Finite difference methods for pricing American options.

Here we study finite difference methods for American options. We begin by displaying the model for these options and by arguing that, in contrast with the models for European options, it is a strongly non-linear one. We then consider the so-called obstacle problem in order to study in a simpler setting how to deal numerically with the same type of nonlinearity. There are mainly two approaches here: the one that uses variational inequalities and the one that uses viscosity solutions. Unfortunately, we cannot develop neither of these approaches here but we are going to point out the difficulties that those approaches would solve. Having done that for the obstacle problem, we come back to our original problem of using finite difference methods for solving American options.

1. The model for American options.

The only difference in the models of European and American options is that whereas for European options we require that

$$\frac{\partial}{\partial t} f + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2}{\partial S^2} f + rS\frac{\partial}{\partial S} f = r f,$$

for American options we require instead that

$$\frac{\partial}{\partial t} f + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2}{\partial S^2} f + rS\frac{\partial}{\partial S} f \le rf, \tag{1.1a}$$

$$f \geq \max\{K-S, 0\} \qquad \text{for put options,} \tag{1.1b)$_p$ ($$

$$f \geq \max\{S-K, 0\} \qquad \text{for call options,} \tag{1.1b)$_p$ ($$

and

$$\left(\frac{\partial f}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2}{\partial S^2} f + rS\frac{\partial}{\partial S} f - rf\right)\left(f - \max\{K-S, 0\}\right) = 0, \tag{1.1c)$_p$ ($$

$$\left(\frac{\partial f}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2}{\partial S^2} f + rS\frac{\partial}{\partial S} f - rf\right)\left(f - \max\{S-K, 0\}\right) = 0, \tag{1.1c)$_c$ ($$

and

$$f(\cdot, t) \in C^1(0, S_n) \qquad \text{for} \quad t \in (0,T). \tag{1.1d}$$

this difference reflects the fact that American options can be excercised at any time. When early exercise is not optimal, the equality holds in (1.1a) and does not in $(1.1b)_p$ or $(1.1b)_c$. We early exercise is optimal, the inequality holds in (1.1a) and equality holds in $(1.1b)_p$ or $(1.1b)_c$. the additional constraint (1.1d) <u>prevents arbitrage</u> and guarantees the uniqueness of the boundary, terminal condition problem for American options.

2. The obstacle problem.

A much simpler model which, however, displays similar features than the above model for American options is the obstacle problem. We are going to study it and with the knowledge gained, we are going to be able to derive finite difference methods for pricing American options.

The solution of the obstacle problem satisfies

$$-u'' \geq 0 \quad \text{in } (0,1), \tag{2.1a}$$

$$u - f \geq 0 \quad \text{in } (0,1), \tag{2.1b}$$

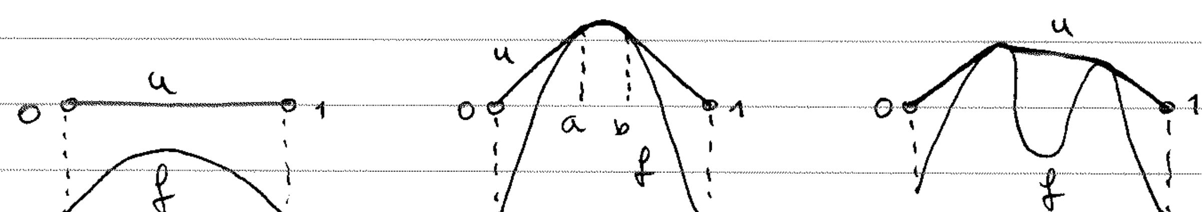
$$(-u'')(u-f) = 0 \quad \text{in } (0,1), \tag{2.1c}$$

$$u(0) = u(1), \tag{2.1d}$$

and

$$u \in C^1(0,1). \tag{2.1e}$$

Here the function f is the so-called obstacle. Below we sketch some solutions for various obstacles:



Let us consider the example in the middle: We see that $-u''=0$ and $u>f$ in $(0,a)$, that $-u''>0$ and $u=f$ in (a,b), and that $-u''=0$ and $u>f$ in $(b,1)$. Note also that we also have

$$u'(a) = f'(a),$$
$$u'(b) = f'(b),$$

in compliance with condition (2.1e). Note also that

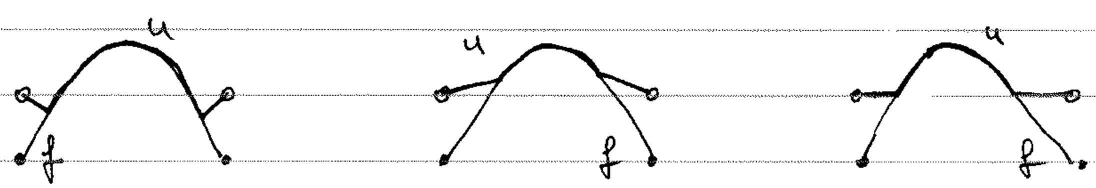
$$-u''(a^-) = 0$$
$$-u''(a^+) = -f''(a^+) > 0$$

and

$$-u''(b^+) = 0$$
$$-u''(b^-) = -f''(b^-) > 0.$$

This shows that $-u''$ has a discontinuity at $x=a$ and at $x=b$. We thus do not require $-u''$ to be defined at all points of $(0,1)$.

This could introduce a problem since we could then argue that the functions



are solutions of the obstacle problem. However, they are not because they do not satisfy the condition (2.1e)! In fact, if we think of the function u as describing an elastic string attached to the points (0,0) and (0,1), we can immediately accept that none of these three functions solves the obstacle problem.

When devising numerical schemes for the obstacle problem, we must take special care in making sure that a discrete version of the condition (2.1e) is satisfied. This is the most difficult aspect of the task.

3. A finite difference method for the obstacle problem.

Let us consider the following discrete version of the equations (2.1a), (2.1b), (2.1c) and (2.1d):

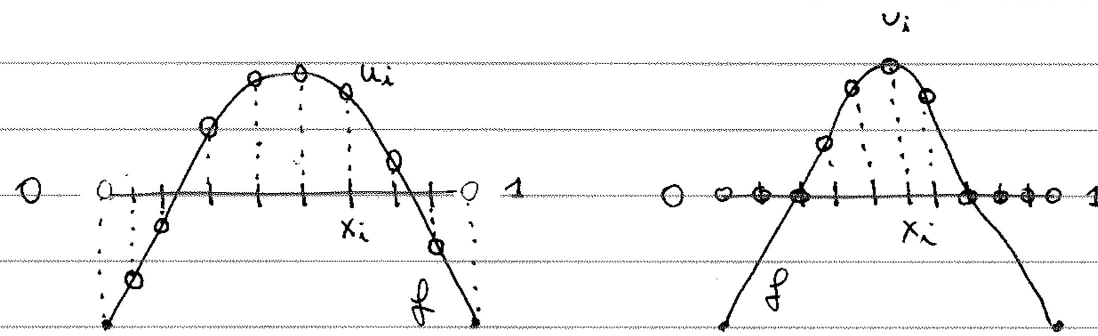
(3.1a) $$-\frac{1}{\Delta x^2}(u_{i+1} - 2u_i + u_{i-1}) \geq 0 \qquad i = 1, \dots, I-1,$$

(3.1b) $$u_i - f_i \geq 0 \qquad i = 1, \dots, I-1,$$

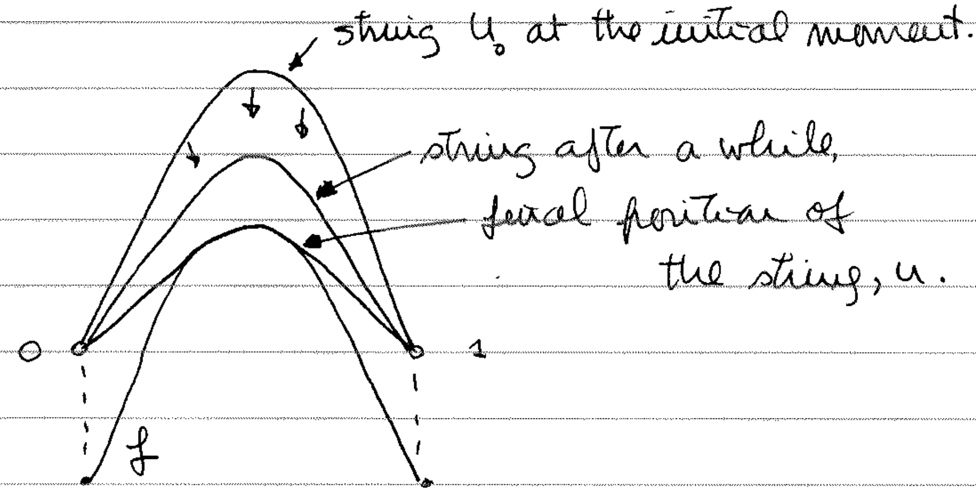
(3.1c) $$\left(-\frac{1}{\Delta x^2}(u_{i+1} - 2u_i + u_{i-1})\right)(u_i - f_i) = 0 \qquad i = 1, \dots, I-1,$$

(3.1d) $$u_0 = u_I = 0.$$

We can immediately see the difficulty one has in dealing with the smoothness condition (2.1e). Indeed, by using finite difference methods such a condition is essentially impossible to enforce directly. We are going to have to use an indirect way to do that in order to avoid having the wrong approximations like the ones below



The indirect way to make sure that we are going to converge to a function satisfying (2.1c) is inspired in our physical interpretation of u as an elastic string. We know that if we pull the string and than we let it go, we are going to see the string evolve in time as in the picture below:



thus, we are going to let "evolve" u_0 in time in such a way that its evolution should stop if (2.1c) is satisfied. Since any function satisfying

$$\max\{f-u,\ r u''\}=0,$$

where r is a positive parameter, satisfies equations (2.1a), (2.1b) and (2.1c), we are going to consider the following evolution problem

(3.2a) $$\frac{\partial}{\partial \tau}u = \max\{f-u,\ r\frac{\partial^2}{\partial x^2}u\} \qquad \text{in } (0,1)\times(0,\infty),$$

(3.2b) $$u(x,0) = u_0(x) \qquad \forall x\in(0,1),$$

(3.2c) $$u(0,t) = u(1,t)=0 \qquad \forall t\in(0,\infty).$$

Here, we take u_0 such that

(3.3) $$f(x)-u_0(x) < 0 \qquad \forall x\in(0,1).$$

Note that

$$\frac{\partial}{\partial \tau}\left(u(x,\tau)-f(x)\right) = \frac{\partial}{\partial \tau}u(x,\tau)$$

$$\geq f(x)-u(x,\tau),$$

and so, setting $\varphi(\tau) = u(x,\tau)-f(x)$, we get that

$$\frac{d}{d\tau}\varphi + \varphi \geq 0.$$

This implies that

$$\varphi(\tau) \geq e^{-\tau}\varphi(0),$$

or, in other words,

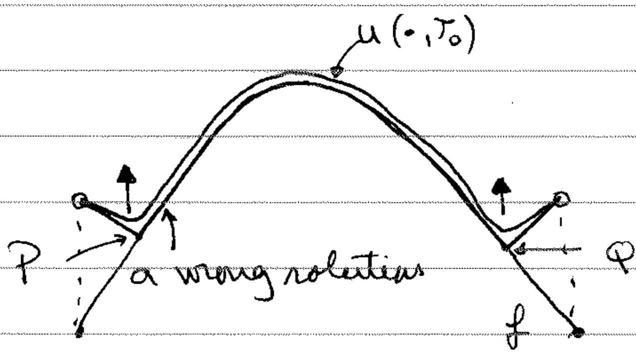
$$u(x,\tau) - f(x) \geq e^{-\tau}\left(u_0(x) - f(x)\right) \qquad \forall \tau > 0$$
$$\forall x \in (0,1).$$

We thus see that the constraint

$$u(x,\tau) - f(x) > 0 \qquad \forall x \in (0,1),$$

is satisfied for all $\tau > 0$. This means that with this choice of initial guess, the "string" $u(\cdot,\tau)$ <u>never</u> <u>touches</u> the obstacle!

Let us argue that this prevents $u(\cdot,\tau)$ to get close to any "wrong" solution of the obstacle problem as τ goes to infinity. Indeed, suppose that at some given time τ_0, we have the following situation:



Since at the points P and Q, $u(\cdot,\tau_0)$ must have a huge positive second-order derivative, we must also have that $u(\cdot,\tau_0)$ around those points will increase since we would then have

$$\frac{\partial}{\partial \tau} u = \gamma \frac{\partial^2}{\partial x^2} u > 0$$

thus $u(\cdot, \tau)$ for $\tau > \tau_0$ will be farther away from the wrong solution around P and Q. In other words, as $\tau \to \infty$, $u(\cdot,\tau)$ will not converge to an incorrect solution of the obstacle problem. In this way, we indirectly imposed the regularity condition (2.1e).

Now, let us discretize the problem (3.2) as follows:

(3.4a) $$\frac{1}{\Delta\tau}\left(u_i^{n+1} - u_i^n\right) = \max\left\{ f_i - u_i^n, \frac{\gamma^n}{\Delta x^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right)\right\}$$

for $i = 1, \dots, I-1$, $n \geq 0$,

(3.4b) $u_0^{n+1} = U_I^{n+1} = 0$ for $n \geq 0$,

(3.4c) $u_i^0 = u_0(i\,\Delta x)$ for $i = 0, \dots, I$.

Note that we have that (3.4a) can be rewritten as

$$u_i^{n+1} = u_i^n + \Delta\tau \max\left\{ f_i - u_i^n, \frac{\gamma^n}{\Delta x^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right)\right\}$$

$i = 1, \dots, I-1$, $n \geq 0$.

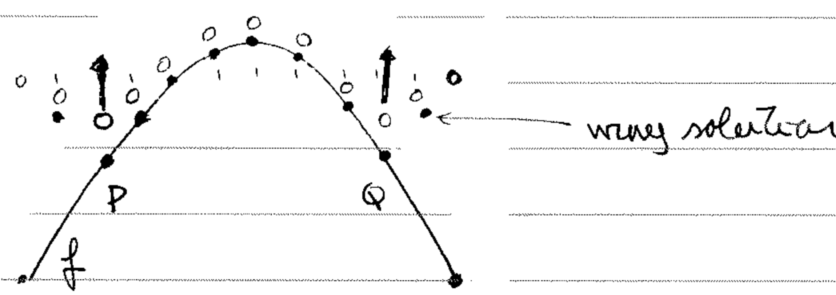
As a consequence

$$\begin{aligned}(u_i^{n+1} - f_i) &\geq (u_i^n - f_i) + \Delta\tau\,(f_i - u_i^n) \\ &= (1-\Delta\tau)\,(u_i^n - f_i)\end{aligned}$$

and, for $\Delta\tau \in [0,1]$, we obtain that

$$(u_i^n - f) \geq (1-\Delta\tau)^n \,(u_0(i\Delta x) - f_i) \qquad \forall n \geq 0.$$

As a consequence the constraint (3.1b) is satisfied by all the iterates $\{u_i^n\}_{i=0}^{I}$ and none of the iterates ever touches the obstacle for $\Delta\tau \in [0,1)$. Again, this prevents convergence to an incorrect solution as we see in the figure below.



Indeed, at the points P and Q the approximate solution will increase.

Next, let us show that the scheme is consistent, that is, that the exact solution $\{u_i\}_{i=0}^{I}$ satisfies it. It is indeed not difficult to see that the sequence

$$\left\{ \{u_i^n := u_i\}_{i=0}^{I} \right\}_{n\geq 0}$$

does satisfy the numerical scheme (3.4). The scheme is thus consistent.

Now let us show that the method converges to a solution of (3.4) under some special conditions.

Set

$$\Delta_i^n = \frac{1}{\Delta x^2}\left(u_{i+1}^n - 2u_i^n + u_{i-1}^n\right).$$

then, for $i = 2, \ldots, I-2$,

$$\Delta_i^{n+1} = \Delta_i^n + \frac{\Delta\tau}{\Delta x^2}\Big(\max\{f_{i+1} - u_{i+1}^n,\ \gamma^n \Delta_{i+1}^n\} - 2\max\{f_i - u_i^n,\ \gamma^n \Delta_i^n\} + \max\{f_{i-1} - u_{i-1}^n,\ \gamma^n \Delta_{i-1}^n\}\Big)$$

Since $f_{i+1} - u_{i+1}^n \leq 0$ and $f_{i-1} - u_{i-1}^n \leq 0$,

$$\Delta_i^{n+1} \leq \Delta_i + \frac{\Delta\tau}{\Delta x^2}\Big(\max\{0,\ \gamma^n \Delta_{i+1}^n\} - 2\gamma\Delta_i^n + \max\{0,\ \gamma^n \Delta_{i-1}^n\}\Big)$$

$$= \left(1 - 2\frac{\Delta\tau}{\Delta x^2}\gamma\right)\Delta_i^n + \frac{\Delta\tau}{\Delta x^2}\max\{0, \gamma^n \Delta_{i+1}^n\} + \frac{\Delta\tau}{\Delta x^2}\max\{0, \gamma^n \Delta_{i-1}^n\}$$

and <u>if we assume</u> that $\Delta_i^n \leq 0$ for $i = 1, \ldots, I$, we obtain that

$$\Delta_i^{n+1} \leq \left(1 - 2\frac{\Delta t}{\Delta x^2}\gamma^n\right)\Delta_i$$

and we get that

$$\Delta_i^{n+1} \le 0 \qquad \text{for } i = 2, \dots, I-2,$$

provided $\frac{\Delta t}{\Delta x^2} r \le \frac{1}{2}$.

Now let us consider Δ_1^{n+1}:

$$\Delta_1^{n+1} = \Delta_1^n + \frac{\Delta T}{\Delta x^2} \Big(\max\{ f_2 - u_2^n, r\,\Delta_2^n \} \\ - 2\max\{ f_1 - u_i^n, r\,\Delta_1^n \} \Big),$$

and by a similar argument, we get that

$$\Delta_1^{n+1} \le 0$$

provided $\frac{\Delta T}{\Delta x^2} r \le \frac{1}{2}$. We can also show that $\Delta_{I-1}^{n+1} \le 0$ under the same condition.

Thus, if we take $\frac{\Delta T}{\Delta x^2} r \le 1/2$, then

$$\Delta_i^{n+1} \le 0 \qquad i = 1, \dots, I-1,$$

provided the same is true for "n+1" replaced by "n".

So, we have established that

$$f_i - u_i^n \leq 0 \qquad i = 0, \dots, I, \quad n \geq 1,$$
$$\Delta_i^n \leq 0 \qquad i = 1, \dots, I-1 \quad n \geq 1,$$

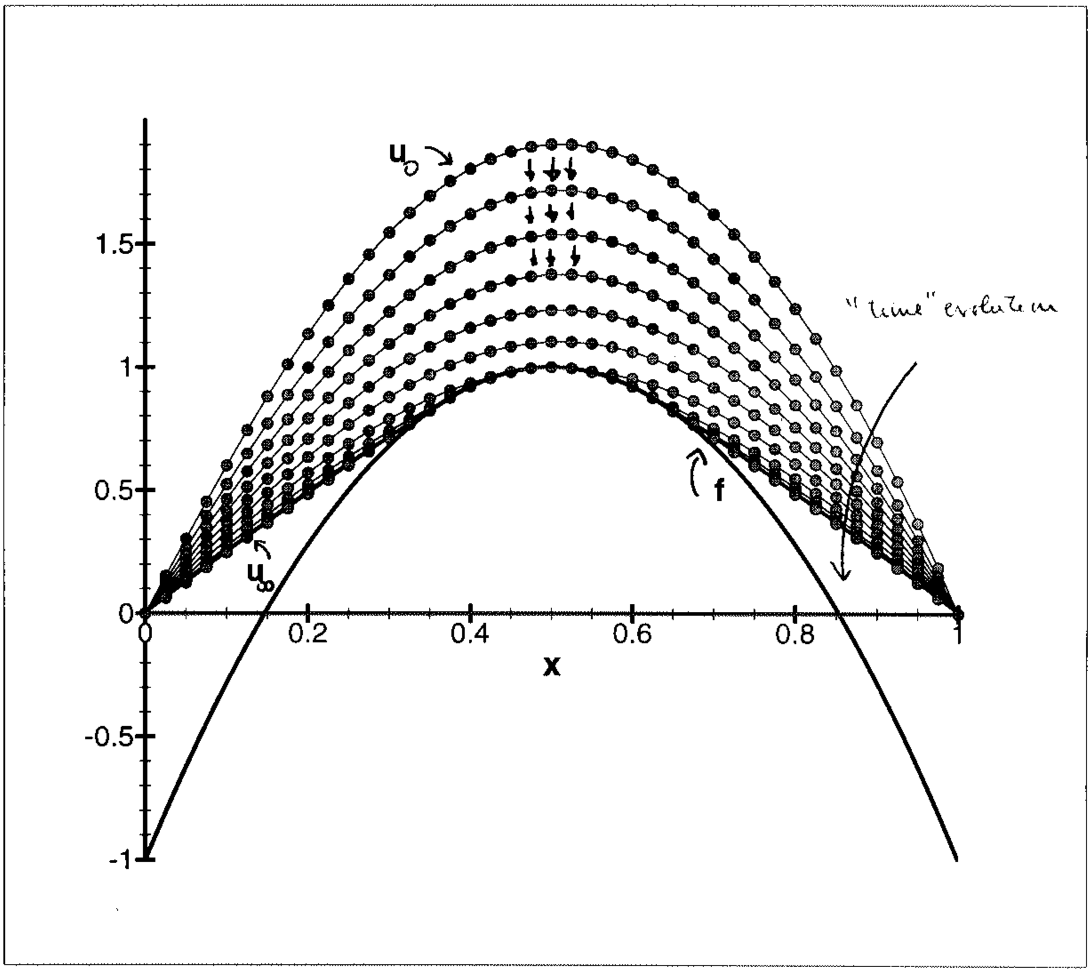
provided $\Delta\tau \in [0,1)$ and $\frac{\Delta\tau}{\Delta x^2} \gamma \leq \frac{1}{2}$, and provided $\Delta_i^0 \leq 0$ for $i = 1, \dots, I-1$.

This implies that

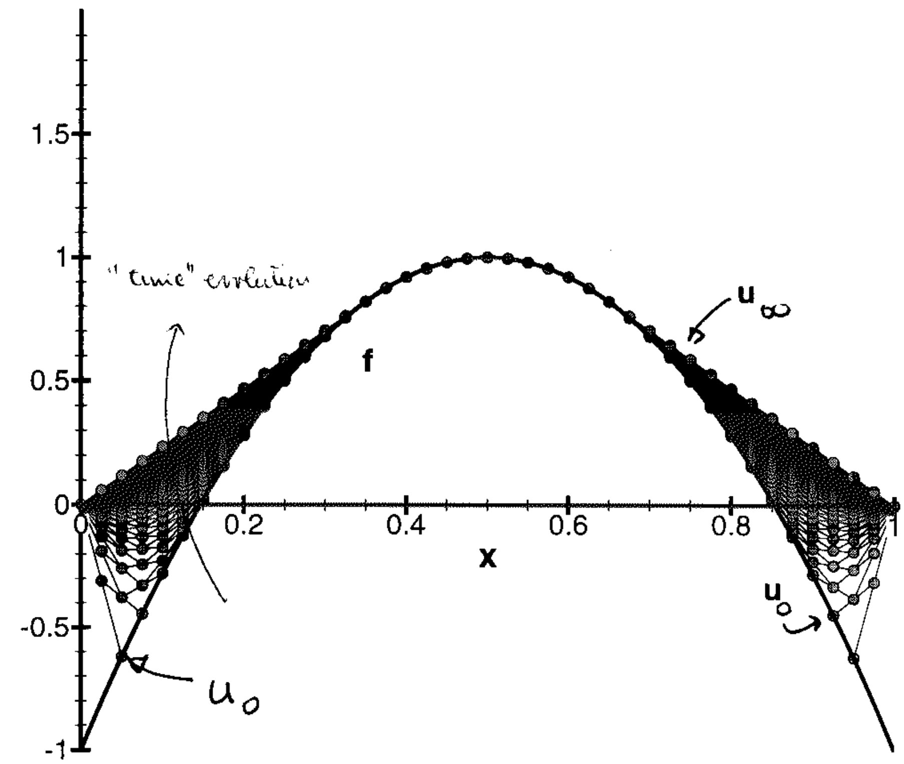
$$u_i^{n+1} \leq u_i^n \qquad \text{for } i = 0, \dots, I \text{ and } n \geq 1.$$

Hence $\{u_i^n\}_{n \geq 1}$ is a non-increasing sequence bounded below by f_i, for each $i = 0, \dots, I$. This implies that it converges to a solution of the obstacle problem!

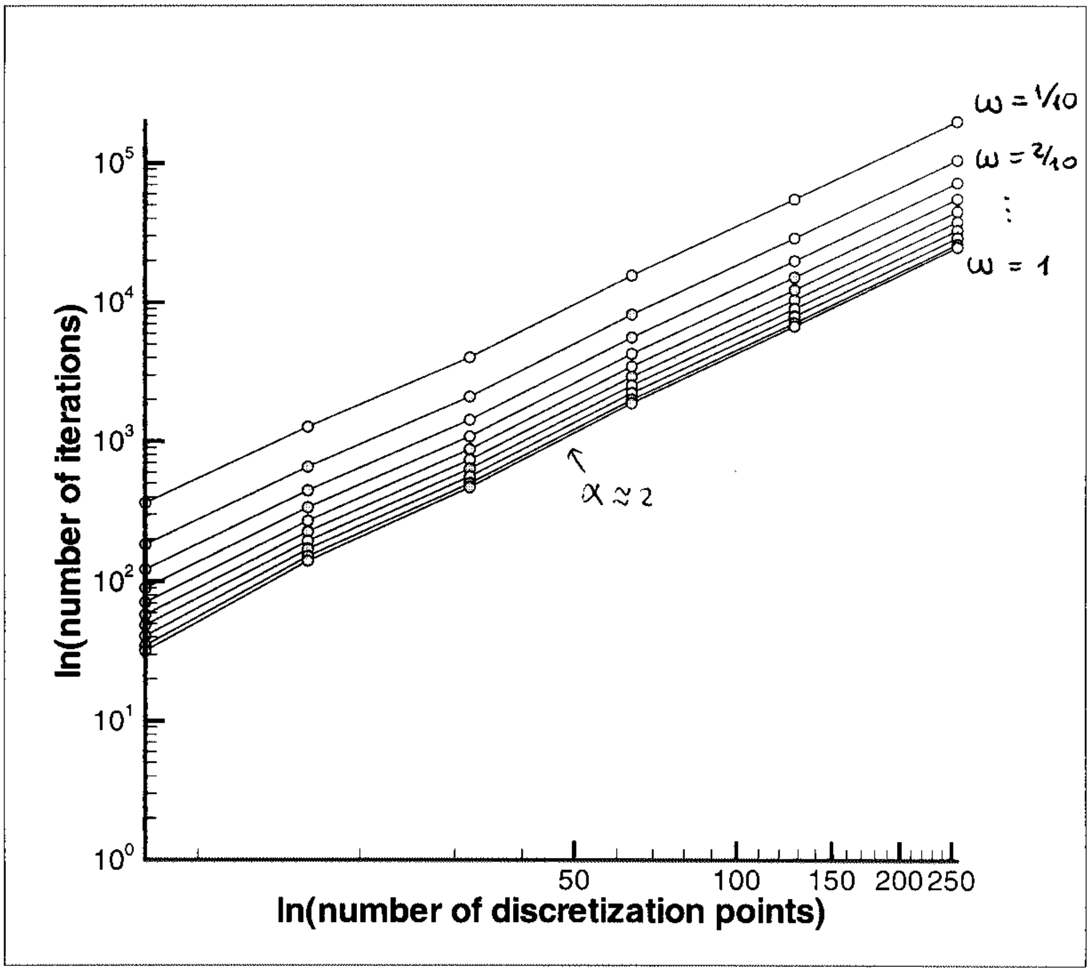
Note that we have not proved that it converges to the correct approximation of the solution of (2.1). To prove this, we must use the theory of viscosity solutions.



This is an example of the application of the method (3.4) for $f(x) = 1 - 8\,(x - \frac{1}{2})^2$. Here we are taking $\Delta T = 1$ and $\frac{\Delta T\,\nu}{\Delta x^2} = \frac{1}{2}$. The initial guess u_0 satisfies the condition $u_0 \geq f$ and $-u_0'' \geq 0$, and so we are under the assumptions that allow us to conclude that the sequence $\{u_i^n\}_{i=0}^{I}$ converges to the solution, here indicated by u_∞, of the discrete obstacle problem (3.1). Note the monotonic evolution of the iterates $\{u_i^n\}_{i=0}^{I}$ as n increases. Note also how the evolution is stopped whenever $u_i^n \cong f_i$ and $\Delta_i^n \approx 0$. The number of discretization points is 40.



Here we use the same scheme as used in the previous page. The method does converge to the correct solution of (3.1) even though our assumptions on u_0 <u>are not</u> satisfied.



Here we plot the number of iterations for the algorithm (3.4) to converge, "N", versus the number of discretization points, "I". We say that we reached convergence if the distance between two consecutive iterates is less than 10^{-10} on each of the discretization points. We are taking $\Delta T = 1$ and various values of $\omega := 2 \frac{\Delta T}{\Delta x^2} \gamma$, namely $i/10$ for $i = 1, \ldots, 10$. We see that the best choice of ω is $\omega = 1$. We also see that there is a linear relation between the numbers we are plotting. So, it is reasonable to conclude that

$$N \cong C(\omega)\, I^{\alpha}.$$

Since the slope of the curves remain invariant as ω changes, it is reasonable to take "α" as a constant.

the data plotted in the previous figure is contained in the Table below.

Ted.

$I \backslash \omega$	$\frac{1}{10}$	$\frac{2}{10}$	$\frac{3}{10}$	$\frac{4}{10}$	$\frac{5}{10}$	$\frac{6}{10}$	$\frac{7}{10}$	$\frac{8}{10}$	$\frac{9}{10}$	1
8	362	183	122	90	71	58	49	41	35	32
16	1267	655	444	336	270	226	194	170	151	140
32	4011	2087	1422	1082	875	735	634	558	498	467
64	15553	8145	5572	4255	3450	2906	2513	2216	1983	1869
128	54761	28800	19752	15101	12268	10347	8958	7906	7080	6699
256	197501	104564	71975	55187	44897	37923	32874	29044	26036	24772

From the data, we can obtain that $\alpha \simeq 2$. this is consistent with our experience when dealing with the Jacobi method (obtained when $\omega = 1$!) when there is no obstacle.

4 The projected SOR method for computing $\{u_i\}_{i=0}^{I}$.

The projected SOR method for computing a solution of (3.1) is strongly related to the method (3.4) we just analyzed. Indeed, if we take $\Delta\tau = 1$ and $\frac{\nu^n}{\Delta x^2} = \frac{\omega}{2}$, and replace u_{i+1}^n by u_{i+1}^{n+1}, we obtain the so-called projected SOR method:

(4.1a) $$u_i^{n+1} = \max\{f_i, u_i^n + \frac{\omega}{2}(u_{i+1}^{n+1} - 2u_i^n + u_{i-1}^n)\} \qquad i = 1,\dots,I-1, \quad n \geq 0,$$

(4.1b) $$u_0^{n+1} = u_I^{n+1} = 0 \qquad n \geq 0,$$

(4.1c) $$u_i^0 = u_0(i\Delta x) \qquad i = 0,\dots,I.$$

Note that, unlike the previous algorithm, once we have computed the iterate $\{u_i^n\}_{i=0}^{I}$, we have to proceed from "right to left" to calculate $\{u_i^{n+1}\}_{i=0}^{I}$.

Let us now study the convergence properties of this method. It is very easy to see that the method is consistent and so if it converges, it converges to a solution of the discrete obstacle problem (3.1).

To find out under what circumstances we can guarantee its convergence, let us proceed as we did to analyze algorithm (3.4).

So, we begin by noting that by (4.1a), we have

(4.2a) $$u_i^{n+1} \geq f_i \qquad \text{for} \quad i = 1, \ldots, I-1 \quad \text{and} \quad n \geq 0;$$

this happens independently of the choice of initial guess u_0!

Next, let us rewrite (4.1a) in a way more convenient for analysis. Thus, setting

$$\Delta_i^n = u_{i+1}^{n+1} - 2u_i^n + u_{i-1}^n \qquad \text{for} \quad i = 1, \ldots, I-1,$$

we can rewrite (4.1a) as follows:

(4.3) $$u_i^{n+1} = u_i^n + \max\left\{ f_i - u_i^n, \frac{\omega}{2} \Delta_i^n \right\} \quad \text{for} \quad i = 1, \ldots, I-1.$$

Let us assume that our initial guess is such that

(4.2b) $$u_{i+1}^0 - 2u_i^0 + u_{i-1}^0 \leq 0 \qquad \text{for} \quad i = 1, \ldots, I-1.$$

Next, we prove that thanks to condition (4.2b), we can find values of ω for which we have that

$$\Delta_i^n \leq 0 \qquad \text{for } i=1,\dots,I-1 \text{ and } n \geq 0.$$

Let us begin by considering the case $n=0$. We then have that, for $i = I-1$,

$$\begin{aligned}\Delta_i^0 &= u_I^1 - 2u_{I-1}^0 + u_{I-2}^0 \\ &= 0 - 2U_{I-1}^0 + U_{I-2}^0\end{aligned}$$

by the boundary condition (4.1b), and so

$$\Delta_i^0 = U_I^0 - 2U_{I-1}^0 + U_{I-2}^0 \leq 0$$

by (4.2b). For $i = I-2,\dots,1$, we have

$$\begin{aligned}\Delta_i^0 &= u_{i+1}^1 - 2u_i^0 + u_{i-1}^0 \\ &= (u_{i+1}^0 - 2u_i^0 + u_{i-1}^0) + \max\{f_{i+1} - U_{i+1}^0, \tfrac{\omega}{2}\Delta_{i+2}^0\}. \\ &\leq \max\{f_{i+1} - U_{i+1}^0, \tfrac{\omega}{2}\Delta_{i+2}^0\} \\ &\leq \max\{0, \tfrac{\omega}{2}\Delta_{i+2}^0\},\end{aligned}$$

by (4.2). We thus see that $\Delta_i^0 \leq 0$ provided $\Delta_{i+1}^0 \leq 0$. Since $\Delta_{I-1}^0 \leq 0$, a simple induction argument shows that we do have that

$$\Delta_i^0 \leq 0 \qquad \text{for } i = 1, \ldots, I-1, \tag{4.4}$$

provided $\omega \geq 0$. Next we show that if $\omega \in [0,1]$, we also have

$$u_{i+1}^1 - 2u_i^1 + u_{i-1}^1 \leq 0 \qquad \text{for } i = 1, \ldots, I-1. \tag{4.5}$$

For $i = 2, \ldots, I-1$, we have, by (4.3),

$$\begin{aligned} u_{i+1}^1 - 2u_i^1 + u_{i-1}^1 &= u_{i+1}^0 - 2u_i^0 + u_{i-1}^0 - 2\max\{f_i - u_i^0, \tfrac{\omega}{2}\Delta_i^0\} \\ &\qquad + \max\{f_{i-1} - u_{i-1}^0, \tfrac{\omega}{2}\Delta_{i-1}^0\} \\ &= \Delta_i^0 - 2\max\{f_i - u_i^0, \tfrac{\omega}{2}\Delta_i^0\} \\ &\qquad + \max\{f_{i-1} - u_{i-1}^0, \tfrac{\omega}{2}\Delta_{i-1}^0\}. \end{aligned}$$

By (4.4), and since we are assuming that $f_i - u_i^0 \leq 0$ for $i = 1, \ldots, I-1$, we get

$$\begin{aligned} u_{i+1}^1 - 2u_i^1 + u_{i-1}^1 &\leq \Delta_i^0 - 2\max\{f_i - u_i^0, \tfrac{\omega}{2}\Delta_i\} \\ &\leq (1-\omega)\Delta_i^0 \\ &\leq 0 \end{aligned}$$

if $\omega \in [0,1]$. For $i = 1$, we have, by (4.1c) and (4.1b),

$$u_2^1 - 2u_1^1 + u_0^1 = u_2^1 - 2u_1^1 + u_0^0$$

$$
\begin{aligned}
&= u_2^1 - 2u_1^0 + u_1^0 - 2\max\{f_1 - u_1^0, \tfrac{\omega}{2}\Delta_1^0\} \\
&= \Delta_1^0 - 2\max\{f_1 - u_1^0, \tfrac{\omega}{2}\Delta_1^0\} \\
&\geq (1-\omega)\Delta_1^0 \\
&\geq 0
\end{aligned}
$$

if $\omega \in [0,1]$.

So we have shown that (4.5) holds. The same argument can be used to show that if

$$u_{i+1}^n - 2u_i^n + u_{i-1}^n \leq 0 \qquad \text{for} \quad i = 1, \dots, I-1,$$

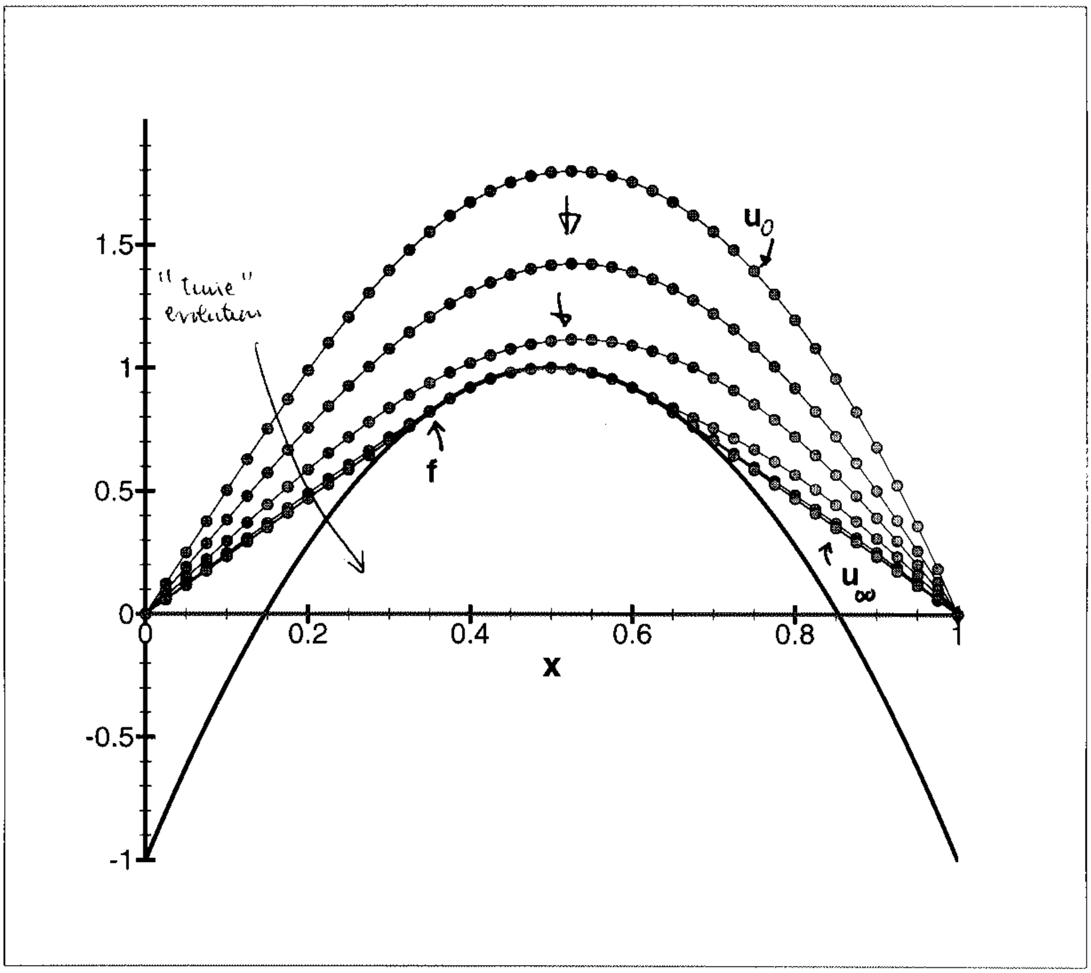
then

$$\Delta_i^n \leq 0 \text{ and } u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1} \leq 0 \quad \text{for} \quad i = 1, \dots, I-1.$$

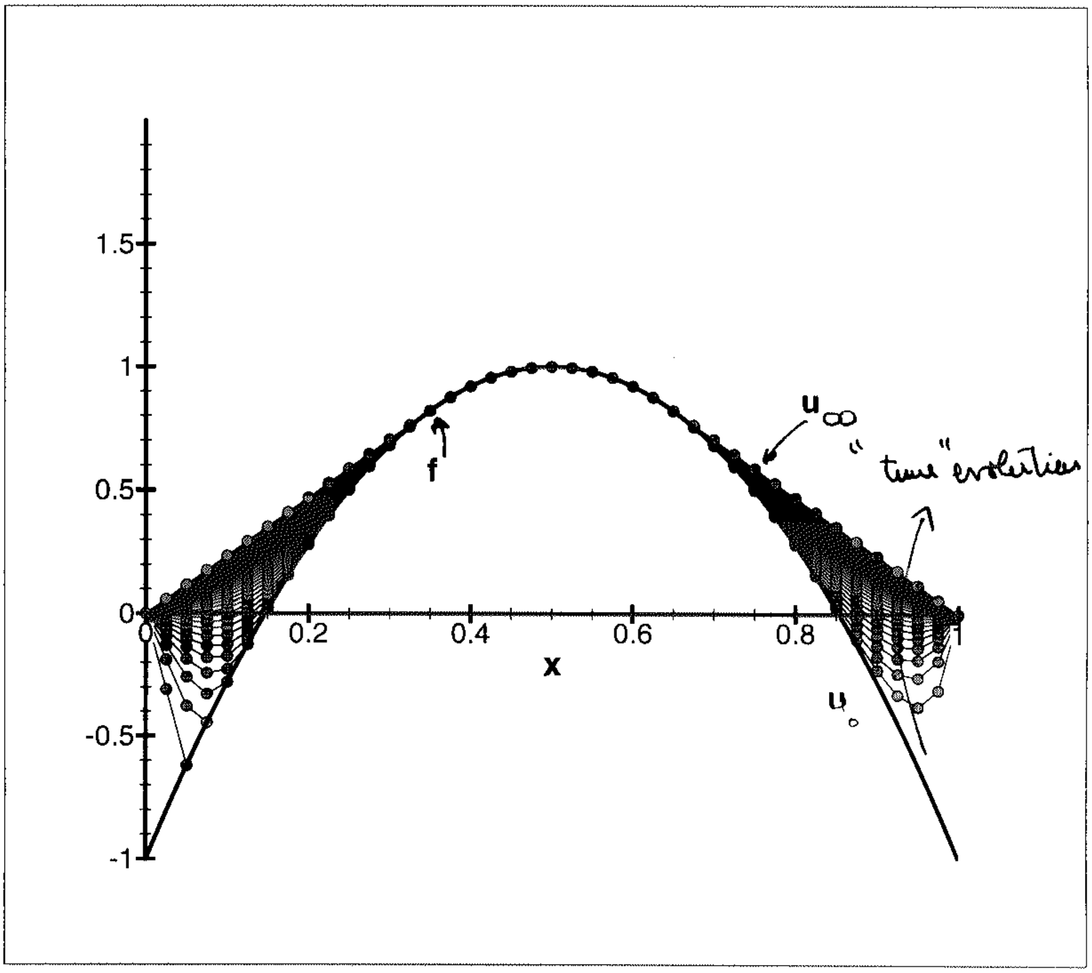
This implies that the sequence $\{u_i^n\}_{n\geq 0}$ is nonincreasing and bounded below since

$$f_i \leq u_i^{n+1} = u_i^n + \max\{f_i - u_i^n, \tfrac{\omega}{2}\Delta_i^n\} \leq u_i^n.$$

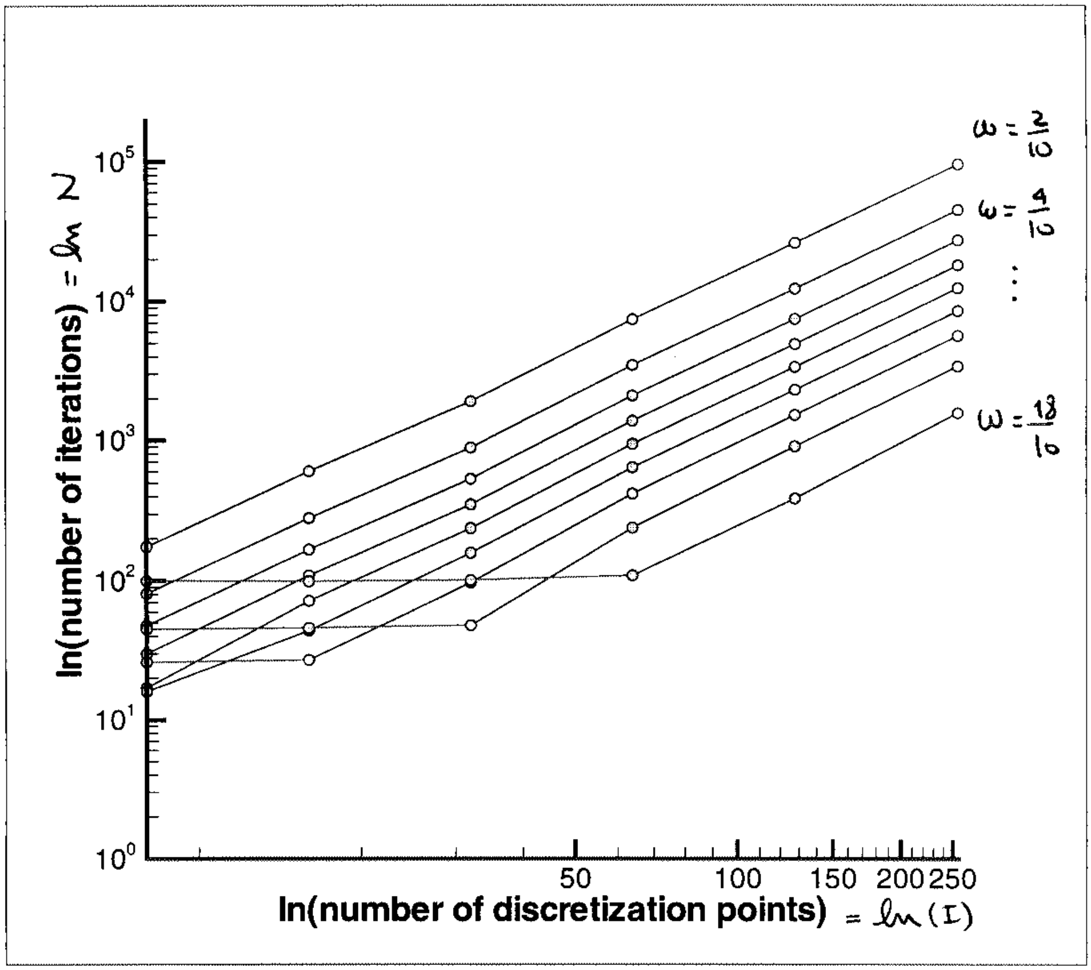
As a consequence, it converges to a limit we denote by u_i. By consistency, $\{u_i\}_{i=0}^{I}$ is a solution of (3.1).



Here we apply the projected SOR with $\omega = 1$ to the obstacle problem (3.1) with $f(x) = 1 - 8(x - \frac{1}{2})^2$. The initial guess satisfies the conditions under which we proved convergence of the method. We see here the monotonic convergence of the iterates, as predicted by our theoretical analysis. Moreover, we see that the method seems to converge to the "correct" solution of (3.1).



Here we show that the projected SOR with $\omega = 1$ converges even if the initial guess v_0 does not satisfy the conditions of our theoretical analysis. To deal with these, more general situation, the more complicated theory of monotone schemes for viscosity solutions would have needed to be used.



I \ ω	$\frac{2}{10}$	$\frac{4}{10}$	$\frac{6}{10}$	$\frac{8}{10}$	1	$\frac{12}{10}$	$\frac{14}{10}$	$\frac{16}{10}$	$\frac{18}{10}$
8	175	81	48	30	17	16	26	45	100
16	602	280	167	109	72	44	27	46	99
32	1900	886	532	350	237	158	97	48	101
64	7394	3463	2089	1379	941	690	417	239	109
128	26132	12286	7429	4914	3361	2297	1514	904	386
256	94932	44926	27271	18099	12409	8504	5629	3395	1566

this is the table of number "N" plotted in the figure above. Here we say we have converged if the distance between two iterates is less than 10^{-10} at every discretization point.

We can see that for I "large", the relation between ln(N) and ln(I) is linear and that slope is the same. So, we can conclude that it is reasonable to have

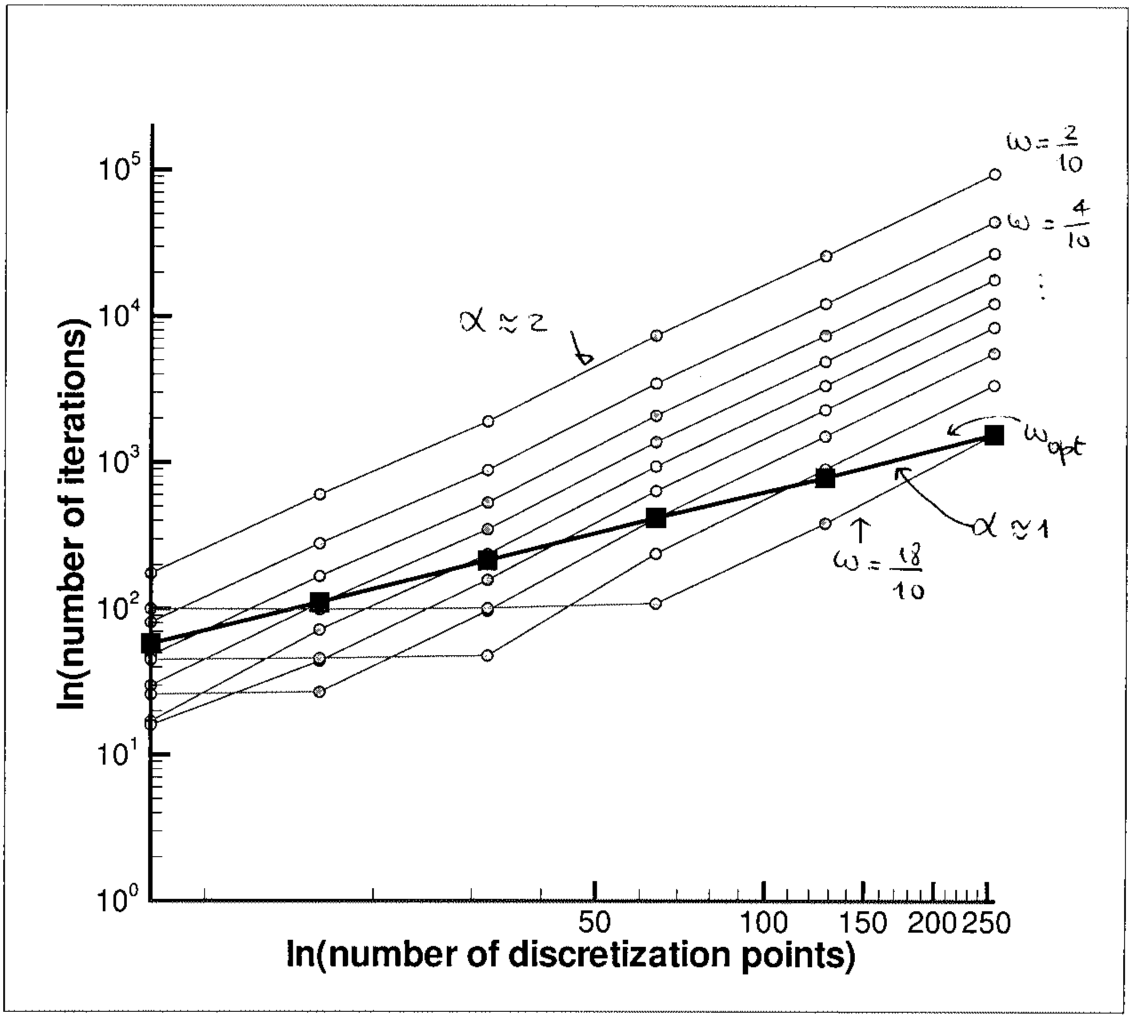
$$N = C(\omega)\, I^{\alpha},$$

where $C(\omega)$ increases as ω increases. We also see that the projected SOR converges for $\omega \in (0,2)$, a result similar to that of the SOR for the same problem without the obstacle. In particular, this means that on that region we have that $\alpha \approx 2$.
Finally, we can see, by comparing the data of the projected SOR algorithm and that of our previous method, that for the same value of ω, the projected SOR is more efficient. For example, for $\omega = 1$ and $n = 256$, the projected SOR takes 12,404 iterations whereas the other method takes 24,772 or twice as many.

If we now take ω as the "optimal" parameter, that is, as

$$\omega = \frac{2}{1 + \sin \frac{\pi}{2I}} \quad \text{(see the notes on iterative methods)},$$

we obtain that $N = C\, I^{\alpha}$ where $\alpha \approx 1$, as we can see from the figure and table in the next page.



I	N
8	58
16	111
32	215
64	420
128	790
256	1568

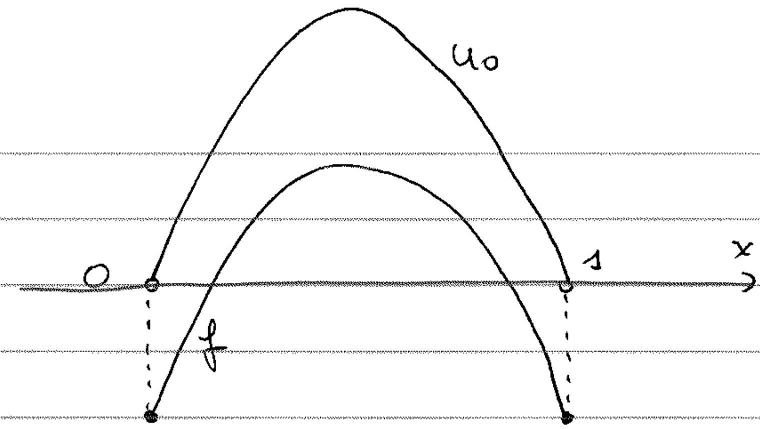
the values in this table are plotted in the thick line and the "■" symbol. For high values of "n", we see that the optimal choice of "ω" results in a much more efficient method that the projected SOR with "fixed" ω.

We have seen in this example that the projected SOR converges for $\omega \in (0,2)$. Moreover, we have also seen that taking ω as the optimum value for the corresponding problem without obstacle reduces the number of iterations from order I^2 to order I. These results are not simple to prove. Consult the book by Elliott and Ockendon entitled "Weak and variational methods for moving boundary problems" for more information and general results about this topic; see, in particular, Chapter IV.

5 <u>Accuracy of the method (3.1) for the obstacle problem.</u>

Let us consider how accurate the method (3.1) can be. Note that if the obstacle is such that $f<0$ in $(0,1)$, then the solution of the problem as well as the approximation given by (3.1) are both equal to zero. As a consequence, the error is zero when "there is no obstacle". However, when the obstacle is "present", the maximum error is of order Δx.

We are not going to present a general proof of this fact. Instead, we are going to prove it for the particular case we have considered in our numerical experiments. So, let us consider an obstacle of the form depicted in the figure below, and consider the we are applying the method (3.4) with $\Delta\tau = 1$ and some $2\frac{\Delta\tau}{\Delta x^2}\gamma \in (0,1]$; see next figure.

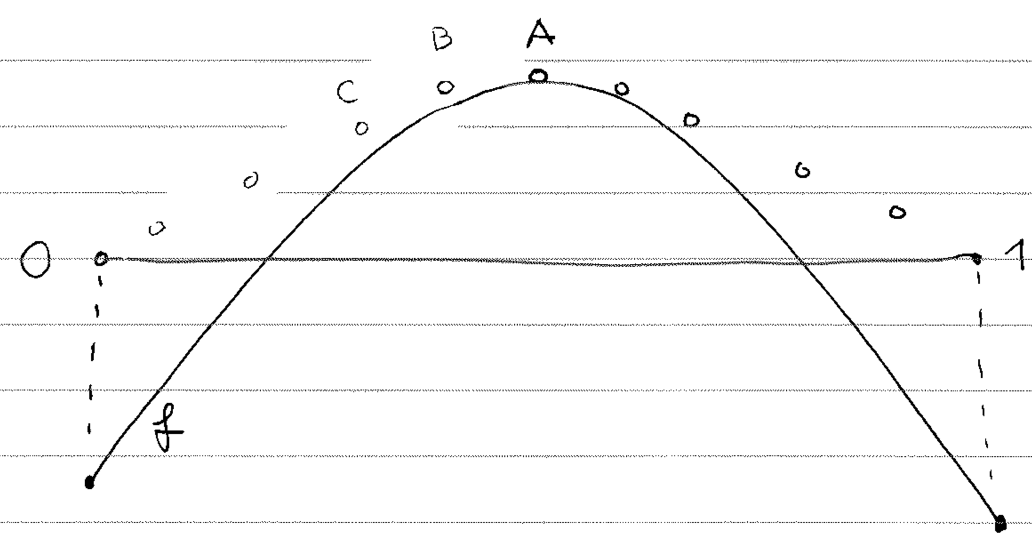


We have shown that the iterates $\{u_i^n\}_{i=0}^{I}$ given by algorithm (3.4) converge to a solution of problem (3.1) as n goes to infinity. We also have shown that the convergence takes place in such a way that

$$u_i^n \geq f_i \qquad \text{for } i=0,\dots,I,$$
$$-(U_{i+1}^n - 2U_i^n + U_{i-1}^n) \geq 0 \qquad \text{for } i=1,\dots,I-1,$$

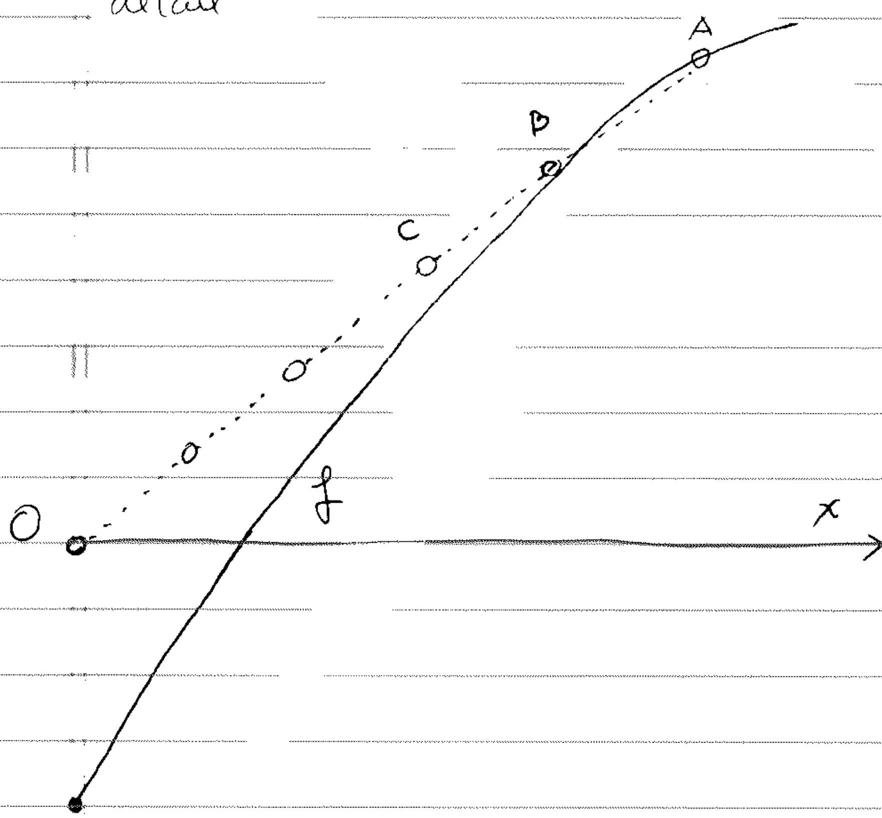
for all $n \geq 0$.

Now, after some iterations, we encounter the following situation:

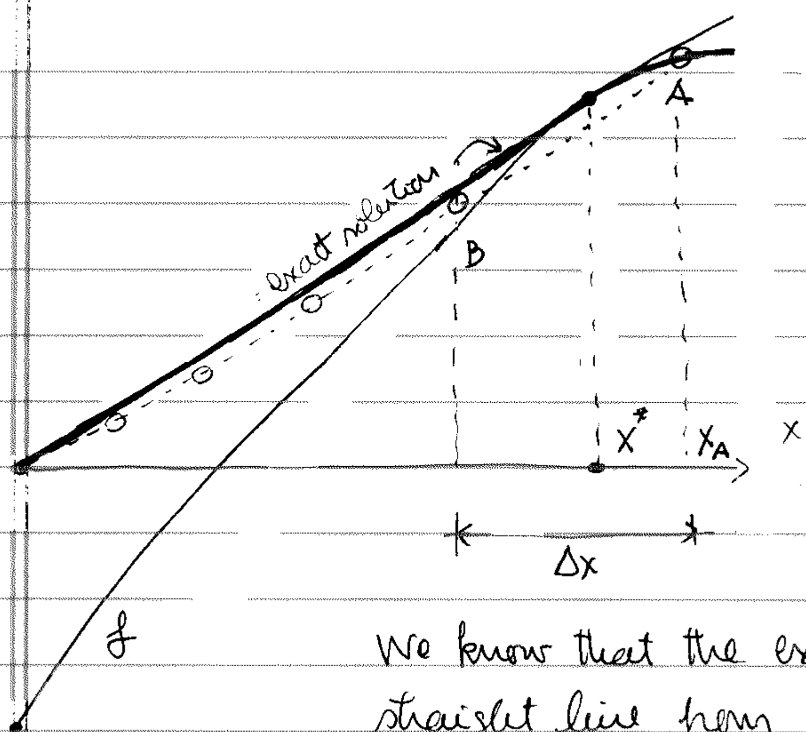


We see that since the point A is extremely close to the obstacle, it is not going to move particularly fast. However, the point B will continue to move since it is "far away" from the obstacle and because $B > \frac{1}{2}(A+C)$.

The question now is how far down is the point B going to move? We can see that it will stop when it hits the obstacle or when $B = \frac{1}{2}(C+A)$, whatever happens first. Let us assume that the first case does not take place. Then we have the following detail



Now, let us add the exact solution to this picture:



We know that the exact solution on $(0, x^*)$ is straight line from $(0,0)$ that is tangent to the obstacle at the point $(x^*, f(x^*))$ and that it coincides with obstacle close to and to the right of x^*; see thick line.

We can then see that, although the error at the point A is clearly equal to zero, this is not the case for the error at B, or for the error at all the points to the left of A for that matter. Since, at the points x_i to the left of x_A, we have

$$\begin{aligned} e_i &= u(x_i) - u_i \\ &= f'(x^*) \cdot x_i - \frac{f(x_A)}{x_A} x_i \\ &= \left[f'(x^*) - \frac{f(x_A)}{x_A} \right] x_i \end{aligned}$$

and since

$$f(x_A) = f(x^*) + (x_A - x^*) f'(x^*) + \tfrac{1}{2}(x_A - x^*)^2 f''(\underline{x}),$$

where $\underline{x}$ is some point in (x^*, x_A), we get that

$$e_i = \left[f'(x^*) - \frac{f(x^*) + (x_A - x^*) f'(x^*)}{x_A} \right] x_i + \tfrac{1}{2}(x_A - x^*)^2 f''(\underline{x}) \frac{x_i}{x_A}$$

$$= \left[\frac{x^* f'(x) - f(x^*)}{x_A} \right] x_i + \tfrac{1}{2}(x_A - x^*)^2 f''(\underline{x}) \frac{x_i}{x_A}.$$

But since at $x = x^*$ the exact solution is tangent to the obstacle, we have that

$$\frac{f(x^*)}{x^*} = f'(x^*).$$

This implies that

$$e_i = \tfrac{1}{2}(x_A - x^*)^2 f''(\underline{x}) \frac{x_i}{x_A}$$

Hence

$$|e_i| \le C \Delta x^2$$

at any point to the left of x_A. Note that if it happens that by chance $x^* = x_A$, the error is identically equal to zero. We just showed that

$$\|e\| = \max_{0 \le i \le I} |e_i| \le C \Delta x^2.$$

6. <u>A finite difference scheme for American options.</u>

Let us now use our knowledge of the numerical approximation of the obstacle problem to devise a numerical method for American option pricing. Let us concentrate on the discretization of equations (1.1); we will leave the discretization of the terminal and boundary conditions to the reader.

Let us discretize the Black-Scholes operator by using a Crank-Nicolson scheme. Thus, the discretized version of

$$\frac{\partial}{\partial t} f + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2}{\partial S^2} f + rS\frac{\partial f}{\partial S} - rf$$

is

$$\frac{1}{\Delta t}\left(f_i^n - f_i^{n-1}\right) + \frac{1}{2}\Big[\frac{1}{2}\sigma^2 S_i^2 \left(f_{i+1}^n - 2f_i^n + f_{i-1}^n\right)\frac{1}{\Delta S^2}$$
$$+ rS_i \frac{1}{2\Delta S}\left(f_{i+1}^n - f_{i-1}^n\right) - rf_i^n\Big]$$
$$+\frac{1}{2}\Big[\frac{1}{2}\sigma^2 S_i^2\left(f_{i+1}^{n-1} - 2f_i^{n-1} + f_{i-1}^{n-1}\right)\frac{1}{\Delta S^2}$$
$$+ rS_i\frac{1}{2\Delta S}\left(f_i^{n-1} - f_i^{n-1}\right) - r f_i^{n-1}\Big].$$

Let us rewrite this expression in a more compact form setting

$$\lambda = \frac{1}{2}\sigma^2\frac{\Delta t}{\Delta S^2}, \quad \gamma' = \frac{\Delta t}{2\Delta S},$$

and

$$\Lambda_i^m = \lambda S_i^2\left(f_{i+1}^m - 2f_i^m + f_{i-1}^m\right) + \gamma' r S_i\left(f_{i+1}^m - f_{i-1}^m\right) - r\Delta t f_i^m,$$

we can rewrite the discretized version of the Black-Scholes operator as

$$\frac{1}{\Delta t}\left(f_i^{n} + \frac{1}{2}\Lambda_i^{n} - f_i^{n-1} + \frac{1}{2}\Lambda_i^{n-1}\right).$$

So, we discretize the equations (1.1a), (1.1b) and (1.1c) as follows:

$$\text{(6.1a)} \qquad f_i^{n-1} - \frac{1}{2}\Lambda_i^{n-1} \geq f_i^{n} + \frac{1}{2}\Lambda_i^{n} \qquad \text{for } i = 1, \dots, I-1,$$

$$\text{(6.1b)}_p \qquad f_i \geq \max\{K - S_i, 0\} \qquad \text{for } i = 1, \dots, I-1,$$

$$\text{(6.1b)}_c \qquad f_i \geq \max\{S_i - K, 0\} \qquad \text{for } i = 1, \dots, I-1,$$

and

$$\text{(6.1c)}_p \qquad \left(\left(f_i^{n-1} - \frac{1}{2}\Lambda_i^{n-1}\right) - \left(f_i^{n} + \frac{1}{2}\Lambda_i^{n}\right)\right)\left(f_i - \max\{K - S_i, 0\}\right) = 0 \quad i = 1, \dots, I-1,$$

$$\text{(6.1c)}_c \qquad \left(\left(f_i^{n-1} - \frac{1}{2}\Lambda_i^{n-1}\right) - \left(f_i^{n} + \frac{1}{2}\Lambda_i^{n}\right)\right)\left(f_i - \max\{S_i - K, 0\}\right) = 0 \quad i = 1, \dots, I-1.$$

To complete these equations, we must add the terminal condition

$$\text{(6.1d)} \qquad f_i^{N} = f_T(S_i) \qquad i = 0, \dots, I,$$

and the boundary conditions

$$\text{(6.1e)} \qquad f_0^{n} = f_0(n\Delta t) \qquad \text{for } n = 0, \dots, N,$$

$$\text{(6.1f)} \qquad f_I^{n} = f_{S_M}(n\Delta t) \qquad \text{for } n = 0, \dots, N.$$

We can now easily see that at each time step we have to compute the solution extremely similar to our finite difference scheme (3.1) for the obstacle problem (2.1). Indeed, having computed $\{f_i^n\}_{i=0}^{I}$, $\{f_i^{n-1}\}_{i=0}^{I}$ is the solution of

(6.2a) $$f_i^{n-1} - \tfrac{1}{2}\Lambda_i^{n-1} - b_i \geq 0 \qquad i=1,\dots,I-1,$$

(6.2b) $$f_i^{n-1} - c_i \geq 0 \qquad i=1,\dots,I-1,$$

(6.2c) $$\left(f_i^{n-1} - \tfrac{1}{2}\Lambda_i^{n-1} - b_i\right)\left(f_i^{n-1} - c_i\right) = 0 \qquad i=1,\dots,I-1,$$

(6.2d) $$f_0^{n-1} = f_0((n-1)\Delta t) \text{ and } f_I^{n-1} = f_{S_M}((n-1)\Delta t),$$

where

(6.2e) $$b_i = f_i^n + \tfrac{1}{2}\Lambda_i^n, \qquad i=1,\dots,I-1,$$

(6.3f)$_p$ $$c_i = \max\{K - S_i, 0\} \qquad i=1,\dots,I-1,$$

(6.3f)$_c$ $$c_i = \max\{S_i - K, 0\} \qquad i=1,\dots,I-1.$$

A direct application of our algorithm of section 3 gives the following algorithm for solving the above problem:

(6.4a) $$u_i^{k+1} = u_i^k + \Delta\tau \max\left\{c_i - u_i^k,\ \gamma\left(b_i - u_i^k + \tfrac{1}{2}\Lambda_i^k\right)\right\} \quad i=1,\dots,I-1,$$

where Λ_i^k is obtained from Λ_i^m by replacing f_j^m by u_j^k for $j=i+1, i$ and $i-1$.

(6.4b) $$u_0^{k+1} = f_0((n-1)\Delta t) \text{ and } u_I^{k+1} = f_T((n-1)\Delta t),$$

(6.4c) $$u_i^0 = f_i^{n-1} \quad \text{for } i=1,\dots,I-1$$

(6.4d) $$u_0^0 = f_0((n-1)\Delta t) \text{ and } u_I^0 = f_T((n-1)\Delta t).$$

Now we can see that the projected SOR method is

(6.5a) $$u_i^{k+1} = \max\left\{ c_i^k ,\ u_i^k + \frac{\omega}{\rho_i}\left(b_i - u_i^k + \tfrac{1}{2}\Lambda_i^k\right)\right\} \quad i=1,\dots,I-1,$$

where

(6.5b) $$\Lambda_i^k = \lambda S_i^2\left(u_{i+1}^{k+1} - 2u_i^k + u_{i-1}^k\right) + \gamma r S_i\left(u_{i+1}^{k+1} - u_{i-1}^k\right) - r\Delta t\, u_i^k,$$

and

(6.5c) $$\rho = 1 + \lambda S_i^2 - \frac{r\Delta t}{2},$$

with the boundary condition (6.4b) and the initial condition (6.4c) and (6.4d).