Chapter 3

The Dynamic-Q optimization algorithm

3.1 Introduction

An efficient *constrained* optimization method is presented in this chapter. The method, called the Dynamic-Q method, consists of applying the dynamic trajectory optimization algorithm (see Appendix B) to successive quadratic approximations of the actual optimization problem. This method may be considered as an extension of the unconstrained SQSD method, presented in Chapter 2, to one capable of handling general constrained optimization problems.

Due to its efficiency with respect to the number of function evaluations required for convergence, the Dynamic-Q method is primarily intended for optimization problems where function evaluations are expensive. Such problems occur frequently in engineering applications where time consuming numerical simulations may be used for function evaluations. Amongst others, these numerical analyses may take the form of a computational fluid dynam-

ics (CFD) simulation, a structural analysis by means of the finite element method (FEM) or a dynamic simulation of a multibody system. Because these simulations are usually expensive to perform, and because the relevant functions may not be known analytically, standard classical optimization methods are normally not suited to these types of problems. Also, as will be shown, the storage requirements of the Dynamic-Q method are minimal. No Hessian information is required. The method is therefore particularly suitable for problems where the number of variables n is large.

In the next section sequential quadratic programming (SQP) methods are briefly discussed to allow for comparison with the proposed method. Next, the Dynamic-Q methodology is presented. Finally the performance of the method is tested and compared to that of an SQP method.

3.2 Sequential quadratic programming methods

Sequential quadratic programming (SQP) methods have been developed over the past thirty years, and are generally considered to be some of the most efficient algorithms available today. Based on Lagrangian methods, it can be shown that the solution $\mathbf{x}^*$ of the nonlinear equality constrained optimization problem

$$\min_{\mathbf{x}} f(\mathbf{x});\ \mathbf{x} = [x_1, x_2, \ldots, x_n]^\top \in \Re^n \quad \text{subject to } \mathbf{h}(\mathbf{x}) = \mathbf{0} \tag{3.1}$$

where $f(\mathbf{x})$ and $\mathbf{h}(\mathbf{x})$ are respectively a scalar and a vector function of $\mathbf{x}$, can be obtained by solving, at successive approximations $\mathbf{x}^i$ to $\mathbf{x}^*$, a sequence of corresponding quadratic programming (QP) subproblems $(QP[i], i = 0, 1, 2, \ldots)$

containing linearized constraints of the following form:

$$\begin{aligned} &\min_{\mathbf{s}} f(\mathbf{x}^i) + \nabla^\top f(\mathbf{x}^i)\mathbf{s} + \frac{1}{2}\mathbf{s}^\top \mathbf{W}^i \mathbf{s} \\ &\text{subject to } \nabla^\top \mathbf{h}(\mathbf{x}^i)\mathbf{s} + \mathbf{h}(\mathbf{x}^i) = \mathbf{0} \end{aligned} \tag{3.2}$$

where $\mathbf{W}^i = \nabla^2 f(\mathbf{x}^i) + \boldsymbol{\lambda}^{i\top} \nabla^2 \mathbf{h}(\mathbf{x}^i)$, with $\boldsymbol{\lambda}^i$ denoting the associated vector of Lagrange multipliers. The solution to subproblem $QP[i]$ is denoted by $\mathbf{s}^i$ and the point at which the next subproblem $QP[i+1]$ is constructed is $\mathbf{x}^{i+1} = \mathbf{x}^i + \mathbf{s}^i$. If successful, the SQP method yields a sequence $\mathbf{x}^0, \mathbf{x}^1, \mathbf{x}^2, \ldots$ that converges to $\mathbf{x}^*$. The particular QP subproblem given here is one of a number of possible forms that may be chosen.

Based on the above argument, a simple SQP algorithm is as follows (Papalambros and Wilde [117]).

Algorithm 3.1 Simple SQP algorithm

Initialization: Select initial point $\mathbf{x}^0$ and initial Lagrange multipliers $\boldsymbol{\lambda}^0$. Set $i := 1$.

Main procedure:

1. Solve the quadratic programming problem $QP[i]$ corresponding to (3.2) to determine $\mathbf{s}^i$ and $\boldsymbol{\lambda}^{i+1}$.
2. Set $\mathbf{x}^{i+1} := \mathbf{x}^i + \mathbf{s}^i$.
3. If termination criteria are satisfied, set $\mathbf{x}^* = \mathbf{x}^{i+1}$ and stop; else set $i := i + 1$ and go to Step 1.

Numerous authors have proposed modifications and variations to the above basic algorithm. There are four areas in which the differences are most prominent. The first of these is the way in which *inequality constraints* are also included in the algorithm. For optimization problems containing inequality constraints an active set strategy may be used. This strategy can be implemented in one of two ways, either on the original problem or by including all

of the inequality constraints in the QP subproblem, and applying an active set strategy to the subproblem. The second point of difference lies in the way the QP subproblem is solved. Almost any method for nonlinear programming, such as the augmented Lagrangian method or the dual method, may be specially adapted to the solution of the QP subproblem. A third way in which SQP algorithms differ from each other is in the computation of second derivatives of the problem. In the above simple SQP algorithm it is necessary to evaluate the second derivatives of the objective function and the constraints in the computation of $\mathbf{W}^i$, which will usually be a computationally intensive process. In any event, the storage of Hessian information is required which implies the availability of $O(n^2)$ storage locations, and the determination and manipulation of the elements of the $n \times n$ Hessian matrix. Some authors have avoided the latter difficulties by applying quasi-Newton updating formulae to approximate the second derivatives. Powell [118] , for example, has proposed using the BFGS formula to approximate these second derivatives. A fourth point of difference lies in dealing with the feasibility or infeasibility of the constructed subproblems. If the QP subproblem (3.2) is constructed at a point far from the solution $\mathbf{x}^*$ of the constrained optimization problem (3.1), then the subproblem may have an unbounded or infeasible solution. For this reason many modern SQP algorithms rather use $\mathbf{s}^i$ as a search direction. Then the point $\mathbf{x}^{i+1}$ at which the next subproblem is constructed is set at $\mathbf{x}^{i+1} := \mathbf{x}^i + \alpha_i \mathbf{s}^i$ with the step size α_i determined by performing a line search on an appropriate merit function in the direction $\mathbf{s}^i$.

3.3 The Dynamic-Q method

Consider the general nonlinear optimization problem:

$$\begin{aligned} &\min_{\mathbf{x}} f(\mathbf{x});\ \mathbf{x} = [x_1, x_2, \ldots, x_n]^\top \in \Re^n \\ &\text{subject to} \\ &g_j(\mathbf{x}) = \mathbf{0};\ j = 1, 2, \ldots, p \\ &h_k(\mathbf{x}) = \mathbf{0};\ k = 1, 2, \ldots, q \end{aligned} \tag{3.3}$$

where $f(\mathbf{x})$, $g_j(\mathbf{x})$ and $h_k(\mathbf{x})$ are scalar functions of $\mathbf{x}$.

In the Dynamic-Q approach, successive subproblems $P[i],\ i = 0, 1, 2, \ldots$ are generated, at successive approximations $\mathbf{x}^i$ to the solution $\mathbf{x}^*$, by constructing *spherically quadratic* approximations $\tilde{f}(\mathbf{x})$, $\tilde{g}_j(\mathbf{x})$ and $\tilde{h}_k(\mathbf{x})$ to $f(\mathbf{x})$, $g_j(\mathbf{x})$ and $h_k(\mathbf{x})$. These approximation functions, evaluated at a point $\mathbf{x}^i$, are given by

$$\begin{aligned} \tilde{f}(\mathbf{x}) &= f(\mathbf{x}^i) + \nabla^\top f(\mathbf{x}^i)(\mathbf{x} - \mathbf{x}^i) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^i)^\top \mathbf{A}(\mathbf{x} - \mathbf{x}^i) \\ \tilde{g}_j(\mathbf{x}) &= g_j(\mathbf{x}^i) + \nabla^\top g_j(\mathbf{x}^i)(\mathbf{x} - \mathbf{x}^i) \\ &\quad + \frac{1}{2}(\mathbf{x} - \mathbf{x}^i)^\top \mathbf{B}_j(\mathbf{x} - \mathbf{x}^i),\ j = 1, \ldots, p \\ \tilde{h}_k(\mathbf{x}) &= h_k(\mathbf{x}^i) + \nabla^\top h_k(\mathbf{x}^i)(\mathbf{x} - \mathbf{x}^i) \\ &\quad + \frac{1}{2}(\mathbf{x} - \mathbf{x}^i)^\top \mathbf{C}_k(\mathbf{x} - \mathbf{x}^i),\ k = 1, \ldots, q \end{aligned} \tag{3.4}$$

with the Hessian matrices $\mathbf{A}$, $\mathbf{B}_j$ and $\mathbf{C}_k$ taking on the simple forms

$$\begin{aligned} \mathbf{A} &= \text{diag}(a, a, \ldots, a) = a\mathbf{I} \\ \mathbf{B}_j &= b_j\mathbf{I} \\ \mathbf{C}_k &= c_k\mathbf{I} \end{aligned} \tag{3.5}$$

Clearly the identical entries along the diagonal of the Hessian matrices indicate that the approximate subproblems $P[i]$ are indeed spherically quadratic.

For the first subproblem ($i = 0$) a linear approximation is formed by setting the curvatures a, b_j and c_k to zero. Thereafter a, b_j and c_k are chosen so

that the approximating functions (3.4) interpolate their corresponding actual functions at both $\mathbf{x}^i$ and $\mathbf{x}^{i-1}$. These conditions imply that for $i = 1, 2, 3, \ldots$

$$\begin{aligned} a &= \frac{2\left[f(\mathbf{x}^{i-1}) - f(\mathbf{x}^i) - \nabla^{\mathrm{T}} f(\mathbf{x}^i)(\mathbf{x}^{i-1} - \mathbf{x}^i)\right]}{\left\|\mathbf{x}^{i-1} - \mathbf{x}^i\right\|^2} \\ b_j &= \frac{2\left[g_j(\mathbf{x}^{i-1}) - g_j(\mathbf{x}^i) - \nabla^{\mathrm{T}} g_j(\mathbf{x}^i)(\mathbf{x}^{i-1} - \mathbf{x}^i)\right]}{\left\|\mathbf{x}^{i-1} - \mathbf{x}^i\right\|^2}, \quad j = 1, \ldots, p \\ c_k &= \frac{2\left[h_k(\mathbf{x}^{i-1}) - h_k(\mathbf{x}^i) - \nabla^{\mathrm{T}} h_k(\mathbf{x}^i)(\mathbf{x}^{i-1} - \mathbf{x}^i)\right]}{\left\|\mathbf{x}^{i-1} - \mathbf{x}^i\right\|^2}, \quad k = 1, \ldots, q \end{aligned} \tag{3.6}$$

If the gradient vectors $\nabla^{\mathrm{T}} f$, $\nabla^{\mathrm{T}} g_j$ and $\nabla^{\mathrm{T}} h_k$ are not known analytically, they may be approximated from functional data by means of first-order forward finite differences.

The particular choice of spherically quadratic approximations in the Dynamic-Q algorithm has implications on the computational and storage requirements of the method. Since the second derivatives of the objective function and constraints are approximated using function and gradient data, the $O(n^2)$ calculations and storage locations, which would usually be required for these second derivatives, are not needed. The computational and storage resources for the Dynamic-Q method are thus reduced to $O(n)$. At most, $4+p+q+r+s$ n-vectors need be stored (where p, q, r and s are respectively the number of inequality and equality constraints and the number of lower and upper limits of the variables). These savings become significant when the number of variables becomes large. For this reason it is expected that the Dynamic-Q method is well suited, for example, to engineering problems such as structural optimization problems where a large number of variables are present.

In many optimization problems, additional simple side constraints of the form $\hat{k}_i \leq x_i \leq \check{k}_i$ occur. Constants $\hat{k}_i$ and $\check{k}_i$ respectively represent lower and upper bounds for variable x_i. Since these constraints are of a simple form (having zero curvature), they need not be approximated in the Dynamic-Q method and are instead explicitly treated as special linear inequality constraints. Constraints corresponding to lower and upper limits are respectively

of the form

$$\begin{aligned} \hat{g}_l(\mathbf{x}) &= \hat{k}_{vl} - x_{vl} \leq 0,\ l = 1, 2, \ldots, r \leq n \\ \check{g}_m(\mathbf{x}) &= x_{wm} - \check{k}_{wm} \leq 0,\ m = 1, 2, \ldots, s \leq n \end{aligned} \tag{3.7}$$

where $vl \in \hat{I} = (v1, v2, \ldots, vr)$ the set of r subscripts corresponding to the set of variables for which respective lower bounds $\hat{k}_{vl}$ are prescribed, and $wm \in \check{I} = (w1, w2, \ldots, ws)$ the set of s subscripts corresponding to the set of variables for which respective upper bounds $\check{k}_{wm}$ are prescribed. The subscripts vl and wm are used since there will, in general, not be n lower and upper limits, i.e. usually $r \neq n$ and $s \neq n$.

In order to obtain convergence to the solution in a controlled and stable manner, move limits are placed on the variables. For each approximate subproblem $P[i]$ this move limit takes the form of an additional single inequality constraint

$$g_\rho(\mathbf{x}) = \left\|\mathbf{x} - \mathbf{x}^{i-1}\right\|^2 - \rho^2 \leq 0 \tag{3.8}$$

where ρ is an appropriately chosen step limit and $\mathbf{x}^{i-1}$ is the solution to the previous subproblem.

The approximate subproblem, constructed at $\mathbf{x}^i$, to the optimization problem (3.4) (plus simple side constraints (3.7) and move limit (3.8)), thus becomes $P[i]$:

$$\begin{gathered} \min_{\mathbf{x}} \tilde{f}(\mathbf{x}),\ \mathbf{x} = [x_1, x_2, ..., x_n]^\top \in \Re^n \\ \text{subject to} \\ \tilde{g}_j(\mathbf{x}) \leq 0,\ j = 1, 2, \ldots, p \\ \tilde{h}_k(\mathbf{x}) = 0,\ k = 1, 2, \ldots, q \\ \hat{g}_l(\mathbf{x}) \leq 0,\ l = 1, 2, \ldots, r \\ \check{g}_m(\mathbf{x}) \leq 0,\ m = 1, 2, \ldots, s \\ g_\rho(\mathbf{x}) = \left\|\mathbf{x} - \mathbf{x}^{i-1}\right\|^2 - \rho^2 \leq 0 \end{gathered} \tag{3.9}$$

with solution $\mathbf{x}^{*i}$. The Dynamic-Q algorithm is given by Algorithm 3.2. In

the Dynamic-Q method the subproblems generated are solved using the dynamic trajectory, or "leap-frog" (LfopC) method of Snyman [101, 102] for unconstrained optimization applied to penalty function formulations (Snyman *et al.* [119], Snyman [103]) of the constrained problem. A brief description of the LfopC algorithm is given in Appendix B.

Algorithm 3.2 Dynamic-Q algorithm

Initialization: Select starting point $\mathbf{x}^0$ and move limit ρ. Set $i := 0$.
Main procedure:

1. Evaluate $f(\mathbf{x}^i)$, $g_j(\mathbf{x}^i)$ and $h_k(\mathbf{x}^i)$ as well as $\nabla f(\mathbf{x}^i)$, $\nabla g_j(\mathbf{x}^i)$ and $\nabla h_k(\mathbf{x}^i)$. If termination criteria are satisfied set $\mathbf{x}^* = \mathbf{x}^i$ and stop.

2. Construct a local approximation $P[i]$ to the optimization problem at $\mathbf{x}^i$ using expressions (3.4) to (3.6).

3. Solve the approximated subproblem $P[i]$ (given by (3.9)) using the constrained optimizer LfopC with $\mathbf{x}^0 := \mathbf{x}^i$ (see Appendix B) to give $\mathbf{x}^{*i}$.

4. Set $i := i + 1$, $\mathbf{x}^i := \mathbf{x}^{*(i-1)}$ and return to Step 2.

The LfopC algorithm possesses a number of outstanding characteristics, which makes it highly suitable for implementation in the Dynamic-Q methodology. The algorithm requires only gradient information and no explicit line searches or function evaluations are performed. These properties, together with the influence of the fundamental physical principles underlying the method, ensure that the algorithm is extremely robust. This has been proven over many years of testing (Snyman [103]). A further desirable characteristic related to its robustness, and the main reason for its application in solving the subproblems in the Dynamic-Q algorithm, is that if there is no feasible solution to the problem, the LfopC algorithm will still find the best possible compromised solution without breaking down. The Dynamic-Q algorithm thus usually converges to a solution from an infeasible remote point

without the need to use line searches between subproblems, as is the case with SQP. The LfopC algorithm used by Dynamic-Q is identical to that presented in Snyman [103] except for a minor change to Lfop which is advisable should the subproblems become effectively unconstrained.

3.4 Numerical results and conclusion

The Dynamic-Q method requires very few parameter settings by the user. Other than convergence criteria and specification of a maximum number of iterations, the only parameter required is the step limit ρ. The algorithm is not very sensitive to the choice of this parameter, however, ρ should be chosen of the same order of magnitude as the diameter of the region of interest. For the problems listed in Table 3.1 a step limit of $\rho = 1$ was used except for problems 72 and 106 where step limits and $\rho = 100$ were used respectively.

Given specified positive tolerances ε_x, ε_f and ε_c, then at step i termination of the algorithm occurs if the normalized step size

$$\frac{\|\mathbf{x}^i - \mathbf{x}^{i-1}\|}{1 + \|\mathbf{x}^i\|} < \varepsilon_x \tag{3.10}$$

or if the normalized change in function value

$$\frac{|f^i - f_{\text{best}}|}{1 + |f_{\text{best}}|} < \varepsilon_f \tag{3.11}$$

where f_{best} is the lowest previous feasible function value and the current $\mathbf{x}^i$ is feasible. The point $\mathbf{x}^i$ is considered feasible if the absolute value of the violation of each constraint is less than ε_c. This particular function termination criterion is used since the Dynamic-Q algorithm may at times exhibit oscillatory behavior near the solution.

In Table 3.1, for the same starting points, the performance of the Dynamic-Q method on some standard test problems is compared to results obtained for Powell's SQP method as reported by Hock and Schittkowski [120]. The

Prob. #	n	f_{act}	SQP			Dynamic-Q		
			N^{fg}	f^*	E^r	N^{fg}	f^*	E^r
2	2	5.04E-02	16∼	2.84E+01	2.70E+01	7*	4.94E+00	<1.00E-08
10	2	-1.00E+00	12	-1.00E+00	5.00E-08	13	-1.00E+00	<1.00E-08
12	2	-3.00E+01	12	-3.00E+01	<1.00E-08	9	-3.00E+01	<1.00E-08
13	2	1.00E+00	45	1.00E+00	5.00E-08	50$	9.59E-01	2.07E-02
14	2	1.39E+00	6	1.39E+00	8.07E-09	5	1.39E+00	7.86E-07
15	2	3.07E+02	5	3.07E+02	<1.00E-08	15*	3.60E+02	5.55E-07
16	2	2.50E-01	6*	2.31E+01	<1.00E-08	5*	2.31E+01	<1.00E-08
17	2	1.00E+00	12	1.00E+00	<1.00E-08	16	1.00E+00	<1.00E-08
20	2	3.82E+01	20	3.82E+01	4.83E-09	4*	4.02E+01	<1.00E-08
22	2	1.00E+00	9	1.00E+00	<1.00E-08	3	1.00E+00	<1.00E-08
23	2	2.00E+00	7	2.00E+00	<1.00E-08	5	2.00E+00	<1.00E-08
24	2	-1.00E+00	5	-1.00E+00	<1.00E-08	4	-1.00E+00	1.00E-08
26	3	0.00E+00	19	4.05E-08	4.05E-08	27	1.79E-07	1.79E-07
27	3	4.00E-02	25	4.00E-02	1.73E-08	28	4.00E-02	9.62E-10
28	3	0.00E+00	5	2.98E-21	2.98E-21	12	7.56E-10	7.56E-10
29	3	-2.26E+01	13	-2.26E+01	8.59E-11	11	-2.26E+01	8.59E-11
30	3	1.00E+00	14	1.00E+00	<1.00E-08	5	1.00E+00	<1.00E-08
31	3	6.00E+00	10	6.00E+00	<1.00E-08	10	6.00E+00	1.43E-08
32	3	1.00E+00	3	1.00E+00	<1.00E-08	4	1.00E+00	<1.00E-08
33	3	-4.59E+00	5*	-4.00E+00	<1.00E-08	3*	-4.00E+00	<1.00E-08
36	3	-3.30E+03	4	-3.30E+03	<1.00E-08	15	-3.30E+03	<1.00E-08
45	5	1.00E+00	8	1.00E+00	<1.00E-08	7	1.00E+00	1.00E-08
52	5	5.33E+00	8	5.33E+00	5.62E-09	12	5.33E+00	1.02E-08
55	6	6.33E+00	1∼	6.00E+00	4.54E-02	2*	6.66E+00	1.30E-09
56	7	-3.46E+00	11	-3.46E+00	<1.00E-08	20	-3.46E+00	6.73E-08
60	3	3.26E-02	9	3.26E-02	3.17E-08	11	3.26E-02	1.21E-09
61	3	-1.44E+02	10	-1.44E+02	1.52E-08	10	-1.44E+02	1.52E-08
63	3	9.62E+02	9	9.62E+02	2.18E-09	6	9.62E+02	2.18E-09
65	3	9.54E-01	11∼	2.80E+00	9.47E-01	9	9.54E-01	2.90E-08
71	4	1.70E+01	5	1.70E+01	1.67E-08	6	1.70E+01	1.67E-08
72	4	7.28E+02	35	7.28E+02	1.37E-08	30	7.28E+02	1.37E-08
76	4	-4.68E+00	6	-4.68E+00	3.34E-09	8	-4.68E+00	3.34E-09
78	5	-2.92E+00	9	-2.92E+00	2.55E-09	6	-2.92E+00	2.55E-09
80	5	5.39E-02	7	5.39E-02	7.59E-10	6	5.39E-02	7.59E-10
81	5	5.39E-02	8	5.39E-02	1.71E-09	12	5.39E-02	1.90E-10
100	7	6.80E+02	20	6.80E+02	<1.00E-08	16	6.80E+02	1.46E-10
104	8	3.95E+00	19	3.95E+00	8.00E-09	42	3.95E+00	5.26E-08
106	8	7.05E+03	44	7.05E+03	1.18E-05	79	7.05E+03	1.18E-05
108	9	-8.66E-01	9*	-6.97E-01	1.32E-02	26	-8.66E-01	3.32E-09
118	15	6.65E+02	∼	∼	∼	38	6.65E+02	3.00E-08
Svan	21	2.80E+02	150	2.80E+02	9.96E-05	93	2.80E+02	1.59E-06

* Converges to a local minimum - listed E^r relative to function value at local minimum;
∼ Fails; $ Terminates on maximum number of steps

Table 3.1: Performance of the Dynamic-Q and SQP optimization algorithms

problem numbers given correspond to the problem numbers in Hock and Schittkowski's book. For each problem, the actual function value f_{act} is given, as well as, for each method, the calculated function value f^* at convergence, the relative function error

$$E^r = \frac{|f_{\text{act}} - f^*|}{1 + |f_{\text{act}}|} \tag{3.12}$$

and the number of function-gradient evaluations (N^{fg}) required for convergence. In some cases it was not possible to calculate the relative function error due to rounding off of the solutions reported by Hock and Schittkowski. In these cases the calculated solutions were correct to at least eight significant figures. For the Dynamic-Q algorithm, convergence tolerances of $\varepsilon_f = 10^{-8}$ on the function value, $\varepsilon_x = 10^{-5}$ on the step size and $\varepsilon_c = 10^{-6}$ for constraint feasibility, were used. These were chosen to allow for comparison with the reported SQP results.

The result for the 12-corner polytope problem of Svanberg [121] is also given. For this problem the results given in the SQP columns are for Svanberg's Method of Moving Asymptotes (MMA). The recorded number of function evaluations for this method is approximate since the results given correspond to 50 outer iterations of the MMA, each requiring about 3 function evaluations.

A robust and efficient method for nonlinear optimization, with minimal storage requirements compared to those of the SQP method, has been proposed and tested. The particular methodology proposed is made possible by the special properties of the LfopC optimization algorithm (Snyman [103]), which is used to solve the quadratic subproblems. Comparison of the results for Dynamic-Q with the results for the SQP method show that equally accurate results are obtained with comparable number of function evaluations.