Reduced Functions, Gradients and Hessians from Fixed Point Iterations for State Equations

Andreas Griewank *
Christèle Faure †

16th January 2002

Abstract In design optimization and parameter identification, the objective, or response function(s) are typically linked to the actually independent variables through equality constraints, which we will refer to as state equations. Our key assumption is that it is impossible to form and factor the corresponding constraint Jacobian, but one has instead some fixed point algorithm for computing a feasible state, given any reasonable value of the independent variables. Assuming that this iteration is eventually contractive we will show how reduced gradients [Jacobians] and Hessians (in other words the total derivatives) of the response[s] with respect to the independent variables can be obtained via algorithmic, or automatic, differentiation (AD).

In our approach the actual application of the so-called reverse, or adjoint differentiation mode is kept local to each iteration step. Consequently the memory requirement is typically not unduly enlarged. The resulting approximating Lagrange multipliers are used to compute estimates of the reduced function values that can be shown to converge twice as fast as the underlying state space iteration. By a combination with the forward mode of AD one can also obtain extra-accurate directional derivatives of the reduced functions as well as feasible state space directions and the corresponding reduced or projected Hessians of the Lagrangian.

Our approach is verified by test calculations on an aircraft wing with two responses, namely lift and drag coefficient, and two variables, namely the angle of attack and the Mach number. The state is a two-dimensional flow field defined as solution of the discretized Euler equation at transonic conditions.

Keywords. fixed point iteration, derivative convergence, algorithmic or automatic differentiation, implicit functions, reduced gradient, reduced Hessian, Q- and R-linear convergence.

1 Introduction

Let us consider a parameter-dependent system of nonlinear equations

$$w \equiv F(z,x) = 0 \quad \text{with} \quad F : \mathbb{R}^l \times \mathbb{R}^n \mapsto \mathbb{R}^l \tag{1}$$

where x represents the vector of independent variables or parameters with respect to which we wish to optimize. The goal is to find some desirable value of the *response* function

$$y = f(z,x) \quad \text{with} \quad f : \mathbb{R}^l \times \mathbb{R}^n \mapsto \mathbb{R}^m \tag{2}$$

*Institute of Scientific Computing, Technical University Dresden, (griewank@math.tu-dresden.de)
†PolySpace Technologies, Paris (christele.faure@polyspace.com)

that evaluates a few key quantities of the *state vector* z. For example the state equation $F(z,x)=0$ may be some discretized version of the Navier–Stokes equation with z representing the flow field around an airfoil and the components of $y = F(z,x)$ being the lift and drag coefficients. This situation serves as a test case in the final Section of our paper. Although this is not necessary for our theoretical statements one may usually assume that the dimension l of the state space is orders of magnitudes larger than the number n of parameters x and the dimension m of y, which we may think of as vector of objectives and soft constraints. In the airfoil example one may include besides the aerodynamic coefficients also volume, manufacturing costs etc. Often some of the design parameters do not enter directly into the state equations but determine some intermediate parameters via a grid generation process. From a conceptual point of view this preprocessing does not effect our basic approach very much, though it may raise some serious implementation issues.

To arrive at desirable response values by varying x we need the reduced function value $y = \varphi(x) \equiv f(z(x),x)$ and its Jacobian $dy/dx = \varphi'(x)$, possibly post- or pre-multiplied by vectors $\dot{x}$ or $\bar{y}$, respectively. In order to effectively eliminate the variables z and w at least theoretically we need the following assumption.

Assumption JR: Jacobian Regularity
On some neighborhood of a solution (z_*,x) with $F(z_*,x)=0$ the Jacobians of $F(z,x)$ and $f(z,x)$ with respect to $z \in \mathbb{R}^l$ and $x \in \mathbb{R}^n$ are once Lipschitz-continuously differentiable and the square matrix

$$F_z(z,x) \;\equiv\; \frac{\partial}{\partial z} F(z,x) \in \mathbb{R}^{l\times l} \qquad \blacksquare$$

is nonsingular at all points in that neighborhood with $\|F_z(z,x)^{-1}\| \le \Gamma$.

Consequences of the Implicit Function Theorem

By the implicit function theorem the Jacobian of $z_* = z_*(x)$ s.t. $F(z_*(x),x)=0$ is given by the matrix

$$Z_* \;\equiv \frac{dz}{dx} \;=\; -\left(\frac{\partial w}{\partial z}\right)^{-1} \frac{\partial w}{\partial x} \;=\; -F_z\left(z_*,x\right)^{-1} F_x\left(z_*,x\right) \;\in \mathbb{R}^{l\times n} \quad . \tag{3}$$

Then it follows by the chain rule for the total derivative of the responses or objectives $y = \varphi(x) \equiv f(z_*(x),x)$ with respect to the design parameters x that

$$\varphi'(x_*) \;\equiv\; \frac{dy}{dx} \;=\; \frac{\partial y}{\partial x} + \frac{\partial y}{\partial z}\,\frac{dz}{dx} \;=\; f_x(z_*,x) - f_z(z_*,x)\,F_z\left(z_*,x\right)^{-1} F_x(z_*,x) \;\in \mathbb{R}^{m\times n}. \tag{4}$$

As we will see below the triple matrix product on the right can be computed in two different ways related to the forward and reverse mode of automatic differentiation. The latter method is more efficient if $m \ll n$ as we may assume and a combination of both yields also second derivatives as well as extra accurate first derivatives.

In order to express second derivatives in standard matrix-vector notation let $\dot{x} \in \mathbb{R}^n$ be an arbitrary but fixed direction in the parameter space and $\dot{z}_* = Z_*\dot{x}$ the corresponding feasible direction in the state space. Denoting differentiation along the straight line $(z_* + t\dot{z}_*, x + t\dot{x})$ with respect to t at $t=0$ by superscript $\cdot$ we obtain from (4) the matrix expression

$$\dot{\varphi}' \equiv \varphi''\dot{x} \;=\; \dot{f}_x - \dot{f}_z\,F_z^{-1}F_x + f_z F_z^{-1}\dot{F}_z F_z^{-1}F_x - f_z F_z^{-1}\dot{F}_x \in \mathbb{R}^{m\times n} \tag{5}$$

where all quantities are evaluated at (z_*, x) and all dotted quantities depend also on $(\dot{z}_*, \dot{x})$ as follows.

$$\dot{f}_x \equiv \dot{f}_x(z_*, x, \dot{z}_*, \dot{x}) \equiv f_{xz}(z_*, x)\dot{z}_* + f_{xx}(z_*, x)\,\dot{x} \in \mathbb{R}^{m\times n} \tag{6}$$

$$\dot{f}_z \equiv \dot{f}_x(z_*, x, \dot{z}_*, \dot{x}) \equiv f_{zz}(z_*, x)\dot{z}_* + f_{zx}(z_*, x)\,\dot{x} \in \mathbb{R}^{m\times l} \tag{7}$$

$$\dot{F}_x \equiv \dot{F}_x(z_*, x, \dot{z}_*, \dot{x}) \equiv F_{xz}(z_*, x)\dot{z}_* + F_{xx}(z_*, x)\,\dot{x} \in \mathbb{R}^{l\times n} \tag{8}$$

$$\dot{F}_z \equiv \dot{F}_z(z_*, x, \dot{z}_*, \dot{x}) \equiv F_{zz}(z_*, x)\dot{z}_* + F_{zx}(z_*, x)\,\dot{x} \in \mathbb{R}^{l\times l} \tag{9}$$

The expansion (5) contains the two triple products $\dot{f}_z\, F_z^{-1}\, F_x$, $f_z\, F_z^{-1}\, \dot{F}_x$ and the quintuple product $f_z\, F_z^{-1}\, \dot{F}_z\, F_z^{-1}\, F_x$, which can be computed in various ways as we shall see. If the feasible direction matrix Z_* defined in (3) and the adjoint matrix of Lagrange multipliers

$$W_* \equiv -f_z(z_*, x)\, F_z^{-1}(z_*, x) \in \mathbb{R}^{m\times l} \tag{10}$$

are known then one can compute the reduced Jacobian in either one of the forms

$$f_x + f_z\, Z_* \;=\; \varphi'(x) \;=\; f_x + W_*\, F_x \in \mathbb{R}^{m\times n}\,.$$

Often one does not need the full reduced Jacobian but only its product with a column vector $\dot{x} \in \mathbb{R}^n$ from the right or a row vector $\bar{y} \in \mathbb{R}^m$ from the left. Then we obtain the *reduced tangent*

$$\dot{y}_* \;=\; \varphi'\,\dot{x} \;=\; f_x\dot{x} - f_z\, F_z^{-1} F_x\dot{x} \;=\; f_x\dot{x} + f_z\,\dot{z}_* \quad \text{with} \quad \dot{z}_* = Z_*\,\dot{x} \tag{11}$$

and *reduced gradient*

$$\bar{x}_* \;=\; \bar{y}\,\varphi' \;=\; \bar{y} f_x - \bar{y}\, f_z\, F_z^{-1} F_x \;=\; \bar{y}\, f_x + \bar{w}_* F_x \quad \text{with} \quad \bar{w}_* = \bar{y}\, W_* \quad . \tag{12}$$

Hence we see that by letting $\dot{x}$ range over the n Cartesian basis vectors in $\mathbb{R}^n$ or $\bar{y}$ range over the m Cartesian basis vectors in $\mathbb{R}^m$ one can compute the reduced Jacobian $\varphi'(x)$ column- or row-wise from the corresponding vectors $\dot{z}_*$ or vectors $\bar{w}_*$, respectively. These then form exactly the columns of the matrix Z_* and the rows of the matrix W_* defined in (3) and (10). Similarly one obtains according to (5) for any feasible direction pair $(\dot{z}_*, \dot{x}) = (Z_*\dot{x}, \dot{x}) \in \mathbb{R}^{l+n}$

$$\dot{\varphi}' \equiv \varphi''\,\dot{x} \;=\; \dot{f}_x + \dot{f}_z\, Z_* + W_*\,\dot{F}_z\, Z_* + W_*\,\dot{F}_x \in \mathbb{R}^{m\times n} \quad . \tag{13}$$

In other words, if Z_* and W_* have been obtained, no more equation solving is required but only directional differentiation in the forward and reverse mode. By letting $\dot{x}$ range over all Cartesian basis vectors in $\mathbb{R}^n$ and thus the corresponding $\dot{z}_*$ over all columns of Z_* one could obtain the full reduced derivative tensor $\varphi''(x) \in \mathbb{R}^{m\times n\times n}$ in form of its n directional contractions $\dot{\varphi}'(x) \equiv \varphi''(x)\,\dot{x}$. It seems doubtful that this enormous effort will often be worthwhile.

The computation of the whole feasible direction matrix Z_* is not needed for calculating the reduced Jacobian φ' and it can also be avoided if one only needs to calculate the product of a given weight vector $\bar{y} \in \mathbb{R}^m$ with the directional derivative(s) $\dot{\varphi}'(x)$. To see this we derive from (5)

$$\begin{aligned} \dot{\bar{x}}_* \equiv \bar{y}\,\dot{\varphi}' \equiv \bar{y}\,\varphi''\dot{x} &= \bar{y}\,\dot{f}_x + \bar{y}\, W_*\,\dot{F}_x + \bar{y}\,\dot{f}_z\, Z_* + \bar{y}\, W_*\dot{F}_z\, Z_* \\ &= \bar{y}\,\dot{f}_x + \bar{w}_*\,\dot{F}_x + \dot{\bar{w}}_* F_x \quad \text{with} \quad \dot{\bar{w}}_* \equiv -\left(\bar{y}\,\dot{f}_z + \bar{w}_*\,\dot{F}_z\right) F_z^{-1}\,. \end{aligned} \tag{14}$$

If $\dot{\bar{x}}_* \in \mathbb{R}^n$ is computed for each feasible direction pair $(\dot{z}_*, \dot{x}) \in \mathbb{R}^{l+n}$ with $\dot{x}$ ranging over the n Cartesian basis vectors in $\mathbb{R}^n$ one obtains the so-called reduced, or *projected* Hessian

$$\bar{y}\,\varphi''(x) \equiv \nabla_x^2[\bar{y}\,\varphi(x)] \in \mathbb{R}^{n\times n} \quad \text{of the Lagrangian} \quad \bar{y}\,\varphi(x) \quad .$$

Normally one would assume that a reduced Hessian is much cheaper to evaluate than the reduced second derivative tensor $\varphi''(x) \in \mathbb{R}^{m\times n\times n}$ because the latter has m times as many elements. The expectation may be wrong in our scenario because the main effort is likely to be the computation of the feasible directions and Lagrange multipliers, which are defined in terms of first derivatives, rather than the actual evaluation of second derivatives of the combined system (F, f).

Optimization Aspects

In optimization calculations with a single objective function $y = f(z, x) \in \mathbb{R}$ the vector $\varphi'(x) \in \mathbb{R}^n$ is called the reduced gradient. At least theoretically, accurate and affordable values of the reduced gradient allow us to treat the optimization of y as an unconstrained problem. This approach leads to a class of schemes for equality constrained optimization problems that are unsurprisingly named *reduced gradient methods*. In a certain sense all rapidly converging constrained optimization methods are ultimately reduced gradients methods, but the wisdom of maintaining more or less exact feasibility earlier on is debatable. On one hand it means that a feasible solution with a somewhat reduced objective function is available, whenever the optimization calculation stops, possibly because the computing resources or the patience of management has been exhausted.

On the other hand, if the optimization can be carried out until the end, then allowing earlier on significant infeasibilities (in the sense of large residuals in the state equation) may reduce the overall runtime significantly. This effect is especially likely to occur when the nature of the state equation $F(z, x) = 0$ is such that (re-)gaining feasibility is a rather slow iterative process. Then one might also be interested in determining quite rapidly whether a suggested change in the design variables x actually leads to a desirably low value of the reduced objective function. The right compromise between feasibility and optimality is the hallmark of a good merit function for judging whether an optimization step has been successful or not. In the context of design optimization gaining feasibility and optimality at the same time has been proposed as one-shot approach by S. Ta'asan [TKS92] and also employed successfully by A. Jameson [Jam95]. In this paper we will not debate these wider issues but concentrate on the task of arriving as fast as possible at accurate values of the reduced function and its first and second derivatives.

Note that we have assumed here and throughout that there is a given partitioning of all variables into a set of state variables z and a set of design variables x with the former being considered as dependent on the latter via the state equation. In the terminology of MINOS and similar nonlinear programming tools the state variables are always basic and the design variables always nonbasic. This fixed partition is only possible due to the regularity Assumption **JR** and because we have excluded the possibility of inequality constraints. If the latter are also present our analysis applies only locally, once all active constraints have been identified.

Preview of Contents

In Section 2 of the paper we set up direct and adjoint sensitivity equations whose solutions can be interpreted as feasible directions and Lagrange multipliers, respectively. Either of them yield immediately the desired reduced first derivatives. In addition we consider a second order sensitivity equation. In view of their size in typical applications the sensitivity equations can usually not be solved exactly and we must instead accept approximate solutions obtained by iterative solvers.

In Section 3 we express reduced function values and their first and second derivatives in terms of approximate feasible directions and/or Lagrange multipliers. Both are needed to obtain first derivatives with enhanced accuracy and second derivatives with normal accuracy. In Section 4 we review some fundamental results on the convergence of contractive fixed point iterations. In the subsequent Section 5 we apply these results first to the direct or forward differentiation of the original iteration loop and then we analyse the new approach of *iterated adjoints*. It differs completely from the iteration that would be obtained by mechanically applying the reverse mode to the original iteration. In particular the memory requirement does not grow with the number of steps taken. As a result we obtain estimates of the reduced function and its directional derivatives that converge twice as fast as the underlying state space iterates. By way of explanation for this superconvergence result we consider in Section 6 the special relations that apply when the state equations are linear and $n = 1 = m$. In Section 7 our approach is numerically validated on an Euler code in two dimensions. The paper concludes with the customary summary and tentative conclusions in Section 8.

2 The Direct, Adjoint and Second Order Sensitivity Equation

Rather than directly dealing with the matrices Z_* and W_* or even second derivative tensors we consider from now on for given $\dot{x} \in \mathbb{R}^n$ and $\bar{y} \in \mathbb{R}^m$ corresponding individual vectors $\dot{z}_* \in \mathbb{R}^n$, $\bar{w}_* \in \mathbb{R}^l$ and $\dot{\bar{w}}_* \in \mathbb{R}^l$ as defined in (11), (12), and (14). They can be characterized as solutions of the following sensitivity equations

$$0 \;=\; \dot{F}(z_*,\, x, \dot{z}_*,\, \dot{x}) \;\equiv\; F_z(z_*,\, x)\dot{z}_* \;+\; F_x(z_*, x)\,\dot{x} \;\in\; \mathbb{R}^l \tag{15}$$

$$0 \;=\; \bar{F}(z_*, x, \bar{w}_*,\, \bar{y}) \;\equiv\; \bar{w}_*\, F_z(z_*, x) + \;\bar{y}\, f_z(z_*,\, x) \;\in\; \mathbb{R}^l \tag{16}$$

and

$$0 \;=\; \dot{\bar{F}}(z_*, x, \dot{z}_*, \dot{x}, \bar{w}_*, \bar{y}, \dot{\bar{w}}_*) \;\equiv\; \dot{\bar{w}}_*\, F_z(z_*, x) + \bar{w}_*\, \dot{F}_z(z_*, x, \dot{z}_*, \dot{x}) + \bar{y}\, \dot{f}_z(z_*, x, \dot{z}_*, \dot{x}) \tag{17}$$

where $\dot{F}_z$ and $\dot{f}_z$ are as defined in (9) and (7). Notice that the last equation can be obtained by formally deriving the previous one with $\dot{\bar{y}}$ assumed zero throughout. Having obtained the vectors $\dot{z}_*$, $\bar{w}_*$ and $\dot{\bar{w}}_*$ one may compute the restricted derivative information $\dot{y}_*$, $\bar{x}_*$, and $\dot{\bar{x}}_*$ defined in (11), (12), and (14).

At least theoretically each of the vectors $\dot{z}_*$, $\bar{w}_*$, and $\dot{\bar{w}}_*$ can be computed by solving one linear system involving the Jacobian $F_z(z_*, x)$ and its transpose, respectively. The corresponding right-hand sides

$$\dot{F}(z_*, x, 0, -\dot{x}) \quad , \quad \bar{F}(z_*, x, 0, -\bar{y}) \quad , \quad \dot{\bar{F}}(z_*, x, \dot{z}_*, \dot{x}, -\bar{w}_*, -\bar{y}, 0) \;\in\; R^l$$

can be evaluated by a forward sweep, a reverse sweep, or a combined forward and reverse sweep on the evaluation procedure for (F, f), respectively. Hence, in all three cases the operations count for the right hand side is a small multiple of that for (F, f) itself. The same applies to the memory requirement for the adjoints $\bar{F}(z_*, x, 0, -\bar{y})$ and $\dot{\bar{F}}(z_*, x, \dot{z}_*, \dot{x}, -\bar{w}_*, -\bar{y}, 0)$, which is also proportional to the basic operations count, unless more sophisticated versions of the reverse mode with checkpointing [Gri00] are employed. We expect this to be necessary only in rather exceptional cases where the evaluation of F itself involves time-like evolutions.

In many applications the main obstacle to solving the sensitivity equations is that the Jacobian cannot be formed and factored at a reasonable cost. We will certainly make this assumption

here, since otherwise one may also perform Newton steps and the whole idea of extracting extra information from the users fixed point iteration becomes moot. The same would still be true if the Jacobian could be preconditioned well enough such that a suitable iterative solver could find Newton steps or the solutions to our sensitivity equations quite rapidly, even when high accuracy was required. Hence we will assume that the iterative solution of the state equation is a rather drawn out process, possibly effected by a legacy code including various tricks of the trade that the current users may not fully be aware of. While this is the scenario to which our approach is applicable in principle, we will in fact inch back towards the Newton-like scenario sketched above when it comes to establishing convergence at certain asymptotic rates. However, there is numerical evidence that the approach works still in cases where the assumptions of our theory are not satisfied, or at least not easily verified. For example, this is the case for the Euler code for which we obtained the numerical results listed in Section 7.

In what one might call a two-phase approach many researchers solve the sensitivity equations separately, after the state z_* has been approximated with satisfactory accuracy. Here we will utilize a piggy-back approach, where these linear equations are solved simultaneously with the original state equation. Whatever methods one uses to generate approximate solutions $\dot{z}$, $\bar{w}$ and $\dot{\bar{w}}$ to the sensitivity equations their quality can be gauged by evaluating the *derivative residuals* $\dot{F}(z_*, \dot{z})$, $\bar{F}(z_*, \bar{w})$, and $\dot{\bar{F}}(z_*, \dot{z}, \bar{w}, \dot{\bar{w}})$ defined in (15),(16), and (17). Here and sometimes in the remainder of this paper we omit the argument vectors $x, \dot{x}$ and $\bar{y}$ because they are always selected as constants. The derivative residual vectors can be obtained just as cheaply as the right hand sides mentioned above and bound the derivative errors as follows

Proposition 1 (Derivative Vector Accuracy Bounds)
Under Assumption **JR** *and with $\dot{x} \in \mathbb{R}^n$ or $\bar{y} \in \mathbb{R}^m$ fixed there exist constants $\delta > 0$ and $\gamma < \infty$ such that with $\dot{F}$ defined in (15)*

$$\|\dot{z} - \dot{z}_*\| \;\le\; \gamma\,(\|F(z,x)\| + \|\dot{F}(z,x,\dot{z},\dot{x})\|)$$

and with $\bar{F}$ defined in (16)

$$\|\bar{w} - \bar{w}_*\| \;\le\; \gamma\,(\|F(z,x)\| + \|\bar{F}(z,x,\bar{w},\bar{y})\|)$$

and with $\dot{\bar{F}}$ defined in (17)

$$\|\dot{\bar{w}} - \dot{\bar{w}}_*\| \;\le\; \gamma\,(\|F(z,x)\| + \|\dot{F}(z,x,\dot{z},\dot{x})\| + \|\bar{F}(z,x,\bar{w},\bar{y})\| + \|\dot{\bar{F}}(z,x,\dot{z},\dot{x},\bar{w},\bar{y},\dot{\bar{w}})\|) \qquad \square$$

for all z with $\|z - z_*\| < \delta$ and $\dot{z}, \bar{w}, \dot{\bar{w}} \in \mathbb{R}^l$ arbitrary.

Proof The first two inequalities were established as Lemma 11.2 on page 285 in [Gri00]. The last follows in a similar fashion as errors in z, $\dot{z}$, $\bar{w}$, and the residual $\dot{\bar{F}}$ itself all perturb the supposed identity (17). ■

The constant γ is a function of local Lipschitz constants and the bound Γ on the size of the inverse $F_z(z,x)^{-1}$. As always in nonlinear equation solving good estimates for these quantities are hard to come by. As one can see in Proposition 1 both derivative vectors $\dot{z}$ and $\bar{w}$ are usually affected by error in the underlying z, which explains why there is often a time-lag in their convergence. It can be expected to be even larger for $\dot{\bar{w}}$, which also depends on the other two derivative vectors $\dot{z}$ and $\bar{w}$. The delay has been observed on most iterative schemes other than Newton's method.

3 Approximating Reduced Functions and Derivatives

As an immediate consequence of Proposition 1 we note that by replacing the exact vectors $\dot{z}_*$, $\bar{w}_*$ and $\dot{\bar{w}}_*$ in the formulas (11), (12), and (14) by approximations $\dot{z}$, $\bar{w}$ and $\dot{\bar{w}}$ one obtains also first order approximations to vectors of first or second reduced derivatives. Moreover, as originally suggested by Christianson in [Chr98], one can use the approximate derivatives obtained to compute corrected reduced function and Jacobian values whose error is essentially that of the first order estimates squared.

Corollary 1 (Corrected Function Estimate)
Under the assumptions of Proposition 1 and with any vector $\bar{w} \in \mathbb{R}^n$ for given $\bar{y} \in \mathbb{R}^l$, the corrected value

$$\sigma \equiv \sigma(z, x, \bar{w}, \bar{y}) \equiv \bar{y} f(z, x) + \bar{w} F(z, x) \tag{18}$$

satisfies

$$\left| \bar{y}\, \varphi(x) - \sigma(z, x, \bar{w}, \bar{y}) \right| \quad \leq \; \Gamma \, \|F(z,x)\| \, \|\bar{F}(z, x, \bar{w}, \bar{y})\| + \mathcal{O}(\|F(z,x)\|^2) \qquad \square$$

Proof The assertion follows from the Taylor expansions

$$\begin{aligned} f(z_*, x) &= f(z, x) + f_z(z, x)(z_* - z) + \mathcal{O}(\|z - z_*\|^2) \quad , \\ 0 = F(z_*, x) &= F(z, x) + F_z(z, x)(z_* - z) + \mathcal{O}(\|z - z_*\|^2) \end{aligned}$$

by the definition of $\bar{F}$ and with Γ as an upper bound on the inverse Jacobian norm. ∎

As we will see in Section 5 one can expect that within an iterative procedure the corrected estimate $\bar{y} f(z, x) + \bar{w} F(z, x)$ converges roughly twice as fast as $\bar{y} f(z, x)$ to the actual reduced function value $\bar{y} f(z_*, x)$. This technique for the doubling of the order of the estimate is related to the methods of Pierce and Giles [PG00] and Venditti and Darmofal [VD00] for obtaining superconvergent approximations to integral quantities from the solution of partial differential equations.

A similar superconvergence result can be obtained for the partial derivatives of the reduced function if Lagrange multipliers $\bar{w}$, feasible directions $\dot{z}$, and second order adjoints $\dot{\bar{w}}$ are available. First we notice that the approximating vectors

$$\dot{y} \;\equiv\; f_x(z, x)\dot{x} + f_z(z, x)\dot{z} \in \mathbb{R}^m \quad \text{and} \quad \bar{x} \;\equiv\; \bar{y} f_x(z, x) + \bar{w} F_x(z, x) \in \mathbb{R}^n \tag{19}$$

yield the following results for the desired reduced partial

$$\bar{y}\, \dot{y} + \mathcal{O}(\|F(z)\| + \|\dot{F}(z, \dot{z})\|) \;=\; \bar{y}\, \dot{y}_* \;\equiv\; \dot{\sigma}_* \;\equiv\; \bar{x}_*\, \dot{x} \;=\; \bar{x}\, \dot{x} + \mathcal{O}(\|F(z)\| + \|\bar{F}(z, \bar{w})\|) \quad . \tag{20}$$

Hence, one may get a first order approximation to each reduced Jacobian component by approximately solving either the direct or the adjoint sensitivity equation. We can get a corresponding second order estimate by applying Corollary 1 to the composite state equation

$$\mathbf{F}(\mathbf{z}, \mathbf{x}) \equiv \begin{bmatrix} F(z, x) \\ \dot{F}(z, x, \dot{z}, \dot{x}) \end{bmatrix}, \tag{21}$$

where

$$\mathbf{x} \;\equiv\; \begin{bmatrix} x \\ \dot{x} \end{bmatrix}, \quad \mathbf{z} \;\equiv\; \begin{bmatrix} z \\ \dot{z} \end{bmatrix}, \tag{22}$$

with the new response function

$$\mathbf{f}(\mathbf{z}, \mathbf{x}) \;\equiv\; \dot{f}(z, x, \dot{z}, \dot{x}) \;\equiv\; f_z(z, x)\; \dot{z} + f_x(z, x)\; \dot{x}, \tag{23}$$

Corollary 2 (Corrected Partial Estimate)
If the Assumption **JR** *holds with F twice Lipschitz-continuously differentiable and $\dot{x} \in \mathbb{R}^n$ and $\bar{y} \in \mathbb{R}^m$ are fixed then the corrected estimate*

$$\dot{\sigma} \equiv \dot{\sigma}(z, \dot{z}, \bar{w}, \dot{\bar{w}}) \;\equiv\; \bar{y}\dot{f}(z, \dot{z}) + \bar{w}\dot{F}(z, \dot{z}) + \dot{\bar{w}}F(z) \tag{24}$$

satisfies

$$\bar{y}\,\varphi'(x)\,\dot{x} \;-\; \dot{\sigma}(z, \dot{z}, \bar{w}, \dot{\bar{w}}) \;=\; \mathcal{O}\big(\|F(z)\| + \|\dot{F}(z, \dot{z})\| + \|\bar{F}(z, \bar{w})\| + \|\dot{\bar{F}}(z, \dot{z}, \bar{w}, \dot{\bar{w}})\|\big)^2 \qquad \square$$

for any vectors $\dot{z}, \bar{w}, \dot{\bar{w}} \in \mathbb{R}^l$.

Proof The Jacobian matrix of the extended system (21) is given by

$$\mathbf{F}_z = \begin{bmatrix} F_z \;,\; 0 \\ \dot{F}_z \;,\; F_z \end{bmatrix} \quad \text{where} \quad \dot{F}_z = F_{zz}\,\dot{z} + F_{zx}\,\dot{x} \quad .$$

By assumption it is locally Lipschitz-continuous and also invertible. Hence we can apply the previous corollary with the adjoint system given by

$$0 \;\equiv\; \bar{\mathbf{F}}(\mathbf{z}, \mathbf{x}, \bar{\mathbf{w}}, \bar{\mathbf{y}}) \;\equiv\; \bar{\mathbf{w}} \begin{bmatrix} F_z \;,\; 0 \\ \dot{F}_z \;,\; F_z \end{bmatrix} + \bar{y}\begin{bmatrix} \dot{f}_z, f_z \end{bmatrix} \quad . \tag{25}$$

Partitioning $\bar{\mathbf{F}} = (\dot{\bar{F}}, \bar{F})$ and $\bar{\mathbf{w}} = (\dot{\bar{w}}, \bar{w})$ we find that (25) is equivalent to the two equations (16), (17), and that

$$\bar{\mathbf{w}}\mathbf{F}(\mathbf{z}) \;=\; \dot{\bar{w}}F(z) + \bar{w}\dot{F}(z, \dot{z})$$

so that $\dot{\sigma} = \boldsymbol{\sigma}$ satisfies indeed the assertion as naturally

$$\|\mathbf{F}(\mathbf{z})\| \;=\; \mathcal{O}\big(\|F(z)\| + \|\dot{F}(z)\|\big) \quad \text{and} \quad \|\mathbf{F}(\mathbf{z}, \bar{\mathbf{w}})\| \;=\; \mathcal{O}\big(\,\|\bar{F}(z, \bar{w})\| + \|\dot{\bar{F}}(z, \dot{z}, \bar{w}, \dot{\bar{w}})\|\,\big) \tag{26}$$

■

It follows immediately from Corollary 2 that the error in the reduced partial $\dot{\sigma}$ is essentially of order $\|F\| + \|\dot{F}\|\,\|\bar{F}\|$ if $\dot{\bar{w}}$ is not computed at all but simply defaults to zero. The resulting synergetic estimate is still likely to be better than either simple first order estimate if the state equation itself has been solved to a significantly higher accuracy than the sensitivity equations.

As we will see below the derivative estimates $\dot{z}$, $\bar{w}$, and $\dot{\bar{w}}$ generated from fixed point contractions converge at about the same linear rate as the underlying iterates z themselves. The same is true for the corresponding residuals $F(z), \dot{F}(z, \dot{z})$, $\bar{F}(z, \bar{w})$, and $\dot{\bar{F}}(z, \dot{z}, \bar{w}, \dot{\bar{w}})$ where we have again omitted the constant arguments $x, \dot{x}$ and $\bar{y}$. Whereas the corrected function estimate σ given in Corollary 1 is probably very useful for an efficient optimization procedure it is not yet clear whether this is also true for the corrected partial derivatives $\dot{\sigma}$. The extra effort needed for the computation of $\dot{z}$ and $\dot{\bar{w}}$ might pay off if $\dot{x}$ is a prospective search direction so that an accurate directional derivative estimate $\dot{\sigma}$ would be helpful in line-searches or in a truncated Newton iteration.

We may distinguish two kinds of cost that are incurred when we evaluate reduced first and second derivatives. Probably the dominant expense is to compute vectors $\dot{z}$, $\bar{w}$ and possibly $\dot{\bar{w}}$ for various settings of $\dot{x}$ and $\bar{y}$ as approximate solutions of the appropriate sensitivity equation (15), (16), or (17). Subsequent collections of these vectors must be substituted into the right hand sides (11), (12), (13), or (14), which effectively means performing some forward and/or reverse differentiation on the function $F(z, x)$. Except when the full second derivative tensor φ'' is evaluated via (13) the number of such local differentiations is not very large and we may assume that their total cost is dominated by the required effort to solve sensitivity equations at least approximately. In Table 1 of computational costs we have only counted the number of such solutions required for computing various derivatives with normal and double accuracy.

Table 1: Cost factors for derivative estimates with single and double accuracy

Symbol	Degree	Components	Single	Double	Name
φ	0	m	1	2	Reduced Function
$\bar{y}\,\varphi'$	1	n	2	n	Reduced Gradient
$\varphi'\,\dot{x}$	1	m	2	m	Reduced Tangent
φ'	1	$m \times n$	$\min(m,n)$	$m\,n$	Reduced Jacobian
$\bar{y}\,\varphi''\,\dot{x}$	2	n	3	–	Second Order Adjoint
φ''	2	$m \times n \times n$	$m+n$	–	Reduced Tensor

4 Contractions and their Convergence Rates

The iterates $z_k \in \mathbb{R}^l$ generated by many practical methods for approximating a solution $z_* = z_*(x)$ with $F(z_*, x) = 0$ satisfy a recurrence of the form

$$z_{k+1} \;=\; H_k(z_k, x) \;\equiv\; z_k \;-\; P_k F(z_k, x)\;. \tag{27}$$

Here the preconditioner P_k is some $l \times l$ matrix that approximates the inverse of the Jacobian $F_z(z_k, x)$. The closer that approximation, the more the iteration resembles Newton's methods with its excellent local convergence properties. As long as $F(z_k, x) \neq 0$ any new iterate z_{k+1} can be written in the form (27) since we have not yet imposed any conditions on P_k. To ensure stable convergence from within a vicinity of z_* we make the following assumption

Assumption CP: Contractive Preconditioning
The preconditioners P_k satisfy

$$\|\!| I - P_k\, F_z(z_*, x) |\!\| \;\leq\; \rho \;<1 \quad \text{for all} \quad k \tag{28}$$

■

with respect to some induced matrix norm $\|\!| \cdot |\!\|$.

Because the norm $\|\!| \cdot |\!\|$ must be independent of k this hypothesis is a little stronger than the condition that the spectral radius ($\equiv$ modulo of largest eigenvalue) of all $[I - P_k F_z(z_*, x)]$ are uniformly bounded below 1. According to Ostrowski's Theorem (see Propositions 10.1.3 and 10.1.4 in [OR70]) it follows from Assumption **CP** that all initial guesses z_0 whose distance to z_* is less than some bound lead to convergence with

$$Q\{z_k - z_*\}_{k\,\in\,\mathbb{N}} \;\equiv\; \limsup_k \frac{\|\!| z_{k+1} - z_* |\!\|}{\|\!| z_k - z_* |\!\|} \;\leq\; \rho\;. \tag{29}$$

Here the vector norm $\|\!| \cdot |\!\|$ must be consistent with the matrix norm used in Assumption **CP** so that for any square matrix $A \in \mathbb{R}^{l\times l}$

$$\|\!| A |\!\| \;\equiv\; \max_{0\neq z\in\mathbb{R}^l} \|\!| Az |\!\| / \|\!| z |\!\|\;.$$

Quotient and Root Convergence Factors

The chosen norm strongly influences the so-called Q-factor $Q\{z_k - z_*\}$ defined by (29) for any iteration sequence $\{z_k - z_*\}$ with $z_k \neq z_*$ for all k. In contrast it follows from the equivalence of all norms on finite dimensional spaces that the R-factor

$$R\{z_k - z_*\}_{k \in \mathbb{N}} \equiv \limsup_k \sqrt[k]{\|z_k - z_*\|} \leq Q\{z_k - z_*\}_{k \in \mathbb{N}} \leq \rho \tag{30}$$

is norm independent. The last inequality holds by (29) and the other one is established as Proposition 9.3.1 in [OR70]. In both (29) and (30) we may replace the uniform bound $\rho < 1$ from Assumption **CP** by the corresponding limit superior

$$\rho_o \equiv \limsup_k \|I - P_k F_z(z_k, x)\| \leq \rho \tag{31}$$

so that in conclusion

$$\rho_* \equiv R\{z_k - z_*\}_{k \in \mathbb{N}} \leq Q\{z_k - z_*\}_{k \in \mathbb{N}} \leq \rho_0 \leq \rho < 1 . \tag{32}$$

When $R\{z_k - z_*\}_{k \in \mathbb{N}} = 0$ the convergence is said to be *R-superlinear* and when even $Q\{z_k - z_*\} = 0$ it is called *Q-superlinear*. The latter, highly desirable property is again norm invariant and can be established for certain secant-updating methods [DS96] without the even stronger condition $\rho_0 = 0$ necessarily being satisfied. This and other situations where $\rho_* < \rho_0$ are particularly advantageous with the respect to the corrected function value. Provided $I - P_0 F_z$ is a compact operator the superlinear convergence property of secant updating schemes can also be established in a Hilbert space setting.

Convergence Rates of Residuals

Except on academic test functions one can normally not compute the solution error $\|z_k - z_*\|$ and must therefore be content to gauge the quality of the current approximation z_k in terms of the residual $F_k \equiv F(z_k, x)$. Under our Assumptions **JR** one may view $\|F(z_k, x)\|$ as an equivalent norm to $\|z_k - z_*\|$ since there must be constants $\delta > 0$ and $1 \leq \gamma < \infty$ such that

$$\frac{1}{\gamma} \leq \frac{\|F(z, x)\|}{\|z - z_*\|} \leq \gamma \quad \text{for} \quad \|z - z_*\| < \delta .$$

This implies for any sequence z_k converging to, but never attaining exactly z_* that

$$R\{F_k\}_{k \in \mathbb{N}} = R\{z_k - z_*\}_{k \in \mathbb{N}} \leq Q\{F_k\}_{k \in \mathbb{N}} \leq \gamma^2 Q\{z_k - z_*\}_{k \in \mathbb{N}} .$$

In particular we have the equivalent superlinear convergence conditions

$$Q\{z_k - z_*\}_{k \in \mathbb{N}} = 0 \quad \Leftrightarrow \quad Q\{F_k\}_{k \in \mathbb{N}} = 0$$

and

$$R\{z_k - z_*\}_{k \in \mathbb{N}} = 0 \quad \Leftrightarrow \quad R\{F_k\}_{k \in \mathbb{N}} = 0 .$$

To succinctly indicate convergence with the R-factor ρ_* we will write

$$z_k = z_* + \tilde{\mathcal{O}}(\rho_*^k) \quad \text{and} \quad F(z_k, x) = \tilde{\mathcal{O}}(\rho_*^k) .$$

In practice one may use the estimates

$$R\{F_k\} \;\approx\; \left(\frac{\|F_k\|}{\|F_0\|}\right)^{\frac{1}{k}} \quad \text{and} \quad Q\{F_k\} \;\approx\; \max\left\{\frac{\|F_k\|}{\|F_{k-1}\|}, \frac{\|F_{k+1}\|}{\|F_k\|}\right\}$$

to track the progress of the iteration. Here we have somewhat arbitrarily chosen to maximize over two successive residual reduction ratios for the Q-factor. The reason is that some kind of alternating approach seems to be a convergence pattern that occurs reasonably often, especially in the vicinity of singularities or severe ill-conditioning [Gri80, NS96].

5 Derivative Recurrences and their Convergence

In this section we attempt to derive from the original fixed point iteration extra recurrences that simultaneously compute the desired derivative quantities in a piggyback fashion. First we consider more or less straight forward differentiation. Suppose the preconditioner matrix P_k is for each k at least locally a smooth function of (z, x); often it will even be constant. Moreover, let x vary along the straight $x = x(t) = x(0) + t\,\dot{x}$ as a function of the scalar parameter $t \approx 0$. Then it follows by induction on k that when $z_k = z_k(t)$ is differentiable in t so is z_{k+1} and the derivatives $\dot{z}_k = \dot{z}_k(t)$ must satisfy the recurrence

$$\dot{z}_{k+1} \;=\; \dot{z}_k - P_k\,\dot{F}(z_k, x, \dot{z}_k, \dot{x}) - \dot{P}_k\,F(z_k, x)\;. \tag{33}$$

Here the matrix $\dot{P}_k \equiv d\,P_k(z_k(t), x(t))/dt$ is the derivative of the preconditioner with respect to t, which exists under the assumption made above. The derivative residual $\dot{F}(z_k, x, \dot{z}_k, \dot{x})$ is defined by (15).

The last term $\dot{P}_k\,F(z_k, x)$ is in some way the most interesting. If the preconditioner is fixed so that (27) reduces to a simple substitution method, the last term vanishes since clearly $\dot{P}_k \equiv 0$. Even if the $\dot{P}_k$ are nonzero but their size is uniformly bounded, the term $\dot{P}_k\,F(z_k, x)$ disappears gradually as $F_k = F(z_k, x)$ converges to zero. This happens for example in Newton's method where $P_k = F_z(z_k, x)^{-1}$ is continuously differentiable in (z_k, x), provided F itself is at least twice continuously differentiable. However, second derivatives should not really come into it at all as the implicit derivative $\dot{z}_*$ is according to the explicit representation $\dot{z}_* = Z_*\dot{x}$ with Z_* given by (3) uniquely defined by the extended Jacobian of F. Hence we may prefer to simply drop the last term and use instead the *simplified recurrence*

$$\dot{z}_{k+1} \;=\; \dot{z}_k - P_k\,\dot{F}(z_k, x, \dot{z}_k, \dot{x})\;. \tag{34}$$

The implementation of this recurrence requires the *deactivation* of P_k when this preconditioner depends on x as it usually does. By this we mean that the dependence on x is suppressed so that it looks as through P_k consists of real entries that have fallen from the sky. Whether and how this can be done depends on the particular AD tool. It should not be made too easy, because the unintentional suppression of active dependencies can lead to wrong derivative values. In the numerical calculations reported in Section 7 no deactivation was performed but we believe that the P_k may be assumed to be piece-wise constant. For our theoretical analysis we make the following assumption

Assumption UL: Lipschitz Continuous Dependence
For the convergent sequence $z_k \to z_*$ the preconditioners P_k are all differentiable with respect to

(z, x) on some neighborhood of (z_k, x) such that the derivative tensors

$$\frac{\partial}{(\partial z, \partial x)} P_k \in \mathbb{R}^{l \times l \times (l+n)} \tag{35}$$

are uniformly bounded over all k. ■

Under this assumption both derivative recurrences lead to the same convergence according to the following generalization of a result by Jean Charles Gilbert [Gil92].

Proposition 2 (Root Convergence Factor of Derivatives) *Under Assumptions* **CP** *and* **UL** *we have for all z_0 sufficiently close to z_* and arbitrary $\dot{z}_0$ applying, either (33) or (34)*

$$R\{\dot{z}_k - \dot{z}_*\}_{k \in \mathbb{N}} \;\leq\; \dot{\rho} \equiv \max\Big(\rho_*,\; R\{\dot{F}(z_k, , x, \dot{z}_k, \dot{x})\}_{k \in \mathbb{N}}\Big) \;\leq\; \rho_o \qquad \square$$

where ρ_o is defined in (31).

Proof See [GBC+93]. ■

According to Proposition 2 the complete (33) and the simplified (34) derivative recurrence yield R-linear convergence to the exact derivative $\dot{z}_*$. In both cases the root convergence factor is bounded by the limiting spectral radius ρ_o, which also bounds the quotient convergence factor of the iterates z_k themselves. If $\rho_o = 0$ we have Q-superlinear convergence of the z_k and the slightly weaker property of R-superlinear convergence for the $\dot{z}_k$. This result applies in particular for Newton's method, where the z_k converge in fact quadratically.

In [GBC+93] a considerable effort was made to extend the result to quasi-Newton methods where the sequence P_k is obtained by secant updating, which leads to Assumption **UL** being violated in general. Nevertheless, it could be shown that due to classical convergence characteristics of secant updating methods the decline of the residuals $F(z_k, x)$ is just fast enough that the last term in (33) still gradually disappears. It should be noted however, that in this situation the derivatives $\dot{z}_k$ (and the adjoints $\bar{w}_k$ discussed below) do in general still only converge linearly, while the z_k themselves converge superlinearly. For many other methods like conjugate gradient type schemes and of course multi-level approaches our assumptions are not easily verified. The simplified piggyback iteration for solving simultaneously the equations $F(z) = 0$ and $\dot{F}(z, \dot{z}) = 0$ is displayed in Table 2 below.

Table 2: Direct Fixed Point Iteration

fix $x, \dot{x} \in \mathbb{R}^n$
initialize $z_0, \dot{z}_0 \in \mathbb{R}^l$
for $k = 0, 1, 2, \ldots$
$\quad w_k = F(z_k, x)$
$\quad \dot{w}_k = \dot{F}(z_k, x, \dot{z}_k, \dot{x})$
$\quad$ stop if $\|w_k\|$ and $\|\dot{w}_k\|$ are small
$\quad z_{k+1} = z_k - P_k\, w_k$
$\quad \dot{z}_{k+1} = \dot{z}_k - P_k\, \dot{w}_k$
$y_k = f(z_k, x)$
$\dot{y}_k = f_z(z_k, x)\dot{z}_k + f_x(z_k, x)\dot{x}$

Table 3: Adjoint Fixed Point Iteration

fix $x \in \mathbb{R}^n, \bar{y} \in \mathbb{R}^m$
initialize $z_0, \bar{w}_0 \in \mathbb{R}^l$
for $k = 0, 1, 2, \ldots$
$\quad [w_k, y_k] = [F(z_k, x), f(z_k, x)]$
$\quad \bar{z}_k = \bar{F}(z_k, x, \bar{w}_k, \bar{y})$
$\quad$ stop if $\|w_k\|$ and $\|\bar{z}_k\|$ are small
$\quad z_{k+1} = z_k - P_k\, w_k$
$\quad \bar{w}_{k+1} = \bar{w}_k - \bar{z}_k\, P_k$
$y_k = f(z_k, x), \;\; \sigma_k = \bar{y}\, y_k + \bar{w}_k\, w_k$
$\bar{x}_k = \bar{w}_k\, F_x(z_k, x) + \bar{y}\, f_x(z_k, x)$

Adjoint Fixed Point Iteration

Except for the omission of the term involving $\dot{P}_k$ and the suggested modification of the stopping criterion, the scheme listed in Table 2 could have been obtained by simply differentiating the original fixed point iteration in the forward mode. This black box approach often yields virtually identical results and it alleviates the need for any code modification by hand. Unfortunately, things are not nearly as convenient in the reverse mode.

Suppose one has a code for executing the update (27) a certain number of T times and subsequently evaluating the response function $y = f(z, x)$ at the final z. If one then applies an adjoint generator nominating x as independent and y as dependent variables the resulting code will have some pretty undesirable features. The main crux is that it will save all intermediate states on the way forward, which means a T-fold increase in memory relative to the original fixed point iteration. The reason for this apparent waste of storage is that AD tools cannot know (and usually cannot be told either) whether an iteration represents a genuine evolution whose complete trajectory is important for the adjoints to be calculated, or whether the early stages are only of passing interest as the trajectory homes in on a fixed point later. In the latter case, which is applicable here, the return sweep of the reverse mode regresses in every sense of the word from good information in the vicinity of the solution point to much earlier iterates where function and derivative values have little to do with the desired implicit derivatives at the limit. Moreover, in contrast to the constructive tests given in Proposition 1, we have no way of gaging the quality of the approximation $\bar{x} \approx \bar{x}_*$ that finally pops out of the black box adjoint procedure. Even if we could test its quality, there would be no apparent way of refining the approximation other than by rerunning both sweeps with an increased T. Some of these arguments against mechanical adjoining can be made even in the linear case as was done in [Gil00]. As the reader might suspect by now this gloomy description only sets the stage for the following enlightenment.

To compute $\bar{w}_*$ one must somehow solve the adjoint sensitivity equation

$$F_z(z_*, x)^T \bar{w}^T \;=\; -f_z(z_*, x)^T \bar{y}^T \;=\; \bar{F}(z_*, x, 0, -\bar{y})^T \tag{36}$$

obtained by transposing (16). The transposed Jacobian $F_z(z_*, x)^T$ has the same size, spectrum, and sparsity characteristics as $F_z(z_*, x)$ itself. Hence the task of solving the adjoint sensitivity equation (36) is almost exactly equivalent to the task of solving the direct sensitivity, equation (15). Because of the similarity relation

$$P_k^T F_z(z_k, x)^T \;=\; P_k^T \left[P_k F_z(z_k, x)\right]^T P_k^{-T} \tag{37}$$

the square matrices $I - P_k F_z(z_k, x)$ and $I - P_k^T F_z(z_k, x)^T$ have the same spectrum. Hence the latter has by Assumption **CP** a spectral norm less or equal to $\rho < 1$ and we may solve the adjoint sensitivity equation (36) by the iteration

$$\bar{w}_{k+1}^T \;=\; \bar{w}_k^T - P_k^T \left[F_z(z_k, x)^T \bar{w}_k^T + f_z(z_k, x)^T \bar{y}^T\right] \;=\; \bar{w}_k^T - P_k^T \bar{F}(z_k, x, \bar{w}_k, \bar{y})^T \,. \tag{38}$$

where $\bar{F}$ is defined as in (16). The recurrence (38) was apparently first analyzed by Christianson [Chr94], albeit with a fixed final preconditioner P_k.

Now the question arises whether the adjoint sensitivity calculations can also be performed in a piggyback fashion, i.e. without setting up a second phase iteration. This means that we can propagate the adjoint vectors $\bar{w}_k$ forward without the need to record the intermediates z_k and the corresponding preconditioners. Only each coupled evaluation $\left[F(z_k, x), f(z_k, x)\right]$ must be reversed to yield the adjoint residual $\bar{F}(z_k, x, \bar{w}_k, \bar{y}) \in \mathbb{R}^l$ at a comparable computational effort. However, it should be noted that the response function f must now be evaluated at each iterate rather than

just at the end. This extra effort is typically quite small and sometimes required anyway for the adjustment of boundary conditions [FE00]. The size of the resulting adjoint residual $\bar{F}$ should be included in the overall stopping criterion, which yields the iteration displayed in Table 3. For a real code this iteration can be obtained by applying an adjoint generator only to the body of the loop without reversing the loops order of execution. In this transformation the design parameters x should be declared as passive with respect to differentiation or otherwise the reduced gradient $\bar{x}_k$ must be reset to zero before it is actually computed after exit from the loop.

Since the transpose $I - F_z(z_k)^T P_k^T$ is the Jacobian of the adjoint fixed point iteration for $\bar{w}_k$ it has the same contraction factor ρ_0 and we obtain the following Corollary of Proposition 2.

Corollary 3 (Root Convergence Factor of Adjoint Derivatives) *Under Assumptions* **CP** *and* **UL** *with F twice Lipschitz continuously differentiable we obtain for all z_0 sufficiently close to z_* and arbitrary $\bar{w}_0$ by applying the iteration of Table 3 infinitely often*

$$R\{\bar{w}_k - \bar{w}_*\}_{k \in \mathbb{N}} \;\le\; \bar{\rho} \equiv \max\left(\rho_*, R\{\bar{F}(z_k, x, \bar{w}_k, \bar{y})\}_{k \in \mathbb{N}}\right) \;\le\; \rho_o$$

□

It must be stressed that contrary to what one might expect the vectors $\bar{w}_k$ and $\bar{z}_k$ are not the adjoints of the intermediate values w_k and z_k in the usual sense. The concept of an adjoint is normally only defined for evaluation procedures that involve an a priori fixed sequence of elemental operations. However, in the linear case discussed in Section 6 there is an interpretation of the barred quantities as adjoints in a more conventional sense.

It is important to note that the adjoint evaluation yielding $\bar{z}_k$ and thus $\bar{w}_{k+1}$ immediately follows a corresponding forward calculation, so that taping is only required temporarily. What we call here *adjoint fixed point iteration* has been referred to as *iterative incremental form* of the adjoint sensitivity equation in the aerodynamic literature [NHJ+92].

Second Order Adjoint Fixed Point Iteration

In order to obtain second order adjoints $\dot{\bar{w}}_k$ occurring in (14), (17), and (24) we simply have to differentiate the adjoint fixed point iteration once more, this time in the forward mode as displayed in Table 4.

As one can see Table 4 contains essentially the union of Table 2 and Table 3 plus four statements and the convergence test involving $\dot{\bar{w}}$. For the numerical results reported in Section 7 it was obtained by applying the source transformation tool Odyssée a second time in direct, or tangent, mode to the code representing the adjoint fixed point iteration listed in Table 3. It would probably be harder to generate a code that involves only the first order quantities $\dot{z}_k$, $\bar{w}_k$ and the corresponding residuals $\dot{w}_k, \bar{z}_k$ but not $\dot{\bar{w}}_k$ and $\dot{\bar{z}}_k$. Since the transpose $I - F_z(z_k)^T P_k^T$ is also the Jacobian of the fixed point iterations for $\dot{\bar{w}}_k$ its has again the same contraction factor and we obtain the following Corollary of Corollary 2.

Corollary 4 (Root Convergence Factor of Second Order Adjoints) *Under Assumptions* **CP** *and* **UL** *with F twice Lipschitz continuously differentiable we have for all z_0 sufficiently close to z_* and arbitrary $\dot{\bar{w}}_0$ the iteration of Table 4 infinitely often*

$$R\{\dot{\bar{w}}_k - \dot{\bar{w}}_*\}_{k \in \mathbb{N}} \;\le\; \dot{\bar{\rho}} \equiv \max\left(\dot{\rho}, \bar{\rho}, R\{\dot{\bar{F}}(z_k, x, \dot{z}_k, \dot{x}, \bar{w}_k, \bar{y}, \dot{\bar{w}}_k)\}_{k \in \mathbb{N}}\right) \;\le\; \rho_o \quad . \tag{39}$$

□

Table 4: Second Order Adjoint Fixed Point Iteration

fix $x, \dot{x} \in \mathbb{R}^n, \bar{y} \in \mathbb{R}^m$
initialize $z_0, \dot{z}_0, \bar{w}_0, \dot{\bar{w}}_0 \in \mathbb{R}^l$
for $k = 0, 1, 2, \ldots$
$\quad [w_k, y_k] = [F(z_k, x), f(z_k, x)]$
$\quad \dot{w}_k = \dot{F}(z_k, x, \dot{z}_k, \dot{x})$
$\quad \bar{z}_k = \bar{F}(z_k, x, \bar{w}_k, \bar{y})$
$\quad \dot{\bar{z}}_k = \dot{\bar{F}}(z_k, x, \dot{z}_k, \dot{x}, \bar{w}_k, \bar{y}, \dot{\bar{w}}_k)$
$\quad$ stop if $\|w_k\|, \|\dot{w}_k\|, \|\bar{z}_k\|$ and $\|\dot{\bar{z}}_k\|$ are small
$\quad z_{k+1} = z_k - P_k\, w_k \qquad \dot{z}_{k+1} = \dot{z}_k - P_k\, \dot{w}_k$
$\quad \bar{w}_{k+1} = \bar{w}_k - \bar{z}_k\, P_k \qquad \dot{\bar{w}}_{k+1} = \dot{\bar{w}}_k - \dot{\bar{z}}_k\, P_k$
$y_k = f(z_k, x) \qquad \sigma_k = \bar{y}\, y_k + \bar{w}_k\, w_k$
$\dot{y}_k = \dot{f}_z(z_k, x, \dot{z}_k, \dot{x}) \quad \dot{\sigma}_k = \bar{y}\, \dot{y}_k + \bar{w}_k \dot{w}_k + \dot{\bar{w}}_k\, w_k$
$\bar{x}_k = \bar{w}_k\, F_x(z_k, x) + \bar{y}\, f_x(z_k, x)$
$\dot{\bar{x}}_k = \dot{\bar{w}}_k F_x(z_k, x) + \bar{w}_k \dot{F}_x(z_k, x, \dot{z}_k, \dot{x}) + \bar{y}\, \dot{f}_x(z_k, x, \dot{z}_k, \dot{x})$

Substituting Proposition 2 and its two Corollaries into Proposition 1 and its two Corollaries we obtain the following list of R-linear convergence results

$$
\begin{aligned}
z_k - z_* &= \mathcal{O}(\|F(z_k)\|) &&= \tilde{\mathcal{O}}(\rho_*^k) \\
y_k - y_* &= \mathcal{O}(\|F(z_k)\|) &&= \tilde{\mathcal{O}}(\rho_*^k) \\
\sigma_k - \bar{y}\, y_* &= \mathcal{O}(\|F(z_k)\|\, \|\bar{F}(z_k)\| + \|F(z_k)\|^2) &&= \tilde{\mathcal{O}}(\rho_*^k \rho_o^k) \\
\bar{w}_k - \bar{w}_* &= \mathcal{O}(\|F(z_k)\| + \|\bar{F}(z_k, \bar{w}_k)\|) &&= \tilde{\mathcal{O}}(\rho_o^k) \\
\bar{x}_k - \bar{x}_* &= \mathcal{O}(\|F(z_k)\| + \|\bar{F}(z_k, \bar{w}_k)\|) &&= \tilde{\mathcal{O}}(\rho_o^k) \\
\dot{z}_k - \dot{z}_* &= \mathcal{O}(\|F(z_k)\| + \|\dot{F}(z_k, \dot{z}_k)\|) &&= \tilde{\mathcal{O}}(\rho_o^k) \\
\dot{y}_k - \dot{y}_* &= \mathcal{O}(\|F(z_k)\| + \|\dot{F}(z_k, \dot{z}_k)\|) &&= \tilde{\mathcal{O}}(\rho_o^k) \\
\dot{\sigma}_k - \bar{y}\, \dot{y}_* &= \mathcal{O}\big(\|F(z_k)\| + \|\dot{F}(z_k, \dot{z}_k)\| + \|\bar{F}(z_k, \bar{w}_k)\| + \|\dot{\bar{F}}(z_k, \dot{z}_k, \bar{w}_k, \dot{\bar{w}}_k)\|\big)^2 &&= \tilde{\mathcal{O}}(\rho_o^{2k}) \\
\dot{\bar{x}}_k - \dot{\bar{x}}_* &= \mathcal{O}(\|F(z_k)\| + \|\dot{F}(z_k, \dot{z}_k)\| + \|\bar{F}(z_k, \bar{w}_k)\| + \|\dot{\bar{F}}(z_k, \dot{z}_k, \bar{w}_k, \dot{\bar{w}}_k)\|) &&= \tilde{\mathcal{O}}(\rho_o^k) \quad .
\end{aligned}
$$

Here we have again omitted the constant vectors $x, \dot{x}$ and $\bar{y}$ as arguments.

Extended Computational Graph

In Figure 1 we have displayed the dependence relations between the various vector quantities in our iteration loops. The oval conditionals $= 0$? indicate that the input vectors coming from the right are checked for size in a suitable norm. If they are sufficiently small the iteration is terminated; otherwise a suitable multiple of the residual vector is incremented to the node values on the left. The right half displays the original state space iteration with directional derivatives being carried along as well. Here, as indicated by the vertical arrows, the vectors x and $\dot{x}$ are given inputs,

whereas y and $\dot{y}$ are the resulting outputs. They may be (re-)calculated throughout or only after the state space iteration has terminated. The left half of the graph represents the adjoint quantities as mirror images of the direct ones. The dotted lines represent the dependence of the corrected response σ and its derivative $\dot{\sigma}$ on both direct and adjoint quantities.

Unless all iterations are indeed globally contractive one may reach different limit points depending on the initialization of z and $\dot{z}$. For simplicity we may assume without too much loss of generality that the state vectors z and $\dot{z}$ are initialized to zero. Then we can distinguish between four kinds of variables and the corresponding graph nodes.

$$\begin{array}{ccc} \text{independents} & \longleftrightarrow & \text{dependents} \\ \text{solutions} & \longleftrightarrow & \text{residuals} \end{array}$$

The solution variables (here z and $\dot{z}$) are initially zero whereas the corresponding residual variables (here w and $\dot{w}$) are supposed to be zero at the end. Moreover, as we have suggested graphically there is a nice duality relationship with the corresponding adjoint nodes in the left half of Figure 1. Independent nodes have dependent adjoints and vice versa. Similarly, solution nodes have residual adjoints and vise versa. This symmetry about the vertical center line is aesthetically very pleasing.

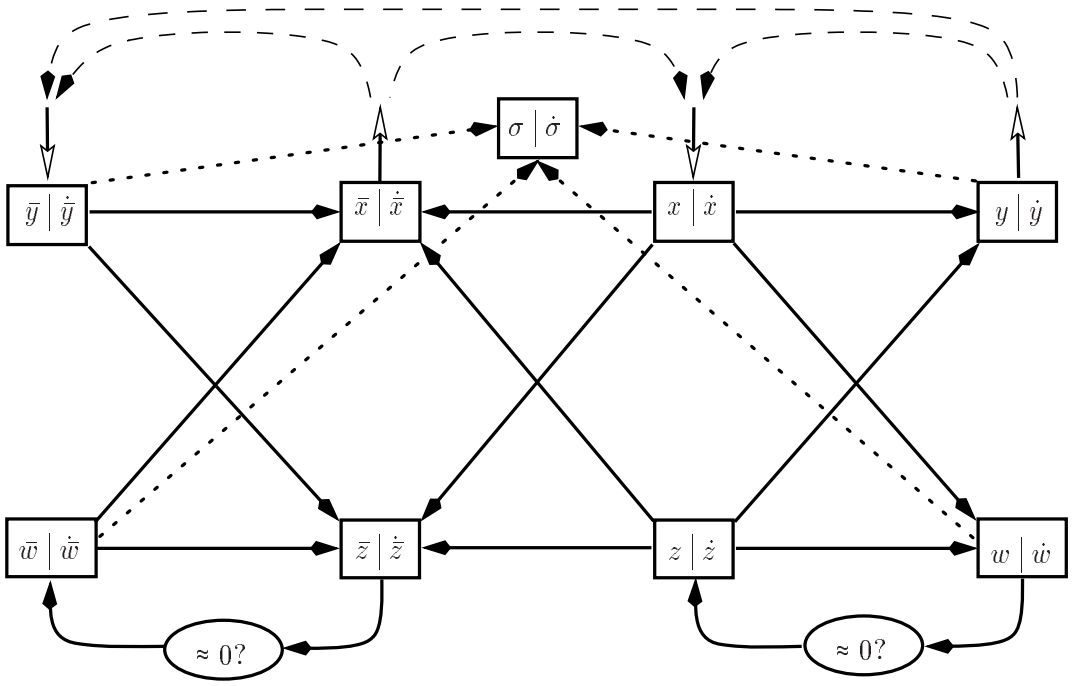

Figure 1: Direct and Adjoint Fixed Point Iterations with Directional Derivatives

Since the adjoint iteration on the left is linear and F_z by assumption regular the adjoint limit values of $(\bar{w}, \dot{\bar{w}})$ are unique. Moreover, these convergence results remain true irrespective of the order in which the updates of the primal and dual variables are carried out. A standard *two-phase* approach is to first iterate the primal variables $(z, \dot{z})$ to convergence with full accuracy and only then to initiate the adjoint iteration. Alternatively one might prefer a "piggyback" approach, where the adjoint iteration is performed simultaneously as embodied in the programs listed in Tables 3 and 4. At least on our example Euler code the adjoint iteration does not seem degraded at all by the early perturbations in the piggyback approach.

Possible Uses in One Shot Optimization

The dashed lines on top of Figure 1 represent data dependencies that arise when the design variables x, the prospective search direction $\dot{x}$ and the multipliers $\bar{y}$ are adjusted in view of the current approximations of the reduced function y, the reduced gradient $\bar{x}$ and their directional derivatives $\dot{y}$ and $\dot{\bar{x}}$. In particular $\bar{y}$ may be a vector of Lagrange multipliers that is dynamically selected by a constrained optimization algorithm. Within the aerodynamic design community specific suggestions for adjusting x simultaneously in a one-shot manner have been used with success in [Jam95] and [TKS92]. As far as we are aware of the use of second derivatives have not been suggested in this context.

As we have mentioned earlier, it is a difficult question to decide how much feasibility in z one should recover keeping x constant via the fixed point iteration, before selecting another optimization step in x. Without strong convexity assumptions that preclude the existence of local minima or other stationary points the result of the design iteration may strongly depend on initializations.

6 Special Relations in the Linear Case

Suppose the state equation $F = 0$ and the response function f are linear with the latter not depending explicitly on x. For fixed $\bar{y}$ and $\dot{x}$ we may furthermore replace f with $\bar{y}f$ and restrict x also to be scalar so that effectively $\bar{y} = 1 = \dot{x}$. Hence we can write without loss of generality

$$F(z,x) \;=\; Az + b\,x \quad \text{and} \quad f(z) \;=\; c\,z \quad , \tag{40}$$

so that $z(x) \;=\; -A^{-1}b\,x$ and $y(x) \;=\; -c\,A^{-1}b\,x$, where $b, c \in \mathbb{R}^l$ and $A \in \mathbb{R}^{l\times l}$ are constant. The sensitivity equations have the explicit solutions

$$\dot{z}_* \;=\; -A^{-1}b \quad , \quad \bar{w}_* \;=\; -cA^{-1} \quad \text{and} \quad \dot{\bar{w}}_* \;=\; 0\,,$$

where the last identity follows from the fact that second derivatives of linear functions vanish identically. Assuming finally that the preconditioners $P_k \equiv P \in \mathbb{R}^{l\times l}$ are the same at each iteration the recurrences for the z_k, $\dot{z}_k$, and $\bar{w}_k$ reduce to

$$\begin{aligned} z_{k+1} \;&=\; z_k - P(Az_k + bx)\,, \quad \dot{z}_{k+1} \;=\; \dot{z}_k - P(A\dot{z}_k + b)\,, \\ \bar{w}_{k+1} \;&=\; \bar{w}_k - (\bar{w}_k A + c)P\,, \quad \dot{\bar{w}}_{k+1} \;=\; \dot{\bar{w}}_k - \dot{\bar{w}}_k AP\,. \end{aligned} \tag{41}$$

Starting from $z_0 = 0 = \dot{z}_0$ and $\bar{w}_0 = 0 = \dot{\bar{w}}_0$ one can easily check by induction that

$$\begin{aligned} z_k \;&=\; -\sum_{j=0}^{k-1}(I - PA)^j P\,bx\,, \quad \dot{z}_k \;=\; -\sum_{j=0}^{k-1}(I - PA)^j P\,b\,, \\ \bar{w}_k \;&=\; -c\,P\sum_{j=0}^{k-1}(I - AP)^j\,, \quad \dot{\bar{w}}_k \;=\; 0\,. \end{aligned} \tag{42}$$

Our contractivity condition **CP** requires exactly that the common spectral radius ρ_o of $I - PA$ and $I - AP$ is less than one, so that we have the convergent Neumann series

$$\sum_{j=0}^{\infty}(I - PA)^j P \;=\; (PA)^{-1}P \;=\; A^{-1} \;= P(AP)^{-1} \;=\; P\sum_{j=0}^{\infty}(I - PA)^j\,.$$

Hence we see that

$$z_k \;=\; -A^{-1}bx + \mathcal{O}(\rho_o^k)\,, \quad \dot{z}_k \;=\; -A^{-1}b + \mathcal{O}(\rho_o^k)\,, \quad \text{and} \quad \bar{w}_k \;=\; -c\,A^{-1} + \mathcal{O}(\rho_o^k)\,.$$

Whereas this rate of convergence applies also in the general, nonlinear case the following exact identities are specific to the linear scenario.

Proposition 3 (Identities between Estimates)
For the linear system (40) we obtain exactly

$$\begin{aligned} y_{2k} \;&=\; c\,z_{2k} \;&=\; \sigma_k \;&=\; y_* + \mathcal{O}(\rho_o^{2k}) \\ \dot{y}_{2k} \;&=\; c\,\dot{z}_{2k} \;=\; \bar{w}_{2k}\,b \;&=\; \dot{\sigma}_k \;&=\; \dot{y}_* + \mathcal{O}(\rho_o^{2k}) \end{aligned}$$

where

$$\sigma_k \;=\; c\,z_k + \bar{w}_k\,w_k \quad \textit{and} \quad \dot{\sigma}_k \;=\; c\,\dot{z}_k + \bar{w}_k\,\dot{w}_k \quad \textit{since} \quad \dot{\bar{w}}_k = 0 \quad . \qquad \square$$

Proof As a consequence of (42) we have

$$\begin{aligned} c\,z_{2k} \;&=\; -c\sum_{j=0}^{2k-1}(I-PA)^j Pbx \;=\; c\left[I+(I-P\,A)^k\right]\left[-\sum_{j=0}^{k-1}(I-PA)^j P\,b\,x\right] \\ &=\; c\,z_k - c\sum_{j=0}^{k-1}(I-P\,A)^j(I-P\,A)^k P\,b\,x \;=\; c\,z_k - c\,P\sum_{j=0}^{k-1}(I-A\,P)^j(I-AP)^k\,b\,x \\ &=\; c\,z_k + \bar{w}_k\,w_k \end{aligned}$$

as $w_k = (I-AP)^k bx$. In order to prove the second assertion we note first that

$$c\,\dot{z}_k \;=\; -c\sum_{j=0}^{k-1}(I-PA)^j P\,b \;=\; -c\,P\sum_{j=0}^{k-1}(I-AP)^j b \;=\; \bar{w}_k b \quad .$$

Simply substituting the explicit expressions for $\dot{z}_k$ and $\bar{w}_k$ into the corrected estimate we obtain with $F \equiv 0$ and the above identity of the first order estimates

$$\begin{aligned} \sigma_k \;&=\; [2\,c + \bar{w}_k A]\,\dot{z}_k \;=\; \left[c - (\bar{w}_{k+1} - \bar{w}_k)P^{-1}\right]\dot{z}_k \;=\; \left[c + c\,P(I-AP)^k P^{-1}\right]\dot{z}_k \\ &=\; c\left[I+(I-PA)^k\right]\left[-\sum_{j=0}^{k-1}(I-AP)^j b\right] \;=\; -c\left[\sum_{j=0}^{2k-1}(I-AP)^j\right] b \;=\; c\,\dot{z}_{2k} \;=\; \bar{w}_{2k}\,b \quad \blacksquare \end{aligned}$$

Thus we see that after k iterations the corrected function and derivative estimates have already reached exactly the values that the simple estimates will only reach after $2k$ iterations. These identity relations do crucially depend on the constancy of the preconditioners as they are already violated for $k = 2$ when $P_0 \neq P_1$. Nevertheless the asserted orders are of course attained under our general assumptions for the nonlinear case. Moreover, in the numerical results reported in Section 7 the estimates for the response function and its derivatives do in fact very closely exhibit the doubling pattern established here under the assumption of linearity and constant preconditioning.

Interpretation of $\bar{z}_k$ and $\bar{w}_k$ as Adjoints

In the linear case we can interpret the iterates $\bar{z}_k$ directly as adjoints in the conventional sense. More specifically one can easily derive from (41) that for all $i \leq k$

$$(z_k - z_*) \;=\; (I - PA)^{k-i}\,(z_i - z_*)$$

and

$$\bar{z}_{k-i} \;=\; c + \bar{w}_{k-i}A \;=\; c\,(I - PA)^{k-i} \quad .$$

Hence we find for $\bar{y}\,y_k = c\,z_k$ that

$$\tilde{z}_i \;\equiv\; \frac{\partial}{\partial z_i}\bar{y}\,y_k \;=\; \bar{z}_{k-i} \tag{43}$$

As $\bar{z}_k \to 0$ this reflects the fact that the influence of an earlier state z_i on a newer state z_k wanes as $k-i$ the number of fixed point iteration in between grows towards infinity. For the nonvanishing $\bar{w}_k$ a corresponding interpretation seems to require a small rewrite of our original recurrence. Namely the residual evaluation $w_k = Az_k + b$ must be replaced by the mathematically equivalent recurrence

$$z_{i+1} -= Pw_i\;, \quad w_{i+1} \;=\; w_i + A(z_{i+1} - z_i) \tag{44}$$

starting from $w_o = b + Az_0$ with arbitrary $z_0 \in \mathbb{R}^l$. It yields for the true adjoint $\tilde{w}_i \equiv \frac{\partial}{\partial w_i}\bar{y}\,y_k$ with $\tilde{w}_k = 0$ the backward recurrence

$$\begin{aligned}\tilde{w}_i \;&=\; \tilde{w}_{i+1} - \tilde{z}_{i+1}P \;=\; -\sum_{j=i}^{k}\tilde{z}_j P \;=\; -\sum_{j=i}^{k}\bar{z}_{k-j}P \\ &=\; -\sum_{j=0}^{k-i}\bar{z}_j P \;=\; -c\sum_{j=0}^{k-i}(I - PA)^{j-1}\,P \;=\; \bar{w}_{k-i} \quad .\end{aligned}$$

Hence observe that similarly to (43)

$$\tilde{w}_i \;=\; \frac{\partial}{\partial z_i}\bar{y}\,y_k \;=\; \bar{w}_{k-i} \quad .$$

The influence of w_i on z_k and thus $\bar{y}\,y_k = c\,z_k$ does not tend to zero because of the lasting impact that w_i has on all w_k for $k > i$ in the incremental form (44). Especially in the nonlinear case, it is probably most appropriate to view the $\dot{z}_k$ $\bar{w}_k$ as iterates in their own right.

7 Numerical Results

The following results were obtained on a 2D Euler code written in Fortran 77 and provided by Vittorio Selmin from Alenia, Torino. The configuration considered is the standard test case of the NACA0012 airfoil with a structured computational mesh of 1385 nodes with about 23 nodes on the skin. The "design" vector $x \in \mathbb{R}^n$ with $n = 2$ consists of the angle of attack and the free stream velocity. They are set to the values $x_1 = 1°$ and $x_2 = M_\infty = 0.8$, respectively. The response vector $y \in \mathbb{R}^m$ with $m = 2$ considered consists solely of the lift coefficient y_1 and the drag coefficient y_2. Correspondingly both the direction $\dot{x}$ and the weight vectors $\bar{y}$ were always chosen as one of the Cartesian basis vectors in $\mathbb{R}^2$. Hence there are in fact four different seeding possibilities, which yield in particular the reduced 2×2 Jacobian with normal and double accuracy as well as the two

reduced Hessians of both lift and drag with respect to angle of attack and free-stream velocity. The code was differentiated repeatedly by Odyssée, which required some modifications by hand to obtain the nonstandard adjoint code with directional derivatives displayed in Table 4 in Section 5. The derivative vectors $\dot{z}, \bar{w}, \dot{\bar{w}}$ were always initialized to zero and the second order weighting vector $\dot{\bar{y}}$ was kept zero throughout.

First to verify Proposition 2 and its Corollaries we monitored the norms of the *state equation residual* F the *direct derivative residual* $\dot{F}$ the *adjoint derivative residual* $\bar{F}$ and the *second order residual* $\dot{\bar{F}}$ as defined in equations (1), (15), (16), and (17) respectively. The logarithms of these norms divided by their initial values were plotted against the iterations counter in Figure 2. As one can see by the four slopes all of them converge asymptotically with a similar rate as predicted by the theory. Also it is noticeable that the second order residual trails behind the other three and the state equation is a little bit ahead as was to be expected in view of the theory in Section 5.

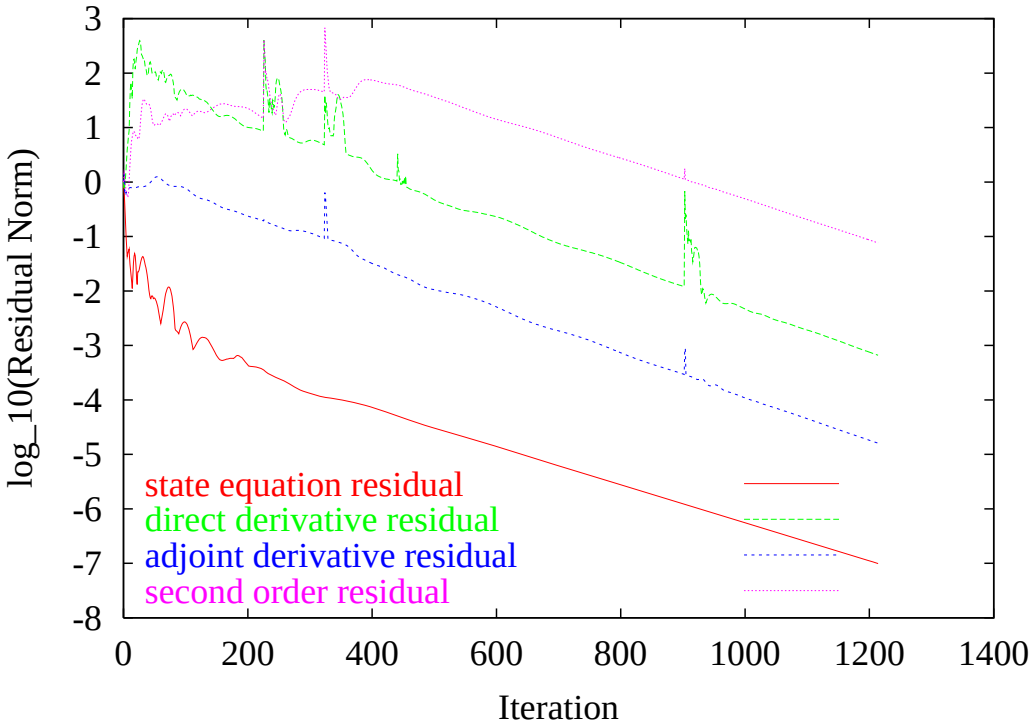

Figure 2: History of Residual Norms F, $\dot{F}$, $\bar{F}$, and $\dot{\bar{F}}$ for $\bar{y} = e_1$ and $\dot{x} = e_2$, i.e. lift coefficient and Mach number

In our piggyback approach the iterations for $\dot{z}_k$ and $\bar{w}_k$ are initially affected by errors in the state z_k itself and this perturbation effect should be even more marked for the second order adjoints $\dot{\bar{w}}_k$. To assess to what extent this effect slowed down their convergence we conducted the following experiment. After 800 iterations when the state equation residual had already been reduced by a factor of about 10^5 the derivative vectors $\dot{z}_k$, $\bar{w}_k$ and $\dot{\bar{w}}_k$ were reset to zero. In other words as far as the derivatives are concerned we start from a point where the state equation is almost exactly solved so that subsequent variations in the state approximation z_k are rather small. In effect this way of computing sensitivities amounts to the two-phase philosophy mentioned just before Proposition 1. As one can see on the right half of Figure 3, the resulting derivative residual histories are rather similar to the ones obtained right from the start. Some peaks have been eliminated but the overall speed of convergence looks the same. Hence we can conclude that at least on this test problem the

piggyback approach does not slow down convergence at all.

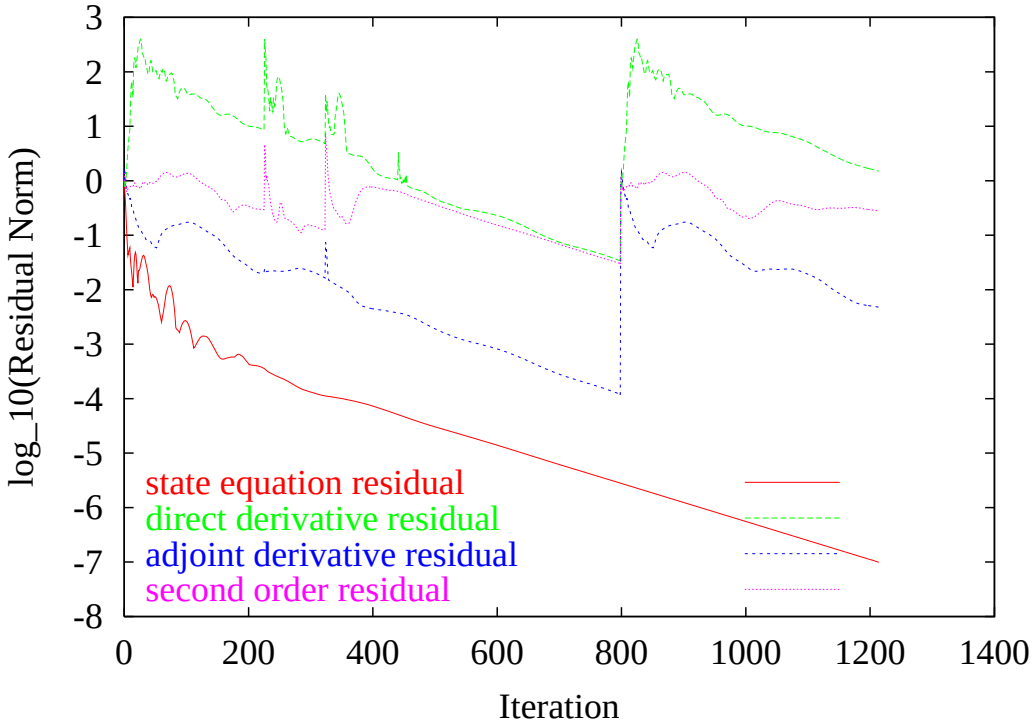

Figure 3: History of Residual Norms for $\bar{y}_1 = e_2$ and $\dot{x} = e_2$, i.e. drag coefficient and angle of attack with reset of $\dot{z}$, $\bar{w}$, and $\dot{\bar{w}}$ to zero after 800 iterations

Moreover as we can see by carrying along adjoint derivatives right from the start we obtain a much improved estimate for the reduced response function as displayed in Figure 4. The curve labeled *double* represents the corrected function value estimate, which converges indeed significantly faster and arrives after some 200 iterations an approximate value that should already be close enough to decide whether the current design parameter settings are competitive or whether this configuration should be rejected. On closer inspection of the distinctive maxima and minima one finds that they occur for the corrected estimate at roughly half the number of iterations for which they occur for the normal estimate. To illustrate this hypothesis we made the scaling of the iterations logarithmic, which shows quite clearly that the two curves are shifted exactly by $1 = \log_2(2)$.

The same desirable situation prevails for the reduced response derivatives as displayed in Figure 5. These observations agree with the relations we derived in Proposition 3 for the special case of a linear system with constant preconditioner. Hence we may expect that for certain smooth nonlinear systems and the resulting iterations the carrying along of first [and possibly second] order adjoint information yields the same response function and derivative estimates obtainable by the original iteration [with directional derivatives] at half the number of iterations. Moreover without any significant extra effort we obtain the full reduced gradient $\bar{x} \equiv \bar{y}\varphi'$ [and the Hessian vector product $\dot{\bar{x}} \equiv \bar{y}\,\varphi''$]. The statements in brackets apply only if the program in Table 4 rather than the simple adjoint in Table 3 is executed.

The reduced gradient approximations of the lift coefficient y_1 and the drag coefficient y_2 with respect to the angle of attack x_1 and the free-stream velocity x_2 after 256, 512 and 1024 iterations are listed in the following Table 5.

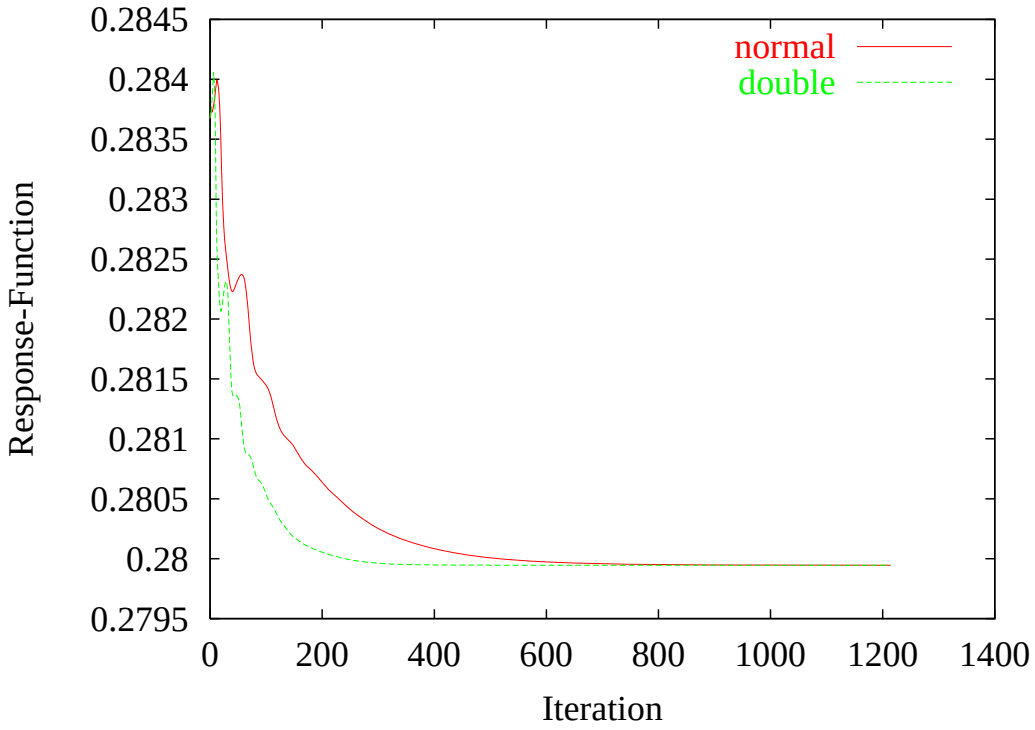

Figure 4: Normal and corrected Response Function estimate for $\bar{y} = e_2$

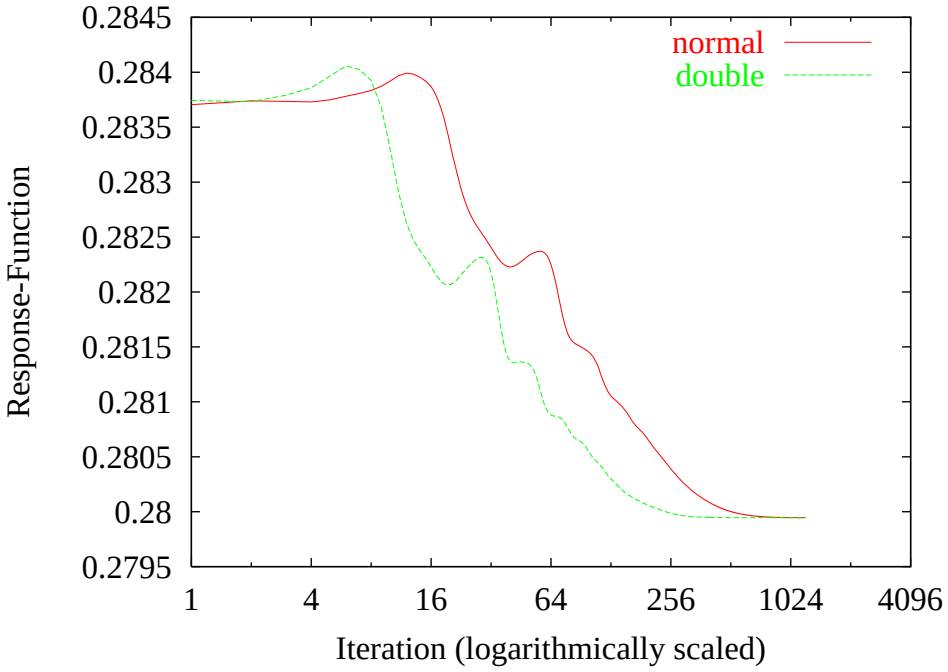

Figure 5: Normal and corrected Response Function estimate for $\bar{y} = e_2$

The entries of Hessians of the drag coefficient y_2 with respect to x_1 and x_2 converge as displayed in Table 6.

Table 5: Reduced Jacobian Entries after 256, 512, and 1024 Iterations

	$\partial y_1/dx_1$	$\partial y_1/\partial x_2$	$\partial y_2/\partial x_1$	$\partial y_2/\partial x_1$
256	2.03469	16.11593	0.51226	1.01937
512	2.03846	16.12873	0.51255	1.01997
1024	2.04191	16.14435	0.51272	1.02073

Table 6: Reduced Hessians Entries after 256, 512, and 1024 Iterations

	$\partial^2 y_2/\partial x_1^2$	$\partial^2 y_2/\partial x_1 \partial x_2$	$\partial^2 y_2/\partial x_2 \partial x_1$	$\partial^2 y_2/\partial x_2^2$
256	31.58015	−6.42395	−5.11607	−5.91265
512	21.94557	−14.98488	−14.45852	−10.25206
1024	20.92140	−15.36294	−15.35948	−10.50083

The entries in the second and third columns of Table 6 should be the same if the reduced Hessians was evaluated exactly. Their discrepancies give an indication of how closely the exact values have been approximated and whether the whole code has been correctly differentiated according to our recipes in the first place.

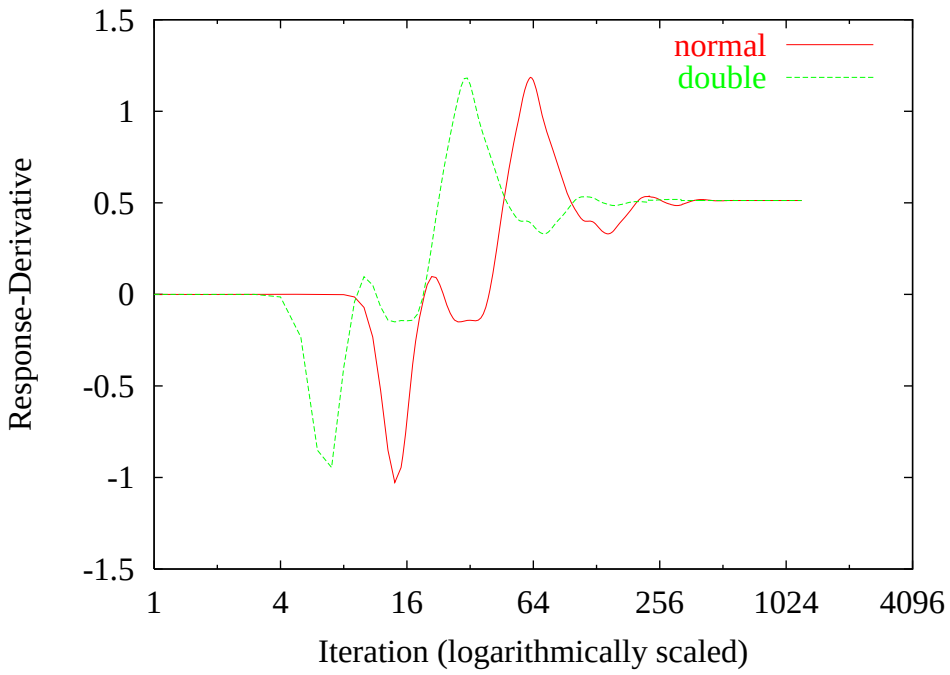

Figure 6: Normal and corrected Estimate of drag Coefficient ($\bar{y} = e_1$) derivative w.r.t. angle of attack

8 Summary and Conclusion

On the basis of a given fixed point iteration for solving a state equation $F(z,x)=0$ with respect to z we wish to calculate rapidly converging estimates for the reduced function $f(z(x),x)$ and its total derivatives with respect to x. This reduced gradient and a corrected version of the reduced function $\varphi(x)=f(z(x),x)$ is obtained from the results of an adjoint fixed point iteration that can be grafted onto the original user supplied solver. The corrected function estimate due to Christianson [Chr98] is shown to converge twice as fast as the underlying fixed point iterations.

The same is true for the corrected estimates of reduced first derivatives, which are obtained by a combination of adjoint and direct differentiation. The latter process generates feasible state space directions, which yield also in combination with the Lagrange multipliers generated by the adjoint differentiation reduced Hessians and other second derivative information. The algorithmic development and convergence analysis was confirmed by test calculations on a $2D$ Euler code. The rapid availability of accurate estimates for the reduced response function and its derivatives should facilitate the development of piggy-back design optimization methods that achieve optimality in addition to feasibility at little extra cost.

The question whether and when derivatives should be carried along with the state iterates themselves in a piggyback fashion cannot be answered in general. The same is certainly true for the evaluation of reduced second derivatives, which are considerably more expensive but may sometimes reduce the number of iterations considerably.

9 Acknowledgement

The authors are indebted to Vittorio Selmin, Alenia for supplying the Euler code and Michael Giles, University of Oxford, for many corrections and suggestions on the basis of an early draft.

References

[Chr94] B. Christianson, *Reverse accumulation and attractive fixed points*, Optim. Methods Softw. **3** (1994), 311–326.

[Chr98] B. Christianson, *Reverse accumulation and implicit functions*, Optim. Methods Softw. **9** (1998), 307–322.

[DS96] J.E. Dennis, Jr. and R.B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, Classics Appl. Math., no. 16, SIAM, Philadelphia, 1996.

[FE00] S.A. Forth and T.P. Evans, *Aerofoil Optimisation via Automatic Differentiation of a Multigrid Cell-Vertex Euler Flow Solver*, Proceedings of Automatic Differentiation 2000: From Simulation to Optimization (Berlin), Springer Verlag, 2000.

[GBC+93] A. Griewank, C. Bischof, G. Corliss, A. Carle, and K. Williamson, *Derivative convergence of iterative equation solvers*, Optimiz. Methods Softw. **2** (1993), 321–355.

[Gil92] J.Ch. Gilbert, *Automatic differentiation and iterative processes*, Optim. Methods Softw. **1** (1992), 13–21.

[Gil00] M.B. Giles, *On the iterative solution of adjoint equations*, Proceedings of Automatic Differentiation 2000: From Simulation to Optimization (Berlin), Springer Verlag, 2000.

[Gri80] A. Griewank, *Starlike domains of convergence for Newton's method at singularities*, Numer. Math. **35** (1980), 95–111.

[Gri00] A. Griewank, *Evaluating Derivatives, Principles and Techniques of Algorithmic Differentiation*, Frontiers in Appl. Math., no. 19, SIAM, Philadelphia, 2000.

[Jam95] A. Jameson, *Optimum aerdynamic design using cfd and control theory*, 12th AIAA Computational Fluid Dynamics Conference, AIAA Paper 95-1729 (San Diego, CA), American Institute of Aeronautics and Astronautics, 1995.

[NHJ+92] P.A. Newman, G.J.-W. Hou, H.E. Jones, A.C. Taylor, and V.M. Korivi, *Observations on computational methodologies for use in large-scale, gradient-based, multidisciplinary design incorporating advanced CFD codes*, Technical Memorandum 104206, NASA Langley Research Center, February 1992, AVSCOM Technical Report 92-B-007.

[NS96] S.G. Nash and A. Sofer, *Linear and nonlinear programming*, McGraw-Hill Series in Industrial Engeneering and Management Science, McGraw-Hill, New York, 1996.

[OR70] J.M. Ortega and W.C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*, Academic Press, New York, 1970.

[PG00] N.A. Pierce and M.B. Giles, *Adjoint recovery of superconvergent functionals from PDE approximations*, SIAM Review **42** (2000), no. 2, 247–264.

[TKS92] S. Ta'asan, G. Kuruvila, and M.D. Salas, *Aerdynamic design and optimization in one shot*, 30th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 91-0025 (Reno, Nevada), American Institute of Aeronautics and Astronautics, 1992.

[VD00] D. Venditti and D. Darmofal, *Adjoint error estimation and grid adaptation for functional outputs: application to quasi-one-dimensional flow*, Journal of Computational Physics **164** (2000), 204–227.