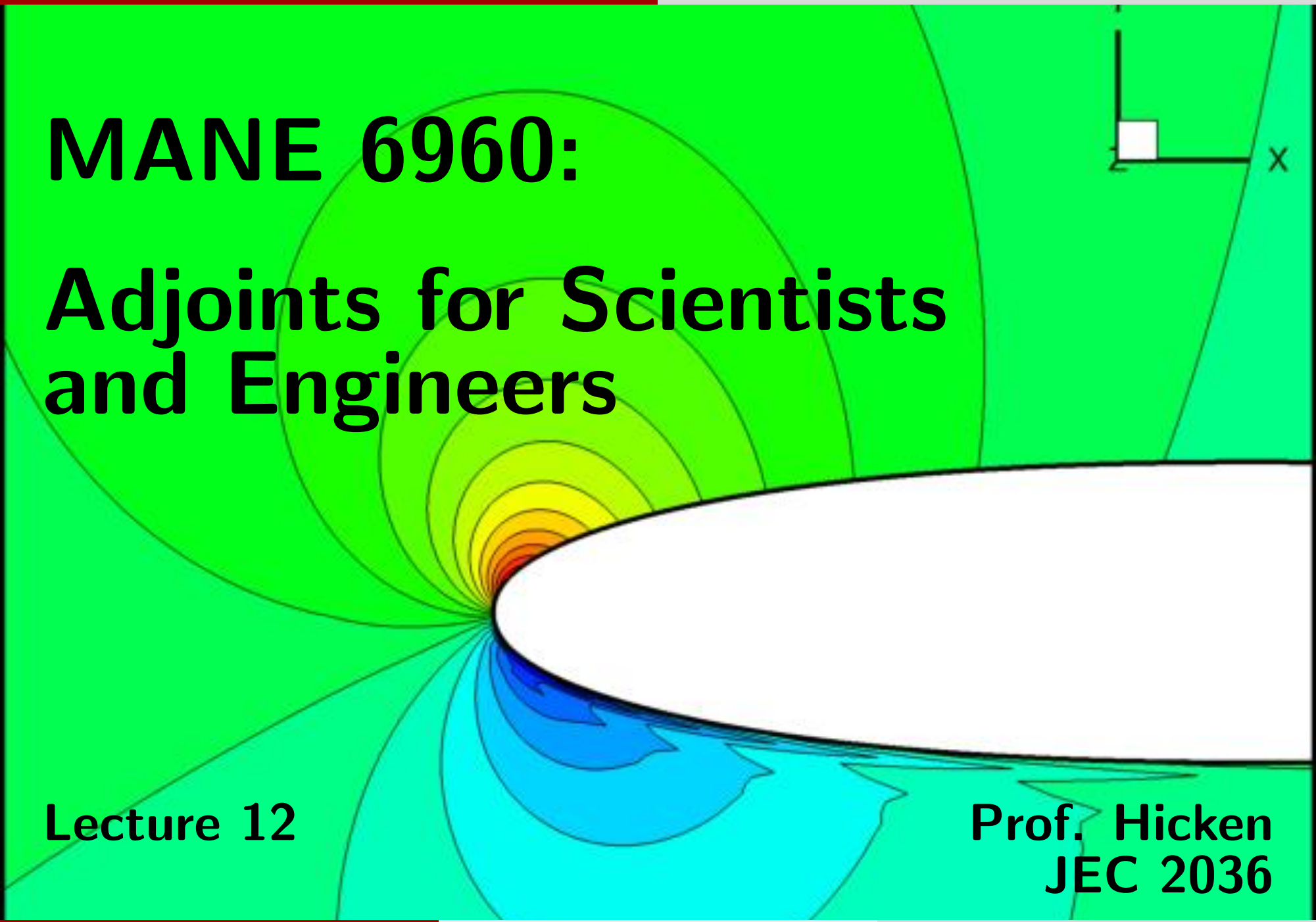
MANE 6960:

Adjoints for Scientists and Engineers



Lecture 12

Prof. Hicken
JEC 2036

Lecture Overview

Once we have the Jacobian $\partial R_h/\partial u_h$ and the derivative $\partial J_h/\partial u_h$, we can solve for the discrete adjoint:

$$\left(\frac{\partial R_h}{\partial u_h}\right)^T \psi_h = \left(\frac{\partial J_h}{\partial u_h}\right)^T.$$

But how should we solve this equation in practice?

- Until now, we have tacitly assumed a direct solve
- But what happens if the Jacobian is too large for a direct solve?

Lecture Overview (cont.)

classes of

In this lecture, we will discuss two popular iterative methods for solving the discrete adjoint equations:

1. exact-dual time marching; and
2. krylov-subspace methods.

Exact-Dual Time Marching Methods

Background

In the following slides I will describe the class of exact-dual time-marching methods first put forward by Giles (see [GDMP03] and the references therein).

- Exact-dual here refers to an adjoint solution method that is the dual of a time-marching method applied to the primal (or linearized primal) problem.
- To describe the method, I closely follow the presentation in [NLPD04], which is based on the implicit Euler method; however, the exact-dual approach can be applied to any time-marching scheme.

Primal Time-Marching Method

Suppose we want to solve the discretized, nonlinear equations

$$R_h(u_h) = 0.$$

If these equations are the steady form of a conservation law, then it is appropriate to consider time-marching the system to steady state:

$$M_h \frac{du_h}{dt} + R_h(u_h) = 0, \qquad u_h(0) = u_{h,0}. \tag{$\dagger$}$$

- M_h is the identity matrix for strong form discretizations, and the mass-matrix for weak-form discretizations.
- If the problem is steady, then $\lim_{t\to\infty} u_h(t)$ will give us the desired solution

Primal Time-Marching Method (cont.)

Following Nielsen *et al.* [NLPD04], we will discretize (†) using the implicit Euler method.

- It is important to emphasize that we do not need to use an accurate (or consistent) time-marching method in this context.

$$\frac{1}{\Delta t} M_h \left(u_h^{(k+1)} - u_h^{(k)} \right) + R_h \left(u_h^{(k+1)} \right) = 0$$

Linearize to get the update eqn

$$\left[\frac{1}{\Delta t} M_h + \frac{\partial R_h}{\partial u_h} \right] \Delta u_h^{(k)} + R_h \left(u_h^{(k)} \right) = 0$$

$$\Delta u_h^{(k)} \equiv u_h^{(k+1)} - u_h^{(k)}$$

Primal Time-Marching Method (cont.)

The above linear system is typically not solved exactly; instead, we replace the inverse system matrix with

$$P^{-1} \approx \left[\frac{1}{\Delta t} M_h + \frac{\partial R_h}{\partial u_h} \right]^{-1}$$

- P^{-1} may be a Gauss-Seidel fixed-point iteration, multigrid, or ILU factorization.

Thus, the final iterative method is

$$u_h^{(k+1)} = u_h^{(k)} - P^{-1} R_h(u_h^{(k)}).$$

Time-Marching the Direct Sensitivity

Recall the direct sensitivity method from last lecture:

Definition: Direct Sensitivity Method

The total derivative of J_h with respect to α_i is

$$\frac{DJ_h}{D\alpha_i} = \frac{\partial J_h}{\partial \alpha_i} + \frac{\partial J_h}{\partial u_h} v_h$$

where $v_h \in \mathbb{R}^s$ is the direct sensitivity and satisfies

$$\frac{\partial R_h}{\partial u_h} v_h + \frac{\partial R_h}{\partial \alpha_i} = 0.$$

Time-Marching the Direct Sensitivity (cont.)

The same iterative method used to solve for u_h can be applied to the unsteady direct sensitivity equation:

$$M_h \frac{dv_h}{dt} + \left(\frac{\partial R_h}{\partial u_h}\right) v_h + \frac{\partial R_h}{\partial \alpha_i} = 0.$$

Starting from implicit Euler, we have

$$\frac{1}{\Delta t} M_h \left(v_h^{(k+1)} - v_h^{(k)}\right) + \left(\frac{\partial R_h}{\partial u_h}\right) v_h^{(k+1)} + \frac{\partial R_h}{\partial \alpha_i} = 0$$

$$\underbrace{\left[\frac{1}{\Delta t} M_h + \left(\frac{\partial R_h}{\partial u_h}\right)\right]}_{\approx P} \Delta v_h^{(k)} + \left[\left(\frac{\partial R}{\partial u_h}\right) v_h^{(k)} + \frac{\partial R_h}{\partial \alpha_i}\right] = 0$$

$$\Delta v_h^{(k)} + P^{-1}\left[\frac{\partial R_h}{\partial u_h} v_h^{(k)} + \frac{\partial R_h}{\partial \alpha_i}\right] = 0$$

Time-Marching the Direct Sensitivity (cont.)

This defines an iterative scheme of the form

$$v_h^{(k+1)} = G v_h^{(k)} + f \qquad (\star)$$

where

$$G = \left[I - P^{-1} \left(\frac{\partial R_h}{\partial u_h} \right) \right], \qquad \text{and} \qquad f = -P^{-1} \frac{\partial R_h}{\partial \alpha_i}.$$

Theorem: [Saa03]

If $\rho(G) < 1$,

Let $\rho(G)$ denote $\max_i |\lambda_i(G)|$, the spectral radius of the matrix G. Then $I - G$ is nonsingular and the iteration $(\star)$ converges for any initial $v_h^{(0)}$, and its convergence rate is given by $-\ln(\rho(G))$.

Exact-Dual Adjoint Method

Last lecture we showed that $DJ_h/D\alpha_i$ is the same regardless of whether we use the direct or adjoint sensitivity method. This was a consequence of the identity

$$\frac{\partial J_h}{\partial u_h} v_h = -\psi_h^T \frac{\partial R_h}{\partial \alpha_i}.$$

However, in general, this identity does not hold for an iterative solution of v_h and ψ_h, that is

$$\frac{\partial J_h}{\partial u_h} v_h^{(k)} \neq -\left(\psi_h^{(k)}\right)^T \frac{\partial R_h}{\partial \alpha_i}.$$

An exact-dual method is one for which the left and right sides are equal for all k

Exact-Dual Adjoint Method (cont.)

Note: $v_h^{(0)} = 0$

To derive the exact dual method, we start with the following identity:

$$\frac{\partial J_h}{\partial u_h} v_h^{(k)} = \frac{\partial J_h}{\partial u_h} v_h^{(k)} - \sum_{j=0}^{k-1} (\lambda^{(j+1)})^T \left(v_h^{(k+1)} - G v_h^{(k)} - f \right)$$

$$= \frac{\partial J_h}{\partial u_h} v_h^{(k)} - \begin{bmatrix} \lambda^{(1)} \\ \lambda^{(2)} \\ \vdots \\ \lambda^{(k)} \end{bmatrix}^T \left\{ \begin{bmatrix} I & & & \\ -G & I & & \\ & \ddots & \ddots & \\ & & -G & I \end{bmatrix} \begin{pmatrix} v_h^{(1)} \\ v_h^{(2)} \\ \vdots \\ v_h^{(k)} \end{pmatrix} - \begin{pmatrix} f \\ f \\ \vdots \\ f \end{pmatrix} \right\}$$

$$= \underbrace{\sum_{j=1}^{k} \lambda^{(j)T} f}_{(\Psi_h^{(k)})^T p} - \begin{bmatrix} v_h^{(1)} \\ v_h^{(2)} \\ \vdots \\ v_h^{(k)} \end{bmatrix}^T \left\{ \begin{bmatrix} I - G^T & & & \\ & I - G^T & & \\ & & \ddots & \\ & & & I \end{bmatrix} \begin{pmatrix} \lambda^{(1)} \\ \lambda^{(2)} \\ \vdots \\ \lambda^{(K)} \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \vdots \\ \left(\partial J_h / \partial u_h\right)^T \end{pmatrix} \right\}$$

Exact-Dual Adjoint Method (cont.)

Defining $\psi_h^{(m)} \equiv \sum_{j=k-m}^{k-1} P^{-T} \lambda^{(j+1)}$ and manipulating the above expression (see [NLPD04]) we arrive at the exact-dual algorithm.

Definition: Exact-dual (Approximate Implicit Euler)

Given the direct-sensitivity iteration ($\star$), the exact-dual iteration for the adjoint is

$$\psi_h^{(k+1)} = \left[I - P^{-T} \left(\frac{\partial R_h}{\partial u_h} \right)^T \right] \psi_h^{(k)} + P^{-T} \left(\frac{\partial J_h}{\partial u_h} \right)^T .$$

or $\psi_h^{(k+1)} = \psi_h^{(k)} - P^{-T} \left[\left(\frac{\partial R_h}{\partial u_h} \right)^T \psi_h^{(k)} - \left(\frac{\partial J_h}{\partial u_h} \right)^T \right]$ adjoint residual

Pros & Cons of the Exact-Dual

- ✓ Jacobian-free, since only products with $(\partial R_h/\partial u_h)^T$ are required, and these can be computed with the reverse-mode of AD
- ✓ memory is limited to that needed to apply P^{-1}
- ✗ linear convergence rate
- ✗ if the linearized problem is unstable, the exact-dual (and direct) iterative schemes will diverge.

Krylov-Subspace Iterative Methods

Krylov Subspace

Let $A \in \mathbb{R}^{n \times n}$ be nonsingular, and consider finding a solution to

$$Ax = b.$$

A Krylov-iterative method finds an approximate solution x_m in the affine space

$$x_0 + \mathcal{K}_m(A, r_0),$$

where x_0 is the initial "guess", $r_0 \equiv b - Ax_0$ is the initial residual, and

$$\mathcal{K}_m(A, r_0) = \underbrace{\text{span}\left\{r_0, Ar_0, A^2 r_0, \ldots, A^{m-1} r_0\right\}}_{\text{Krylov subspace}}$$

Arnoldi's Method

In order to represent the Krylov subspace, we need a stable basis

- The basis $\{r_0, Ar_0, \ldots, A^{m-1}r_0\}$ is not a stable basis, because its vectors become closer and closer to being linearly dependent.

Idea: build a stable basis for the Krylov subspace by applying Gram-Schmidt orthogonalization dynamically, i.e. "on the fly."

- This is called Arnoldi's Method

Arnoldi's Method (cont.)

modified Gram-Schmidt

Algorithm: Arnoldi's Method (MGS Version)

Data: initial residual, $r_0 = b - Ax_0$

Result: V_{m+1}, orthogonal basis for $\mathcal{K}_{m+1}(A, r_0)$

set $v_1 = r_0/\beta$, where $\beta = \|r_0\|$
for $k = 1, 2, \ldots, m$ **do**
  $w = Av_k$
  for $j = 1, 2, \ldots, k$ **do**
    $h_{j,k} = (w, v_j)$
    $w \leftarrow w - h_{j,k}v_j$
  end
  $h_{k+1,k} = \|w\|$
  set $v_{k+1} = w/h_{k+1,k}$
end

Arnoldi's Method (cont.)

It is straightforward to show [Saa03] that Arnoldi's method produces the following identity:

$$AV_m = V_{m+1}\bar{H}_m$$

where

$$V_m = \begin{bmatrix} v_1 & v_2 & \dots & v_m \end{bmatrix}$$

$V_m^T V_m = I_m$

$$\bar{H}_m = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,m} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,m} \\ & h_{3,2} & \cdots & h_{3,m} \\ & & \ddots & \vdots \\ & & & h_{m+1,m} \end{bmatrix} = H_m \in \mathbb{R}^{m \times m}$$

$\in \mathbb{R}^{(m+1)\times m}$

Generalized Minimal Residual Method

The generalized minimal residual (GMRES) method consists of finding the approximate solution $x_m \in x_0 + \mathcal{K}_m(A, r_0)$ that minimizes the 2-norm of the residual.

basis for

$$x_m = x_0 + V_m y_m$$

$$\begin{aligned} \|b - A x_m\|_2 &= \|b - A x_0 - A V_m y_m\| \\ &= \|r_0 - V_{m+1} \bar{H}_m y_m\| \\ &= \|V_{m+1} (\beta e_1 - \bar{H}_m y_m)\| \\ &= \|\beta e_1 - \bar{H}_m y_m\| \end{aligned}$$

minimize this small least-squares problem

Recall: $r_0 = \beta v_1$

$$e_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^{m+1}$$

Full Orthogonalization Method

The full-orthogonalization method (FOM) is obtained by performing a Galerkin projection of r_m onto $\mathcal{K}_m(A, r_0)$ and setting the result to zero.

$$x_m = x_0 + V_m y_m$$

$$V_m^T r_m = 0 \Rightarrow V_m^T (b - Ax_0 - AV_m y_m) = 0$$

$$\Rightarrow V_m^T (V_{m+1} (\beta e_1 - \bar{H}_m y_m)) = 0$$

$$\Rightarrow \beta e_1 - H_m y_m = 0$$

square

$$\Rightarrow y_m = H_m^{-1} \beta e_1$$

$V_m^T V_{m+1}$

$$= \begin{bmatrix} 1 & & 0 \\ & 1 & 0 \\ & \ddots & \vdots \\ & & 1 \; 0 \end{bmatrix}$$

Symmetric Variants

If A is symmetric, then it is easy to show that $H_m^T = H_m$; therefore, H_m is tridiagonal.

$$A V_m = V_{m+1} \bar{H}_m$$

Square

$$\Longrightarrow \quad V_m^T A V_m = V_m^T V_{m+1} \bar{H}_m = H_m$$

but

$$H_m = V_m^T A V_m = V_m^T A^T V_m = H_m^T$$

Furthermore,

$$H_m = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \end{bmatrix} = \begin{bmatrix} \times & \times & & \\ \times & \times & \times & \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} = H_m^T$$

Symmetric Variants (cont.)

In this case, the tridiagonal structure of H_m implies that we do not need to orthogonalize against all previous vectors in Gram-Schmidt.

- Cost per iteration becomes constant
- Memory does not grow with m, number of iterations

For symmetric A, Arnoldi's method becomes the symmetric Lanczos method and

- GMRES becomes MINRES
- FOM becomes the conjugate gradient (CG) method

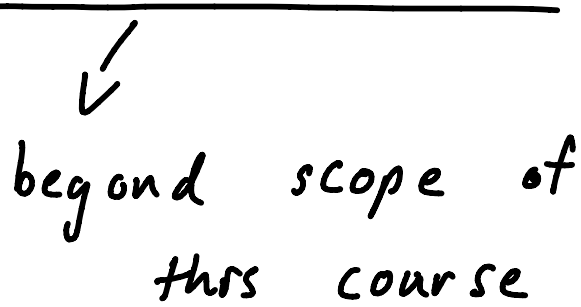
Preconditioning

Krylov methods need to be preconditioned in practice, that is, we need a matrix (or operation) P such that

- $P^{-1} \approx A^{-1}$ and
- P^{-1} is relatively inexpensive to apply.

Krylov iterative methods have preconditioned variants, so the incorporation of P^{-1} is not difficult.

- The challenge is finding an efficient preconditioner.



Krylov Methods and the Adjoint Equation

A few remarks are in order regarding the use of Krylov methods to solve the Adjoint equation,

$$\left(\frac{\partial R_h}{\partial u_h}\right)^T \psi_h = \frac{\partial J_h}{\partial u_h}.$$

preconditioner might not be matrix free

1. Krylov methods are "matrix-free," since Arnoldi's method does not need anything other than matrix-vector products and preconditioning operations.
2. If a Newton-Krylov method is used to solve $R_h(u_h) = 0$, the preconditioner for the linearized primal problem can be reused to solve the adjoint problem (in its transposed form, of course).

Krylov Methods and the Adjoint Equation (cont.)

The linearized primal problem in a Newton-Krylov approach may only need 1–2 orders reduction in its residual norm.

- This is because the Newton step is only an approximation, and there is no need to solve the linearized problem tightly.

In contrast, the adjoint equation may need to be solved very accurately (6-8 orders reduction in residual norm), in order to ensure accurate gradients.

- Such tight tolerances may require many (order 100) Krylov iterations, and this may be impractical in terms of memory.
- The solution is to use restarted, truncated, or deflated Krylov methods; however, be cautious of using bare-bones restarted methods, which can stall.

Pros & Cons of Krylov-based Adjoint Solves

- ✓ Jacobian-free, since only products with $(\partial R_h/\partial u_h)^T$ are required, and these can be computed with the reverse-mode of AD
- ✓ potentially rapid (superlinear) convergence rate
- ✓ Can solve the adjoint even if the linearized problem is unstable
- ✗ Memory can be an issue unless some form of truncation or deflation is used, and then stalling can become a risk.

References

[GDMP03] Michael B. Giles, Mihai C. Duta, Jens-Dominik Müller, and Niles A. Pierce, *Algorithm developments for discrete adjoint methods*, AIAA Journal **41** (2003), no. 2, 198–205.

[NLPD04] Eric J. Nielsen, James Lu, Michael A. Park, and David L. Darmofal, *An implicit, exact dual adjoint solution method for turbulent flows on unstructured grids*, Computers and Fluids **33** (2004), 1131–1155.

[Saa03] Yousef Saad, *Iterative Methods for Sparse Linear Systems*, second ed., SIAM, Philadelphia, PA, 2003.