# *nlTGCR: A nonlinear acceleration procedure based on Generalized Conjugate Residuals*

*Yousef Saad*
University of Minnesota

NASCA 2023

$A\theta\acute{\eta}\nu\alpha$, Jul. 3-6, 2023

# *First:*

- Joint work with: Yuanzhe Xi (Emory), Shifan Zhao, Huan He (Harvard), Ziyuan Tang (Minnesota)
- Work supported by NSF.
- Related articles:
  - *NLTGCR: a class of nonlinear acceleration procedures based on Conjugate Residuals,* Huan He, Ziyuan Tang, Shifan Zhao, YS, and Yuanzhe Xi
  - *Shanks sequence transformations and Anderson acceleration,* C. Brezinski, M. Redivo-Zaglia, YS - SIAM Review, 2018

## *Introduction & Background*

- Accelerators for linear systems: Conjugate Gradient, Conjugate Residual, GCR, ORTHOMIN, GMRES, BiCGSTAB, IDR, ..
- Krylov subspace methods
- Picture for solving nonlinear equations is more complex

*(a)* Linear accelerators invoked when solving Jacobian systems iteratively in Newton $\rightarrow$ Inexact Newton methods

*(b)* Quasi-Newton methods, BFGS, LBFGS, ..., : approximate Jacobian/ inverse with Low-rank updates

*(c)* Anderson acceleration, Pulay mixing, ... nonlinear acceleration viewpoint + (rough) a linear model

- This talk: take the viewpoint of extending nonsymmetric Krylov methods [GCR, ORTHOMIN, ..] to nonlinear setting
- Many many possible options and viewpoints
- Can exploit models that are locally more accurate; can exploit known results on global convergence; etc.
- Possible to derive methods that emcompass all three viewpoints (a), (b), (c) shown above.
- One specific goal: unravel algorithms with short-term recurrence

... Let us begin with some background

## *Extrapolation and Acceleration: A few historal landmarks*

Extraplotion: given sequence $(s_j)$
- define extrapolated sequence:

$$\boxed{t_k^{(j)} = \sum_{i=0}^{k} \alpha_i s_{j+i} \quad \text{with} \quad \sum \alpha_i = 1}$$

- ➤ Richardson's 'deferred approach to the limit' 1910, 1927.
- ➤ Aitken [1926] – initially to compute zeros of polynomials.
- ➤ Romberg [1955] – integration, ...
- ➤ Shanks [1955] generalizes Aitken's method
- ➤ Wynn [1956]: Elegant implementation of Shanks transform $\rightarrow$ $\epsilon$-algorithm
- ➤ Discovery ignited substantial following in late 1960s - early 1970s
- ➤ C. Brezinski, H. Sadok, K. Jbilou, M. Redivo Zaglia, Germain-Bonne, G. Walz, A. Sidi and co-workers, ...

- In physics: Different approaches - e.g., Anderson mixing, DIIS, ..., were developed - with a similar goal

- Viewpoint closer to quasi-Newton than to extrapolation

- In Numerical Linear Algebra: Acceleration for linear systems : Chebyshev acceleration (old), but also Minimal Polynomial Extrapolation (MPE- Cabay-Jackson); Reduced Rank Extrapolation, many others

# *Acceleration*

➤ Common situation: A (complex) physical simulation leading to a sequence of a physical quantity (charge densities, potentials, pressures, ...)

➤ Common approach: fixed point iteration

$$x_{k+1} = g(x_k)$$

- Acceleration methods try to solve the system $x - g(x) = 0$ by creating a sequence that invokes function $g$ and the previous iterates.
- In essence we seek to solve $f(x) = 0$ where $\boxed{f(x) \equiv x - g(x)}$
- With one restriction: use only function evaluations and lin. combinations

# *Acceleration, Extrapolation, Quasi-Newton*

Extrapolation

$x_1, x_2, \cdots, x_n \to$
$t_k^{(n)}, n = 1, 2, ..$
Shanks formula,
$\epsilon$-Algorithm, ...

Quasi–Newton:

$(f(x) = 0\ )$
$x \leftarrow x - M^{-1} f(x)$
$M$ approximates
Jacobian using
$\Delta x_1, \Delta x_2, \cdots, \Delta x_j$
$\Delta f_1, \Delta f_2, \cdots, \Delta f_j$

Anderson-Pulay

$(f(x) = 0)$
$\sim Min||f(x + \Delta X y)||$
Approximate
$f(x + \Delta X y)$ using
$\Delta x_1, \Delta x_2, \cdots, \Delta x_j$
$\Delta f_1, \Delta f_2, \cdots, \Delta f_j$

# Inexact Newton, Quasi-Newton, Krylov-Newton

We now focus on solving $f(x) = 0$ ($f : \mathbb{R}^n \to \mathbb{R}^n$) Newton Approach

Set $x_0 =$ an initial guess.
For $n = 0, 1, 2, \cdots$ until conv. do:
  Solve: $J(x_j)\delta_j = -f(x_j)$   (*)
  Set: $x_{j+1} = x_j + \delta_j$

$\leftarrow f(x_j + \delta) \approx f(x_j) + J(x_j)\delta$

with $J(x_j) = f'(x_j)$ = Jacobian at $x_j$

Standard Newton: solve (*) exactly

Inexact Newton methods: solve system (*) approximately.

Quasi-Newton methods: solve system (*) in which Jacobian is replaced by an estimate obtained from previous iterates.

Newton-Krylov methods: solve system (*) by a Krylov subspace method

Note: In Krylov-Newton, Jacobian of $f$ not needed explicitly.

- Compute $Jv$ via finite difference approximation: $\frac{\partial f}{\partial x}v \approx \frac{f(x+\epsilon v)-f(x)}{\epsilon}$

- Can use Newton-Krylov to accelerate sequence: $\boxed{x_{j+1} = g(x_j)}$

.. by solving $f(x) = 0$ where $f(x) = x - g(x)$

*Important consideration:* need to compute $f(x_j + \epsilon v)$ for arbitrary $v$ ..

- ... instead of using only the $x_j$'s and $f_j$'s that are available

# *Inexact Newton, Quasi-Newton, Anderson Acceleration*

Problem: Find $x \in \mathbb{R}^n$ such that $f(x) = 0$

Or solve: $\min \phi(x)$; Then $f(x) = \nabla\phi(x)$

Recall: *Newton Krylov:* $x_{j+1} = x_j + \delta_j$ where

$\delta_j \equiv$ approx. solution of $\boxed{J(x_j)\delta + f(x_j) = 0}$ by a Krylov subspace method

- Notation $J \equiv J(x_j)$ - So Newton system is

$$J\delta = -f(x_j)$$

- Let $V_l$ is an orthonormal basis of the Krylov subspace

$$K_l = \mathbf{span}\{v, Jv, \cdots, J^{l-1}v\}, \quad \text{where} \quad v \equiv -f(x_j)$$

- Then approximate solution is in the form $\boxed{\delta_j = V_l y_l}$

- For example, if the method invoked is FOM, then:

$$\boxed{\delta_j = V_l(V_l^T J V_l)^{-1} V_l^T(-f(x_j))}$$

- In essence: inverse Jacobian approximated by the matrix

$$\boxed{B_{j,IOM} = V_l(V_l^T J V_l)^{-1} V_l^T}$$

- For GMRES / GCR, inverse Jacobian approximation is:

$$\boxed{B_{j,GMRES} = V_l(JV_l)^{\dagger}.}$$

*Important observation:* approximations are for step $j$ only – discarded in next step. The process has no ‘memory’

# *Inexact Newton, Quasi-Newton, Anderson Acceleration*

- Quasi-Newton (QN) methods: build approximations to $J(x_j)$ or $J(x_j)^{-1}$, *progressively* using previous iterates

- Notation: $\Delta x_j \equiv x_{j+1} - x_j$ , $\Delta f_j \equiv f(x_{j+1}) - f(x_j)$,

- Secant condition:

- *No-change condition:*

$$J_{j+1}\Delta x_j = \Delta f_j,$$

$$J_{j+1} q = J_j q, \quad \forall q \quad \text{such that} \quad q^T \Delta x_j = 0.$$

- Broyden: $\exists!$ $J_{j+1}$ that satisfies both conditions. Calculated as:

$$J_{j+1} = J_j + (\Delta f_j - J_j \Delta x_j) \frac{\Delta x_j^T}{\Delta x_j^T \Delta x_j}.$$

- Type II Broyden: Inverse Jacobian approximated by $G_j$ at step $j$
- Secant condition:
- *No-change condition:*

$$G_{j+1}\Delta f_j = \Delta x_j,$$

$$G_{j+1}q = G_j q, \quad \forall q \quad \text{such that} \quad q^T \Delta f_j = 0.$$

- Broyden (II): $\exists!$ $G_{j+1}$ that satisfies both conditions. Calculated as:

$$G_{j+1} = G_j + (\Delta x_j - G_j \Delta f_j)\frac{\Delta f_j^T}{\Delta f_j^T \Delta f_j},$$

Note: Common feature of QN methods: The sequence of pairs of $\Delta x_i, \Delta f_i$ used to update previous approximation to $J(x_j)$ or $J(x_j)^{-1}$.

- Progressive low-rank approximation ...
- ... 'One rank at a time'

## *Anderson Acceleration*

- Want fixed point of $g(x) : \mathbb{R}^n \to \mathbb{R}^n$. Let $f(x) = g(x) - x$.
- Select $x_0$ and define $x_1 = x_0 + \beta f_0$ [$\beta$ is a parameter]

Given: $x_i$ and $f_i = f(x_i)$ for $i = j - m, \cdots, j$

Let: $\Delta x_i = x_{i+1} - x_i, \quad \Delta f_i = f_{i+1} - f_i$ for $i = 0, 1, \cdots, j - m$

$\mathcal{X}_j = [\Delta x_{j-m} \ \cdots \ \Delta x_{j-1}], \quad \mathcal{F}_j = [\Delta f_{j-m} \ \cdots \ \Delta f_{j-1}]$.

Compute: $\boxed{x_{j+1} = \bar{x}_j + \beta \bar{f}_j}$ where: $\boxed{\bar{x}_j = x_j - \mathcal{X}_j \, \theta^{(j)}, \ \bar{f}_j = f_j - \mathcal{F}_j \, \theta^{(j)}}$

And: $\boxed{\theta^{(j)} = \mathrm{argmin}_{\theta \in \mathbb{R}^m} \| f_j - \mathcal{F}_j \, \theta \|_2}$

Note: Original article formulated problem in the standard 'acceleration' form

$$\bar{x}_j = \sum_{i=j-k}^{j} \mu_i^{(j)} x_i \quad \text{with} \quad \sum \mu_i^{(j)} = 1$$

- The $\mu_i^{(j)}$'s must now minimize $\left\| \sum_{i=j-k}^{j} \mu_i^{(j)} f_i \right\|_2^2$
- Mathematically equivalent to previous formulation

*Q* Any relation to extrapolation?

- Above formulation is very similar to expressions used for extrapolation.
- Anderson was very much inspired by litterature in extrapolation methods.

## *Relation with other methods*

- In "generalized Broyden methods" [Louis & Vanderbilt'84, Eyert'96] approximate Jacobian $G_j$ satisfies $m$ secant conditions at once:

$$G_j \Delta f_i = \Delta x_i \text{ for } i = j-m, \ldots, j-1.$$

- Matrix form:

$$G_j \mathcal{F}_j = \mathcal{X}_j$$

- No-change condition:

$$(G_j - G_{j-m})q = 0 \quad \forall q \in \text{Span}\{\Delta f_{j-m}, \ldots, \Delta f_{j-1}\}^{\perp}$$

- After calculations we get a rank-$k$ update formula:

$$G_j = G_{j-m} + (\mathcal{X}_j - G_{j-m}\mathcal{F}_j)(\mathcal{F}_j^T \mathcal{F}_j)^{-1}\mathcal{F}_j^T.$$

... and an update of the form:

$$\boxed{x_{j+1} = x_j - G_{j-m} f_j - (\mathcal{X}_j - G_{j-m}\mathcal{F}_j)\gamma_j; \quad \gamma_j = \mathcal{F}_j^{\dagger} f_j}$$

- Setting $G_{j-m} = -\beta I$ yields exactly Anderson's original method [which includes a parameter $\beta$]
- Result shown by Eyert (1996) [See also H-r Fang and YS (2009)]
- Note $\bar{x}_j = x_j - \mathcal{X}_j \mathcal{F}_j^{\dagger} f_j$ and $\bar{f}_j = f_j - \mathcal{F}_j \mathcal{F}_j^{\dagger} f_j$
- Walker and Ni'11: Equivalence with GMRES in linear case.

# NONLINEAR TRUNCATED GCR

## *Revisiting old friends: The GCR method*

*Recall main goal:* start with accelerators in linear case - then see how to extend them to nonlinear case

*Class of Krylov subspace methods:*

- Conjugate gradient (Hestenes and Stiefel, '51), Conjugate Residual (Stiefel '55), Lanczos (51), Bi-CG (Fletcher 76)
- Accelerators developed in 1980s, 1990s: GCR, ORTHOMIN, GMRES, BiCGSTAB, IDR, ..

➤ We consider the *Generalized Conjugate Residual* (GCR) [Eisenstat, Elman, Schultz, '83]

# *GCR for linear case:* $Ax = b$

ALGORITHM : 1. *GCR*

**Input**: *Matrix* $A$, *RHS* $b$, *initial* $x_0$.
Set $p_0 = r_0 \equiv b - Ax_0$.
**for** $j = 0, 1, 2, \cdots,$ *Until convergence* **do**
    $\alpha_j = (r_j, Ap_j)/(Ap_j, Ap_j)$
    $x_{j+1} = x_j + \alpha_j p_j$
    $r_{j+1} = r_j - \alpha_j Ap_j$
    $p_{j+1} = r_{j+1} - \sum_{i=0}^{j} \beta_{ij} p_i$   *where*   $\beta_{ij} := (Ar_{j+1}, Ap_i)/(Ap_i, Ap_i)$
**end for**

➤ Recall: the set $\{Ap_i\}_{i=0,\cdots,j}$ is orthogonal

- Two practical variants

| | |
|---|---|
| *Restarting* | GCR(k) - restart every $k$ steps |
| *Truncation* | TGCR(m,k) - Truncated GCR: Orthogonalize against $m$ most recent vectors only + restart dimension of $k$ |

- In TGCR(m,k) Line 7 becomes: [Notation: $j_m = \max\{0, j - m + 1\}$]

$$p_{j+1} = r_{j+1} - \sum_{i=j_m}^{j} \beta_{ij} p_i \quad \text{where} \quad \beta_{ij} := (Ar_{j+1}, Ap_i)/(Ap_i, Ap_i)$$

- GCR(k): Eisenstat, Elman and Schultz [83] - equivalent to GMRES(k)

- TGCR initially developed by Vinsome '76 (as *ORTHOMIN*), analyzed in 1983 GCR paper

# Properties of (full) GCR in linear case

Notation: $\boxed{P_k = [p_0, p_1, \cdots, p_k]}$ $\boxed{R_k = [r_0, r_1, \cdots, r_k],}$ $\boxed{V_k = AP_k}$

Property: (Eisenstat-Elman-Schultz) *The residual vectors produced by (full) GCR are semi-conjugate, i.e.,* $(r_j, Ar_i) = 0$ *for* $i < j$.

Corollary: When $A = A^T$ residuals are conjugate

Property: *When* $A$ *is symmetric real, then the matrix* $(AR_k)^T(AP_k)$ *is lower bidiagonal.*

Property: *When* $A$ *is nonsingular, (full) GCR breaks down iff it produces an exact solution.*

breakdown $\leftrightarrow$ 'lucky breakdown'

Property: Approximate solution at $k$-th step is $\boxed{x_{k+1} = x_0 + P_k V_k^T r_0}$

- We say that the algorithm induces the 'approximate inverse' $B_k = P_k V_k^T$ - a rank-k matrix. Let $\boxed{\mathcal{L}_k = \text{Span}(V_k)}$ and $\boxed{\pi = V_k V_k^T}$. Then

> - $B_k = A^{-1}\pi \;\; \rightarrow \;\; B_k$ inverts $A$ exactly in $\mathcal{L}_k$, i.e., $B_k \pi = A^{-1}\pi$.
> - $AB_k = \pi$.
> - When $A$ is symmetric then $B_k$ is self-adjoint when restricted to $\mathcal{L}_k$.
> - $B_k A x = x$ for any $x \in \text{Span}\{P_k\}$, i.e., $B_k$ inverts $A$ exactly from the left when $A$ is restricted to the range of $P_k$.
> - $B_k A$ is the projector onto $\text{Span}\{P_k\}$ and orthogonally to $A^T \mathcal{L}_k$.

- Reminescent of Moore-Penrose properties

# Nonlinear case: Inexact-Newton with GCR

Problem : $f(x) = 0$

Inexact Newton: $\boxed{\begin{array}{l} x_{j+1} = x_j + \delta_j \quad \text{where:} \\ \|J\delta_j + f_j\| \le \eta_j \|f_j\| \end{array}}$ $\quad f_j \equiv f(x_j)$, $\quad J \equiv Df(x_j))$

- Dembo-Eisenstat-Steihaug ’82, Dembo-Steihaug ’83, ...,
- Inexact-Newton GCR : solve systems approximately with TGCR(m,k)
- Inexact Newton is a simple, well-understood framework.
- Lots of results with linesearch + trust-region global strategies.
- Newton-GMRES [Brown & YS, 1990]; Convergence results [Brown & YS, 1994, Eisenstat & Walker ’94]

## *Next: Multisecant viewpoint*

- Linear TGCR builds $m$ directions such that:

  $\{Ap_{j_m}, \cdots Ap_j\}$ is orthogonal

- In nonlinear case we can still use this basis– where $A$ is 'some' Jacobian.

- This is done in inexact Newton where: $\boxed{A = J(x_0)}$ - fixed.

- Here: we assume that at step $j$ we have a set of (at most) $m$ current 'search' directions $\{p_i\}$ for $i = j_m, j_m + 1, \cdots, j$

- Along with $v_i \equiv J(x_i)p_i$, $i = j_m, j_m + 1, \cdots, j$

- Set:

$$P_j = [p_{j_m}, p_{j_m+1}, \cdots, p_j], \qquad V_j = [v_{j_m}, v_{j_m+1}, \cdots, v_j].$$

- Note: In Linear Case or Inexact Newton case $v_i = Jp_i$ ($J$ is fixed)

- Here $J$ varies with iterate - $v_i = J(x_i)p_i$ $(== Ap_i$ in TGCR)
- $p_i$ and $v_i$ are 'paired' much like the $\Delta f_j$ and $\Delta x_i$ of QN and AA
- Notation $\boxed{V_j = [J]P_j}$

## Main Idea of Nonlinear Extension:

- Just build orthonormal basis $V_j$ as in TGCR
- Do usual projection step to minimize 'linear residual' - i.e.,

$$x_{j+1} = x_j + P_j y_j \qquad \text{where} \qquad y_j = \text{argmin}_y \| f(x_j) + V_j y \|$$

- Note: $V_j$ orthonormal $\rightarrow$ $y_j = V_j^T(-f(x_j)) \equiv V_j^T r_j$

## ALGORITHM : 2. *nlTGCR(m,k)*

**Input**: $f(x)$, initial $x_0$.
Set $r_0 = -f(x_0)$.
Compute $v = Jr_0$; ▷ *Use Frechet*
$v_0 = v/\|v\|$, $p_0 = r_0/\|v\|$;
**for** $j = 0, 1, 2, \cdots$, Until convergence **do**
    $y_j = V_j^T r_j$
    $x_{j+1} = x_j + P_j y_j$ ▷ *Scalar $\alpha_j$ becomes vector $y_j$*
    $r_{j+1} = -f(x_{j+1})$ ▷ *Replaces linear update: $r_{j+1} = r_j - V_j y_j$*
    Set: $p := r_{j+1}$; and $i_0 = \max(0, j - m + 1)$
    Compute $v = Jp$ ▷ *Use Frechet*
    Compute $[p_{j+1}, v_{j+1}] = bOrth(P_j, V_j, v, m)$
    If mod(j,k) == 0, restart
**end for**

# A few properties

➤ Notation: $\tilde{r}_{j+1} = r_j - V_j y_j$ (Linear Residual) ; $z_j = \tilde{r}_j - r_j$

The following properties are satisfied by the vectors produced by ***nlTGCR***:

1. The system $[v_{j_m}, v_{j_m+1}, \cdots, v_{j+1}]$ is orthonormal.
2. $(\tilde{r}_{j+1}, v_i) = 0$ for $j_m \le i \le j$, i.e., $V_j^T \tilde{r}_{j+1} = 0$.
3. $\|\tilde{r}_{j+1}\|_2 = \min_y \|f(x_j) + [J] P_j y\|_2 = \min_y \|f(x_j) + V_j y\|_2$
4. $(v_{j+1}, \tilde{r}_{j+1}) = (v_{j+1}, r_j)$
5. $V_j^T r_j = (v_j, \tilde{r}_j) e_1 - V_j^T z_j$ where $e_1 = [1, 0, \cdots, 0]^T \in \mathbb{R}^{m_j}$ with $m_j \equiv \min\{m, j+1\}$.

➤ What can we say about the deviation $z_j$?

# A few properties (cont.)

Define: $\boxed{\begin{array}{l} s_j = f(x_{j+1}) - f(x_j) - J(x_j)(x_{j+1} - x_j). \\ w_i = (J(x_j) - J(x_i))p_i \; ; \quad \text{and} \quad W_j = [w_{j_m}, \cdots, w_j]. \end{array}}$

The difference $z_{j+1} = \tilde{r}_{j+1} - r_{j+1}$ satifies the relation:

$$\tilde{r}_{j+1} - r_{j+1} = W_j y_j + s_j = W_j V_j^T r_j + s_j$$

and therefore:

$$\|\tilde{r}_{j+1} - r_{j+1}\| \le \|W_j\|_2 \, \|r_j\|_2 + \|s_j\|_2$$

- All this means is that the difference is of “second order”
- Hence: can switch to linear form of residual at some point
- Saves one fun. eval

- Let $d_j = x_{j+1} - x_j = P_j y_j$. One may ask: Is this a descent direction?

Let $f(x) = \frac{1}{2}\|f(x)\|_2^2$ and let $\tilde{v}_{j_m}, \cdots, \tilde{v}_j$ be the columns of:

$$\tilde{V}_j \equiv J(x_j)P_j.$$

Then,

$$(\nabla f(x_j), d_j) = -(v_j, r_j)^2 - \sum_{i=j_m}^{j-1} (v_i, r_j)(\tilde{v}_i, r_j)$$

## *Multisecant property*

- Observe that the update at step $j$ takes the form:

$$\boldsymbol{x}_{j+1} = \boldsymbol{x}_j + \boldsymbol{P}_j \boldsymbol{V}_j^T \boldsymbol{r}_j = \boldsymbol{x}_j + \boldsymbol{P}_j \boldsymbol{V}_j^T (-\boldsymbol{f}(\boldsymbol{x}_j))$$

- Thus, we are in effect using a secant-type method with the Approximate inverse Jacobien:

$$\boldsymbol{G}_{j+1} = \boldsymbol{P}_j \boldsymbol{V}_j^T$$

- In addition: The unique solution to the problem

$$\min\{\|\boldsymbol{B}\|_F \text{ subject to: } \quad \boldsymbol{B}\boldsymbol{V}_j = \boldsymbol{P}_j\}$$

is achieved by the matrix $\boldsymbol{G}_{j+1} = \boldsymbol{P}_j \boldsymbol{V}_j^T$.

- Yet another multi-secant type method, but ...

- The method shares also characteristics of inexact Newton
- In particular: possible to add global convergence strategies – e.g. back-tracking [unlike AA]
- The relation $\boldsymbol{v}_j = \boldsymbol{J}(\boldsymbol{x}_j)\boldsymbol{p}_j$ is accurate - [Frechet diff.]
- Contrast with the relation $\boldsymbol{\Delta f}_j \approx \boldsymbol{J}\boldsymbol{\Delta x}_j$ (Anderson, QN)
- Two function evaluations per iteration but ...
- ... can be reduced to one as soon as $\boldsymbol{r}_j$ becomes close to $\tilde{\boldsymbol{r}}_j$ (linear)

## *General GCR framework*

- There are situations where Anderson does amazingly well..

- Example Picard iteration for Navier Stokes. [A form of Preconditioned fixed-pt iter.]

*Q:* Can we implement Anderson acceleration in the form of GCR? The two are fairly close

*A:* Yes -

- Details skipped -

## *Experiments - Bratu problem*

- Illustrates the importance of exploiting symmetry [Recall: in linear symmetric case GCR becomes CR, requires window-size of 2]

- .. and importance of adaptive version

Nonlinear eigenvalue problem (Bratu)

- Take $\lambda = 0.5$.

$$-\Delta u = \lambda e^u \text{ in } \Omega = (0,1) \times (0,1)$$
$$u(x,y) = 0, \text{ for } (x,y) \in \partial\Omega$$

- FD discretization with grid of size $100 \times 100$ $\to$r Problem size = $n = 10,000$

- Tested: *nlTGCR*, *anderson*, and a basic *adaptive gradient method* (step-length dynamically adapted)

## The Adaptive update version

- Bratu problem is almost linear – also true for all problems near convergence

- Idea: exploit the linearized update version of nlTGCR to cut number of func. evals. by $\approx$ half

- Need an adaptive mechanism: switch from the nonlinear to linear updates - [$\approx$ linear regime]

- and switch back when needed

- Define the nonlinear and nonlinear res. at step $j$:

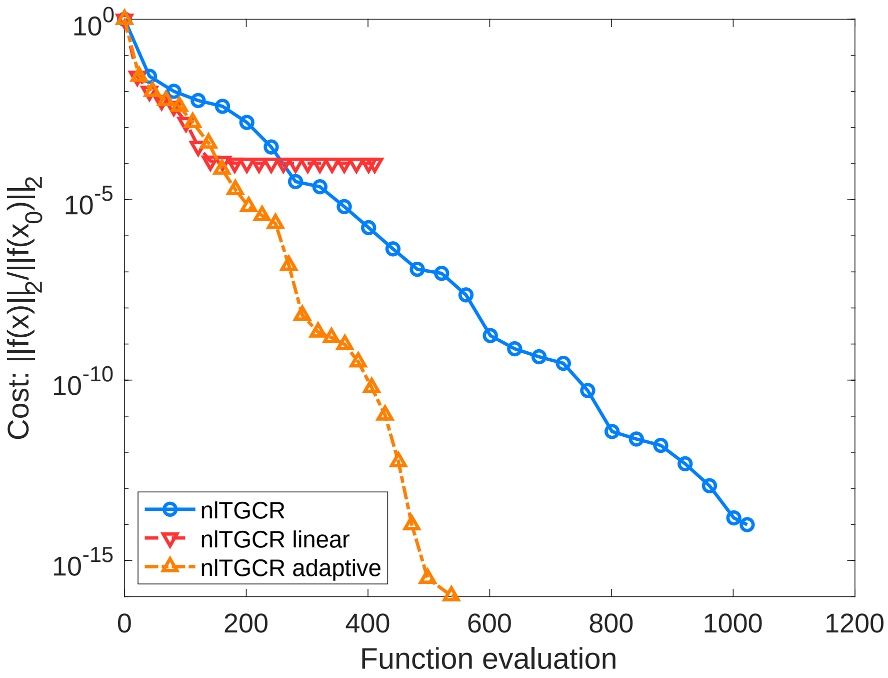
$$\boxed{\begin{aligned} r_{j+1}^{nl} &= -f(x_{j+1}), \\ r_{j+1}^{lin} &= r_j^{nl} - V_j y_j. \end{aligned}}$$

- Criterion will use the angular distance between the two vectors:

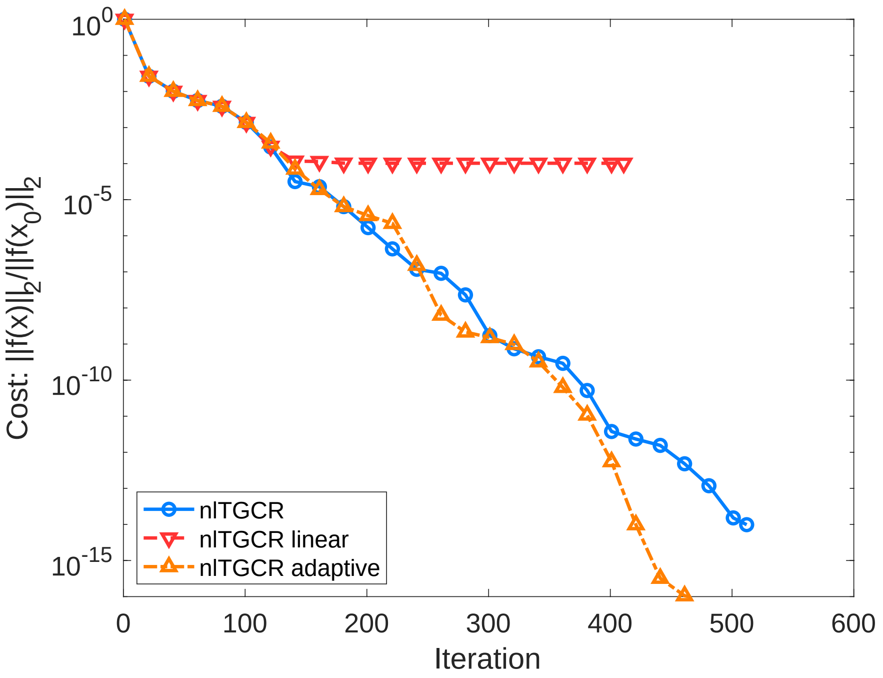
$$d_j := 1 - \frac{(r_j^{nl})^T r_j^{lin}}{\|r_j^{nl}\|_2 \cdot \|r_j^{lin}\|_2}$$

- Linear updates turned on when $d_j < \tau$, where $\tau$ is a threshold
- Check $d_j$ regurlarly, for example, every 10 iterations,
- Switch back to nonlinear updates when $d_j \geq \tau$
- In experiments, we set the threshold to $\tau = 0.01$.

- Window size $m = 1$,



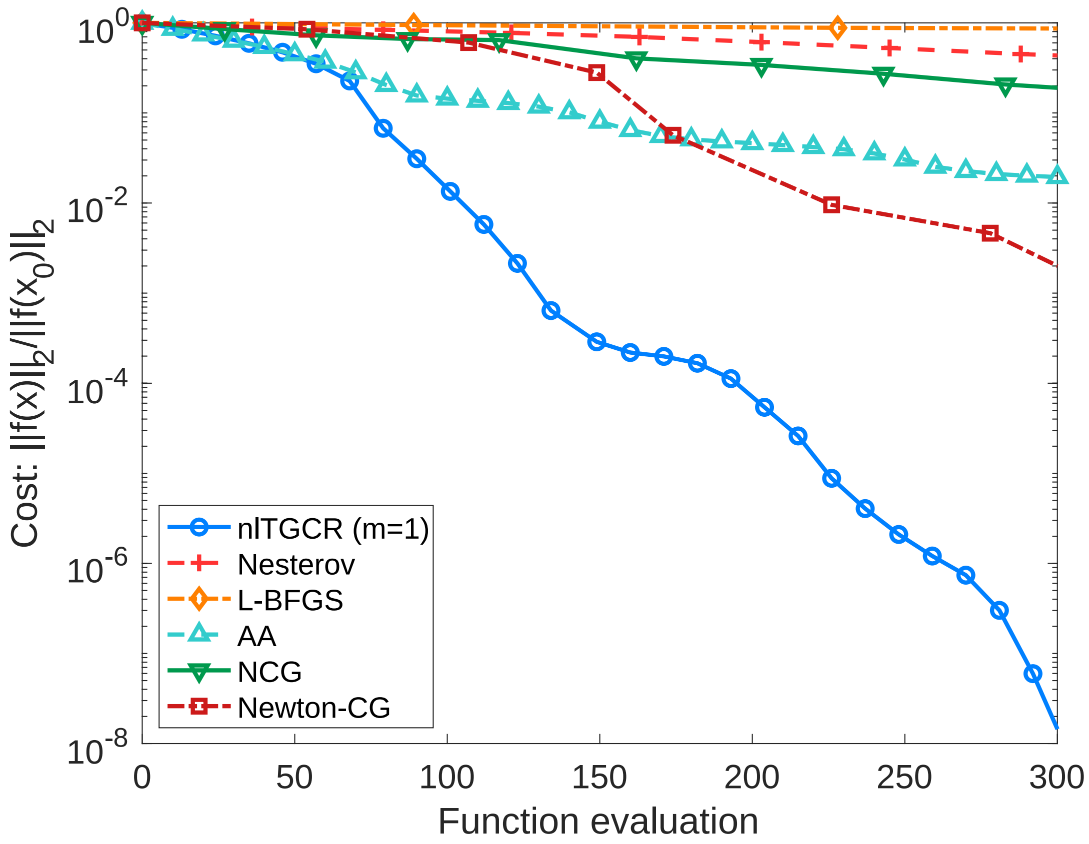

*Function evaluations.*




*Iterations*

# *Exploiting symmetry*

Bratu problem with:
AA, L-BFGS, Nonlinear CG (NCG), [fletcher reeves], and Inexact Newton with CG (Newton-CG).



## *Molecular optimization with Lennard-Jones potential*$^{(*)}$

- Illustrates the importance of a global strategy - linesearch / backtracking + exploiting the Jacobian at multiple points

- Goal: find atom positions that minimize total potential enery:

Lennard-Jones Potential ($x_i$ = position of atom $i$)

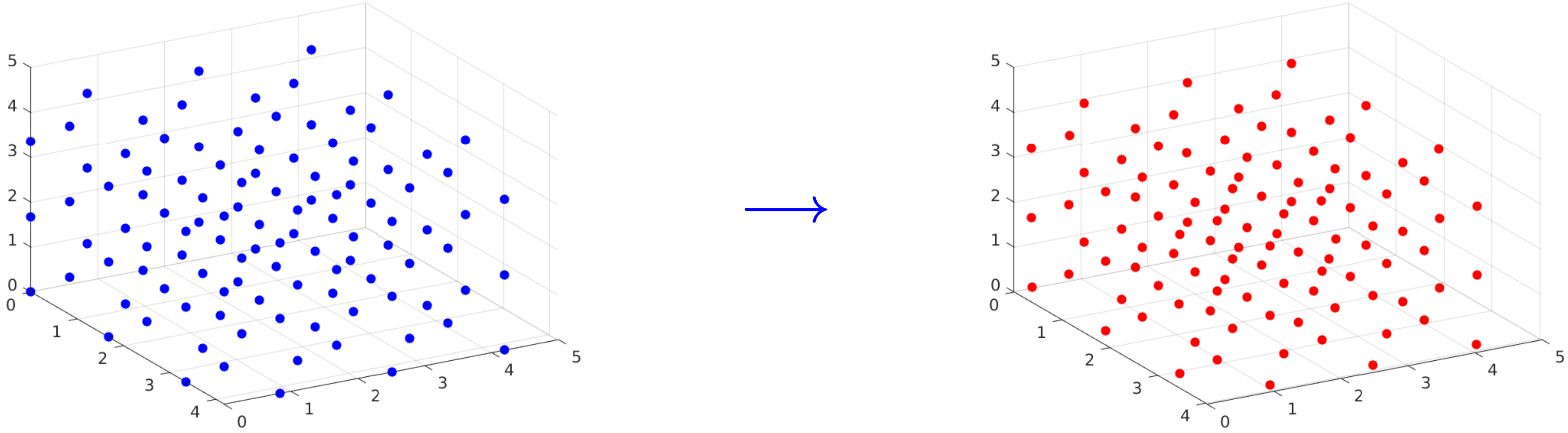
$$E = \sum_{i=1}^{Nat} \sum_{j=1}^{i-1} 4 \times \left[ \frac{1}{\|x_i - x_j\|^{12}} - \frac{1}{\|x_i - x_j\|^{6}} \right]$$

$\boxed{\text{Initial Config}} \rightarrow \boxed{\text{Iterate to mininmize } \|\nabla E\|^2} \rightarrow \boxed{\text{Final Config}}$

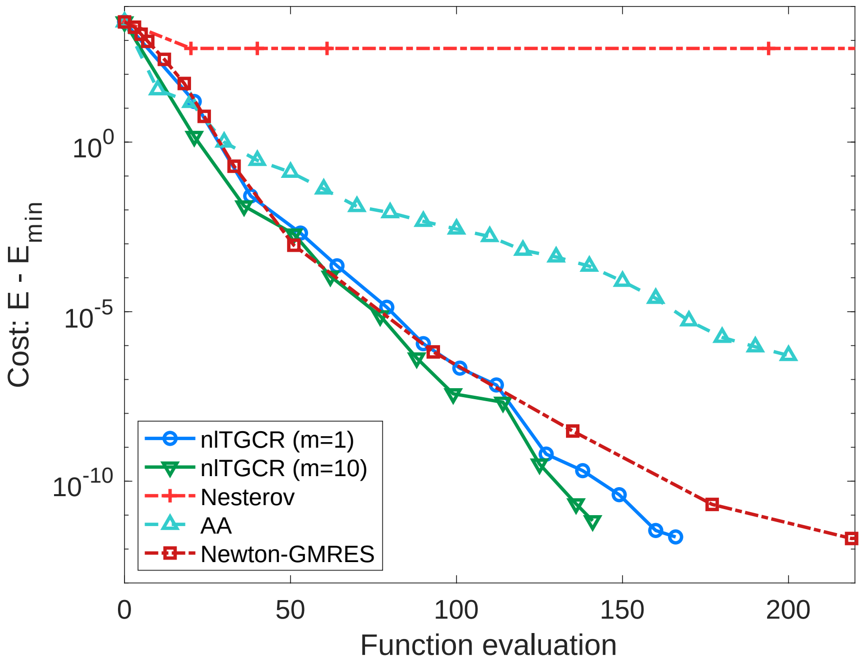
- Difficult problem due to high powers $\rightarrow$ Backtracking essential

---

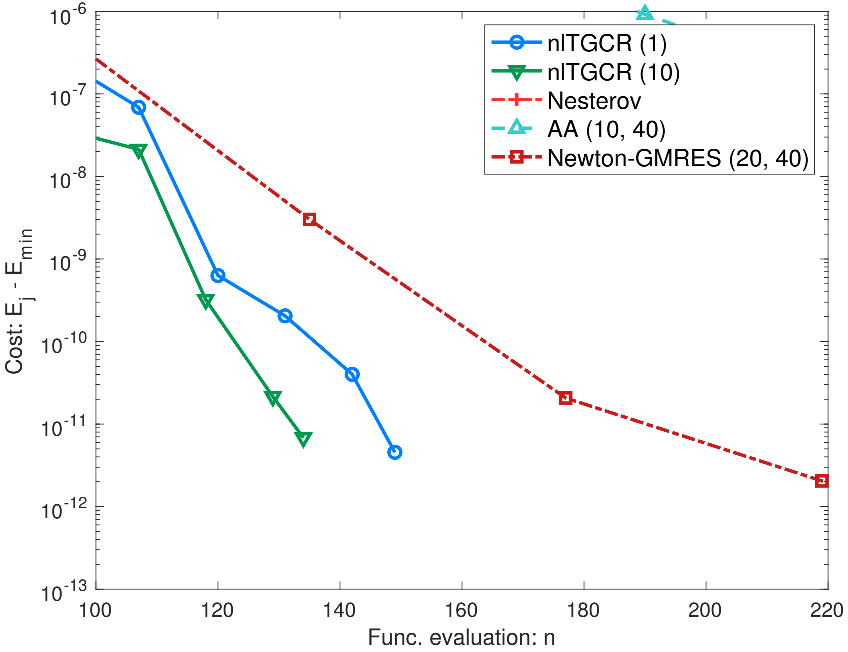
(*) *Thanks:* Stefan Goedecker's course site - Basel Univ.




- Initial geometry: 'Face-Centered Cube' + perturbation
- Adaptive gradient method: $\boxed{x_{j+1} = x_j - t_j \nabla E(x_j)}$ – with $t_j$ adapted – can be made to work fairly well.
- AA will fail unless underlying fixed point iteration selected carefully: $\boxed{x_{j+1} = x_j - \mu \nabla E(x_j)}$ where $\mu \sim 10^{-3}$. Also must take $\beta \sim 10^{-2}$.



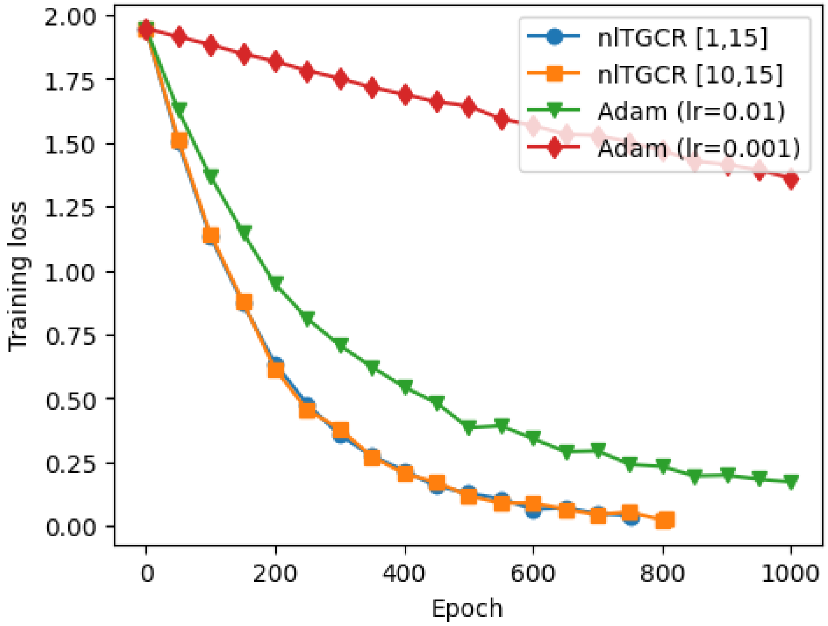

*Lennard-Jones problem. )*

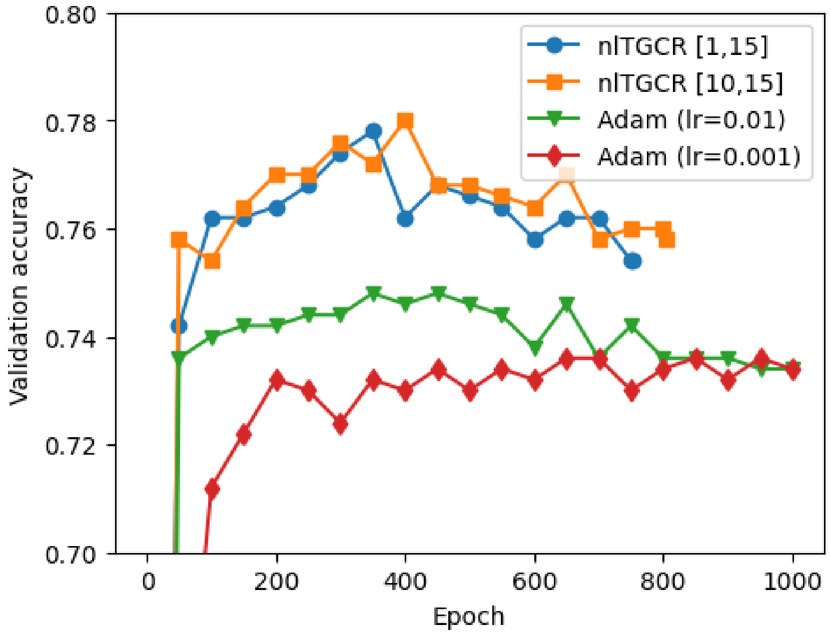



*Zoom near convergence*

# *Graph Convolutional Network*

Dataset: ***Cora*** [2708 scientific pubs., 5429 links, 7 classes]. Goal: node classification [topic of paper from words and links]




*nlTGCR vs. Adam: training loss and validation accuracy*

# A few references


- Eyert, V. (1996). *A comparative study on methods for convergence acceleration of iterative vector sequences.* *J. Computational Phys.*, 124:271–285.
- H. ren Fang and Y. Saad (2009). *Two classes of multisecant methods for nonlinear acceleration.* *Numerical Linear Algebra with Applications*, 16(3)
- Sterck, H. D. and He, Y. (2021). *On the asymptotic linear convergence speed of Anderson acceleration, Nesterov acceleration, and nonlinear GMRES.* *SIAM Journal on Scientific Computing*, 43(5):S21–S46.
- Walker, H. F. and Ni, P. (2011). *Anderson acceleration for fixed-point iterations.* *SIAM J. Numer. Anal.*, 49(4)
- Zhang, J., O'Donoghue, B., and Boyd, S. (2020). *Globally convergent type-I Anderson acceleration for nonsmooth fixed-point iterations.* *SIAM Journal on Optimization*, 30(4) (2020)
- D. Scieur, A. D'Aspremont, F. Bach, *Regularized nonlinear acceleration,* Math. Program., 179 (2020) 47-83.
- A. Toth, C.T. Kelley, *Convergence analysis for Anderson Acceleration,* SIAM J. Numer. Anal., 53(2) (2015)
- T. Rohwedder T, R. Schneider, *An analysis for the DIIS acceleration method used in Quantum Chemistry calculations,* *J. Math. Chem.,* 49(9) (2011) 1889-1914.

## *Concluding remarks*

- Method can be adapted to context of stochastic gradient-type methods
- In deep learning: build $P_j, V_j$ across different batches
- i.e., ignore the fact that the objective function varies with each batch
- Challenge: QN-type methods exploit smoothness but ...
- ... Stochastic character limits smoothness.
- Future:
  - 1) Adapt a few more of the Krylov methods developed in the 1980s
  - 2) Adapt nltgcr to non-smooth context [more to be done here]