# POPDIP:
# a POsitive-variables Primal-Dual Interior Point method

Ed Bueler

August 24, 2022

**Abstract**

The algorithm documented here is a version of a Newton-type primal-dual interior point algorithm in [5], namely Algorithm 16.1 in section 16.7. The version here minimizes a smooth nonlinear function subject to the simple constraints that all the variables are nonnegative. That is, it is a specialized algorithm for nonnegativity constraints; it is not suitable for general inequality constraints.

These short notes are not research as this algorithm is simply a special case of a well-known algorithm. Furthermore "POPDIP" is just a name I made up; it is not in common use! However, this class of algorithms is new to me so I am documenting it fully.

## Introduction

Consider a nonlinear optimization problem with nonnegativity (informally: positivity) constraints on the variables:

$$\begin{array}{ll}\text{minimize} & f(x) \\ \text{subject to} & x \ge 0\end{array} \tag{1}$$

As usual, "$x \ge 0$" means that each entry of $x \in \mathbb{R}^n$ is nonnegative. The feasible set for (1) is the convex and closed set $S = \{x \in \mathbb{R}^n \,:\, x \ge 0\}$ with interior $S^\circ = \{x \in \mathbb{R}^n \,:\, x > 0\}$.

Here $f : \mathbb{R}^n \to \mathbb{R}^n$ is a continuous and smooth function. In fact this algorithm only assumes $f$ is defined and smooth on $S^\circ$, but in practice, so as to have a chance of good performance, $f$ should be well-behaved near the boundary of $S$.

One can start the derivation by considering a logarithmic barrier function. Let $\mu > 0$. If $x \in S^\circ$ then the following function is well-defined and finite:

$$\beta_\mu(x) = f(x) - \mu \sum_{i=1}^n \ln x_i \tag{2}$$

Let $\{e_1, \dots, e_n\}$ be the standard basis of $\mathbb{R}^n$. The first-order necessary condition for the unconstrained problem of minimizing $\beta_\mu$, namely $\nabla \beta_\mu(x) = 0$ for $x \in S^\circ$, is

$$\nabla f(x) - \mu \sum_{i=1}^n \frac{1}{x_i} e_i = 0 \tag{3}$$

Conditions (3) can be reformulated by defining additional variables

$$\lambda_i = \frac{\mu}{x_i}$$

so that $\lambda \in \mathbb{R}^n$. Note that $\lambda > 0$ if and only if $x > 0$ because $\lambda_i x_i = \mu > 0$. Then (3), plus feasibility for $x$, is precisely equivalent to the following nonlinear system of equations and inequalities:

$$\begin{aligned} x &\geq 0 \\ \lambda &\geq 0 \\ \nabla f(x) - \lambda &= 0 \\ \lambda_i x_i &= \mu, \qquad i = 1, \ldots, n \end{aligned} \tag{4}$$

The feasible set for $x$ and for $\lambda$ is the same, namely $S \subset \mathbb{R}^n$. Because of the last condition in (4), both $x$ and $\lambda$ are positive and thus in the interior $S^\circ$. By contrast, for the general primal-dual interior point Algorithm 16.1 [5, section 16.7], the feasible set for the primal variable $x$ is different from the dual feasible set for $\lambda$. For example, generally the dimension is different.

The third condition in (4) can be written using a Lagrangian function for (1), namely

$$\mathcal{L}(x, \lambda) = f(x) - \sum_{i=1}^{n} \lambda_i x_i, \tag{5}$$

in which case the third condition states that $\nabla_x \mathcal{L}(x, \lambda) = 0$. In fact, the KKT conditions [5, sections 14.4, 14.5] are nearly the same as (4) but with the last equation replaced by a condition of complementary slackness,

$$\begin{aligned} x &\geq 0 \\ \lambda &\geq 0 \\ \nabla f(x) - \lambda &= 0 \\ \lambda_i x_i &= 0, \qquad i = 1, \ldots, n \end{aligned} \tag{6}$$

System (4) modifies these KKT conditions by changing complementary slackness to a connection between the primal and dual variables. Their product is set to a positive constant. Thus system (4) describes a solution which is different from the KKT conditions for (1).

The stationary-point conditions for Lagrangian (5) which would apply in the absence of positivity constraints, namely $\nabla_x \mathcal{L}(x, \lambda) = 0$ and $\nabla_\lambda \mathcal{L}(x, \lambda) = 0$, do not generally apply, of course. (Making the inequality constraints into equalities would make the problem trivial!) However, $x_i = 0$ holds by definition when the $i$th constraint is active. The algorithm here will not keep track of the active set but rather allow determination of the active set upon convergence, by use of a near-zero tolerance.

One of the KKT conditions (6) is $\nabla f(x) - \lambda = 0$ so $\lambda$ can be easily eliminated if desired. In fact, writing the conditions without the dual variables gives a *nonlinear complementary problem* (NCP) [3],

$$x \geq 0, \qquad F(x) \geq 0, \qquad x_i F(x)_i = 0$$

where $F(x) = \nabla f(x)$ is a vector-valued function. Thus we can regard our method as solving the barrier-modified NCP

$$x \geq 0, \qquad F(x) \geq 0, \qquad x_i F(x)_i = \mu.$$

However, a primal-dual method for this NCP would re-introduce the dual variables so there is no benefit here of an NCP formulation other than to note the connection for future reference. (An NCP formulation is appropriate if the problem were a variational inequality [1, 2] instead of actual optimization.)

## Algorithm design

Algorithm 16.1 [5] applies to (1). The POPDIP algorithm proposed below is the simplification for $g_i(x) = x_i$. All such primal-dual interior point algorithms [6] compute approximate solutions to systems like (4) for a sequence $\{\mu_k > 0\}$ going to zero. The limit where $\mu = 0$, never achieved because the algorithm is terminated after a finite number of steps, solves the KKT conditions (6).

Each step of the algorithm is a Newton step for the equalities in (4), namely

$$\begin{aligned} \nabla f(x) - \lambda &= 0 \\ \lambda_i x_i &= \mu_k, \qquad i = 1, \dots, n. \end{aligned} \tag{7}$$

The step updates both $x$ and $\lambda$ using the linearization of these equations.

Because of the second equation, (7) is always a nonlinear system. However, if $f$ is quadratic then the first equation is linear.

To describe the Newton step let $x = x_k + \Delta x$ and $\lambda = \lambda_k + \Delta\lambda$. We assume that the current iterate $(x_k, \lambda_k)$ does not solve (7). The unknowns in the Newton step form the search direction $p = (\Delta x, \Delta\lambda)$. Substituting into (7) and expanding to first order gives

$$\begin{aligned} \nabla f(x_k) + \nabla^2 f(x_k)\Delta x - \lambda_k - \Delta\lambda &= 0 \\ (\lambda_k)_i (x_k)_i + (x_k)_i (\Delta\lambda)_i + (\lambda_k)_i (\Delta x)_i &= \mu_k, \qquad i = 1, \dots, n \end{aligned} \tag{8}$$

Rearranging as a linear block system for the search direction, and suppressing the current-iterate subscript, we get step equations

$$\begin{bmatrix} \nabla^2 f(x) & -I \\ \Lambda & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} -\nabla f(x) + \lambda \\ -\Lambda x + \mu_k e \end{bmatrix} \tag{9}$$

Here $I$ is the $n \times n$ identity matrix and the other notation is as follows:

$$\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix}, \qquad X = \begin{bmatrix} x_1 & & \\ & \ddots & \\ & & x_n \end{bmatrix}, \qquad e = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}.$$

Given a solution of (9) the update formulas are

$$\begin{aligned} x_{k+1} &= x_k + \alpha\Delta x \\ \lambda_{k+1} &= \lambda_k + \alpha\Delta\lambda \end{aligned}$$

A single maximum step size $\alpha$ for both the primal and dual variables is determined by a condition of (strict) positivity for both $x_{k+1}$ and $\lambda_{k+1}$ [6]. (A ratio test suffices to keep $x$ feasible, in common with how the slack variables are limited in a general primal-dual interior point method [5, subsection 16.7.2]. Back-tracking is not needed to maintain feasibility of the primal variables because of the linearity of the constraint functions in (1), namely $g_i(x) = x_i$.) Because this is a Newton search one uses $\alpha = 1$ for the largest allowed step. Note we use $p = (\Delta x, \Delta\lambda) \in \mathbb{R}^{2n}$ as a search direction.

The optimality test is the same as in Algorithm 16.1 [5]. It uses the merit function

$$\nu(x, \lambda) = \max\{\|\nabla f(x) - \lambda\|, \|\Lambda x\|\} \tag{10}$$

where $\|\cdot\|$ denotes the usual $L^2$ norm on $\mathbb{R}^n$. Note that once $\mu_k \to 0$ we have $\nu(x_*, \lambda_*) = 0$ for the exact optimum, but when $\mu_k \neq 0$ then the exact solution of (7) does not make $\nu(x, \lambda)$ zero. In fact $\nu(x_*, \lambda_*) = \sqrt{n}\,\mu_k$ for the exact solution of (7); if the merit function value $\nu(x_k, \lambda_k)$ is close to $\sqrt{n}\,\mu_k$ then we should decrease $\mu_k$ more rapidly.

Subsection 16.7.2 of [5] describes how Algorithm 16.1 should be modified so that, by Theorem 16.17, the algorithm will exhibit the local quadratic convergence.[1] They describe how the general inequality constraints should be augmented by slack (excess) variables which have simple nonnegativity constraints, but in our case the constraints are already of that type.[2] However, Theorem 16.17 also uses a specific barrier parameter update for $\mu_k$ and a specific parameter choice for $\kappa$ in the ratio test. In our case these are equations

$$\begin{aligned} \mu_k &= \min\{\theta\nu(x_k, \lambda_k), \nu(x_k, \lambda_k)^2\}, \\ \kappa &= \max\{\bar\kappa, 1 - \nu(x_k, \lambda_k)\}, \end{aligned}$$

for parameters $0 < \theta < 1$ and $0 < \bar\kappa < 1$. We choose these parameters based on the minimal hints in Example 16.16 on pages 643–644.

The method by which the initial dual variables are determined (below) is a somewhat ad hoc construction for which I have no reference. This issue of initialization is discussed in [4].

## Algorithm

We can now present a pseudocode for our algorithm.

ALGORITHM POPDIP.

```
    inputs  primal initial values $x_0$ such that $x_0 > 0$
            smooth function $f$ returning $f(x)$, $\nabla f(x)$, and $\nabla^2 f(x)$
parameters  $0 <$ tol [default tol $= 10^{-4}$]
            maxiters $> 0$ [default maxiters $= 200$]
            $0 < \theta < 1$ [default $\theta = 0.1$]
```

[1] One would want this property for a Newton-type algorithm, but quadratic convergence is much harder to achieve in a constrained problem. Thus I am interested in this algorithm!

[2] The general replacement "$g_i(x) \geq 0$" by "$g_i(x) - s_i = 0$, $s_i \geq 0$" would be, in our case, the replacement "$x_i \geq 0$" by "$x_i - s_i = 0$, $s_i \geq 0$." Clearly this is a meaningless renaming of the existing variables.

```
        $0 < \bar\kappa < 1$ [default $\bar\kappa = 0.9$]
output  an estimate $(x_k, \lambda_k)$ of the solution
    • determine initial dual variables:
        (i) $y_0 = \nabla f(x_0)$
        (ii) if $y_0 \le 0$ then $\mu_0 = 1$; otherwise $\mu_0$ is average of those products $(x_0)_i(y_0)i$
             where $(y_0)_i > 0$
        (iii) $(\lambda_0)_i = \mu_0/(x_0)_i$ for $i = 1, \dots, n$
    • for $k = 0, 1, 2, \dots, \texttt{maxiters} - 1$
        (i) optimality test: if $\nu(x_k, \lambda_k) < \texttt{tol}$ then stop
        (ii) barrier parameter: $\mu_k = \min\{\theta\nu(x_k, \lambda_k), \nu(x_k, \lambda_k)^2\}$
        (iii) compute Newton step by solving this system for $(\Delta x, \Delta\lambda)$:
              $\begin{bmatrix} \nabla^2 f(x_k) & -I \\ \Lambda_k & X_k \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} -\nabla f(x_k) + \lambda_k \\ -\Lambda_k x_k + \mu_k e \end{bmatrix}$
        (iv) ratio test for step sizes to keep $x_{k+1}, \lambda_{k+1}$ positive:
              $\kappa = \max\{\bar\kappa, 1 - \nu(x_k, \lambda_k)\}$
              $\alpha = \min_{1\le i\le n} \left\{1,\ -\kappa\frac{(x_k)_i}{(\Delta x)_i}\ :\ (\Delta x)_i < 0,\ -\kappa\frac{(\lambda_k)_i}{(\Delta\lambda)_i}\ :\ (\Delta\lambda)_i < 0\right\}$
        (v) the update:
              $x_{k+1} = x_k + \alpha\Delta x$
              $\lambda_{k+1} = \lambda_k + \alpha\Delta\lambda$
```

This algorithm is implemented by a MATLAB code with signature

```
function [xk,lamk,xklist,lamklist] = popdip(x0,f,tol,maxiters,theta,kappabar)
```

Only inputs `x0,f` are required. The parameters have the default values listed above. If the outputs `xklist` and `lamklist` are not requested then they are not saved.

Download the code at

bueler.github.io/M661F18/matlab/popdip.m

## Example 1

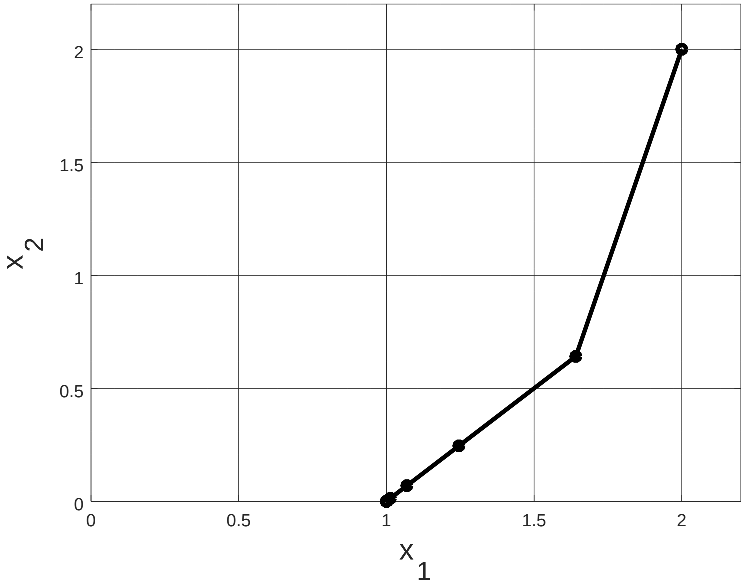
We first try an easy 2D test problem in `testpopdip.m`:

$$\begin{array}{ll} \text{minimize} & f(x) = \frac{1}{2}(x_1 - 1)^2 + \frac{1}{2}(x_2 + 1)^2 \\ \text{subject to} & x \ge 0 \end{array} \tag{11}$$

For this objective function $f$ the unconstrained minimum is the infeasible point $\hat{x} = (1, -1)^\top$. A sketch shows that the exact solution of the constrained problem is $x_* = (1, 0)^\top$. Running the code with the given initial iterate $x_0 = (2, 2)^\top$ and $\texttt{tol} = 10^{-14}$, and otherwise using the default parameters in POPDIP, gives

```
>> testpopdip([2 2]’,1.0e-14)
  1:    2.000000000000000    2.000000000000000
  2:    1.641421356237309    0.641421356237310
  3:    1.245446520462486    0.245446520462486
  4:    1.069404818969903    0.069404818969903
  5:    1.013447008853577    0.013447008853577
  6:    1.000537794151783    0.000537794151783
  7:    1.000000867357066    0.000000867357066
  8:    1.000000000002257    0.000000000002257
  9:    1.000000000000000    0.000000000000000
```

The printed columns are $(x_k)_1$ and $(x_k)_2$. Note the apparent quadratic, or at least strongly superlinear, convergence. The iterates $x_k$ are graphed in the following figure.



### Example 2

FIXME obstacle problem in 1D

$$f[u] = \int_0^1 \frac{1}{2}|u'(x)|^2 - q(x)u(x)\,dx \tag{12}$$

$\Delta x = 1/(n+1)$

$$f(u) = \Delta x \sum_{i=0}^{n} \left( \frac{1}{2} \left( \frac{u_{i+1} - u_i}{\Delta x} \right)^2 - q(x_{i+1/2}) \frac{u_i + u_{i+1}}{2} \right) \tag{13}$$

where $u_0 = 0$ and $u_{n+1} = 0$ when they appear

gradient components for $i = 1, \dots, n$

$$\nabla f(u)_i = \frac{1}{\Delta x} \left\{ \begin{matrix} 2u_1 - u_2 & (i = 1) \\ -u_{i-1} + 2u_i - u_{i+1} & (1 < i < n) \\ -u_{n-1} + 2u_n & (i = n) \end{matrix} \right\} - \frac{\Delta x}{2}(q(x_{i-1/2}) + q(x_{i+1/2})) \tag{14}$$

Hessian matrix

$$\nabla^2 f(u) = \frac{1}{\Delta x} \begin{bmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & & \ddots & \\ & & -1 & 2 \end{bmatrix} \tag{15}$$

### Possible improvements

We may consider possible improvements of our algorithm. First, in Algorithm 16.1 the computation of the Newton search direction is followed by separate line searches in $x$ and in $\lambda$. These line searches only maintain nonnegativity and they do not seek sufficient decrease of $f(x)$; they only use ratio tests. Secondly, equation (9) can be symmetrized by multiplying the second half of the equations by $-\Lambda^{-1}$:

$$\begin{bmatrix} \nabla^2 f(x) & -I \\ -I & -\Lambda^{-1}X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} -\nabla f(x) + \lambda \\ x - \mu_k \Lambda^{-1} e \end{bmatrix} \tag{16}$$

FIXME further simplify into system of $n$ equations for $\lambda$ only

These facts suggests two possible changes:

1. Back-tracking line search is appropriate as a globalization even for unconstrained optimization. Thus there must be cases where it is appropriate for problem (1) as well. Once the ratio tests are applied, further back-tracking could be used based on sufficient decrease. Compare the modified back-tracking line searches in [1].

2. One can replace linear system (9) with symmetrized system (16).

Determining if these are actual improvements would require testing which we have not done.

## References

[1] S. Benson and T. Munson, *Flexible complementarity solvers for large-scale applications*, Optimization Methods and Software, 21 (2006), pp. 155–168.

[2] E. Bueler, *Stable finite volume element schemes for the shallow ice approximation*, J. Glaciol., 62 (2016), pp. 230–242.

[3] F. Facchinei and J.-S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems*, Springer, 2007.

[4] M. Gertz, J. Nocedal, and A. Sartenar, *A starting point strategy for nonlinear interior methods*, Applied Mathematics Letters, 17 (2004), pp. 945–952.

[5] I. Griva, S. G. Nash, and A. Sofer, *Linear and Nonlinear Optimization*, SIAM Press, 2nd ed., 2009.

[6] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, New York, 2nd ed., 2006.