

Lecture 27: Consensus Equilibrium for Two Models

Charles A. Bouman

Purdue University

Wednesday, October 28, 2026

Outline

- 1 If H is not a proximal map, what is PnP doing?
- 2 The consensus equilibrium (CE) equations
- 3 When F and H are proximal maps: CE solution = MAP estimate
- 4 Equivalence of CE and MAP estimates, with proof
- 5 When F and H are general agents: force balance
- 6 Solving the CE equations with serial agents: $T = (2H - I)(2F - I)$, Douglas–Rachford, Mann iterations
- 7 Solving the CE equations with parallel agents: $\mathbf{T} = (2\mathbf{G} - I)(2\mathbf{F} - I)$

Core idea

When H is not a proximal map, PnP is **no longer solving an optimization problem**. Instead it solves the **CE equations**, which replace optimization as the defining criterion for the solution.

Today's Big Ideas

- 1 The consensus equilibrium (CE) equations are the two equations that PnP-ADMM satisfies at its convergence point, and they replace optimization as the defining criterion for the solution.
- 2 When both agents are proximal maps of convex functions, the solution of the CE equations is exactly the MAP estimate.
- 3 When the agents are general operators, the CE equations describe a balance between opposing actions that minimizes no cost function, and this mixing of actions and objectives is the essence of the plug-and-play method.
- 4 The serial solution rewrites the CE equations as a fixed-point problem that Douglas–Rachford with Mann iterations solves, and that argument also supplies the convergence proof for PnP-ADMM.
- 5 The parallel solution stacks the two states and uses an averaging operator, which gives an algorithm that keeps a well-defined solution when the agents are not proximal maps and that can converge faster than consensus ADMM.

Where we are, and what is unresolved

- In Section 10.1 we introduced the PnP-ADMM algorithm and argued that it would work best if we replaced the proximal map, $H(x)$, with a more general denoising agent. In practice, this works quite well.
- In Section 10.2 we showed that a typical denoiser would **not** be a proximal map.

The questions this raises

- So if $H(x)$ is not a proximal map, then what exactly is the PnP algorithm doing?
- In fact, under what conditions does it even converge?
- And if it does converge, what does it converge to?

In some sense the core issue is that we started this discussion by trying to minimize the MAP cost function, but if $H(x)$ is not a proximal map, then **are we still even solving an optimization problem?**

Consensus equilibrium: the framework of this section

Consensus equilibrium (CE) is the theoretical framework we introduce in this section to answer these questions.

- We will see that, in general, PnP is no longer solving an optimization problem.
- Instead, it is an algorithm for finding a solution to a more general formulation that has the form of an **equilibrium equation**.

Why this matters in practice

CE will allow us to fuse sets of heterogeneous models represented by **both actions and objectives** using a consistent theoretical framework. This is important because it can serve as a basis for integrating machine learning and physics-based models when solving inverse problems.

The CE equations

Look more closely at the PnP-ADMM algorithm:

$$x \leftarrow F(v - u) , \quad v \leftarrow H(x + u) , \quad u \leftarrow u + (x - v) .$$

When the algorithm converges, it must be that $x = v$. Therefore, we know that the following two equations must hold at convergence:

$$\begin{aligned} x^* &= F(x^* - u^*) , \\ x^* &= H(x^* + u^*) . \end{aligned}$$

The defining criterion

The **CE equations** for our inverse problem are the two equations above. They are the defining criteria or formulation for the solution, x^* . This set of equations **replaces the optimization** $x^* = \arg \min_x \{f(x) + h(x)\}$ as our goal in solving the inverse problem.

When F and H are proximal maps

In the special case when $F(x)$ and $H(x)$ are proximal maps corresponding to the convex functions $f(x)$ and $h(x)$, then the solution to the CE equations, x^* , is **exactly the solution to the MAP optimization problem**.

- This is true because we know that the ADMM algorithm converges to the global minimum of a convex problem.
- So the equilibrium between the **action** operations of the CE equations is equivalent to minimizing the **objective** functions of the MAP estimate.

More directly, take the gradient of the CE equations with respect to x^* ; we can easily show that

$$\nabla [f(x) + h(x)]|_{x=x^*} = 0 ,$$

where x^* denotes the solution to the CE equations. So if $f(x)$ and $h(x)$ are convex, then x^* must also be the MAP estimate.

In some important cases, such as a total variation prior, f and h may be nonconvex as well as not continuously differentiable. We can still relate the CE equations to the MAP estimate, but we must use the concept of **subgradients**.

Theorem: Equivalence of CE and MAP Estimates

Equivalence of CE and MAP Estimates

Let F and H be proximal maps of proper closed convex functions f and h . Then x^* is a solution to the CE equations if and only if x^* is a global minimum to the optimization problem $x^* = \arg \min_x \{f(x) + h(x)\}$.

Proof. We first prove the **forward direction**. It is easily shown that if (x^*, u^*) is a solution to the CE equations, then we have that

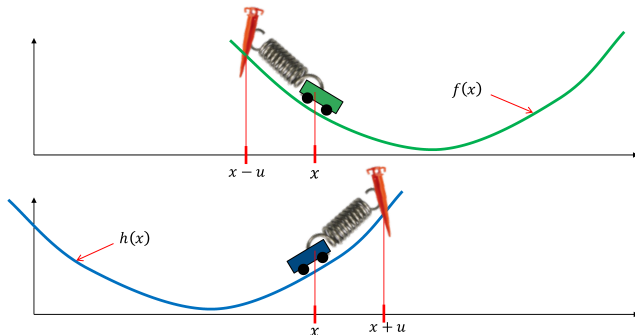
$$0 \in \nabla [f(x) + h(x)]|_{x=x^*}.$$

This then implies that x^* must be a global minimum to the proper closed convex function $f(x) + h(x)$.

We next prove the **reverse direction**. If x^* is a global minimum of $f(x) + h(x)$ and we choose $u^* \in -\sigma^2 \nabla f(x^*)$, then it can be shown that (x^*, u^*) satisfies the CE equations.

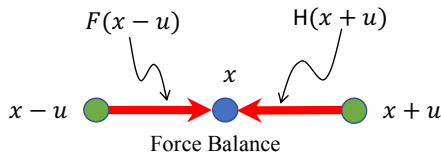
□

CE solution when the agents are proximal maps



The figure shows the two functions $f(x)$ and $h(x)$ separately, each with its own proximal map solution represented by the position of a cart, with each spring stretched by the same amount but in the opposite directions of $\pm u$. The minimum of the sum of the two functions occurs when the two carts fall to the **same location** and their respective stakes are located at $x^* - u$ and $x^* + u$ for some u . In this case it is easy to show that $\nabla[f(x) + h(x)] = 0$, so for strictly convex problems the CE solution is the unique solution to the MAP estimate.

When F and H are general agents



- In this case, the CE equations represent a **force balance equilibrium** in which agents are acting in opposing directions with equal action.
- However, this equilibrium *does not* result from the minimization of any underlying cost function, because the agents are not proximal maps.

The essence of PnP

The action operations used in the CE equations are much more general than the objective functions of the MAP estimate. This **mixing of actions and objectives is the essence of the PnP method**, and it allows us to fuse prior models represented by the actions of DNNs with the objectives of traditional sensor models. Whereas the convergence of the PnP algorithm was questionable, the CE equations give us a firm criterion for stating the solution.

Serial agents: a change of variables

We have expressed the solution of our problem as the solution to the CE equations. But: **do the CE equations have a solution? And if they do, how can we compute it?**

We first reorganize the CE equations into a more elegant form. Introduce the change of variables $w_1 = x - u$ and $w_2 = x + u$, so the CE equations become

$$F(w_1^*) = \frac{w_1^* + w_2^*}{2},$$

$$H(w_2^*) = \frac{w_1^* + w_2^*}{2}.$$

We can always convert back to the original variables using

$$x^* = \frac{w_1^* + w_2^*}{2}, \quad u^* = \frac{w_2^* - w_1^*}{2}.$$

Reflection form and the fixed-point operator T

Simple algebra rewrites these equations in the new form

$$(2F - I) w_1^* = w_2^*,$$

$$(2H - I) w_2^* = w_1^*,$$

where for simplicity we use the notation $Fw = F(w)$ to denote the application of the function F . Substituting in for w_2 results in

$$(2H - I)(2F - I) w_1^* = w_1^*.$$

Fixed point

In other words, w_1^* is the CE solution **if and only if** it is a fixed point of the operator $T = (2H - I)(2F - I)$, where a fixed point of T is any solution to $Tw_1^* = w_1^*$.

This is important because there are many robust algorithms for computing the fixed points of operators.

Douglas–Rachford with Mann iterations

The **Douglas–Rachford** (DR) algorithm with **Mann iterations** solves the fixed-point problem. The Mann iterations have the form

$$w_1 \leftarrow (1 - \rho)w_1 + \rho Tw_1 ,$$

where $\rho \in (0, 1)$ is a user-selectable parameter.

Convergence condition

When the operator T is **nonexpansive**, then the Mann iterations are guaranteed to converge to a fixed point of T if one exists.^a

^aIf T is contractive, then a fixed point is also guaranteed to exist and $\rho \in (0, 1]$.

Once w_1^* is computed, w_2^* is obtained from $w_2^* = (2F - I)w_1^*$, and then the solution to the CE equations follows from $x^* = \frac{w_1^* + w_2^*}{2}$ and $u^* = \frac{w_2^* - w_1^*}{2}$.

The PnP-DR-Serial algorithm

PnP-DR-Serial Algorithm

Initialize Reconstruction: $w_1 \in \mathbb{R}^N$

Until Converged {

$$w_1' \leftarrow (2H - I)(2F - I)w_1$$

$$w_1 \leftarrow (1 - \rho)w_1 + \rho w_1',$$

}

$$w_2^* \leftarrow (2F - I)w_1^*$$

$$x^* \leftarrow \frac{w_1^* + w_2^*}{2}$$

Return(x^*)

This computes the solution to the CE equations for plug-in forward or prior agents $F(x)$ and $H(x)$, with user-selectable parameter $\rho \in (0, 1)$. It is provably convergent when $T = (2H - I)(2F - I)$ is nonexpansive.

Theorem: Serial PnP Convergence

Serial PnP Convergence

If the operator $T = (2H - I)(2F - I)$ is nonexpansive and T has a fixed point, then the PnP-DR-serial algorithm converges to a solution of the CE equations.

Proof. This is a direct result of the Mann iteration theorem of Appendix B. □

- So when we plug in a denoising agent for H , a **sufficient** condition for convergence of the PnP algorithm is that T is nonexpansive.
- In practice, the PnP algorithm will typically numerically converge even when T is not nonexpansive, but in this case it is more difficult to guarantee theoretical convergence.

What the serial result gives us

- When F and H are both proximal maps, it can be shown that $T = (2H - I)(2F - I)$ is nonexpansive and the PnP-DR-serial algorithm converges to **both the CE solution and the MAP estimate**.
- When $\rho = 0.5$, the PnP-DR-serial algorithm is **exactly equivalent** to the PnP-ADMM algorithm. Therefore, this is really a proof of the convergence of the ADMM algorithm, which we never really gave in Chapter 9.
- PnP has a clear interpretation even when F or H is a general agent. It can no longer be viewed as solving the MAP optimization problem, but it can still be viewed as a method for solving the CE equations.

A second advantage of PnP-DR-serial is that, unlike ADMM, it is not restricted to $\rho = 0.5$. In practice, it is often possible to get much more rapid convergence by using **larger values of ρ** , thereby saving computation.

Parallel agents: stacked variables and operators

Stack the two states w_1 and w_2 into a new variable $\mathbf{w}$, and stack the two operators F and H into a corresponding operator $\mathbf{F}(\mathbf{w})$:

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, \quad \mathbf{F}(\mathbf{w}) = \begin{bmatrix} F(w_1) \\ H(w_2) \end{bmatrix}.$$

Next we introduce a subtly important new operator. The operator $\mathbf{G}(\mathbf{w})$ computes the **average** of the two states in $\mathbf{w}$:

$$\mathbf{G}(\mathbf{w}) = \begin{bmatrix} \bar{w} \\ \bar{w} \end{bmatrix}, \quad \text{where} \quad \bar{w} = \frac{w_1 + w_2}{2}.$$

The CE equations in compact form, and the key property of **G**

With this notation, the CE equations take a much more compact and elegant form:

$$\mathbf{F}(\mathbf{w}^*) = \mathbf{G}(\mathbf{w}^*) .$$

The averaging function is deceptively simple but powerful

A key property of **G** is that if you apply it twice, you get the same result, because the average of the average is the average:

$$\mathbf{G}\mathbf{G} = \mathbf{G} .$$

This also means that $(2\mathbf{G} - I)^2 = 4\mathbf{G}\mathbf{G} - 4\mathbf{G} + I = I$, which in turn implies that $(2\mathbf{G} - I)^{-1} = (2\mathbf{G} - I)$.

The parallel fixed-point operator $\mathbf{T}$

Using this fact, we get the following important results:

$$\begin{aligned}\mathbf{F}\mathbf{w}^* &= \mathbf{G}\mathbf{w}^* , \\ (2\mathbf{F} - I)\mathbf{w}^* &= (2\mathbf{G} - I)\mathbf{w}^* , \\ (2\mathbf{G} - I)(2\mathbf{F} - I)\mathbf{w}^* &= \mathbf{w}^* .\end{aligned}$$

So $\mathbf{w}^*$ is the CE solution if and only if it is a fixed point of $\mathbf{T} = (2\mathbf{G} - I)(2\mathbf{F} - I)$, that is $\mathbf{T}\mathbf{w}^* = \mathbf{w}^*$.

Again we use Mann iterations to solve for this fixed point:

$$\mathbf{w} \leftarrow \rho \mathbf{T}\mathbf{w} + (1 - \rho)\mathbf{w} , \quad \rho \in (0, 1) .$$

When $\mathbf{T}$ is nonexpansive, the Mann iterations are guaranteed to converge to a fixed point of $\mathbf{T}$ if one exists. Once converged, the solution is

$$\mathbf{x}^* = \frac{\mathbf{w}_1^* + \mathbf{w}_2^*}{2} .$$

The PnP-DR-Parallel algorithm

PnP-DR-Parallel Algorithm

Initialize Reconstruction: $x^{(0)} \in \mathbb{R}^N$

$$\mathbf{w} \leftarrow \begin{bmatrix} x^{(0)} \\ x^{(0)} \end{bmatrix}$$

Until Converged {

$$\mathbf{w}' \leftarrow (2\mathbf{G} - \mathbf{I})(2\mathbf{F} - \mathbf{I})\mathbf{w}$$

$$\mathbf{w} \leftarrow \rho\mathbf{w}' + (1 - \rho)\mathbf{w} ,$$

}

$$x^* \leftarrow \frac{w_1 + w_2}{2}$$

Return(x^*)

When is $\mathbf{T}$ nonexpansive? If F and H are both **firmly nonexpansive**, as is the case when they are proximal maps, then $\mathbf{T}$ is nonexpansive.

Two advantages over consensus ADMM

When $\rho = 0.5$ and F and H are proximal maps, $\mathbf{T}$ is nonexpansive and PnP-DR must converge to a fixed point. That fixed point is the solution to the underlying MAP optimization problem using variable splitting along with the **consensus ADMM** algorithm.

First advantage

PnP-DR has a well-defined solution **even when F and H are not proximal maps**. ADMM is designed to minimize a cost function, but if F and H are not proximal maps, then there is no underlying cost function to minimize. PnP-DR is still designed to solve the CE equations.

Second advantage

Using **larger values of ρ** typically gives much more rapid convergence, saving computation. Moreover, we can borrow known properties of the DR algorithm to understand both convergence and the existence of a CE solution. PnP-DR-parallel converges when $\mathbf{T}$ is nonexpansive — a far weaker condition than being a proximal map, which is assumed by ADMM.

Summary

- Setting $x = v$ at the convergence point of PnP-ADMM gives the **CE equations** $x^* = F(x^* - u^*)$ and $x^* = H(x^* + u^*)$. They replace optimization as the defining criterion for the solution.
- If F and H are proximal maps of proper closed convex f and h , then x^* solves the CE equations if and only if x^* is a global minimum of $f(x) + h(x)$.
- For general agents, the CE equations are a force balance equilibrium that does not result from minimizing any underlying cost function. Mixing actions and objectives is the essence of PnP.
- Serial solution: with $w_1 = x - u$ and $w_2 = x + u$, w_1^* is the CE solution if and only if it is a fixed point of $T = (2H - I)(2F - I)$. Douglas–Rachford with Mann iterations converges when T is nonexpansive. At $\rho = 0.5$ this is exactly PnP-ADMM, which supplies the ADMM convergence proof.
- Parallel solution: with stacked $\mathbf{w}$, stacked $\mathbf{F}$, and the averaging operator $\mathbf{G}$, the CE equations are $\mathbf{F}(\mathbf{w}^*) = \mathbf{G}(\mathbf{w}^*)$. Since $\mathbf{G}\mathbf{G} = \mathbf{G}$, the solution is a fixed point of $\mathbf{T} = (2\mathbf{G} - I)(2\mathbf{F} - I)$.

Reading: Chapter 10.3. Next lecture: Chapter 10.4, Multiagent consensus equilibrium.