it consists of $\{n'\}$ such that $n'\delta S \equiv S_0 \equiv$ the current asset price. With the objective of minimizing $\theta(\boldsymbol{x})$, our goal in this set of experiments is to investigate whether IMPA or PIPA is able to obtain an objective value of θ close to zero. Ideally, we would like the algorithms to reproduce the constant volatility $\boldsymbol{\sigma}_0$. In reality, since the inverse MPEC (8) does not necessarily have a unique optimal solution, we do not expect a constant volatility to be produced by the algorithms; instead, we will be satisfied if the algorithms can produce a small θ value. We further take $\boldsymbol{\Gamma}$ to be a rectangular box, meaning that the unknown volatility vector $\boldsymbol{\sigma}$ is subject to simple upper and lower bounds. In particular, we set an upper bound of 1.0 and a lower bound of 0.0.

Before presenting the computational results, we give some details involved in the implementation of the two algorithms. In IMPA, the major computations in each iteration are (i) the direction generation step which involves solving the convex quadratic program (13), and (ii) the evaluation of $\boldsymbol{x}^\nu(\tau)$ for various step sizes $\tau = \rho^\ell$. By using the equality constraints in (13), we can solve for the variables $d\boldsymbol{x}_i$ for $i \in \alpha_\nu \cup \beta'$ in terms of $d\boldsymbol{\sigma}$, thereby converting (13) into a strictly convex quadratic program in the variable $d\boldsymbol{\sigma}$ alone. In the experiments below, we have chosen each Q_ν to be the identity matrix. This choice greatly simplifies the resulting quadratic program. In fact, with the feasible set $\boldsymbol{\Gamma}$ being a rectangular box, the solution of the latter program becomes trivial.

The evaluation of $\boldsymbol{x}^\nu(\tau)$ involves solving the forward option LCP (7) corresponding to $\boldsymbol{\sigma}^\nu(\tau)$. This LCP decomposes into K independent linear complementarity subproblems each corresponding to a given option. In turn, as mentioned before, each single-option LCP can be solved backward in time. Thus, the evaluation of $\boldsymbol{x}^\nu(\tau)$ amounts to solving KM time-stepped LCPs each of the form (4). In principle, there are many methods applicable for solving the latter LCP (4); see [10, 17]. In a separate experiment, we have compared several of these methods and concluded that when N (the number of discretized asset prices) is large, an interior point algorithm [30] is the preferred method for solving the LCP (4). Thus, this interior point algorithm is chosen as the principal tool for evaluating $\boldsymbol{x}^\nu(\tau)$.

The main computation in each iteration of PIPA is the solution of the convex quadratic program (15). Unlike IMPA, there is no need to solve any forward option LCP. As in IMPA, the program (15) can be converted into one in the variable $d\boldsymbol{\sigma}$ alone. In both cases, the highly special structure of the matrices involved is exploited to facilitate the computations.

The termination test for IMPA is

$$\min(\, \| \, d\boldsymbol{\sigma} \, \|, \, \theta(\boldsymbol{x}^\nu, \boldsymbol{\sigma}^\nu) \,) \, \leq \, 10^{-6}. \tag{16}$$

A maximum number of 80 iterations is also imposed. Since IMPA maintains the feasibility of the pair $(\boldsymbol{\sigma}^\nu, \boldsymbol{x}^\nu)$ to the MPEC (8), if IMPA terminates after 80 iterations without satisfying (16), we obtain an implied volatility vector $\boldsymbol{\sigma}^\nu \in \boldsymbol{\Gamma}$ and associated American option prices $\boldsymbol{x}^\nu$ that are deemed satisfactory. As we shall see, the objective values $\theta(\boldsymbol{x}^\nu)$ at termination of IMPA in all runs are invariably very small.

The termination test for PIPA is

$$\| \, (\, d\boldsymbol{\sigma}, \, d\boldsymbol{x}, \, d\boldsymbol{w} \,) \, \| + r_\nu \, \leq \, 10^{-8}. \tag{17}$$

At termination, the iterate $(\boldsymbol{\sigma}^\nu, \boldsymbol{x}^\nu)$ produced by PIPA is an approximate feasible solution to the MPEC (8) with the feasibility accuracy being less than 10^{-8}. The objective value $\theta(\boldsymbol{x}^\nu)$ is reported in each run.

5.1 Example 1

Two observed put option values are given ($K = 2$):

k	(E_k, T_k)	$V^{\mathrm{obs},k}$
1	$(8, 0.5)$	0.8003
2	$(8, 1.0)$	1.1217

The current asset price is $S_0 = 8$. We use an initial value $\boldsymbol{\sigma}^0 = \boldsymbol{\sigma}_{\text{init}} = 0.255$ to start both algorithms. For PIPA, this initial value $\boldsymbol{\sigma}^0$ induces an initial pair $(\boldsymbol{x}^0, \boldsymbol{w}^0)$ that is obtained by slightly perturbing $(\boldsymbol{x}(\boldsymbol{\sigma}_{\text{init}}), \boldsymbol{w}(\boldsymbol{\sigma}_{\text{init}}))$ in order to satisfy the positivity condition (b); in turn, a scalar η is then defined so that the condition (c) is also satisfied. The discretized volatility surfaces computed by IMPA and PIPA are shown in Figures 1 and 2, respectively.

The outputs from the two algorithms are summarized in the table below. Note that the objective value obtained by PIPA is larger than that obtained by IMPA even though the residual is quite small. The two surfaces in Figure 1 and Figure 2 have similar shape, but are not identical.

	# iter	residual	$\|d\boldsymbol{\sigma}\|$	θ
PIPA	31	3.0995e-09	1.2694e-11	1.0832e-02

	# iter	$\|d\boldsymbol{\sigma}\|$	θ
IMPA	73	4.7601e-04	8.7225e-07

Since PIPA and IMPA did not produce identical surfaces and neither recovered the original constant volatility surface used to generate the observed option data, as a test to determine how close the option prices calculated from given volatility are being reproduced, we use the volatility surfaces obtained by PIPA and IMPA now as parameters in the forward problem to re-calculate the option values. For given strike prices $E = 8$ and $E = 9$, we plot the three set of option values computed using: 1) $\boldsymbol{\sigma}(S,t) = 0.4$, 2) $\boldsymbol{\sigma}(S,t) = \boldsymbol{\sigma}_{\text{PIPA}}(S,t)$, and 3) $\boldsymbol{\sigma}(S,t) = \boldsymbol{\sigma}_{\text{IMPA}}(S,t)$. See Figures 3–4 where we use "V obs" to mean the option values calculated using $\boldsymbol{\sigma}_0(S,t) = 0.4$. ¿From these figures, we see that IMPA reproduces the option prices $\boldsymbol{V}_{\text{obs}}$ rather closely when the asset price is close to the strike price.

5.2 Example 2

Eleven observed put option values are given ($K = 11$):

k	(E_k, T_k)	$V^{\text{obs},k}$	k	(E_k, T_k)	$V^{\text{obs},k}$
1	(3, 1.0)	0.0041	7	(9, 1.0)	1.7287
2	(4, 1.0)	0.0288	8	(10, 1.0)	2.4339
3	(5, 1.0)	0.1156	9	(11, 1.0)	3.2277
4	(6, 1.0)	0.3131	10	(12, 1.0)	4.0994
5	(7, 1.0)	0.6497	11	(13, 1.0)	5.0117
6	(8, 1.0)	1.1217			

The corresponding parameters are the same as those given in Example 1. Following the analysis of Example 1, we give the results below.

	# iter	residual	$\|d\boldsymbol{\sigma}\|$	θ
PIPA	73	7.0148e-09	6.8803e-15	2.2206e-03

	# iter	$\|d\boldsymbol{\sigma}\|$	θ
IMPA	80	1.2632e-03	1.8398e-04

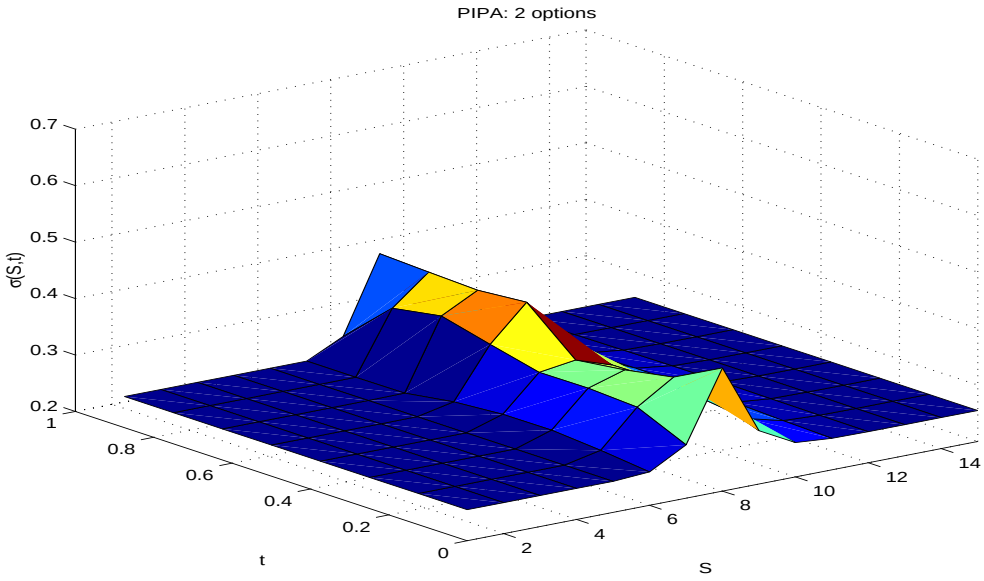

Figure 1: Example 1, volatility surface produced by PIPA, 2 observed options

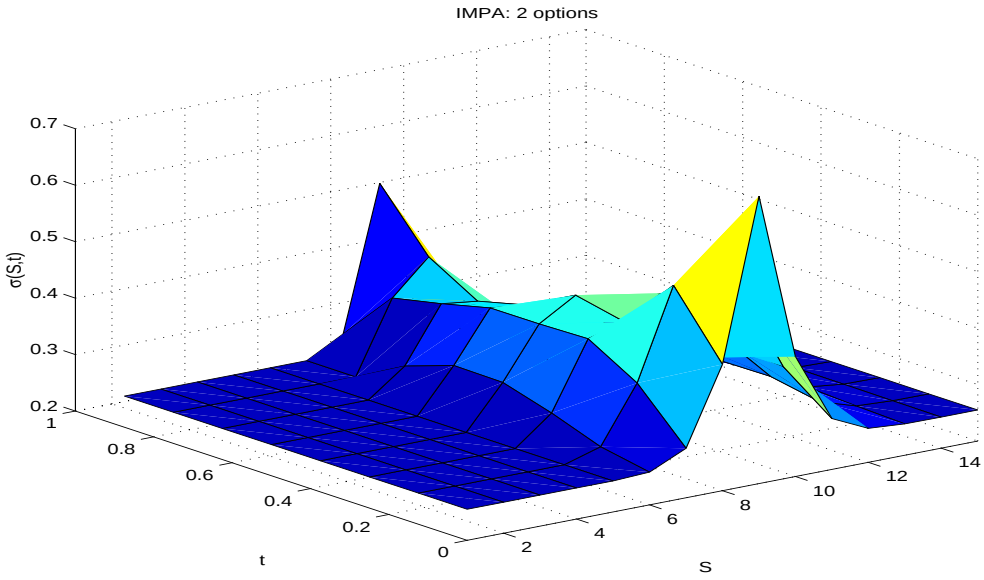

Figure 2: Example 1, volatility surface produced by IMPA, 2 observed options

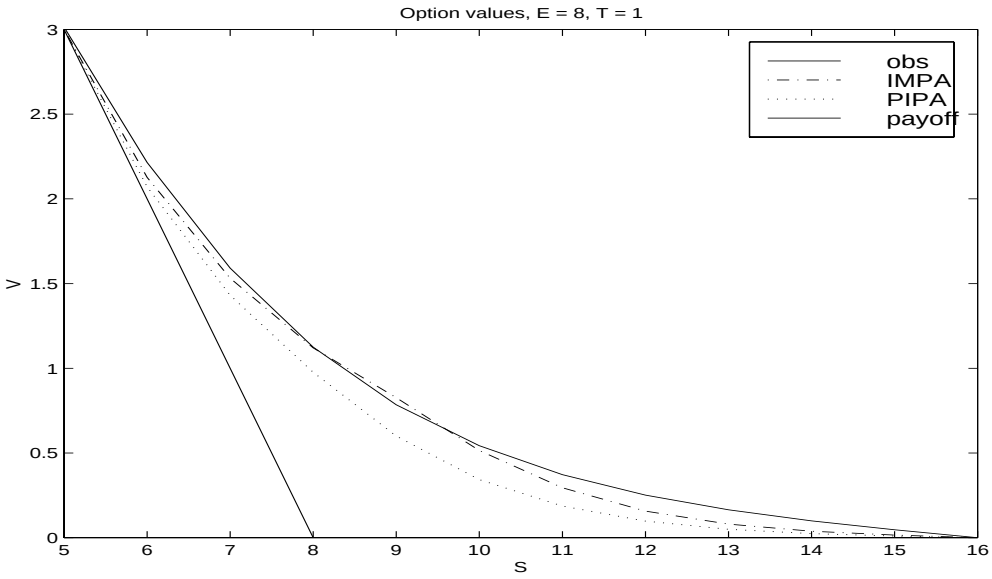

Figure 3: Example 1, options values calculated using constant/PIPA/IMPA volatility surfaces, E = 8

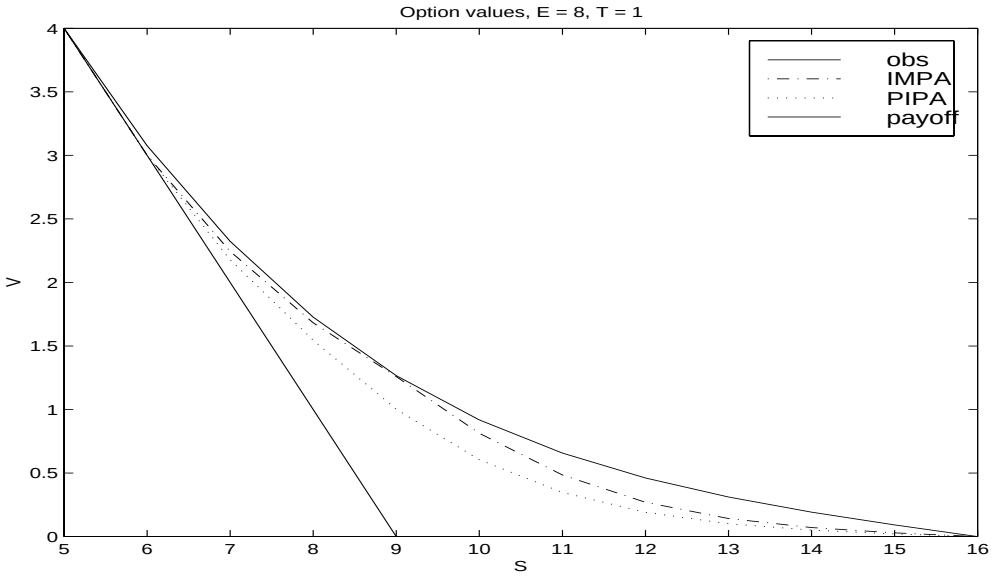

Figure 4: Example 1, options values calculated using constant/PIPA/IMPA volatility surfaces, E = 9

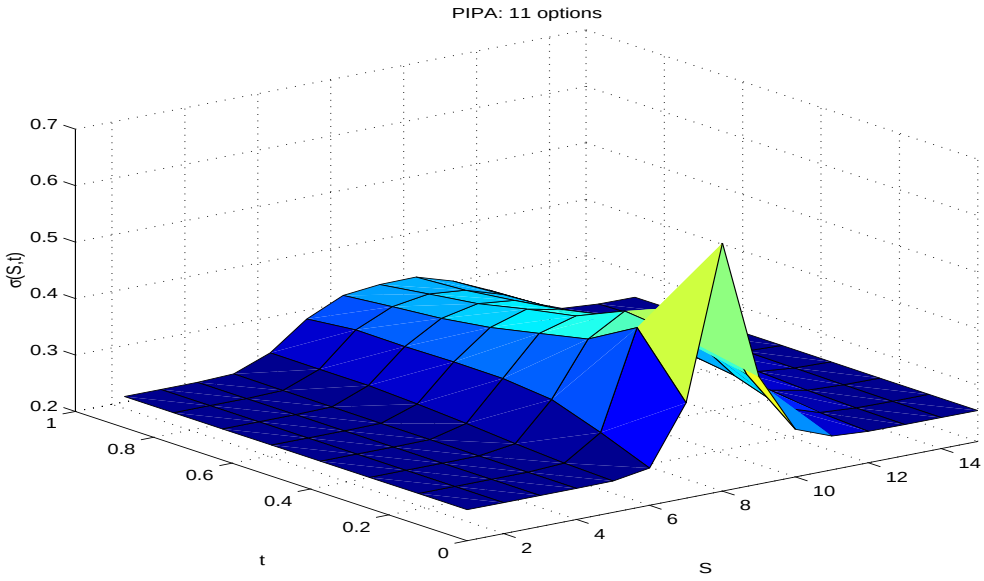

Figure 5: Example 2, volatility surface produced by PIPA, 11 observed options

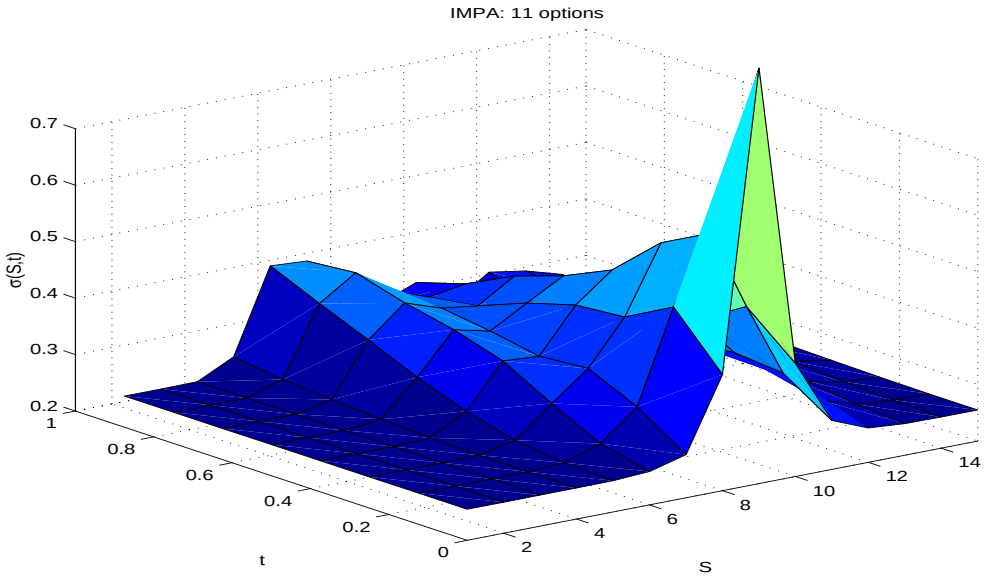

Figure 6: Example 2, volatility surface produced by IMPA, 11 observed options

Again, we plot the re-calculated option values for strike prices $E = 8$ and $E = 9$, see Figures 7–8. In this case, both IMPA and PIPA reproduce the observed option curve closely when the asset price is close to the strike price.

5.3 Example 3

Fourteen observed put option values are given ($K = 14$):

k	(E_k, T_k)	$V^{\text{obs},k}$	k	(E_k, T_k)	$V^{\text{obs},k}$
1	$(5, 0.5)$	0.0279	8	$(5, 1.0)$	0.1156
2	$(6, 0.5)$	0.1240	9	$(6, 1.0)$	0.3131
3	$(7, 0.5)$	0.3884	10	$(7, 1.0)$	0.6497
4	$(8, 0.5)$	0.8003	11	$(8, 1.0)$	1.1217
5	$(9, 0.5)$	1.4496	12	$(9, 1.0)$	1.7287
6	$(10, 0.5)$	2.1913	13	$(10, 1.0)$	2.4339
7	$(11, 0.5)$	3.0533	14	$(11, 1.0)$	3.2277

We give the results below. For detailed explanation, the reader is referred to the discussion in Example 1.

PIPA	# iter	residual	$\|d\boldsymbol{\sigma}\|$	θ
	66	2.0231e-09	1.3517e-15	2.6059e-03

IMPA	# iter	$\|d\boldsymbol{\sigma}\|$	θ
	80	1.7086e-03	1.5764e-04

The re-calculated option values for strike prices $E = 8$ and $E = 9$ are plotted in Figures 11–12.

It is interesting to note that in all three examples, the volatility surface calculated by PIPA is lower than that calculated by IMPA. This is also reflected in the plots of the option values: the option values calculated from $\boldsymbol{\sigma}_{\text{PIPA}}(S, t)$ are below those calculated from $\boldsymbol{\sigma}_0 = 0.4$ and $\boldsymbol{\sigma}_{\text{IMPA}}$.

5.4 Example 4

Again taking the option values calculated from the constant volatility ($\boldsymbol{\sigma}_0 = 0.4$) to be the observed values, we now perturb each observed values by a small amount; i.e.

$$\tilde{V}_n^{\text{obs},k} \equiv V_n^{\text{obs},k} + 0.0707$$

Thus, the new objective function is given by

$$\begin{aligned}\theta(\boldsymbol{x}) &= \tfrac{1}{2}\sum_{k=1}^{K}\sum_{n\in\mathcal{N}_k}\left(V^k(n\delta S, 0) - \tilde{V}_n^{\text{obs},k}\right)^2 \\ &= \tfrac{1}{2}\sum_{k=1}^{K}\sum_{n\in\mathcal{N}_k}\left(V^k(n\delta S, 0) - V_n^{\text{obs},k} - 0.0707\right)^2\end{aligned}$$

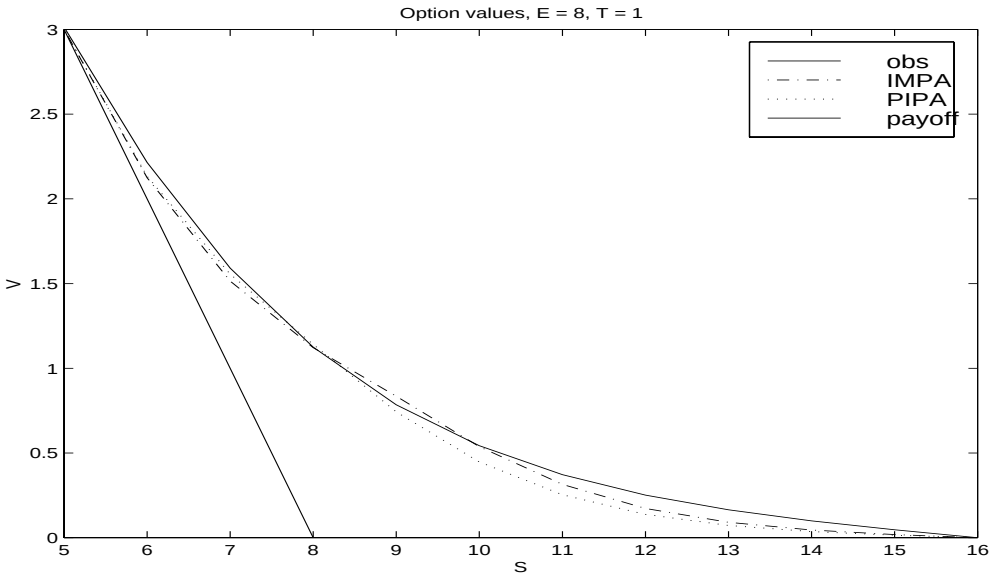

Figure 7: Example 2, options values calculated using constant/PIPA/IMPA volatility surfaces, E = 8

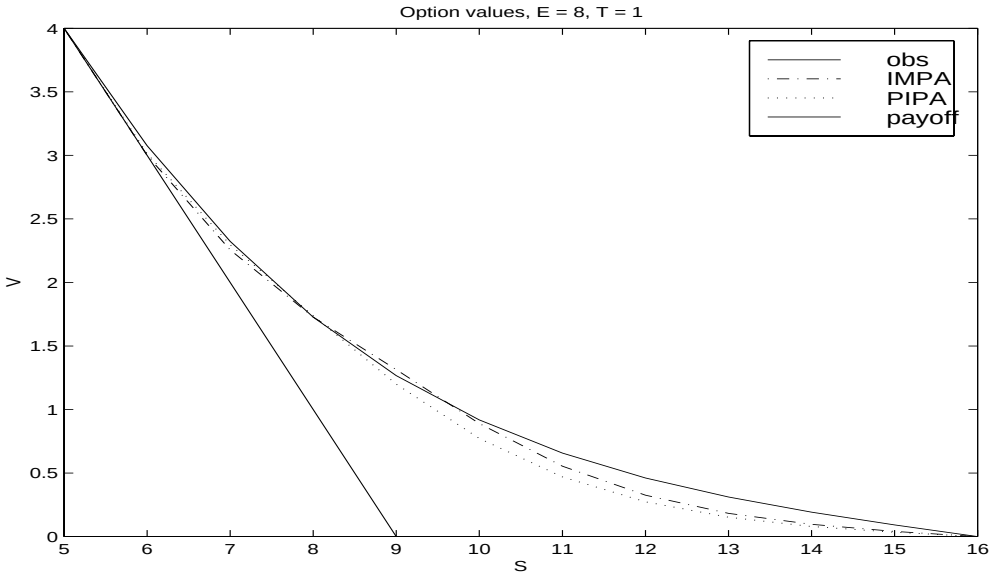

Figure 8: Example 2, options values calculated using constant/PIPA/IMPA volatility surfaces, E = 9

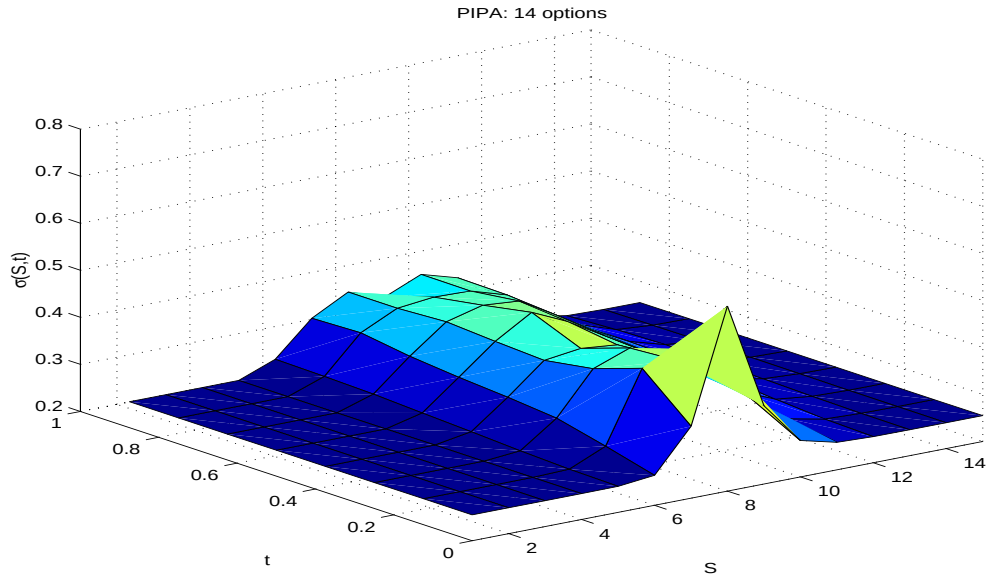

Figure 9: Example 3, volatility surface produced by PIPA, 14 observed options

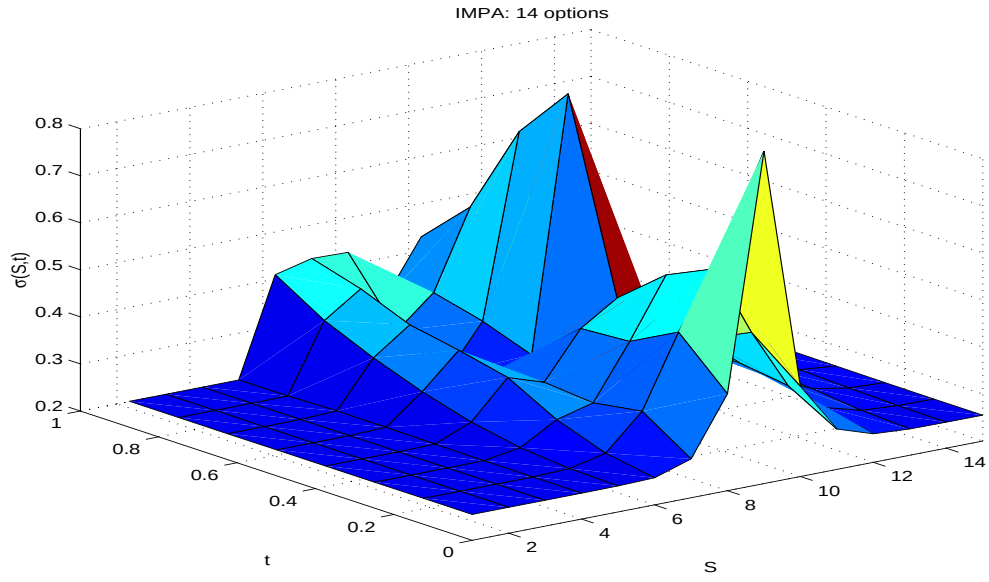

Figure 10: Example 3, volatility surface produced by IMPA, 14 observed options

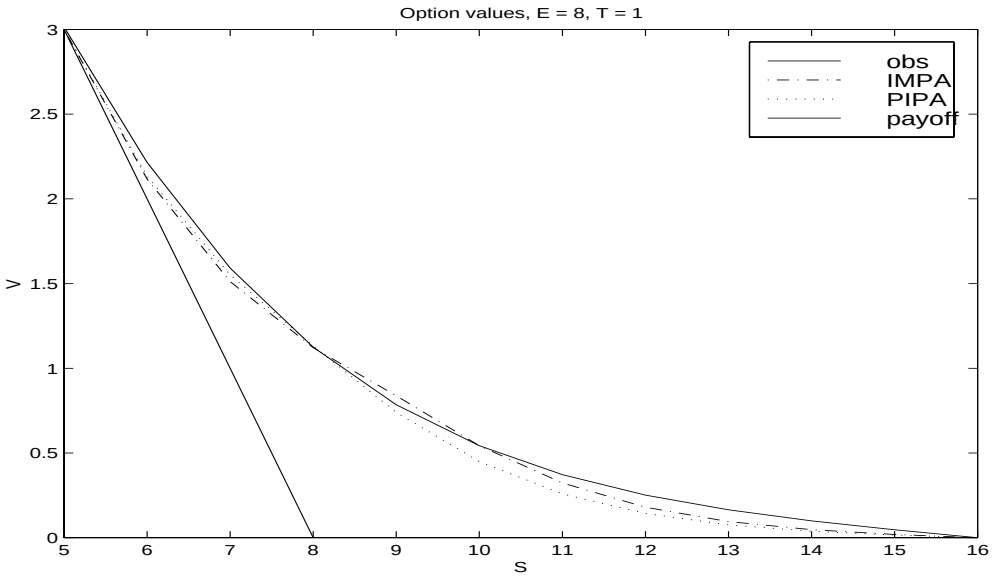

Figure 11: Example 3, options values calculated using constant/PIPA/IMPA volatility surfaces, E = 8

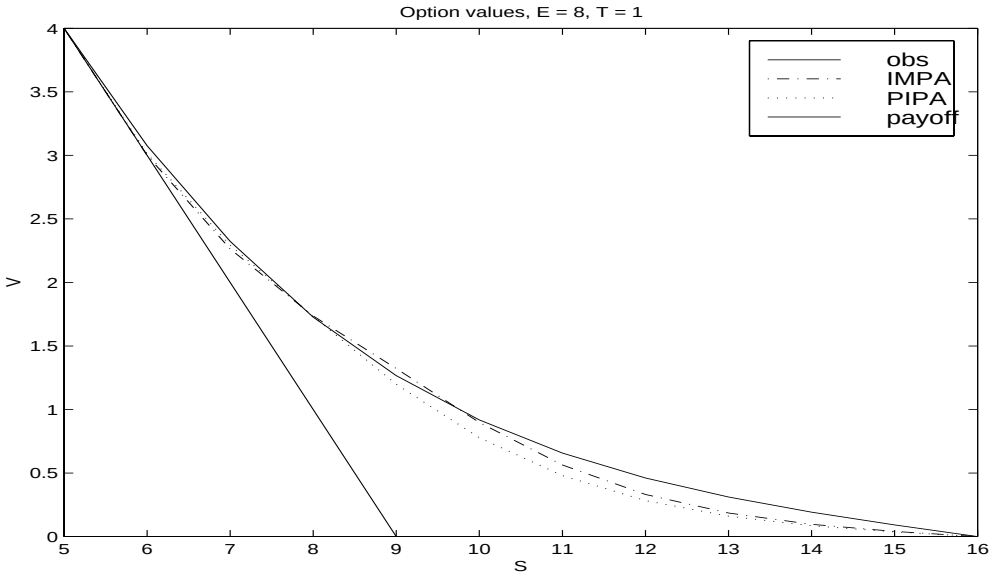

Figure 12: Example 3, options values calculated using constant/PIPA/IMPA volatility surfaces, E = 9

Clearly, $\boldsymbol{\sigma}(S,t) = \boldsymbol{\sigma}_0$ for all (S,t) is a feasible solution to the optimization problem (9). This gives an upper bound on the optimal objective value:

$$\theta_{\text{optimal}} \leq K(0.0707)^2 \equiv UB.$$

We repeat Examples 1–3 using this new objective function. Our goal is to investigate how well IMPA and PIPA perform in relation to this upper bound value UB, starting at $\boldsymbol{\sigma}_{\text{init}} = 0.255$. The results are summarized below. With the exception of PIPA on the 2-option example, in each case the algorithm is able to produce an objective value that is lower that the corresponding upper bound, in some cases much lower.

	PIPA	IMPA[a]	UB
2 options	2.5859e-02	8.8725e-07	4.9985e-03
11 options	1.5442e-02	4.3301e-03	2.7492e-02
14 options	1.0045e-02	6.9930e-04	3.4989e-02

[a] Maximum 80 iterations

5.5 On the issue of a smooth volatility surface

We emphasize that the above four sets of experiments are selected to demonstrate the viability of the PIPA and IMPA algorithms. The general methodology is applicable to any objective function $\theta(\boldsymbol{\sigma}, \boldsymbol{x})$ and any realistic constraint set $\boldsymbol{\Gamma}$. For computational simplicity, we have chosen some straightforward objective functions that do not contain any smoothing terms. Adding a smoothing term such as

$$|| \nabla \boldsymbol{\sigma} ||^2 = \int \left[\left(\frac{\partial \sigma}{\partial S} \right)^2 + \left(\frac{\partial \sigma}{\partial t} \right)^2 \right] dS dt$$

to the objective function will not compromise the validity of the method. Indeed, only the linear algebra involved need to be modified. The roughness of the surfaces plotted in Figures 1–10 is also partly due to the coarseness of the grid: $dS = 1$ and $dt = 0.125$. In addition, these plots are generated by the built-in MATLAB graphics routine, which we believe does not perform any type of high-order interpolation/smoothing on the data points. Obviously, if we refine the grid and employ some improved graphics routines, then the resulting surfaces can be expected to become "smoother". In what follows, we report the results of some attempts to obtain a smoother volatility surface by a straightforward modification of the PIPA and IMPA computer codes.

In running the above four examples, we have required the volatility to lie in a unit hypercube, i.e.

$$0 \leq \boldsymbol{\sigma} \leq \mathbf{1}\,.$$

These bounds may prove to be too loose. Indeed, if we tighten the bounds so that instead

$$0.001 \leq \boldsymbol{\sigma} \leq 0.5\,,$$

we expect the spikes on the surfaces will be less prominent. To verify, we re-run Example 3 using the same starting point $\boldsymbol{\sigma}_{\text{init}} = 0.255$ but with the tightened bounds on $\boldsymbol{\sigma}$. The surfaces are plotted in Figures 13 and 14.

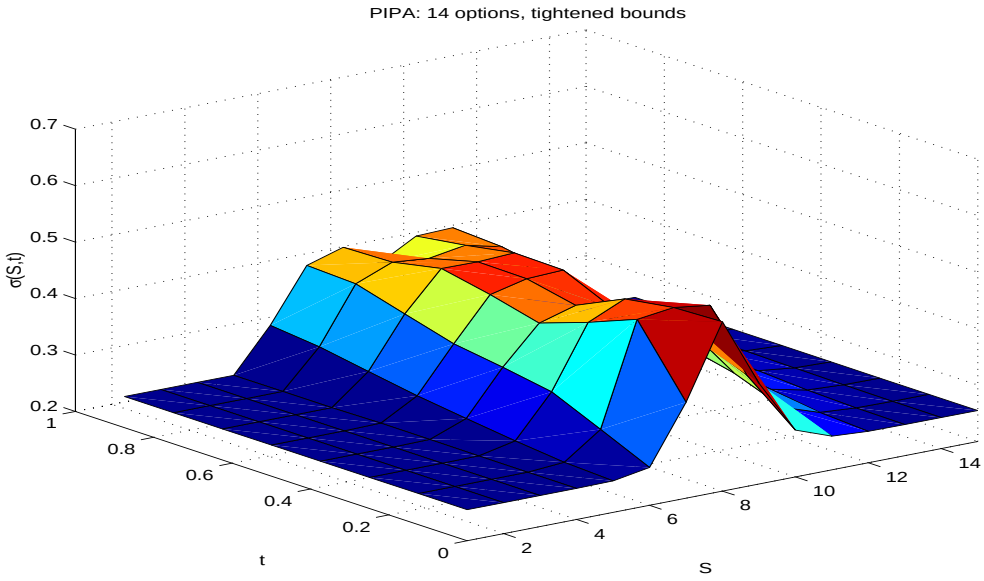

Figure 13: Example 3, volatility surface produced by PIPA with tightened bounds

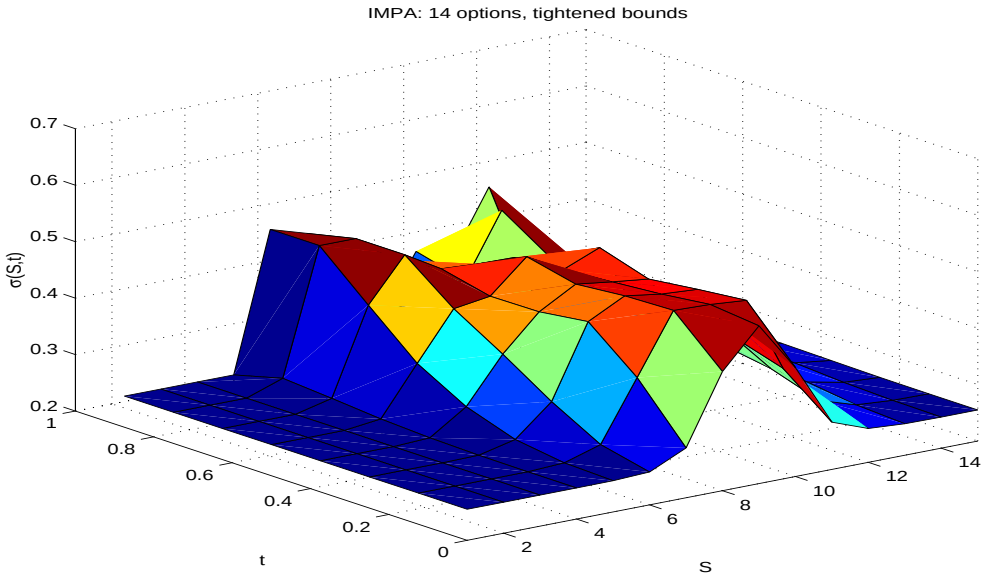

Figure 14: Example 3, volatility surface produced by IMPA with tightened bounds

	# iter	residual	$\|d\boldsymbol{\sigma}\|$	θ
PIPA	82	7.4863e-10	4.0431e-15	1.0476e-02

	# iter	$\|d\boldsymbol{\sigma}\|$	θ
IMPA	80	2.1531e-03	1.3280e-03

In addition, suppose it becomes reasonable to require the computed volatility surface to be close to a set of historical volatility data, we may incorporate this requirement into the objective function as well. Again, we re-run Example 3 using the same starting point as before but with the modified objective function

$$\theta(\boldsymbol{x}) \equiv \frac{1}{2}\sum_{k=1}^{K}\sum_{n\in\mathcal{N}_k}(\,V^k(n\delta S,0)-V_n^{\mathrm{obs},k}\,)^2+\frac{1}{2}\sum_{k=1}^{K}\sum_{(m,n)\in\mathcal{S}}(\sigma_{mn}-\sigma_{mn}^{\mathrm{his}})^2,$$

where $\mathcal{S}=\{1,\ldots,M-1\}\times\{1,\ldots,N-1\}$ and $\boldsymbol{\sigma}^{\mathrm{his}}=0.35$. The results are plotted in Figures 15 and 16.

	# iter	residual	$\|d\boldsymbol{\sigma}\|$	θ
PIPA	82	1.0941e-10	1.1360e-14	7.8159e-02

	# iter	$\|d\boldsymbol{\sigma}\|$	θ
IMPA	80	5.4872e-01	1.8489e-01

¿From the above set of tests, we see that the volatility surface has much less fluctuation if we impose more stringent conditions either by restricting the feasible region $\boldsymbol{\Gamma}$ or by including additional terms to the objective function. This also gives a good indication that should a smoothing term be included into the objective function, the volatility surface can be expected to be much less spiky.

6 Conclusion

We have introduced a new, mathematically sound approach to compute an implied volatility surface of American options. An inverse optimization problem that is an instance of an MPEC is formulated for computing this surface. Two different algorithms for solving the MPEC, namely PIPA and IMPA, are presented and tested. Numerical results demonstrate that these algorithms are able to reproduce the observed option values and meet the prescribed objectives closely.

A Computing the American option prices

The discretized American option pricing problem is a series of time-stepped LCPs. Many algorithms are available for solving linear complementarity problems. The algorithms fall into two general classes: iterative algorithms and pivotal algorithms. The former algorithms include the projected successive over-relaxation (PSOR) method and interior-point (IP) methods. Algorithms such as Lemke's and parametric principal pivoting are pivotal in nature. Because the matrix $\boldsymbol{Q}$ in LCP (4) may be of large order but sparse–it is typically tri-diagonal, we choose to use iterative algorithms

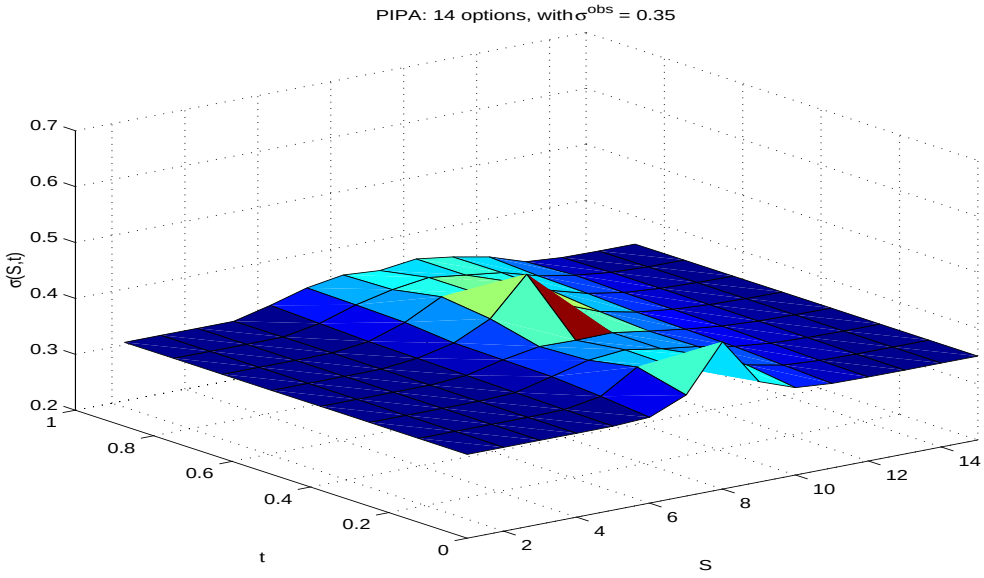

Figure 15: Example 3, volatility surface produced by PIPA with $\boldsymbol{\sigma}^{\text{his}} = 0.35$

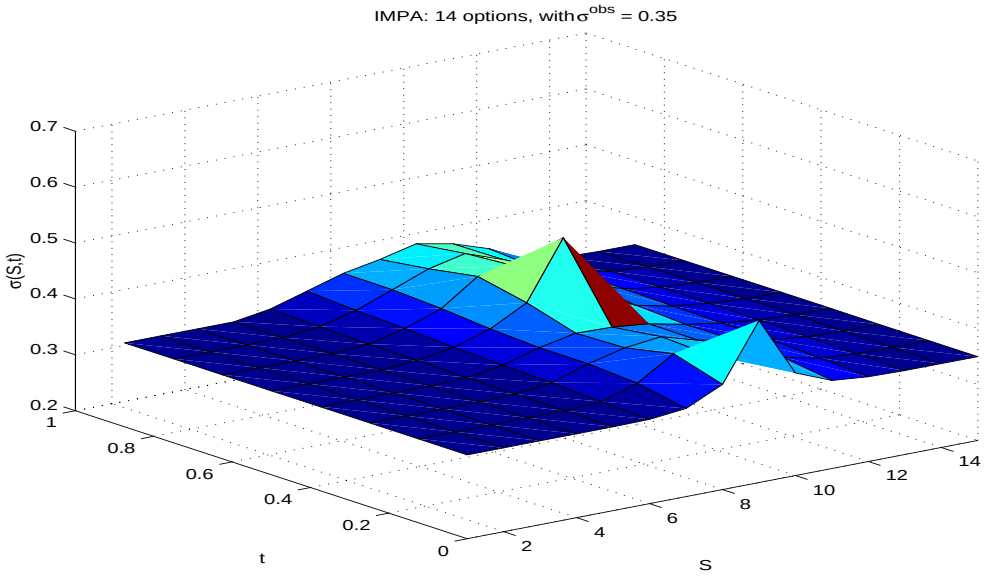

Figure 16: Example 3, volatility surface produced by IMPA with $\boldsymbol{\sigma}^{\text{his}} = 0.35$

over pivotal algorithms as the former can take advantage of the sparsity more effectively. As an experiment, both the PSOR algorithm and the interior point algorithm are implemented on the same set of test problems so that we can compare their relative performance. The implementations are done in MATLAB and are run on a SPARC ULTRA I/140. Table 1 summarizes the results. The parameters used for the tests are: $\delta S = 1$, $\delta t = 0.25$, $M = 4(T = 1)$, $r = 0.05$, $D_0 = 0.02$. Crank-Nicolson finite-difference is used with $\theta = 0.5$. We run both PSOR and IP for various sizes of the matrix $\boldsymbol{M}$, which is of order $(N-1)$. For IP, a termination criterion similar to (17) is imposed. For PSOR, the termination rule is

$$\text{Either } ||\min(\boldsymbol{V}^\nu, \boldsymbol{q} + \boldsymbol{Q}\boldsymbol{V}^\nu)||_\infty < 10^{-8}\,, \quad \text{or} \quad \text{max iteration } > 900\,.$$

The performance of the PSOR algorithm depends on the relaxation parameter ω. To further illustrate this, we run the PSOR algorithm for all test problems first with ω being fixed at 1.1 and then with a "best" ω value. (For simplicity, we run each problem for different values of $\omega \in (1, 2]$ at an increment of 0.1 and choose the best value from the set.) For PSOR, we report the average number of iterations required for each time step. As noted before, the option values produced by both algorithms agree almost exactly. Using the best ω, we report the "max diff" defined as

$$\text{max diff } = ||\boldsymbol{V}_0^{\text{sor}} - \boldsymbol{V}_0^{\text{ip}}||_\infty\,.$$

¿From the test results, we notice that when the size of the matrix is small, PSOR algorithm tends to work faster than the interior point algorithm; however, as the size of the matrix increases, the interior point algorithm is more robust than the PSOR algorithm. In particular, the interior point algorithm is more user friendly in the sense that it is not parameter dependent.

Acknowledgement. The authors are grateful to the Co-Editor Mark Broadie for helpful discussions on the topic of this paper and for several related references.

References

[1] M. Avellaneda, "Minimum-entropy calibration of asset-pricing models", *manuscript*, Courant Institute of Mathematical Sciences, New York University (1997).

[2] M. Avellaneda, C. Friedman, R. Holmes and D. Sampeir, "Callibrating volatility surfaces via relative entropy minimization", *Applied Mathematical Finance* 4 (1997) 37–64.

[3] S. Barle and N. Cakici, "Growing a smiling tree", in M. Broadie and P. Glasserman, eds., *Hedging with Trees: Advances in Pricing and Risk Managing Derivatives*, Risk Books (1998) pp. 173–178.

[4] S. Beckers, "Standard deviation implied in option prices as predictors of future stock price volatility", *Journal of Banking and Finance* 5 (1981) 363–381.

[5] F. Black and M. Scholes, "The pricing of options and corporate liabilities", *Journal of Political Economy* 81 (1973) 647–659.

[6] J.N. Bodurtha and M. Jermakyan, "Nonparametric estimation of an implied volatility surface", *The Journal of Computational Finance* 2 (1999) 29–60.

		IP		PSOR ($\omega = 1.1$)		PSOR (best ω)			
N	E	iter	CPU	iter	CPU	ω	iter	CPU	max diff
16	8	30	1.83	10	0.11	1.1	10	0.11	7.8353e-10
32	16	36	2.77	40	0.63	1.3	21	0.33	4.1985e-10
50	25	43	4.13	99	2.13	1.5	35	0.77	3.4150e-10
60	30	47	5.08	144	3.72	1.5	42	1.08	2.6963e-09
80	40	55	7.54	263	9.41	1.6	56	1.93	2.6029e-09
90	45	51	7.55	336	13.44	1.7	68	2.62	3.5666e-10
100	50	53	8.30	417	18.11	1.7	70	3.04	3.0367e-10
140	70	62	12.63	838	49.12	1.8	109	6.32	5.8987e-10
200	100	72	21.28	900[a]	NA	1.8	171	16.01	3.2441e-09
300	150	72	29.74	900[a]	NA	1.9	238	32.03	3.9989e-10
400	200	80	42.79	900[a]	NA	1.9	317	56.87	3.3456e-09
700	350	111	83.61	900[a]	NA	(1,2]	900[b]	NA	NA

[a] PSOR failed to converge after 900 iterations, $\omega = 1.1$

[b] PSOR failed to converge after 900 iterations for all values of $\omega \in (1, 2]$. This is the case when the matrix M fails to be strictly diagonally dominant: $\theta(r - D_0)N = 10.5 > 8.05$.

Table 1: Comparison of the PSOR algorithm and the interior point algorithm, $r = 0.05$, $D_0 = 0.02$, $dS = 1$, $T = 0.5$, $d\tau = 0.125$

[7] I. Bouchouev and V. Isakov, "The inverse problem of option pricing", *Inverse Problems* 13 (1997) L11–L17.

[8] N. Chriss, "Transatlantic trees", in M. Broadie and P. Glasserman, eds., *Hedging with Trees: Advances in Pricing and Risk Managing Derivatives*, Risk Books (1998) pp. 179–184.

[9] T. Coleman, Y. Li and A. Verma, "Reconstructing the unknown local volatility function", *The Journal of Computational Finance* 2 (1999) 77–102..

[10] R.W. Cottle, J.S. Pang, and R.S. Stone, *The Linear Complementarity Problem*, Academic Press, Boston (1992).

[11] M.A.H. Dempster and J.P. Hutton, "Fast numerical valuation of American, exotic and complex options", *Applied Mathematical Finance* 4 (1997) 1–20.

[12] M.A.H. Dempster and J.P. Hutton, "Pricing American stock options by linear programming", *Mathematical Finance*

[13] M.A.H. Dempster, J.P. Hutton, and D.G. Richards, "LP valuation of exotic American options exploiting structure", *Journal of Mathematical Finance* 2 (1998) 61–84.

[14] E. Derman and I. Kani, "Riding on a smile", *Risk* 7 (1994) 32–39.

[15] B. Dupuire, "Pricing with a smile", *RISK* 7 (1994) 18–20.

[16] R. Gibson, *Option Valuation: Analyzing and Pricing Standardized Option Contracts*, McGraw-Hill, New York (1991).

[17] J. Huang and J.S. Pang, "Option pricing and linear complementarity" *Journal of Computational Finance* 2 (1998) 31–60.

[18] J. Hull, *Options, Futures, and Other Derivative Securities*, Second Edition, Prentice Hall, New Jersey (1989).

[19] J. Hull and A. White, "The pricing of options on assets with stochastic volatilities", *The Journal of Finance* 42 (1987) 281–300.

[20] P. Jaillet, D. Lamberton, and B. Lapeyre, "Variational inequalities and the pricing of American options", *Acta Applicandae Mathematicae* 21 (1990) 263–289.

[21] H. Johnson and D. Shanno, "Option pricing when the variance is changing", *Journal of Financial and Quantitative Analysis* 22 (1987) 143–151.

[22] R. Lagnado and S. Osher, "A technique for calibrating derivative security pricing models", *Journal of Computational Finance* 1 (1997) 13–25.

[23] R. Lagnado and S. Osher, "Reconciling differences", *RISK* 10 (1997) 79–83.

[24] H.A. Latané and R.J. Rendleman, "Standard deviations of stock price ratios implied in option prices", *The Journal of Finance* 31 (1976) 369–381.

[25] Z.Q. Luo, J.S. Pang, and D. Ralph, *Mathematical Program with Equilibrium Constraints*, Cambridge University Press, Cambridge (1996).

[26] J.S. Pang, S.P. Han and N. Rangaraj, "Minimization of locally Lipschitzin functions", *SIAM Journal on Optimization* 1 (1991) 57–82.

[27] M. Rubinstein, "Implied binomial trees", *The Journal of Finance* 69 (1994) 771–818.

[28] C. Vazquez, "An upwind numerical approach for an American and European option pricing model", *Applied Mathematics and Computation* 97 (1998) 273–386.

[29] P. Wilmott, J.N. Dewynne and S. Howison, *Option Pricing: Mathematical Models and Computation*, Oxford Financial Press, Oxford (1993).

[30] T. Wang, R.D.C. Monteiro and J.S. Pang, "An interior point potential reduction method for constrained equations", *Mathematical Programming* (1996) 159–195.

[31] R. Zvan, P.A. Forsyth, and K.R. Vetzal, "Penalty methods for American options with stochastic volatility", *Journal of Computational and Applied Mathematics* 91 (1998) 199–218.