Acceleration methods for fixed point coupled problems iterations

Coupled Problems 2017
I. RAMIÈRE, T. HELFER (DEN, DEC, SESC)

12-14 JUNE 2017

Numerical Methods in Coupled Problems
Fixed point iterations acceleration

A generic residual-based acceleration method

Numerical experiments

Conclusions and perspectives
Fixed point iterations acceleration
Fixed point problems

Fixed point iterations

\[ X_{n+1} = F(X_n), \text{ until } \|F(X_n) - X_n\| \leq \varepsilon \]

are still the **most common approach** to dealing with a variety of numerical problems:

- **coupled problems**: partitioning multi-physics, domain decomposition, proper generalized decomposition,...
- **nonlinear problems**: heat transfer, nonlinear mechanics, electronic structure,...
Fixed point problems

Fixed point iterations

\[ X_{n+1} = F(X_n), \text{ until } \|F(X_n) - X_n\| \leq \varepsilon \]

are still the most common approach to dealing with a variety of numerical problems

😄: generic, no a priori knowledge on \( F \) (derivative,...)

😢: often slow and poor (linear) convergence
Fixed point problems

Fixed point iterations

\[ X_{n+1} = F(X_n), \text{ until } \|F(X_n) - X_n\| \leq \varepsilon \]

are still the most common approach to dealing with a variety of numerical problems

😊 : generic, no a priori knowledge on \( F \) (derivative,...)
😊 : often slow and poor (linear) convergence

\[ X_{n+1} = G(F(X_n), F(X_{n-1}), \ldots, X_n, X_{n-1}, \ldots), \text{ until } \|F(X_n) - X_n\| \leq \varepsilon \]

which aim to converge faster (in less iterations and computational cost) to the solution \( \bar{X} \).
Most popular and efficient acceleration method for first order sequences remains the $\Delta^2$ of Aitken (1926):

Aitken $\Delta^2$-method (static sequence transformation)

Assuming

$$x_{n+1} - \bar{x} = (\lambda + \varepsilon_n)(x_n - \bar{x}), \quad \lim_{n \to \infty} \varepsilon_n = 0$$

Then

$$y_n = x_n - \frac{(\Delta x_n)^2}{\Delta^2 x_n}$$

with

$$\Delta x_n = x_{n+1} - x_n$$

$$\Delta^2 x_n = \Delta x_{n+1} - \Delta x_n$$
Scalar sequences acceleration

Most popular and efficient acceleration method for first order sequences remains the $\Delta^2$ of Aitken (1926):

**Aitken $\Delta^2$-method (static sequence transformation)**

Assuming
\[
x_{n+1} - \bar{x} = (\lambda + \varepsilon_n)(x_n - \bar{x}), \quad \lim_{n \to \infty} \varepsilon_n = 0
\]

Then
\[
y_n = x_n - \frac{(\Delta x_n)^2}{\Delta^2 x_n}
\]

with
\[
\begin{align*}
\Delta x_n &= x_{n+1} - x_n \\
\Delta^2 x_n &= \Delta x_{n+1} - \Delta x_n
\end{align*}
\]

Dynamic (/cycling/recursive) sequence transformations based on the $\Delta^2$-method
Most popular and efficient acceleration method for first order sequences remains the $\Delta^2$ of Aitken (1926):

**Aitken $\Delta^2$-method (static sequence transformation)**

Assuming $x_{n+1} - \bar{x} = (\lambda + \varepsilon_n)(x_n - \bar{x})$, \[\lim_{n \to \infty} \varepsilon_n = 0\]

Then $y_n = x_n - \frac{(\Delta x_n)^2}{\Delta^2 x_n}$ with \[
\begin{align*}
\Delta x_n &= x_{n+1} - x_n \\
\Delta^2 x_n &= \Delta x_{n+1} - \Delta x_n
\end{align*}
\]

**Dynamic (/cycling/recursive) sequence transformations based on the $\Delta^2$-method**

- Steffensen algorithm

  \[
x_{n+1} = x_n - \frac{(f(x_n) - x_n)^2}{f(f(x_n)) - 2f(x_n) + x_n}
\]

- two-steps method
- second-order method
- but efficacy index 1.4
- converges even when the basic fixed point iterations diverge!
Scalar sequences acceleration

- Most popular and efficient acceleration method for first order sequences remains the $\Delta^2$ of Aitken (1926):

**Aitken $\Delta^2$-method (static sequence transformation)**

Assuming $x_{n+1} - \bar{x} = (\lambda + \varepsilon_n)(x_n - \bar{x})$, $\lim_{n \to \infty} \varepsilon_n = 0$

Then $y_n = x_n - \frac{(\Delta x_n)^2}{\Delta^2 x_n}$ with

$$
\begin{align*}
\Delta x_n &= x_{n+1} - x_n \\
\Delta^2 x_n &= \Delta x_{n+1} - \Delta x_n
\end{align*}
$$

- Dynamic (/cycling/recursive) sequence transformations based on the $\Delta^2$-method
  - **Secant method**
    $$
    x_{n+1} = x_n - \frac{(x_n - x_{n-1})(f(x_n) - x_n)}{(f(x_n) - x_n) - (f(x_{n-1}) + x_{n-1})}
    $$
    - also called dynamic relaxation, Aitken relaxation,...
    - one-step method
    - order : $(1 + \sqrt{5})/2 \simeq 1.6$
    - efficacy index $1.6 >$ Steffensen method
Most popular and efficient acceleration method for first order sequences remains the $\Delta^2$ of Aitken (1926):

**Aitken $\Delta^2$-method (static sequence transformation)**

Assuming $x_{n+1} - \bar{x} = (\lambda + \varepsilon_n)(x_n - \bar{x})$, \[ \lim_{n \to \infty} \varepsilon_n = 0 \]

Then $y_n = x_n - \frac{(\Delta x_n)^2}{\Delta^2 x_n}$ with \[
\begin{align*}
\Delta x_n &= x_{n+1} - x_n \\
\Delta^2 x_n &= \Delta x_{n+1} - \Delta x_n
\end{align*}
\]

Dynamic (/cycling/recursive) sequence transformations based on the $\Delta^2$-method

- Many other generalizations of the Aitken’s process: $\varepsilon$-algorithm,...
Correspond to the majority of the problems of interest

Generalization of scalar transformations to vector transformations are non-unique (convergence properties ?...)

Extensively studied in the literature in the 60s and since the 90s.
Vector sequences acceleration

- Correspond to the majority of the problems of interest
- Generalization of scalar transformations to vector transformations are non-unique (convergence properties ?...)
- Extensively studied in the literature in the 60s and since the 90s.
  - Most of them are scalar extension using
    \[ X^{-1} = \frac{X}{\|X\|^2} \]
    or more generally
    \[ X^{-1} = \frac{Y}{Y \cdot X} \]

- Steffensen → Irons and Tuck, Lemaréchal, Graves-Morris, A-algorithm, Jennings, Zienkiewicz and Lohner,...
- Secant → many variant of the vector secant method, dynamic relaxation method,...
- \(\varepsilon\)-algorithm → vector \(\varepsilon\)-algorithm, topological \(\varepsilon\)-algorithm
Vector sequences acceleration

- Correspond to the majority of the problems of interest
- Generalization of scalar transformations to vector transformations are non-unique (convergence properties ?...)
- Extensively studied in the literature in the 60s and since the 90s.
  - Other ones are directly dedicated to vector sequences and often based on a minimization process (or projection)
    - Nonlinear hybrid procedure and the associated $\Delta^k$-method
    - Anderson method (equivalent to the interface quasi-Newton method),
    - Reduced rank extrapolation method, Minimal Polynomial extrapolation method,...

These methods can generally reduce (for special choices of parameters) in the scalar case to the Aitken $\Delta^2$ method.

$\leftrightarrow$ Generalization of Aitken’s method for vector sequences.
Correspond to the majority of the problems of interest

Generalization of scalar transformations to vector transformations are non-unique (convergence properties ?...)

Extensively studied in the literature in the 60s and since the 90s.

We propose a common framework to build new vector acceleration methods and to recover the most popular and efficient extrapolation methods
A generic residual-based acceleration method
**Principle of the static method**

- **A generic formalism** for vector sequences

**Residual-based acceleration method**

Assuming

- \( Y_n = X_n - \sum_{i=1}^{M} \lambda_i^n Z_i^n \), \( \lim_{n \to \infty} Z_i^n = 0, \lambda_i^n \in \mathbb{R} \)
- \( Y_{n+1} = X_{n+1} - \sum_{i=1}^{M} \lambda_i^n Z_{i,n+1} \)

Then the minimization of \( \delta Y_n = Y_{n+1} - Y_n \) gives \( \lambda_i^n \) and an estimation of \( \bar{X} \)

\[
Y_n = X_n - Z_n (\delta Z_n^T \delta Z_n)^{-1} \delta Z_n^T \delta X_n
\]

- Generalization for \( M > 1 \) of the nonlinear hybrid procedure
- Very close to the polynomial extrapolation formalism: reduced rank extrapolation method recovered for \( Z_i^n = \delta X_{n+i-1} \)
- For \( Z_i^n = \delta X_{n-i} \): basic idea of Anderson method
Two classes of dynamic methods

- **Iterative acceleration for fixed point iterations:**
  - Two sequences $(X_n)_n$ and $(F(X_n))_n$ are generated
  - Two main formalisms of iterative residual-based methods are available

- **One-step iterative methods** (easily transportable to multi-step methods)
Two classes of dynamic methods

- Iterative acceleration for fixed point iterations:
  - two sequences \((X_n)_n\) and \((F(X_n))_n\) are generated
  - two main formalisms of iterative residual-based methods are available

- One-step iterative methods (easily transportable to multi-step methods)

Crossed sequences method

\[
Y_n = F(X_{n-1}) - \sum_{i=1}^{M} \lambda_i^n Z_i^n
\]

\[
Y_{n+1} = F(X_n) - \sum_{i=1}^{M} \lambda_i^n Z_i^{n+1}
\]

- Focus on the basic fixed point sequence \((F(X_n))_n\)
- Take generally into account \((X_n)_n\) in the definition of \(Z_i^n\)
- For \(Z_i^n = F(X_{n-i}) - X_{n-i}\) and \(M = 1\) : standard vector secant (dynamic relaxation) and Irons-Tuck if two-step method.
Two classes of dynamic methods

- **Iterative acceleration for fixed point iterations**:
  - two sequences \((X_n)_n\) and \((F(X_n))_n\) are generated
  - two main formalisms of iterative residual-based methods are available

- **One-step iterative methods** (easily transportable to multi-step methods)

### Alternate sequences method

\[
Y_n = X_n - \sum_{i=1}^{M} \lambda_i^n Z_i^n
\]

\[
Y_{n+1} = F(X_n) - \sum_{i=1}^{M} \lambda_i^n Z_i^{n+1}
\]

- \(Y_n\) is only concerned by \((X_n)_n\) whereas \(Y_{n+1}\) depends on \((F(X_n))_n\)
- Then \(\delta Y_n\) is a linear combination of fixed point residual \((F(X_n) - X_n)\) with the same coefficients than the sequence coefficients.
- For \(Z_i^n = X_{n-i+1} - X_{n-i}\) and \(Z_i^{n+1} = F(X_{n-i+1}) - F(X_{n-i})\) : Anderson method.
Numerical experiments
Nonlinear mechanics test case

- Point-wise solver used to perform nonlinear mechanical behaviour unit testings: MTest (MFront) tool, developed in the PLEIADES platform.
- Solved by a quasi-Newton fixed point equation

Equilibrium resolution

Let \( X = E|_{t+\Delta t} - E|_t \), then solve
\[
R(X) = \Sigma|_{t+\Delta t}(X, \Delta t, V|_t) - \Sigma|_{t+\Delta t}^{imp} = 0
\]
using
\[
X_{n+1} = X_n - H^{-1}R(X_n)
\]
with \( H \): elastic operator

😊: \( H \) easy to compute, symmetrical and definite positive and can be factorized only once (or updated from time to time)
😊: Very slow fixed point linear convergence.
Fixed point iterations converge very slowly ($\approx 1.000$ iterations per time-step)
Chaboche plastic behaviour

- Fixed point iterations converge very slowly ($\approx 1.000$ iterations per time-step)
- Two iteration residual methods comparison + second-order Newton method - $\varepsilon = 10^{-8}$
Chaboche plastic behaviour

- Fixed point iterations converge very slowly ($\sim 1.000$ iterations per time-step)
- Two iteration residual methods comparison + second-order Newton method - $\varepsilon = 10^{-8}$
Fixed point iterations converge very slowly (≈ 1.000 iterations per time-step)

Three iteration residual methods are not better than two iteration methods: Alternate 2 – $\delta \simeq$ Alternate secant.
Fixed point iterations converge very slowly ($\approx 1.000$ iterations per time-step)

Three iteration residual methods are not better than two iteration methods: Alternate 2 $-\delta \approx$ Alternate secant. Crossed 2-$\delta$ method converges but not to the fixed point.
Fixed point iterations converge faster

Jacobian matrix unavailable $\Rightarrow$ no Newton method
- Fixed point iterations converge faster
- Jacobian matrix unavailable \(\Rightarrow\) no Newton method
- Two iteration residual methods comparison - \(\varepsilon = 10^{-8}\)
Polycrystals test case

- Fixed point iterations converge faster
- Jacobian matrix unavailable $\Rightarrow$ no Newton method
- Two iteration residual methods comparison - $\varepsilon = 10^{-8}$
Polycrystals test case

- Fixed point iterations converge faster
- Jacobian matrix unavailable \( \Rightarrow \) no Newton method
- Three iteration residual methods comparison
Polycrystals test case

- Fixed point iterations converge faster
- Jacobian matrix unavailable ⇒ no Newton method
- Three iteration residual methods comparison

![Graph comparing residual vs. number of iterations for different methods]

Legend:
- Fixed Point
- Crossed Secant
- Alternate Secant
- Alternate Delta2
- Crossed 2Delta
- Alternate 2Delta

Residual vs. Number of iterations plot showing convergence rates for different methods.
Performances of the acceleration method depend on the test case.

Secant methods seem really efficient: they can concurrence the second-order Newton method!

Alternate $M - \delta$ (\equiv Anderson) methods seem the best $M$-iterates methods.
Multi-physics coupling

- Neutronic
- Thermic
- Mechanics
- Fission products

Time evolution

- Mult-physics loop
- Gauss-Seidel multi-physics fixed point iterations

Partitioning coupling

platform dedicated to nuclear fuel behaviour simulation
Crossed Secant

Fixed point iterations

Crossed secant (dynamic relaxation)
Irons and Tuck method

- Crossed secant (dynamic relaxation) not always converge in case of fixed point iterations divergence
- Irons and Tuck method seems always converge but sometimes slower than the fixed point.
Irons and Tuck method

- Crossed secant (dynamic relaxation) not always converge in case of fixed point iterations divergence
- Irons and Tuck method seems always converge but sometimes slower than the fixed point.
Irons and Tuck method

- Crossed secant (dynamic relaxation) not always converge in case of fixed point iterations divergence
- Irons and Tuck method seems always converge but sometimes slower than the fixed point.
- Currently trying alternate $M - \delta$ methods...
Conclusions
A generic residual-based approach has been introduced to build sequence acceleration processes

- generalization of various existing vector acceleration methods
- derived in two main classes of dynamic acceleration methods: crossed and alternate approaches

A set of iterative acceleration approaches have been tested on nonlinear mechanics resolution and multi-physics simulation

- the crossed approach is only interesting in its secant \((1 - \delta X_n)\) version
- in particular alternate \(M - \delta\) (Anderson) approaches seem the most efficient

This generic acceleration formalism has already been successfully applied on other applications: FFT, PGD,...
Thank you for your attention