Monotone Operators: Primal-Dual Splitting

Mert Pilanci

EE364b, Stanford University

Outline

- Primal-Dual Hybrid Gradient (PDHG)
- Condat-Vu Splitting

Goal

We want to derive primal-dual algorithms by starting from the optimality conditions and rewriting them as fixed point equations.
The guiding idea is simple:

$$0 \in A(z)$$

can be rewritten as

$$z \in z + \lambda A(z).$$

Therefore

$$z = (I + \lambda A)^{-1}(z).$$

When

$$A = \partial f,$$

we have

$$(I + \lambda \partial f)^{-1} = \mathbf{prox}_{\lambda f}.$$

So proximal maps appear by algebraically rewriting optimality conditions, not by changing the original problem.

The basic proximal identity

Let q be a closed convex function. Suppose

$$0 \in \partial q(u) + a.$$

Then

$$-a \in \partial q(u).$$

Multiply by $\lambda > 0$:

$$-\lambda a \in \lambda \partial q(u).$$

Add u to both sides:

$$u - \lambda a \in u + \lambda \partial q(u).$$

Thus

$$u - \lambda a \in (I + \lambda \partial q)(u).$$

Applying the inverse operator gives

$$u = (I + \lambda \partial q)^{-1}(u - \lambda a).$$

Therefore

$$u = \mathrm{prox}_{\lambda q}(u - \lambda a).$$

Rule to remember

The inclusion

$$0 \in \partial q(u) + a$$

is equivalent to the fixed point equation

$$u = \mathrm{prox}_{\lambda q}(u - \lambda a).$$

This is the only algebraic rule we need.

$$\boxed{0 \in \partial q(u) + a \quad \Longleftrightarrow \quad u = \mathrm{prox}_{\lambda q}(u - \lambda a)}$$

The vector a can be a gradient, a linear coupling term, or a dual variable.

Warm-up: proximal gradient

Consider

$$\min_x f(x) + g(x),$$

where f is differentiable and g is proximable.
The optimality condition is

$$0 \in \nabla f(x^\star) + \partial g(x^\star).$$

Rewrite it as

$$0 \in \partial g(x^\star) + \nabla f(x^\star).$$

Using the proximal identity with

$$q = g, \qquad a = \nabla f(x^\star),$$

we get

$$x^\star = \mathrm{prox}_{\lambda g}\left(x^\star - \lambda \nabla f(x^\star)\right).$$

This suggests the iteration

$$x_{k+1} = \mathrm{prox}_{\lambda g}\left(x_k - \lambda \nabla f(x_k)\right).$$

Why introduce a primal-dual form?

Proximal gradient is natural for

$$\min_x f(x) + g(x).$$

But many problems have the form

$$\min_x f(x) + g(Kx).$$

Examples:

$$\min_x f(x) + \lambda \|Kx\|_1,$$

$$\min_x f(x) + \lambda \, \mathrm{TV}(x),$$

$$\min_x f(x) + I_{\mathcal{C}}(Kx).$$

The difficulty is that even if prox_g is easy,

$$\mathrm{prox}_{g \circ K}$$

may be hard.

Primal-dual methods avoid computing $\mathrm{prox}_{g \circ K}$.

Fenchel representation

For a closed convex function g,

$$g(s) = \sup_y \left\{ y^T s - g^*(y) \right\}.$$

Therefore

$$g(Kx) = \sup_y \left\{ y^T Kx - g^*(y) \right\}.$$

So

$$\min_x f(x) + g(Kx)$$

can be written as the saddle problem

$$\min_x \max_y \left\{ f(x) + y^T Kx - g^*(y) \right\}.$$

Equivalently,

$$\min_x \max_y \left\{ f(x) + x^T K^T y - g^*(y) \right\}.$$

Saddle optimality conditions

Define

$$L(x, y) = f(x) + y^T K x - g^*(y).$$

At a saddle point $(x^\star, y^\star)$, we need

$$0 \in \partial_x L(x^\star, y^\star),$$

and

$$0 \in -\partial_y L(x^\star, y^\star).$$

For the primal variable,

$$\partial_x L(x, y) = \partial f(x) + K^T y.$$

For the dual variable,

$$-\partial_y L(x,y) = \partial g^*(y) - Kx.$$

Therefore the saddle optimality conditions are

$$0 \in \partial f(x^\star) + K^T y^\star,$$

$$0 \in \partial g^*(y^\star) - Kx^\star.$$

First fixed point equation

Start with the primal optimality condition

$$0 \in \partial f(x^\star) + K^T y^\star.$$

This has the form

$$0 \in \partial q(u) + a$$

with

$$q = f, \qquad u = x^\star, \qquad a = K^T y^\star.$$

Therefore

$$x^\star = \mathrm{prox}_{\tau f}\left(x^\star - \tau K^T y^\star\right).$$

This is the primal fixed point equation.

Second fixed point equation

Start with the dual optimality condition

$$0 \in \partial g^*(y^\star) - Kx^\star.$$

This has the form

$$0 \in \partial q(u) + a$$

with

$$q = g^*, \qquad u = y^\star, \qquad a = -Kx^\star.$$

Therefore

$$y^\star = \mathrm{prox}_{\sigma g^*}\left(y^\star - \sigma(-Kx^\star)\right).$$

Hence

$$y^\star = \mathrm{prox}_{\sigma g^*}\left(y^\star + \sigma K x^\star\right).$$

This is the dual fixed point equation.

The primal-dual fixed point

The saddle point satisfies

$$x^\star = \mathrm{prox}_{\tau f}\left(x^\star - \tau K^T y^\star\right),$$

$$y^\star = \mathrm{prox}_{\sigma g^*}\left(y^\star + \sigma K x^\star\right).$$

So the pair $(x^\star, y^\star)$ is a fixed point of

$$F(x,y) = \begin{bmatrix} \mathrm{prox}_{\tau f}(x - \tau K^T y) \\ \mathrm{prox}_{\sigma g^*}(y + \sigma K x) \end{bmatrix}.$$

A naive fixed point iteration would be

$$x_{k+1} = \mathrm{prox}_{\tau f}\left(x_k - \tau K^T y_k\right),$$

$$y_{k+1} = \mathrm{prox}_{\sigma g^*}\left(y_k + \sigma K x_k\right).$$

Why the naive iteration is not enough

The primal-dual coupling is

$$K^T y \qquad \text{and} \qquad -Kx.$$

This is a skew-symmetric coupling:

$$\begin{bmatrix} 0 & K^T \\ -K & 0 \end{bmatrix}.$$

Forward steps on skew-symmetric operators tend to rotate.
A simple analogy is

$$\dot{z} = Jz, \qquad J^T = -J.$$

The dynamics rotate instead of contract.
So we need a slightly more careful fixed point iteration. PDHG uses a correction through a fresh dual variable and an extrapolated primal point.

A more stable ordering

The fixed point equations are

$$x = \mathrm{prox}_{\tau f}\left(x - \tau K^T y\right),$$

$$y = \mathrm{prox}_{\sigma g^*}\left(y + \sigma K x\right).$$

Instead of updating both using old values, use the new dual variable in the primal step:

$$y_{k+1} = \mathrm{prox}_{\sigma g^*}\left(y_k + \sigma K x_k\right),$$

$$x_{k+1} = \mathrm{prox}_{\tau f}\left(x_k - \tau K^T y_{k+1}\right).$$

At a fixed point this still gives

$$x_{k+1} = x_k = x^\star, \qquad y_{k+1} = y_k = y^\star.$$

The extrapolation idea

PDHG usually replaces x_k in the dual step by an extrapolated point

$$\bar{x}_k = x_k + \theta(x_k - x_{k-1}).$$

Then

$$y_{k+1} = \mathrm{prox}_{\sigma g^*}\left(y_k + \sigma K \bar{x}_k\right),$$

$$x_{k+1} = \mathrm{prox}_{\tau f}\left(x_k - \tau K^T y_{k+1}\right).$$

The extrapolation is not changing the optimality condition.
At a fixed point,

$$x_k = x_{k-1} = x^\star,$$

so $\bar{x}_k = x^\star$.

PDHG algorithm

For

$$\min_x f(x) + g(Kx),$$

PDHG is

$$\bar{x}_k = x_k + \theta(x_k - x_{k-1}),$$

$$y_{k+1} = \mathrm{prox}_{\sigma g^*}\left(y_k + \sigma K \bar{x}_k\right),$$

$$x_{k+1} = \mathrm{prox}_{\tau f}\left(x_k - \tau K^T y_{k+1}\right).$$

Equivalently, one often writes

$$\bar{x}_{k+1} = x_{k+1} + \theta(x_{k+1} - x_k).$$

A common convergence condition is

$$\tau\sigma\|K\|_2^2 < 1,$$

with

$$0 \leq \theta \leq 1.$$

PDHG in one line

The optimality system is

$$\begin{cases} 0 \in \partial f(x) + K^T y, \\ 0 \in \partial g^*(y) - Kx. \end{cases}$$

Apply the proximal identity separately:

$$\begin{cases} x = \mathrm{prox}_{\tau f}(x - \tau K^T y), \\ y = \mathrm{prox}_{\sigma g^*}(y + \sigma K x). \end{cases}$$

Turn the fixed point into an iteration:

$$\begin{cases} y_{k+1} = \mathrm{prox}_{\sigma g^*}(y_k + \sigma K \bar{x}_k), \\ x_{k+1} = \mathrm{prox}_{\tau f}(x_k - \tau K^T y_{k+1}). \end{cases}$$

This is PDHG.

Special case: original composite problem

For

$$\min_x f(x) + g(x),$$

take

$$K = I.$$

Then

$$\min_x f(x) + g(x) = \min_x f(x) + g(Ix).$$

The saddle form is

$$\min_x \max_y \left\{ f(x) + y^T x - g^*(y) \right\}.$$

The optimality conditions are

$$0 \in \partial f(x^\star) + y^\star,$$

$$0 \in \partial g^*(y^\star) - x^\star.$$

The fixed point equations are

$$x^\star = \mathrm{prox}_{\tau f}(x^\star - \tau y^\star),$$

$$y^\star = \mathrm{prox}_{\sigma g^*}(y^\star + \sigma x^\star).$$

Now add a smooth term

Condat–Vu handles

$$\min_x f(x) + h(x) + g(Kx),$$

where

f is proximable,

h is smooth convex,

g is proximable through g^*.

The saddle form is

$$\min_x \max_y \left\{ f(x) + h(x) + y^T K x - g^*(y) \right\}.$$

The only new term is the smooth gradient

$$\nabla h(x).$$

Condat–Vu optimality system

The saddle function is

$$L(x, y) = f(x) + h(x) + y^T Kx - g^*(y).$$

The saddle optimality conditions are

$$0 \in \partial f(x^\star) + \nabla h(x^\star) + K^T y^\star,$$

$$0 \in \partial g^*(y^\star) - Kx^\star.$$

This is the same structure as PDHG, except that the primal inclusion now contains

$$\nabla h(x^\star).$$

Primal fixed point for Condat–Vu

Start with

$$0 \in \partial f(x^\star) + \nabla h(x^\star) + K^T y^\star.$$

This has the form

$$0 \in \partial q(u) + a$$

with

$$q = f, \qquad u = x^\star,$$

and

$$a = \nabla h(x^\star) + K^T y^\star.$$

Therefore $x^\star = \mathrm{prox}_{\tau f}\left(x^\star - \tau \nabla h(x^\star) - \tau K^T y^\star\right)$

Dual fixed point for Condat–Vu

The dual inclusion is unchanged:

$$0 \in \partial g^*(y^\star) - Kx^\star.$$

Using the proximal identity,

$$y^\star = \mathrm{prox}_{\sigma g^*}\left(y^\star + \sigma K x^\star\right).$$

Thus the fixed point equations are

$$x^\star = \mathrm{prox}_{\tau f}\left(x^\star - \tau \nabla h(x^\star) - \tau K^T y^\star\right),$$

$$y^\star = \mathrm{prox}_{\sigma g^*}\left(y^\star + \sigma K x^\star\right).$$

Naive primal-dual forward-backward

A direct fixed point iteration would be

$$x_{k+1} = \mathrm{prox}_{\tau f}\left(x_k - \tau \nabla h(x_k) - \tau K^T y_k\right),$$

$$y_{k+1} = \mathrm{prox}_{\sigma g^*}\left(y_k + \sigma K x_k\right).$$

This is the most literal iteration from the fixed point equations.
But again the coupling through K can create rotation.
Condat–Vu uses a reflected primal point in the dual update.

The reflection trick

After computing x_{k+1}, form

$$\tilde{x}_{k+1} = 2x_{k+1} - x_k.$$

This is a reflected or extrapolated point.
Then use it in the dual update:

$$y_{k+1} = \operatorname{prox}_{\sigma g^*}\left(y_k + \sigma K \tilde{x}_{k+1}\right).$$

That is,

$$y_{k+1} = \operatorname{prox}_{\sigma g^*}\left(y_k + \sigma K(2x_{k+1} - x_k)\right).$$

At a fixed point,

$$x_{k+1} = x_k = x^\star,$$

so

$$2x_{k+1} - x_k = x^\star.$$

Thus the reflection changes the path, not the solution.

Condat–Vu algorithm

For

$$\min_x f(x) + h(x) + g(Kx),$$

the Condat–Vu iteration is

$$x_{k+1} = \mathrm{prox}_{\tau f}\left(x_k - \tau \nabla h(x_k) - \tau K^T y_k\right),$$

$$y_{k+1} = \mathrm{prox}_{\sigma g^*}\left(y_k + \sigma K(2x_{k+1} - x_k)\right).$$

The fixed point equations are exactly the optimality conditions:

$$0 \in \partial f(x^\star) + \nabla h(x^\star) + K^T y^\star,$$

$$0 \in \partial g^*(y^\star) - Kx^\star.$$

Condat–Vu with relaxation

A relaxed version uses

$$\tilde{x}_{k+1} = x_{k+1} + \theta(x_{k+1} - x_k),$$

where typically

$$0 \le \theta \le 1.$$

Then

$$y_{k+1} = \mathrm{prox}_{\sigma g^*}\left(y_k + \sigma K \tilde{x}_{k+1}\right).$$

For

$$\theta = 1,$$

we get the standard reflected point

$$\tilde{x}_{k+1} = 2x_{k+1} - x_k.$$

For

$$\theta = 0,$$

there is no reflection:

$$\tilde{x}_{k+1} = x_{k+1}.$$

Why the reflection is like acceleration

The reflected point

$$2x_{k+1} - x_k$$

is a first-order prediction of where the primal variable is heading.
It is the same simple extrapolation idea as

$$x_{k+1} + (x_{k+1} - x_k).$$

So the dual variable sees not only the new primal point, but also the local direction of motion.

This helps compensate for the skew-symmetric coupling

$$K^T y \qquad \text{and} \qquad -Kx.$$

The reflection is therefore an extragradient-like correction.

PDHG versus Condat–Vu

PDHG solves

$$\min_x f(x) + g(Kx).$$

Its fixed point comes from

$$0 \in \partial f(x) + K^T y,$$

$$0 \in \partial g^*(y) - Kx.$$

Condat–Vu solves

$$\min_x f(x) + h(x) + g(Kx).$$

Its fixed point comes from

$$0 \in \partial f(x) + \nabla h(x) + K^T y,$$

$$0 \in \partial g^*(y) - Kx.$$

So Condat–Vu is PDHG plus a forward gradient step on the smooth term h.

Recovering proximal gradient from Condat–Vu

Take the original problem

$$\min_x h(x) + g(x),$$

where h is smooth and g is nonsmooth.
This is the Condat–Vu problem with

$$f = 0, \qquad K = I.$$

The optimality conditions are

$$0 = \nabla h(x^\star) + y^\star,$$

$$0 \in \partial g^*(y^\star) - x^\star.$$

The first condition says

$$y^\star = -\nabla h(x^\star).$$

The second condition says

$$x^\star \in \partial g^*(y^\star).$$

Equivalently,

$$y^\star \in \partial g(x^\star).$$

Thus

$$-\nabla h(x^\star) \in \partial g(x^\star),$$

which gives

$$0 \in \nabla h(x^\star) + \partial g(x^\star).$$

Using Moreau to connect to proximal gradient

Moreau's identity says

$$\text{prox}_{\sigma g^*}(y) = y - \sigma\, \text{prox}_{g/\sigma}(y/\sigma).$$

In the case

$$\min_x h(x) + g(x),$$

the dual update is

$$y_{k+1} = \text{prox}_{\sigma g^*}\left(y_k + \sigma(2x_{k+1} - x_k)\right).$$

Using Moreau's identity,

$$y_{k+1} = y_k + \sigma(2x_{k+1} - x_k) - \sigma\, \text{prox}_{g/\sigma}\left(\frac{1}{\sigma} y_k + 2x_{k+1} - x_k\right).$$

Thus the dual update can be written using the prox of g.

The main message

All of these methods come from the same algebra.
Start with an inclusion:

$$0 \in \partial q(u) + a.$$

Rewrite it as a fixed point:

$$u = \mathrm{prox}_{\lambda q}(u - \lambda a).$$

For PDHG, the inclusions are

$$0 \in \partial f(x) + K^T y,$$

$$0 \in \partial g^*(y) - Kx.$$

For Condat–Vu, the inclusions are

$$0 \in \partial f(x) + \nabla h(x) + K^T y,$$

$$0 \in \partial g^*(y) - Kx.$$

The extrapolation terms

$$\bar{x}_k = x_k + \theta(x_k - x_{k-1})$$

and

$$2x_{k+1} - x_k$$

change the iteration trajectory, but not the fixed points.

Compact summary

PDHG:

$$\min_x f(x) + g(Kx)$$

$$\begin{cases} 0 \in \partial f(x) + K^T y, \\ 0 \in \partial g^*(y) - Kx. \end{cases}$$

$$\begin{cases} y_{k+1} = \operatorname{prox}_{\sigma g^*}\left(y_k + \sigma K \bar{x}_k\right), \\ x_{k+1} = \operatorname{prox}_{\tau f}\left(x_k - \tau K^T y_{k+1}\right), \\ \bar{x}_{k+1} = x_{k+1} + \theta(x_{k+1} - x_k). \end{cases}$$

Condat–Vu:

$$\min_x f(x) + h(x) + g(Kx)$$

$$
\begin{cases} x_{k+1} = \operatorname{prox}_{\tau f}\left(x_k - \tau \nabla h(x_k) - \tau K^T y_k\right), \\ y_{k+1} = \operatorname{prox}_{\sigma g^*}\left(y_k + \sigma K(2x_{k+1} - x_k)\right). \end{cases}
$$