Хорошо. В таком «авральном» случае возьмем вариационные неравенства:

$$\text{найти } x \in C: \quad \left(Ax, y-x\right) \geq 0 \quad \forall y \in C. \tag{1}$$

Предположим, что выполнены следующие условия:

- множество $C \subseteq H$ – выпуклое и замкнутое;
- оператор $A: H \to H$ – монотонный и липшицевый (L – константа Липшица);
- множество решений (1) S не пусто.

Из многообразия методов решения (1) возьмем три базовых метода.

Алгоритм 1 (Корпелевич).

Инициализация. Выбираем элементы x_1, $\lambda \in \left(0, \frac{1}{L}\right)$,. Полагаем $n = 1$.

Шаг 1. Вычислить

$$y_n = P_C\left(x_n - \lambda A x_n\right).$$

Если $x_n = y_n$, то СТОП и x_n – решение, иначе перейти на шаг 2.

Шаг 2. Вычислить

$$x_{n+1} = P_C\left(x_n - \lambda A y_n\right).$$

Положить $n := n+1$ и перейти на шаг 1.

Алгоритм 2 (P. Tseng).

Инициализация. Выбираем элементы x_1, $\lambda \in \left(0, \frac{1}{L}\right)$,. Полагаем $n = 1$.

Шаг 1. Вычислить

$$y_n = P_C\left(x_n - \lambda A x_n\right).$$

Если $x_n = y_n$, то СТОП и x_n – решение, иначе перейти на шаг 2.

Шаг 2. Вычислить

$$x_{n+1} = y_n - \lambda\left(A y_n - A x_n\right).$$

Положить $n := n+1$ и перейти на шаг 1.

Алгоритм 3 (Попов).

Инициализация. Выбираем элементы x_1, $y_0 \in C$, $\lambda \in \left(0, \frac{1}{3L}\right)$. Полагаем $n = 1$.

Шаг 1. Вычислить

$$y_n = P_C\left(x_n - \lambda A y_{n-1}\right).$$

Шаг 2. Вычислить

$$x_{n+1} = P_C\left(x_n - \lambda A y_n\right).$$

Если $x_{n+1} = x_n = y_n$, то остановить и x_n – решение. Иначе положить $n := n+1$ и перейти на шаг 1.

В последнее время появились адаптивные схемы на базе описанных, не требующие знания константы Липшица. Привожу их.

Алгоритм 4 (адапт. Корпелевич).

Инициализация. Выбираем элемент $x_1 \in C$, $\tau \in (0,1)$, $\lambda_1 \in (0, +\infty)$. Полагаем $n = 1$.

Шаг 1. Вычислить

$$y_n = P_C\left(x_n - \lambda_n A x_n\right).$$

Если $x_n = y_n$, то остановить и x_n – решение. Иначе перейти на шаг 2.

Шаг 2. Вычислить

$$x_{n+1} = P_C\left(x_n - \lambda_n A y_n\right).$$

Шаг 3. Вычислить

$$\lambda_{n+1} = \begin{cases} \lambda_n, & \text{если} \quad \left(Ax_n - Ay_n, x_{n+1} - y_n\right) \le 0, \\ \min\left\{\lambda_n, \dfrac{\tau}{2} \dfrac{\|x_n - y_n\|^2 + \|x_{n+1} - y_n\|^2}{\left(Ax_n - Ay_n, x_{n+1} - y_n\right)}\right\}, & \text{иначе.} \end{cases}$$

Положить $n := n+1$ и перейти на шаг 1.

Замечание. В алгоритме 4 можно делать и так

$$\lambda_{n+1} = \begin{cases} \lambda_n, & \text{если} \quad Ax_n - Ay_n = 0, \\ \min\left\{\lambda_n, \tau \dfrac{\|x_n - y_n\|}{\|Ax_n - Ay_n\|}\right\}, & \text{иначе.} \end{cases}$$

Алгоритм 5 (адапт. P. Tseng).

Инициализация. Выбираем элемент $x_1 \in C$, $\tau \in (0,1)$, $\lambda_1 \in (0,+\infty)$. Полагаем $n=1$.

Шаг 1. Вычислить

$$y_n = P_C\left(x_n - \lambda_n A x_n\right).$$

Если $x_n = y_n$, то остановить и x_n – решение. Иначе перейти на шаг 2.

Шаг 2. Вычислить

$$x_{n+1} = y_n - \lambda_n\left(Ay_n - Ax_n\right).$$

Шаг 3. Вычислить

$$\lambda_{n+1} = \begin{cases} \lambda_n, & \text{если} \quad Ax_n - Ay_n = 0, \\ \min\left\{\lambda_n, \tau \dfrac{\|x_n - y_n\|}{\|Ax_n - Ay_n\|}\right\}, & \text{иначе.} \end{cases}$$

Положить $n := n+1$ и перейти на шаг 1.

Алгоритм 6 (адапт. Попов).

Инициализация. Выбираем элементы x_1, $y_0 \in C$, $\tau \in \left(0,\frac{1}{3}\right)$, $\lambda_1 \in (0,+\infty)$. Полагаем $n=1$.

Шаг 1. Вычислить

$$y_n = P_C\left(x_n - \lambda_n A y_{n-1}\right).$$

Шаг 2. Вычислить

$$x_{n+1} = P_C\left(x_n - \lambda_n A y_n\right).$$

Если $x_{n+1} = x_n = y_n$, то остановить и x_n – решение. Иначе перейти на шаг 3.

Шаг 3. Вычислить

$$\lambda_{n+1} = \begin{cases} \lambda_n, & \text{если} \quad \left(Ay_{n-1} - Ay_n, x_{n+1} - y_n\right) \le 0, \\ \min\left\{\lambda_n, \dfrac{\tau}{2} \dfrac{\|y_{n-1} - y_n\|^2 + \|x_{n+1} - y_n\|^2}{\left(Ay_{n-1} - Ay_n, x_{n+1} - y_n\right)}\right\}, & \text{иначе.} \end{cases}$$

Положить $n := n+1$ и перейти на шаг 1.

Замечание. В алгоритме 6 можно делать и так

$$\lambda_{n+1} = \begin{cases} \lambda_n, & \text{если} \quad Ay_{n-1} - Ay_n = 0, \\ \min\left\{\lambda_n, \tau \dfrac{\|y_{n-1} - y_n\|}{\|Ay_{n-1} - Ay_n\|}\right\}, & \text{иначе.} \end{cases}$$

А недавно Юра Малицкий предложил такую схему (мне очень нравится)

Алгоритм 7 (Ю. Малицкий, М. Tam).

Инициализация. Выбираем элементы x_1, $x_0 \in H$, $\lambda \in \left(0, \frac{1}{2L}\right)$.

$$x_{n+1} = P_C\left(x_n - \lambda A x_n - \lambda\left(Ax_n - Ax_{n-1}\right)\right).$$

Можно нарисовать такую схему с адаптацией.

Алгоритм 8 (адапт. Ю. Малицкий, М. Tam).

Инициализация. Выбираем элементы x_1, $x_0 \in H$, λ_1, $\lambda_0 > 0$, $\tau \in \left(0, \frac{1}{2}\right)$.

Шаг 1. Вычислить

$$x_{n+1} = P_C\left(x_n - \lambda_n A x_n - \lambda_{n-1}\left(Ax_n - Ax_{n-1}\right)\right).$$

Если $x_{n+1} = x_n = x_{n-1}$, то остановить и x_n – решение. Иначе перейти на шаг 2.

Шаг 2. Вычислить

$$\lambda_{n+1} = \min\left\{\lambda_n, \tau \frac{\|x_{n+1} - x_n\|}{\|Ax_{n+1} - Ax_n\|}\right\}$$

Положить $n := n + 1$ и перейти на шаг 1.

Предлагаю провести соревнование между этими алгоритмами на задачах из посланных статей. Нужно выяснить, какой из этих методов лучший. И еще: возможна настроика алг. 8 так, что он будет быстрее метода «Algorithm 1 The forward-reflected-backward method with linesearch» из статьи Malitsky Y, Tam MK A Forward-Backward splitting method for monotone inclusions without cocoercivity. arXiv:1808.04162. 2018?