Here are some remarks on Example 4.4 in [1]. I have corrected the formulae in the example and support my results with a grid refinement study.

I am planning to test some solvers and use initial iterates that satisfy the boundary conditions next. An initial iterate that does not satisfy the boundary conditions is not a good choice for a PDE problem. It is easy to compute one that does by solving $\nabla^2 w = 0$ with the boundary conditions.

The example is on $\Lambda = (0,1) \times (0,1)$

$$-\nabla^2 u + N(u) = c \tag{1}$$

subject to boundary conditions

$$u = v \text{ on } \partial\Lambda \tag{2}$$

and a free boundary condition

$$u = 0 \text{ in } \Lambda_0 \text{ and } u = \|\nabla u\| = 0 \text{ on } \partial\Lambda_0. \tag{3}$$

The domain Λ_0 is unknown. As is stands this problem is not well posed unless you allow Λ_0 to be empty because for a general N and c there is no guarantee that $u = 0$ anywhere in Λ.

In this particular case

$$N(u) = \frac{9}{(1-p)^2} u^p + \delta e^{-u}.$$

and you generate c so the solution of the problem is

$$v(\xi,\zeta) = \left(\frac{3r-1}{2}\right)^{2p/(1-p)} \max(0, r - 1/3)$$

So $\Lambda_0 = \{(\xi,\zeta), |, \sqrt{\xi^2+\zeta^2} > 1/3\}$

Once you have done this, the free boundary has been encoded in the right side c and the boundary conditions, so simply solving the equation will recover the free boundary with no additional work.

So, we must have

$$c = -\nabla^2 v + N(v). \tag{4}$$

I will show that

$$\nabla^2 v = (3/2)\sigma(\sigma+1)T^{\sigma-1} + \frac{1}{r}(\sigma+1)T^{\sigma}$$

Here

$$T = \frac{3r-1}{2} \max(0, r-1/3)\chi^*(r) \text{ and } \sigma = 2p/(1-p)$$

where χ^* is the characteristic function of $(1/3, \infty)$. So, we have

$$v = T^{\sigma} \max(0, r-1/3) = \frac{2}{3} T^{\sigma+1}.$$

1 Laplacian in Polar Coordinates

We will use the fact that v depends only of r and not the angular variable to obtain

$$\nabla^2 v = v_{rr} + \frac{1}{r} v_r$$

For $r > 1/3$, $T_r = 3/2$ so

$$v_r = (\sigma+1)T^{\sigma}$$

and

$$v_{rr} = \sigma(\sigma+1)T^{\sigma-1}(3/2).$$

So my claim is that

$$\nabla^2 v = (\sigma+1)(T^\sigma + (3/2)\sigma T^{\sigma-1}). \tag{5}$$

This is a calculation that is easy to compute incorrectly because one must get the boundary conditions incorporated in to the discrete equations in the proper order. You do this in [1] with the vector $\tilde{v}$ which you add to c to form the discrete right hand side. The paper [1] does not define $\tilde{v}$, but I think I know what you did and it should be correct. Numerical testing should confirm that and demonstrate that the discretization is correct. I do it by implicitly adding $\tilde{v}$ to the discretization of c as part of the discretization of c.

I will test the Laplacian with a grid refinement study in the following way using $p = .8$ and $\delta = 1$.

1. Evaluate v on an $N \times N$ numerical grid with stepisze $h = 1/(N+1)$ on Λ.
   (a) Compute the analytic Laplacian with (5) L_e.
   (b) Compute the finite difference Laplacian L_d. This computation also tests the boundary data.
2. Tabulate the relative error $E_r = \|L_e - L_d\|/\|L_e\|$ as a function of N for the $\ell^1, \ell^2, \ell^\infty$ norms.

We should see the errors decay by a factor of roughly four in all cases, and they do.

Table 1: Laplacian errors

N	ℓ^1	ℓ^2	ℓ^∞	
10	3.894e-02	2.971e-02	2.400e-02	
20	9.974e-03	7.563e-03	5.808e-03	:w
40	2.534e-03	1.917e-03	1.431e-03	
80	6.394e-04	4.832e-04	3.552e-04	

2 The equations

All seems to be fine in [1] except for the right hand side function c. It should be

$$c = -\nabla^2 v + N(v) = (3/2)\sigma(\sigma+1)T^{\sigma-1} + \frac{1}{r}(\sigma+1)T^\sigma + N(v). \tag{6}$$

Recall that

$$T = \frac{3r-1}{2}\max(0, r-1/3)\chi^*(r), \sigma = 2p/(1-p)$$

χ^* is the characteristic function of $(1/3, \infty)$, and so

$$v = T^\sigma \max(0, r-1/3) = \frac{2}{3}T^{\sigma+1}.$$

In my discretization I do not multiply by h^2 as you do at the top of page 19 in [1]. The reason for this is that I will want to look at preconditioning and fast solvers expect the h^{-2} to be part of the operator.

I will do another grid refinement study by computing the discrete form of H (which includes the boundary conditions) and evaluating the discrete form of $H(v)$. This should be $O(h^2)$. This study confirms that the discretization is consistent and that v and c are also consistent.

Table 2: Values of $\|H(v)\|/\|v\|$ as a function of N

n	ℓ^1	ℓ^2	ℓ^∞
10	4.705e+00	2.934e+00	2.082e+00
20	1.127e+00	6.923e-01	4.454e-01
40	2.772e-01	1.694e-01	1.030e-01
80	6.884e-02	4.199e-02	2.476e-02

3 Solving with AA

We will consider two formulations. The original approach from [1] is a differential equations formulation and we begin with that one in § 3.1.

The natural approach, which leads directly to the integral equations formulation in § 3.2, is to use an initial iterate that satisfies the boundary conditions, then the line search is not needed and one can use AA directly. In particular, the intial iterate will be the solution of Laplace's equation

$$-\nabla^2 u_0 = 0 \text{ in } \Lambda \; u_0 = v \text{ on } \partial\lambda. \tag{7}$$

We can use (7) to write the solution as in integral form. We define $\mathcal{S}(f; v)$ to be the solution operator of (7). If $v = 0$ the $\mathcal{S}(f; 0) = \mathcal{L}(f)$ is an linear integral operator

$$\mathcal{L}f(x, y) = \int_\Lambda G(x, y, \xi, \eta) f(\xi, \eta) dA$$

where

$$G(x, y, \xi, \eta) = \frac{1}{4\pi} \ln((\xi - x)^2 + (\eta - y)^2).$$

Then

$$\mathcal{S}(f; v) = \mathcal{L}(f) + \mathcal{S}(0; v).$$

Of course, when we compute $\mathcal{L}(f)$ we will use an efficient solver rather than evaluate the two dimensional integral.

3.1 Differential Equations Formulation

For the computions in this section I report in this section I use (in your terminology)

$$\gamma = .1$$

and multiply the discrete residual by h^2 as you do.

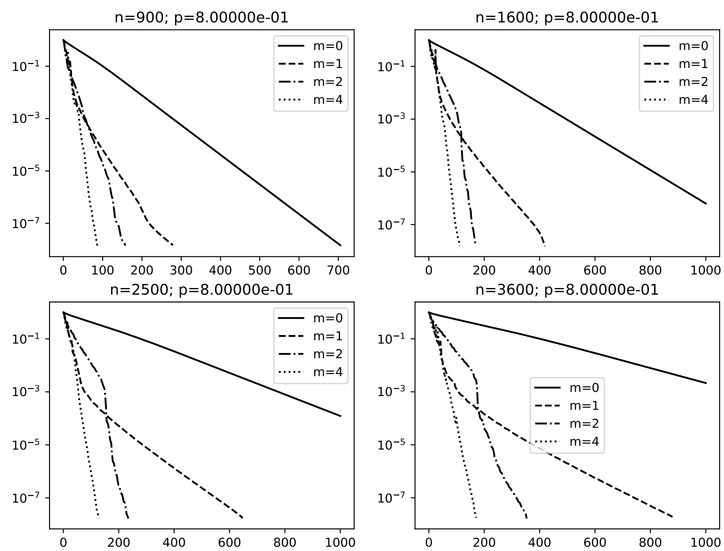
My discretiztion is (I think) exactly the same as yours.

With this intial iterate AA does very well. Here are some results for $p = .8$, $m = 0, 1, 2, 4$, and $n = 900, 1600, 2500$.

It's interesting to see that the performance of $AA(2)$ and $AA(4)$ does not degrade as rapidly with n as $AA(0)$ and $AA(1)$.

3.2 Integral Equation Formulation

Figure 1: AA Performance: Original formulation



References

[1] X. Qu, W. Bian, and X. Chen, *An extra gradient Anderson-accelerated algorithm for pseudomonotone variational inequalities*, 2024, https://arxiv.org/abs/2408.06606, https://arxiv.org/abs/2408.06606.