NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:
The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

A Relaxed Reduced Space SQP Strategy for Dynamic Optimization Problems

J.S. Logsdon, LT. Biegler

EDRC 06-135-92

A Relaxed Reduced Space SQP Strategy for Dynamic Optimization Problems

J.S. Logsdon and LJ. Biegler
Department of Chemical Engineering
Carnegie Mellon University, Pittsburgh, PA 15213

Recently, strategies have been developed to solve dynamic simulation and optimization problems in a simultaneous manner by applying orthogonal collocation on finite elements and solving the nonlinear program (NLP) with a reduced space Successive Quadratic Programming (SQP) approach. In this paper we develop a relaxed simultaneous approach that leads to faster performance. The method operates in the reduced space of the control variables and solves the collocation equations inexactly at each SQP iteration. Unlike previous simultaneous formulations, it is able to consider the state variables one element at a time. Also, this approach is compared on two process examples to the reduced gradient, feasible path approach outlined in Logsdon and Biegler (1992). Here nonlinear programs with up to 5500 variables are solved with only 40% of the effort. Finally, a theoretical analysis of this approach is provided.

NLP Formulation

Consider the following general controlI problem forte[0, tf]:

$$\min_{u(t),z(t)} \wedge(tf)) \qquad (1)$$

$$\text{s.t. } \dot{z}(t) = F(z(t), u(t)), \quad z(0) = z_0$$

$$g(z(t),u(t)) \leq 0 \qquad gf<z(tf)) \leq 0$$

$$z(t)^L \leq z(t) * z(t)^U$$

$$u(t)^L \leq u(t) \wedge u(t)^U$$

where *F(z(tf)) is the objective function; g(u(t), z(t)) is the inequality design constraint vector over time; z(t) is the state profile vector with initial conditions zo and upper and lower bounds $z(t)^u$, $z(t)^L$; u(t) are the control profiles with upper and lower bounds $u(t)^u$ and $u(t)^L$; and gf are inequality constraints at final conditions. As described in Logsdon and Biegler (1992), we discretize the DAE system by applying orthogonal collocation and replace the differential equations by algebraic residual equations. These residuals are evaluated at the shifted roots of an orthogonal Legendre polynomial over a finite element, i, in time, t e [&, Ct+ll- Thus for the initial value problem, i = F(x, u(t), z(t)), t £ (0, tf), z(0) = ZQ one can write the following residual equation at points tik (at point k in element i):

$$\sum_{j=0}^{K} z_{ij} \; \dot{\phi}_j(\tau_k) - \Delta C_i \, F(z_{ifa} U_{fc}) = 0 \qquad i = 1,\ldots NE, \; k = 1,\ldots K \qquad (2)$$

where the polynomial basis functions <}j Ok) and <}j (tk), are calculated beforehand and depend only on the Legendre polynomial root locations. Here AC is the element length and zft and U& are Lagrange-type polynomial coefficients for the state and control variable profiles, respectively, at collocation point k and element i. Also, note that tik = Ci + A& Xk# In our discretization, we enforce the continuity of the states at element endpoints but allow the control profiles to have discontinuities at these endpoints. Here, for i = 2....NE:

$$z_{i0} = \sum_{j=0}^{K} z_{i-1,j}\,\phi_j(\tau = 1) \qquad (3)$$

These endpoints also provide the initial conditions for the next element states. Substituting the discretized equations for the ODEs yields the NLP formulation for (1). It consists of the discretized DAE model, the

continuity equations for states variables, and additional equality and inequality constraints. Finally, approximation error constraints are added that ensure the accuracy of the solution profiles. These can range from simple bounds on element lengths to constraints based on detailed approximation error estimates (see Vasantharajan and Biegler, 1990).

Min **Y(z$_f$,p)**
zy , Uy , z$_f$, A& (4)

^{s}-t- **$\overset{K}{2}$^ ^{z}ij ^k)-A^i F (z^uik) = 0 ^{z}io - ^{z}o - °**
i=0

g (zy, Uij, Ad) ≤ 0, Zio = $\sum^{K}_{j=0}$ z$_{i-1,j}$ϕ$_j$ (τ=1)

gf(Zf) ≤ 0, Zf = $\overset{K}{2}$,ZNE,j<l>j(X=l)
j=0

^{z}ij^L s ^{z}ij^U £ ^{z}ij, ^{u}ij^L * ^{u}ij * ^{u}ij^U,

ACi ^ ACi ^ **ACi** ^ ACi = tf
i= 1

Variables in (4) include A£$_r$, the finite element lengths for i =1,...NE, z$_f$, the value of the state at the final **time, and** z^, **u^, the collocation coefficients for the state and control profiles, respectively. Problem (4) can be solved by any large scale nonlinear programming solver. However, it is quite large and some form of decomposition is desirable in order to solve this problem efficiently. To motivate our approach we define and partition the variables in (4) as follows:** yi **are variables defined at the beginning of element i (zip), as well as initial and final conditions (zo and zf); wi are internal state variables in the collocation** equations, (zflc, **k=l,K), for element i; vi are control variables in the collocation equations,** (ujk, k=l,K) **for** element i **as well as element lengths** (ACi). **Problem (4) can be rewritten as:**

Min ¥(yNE+i) (5)
Yi» Wi, Vi
s.t hi(yi, wi, v0 = Ayi + qi(wi, vi) = 0 (6)
yi+l-ocyi-Cwi = O, yi= zo (7)
g(wi,Vi)<0 (8)
vi^L£vi£vi^u wi^L^wi^wi^u i = l,NE

where a is a scalar and A and C are matrices composed of elements from the Lagrange basis functions. A straightforward approach to solving (5)-(8) is to apply a general purpose large-scale NLP algorithm (e.g. RND/SQP or MINOS) directly and linearize (6)-(8) at the jth SQP iteration. Linearizing (6) yields:

(Ayjl + qKwp, vp)) + D(wii ,$_{Vi}$J) ($_{Wi}$ - wp) + G(wil ,vp) ($_{Vi}$ - $_{Vi}$J) = 0 (9)

Here the matrices are defined by D = V$_w$hi^T and G = V$_v$hi^T. Note, however, that the variables for ***all*** **of the elements must be stored and recovered in order to be used for die next linearization. Also, to calculate Lagrange multiplier estimates for (9) the matrices D(wp ,vii) and G(wii ,ViJ) need to be stored as well. For large systems of ODFs which require many finite elements, this requirement can be prohibitive.**

As an alternative to this *direct simultaneous approach* we note from the above formulation that the equations (6) and (7) can be solved sequentially for wi and yi+i, i = 1,... NE, once vi has been updated by SQP. This forms the basis of our decomposition. In the *feasible path approach* developed in Logsdon and Biegler (1992), (6) is a square system solved by a Newton method for wi, with vi and yi given. To motivate a relaxation of this method, we first state the *feasible path algorithm* in Logsdon and Biegler (1992):

Feasible Path Algorithm:

0. Choose the number of elements (NE) and the corresponding number (K) of collocation points. Initialize the control variables, state variables and element lengths.

1. For values of the control variables and element lengths at SQP iteration j, and initial conditions for the state variables, perform the following for each element i (i = 1,...NE):

   1.1 Using the initial conditions of element i, yji, as starting guesses, solve the residual equations (6) by a Newton algorithm to obtain the interior states, Wji.

   1.2 From linearization of (6) (e.g., (9)) calculate the sensitivity of this element's dependent variables with respect to y* and vi:

   $$|\wedge = - D(wi_{i}vi)^{-1}G(w|,v|), \quad \wedge i \underset{i}{=} - D(wj,vj)^{"1} A$$

   Note that A, C, D, and G can be determined analytically from the differential equations and $D^{"1}$ is available from step 1.1.

   1.3 Apply the continuity equations and solve for the next element's initial conditions:

   $$y_{i+1} - \alpha y_i - C\ wi = 0, \quad \wedge- = - C\ D(w|,vj\wedge G\wedge vj), \quad \wedge- = \alpha I - C\ D(w|,v|)^{-1}A$$

   1.4 Chainrule **the** derivatives from previous elements and update:

   $$\frac{\partial y_{i+1}}{dv_m} = \frac{\partial y_{m+1}}{dv_m}\frac{\partial y_{m+2}}{3y_m+1} \#" \frac{\partial y_{i+1}}{dyi} \quad \text{form} = 1, \dots i-1$$

2. **Continue until an intermediate element is reached that influences an** inequality ($g(w_c,vc)$), or until **the last element is reached. Determine the reduced gradients of the objective and inequality constraint functions with respect to v (i=l,...NE) according to equation (10),** where $v^T = [vi^T, v2^T, \dots v_{NE}{}^T]$.

$$Z^T\nabla\psi = \frac{\partial y_f}{\partial v}\frac{\partial \psi}{\partial y_f} \qquad Z^T\nabla g = \frac{\partial w_c}{\partial v}\frac{\partial g}{\partial w_c} + \frac{\partial g}{\partial v_c} \qquad \textbf{(10)}$$

3. **Call the SQP algorithm. If Kuhn - Tucker conditions are satisfied, STOP.** Otherwise **SQP** creates **and solves the following quadratic program (11) at iteration j:**

$$\text{Min}_{\Delta v}\ (\nabla\psi^T Z)^j\ \Delta v + \frac{1}{2}\Delta v^T(Z^TBZ)^j\Delta v \qquad \textbf{(11)}$$

$$\text{s.t} \quad g^j + (\nabla g^T Z)^j\,\Delta v \le 0$$

to determine the search direction, Av. Note that this QP contains all of the state and control variable inequality constraints. In addition, our SQP method also updates the reduced Hessian matrix, (Z^TBZ), based on the BFGS formula, and performs a line search to determine the steplength *X*, for the decision variables (see Cuthrell and Biegler, 1985).

5. Set vJ+l= vJ 4- *X* Av and j = j+1. Return to step 1 with a new set of decision variables from SQP.

Unlike the *direct simultaneous approach*, this *feasible path approach* has the advantage that only one finite element needs to be considered at a time in step 1. In addition, since hj has the same structure for each element, the sparsity of large systems can easily be exploited. This decomposition procedure has been applied widely in the solution of ordinary and partial differential equations, where it is referred to as *parameter condensation* or *compactification* (Ascher et al, 1988). For the special case where the state ·variables occur linearly in the differential equations, Logsdon and Biegler (1992) note that (6) becomes: 'hi(yi, wi, vi) = Ayi + Dwi + si(vi) = 0 and wj = - D^{-1}(Ayi + Sj(vi)). If (6) is generally nonlinear, however, solving for WJ may be expensive. Instead, we consider an *relaxed approach* to the solution of (5) - (8). Instead of solving (6) completely for each element at each iteration, we set all of the internal state " variables $\dot{w}p$ to $\dot{y}p$ (the value at the beginning of the element) and linearize (6) about this point and $\dot{v}p$. Starting at this point we execute a limited number of Newton iterations for wji to obtain $\overline{w}$~j. Equations (6) are then linearized about this point to yield:

$$(Ay_i^j + q_i(\overline{w}''i, vji)) + D(\overline{v}T_{\{}, v_iJ)(w_i - \overline{w}''i) = 0 \qquad (12)$$

We set wji **to $\overline{w}$~i and equation (7) is then solved sequentially** to **determine yi+ii. Here the** key question is:

How many Newton iterations are sufficient to determine W[and still allow for reliable and fast convergence of the solution to (4)?

A sufficient **condition to prove convergence requires that the collocation equations be forced to converge monotonically as the number of SQP iterations increases. Thus, our *relaxed SQP algorithm* is essentially the** same **as the one presented above, except that the solution step in 1.1 is modified as follows:**

Step 1.1 (Modified) For a set of fixed decision variables, begin with the initial conditions for the state variables in element i and determine the interior state variables, wji so that the following inequality is satisfied for constants C>0,0<r<l and* Y *> 1.

$$\left\|h_i(w_i^j, v_i^j)\right\| \le C r^{\gamma j} \qquad i = 1, \ldots NE \qquad (13)$$

Clearly, this condition can be satisfied by performing a small number of Newton iterations. Moreover, simpler fixed point iterations may also be substituted. Note that the collocation equations are converged more tightly as the NLP algorithm proceeds toward the optimum. Thus, if the optimization problem is difficult to converge, condition (13) effectively leads to a feasible path algorithm. Otherwise, if a few Newton iterations are sufficient to allow simultaneous convergence, both the collocation equations and the NLP converge quickly.

Relaxed methods that apply a fixed number of Newton iterations to the equalities have also been tried previously on various practical optimization problems. Specifying this number *a priori* is difficult and we are aware of no convergence proof in the literature for such methods, without a condition like (13). Moreover, in this study we develop a superlinear convergence property that does not depend on specific values of C, *y* and r. Intuitively, one can argue that by ensuring that the collocation equations converge along a monotonic sequence, and by enforcing a line search for the control variables and element lengths, we ensure that we have a convergent SQP algorithm. This argument is shown rigorously by the following property.

Local Convergence Property: Given the infeasible path algorithm described above, which satisfies (13) and is made globally convergent by a line search algorithm. Then for unit steplengths, a quasi-Newton method that satisfies the Dennis-More[1] (see Fletcher, 1987) conditions and *any* values for C > 0, 0 < r < 1 and y> 1, the rate of convergence of the **SQP** algorithm is R-superlinear, i.e:

$$||x_k - x^*| \leq \hat{C} \hat{r}^{Tk}$$

with values for the following constants: $\hat{C} > 0$, $\hat{y} > 1$. Here $x^T = [w^T, y^T, v^T]$.

Proof: The proof of the Local Property is given in the Appendix.

Note that although the R-superlinear property is not as strong as the Q-superlinear properties for conventional SQP methods, this rate of convergence still leads to acceptable performance. To enforce *convergence from poor starting points* we perform a line search, along the direction obtained from (QP3), in order to find a steplength that decreases an augmented Lagrangian function (Cuthrell and Biegler, 1985). Such a point can be guaranteed if **the** collocation equations are converged tightly enough, i.e. in the later stages **of the infeasible path algorithm (and, of, course,** in **the feasible path algorithm). Moreover, early** in **the algorithm, line search failures can be overcome by** restarting **the problem with converged** collocation **equations, and using smaller values of C and r in (13). This restart was not necessary for the** examples **presented in this study.**

Clearly the tradeoff between *the feasible path* and *relaxed* methods is **the cost of converging the** collocation **equations vs. a slower convergence rate for SQP.** When equations (6) are easy to converge (e.g. our first **example) both approaches are about the same. On the other hand, if the collocation equations are highly nonlinear and expensive to solve (the second example), considerable savings are obtained with the relaxed approach. The tradeoff between the *direct simultaneous SQP approach* and the *relaxed method* is the cost of storage of the intermediate variables vs. additional Newton steps within each SQP iteration. For problems with many state variables and/or many finite elements, storage costs for the direct simultaneous approach can be prohibitive. In the next section we demonstrate and compare the performance of these algorithms on dynamic process optimization problems.**

Example Problems

We consider two dynamic process optimization problems to compare the feasible path and relaxed algorithms. We consider first a small problem where (6) is easy to converge and then solve an example where (6) is much larger and more nonlinear. In the first case we observe little difference between the two approaches while in the second there is a significant advantage to the relaxed approach.

Consider the nonlinear batch reactor (Ray, 1981) with temperature as the control variable; it is desired to maximize one of the products after a fixed reaction time. Here we consider the following reaction, A — > B —> CThe problem is nonlinear in the rate equations for the concentration of A. Letting ci and C2 represent the concentration of A and B, respectively, the optimal control problem is:

$$\mathbf{Max} \quad c_2(1.0)$$

$$\mathbf{s.t} \quad \wedge = -4000 \exp(-2500\, T) c?$$

$$\wedge = 4000 \exp(-2500\, T) <\% - 6.2 \times 10^5 \exp(-5000\, T) c_2$$

$$C!(0) = 1.0, \quad c_2(0) = 0 \quad 298 <£\, T <\geq 398$$

Using two point collocation, six finite elements and the feasible path approach, Logsdon and Biegler (1992) achieve convergence in 18 SQP iterations. Here, 3 to 4 Newton iterations weie required initially for each element and then 2-3 iterations were required to converge the state variables for subsequent control variable movements from SQP. The temperature profile was initialized as a constant profile at 300° and the steep optimal profiles are shown in Logsdon and Biegler (1992). In the relaxed approach, it becomes clear that a wide range of values for C, r and y will lead to similar performance because, with either one or two Newton iterations of (6) we obtain monotonic convergence for the collocation equations. This arises

because the ODEs are only mildly nonlinear in the state variables and thus relatively easy to converge. Therefore, we only need to compare the performance of the feasible path method with relaxed approaches that use only one or two Newton iterations on the state variables. All three methods converge to virtually identical profiles with about the same performance, as seen in Table 1

Table 1: Comparison of Approaches for Example Problems

Method	SQP Iterations	CPU Seconds (Vaxstation 3200)	Objective Function
Nonlinear Batch Reactor (18 control variables, 36 state variables, 6 finite elements)			
Feasible Path	18	34.65	0.610775 (final concentration)
Two Newton Iters.	16	33.36	0.610667
~~One Newton Iters.~~	~~28~~	~~40.62~~	~~0.610667~~
Maximum Distillate Problem (36 control variables, 5460 state variables, **12** finite elements)			
Feasible Path	51	4184	38.615 (moles recovered)
Two Newton Iters.	61	1663	38.615
Monotonic Sequence for h(x)	58	1595	38.615

To demonstrate the performance of the relaxed algorithm on a larger problem, we consider a **cyclohexane/hexane batch distillation which was simulated successfully simulated by Domenech and Enjalbert (1980). Here we consider the maximum distillate problem for a batch time of one hour.** The **simulation model described by Domenech and Enjalbert (1980) includes tray and condenser holdups and assumes equimolar overflow (no enthalpy balances). For this 12-stage column we specify a boilup rate** of **120 moles/hr, plate and condenser holdups of one and ten models, respectively, and a 45/55% charge of a 200 mole cyclohexane/toluene mixture. Antoine equations are used to describe the equilibium K values and ideal mixtures are assumed Further details of this optimization problem can also be found in Logsdon (1990), This problem is complicated by a state variable inequality constraint for the top plate cyclohexane concentration (greater than 99.5 % for each point in time). This constraint requires the use of orthogonal collocation, as state variable constraints are very difficult to enforce otherwise.**

To form (4) we write explicit expressions for the derivative terms and the right hand sides of the differential equations at each of the collocation points. This requires expressions for the equilibrium constants, and additionally the bubble point constraints must be enforced to determine the temperature profiles. To complete the model, we add the continuity equations for the state profiles. A total of 12 elements and 3 collocation points per element were chosen; details of this choice are given in Logsdon (1990). The resulting NLP (4) has 429 constraints with 457 variables *for each element.* Also, given the size of the system of equations, the ITU decomposition of D(wi, v0 in step 1 was performed by the MA28 sparse matrix solver.

For the feasible path, reduced gradient method an average of 6 to 8 Newton iterations are required to 'converge the state variables in each element Unlike the small example, we see that the state variables are quite nonlinear and the collocation equations are harder to converge. Next we solve this problem with a heuristic relaxed method where only two Newton iterations are applied to (5) in order to yield reasonable 'linearizations for the SQP method. Finally, we apply the relaxed method with constraints (13) that force monotonic convergence of (5). Here we choose, C = r = 0.8 and y = 1.1. As a result, the number of Newton iterations for each element averages between one and three iterations. Convergence is not sensitive to this choice of parameters since a wide range of constants will guarantee monotonic convergence and give similar performance. However, use of (13) is critical for convergence of SQP. For example, if only one Newton iteration is applied to (5) and (13) is not enforced, the SQP algorithm terminates in a line search failure. Here, the control profiles for all three methods are virtually identical to the ones presented in Logsdon (1990) as are the objective function values in Table 1. Solution times are also listed in Table 1 for

the different methods. Note that both relaxed approaches require only 40% of the effort required by the feasible path case.

Summary and Conclusions

This paper presents a numerical method for obtaining optimal control profiles that is structured to the discretized ODE equations. Here, orthogonal collocation is used within an NLP framework in order to solve for the control profiles, although other ODE discretizations may be suitable as well. Moreover, the NLP framework allows us to enforce state path constraints and control path constraints. Also, switching times and integration step lengths can be posed as optimization variables to obtain accurate solution of the optimal control profile. This work thus enhances previous optimization-based studies in that we explore a decomposition technique in order to reduce the problem size, tailor the decomposition to the problem structure and develop a relaxed, reduced gradient algorithm. In addition, a local convergence analysis is provided which shows that the infeasible path approach can converge at an R-superlinear rate. Two example problems were considered in the paper. In the first example, the collocation equations were easy to solve in each element and we found that the relaxed approach performed as well as the feasible path approach of Logsdon and Biegler (1992). However, on a much larger batch distillation optimization, the benefits of the relaxed approach are clear. For a 12-stage binary column, the resulting formulation had about 5500 variables and 36 degrees of freedom. Here the relaxed approach was about 2.5 times faster than the feasible path approach and again converged to a virtually identical solution.

Acknowledgements

Financial support from the Engineering Design Research Center, an NSF supported Engineering Center at Carnegie Mellon University is gratefully acknowledged. The second author is also grateful for partial support provided by the Math and Computer Science Division of Argonne National Laboratory for the later stages of this work.

References

Ascher, U.M., R.M. Mattheij and R. D. Russell, Numerical Solution of Boundary Value Problems for Ordinary Differential Equations, Prentice-Hall, Englewood Cliffs, NJ (1988)

Cuthrell, J. E., and L.T. Biegler, '* Improved Infeasible Path Optimization For Sequential Modular Simulators-II The Optimization Algorithm ", Computers and Chemical Engineering , Vol. 9, No. 3, pp. 257 (1985)

Domenech, S. and M. Enjalbert , Computers & Chemical Engineering ,Vol. 5, p. 181 (1981)

Fletcher, R., Practical Methods of Optimization, Wiley, Chichester (1987)

Logsdon, J. S., PhD Thesis, Carnegie Mellon University, Pittsburgh, PA (1990)

Logsdon, J.S. and L.T. Biegler, "Decomposition Strategies for Large Scale Dynamic Optimization Problems," ***Chem. Engr. ScL,*** **47, 4, p. 851 (1992)**

Ray, W.H., Advanced Process Control, McGraw - Hill , New York (1981)

Vasantharajan, S. and L. T. Biegler, "Simultaneous Strategies for Parameter Optimizxation and Accurate Solution of Differential-Algebraic Systems," ***Computers and Chemical Engineering,*** **14, 8, p. 1083 (1990)**

Appendix: Proof of *Local* Convergence Property

We start with the SQP iteration j for all the variables (with a steplength of unity): XJ+I = XJ + dj, where we partition the search direction into null and range space directions, dj = Zj Av + Yj pY- Here the first term, is due to the solution of (11) and the second is due to changes in the dependent variables in order to converge the collocation equations. Now define the distance to the optimal solution, ej, by XJ - x* and substitute the expression from (11). For simplicity we treat the case where the inequality constraints are inactive (Active constraints can also be handled here at the expense of additional algebraic manipulations.). Also, we note that second order optimality conditions are satisfied and that Z*T\Y*Z*, the projected Hessian at the optimum is positive semidefinite. The matrix Zj^BjZj = Mj is maintained positive definite and bounded by the quasi-Newton updates in SQP. From (11) we have

:AVJ = $-(M_j)^{-1}$ZJV\|/$_j$ and

$$e_{j+1} = x_{j+i} \sim x*\mp\ x_j - x* + [Zj\ \ Yj] \begin{bmatrix} -(Mj\wedge zJVVj1 \\ PY\ \ \ J \end{bmatrix} = e_j + [Zj\ \ Yj] \begin{bmatrix} -(M_j)^{-1} Z_j^T \nabla \psi_j \\ PY \end{bmatrix}$$

Now if.we partition e; and ej+-1 asi**fbllows:**

$$\begin{bmatrix} e_j^u \\ <\!* \end{bmatrix} = [Zj\wedge Yj]^{-1}\ {}^e J \qquad \begin{bmatrix} {}^c j+1 \\ e_{j+1}^y \end{bmatrix} = [Z_j\ \ Y_j]^{-1}\ e_{j+1}$$

then we obtain the relations:

$$\begin{bmatrix} "\ {}^e y+i" \\ e_{j+1}^y \end{bmatrix} = \begin{bmatrix} e_j^u \\ e_j^y \end{bmatrix} + \begin{bmatrix} -(Mj)^{"1z}I^V Vj \\ PY \end{bmatrix}$$

For ej $_{we}$ note the following property, because Z(XJ) lies in the null space of Vh(xj)^T.

0 = h(x*) = h($_{xj}$) + V h ($_{X\ j}$) V O| |ej|p = h(xj) + Vh(xj)^{T}Y$_{je}$J+ O| |ej||^2

Because Yj is chosen so that Vh(xj)^{T}Yj is nonsingular, we have that e? = O||h(xj)|| and by (13), e^v is R-superlinearly convergent.

We now consider the e^u component. From above we have: Mje^$_{+1}$ = Mjej^1 - ZjVx|/j and by Taylor series:

$M_j e_{j+1}^u$ =M$_j$e]^1 - ZlVy* - Zj\V*(Z$_{je}$y+Y$_{je}$J) + q |$_{ej}$||^2

where W* is the Hessian of die Lagrange function at the solution. Now the following relations derive from above and the definition of ej$_{+1}$:

$M_j e_{j+1}^u$ = (Mj - ZIW*Z$_j$)eY$_{+1}$ - (Mj - ZIW*ZJ)AVJ - zJw*YjeJ + O| |ej||^2

$Z_*^T W_*$Zkey$_{+1}$ = (Mj - ZlW*Zj)Avj - ZjW.YjeJ + O| |ej|p

The remainder of the proof is based on the quasi-Newton condition originally due to Dennis and More' (see Fletcher, 1987) I I (MJ-ZJW«ZJ)AVJ||£ x| |dj|| £ K(| |$_{ej}$|| +| |e$_{j+}$i||) where K>0andlim$_{j\_»oo}$ K= 0. Substituting this expression above and taking norms gives the following order terms: co| lej^H <, K (| |e^u||) + C rYJ where © > 0, and the last term on the right hand side comes from (13) and the order result for ey.The desired result now follows from the following induction. We assume that 11 e^{u}j11 is bounded above by D *J* J, for some D > 0 and 1< f < 7, and show that | \$_t$*$_{jl}$\\ <; D r*0+D. Substituting into the above order result shows that: 11 ^{eu}j+i 11 ^^{D} ^^{j}(^{K} + C/D kY - 0 i) ^ D rf 0+D $_{smC}$e the terms in parentheses in the middle expression go to zero as j goes to infinity. For j sufficiently large, they are therefore less than r. Thus, both components of ej have the desired convergence rates and the algorithm is R-superlinearly convergent