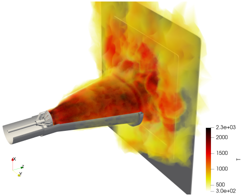
High-order adaptive multistep coupling scheme for multiphysics applications

[Minisymposium] Méthodes parallèles pour la résolution et la simulation d'équations différentielles

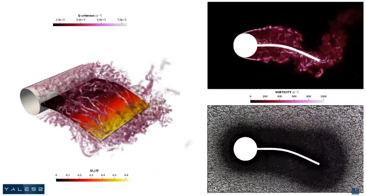
Antoine Simon[1,2], Laurent François[2], Marc Massot[1]

[1]CMAP – École polytechnique, [2]ONERA

Code coupling for unsteady multiphysics simulations

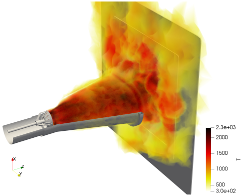


Fluid – thermal ([1] Letournel *et al* 2025)

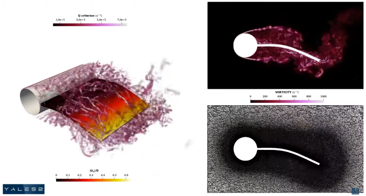


Fluid – structure ([2] Fabbri *et al*)

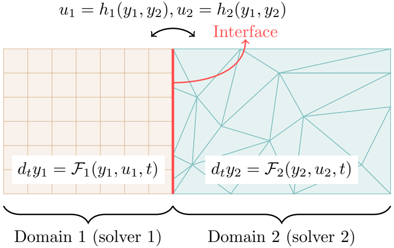
Code coupling for unsteady multiphysics simulations



Fluid – thermal ([1] Letournel *et al* 2025)



Fluid – structure ([2] Fabbri *et al*)



Multiphysics system —model→ Coupled PDEs —space discretization→ Coupled ODEs

↓

$$\begin{cases} d_t y_i = \mathcal{F}_i(y_i, u_i, t) & i = 1 \dots M \\ u_i = h_i(y_1, \dots, y_M) \end{cases}$$

(u_i: coupling variables)

← *Partitionned* approach

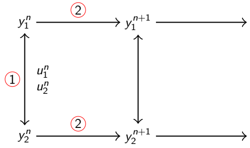
Parallel integration of each subsystem i using a specific solver

The high-order time adaptive multistep coupling scheme

- Usually:

Integration of
$$d_t y_i = \mathcal{F}_i(y_i, u_i^n, t)$$
constant

$\Rightarrow$ Convergence **at order 1**



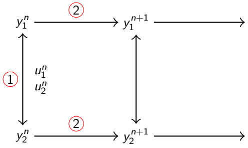
Conventional Parallel Staggered coupling scheme ([3] Farhat and Lesoinne 2000)

The high-order time adaptive multistep coupling scheme

- Usually:

Integration of
$$d_t y_i = \mathcal{F}_i(y_i, u_i^n, t)$$
constant

$\Rightarrow$ Convergence **at order 1**



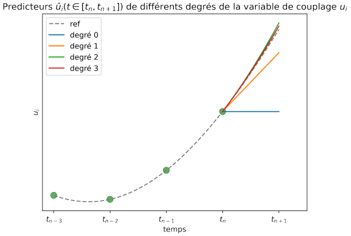
Conventional Parallel Staggered coupling scheme ([3] Farhat and Lesoinne 2000)

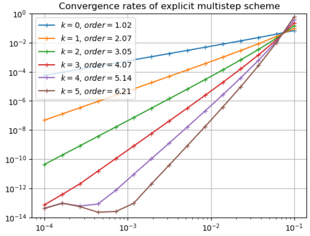
- The high-order multistep coupling scheme (explicit) ([4] François and Massot 2023, [5] Simon *et al* 2025):

Integration of
$$d_t y_i = \mathcal{F}_i(y_i, \hat{u}_i^n, t)$$
polynomial extrapolated from the past $u_i^{n-k} \ldots u_i^n$

$\Longrightarrow$ Convergence **at order k + 1**

k = degree of the interpolation polynomial



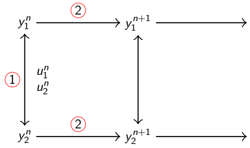


The high-order time adaptive multistep coupling scheme

- Usually:

Integration of
$d_t y_i = \mathcal{F}_i(y_i, u_i^n, t)$
constant

$\Rightarrow$ Convergence **at order 1**



Conventional Parallel Staggered coupling scheme ([3] Farhat and Lesoinne 2000)

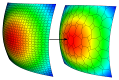
- The high-order multistep coupling scheme (explicit) ([4] François and Massot 2023, [5] Simon *et al* 2025):

Integration of
$d_t y_i = \mathcal{F}_i(y_i, \hat{u}_i^n, t)$
polynomial extrapolated from the past $u_i^{n-k} \dots u_i^n$

$\Longrightarrow$ Convergence **at order k + 1**

k = degree of the interpolation polynomial

Implementations:



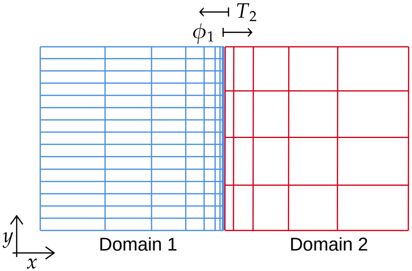
CWIPI

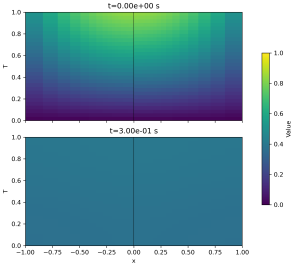
C++ HPC coupling library developed at ONERA (https://w3.onera.fr/cwipi/)

RHAPSOPY

Python coupling library for prototype purposes (https://github.com/hpc-maths/Rhapsopy)

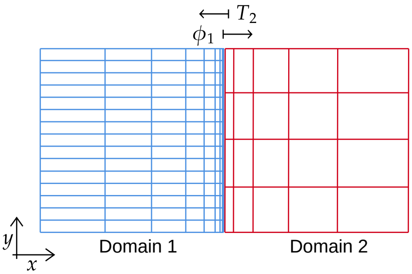
A representative conjugate heat transfer test case

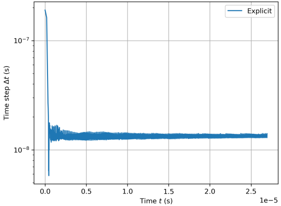




- Non-uniform Cartesian meshes (finite volume scheme)
- Neumann – Dirichlet interface BCs
- Homogeneous Neumann BCs on the exterior
- Non uniform initial temperature field

A representative conjugate heat transfer test case





Time steps size adaptive order 3 explicit multistep coupling scheme

- Non-uniform Cartesian meshes (finite volume scheme)
- Neumann – Dirichlet interface BCs
- Homogeneous Neumann BCs on the exterior
- Non uniform initial temperature field

Explicit: ✖ stability issues ([6] Simon *et al* 2026)

Towards the implicit multistep coupling scheme

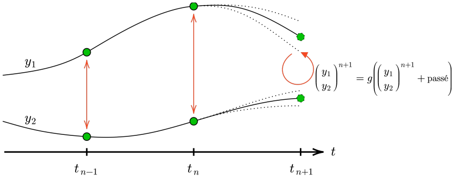
The multistep scheme: $$y_i^{n+1} = y_i^n + \int_{t_n}^{t_{n+1}} \mathcal{F}_i(y_i(\eta), \hat{u}_i^n(\eta), \eta)d\eta \quad (1)$$

- Explicit $\hat{u}_i^n = \Psi(u_i^n, \dots, u_i^{n-k}, t_n, \dots, t_{n-k})$ $\quad \hat{u}_i^n(t) \rightsquigarrow$ interpolates $u_i^{n-k} \dots u_i^n$

Towards the implicit multistep coupling scheme

The multistep scheme: $$y_i^{n+1} = y_i^n + \int_{t_n}^{t_{n+1}} \mathcal{F}_i(y_i(\eta), \hat{u}_i^n(\eta), \eta)d\eta \quad (1)$$

- Explicit $\hat{u}_i^n = \Psi(u_i^n, \ldots, u_i^{n-k}, t_n, \ldots, t_{n-k})$ $\quad \hat{u}_i^n(t) \rightsquigarrow$ interpolates $u_i^{n-k} \ldots u_i^n$
- Implicit $\hat{u}_i^n = \Psi(u_i^{n+1}, \ldots, u_i^{n-k+1}, t_{n+1}, \ldots, t_{n-k+1})$ $\quad \hat{u}_i^n(t) \rightsquigarrow$ interpolates $u_i^{n-k+1} \ldots u_i^{n+1}$



Trade-off: computation cost vs stability and accuracy

Implicit coupling as a fixed-point problem

- $U_n \equiv (\underbrace{u_1^n, \dots, u_M^n}_{\text{present}}, \underbrace{\dots, u_1^{n-k}, \dots, u_M^{n-k}}_{\text{past}})^t$

$$y_i^{n+1} = y_i^n + \int_{t_n}^{t_{n+1}} \mathcal{F}_i(y_i(\eta), \hat{u}_i^n(\eta), \eta)d\eta$$

$$\downarrow$$

Implicit: $\hat{u}_i^n = \Psi(u_i^{n+1}, \dots, u_i^{n-k+1}, t_{n+1}, \dots, t_{n-k+1})$

Implicit coupling as a fixed-point problem

$$y_i^{n+1} = y_i^n + \int_{t_n}^{t_{n+1}} \mathcal{F}_i(y_i(\eta), \hat{u}_i^n(\eta), \eta) d\eta$$

$$\downarrow$$

Implicit: $\hat{u}_i^n = \Psi(u_i^{n+1}, \ldots, u_i^{n-k+1}, t_{n+1}, \ldots, t_{n-k+1})$

- $U_n \equiv (\underbrace{u_1^n, \ldots, u_M^n}_{\text{present}}, \underbrace{\ldots, u_1^{n-k}, \ldots, u_M^{n-k}}_{\text{past}})^t$
- $H \equiv$ state vector $\rightarrow$ coupling variables

Implicit coupling as a fixed-point problem

$$y_i^{n+1} = y_i^n + \int_{t_n}^{t_{n+1}} \mathcal{F}_i(y_i(\eta), \hat{u}_i^n(\eta), \eta)d\eta$$

$$\downarrow$$

Implicit: $\hat{u}_i^n = \Psi(u_i^{n+1}, \dots, u_i^{n-k+1}, t_{n+1}, \dots, t_{n-k+1})$

- $U_n \equiv (\underbrace{u_1^n, \dots, u_M^n}_{\text{present}}, \underbrace{\dots, u_1^{n-k}, \dots, u_M^{n-k}}_{\text{past}})^t$
- $H \equiv$ state vector $\rightarrow$ coupling variables
- $\Phi \equiv$ partitionned integration operator

Implicit coupling as a fixed-point problem

$$y_i^{n+1} = y_i^n + \int_{t_n}^{t_{n+1}} \mathcal{F}_i(y_i(\eta), \hat{u}_i^n(\eta), \eta) d\eta$$

$$\downarrow$$

Implicit: $\hat{u}_i^n = \Psi(u_i^{n+1}, \ldots, u_i^{n-k+1}, t_{n+1}, \ldots, t_{n-k+1})$

- $U_n \equiv (\underbrace{u_1^n, \ldots, u_M^n}_{\text{present}}, \underbrace{\ldots, u_1^{n-k}, \ldots, u_M^{n-k}}_{\text{past}})^t$
- $H \equiv$ state vector $\rightarrow$ coupling variables
- $\Phi \equiv$ partitionned integration operator
- $\Psi \equiv$ interpolation operator

Implicit coupling as a fixed-point problem

$$y_i^{n+1} = y_i^n + \int_{t_n}^{t_{n+1}} \mathcal{F}_i(y_i(\eta), \hat{u}_i^n(\eta), \eta)d\eta$$

$$\downarrow$$

Implicit: $\hat{u}_i^n = \Psi(u_i^{n+1}, \dots, u_i^{n-k+1}, t_{n+1}, \dots, t_{n-k+1})$

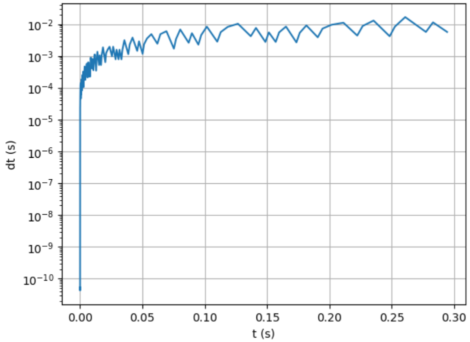
- $U_n \equiv (\underbrace{u_1^n, \dots, u_M^n}_{\text{present}}, \underbrace{\dots, u_1^{n-k}, \dots, u_M^{n-k}}_{\text{past}})^t$
- $H \equiv$ state vector $\rightarrow$ coupling variables
- $\Phi \equiv$ partitionned integration operator
- $\Psi \equiv$ interpolation operator

$$\Longrightarrow U_{n+1} = H \circ \Phi \circ \Psi(U_{n+1}, t_{n+1}, \dots, t_{n-k+1}) \quad (2)$$

$\rightsquigarrow$ In practice, assembling the jacobian of such system is **not possible**

Fixed-point acceleration necessity

Back to the conjugated heat transfer test case

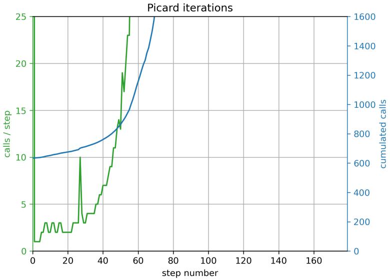


Time steps size implicit multistep coupling scheme (Picard iterations)

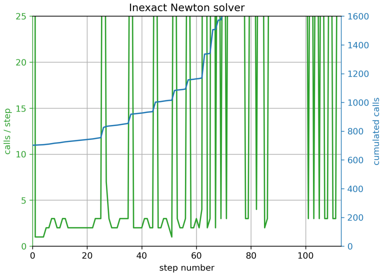
Fixed-point acceleration necessity

Back to the conjugated heat transfer test case

Order 3 implicit multistep coupling scheme ($t_f = 0.2$s)
Number of calls to the fixed-point function
[green: per time step ; blue: cumulated]



Picard iterations



Inexact Newton

$\Longrightarrow$ Inexact Newton unusable in practice BUT useful for comparison on small test case

Acceleration procedures for fixed-point problems

- ✔ Low-order steady / unsteady code coupling + acceleration ([7] Ramière and Helfer 2015)
- ✔ High-order implicit non-adaptive code coupling + quasi-Newton acceleration ([8] Rodenberg and Uekermann 2025)
- ✘ High-order adaptive implicit multistep coupling scheme + acceleration

Acceleration procedures for fixed-point problems

- ✔ Low-order steady / unsteady code coupling + acceleration ([7] Ramière and Helfer 2015)
- ✔ High-order implicit non-adaptive code coupling + quasi-Newton acceleration ([8] Rodenberg and Uekermann 2025)
- ✘ High-order adaptive implicit multistep coupling scheme + acceleration

Broad range of techniques with many variants:

- Extrapolation methods: Anderson Acceleration ...
- Krylov methods: GCR, GMRES ...
- Jacobian approximation methods: Quasi-Newton, Interface Quasi-Newton ...

Acceleration procedures for fixed-point problems

✓ Low-order steady / unsteady code coupling + acceleration ([7] Ramière and Helfer 2015)

✓ High-order implicit non-adaptive code coupling + quasi-Newton acceleration ([8] Rodenberg and Uekermann 2025)

✗ High-order adaptive implicit multistep coupling scheme + acceleration

Broad range of techniques with many variants:

- Anderson Acceleration / GCR / Quasi-Newton methods

At what cost ?

- Storing a Jacobian / set of iterates
- What parameters for AA, nlTGCR etc ?
- Scaling of coupling variables ?

Numerical results

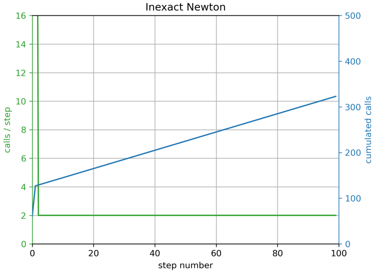
Back to the CHT test case: the setup and the references

Constant time steps $\Delta t = 3 \times 10^{-3}$s, final time $t_f = 0.3$s, order 3 implicit.
Number of calls to the fixed-point function
[green: per time step ; blue: cumulated]

- Reference: Picard iterations

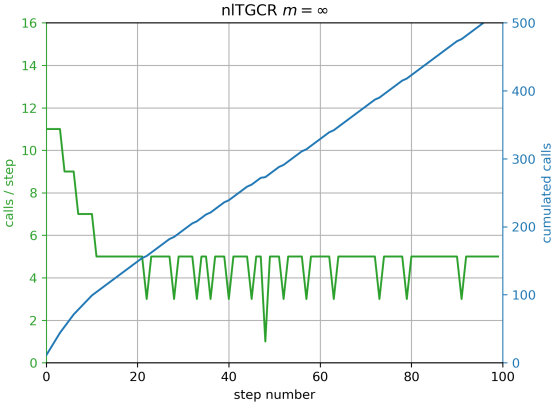
  ✖ First step does **not** converge !

- Reference: Inexact Newton method



Numerical results

Generic nlTGCR method ([9] He *et al* 2024)

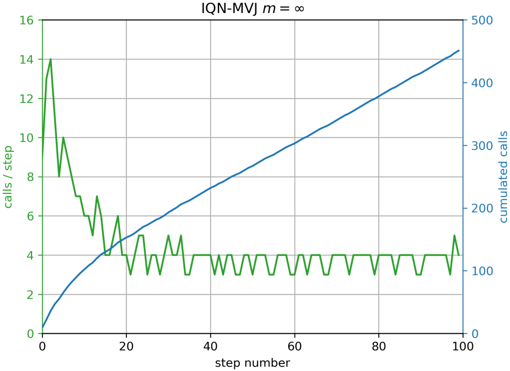


nlTGCR(∞): $n_{\text{calls}} \rightsquigarrow 5$

Numerical results

Quasi-Newton: IQN-MVJ ([10] Lindner *et al* 2015)

Evaluated in the context of code coupling in ([10] Lindner *et al* 2015).

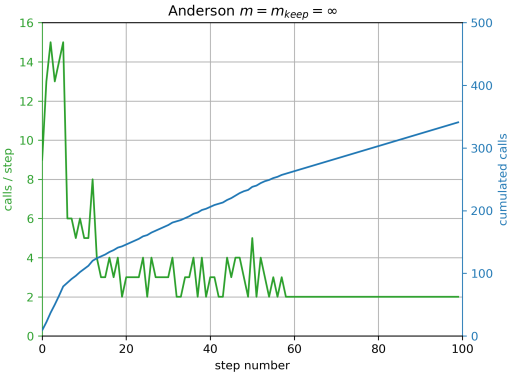


IQN-MVJ: $n_{\text{calls}} \rightsquigarrow 4$

Numerical results

Anderson Acceleration with reuse strategy

m_{keep}: number of vectors kept from last time step(s)



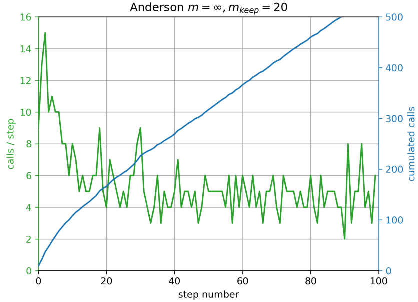
$$\text{AA}(m = m_{keep} = \infty)\text{: } n_{\text{calls}} \rightsquigarrow 2$$

Numerical results

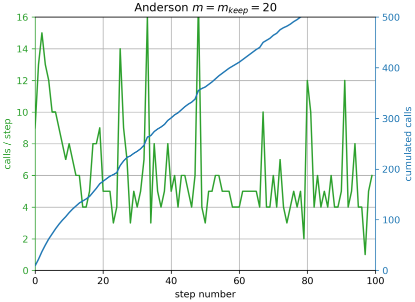
Anderson Acceleration with reuse strategy

m_{keep}: number of vectors kept from last time step(s)

Towards realistic choices for m and m_{keep}



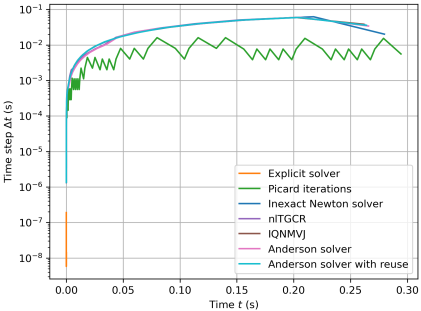
$\text{AA}(m = \infty, m_{keep} = 20)$: $n_{\text{calls}} \rightsquigarrow 5$



$\text{AA}(m = m_{keep} = 20)$: $n_{\text{calls}} \rightsquigarrow 6$

Adaptive multistep coupling order 3 accelerated

Adaptive tolerances : atol $= $ rtol $= 10^{-2}$ and atol$_{WR} =$ rtol$_{WR} = 10^{-4}$



Time steps Δt for adaptive order 3 multistep coupling scheme (embedded 4^{th}-order error estimate)

	Picard	Inexact Newton	nlTGCR	IQN-MVJ	**Anderson**	Anderson reuse
Total n_{calls}	8403	3341	472	682	**388**	594

Conclusion and perspectives

Result:

- A first overview of acceleration methods with the present benchmark

Ongoing investigation:

- Understanding the effect of the partitionned approach on the convergence of the different methods
- Information reuse when time step changes
- Domain of convergence of acceleration methods
- Effect of the scaling of the coupling variables

- Enhance benchmark with nonlinear solid – fluid coupling
- Precise characterization of “strong” / “stiff” couplings

Questions ?

References

[1] R. Letournel, A. Grenouilloux, K. Bioche, R. Mercier and V. Moureau,
Coupling strategies for multi-physics Large-Eddy Simulation: towards fire certification of composite material,
In *International Conference on Numerical Combustion*, Rome, Italy (October 2025),
https://hal.science/hal-05536669.

[2] T. Fabbri, G. Balarac, V. Moureau and P. Benard,
Design of a high fidelity Fluid–Structure Interaction solver using LES on unstructured grid,
Computers & Fluids, 265, p. 105963 (2023),
https://www.sciencedirect.com/science/article/pii/S0045793023001883.

[3] C. Farhat and M. Lesoinne,
Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems,
Computer methods in applied mechanics and engineering, 182 (3-4), pp. 499–515 (2000).

[4] L. François and M. Massot,
Multistep interface coupling for high-order adaptive black-box multiphysics simulations,
In *10th edition of the International Conference on Computational Methods for Coupled Problems in Science and Engineering*. CIMNE (2023),
https://www.scipedia.com/public/2023w.

[5] A. E. Simon, L. François and M. Massot,
High-order multistep coupling: convergence, stability and PDE application,
Comptes Rendus. Mécanique, 353, pp. 1159–1184 (2025).

[6] A. E. Simon, L. François, A. Asad, É. Quémerais and M. Massot,
Development of a high-order time-adaptive multistep coupling scheme in an HPC library for multiphysics applications,
Technical report NASA (2026).

[7] I. Ramière and T. Helfer,
Iterative residual-based vector methods to accelerate fixed point iterations,
Computers & Mathematics with Applications, 70 (9), pp. 2210–2226 (2015).

[8] B. Rodenberg and B. Uekermann,
A waveform iteration implementation for black-box multi-rate higher-order coupling,
arXiv preprint arXiv:2511.07616 (2025).

[9] H. He, Z. Tang, S. Zhao, Y. Saad and Y. Xi,
nltgcr: A class of nonlinear acceleration procedures based on conjugate residuals,
SIAM Journal on Matrix Analysis and Applications, 45 (1), pp. 712–743 (2024).

[10] F. Lindner, M. Mehl, K. Scheufele and B. Uekermann,
A comparison of various quasi-Newton schemes for partitioned fluid-structure interaction,
In *Coupled VI: Proceedings of the VI International Conference on Computational Methods for Coupled Problems in Science and Engineering*, pp. 477–488. CIMNE (2015).

[11] C. Brezinski, M. Redivo-Zaglia and A. Salam,
On the kernel of vector ε-algorithm and related topics,
Numerical Algorithms, 92 (1), pp. 207–221 (2023).

[12] S. C. Eisenstat, H. C. Elman and M. H. Schultz,
Variational iterative methods for nonsymmetric systems of linear equations,
SIAM Journal on numerical analysis, 20 (2), pp. 345–357 (1983).

[13] H. C. Elman,
Iterative methods for large, sparse, nonsymmetric systems of linear equations,
Yale University (1982).

[14] Y. Saad,
Acceleration methods for fixed-point iterations,
Acta Numerica, 34, pp. 805–890 (2025).