Chapter 11

Trust-region interpolation-based methods

11.1 Common features and considerations

In this chapter we present several practical algorithms for derivative-free optimization based on the trust-region framework described in Chapter 10. Although the original algorithms differ from what is presented here, we try to preserve what we see as the main distinguishing ideas, whilst casting the algorithms in a form which is as close as possible to the convergent framework of Chapter 10. What these algorithms have in common is the use of the trust region and of the quadratic models based on polynomial interpolation. The differences between the algorithms lie mainly in the handling of the sample (interpolation) set and in the building of the corresponding model.

The following is a list of questions one needs to answer when designing a trust-region interpolation-based derivative-free method. We will see how these questions are answered by the different methods discussed in this chapter.

1. How many points should be included in the sample set? We know that a quadratic model is desirable, but a completely determined quadratic model based on polynomial interpolation requires $(n+1)(n+2)/2$ function evaluations. It might be too expensive to require such accuracy at each step; hence it may be desirable to build models based on fewer interpolation points, as discussed in Chapter 5. Naturally, another question arises: should the number of points in the sample set be static throughout the algorithm or should it be dynamic? For instance, the algorithm described in Section 11.2 allows a dynamic number of points, whereas the algorithm described in Section 11.3 requires the sample set to have exactly $p_1 \in [n+2,(n+1)(n+2)/2]$ points at each iteration, with p_1 fixed across all iterations.

2. Should the sample set be Λ-poised in $B(x_k;\Delta_k)$ at each iteration and, if not, how should this requirement be relaxed? To enforce Λ-poisedness, even occasionally, one needs to develop a criterion for accepting points into a sample set. This criterion is typically based on some threshold value. What value for the threshold should be chosen and when and how should it be improved if it is chosen badly? If the threshold is chosen too strictly, it might not be possible to find a suitable sample set, and if it is

chosen too loosely, the resulting set could be badly poised. Here we will rely on the material from Chapter 6.

3. Should the sample set always be contained in $B(x_k;\Delta_k)$? The framework in Chapter 10 assumes this,[18] but it might require recomputing too many sample points each time the iterate changes or the trust-region radius is reduced. Hence, the requirement of the sample sets lying in the current trust region may be relaxed in a practical algorithm. In this case, it could be restricted to $B(x_k;r\Delta_k)$, with some fixed $r \geq 1$, or it could be not restricted at all, but updated in such a way that points which are far from x_k are replaced by points in $B(x_k;\Delta_k)$ whenever appropriate.

4. The framework of Chapter 10 allows us to accept new iterates, if any amount of decrease is achieved and the model is sufficiently good, by setting $\eta_0 = 0$ in the notation of Algorithms 10.1 and 10.3. In all the algorithms discussed in this chapter we will allow accepting new iterates based on a simple decrease condition, since it is a universally desirable feature for a derivative-free algorithm. However, each of these algorithms can be trivially modified to accept only new iterates based on a sufficient decrease condition.

11.2 The "DFO" approach

The algorithm described in this section makes use of the machinery developed in the book for the purposes of proving global convergence. It will rely on the material of Chapters 6 and 10. The following are the main distinguishing features of the algorithm, developed by Conn, Scheinberg, and Toint [59, 61] and referred to as "DFO" (derivative-free optimization).

1. It makes use of as many sample points (up to $(n+1)(n+2)/2$) as are available that pass the criterion for Λ-poisedness (described below). In other words the number of points in the sample set Y_k is dynamic.

2. The model is a minimum Frobenius norm quadratic interpolation model as described in (the beginning of) Section 5.3.

3. It always maintains a Λ-poised set of sample points in $B(x_k;r\Delta_k)$ (in the linear or minimum Frobenius norm interpolation sense or in the quadratic interpolation sense), with r a small scaling factor greater than or equal to 1. Making $r \geq 2$ makes the algorithm more practical since it allows the sample points to remain in the current sample sets for a few subsequent iterations even if the current iteration is moved or the trust-region radius is reduced.

4. The original algorithm described in [59] relies on NFPs to measure and maintain Λ-poisedness of sample sets. But the model-improvement algorithm used is very similar to the pivoting algorithms described in Chapter 6. In this section, we discuss a globally convergent version that relies on pivoting algorithms to maintain its sample sets.

[18] The presentation of Chapter 10 could have been adapted to handle the case where the sample set is contained in $B(x_k;r\Delta_k)$, with $r > 0$ a constant fixed across all iterations.

5. At each iteration, the "DFO" algorithm updates the sample set Y_k via the following steps relying on the pivotal algorithms (Algorithms 6.4, 6.5, and 6.6) described in Section 6.3:

   (i) At the beginning of the algorithm, two different threshold values are selected: $0 < \xi_{\text{acc}} < 1/4$ for the acceptance of a point into an interpolation set and $\xi_{\text{imp}} > 1$ for the improvement of the current interpolation set. The value of ξ_{acc} is typically smaller than 0.1. The region within which the pivot polynomials are optimized is a ball of radius Δ_k centered at the current iterate. If we initiate Algorithms 6.4, 6.5, and 6.6 with our monomial basis, then the value of the first pivot is always one; hence it is of no matter which interpolation point is associated with the first pivot. Given any set (possibly, not poised) of interpolation points that contains the current iterate x_k, the algorithms can assign x_k to the first pivot and then select the remaining points so that the resulting set is Λ-poised. This guarantees that the current iterate (and the center of the trust region) can always be included in the well-poised sample set.

   (ii) Consider a sample point $x_k + s_k$ generated during the trust-region minimization step. If this point provides a decrease of the objective function (see Step 3 of Algorithm 11.1 below), then it becomes a new iterate. The center of the trust region is moved, and Algorithm 6.5 is applied (with threshold ξ_{acc}) to all available sample points that lie in the ball of radius $r\Delta_k$ centered at $x_k + s_k$. This procedure will either select a well-poised set of up to $(n+1)(n+2)/2$ points or will stop short due to the lack of suitable sample points.

   (iii) If the new point $x_k + s_k$ does not provide a decrease of the objective function (see Step 3 of Algorithm 11.1 below), then it may still be desirable to add it to the interpolation set, since it brings information which was clearly not provided by the old model. Moreover, the possibly costly function evaluation has already been made. Consequently, we simply add $x_k + s_k$ to Y_k and apply Algorithm 6.5. If in the end some points of the new Y_k were unused, then those points are discarded. Notice that it may happen that $x_k + s_k$ is discarded and the resulting Y_k remains the same as before. This is acceptable from the point of view of the global convergence of the algorithm. It is also acceptable from a practical point of view because it happens very rarely. To avoid such a situation some heuristic approaches can be applied within Algorithm 6.5. For instance, one may put some preference on the points that are closer to the current iterate in the case where a pivot polynomial has similar absolute values at several points.

   (iv) When a model-improvement step is desired after a trust-region step has been computed, then a new point is computed solely with the poisedness of the sample set in mind. If the current interpolation set Y_k is not complete (for linear or minimum Frobenius norm interpolation or quadratic interpolation, depending on the desired accuracy), then one step of Algorithm 6.4 is applied to maximize the absolute value of the appropriate pivot polynomial and increase the size of the interpolation set by one point. Otherwise, we apply one step of Algorithm 6.6 to replace one point of Y_k. The replacement is accepted only if the value of the last pivot polynomial is increased by at least a factor of ξ_{imp}. Otherwise, it is considered that the model-improvement step failed.

6. The pivotal algorithms assume that the set of interpolation points is always shifted and scaled to lie in a ball of radius 1 around the origin. Hence, every time the center or the radius of the trust region changes, the new shifting and scaling have to be introduced and the pivot polynomials have to be computed for the new shifted and scaled set. However, in practice, it is sufficient to change the shifting and scaling only every several iterations. Then one can simply update the set of pivot polynomials whenever no shifting and scaling occurs, which saves considerable computation for larger interpolation set sizes.

Now we present the modified "DFO" algorithm for both the situations where one would like to attain global convergence to first-order stationary points or to second-order stationary points. From the theory of Chapters 3–6, we know that a model is fully linear (FL), resp., fully quadratic (FQ), in $B(x_k; r\Delta_k)$ if it is based on a Λ-poised set with at least $n+1$, resp., $(n+1)(n+2)/2$, points in $B(x_k; r\Delta_k)$—Definitions 6.1 and 6.2 would have to take into account the trust-region radius factor r. When we can guarantee that a model is FL, resp., FQ, we say that the model is certifiably fully linear (CFL), resp., certifiably fully quadratic (CFQ).

Algorithm 11.1 (Modified "DFO" algorithm).

Step 0 (initialization): Choose an initial point x_0, a maximum radius $\Delta_{max} > 0$, and an initial trust-region radius $\Delta_0 \in (0, \Delta_{max}]$. Choose a set Y_0 and compute the initial model $m_0(x)$.

The constants η_1, γ, γ_{inc}, ϵ_c, β, and μ are also chosen and satisfy the conditions $\eta_1 \in (0,1)$, $0 < \gamma < 1 < \gamma_{inc}$, $\epsilon_c > 0$, and $\mu > \beta > 0$. Choose positive pivot thresholds ξ_{acc} and ξ_{imp} and the trust-region radius factor $r \geq 1$. Set $k = 0$.

Step 1 (criticality step): This step is as in Algorithm 10.2 or 10.4, depending on the desired convergence result. Note that m_k, Y_k, and Δ_k might change in the course of this step.

Step 2 (step calculation): This step is as in Algorithm 10.1 or 10.3.

Step 3 (acceptance of the trial point): Check if $m_k(x)$ is CFL/CFQ, which, again, is guaranteed if Y_k contains at least $n+1$, resp., $(n+1)(n+2)/2$, points in $B(x_k; r\Delta_k)$.

Compute $f(x_k + s_k)$ and define

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

If $\rho_k \geq \eta_1$ or if $\rho_k > 0$ and the model is CFL/CFQ, then $x_{k+1} = x_k + s_k$. Otherwise, the model and the iterate remain unchanged ($x_{k+1} = x_k$). Apply the procedure described in items 5(ii) and 5(iii) above to include $x_k + s_k$ and update the sample set. Let the new model and sample set be m_{k+1} and Y_{k+1}.

Step 4 (model improvement): If $\rho_k < \eta_1$, then attempt a model improvement by suitable improvement steps (described in 5(iv) above). Define m_{k+1} and Y_{k+1} to be the (possibly improved) model and sample set.

Step 5 (trust-region radius update): This step is as in Algorithm 10.1 or 10.3. Increment k by one and go to Step 1.

Notes on the algorithm

Due to the use of Algorithms 6.4, 6.5, and 6.6 and the fact that Y_k contains only points which provide acceptable pivot values in those algorithms, we know that as long as Y_k lies in $B(x_k; r\Delta_k)$ it is always Λ-poised, for some unknown, but fixed, Λ. It is not difficult to observe that the model-improvement procedure generates, in a finite and uniformly bounded number of steps, an interpolation set which is Λ-poised for linear or minimum Frobenius norm interpolation or quadratic interpolation, depending on the choice of desired accuracy. From the material in Chapters 3–6, we know that the models that are used are FL or FQ, and the rest of the algorithm fits closely to the framework discussed in Chapter 10. Global convergence to first- or second-order stationary points follows as a simple conclusion.

At each iteration of the algorithm only one or two new sample points are generated; hence at most two function evaluations are required.

The algorithm is very flexible in the number of sample points one can use per iteration. In fact it is easy to modify this algorithm to allow for the use of least-squares regression models.

Aside from the function evaluations and the trust-region step the main computational effort is in computing the pivot polynomials. Generally, if $\mathcal{O}(n^2)$ points are used, then the pivotal algorithm may require $\mathcal{O}(n^6)$ operations. By keeping and reusing the pivot polynomials whenever possible we can reduce the empirical complexity of many iterations; however, some iterations may still require a large numerical effort. For small values of n this effort is often negligible compared to the expense of one function evaluation. For larger values of n, it makes sense to restrict the number of points for each model to $\mathcal{O}(n)$. The models based on the minimum Frobenius norm of the change of the model Hessian work well for a fixed number of points of $\mathcal{O}(n)$. It is natural to use minimum Frobenius norm Lagrange polynomials in the context of such models, which is what is done by the algorithm described in the next section.

11.3 Powell's methods

We now describe an algorithm which is a modification of the algorithms developed by Powell in [188, 189, 190, 191, 192]. We will combine here the features of Powell's most successful algorithm in practice with the trust-region framework of Chapter 10. The trust-region management that Powell is using is different from the one followed in this book. In particular the trust-region radius and the radius of the interpolation set are not the same or related by a constant multiple. While the radius of the interpolation set eventually converges to zero (in theory) the trust-region radius is allowed to remain bounded away from zero. In some respects, the two-region approach is closer to classical trust-region methods (see [57]) for derivative-based optimization. The reason for the trust-region radius to remain bounded away from zero is to allow relatively large steps close to optimality in order to achieve a superlinear local rate of convergence. The theoretical implication of such a choice for our framework is that to obtain a lim-type global convergence result as in Theorem 10.13 one needs to impose a sufficient decrease condition on the trust-region step

acceptance. It is possible to state and analyze the framework of Chapter 10 for the case when the trust-region radius is not converging to zero, but we leave that to the interested reader.

In fact the trust-region management in Powell's methods is more complicated than simply keeping two different radii. The smaller of the two radii is also used to force the interpolation points to be sufficiently far apart to avoid the influence of noise in the function values. Also, the trust-region updating step is more complicated than that of the framework of Chapter 10. These additional schemes are developed to improve the practical performance of the algorithms. They do not significantly affect the main global convergence results but make the analysis quite complex. We present here Powell's method of handling interpolation sets via minimum Frobenius norm Lagrange polynomials embedded into the framework of Chapter 10. The model-improvement algorithm is augmented by a test and a possible extra step which helps ensure that FL models can be constructed in a finite uniformly bounded number of steps. For full details on the original Powell methods we refer the reader to [188, 189, 190, 191, 192], in particular his UOBYQA [189] and NEWUOA [192] methods. The following are the distinguishing features of the algorithm:

1. All sample sets on all iterations contain $p_1 \in [n+2, (n+1)(n+2)/2]$ points. The value $p_1 = 2n+1$ is a natural choice and is the default value in the NEWUOA software package [192].

2. The models are quadratic interpolation models with the remaining degrees of freedom taken by minimizing the Frobenius norm of the *change* in the model Hessian, with respect to the model used in the previous iteration, as described in (the last part of) Section 5.3.

3. Sample set maintenance is based on Algorithm 6.3 modified to handle the minimum Frobenius norm Lagrange polynomials as discussed in Chapter 6.

4. There is no need to involve scaling, as Lagrange polynomials scale automatically. Shifts also do not have any effect. However, shifting the points with respect to the current center of the trust region has an important effect on numerical accuracy and hence is performed regularly.

5. Updates of the sample set in the original Powell algorithms are performed via the following steps:

   (i) The set of minimum Frobenius norm Lagrange polynomials $\ell_i(x)$, $i = 0, \ldots, p$, is maintained at every iteration.

   (ii) If the trust-region minimization of the kth iteration produces a step s_k which is not too short compared to the maximum distance between the sample points and the current iterate, then the function f is evaluated at $x_k + s_k$ and the new point becomes the next iterate x_{k+1} if the reduction in f is sufficient, or just positive and the model is guaranteed to be FL. The quality of the model m_k is established at the end of the preceding iteration. If the new point $x_k + s_k$ is accepted as the new iterate, it is included into Y_k, by removing the point y^i such that the distance $\|x_k - y^i\|$ and the value $|\ell_i(x_k + s_k)|$ are both, in the sense that follows, as large as possible. The trade off between these two objectives is achieved by maximizing the weighted absolute value $w_i|\ell_i(x_k + s_k)|$, where w_i reflects

the distance $\|x_k - y^i\|$. In fact, in [191], instead of maximizing $|\ell_i(x_k + s_k)|$, Powell proposes optimizing the coefficient of the rank-two update of the system defining the Lagrange polynomials, to explicitly improve the conditioning of that system. However, operating with the values of the Lagrange polynomial serves the same purpose and makes it easier for us to use the theory developed in Chapters 5 and 6.

(iii) When the step s_k is rejected, the new point $x_k + s_k$ can still be accepted into Y_k, by removing the point y^i such that the value $w_i|\ell_i(x_k + s_k)|$ is maximized, where w_i reflects the distance $\|x_k - y^i\|$, as long as either $|\ell_i(x_k + s_k)| > 1$ or $\|y^i - x_k\| > r\Delta_k$.

(iv) If the improvement in the objective function is not sufficient, and it is believed that the model needs to be improved, then the algorithm chooses a point in Y_k which is the furthest from x_k and attempts to replace it with a point which maximizes the absolute value of the corresponding Lagrange polynomial in the trust region (or in the smaller interpolation set region, as is done in [188, 189, 190, 191, 192]).

For global convergence we need the criticality step (see Chapter 10), where one may need to construct an FL model. An analogue of this step can be found in Powell's work, and is related to improving geometry when the step s_k is much smaller than Δ_k, which occurs when the gradient of the model is small relative to the Hessian. Here we use the same step that was used in the globally convergent framework of Chapter 10; that is, we use the size of the gradient as the criticality test. Scaling with respect to the size of the Hessian is also possible, as long as arbitrarily small or large scaling factors are not allowed. The modified algorithm is presented below.

Algorithm 11.2 (Minimum Frobenius norm Lagrange polynomial-based algorithm).

Step 0 (initialization): Select $p_1 \in [n+2, (n+1)(n+2)/2]$. Choose an initial point x_0, a maximum radius $\Delta_{max} > 0$, and an initial trust-region radius $\Delta_0 \in (0, \Delta_{max}]$. Choose the trust-region radius factor $r \geq 1$. Choose a well-poised set Y_0 with cardinality p_1. Compute the minimum Frobenius norm Lagrange polynomials $\ell_i(x)$, $i = 0, \ldots, p$, associated with Y_0 and the corresponding quadratic model m_0.

Select a positive threshold value for the improvement step, $\Lambda > \max\{|\ell_i(x)| : i = 0, \ldots, p,\ x \in B(x_0; \Delta_0)\}$.

The constants η_1, γ, γ_{inc}, ϵ_c, τ, β, and μ are also given and satisfy the conditions $\eta_1 \in (0,1)$, $0 < \gamma < 1 < \gamma_{inc}$, $\epsilon_c > 0$, $0 < \tau < 1$, and $\mu > \beta > 0$. Set $k = 0$.

Step 1 (criticality step): This step is as in Algorithm 10.1 (the model-improvement algorithm relies on minimum Frobenius norm Lagrange polynomials). Note that m_k, Y_k, and Δ_k might change in the course of this step.

Step 2 (step calculation): This step is as in Algorithm 10.1.

Step 3 (acceptance of the trial point): If $\|s_k\| \geq \tau \max\{\|y^j - x_k\| : y^j \in Y_k\}$, then compute y^i, where $i \in \operatorname{argmax}_j\{w_j|\ell_j(x_k + s_k)| : y^j \in Y_k\}$ and $w_j > 0$ are weights chosen to give preference to points that are further from x_k.

Compute $f(x_k + s_k)$ and define

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

Set $x_{k+1} = x_k + s_k$ if $\rho_k \geq \eta_1$ or if $\rho_k > 0$ and it is known that m_k is CFL in $B(x_k; r\Delta_k)$. Include $x_k + s_k$ into the sample set Y_{k+1} by replacing y^i. Otherwise, $x_{k+1} = x_k$, and if either $\|y^i - x_k\| > r\Delta_k$ or $|\ell_i(x_k + s_k)| > 1$, then accept $x_k + s_k$ into the sample set Y_{k+1} by replacing y^i.

Step 4 (model improvement): (This step is executed only if $x_{k+1} = x_k$ after Step 3.) Choose $y^i \in \operatorname{argmax}_j\{\|y^j - x_k\| : y^j \in Y_k\}$ and find a new $y^i_* \in \operatorname{argmax}\{|\ell_i(x)| : x \in B(x_k; \Delta_k)\}$. If $\|y^i - x_k\| > r\Delta_k$ or if $|\ell_i(y^i_*)| > \Lambda$, then replace y^i by y^i_* in Y_{k+1}. Otherwise, consider the next furthest point from x_k and repeat the process. If eventually a point y^i is found in Y_{k+1} such that $\max\{|\ell_i(x)| : x \in B(x_k; \Delta_k)\} > \Lambda$, then this point is replaced. If no such point is found, then there is no need for improvement because Y_k is Λ-poised in $B(x_k; r\Delta_k)$. Hence, we conclude that m_{k+1} based on Y_{k+1} is CFL in $B(x_k; r\Delta_k)$.

Step 5 (trust-region radius update): This step is as in Algorithm 10.1. Update the model m_k to obtain m_{k+1} and recompute the minimum Frobenius norm Lagrange polynomials. Increment k by one and go to Step 1.

Notes on the algorithm

The steps for updating the interpolation set described above are sufficient to guarantee that FL models can be constructed in a finite, uniformly bounded number of steps, as required by the convergence analysis in Chapter 10. It is easy to see that the model-improvement step first attempts to replace any interpolation point that is too far from the current iterate (outside the $r\Delta_k$ radius). Clearly, this can be accomplished in at most p steps. Once all the interpolation points are close enough, then the model-improvement step checks the Λ-poisedness of the interpolation set, by maximizing the absolute values of the minimum Frobenius norm Lagrange polynomials one by one. If the set is already Λ-poised, then there is nothing to improve. If the set is not Λ-poised, then one point in the set is replaced. We know from Chapter 6 and from Theorem 6.3 that after a finite and uniformly bounded number of steps, a Λ-poised set is obtained in $B(x_k; r\Delta_k)$ and, hence, an FL model is obtained. Notice that Step 3 may change the interpolation set, but, in the case when the trust-region step is successful, it is not important (from a theoretical perspective) what this change does to the model. On the other hand, when the current iterate does not change, the change of Y_k allowed by Step 3 may only either improve the poisedness of Y_k or replace a far away point of Y_k by $x_k + s_k$. Hence, if the model m_k is FL at the beginning of an unsuccessful Step 3, then m_{k+1} is so at the end of it; that is, an unsuccessful Step 3 may only improve the model, while Step 4 is guaranteed to produce an FL model in a finite number or iterations. Hence, we conclude that the global convergence theory of Chapter 10 (for first-order stationary points) applies to Algorithm 11.2.

In the original methods proposed by Powell [189, 192] there is no explicit check in the model-improvement step that $|\ell_i(y^i_*)| > \Lambda > 1$; instead the model-improvement step is far more complex, but it is aimed at replacing the old interpolation points by the points within

the trust region which give a large value for the corresponding Lagrange polynomials. One of the features of this procedure is to avoid the exact global optimization of the Lagrange polynomials. Consider, now, replacing an interpolation point by another point for which the absolute value of the appropriate Lagrange polynomial exceeds $\Lambda > 1$, rather than seeking the exact global maximizer of such absolute value. From the discussion after the proof of Theorem 6.3 we know that such a step provides an improvement of the poisedness of the interpolation set. On the other hand, we do not know if such replacements steps result in a model-improvement algorithm that terminates in a uniformly bounded finite number of steps. It is possible to modify the model-improvement algorithm to allow several such "cheap" improvement steps and switch to global optimization only when the maximum number of consecutive cheap steps is exceeded.

In the original Powell methods a simple decrease step is always accepted as a new iterate. We allow this in Algorithm 11.2 also but only when we can guarantee that the sample set Y_k on which the model m_k was constructed is Λ-poised in $B(x_k; r\Delta_k)$. Hence we may need to have either an extra check or a flag which can be set by Step 4 to indicate that a Λ-poised set is at hand. There is a trade off between how often such a check is performed and how often a simple decrease can be accepted. The exact implementation is a matter of practical choice.

Although not guaranteed, it is expected that the interpolation set remains well poised throughout the algorithm as long as it is in $B(x_k; r\Delta_k)$. In practice one can start with a simple well-poised interpolation set, such as is used by Powell:

$$Y_0 = \{x_0, x_0 + \Delta e_1, \ldots, x_0 + \Delta e_n, x_0 - \Delta e_1, \ldots, x_0 - \Delta e_n\},$$

where e_i is the ith vector of the identity (note that this is nothing else than (complete) polling using the positive basis $D_{\oplus}$). The advantage of this set is that due to its special structure it takes $\mathcal{O}(n^2)$ operations to construct the initial set of minimum Frobenius norm Lagrange polynomials [192].

11.4 Wedge methods

The two methods we have considered so far in this chapter are closely related in that they generate two kinds of sample points, those that are aimed at reducing the function value and those that are aimed at improving the poisedness of the sample set. A natural question is: is it possible to achieve both of these objectives at once, by generating one kind of points? These two objectives may be conflicting, but is it possible to find a point which improves the value of the model while maintaining an acceptable level of poisedness? An algorithm based on this idea was proposed by Marazzi and Nocedal in [163]. This algorithm follows the approach of attempting to generate points which *simultaneously* provide sufficient decrease for the model (and, thus, hopefully, provide decrease for the objective function) and satisfy the Λ-poisedness condition. At every iteration, the trust-region subproblem minimization is augmented by an additional constraint which does not allow the new point to lie near a certain manifold. This manifold is defined by a subset of the sample set that is fixed for the given iteration and allows for all possible nonpoised sets that contain the fixed subset of sample points. In the original method the constraint which defined the proximity to such a manifold (in the linear case) has the shape of a wedge, hence the name of the method. To provide theoretical guarantees for the method we have to replace the

"wedge"-shaped constraint by a "strip"-shaped one. We will use the terminology "wedge method" to refer to its origin.

Let Y_k be the interpolation set at iteration k, and let $m_k(x_k+s)$ be the model based on Y_k,

$$m_k(x_k+s) \;=\; m_k(x_k)+g_k^\top s+\frac{1}{2}s^\top H_k s.$$

Before a trust-region step is taken we need to identify which point of Y_k is going to be replaced with the new point. Notice that the other two methods of this chapter first make a trust-region step and then decide which point of Y_k to replace with the new point, if appropriate. To choose a point to leave Y_k we use the following algorithm, which is a combination of Algorithms 6.5 and 6.6.

Algorithm 11.3 (Selection of the outgoing point in Y_k).

Initialization: Shift and scale Y_k and consider the new set $\hat{Y}_k=\{(y^i-x_k)/\Delta_k, i=0,\dots,p\}$. Set $u_i(x)=\bar{\phi}_i(x)$, $i=0,\dots,p$. Choose $\xi>0$ and $r\geq 1$. Assume that $\hat{Y}_k$ contains p_1 poised points. Set $i_k=p$.

For $i=0,\dots,p-1$

1. **Point selection:** Find $j_i=\operatorname{argmax}_{j\in\{i,\dots,p\}:\,\|\hat{y}^j\|\leq r}|u_i(\hat{y}^j)|$.
2. **Point acceptance:** If $|u_i(\hat{y}^{j_i})|<\xi$, then $i_k=i$ and stop. Otherwise, swap points $\hat{y}^i$ and $\hat{y}^{j_i}$ in the set $\hat{Y}_k$.
3. **Gaussian elimination:** For $j=i+1,\dots,p$

$$u_j(x) \;\leftarrow\; u_j(x)-\frac{u_j(\hat{y}^i)}{u_i(\hat{y}^i)}u_i(x).$$

This algorithm selects the points in $\hat{Y}_k$ that lie in $B(0;r)$ and which give pivot values of at least ξ. It also computes the first i_k pivot polynomials. The index i_k is the index of the first point that needs to be replaced. By the logic of the algorithm such a point either does not lie in $B(0;r)$ or does not give a large enough pivot value. In case such points do not exist, then $i_k=p$ and the result of Algorithm 11.3 is similar to the result of one round of Algorithm 6.6; in that case $\hat{y}^p$ is simply a good point to be replaced.

The purpose of Algorithm 11.3 is to identify points which should be replaced either because they are far away from the current iterate or because replacing them may make the poisedness of $\hat{Y}_k$ better. Once such a point $\hat{y}^{i_k}$ is identified and the corresponding pivot polynomial is computed, the following problem is solved to produce the trust-region step:

$$\begin{array}{ll}\min\limits_{s\in\mathbb{R}^n} & m_k(x_k)+g_k^\top s+\frac{1}{2}s^\top H_k s\\ \text{s.t.} & \|s\| \;\leq\; \Delta_k,\\ & |u_{i_k}(s/\Delta_k)|\geq\xi.\end{array}\tag{11.1}$$

The above problem is feasible for small enough ξ; hence an optimal solution exists. However, what we are interested in is an approximate solution, which is feasible and satisfies the fraction of Cauchy decrease condition (10.9). We will show that such a solution

also exists, for small enough ξ, in the section of the analysis of the wedge algorithm. The actual approach to (approximately) solve such a problem can be found in [163]. We will not repeat the details here and will simply assume that if such a solution exists, then it can be found. This might require a large computational effort, but it does not require any function evaluations for f, which is normally the main concern of a derivative-free algorithm.

Recall the example of an "ideal" interpolation set illustrated in Figure 3.4. We shift and scale this set so that it lies in $B(0;1)$. Let us apply Algorithm 6.5 to construct the pivot polynomials and then consider the constraint $|u_p(s/\Delta_k)| \geq \xi$ for the last quadratic pivot polynomial. In Figure 11.1 we can see the area of the trust region which is being discarded by this constraints for the cases when $\xi = 0.01$ and $\xi = 0.1$.

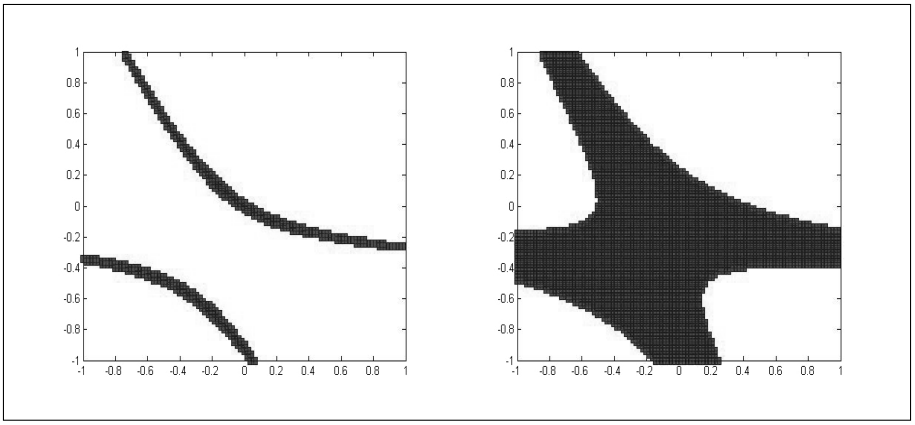

Figure 11.1. *Areas of the (squared) trust region which are forbidden by the wedge constraint $|u_p(s/\Delta_k)| \geq \xi$ for the cases $\xi = 0.01$ (left) and $\xi = 0.1$ (right). The example refers to the sample set of Figure* 3.4.

The following is a summary of the main distinguishing features of the algorithm.

1. The algorithm, as originally proposed in [163], uses exactly $n+1$ points in the linear case and $(n+1)(n+2)/2$ points in the quadratic case. It is possible to extend it to use minimum Frobenius norm models, but we will not do so here.

2. Unlike the two methods discussed in the previous sections, in this case the point which will leave Y_k is selected *before* each trust-region minimization step.

3. The algorithm that we present here aims at maintaining a Λ-poised set of sample points as long as the points are in $B(x_k; r\Delta_k)$, with $r \geq 1$. Due to the use of Algorithm 11.3, and assuming that the trust-region subproblem (11.1) can be solved successfully, we can show that such a step can be obtained after a finite number of iterations (see point 5 below).

   The original wedge algorithm does not employ Algorithm 11.3; it simply selects the point which is the furthest from the current iterate as the point that should be replaced. For the global convergence theory to apply, we need to make sure that the interpolation set can become Λ-poised in $B(x_k; r\Delta_k)$ after a finite number of

improvement steps before the trust-region radius can be reduced (in case the trust-region step is unsuccessful). If we insist that all interpolation points lie in $B(x_k;\Delta_k)$ in order to consider the model FL or FQ, then we would have to replace a lot of points each time the iterate changes or the trust region is reduced. Making r large, say $r \geq 2$, makes the algorithm more practical.

4. There are no direct attempts to improve geometry, but there is a heuristic way of updating the balance between the required poisedness of the sample set and the required fraction of the sufficient reduction in the model (see [163]).

5. If i_k produced by Algorithm 11.3 is smaller than p, then, after the trust-region step (with the wedge constraint), the new Y_k may not be Λ-poised in $B(x_k;\Delta_k)$. However, if the wedge trust-region subproblem is solved successfully (namely, a solution for (11.1) is found for the current value of ξ_k and it satisfies a fraction of Cauchy decrease), then, if x_k and Δ_k remain unchanged, then on the next iteration i_k increases by 1. Hence, eventually either a successful trust-region step is achieved or i_k equals p. In this case we know that if the wedge trust-region subproblem (11.1) is solved successfully, then the resulting new model is CFL or CFQ (depending on p) in $B(x_k;\Delta_k)$. Notice that it may not be so in $B(x_{k+1};\Delta_{k+1})$, but, unless sufficient decrease of the objective function is obtained by the trust-region step, $B(x_{k+1};\Delta_{k+1}) = B(x_k;\Delta_k)$ and a CFL/CFQ model is obtained for certain.

6. The criticality step is crucial for the global convergence properties, just as in the cases of the other two algorithms described in this chapter. A criticality step needs to provide an FL/FQ model in a sufficiently small ball around the current iterate. It is possible to construct such a set by repeatedly solving the problem (11.1). However, since many interpolation points may need to be replaced, the intermediate models are likely to be inaccurate. A more practical approach is to generate a Λ-poised set directly, without consideration for the model decreases, by applying Algorithm 6.5, for instance.

We present the modified wedge algorithm for the quadratic interpolation case, but the analysis that follows mainly concerns global convergence to first-order stationary points.

Algorithm 11.4 (Modified wedge algorithm).

Step 0 (initialization): Choose an initial point x_0, a maximum radius $\Delta_{max} > 0$, and an initial trust-region radius $\Delta_0 \in (0, \Delta_{max}]$. Choose a set Y_0 and compute the initial model $m_0(x)$.

The constants η_1, γ, γ_{inc}, ϵ_c, β, and μ are also given and satisfy the conditions $\eta_1 \in (0,1)$, $0 < \gamma < 1 < \gamma_{inc}$, $\epsilon_c > 0$, and $\mu > \beta > 0$. Choose a positive pivot threshold ξ_0, a Cauchy decrease coefficient $0 < \kappa_{fcd} < 1$, and a trust-region radius factor $r \geq 1$. Set $k = 0$.

Step 1 (criticality step): This step is as in Algorithm 10.1 (note that m_k, Y_k, and Δ_k might change in the course of this step).

Step 2 (step calculation): Apply Algorithm 11.3 to Y_k to select the pivot polynomial $u_{i_k}(x)$. Solve problem (11.1) to obtain s_k.

Step 3 (wedge management): Check if the new solution satisfies the fraction of Cauchy decrease condition:

$$m_k(x_k) - m_k(x_k + s_k) \geq \frac{\kappa_{fcd}}{2} \|g_k\| \min\left\{ \frac{\|g_k\|}{\|H_k\|}, 1 \right\}.$$

If so, let $y^{i_k} = x_k + s_k$ and proceed to the next step. Otherwise, the wedge constraint is relaxed by reducing $\xi_k \leftarrow \xi_k/2$ and return to Step 2.

Step 4 (acceptance of the trial point): Compute $f(x_k + s_k)$ and define

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

If $\rho_k \geq \eta_1$ or if $\rho_k > 0$ and $Y_k \subset B(x_k; r\Delta_k)$, then $x_{k+1} = x_k + s_k$. Otherwise, $x_{k+1} = x_k$.

Step 5 (trust-region radius update): This step is as in Algorithm 10.1. Update the model m_k based on the new sample set Y_k. Increment k by one and go to Step 1.

Notes on the algorithm

The computational effort per iteration of Algorithm 11.4 is $\mathcal{O}(p^3)$ for each Step 2, to which we need to add the computational effort of solving a wedge subproblem. Solving the wedge subproblem can be expensive. It would be justifiable if the overall number of function evaluations is decreased. Marazzi and Nocedal [163] have reported on computational results which showed that the method is comparable with other derivative-free methods, but there was no particularly strong advantage, and other derivative-free methods, in particular that of Powell [192], have recently improved significantly. However, we note that Marazzi and Nocedal's implementation was relying on building complete quadratic models, rather than minimum Frobenius norm models. The latter are used in `DFO` and `NEWUOA` and often give an advantage to a trust-region interpolation-based algorithm. Hence another implementation of the wedge method might perform better.

As with the other algorithms we discussed above, there are numerous ways in which the wedge algorithm can be improved. For instance, one can consider replacing several different points in Y_k and solving several instances of the wedge trust-region subproblem (11.1) to obtain the best trade off between geometry and model decrease. This will increase the cost per iteration, but may decrease the overall number of iterations, and hence the overall number of function evaluations.

Validity of the wedge algorithm for the first-order case

We have described a version of the wedge method which uses quadratic interpolation models. However, Steps 1 and 3 indicate that the method tries only to achieve global convergence to a first-order stationary point. From the results of Chapter 10, we can conclude that all we need to show in order to establish such a convergence is that an FL model can be achieved after a finite, uniformly bounded number of unsuccessful iterations. Below we will show that if only a fraction of Cauchy decrease is required from the approximate

solution to (11.1), then this is indeed the case. Moreover, we can show that an FQ model can also be achieved in a finite, uniformly bounded number of unsuccessful iterations while keeping the fraction of Cauchy decrease requirement.

However, to establish global convergence to a second-order stationary point one can no longer be satisfied with only the fraction of Cauchy decrease, and the fraction of optimal decrease (10.13) has to be considered. Thus, second-order criteria would have to be applied in Steps 1 and 3. This complicates the proofs and, as we will discuss at the end of the chapter, requires an additional safeguard step to be introduced into the algorithm.

Consider a step of Algorithm 11.3 (which is a modification of Algorithm 6.5 described in Chapter 6). Recall the vector v, the vector of the coefficients (with respect to the natural basis $\bar{\phi}$) of the polynomial $u_i(x)$ which is computed during the pivot computation. We know that $\|v\|_\infty \geq 1$. In general we know that for any $0 < \xi < 1/4$ there exists $\hat{s}_k$ such that $\|\hat{s}_k\| \leq 1$ and that $|v^\top \bar{\phi}(\hat{s}_k)| \geq \xi$ (see Lemmas 3.10 and 3.12). The purpose of the analysis below is to show that there exists a constant $\xi > 0$, such that for all iterations it is possible to find a step s_k simultaneously satisfying the following:

- $\|s_k\| \leq \Delta_k$,
- s_k provides at least a fixed fraction of Cauchy decrease of the model,
- $|v^\top \bar{\phi}(s_k/\Delta_k)| \geq \xi$.

From this it will follow that the model can be made FL or FQ by the wedge method after a finite, uniformly bounded number of unsuccessful iterations.

Let us consider the unscaled trust-region subproblem, without the wedge constraint. We index the elements of the unscaled problem by "u" and omit the index k from now on:

$$\begin{array}{rl} \min\limits_{s_u \in \mathbb{R}^n} & c + g_u^\top s_u + \frac{1}{2} s_u^\top H_u s_u \;=\; m_u(s_u) \\ \text{s.t.} & \|s_u\| \;\leq\; \Delta. \end{array}$$

If we scale the trust region so it has radius one, we have

$$\begin{array}{rl} \min\limits_{s \in \mathbb{R}^n} & c + g^\top s + \frac{1}{2} s^\top H s \;=\; m(s) \\ \text{s.t.} & \|s\| \;\leq\; 1, \end{array}$$

where $s = s_u/\Delta$, $g = \Delta g_u$, and $H = \Delta^2 H_u$.

Note that if s satisfies a fraction of Cauchy-decrease-type condition of the form

$$m(0) - m(s) \;\geq\; \frac{\kappa_{fcd}}{2} \|g\| \min\left\{ \frac{\|g\|}{\|H\|}, 1 \right\}, \tag{11.2}$$

then $s_u = \Delta s$ also satisfies a fraction of Cauchy-decrease-type condition of the form

$$m_u(0) - m_u(s_u) \;\geq\; \frac{\kappa_{fcd}}{2} \|g_u\| \min\left\{ \frac{\|g_u\|}{\|H_u\|}, \Delta \right\}.$$

The Cauchy step itself (defined as the minimizer of $m(s)$ along $-g$ and inside the trust region) satisfies inequality (11.2) with $\kappa_{fcd} = 1$.

We want to show that there exists an approximate solution for the problem

$$\begin{aligned} \min_{s\in\mathbb{R}^n} \quad & m(s) \\ \text{s.t.} \quad & \|s\| \le 1, \\ & |\phi(s)^\top v| \ge \xi > 0, \end{aligned} \tag{11.3}$$

yielding a decrease on $m(s)$ as good as a fixed fraction of the decrease obtained by the Cauchy step. The value of ξ is not yet determined, but it must be bounded away from zero for all problem data considered (meaning v, g, and H).

For a moment let us instead consider the following problem for a fixed $\kappa_{fcd} \in (0,1)$:

$$\begin{aligned} \max_{s\in\mathbb{R}^n} \quad & |\phi(s)^\top v| \\ \text{s.t.} \quad & \|s\| \le 1, \\ & m(0) - m(s) \ge \tfrac{\kappa_{fcd}}{2}\|g\| \min\left\{\tfrac{\|g\|}{\|H\|}, 1\right\}. \end{aligned} \tag{11.4}$$

If we can show that the optimal value for this problem is always above a positive threshold $\xi_* > 0$, for all vectors v such that $\|v\|_\infty \ge 1$ and for all vectors g and symmetric matrices H, then we have shown that there exists an approximate solution for problem (11.3) satisfying a fraction of Cauchy decrease for small enough $\xi > 0$. To do so we need to show that the polynomial $\phi(s)^\top v$ does not nearly vanish on the feasible set of (11.4). The difficulty is that when $\|g\|/\|H\| \to 0$ the feasible set of (11.4) may converge to a singular point, on which a suitably chosen polynomial may vanish. What we will show is that for the problem (11.3) arising during the wedge algorithm it always holds that $0 < g_{min} \le \|g\|/\|H\|$, where $g_{min} \in (0,1)$ is a fixed constant.

Recall that $g = g_u\Delta$ and $H = H_u\Delta^2$, and this implies that

$$\frac{\|g\|}{\|H\|} = \frac{\|g_u\|}{\Delta\|H_u\|}.$$

From the standard assumption on the boundedness of the model Hessian we know that $\|H_u\| \le \kappa_{bhm}$. Recall also that $\Delta \le \Delta_{max}$. The criticality step of the modified wedge algorithm ensures that any time the wedge trust-region subproblem is solved at Step 2 we have $\|g_u\| \ge \min\{\epsilon_c, \mu^{-1}\Delta\}$, and

$$\frac{\|g\|}{\|H\|} = \frac{\|g_u\|}{\Delta\|H_u\|} \ge g_{min}, \tag{11.5}$$

where

$$g_{min} = \min\left\{\frac{\epsilon_c}{\Delta_{max}\kappa_{bhm}}, \frac{1}{\mu\kappa_{bhm}}\right\}.$$

Now we prove that is possible to achieve good uniform geometry and, simultaneously, a fraction of Cauchy decrease for all g and H such that $g_{min}\|H\| \le \|g\|$, with $g_{min} \in (0,1)$.

Theorem 11.1. *Let $g_{min} \in (0,1)$ be a given constant. There exists a positive constant ξ_* depending on both g_{min} and κ_{fcd} such that the optimal value of* (11.4) *satisfies*

$$\xi(v; g, H) \ge \xi_* > 0$$

for all vectors $v \in \mathbb{R}^p$ such that $\|v\|_\infty \geq 1$ and for all vectors $g \in \mathbb{R}^n$ and symmetric matrices $H \in \mathbb{R}^{n\times n}$ such that

$$g_{min} \leq \frac{\|g\|}{\|H\|}.$$

Proof. We start by defining

$$a(v;g,H) = \max\left\{|\phi(s)^\top v| : \|s\| \leq 1, \quad m(0)-m(s) \geq \frac{\kappa_{fcd}}{2}\|g\|\right\}$$

when $\|g\| \geq \|H\|$ and

$$b(v;g,H) = \max\left\{|\phi(s)^\top v| : \|s\| \leq 1, \quad m(0)-m(s) \geq \frac{\kappa_{fcd}}{2}\frac{\|g\|^2}{\|H\|}\right\}$$

when $g_{min}\|H\| \leq \|g\| \leq \|H\|$. The optimal value of (11.4) is given by

$$\xi(v;g,H) = \begin{cases} a(v;g,H) & \text{when } \|g\| \geq \|H\|, \\ b(v;g,H) & \text{when } g_{min}\|H\| \leq \|g\| \leq \|H\|. \end{cases}$$

We will prove that the following minimum is attained:

$$\min_{(v;g,H):\|v\|_\infty\geq 1,\ \|g\|\geq\|H\|} a(v;g,H) > 0.$$

We can perform the change of variables $g' = g/\|g\|$ and $H' = H/\|g\|$. Thus, minimizing $a(v;g,H)$ in g and H, with $\|g\| \geq \|H\|$, is equivalent to minimizing $a'(v;g',H')$ in g' and H', with $\|g'\| = 1$ and $\|H'\| \leq 1$, where

$$a'(v;g',H') = \max\left\{|\phi(s)^\top v| : \|s\| \leq 1, \quad -{g'}^\top s - \frac{1}{2}s^\top H' s \geq \frac{\kappa_{fcd}}{2}\right\}.$$

Since $-s^\top H' s \geq -\|s\|^2$ when $\|H'\| \leq 1$ and $\kappa_{fcd} < 1$, it is always possible, for any g' and H' such that $\|g'\| = 1$ and $\|H'\| \leq 1$, to find s (by setting $s = -tg'$ with t slightly less than 1) such that both constraints that define $a'(v;g',H')$ are satisfied strictly. Thus, the feasible set always has an interior; hence a ball can be inscribed in this feasible region, whose positive radius depends continuously on g and H. Since g and H range over a compact set, so does the radius of the inscribed ball. Hence, there exists a smallest positive radius r_a^* such that the feasible set that defines $a'(v;g',H')$ always contains a ball of that radius, say $B(x;r_a^*)$, for some $x \in B(0;1)$ dependent on g' and H'.

Similarly to the proof of Lemma 3.10 we can show that there exists $\xi_*^a > 0$ such that

$$\min_{x\in B(0;1)} \max_{s\in B(x;r_a^*)} |v^\top \phi(s)| \geq \xi_*^a,$$

by claiming that the above quantity is a norm on v. Hence, $a'(v;g',H') \geq \xi_*^a$ for any v, g', and H' such that $\|v\|_\infty \geq 1$, $\|g'\| = 1$, and $\|H'\| \leq 1$.

We also need to prove that the following minimum is attained:

$$\min_{(v;g,H):\|v\|_\infty\geq 1,\ g_{min}\|H\|\leq\|g\|\leq\|H\|} b(v;g,H) > 0.$$

Once again, the structure of the problem appears to be favorable since $m(0) - m(s) \geq (\kappa_{fcd}/2)\,\|g\|^2/\|H\|$ is equivalent to

$$-\left(\frac{g}{\|H\|}\right)^\top s - \frac{1}{2} s^\top \left(\frac{H}{\|H\|}\right) s \;\geq\; \frac{\kappa_{fcd}}{2} \frac{\|g\|^2}{\|H\|^2}.$$

The change of variables in this case is $g' = g/\|H\|$ and $H' = H/\|H\|$. Thus, minimizing $b(v; g, H)$ in v, g, and H, with $g_{min}\|H\| \leq \|g\| \leq \|H\|$, is equivalent to minimizing $b'(v; g', H')$ in v, g', and H', this time with $g_{min} \leq \|g'\| \leq 1$ and $\|H'\| = 1$, where

$$b'(v; g', H') \;=\; \max \left\{ |\phi(s)^\top v| : \|s\| \;\leq\; 1, \quad -{g'}^\top s - \frac{1}{2} s^\top H' s \;\geq\; \frac{\kappa_{fcd}}{2} \|g'\|^2 \right\}.$$

Since $-s^\top H' s \geq -\|s\|^2$ when $\|H'\| = 1$ and $\kappa_{fcd} < 1$, it is always possible, for any g' and H' such that $g_{min} \leq \|g'\| \leq 1$ and $\|H'\| = 1$, to find s (by setting $s = -t g'$ with t slightly less than 1) such that both constraints that define $b'(v; g', H')$ are satisfied strictly. Thus, the feasible set of the problem that defines $b'(v; g', H')$ always has an interior. Hence there is a smallest radius r_b^* such that the feasible region always contains a ball of at least such a radius. Repeating the arguments for $a'(v; g', H')$ we conclude that there exists $\xi_*^b > 0$ such that $b'(v; g', H') \geq \xi_*^b$ for all v, g', and H', such that $\|v\|_\infty \geq 1$, $g_{min} \leq \|g'\| \leq 1$, and $\|H'\| = 1$.

The value of ξ_* that we are looking for is therefore

$$\xi_* \;=\; \min \left\{ \xi_*^a, \xi_*^b \right\} \;>\; 0. \quad \square$$

Validity of the wedge algorithm for the second-order case

To be able to prove global convergence to second-order critical points for the wedge method one first of all needs to strengthen the fraction of Cauchy decrease requirement by the fraction of the eigenstep decrease requirement. Recalling the scaled and unscaled trust-region subproblems, we can see that if s satisfies a fraction of eigenstep-decrease-type condition of the form

$$m(0) - m(s) \;\geq\; \frac{\kappa_{fed}}{2} \max\{-\lambda_{\min}(H), 0\}, \tag{11.6}$$

then $s_u = \Delta s$ also satisfies a fraction of eigenstep-decrease-type condition of the form

$$m_u(0) - m_u(s_u) \;\geq\; \frac{\kappa_{fed}}{2} \max\{-\lambda_{\min}(H_u), 0\}\, \Delta^2.$$

When $\lambda_{min}(H) \geq 0$, any step s such that $m(0) \geq m(s)$, and in particular the Cauchy step, satisfies inequality (11.6) for $\kappa_{fed} > 0$. When $\lambda_{min}(H) < 0$, the eigenstep (defined as an eigenvector u of H associated with the minimal eigenvalue of H, appropriately scaled such that $\|u\| = 1$ and $u^\top g \leq 0$) satisfies inequality (11.6) with $\kappa_{fed} = 1$.

The analogue of problem (11.4) for the second-order case is

$$\begin{array}{ll} \max\limits_{s \in \mathbb{R}^n} & |\phi(s)^\top v| \\ \text{s.t.} & \|s\| \;\leq\; 1, \\ & m(0) - m(s) \;\geq\; \frac{\kappa_{fcd}}{2} \|g\| \min\left\{ \frac{\|g\|}{\|H\|}, 1 \right\}, \\ & m(0) - m(s) \;\geq\; \frac{\kappa_{fed}}{2} \max\{-\lambda_{min}(H), 0\}, \end{array} \tag{11.7}$$

with $\kappa_{fcd}, \kappa_{fed} \in (0, 1)$.

We can show an extended result of Theorem 11.1 for the second-order case.

Theorem 11.2. *Let $\sigma_{min} \in (0,1)$ be a given constant. There exists a positive constant ξ_* depending on σ_{min}, κ_{fcd}, and κ_{fed} such that the optimal value of* (11.7) *satisfies*

$$\xi(v;g,H) \geq \xi_* > 0$$

for all vectors $v \in \mathbb{R}^p$ such that $\|v\|_\infty \geq 1$ and for all vectors $g \in \mathbb{R}^n$ and symmetric matrices $H \in \mathbb{R}^{n\times n}$ such that

$$\sigma_{min} \leq \max\left\{\frac{\|g\|}{\|H\|}, \frac{-\lambda_{min}(H)}{\|H\|}\right\}. \tag{11.8}$$

However, this result is not enough anymore because, unlike the first-order case, there is no guarantee that the bound σ_{min} in Theorem 11.2 exists. We know that the scaled quantity $\|g\|/\|H\|$ is bounded away from zero in the first-order case due to the criticality step. In the second-order case, what we obtain from the criticality step is

$$\max\left\{\frac{\|g\|}{\|H\|}, \frac{1}{\Delta}\frac{-\lambda_{min}(H)}{\|H\|}\right\} = \frac{\sigma_u^m}{\Delta\|H_u\|} \geq \min\left\{\frac{\epsilon_c}{\Delta_{max}\kappa_{bhm}}, \frac{1}{\mu\kappa_{bhm}}\right\}. \tag{11.9}$$

Unfortunately, the boundedness away from zero of the quantity involving Δ, given by $\max\{\|g\|/\|H\|, -\lambda_{\min}(H)/(\Delta\|H\|)\}$, does not imply the boundedness away from zero of $\max\{\|g\|/\|H\|, -\lambda_{\min}(H)/\|H\|\}$, which is what we need in Theorem 11.2. When Δ is small, it is possible that both $\|g\|/\|H\|$ and the most negative eigenvalue of $H/\|H\|$ approach zero. In this case, the part of the trust region where a fraction of eigenstep decrease can be obtained may shrink to a region of empty interior. This means that a polynomial may nearly vanish on such a region and, hence, there is no threshold on the pivot value that can be guaranteed to be achievable. Therefore, there might be no step computed in Step 2 of Algorithm 11.4 for which a fraction of eigenstep decrease is attained.

To show this in more detail let us assume that after the criticality step one has

$$g_u = \begin{bmatrix} 0 \\ \Delta^{\frac{3}{2}} \end{bmatrix} \quad \text{and} \quad H_u = \begin{bmatrix} -\Delta^{\frac{1}{2}} & 0 \\ 0 & 1 \end{bmatrix}.$$

We can see that $\Delta \leq \mu\sigma_u^m$ holds for sufficiently small Δ, since

$$\Delta \leq \mu \max\{\|g_u\|, -\lambda_{min}(H_u)\} = \mu\Delta^{\frac{1}{2}}.$$

Since $\|H_u\| \leq \max\{\Delta_{max}^{\frac{1}{2}}, 1\}$, we see that the model Hessian is bounded. Now, when we consider the scaled quantities $\|g\|$ and $\|H\|$, we obtain, for $\Delta \leq 1$,

$$\frac{\|g\|}{\|H\|} = \frac{\|g_u\|}{\Delta\|H_u\|} = \Delta^{\frac{1}{2}}$$

and

$$\frac{-\lambda_{min}(H)}{\|H\|} = \Delta^{\frac{1}{2}},$$

and therefore the right-hand side of (11.8) converges to zero when Δ approaches zero. Hence, the current criticality step does not guarantee the existence of a positive lower bound on $\max\{\|g\|/\|H\|, -\lambda_{min}(H)/\|H\|\}$. Notice that if the Hessian of f is nonsingular at optimality, then so are all the model Hessians close to optimality and this situation does not occur. For the situation of a singular Hessian, a different criticality step may need to be devised that can provide necessary bounds, while maintaining all other properties required for global convergence.

11.5 Other notes and references

The DFO algorithm described in [59, 61] was implemented by Scheinberg as an open source package. UOBYQA [189] and NEWUOA [192] were implemented by Powell and are distributed by the author. WEDGE is Marazzi's MATLAB® [1] implementation of the Wedge method [163] and is available for free download. Berghen [33] and Berghen and Bersini [34] have implemented a version of the UOBYQA algorithm of Powell [189] in a parallel environment. Their code is called CONDOR. The web addresses of these packages are given in the appendix. The paper [34] also contains numerical comparisons among UOBYQA, CONDOR, and DFO. Uğur et al. [221] report numerical results of CONDOR and DFO on a practical application.

Fasano, Morales, and Nocedal [89] studied the performance of a trust-region interpolation-based algorithm that dispenses with any control of the geometrical positions of the sample points. They reported good numerical results compared to NEWUOA when using, in both algorithms, quadratic models built by $(n+1)(n+2)/2$ points. It seems that, although the condition number of the matrices of the interpolation systems grows considerably during the course of the iterations, it tends to stabilize at tolerable levels and the algorithm is able to make progress towards the solution. However, further research is needed to better understand such a behavior, and more testing is necessary, in particular, for the more practical scenario where the number of points for model building is linear in n.

The area is still quite active, and new papers are likely to appear soon (see, e.g., Custódio, Rocha, and Vicente [69] and Wild [225]).