If the preconditioner row $Y_1 = [0 - 1, 1]$ is used, the resulting equation is

$$[-0.8, -0.2]x_1 = -1,$$

whence $x_1 \in \tilde{\boldsymbol{x}}_1 = [1.25, 5]$. Since $\tilde{\boldsymbol{x}}_1 \cap \boldsymbol{x}_1 = \emptyset$, this computation proves that there are no feasible points X, $C(X) = 0$ in $\boldsymbol{X}$.

5.2.5 The Fritz John Conditions

In certain circumstances, we wish to simultaneously verify feasibility and local optimality. For instance, a conventional floating point constrained optimizer may have been used to obtain an approximate global optimum $\check{X}$. We may then wish to verify uniqueness of a critical point in as large a box $\boldsymbol{X}$ as possible about $\check{X}$, so that $\boldsymbol{X}$ may be excluded from further consideration in the exhaustive search[9]. In other cases, we wish to use interval Newton iteration to eliminate boxes $\boldsymbol{X}$ with $\boldsymbol{X} \cap \boldsymbol{N}(F; \boldsymbol{X}, \check{X}) = \emptyset$. In these cases, the system of equations to be used in the interval Newton method must therefore embody the necessary conditions for constrained optima. Such general conditions, not requiring "constraint qualification" assumptions, are the Fritz John conditions.

The Fritz John conditions have been advocated by Hansen *et al.* in [77] and [79]. Their use is thoroughly explained in those works, although empirical results are lacking. Here, we present and discuss first a variant we have found good, then a full variant that includes the bound constraints. That is, we give the function F and derivative matrix ∇F corresponding to the Fritz John conditions for our formulation in Equation (1.3).

The variables in our system are $X = (x_1, \ldots, x_n)$, $V = (v_1, \ldots, v_m)$, and u_0, for a total of $n + m + 1$ variables. We will write $W = (X, u_0, V)$. The function corresponding to the gradient of the Lagrange function is:

$$F(W) = \begin{pmatrix} u_0 \nabla\phi(X) + \sum_{i=1}^{m} v_i \nabla c_i(X) \\ c_1(X) \\ \vdots \\ c_m(X) \\ \left(u_0 + \sum_{i=1}^{m} v_i^2\right) - 1 \end{pmatrix} = 0, \tag{5.3}$$

The v_i are unconstrained and represent Lagrange multipliers for the equality constraints $c_i = 0$. The last equation is a normalization condition.

By not including the bound constraints $\underline{x}_{i_j} \le x_{i_j}$ and $x_{i_j} \le \overline{x}_{i_j}$ in this function, we reduce the size of the system by q. Furthermore, it is more flexible to

[9] This is done in the unconstrained case in [28] and for general nonlinear systems in [114].

include the bound constraints through the process reviewed in §5.2.3. Thus, Equation (5.3) is applicable for points X when none of the bound constraints are active. When one or more bound constraints are active, an analogue of Equation (5.3) in the appropriate lower-dimensional subspace is used. The technique also avoids singularities in the Jacobi matrix of the Fritz–John function that often occur when bound constraints are included.

The Jacobi matrix of F, used to form the matrix $\boldsymbol{A}$ for interval Newton methods, is:

$$H(W) = \left(\begin{array}{c|c|ccc} u_0 \nabla^2 \phi(X) + \sum_{i=1}^{m} v_i \nabla^2 c_i(X) & \nabla\phi(X) & \nabla c_1(X) & \ldots & \nabla c_m(X) \\ \hline (\nabla c_1(X))^T & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ (\nabla c_m(X))^T & 0 & 0 & \ldots & 0 \\ \hline 0 & 1 & 2v_1 & \ldots & 2v_m \end{array} \right). \quad (5.4)$$

The last equation in (5.3) is a normalization condition that can be chosen in various ways. In particular, Hansen and Walster [79] recommend that it be chosen as

$$\left(u_0 + \sum_{i=1}^{m} v_i [1, 1+\epsilon] \right) - 1, \quad (5.5)$$

where ϵ is on the order of the computational precision. If this is done, the last row of the matrix H in (5.4) becomes

$$\left(\begin{array}{c|c|ccc} 0 & 1 & [1, 1+\epsilon], & \ldots & [1, 1+\epsilon] \end{array} \right). \quad (5.6)$$

If Hansen slopes (of §1.3.2 on page 30) are then used to compute an interval analogue of the matrix (5.4), the "v" variables do not appear as intervals. Thus, if Gaussian elimination is used to bound the solution set to the corresponding preconditioned interval linear system, initial bounds on the Lagrange multipliers u_0 and v_i are not required. However, some of our experiments [121] indicate that the search algorithm is more efficient with the quadratic normalization in (5.4).

In theory, the bound constraints can also be included in the Lagrange function. If this is done, (5.3) becomes

$$F(W) = \begin{pmatrix} u_0 \nabla \phi(X) + \sum_{j=1}^{q} u_j \nabla p_j + \sum_{i=1}^{m} v_i \nabla c_i(X) \\ u_1 p_1 \\ \vdots \\ u_q p_q \\ c_1(X) \\ \vdots \\ c_m(X) \\ \left(u_0 + \sum_{j=1}^{q} u_j + \sum_{i=1}^{m} v_i^2\right) - 1 \end{pmatrix} = 0, \tag{5.7}$$

where $u_i \geq 0$, $0 \leq i \leq q$, where q is the total number of bound constraints, and $p_j(X) = \underline{x}_{i_j} - x_{i_j}$ for $1 \leq j \leq \mu$, $p_j(X) = x_{i_j} - \overline{x}_{i_j}$ for $\mu + 1 \leq j \leq q$; see, for example, [60, §3.4.2]. Thus, $\nabla p_j = E_{i_j}$ for $1 \leq j \leq \mu$ and $\nabla p_j = -E_{i_j}$ for $\mu + 1 \leq j \leq q$, where E_i is the i-th coordinate vector.

The analogue to the matrix (5.4) is thus

$$\left(\begin{array}{c|c|ccc|ccc} \begin{matrix} u_0 \nabla^2 \phi(X) + \sum_{j=1}^{q} u_j \nabla^2 p_j \\ + \sum_{i=1}^{m} v_i \nabla^2 c_i(X) \end{matrix} & \nabla \phi(X) & \nabla p_1 & \cdots & \nabla p_q & \nabla c_1(X) & \cdots & \nabla c_m(X) \\ \hline u_1 (\nabla p_1)^T & 0 & p_1 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \cdots & \vdots \\ u_q (\nabla p_q)^T & 0 & 0 & 0 & p_q & 0 & \cdots & 0 \\ \hline (\nabla c_1(X))^T & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ (\nabla c_m(X))^T & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \hline 0 & 1 & 1 & \cdots & 1 & 2v_1 & \cdots & 2v_m \end{array}\right). \tag{5.8}$$

For example, we may plug in the coordinates $x_1 = 0.5$, $x_2 = 1$ for the optimizer for Example 5.2 into (5.7), to get $u_1 = u_2 = u_3 = 0$, $v_1 = \frac{13-\sqrt{173}}{2}$, $u_0 = -4v_1$, and $u_4 = -9v_1$. The matrix (5.8) corresponding to these values becomes approximately

$$\left(\begin{array}{cc|c|cccc|c} 0 & 0 & -1 & -1.0 & 0 & 1.0 & 0 & -4.0000 \\ 0 & -0.6118 & -2 & 0 & -1 & 0 & 1 & 1.0000 \\ \hline 0 & 0 & 0 & -1.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.5 & 0 & 0 \\ 0 & -0.6883 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline -4 & 1.0000 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 1.0 & 1 & 1.0 & 1 & -0.1529 \end{array}\right),$$

with condition number (in the 2-norm) of approximately 11.8.

Method	# Boxes	CPU time
`RUN_ROOTS_DELETE`	941	20.21
`RUN_GLOBAL_OPTIMIZATION` with (5.7)	10	3.02
`RUN_GLOBAL_OPTIMIZATION` with (5.3)	11	0.16

Table 5.3 Summary of 3 methods of handling constraints for Example 5.2

When dealing with the system (5.7), initial bounds of $[0, 1]$ may be used for u_j, $0 \le j \le q$, and bounds of $[-1, 1]$ may be used for v_i, $1 \le i \le m$. When these bounds were used for Example 5.2 in the program `RUN_ROOTS_DELETE` (cf. §4.4 on page 159), three critical points (including the two optima) were found with 941 boxes (in 8-dimensional space) considered and 20.21 CPU seconds on a Sun Sparc 20 model 51. When the general optimization code `RUN_GLOBAL_OPTIMIZATION` (see §5.3 below), configured to use (5.7) in interval Newton iteration, was used, then both optimizers were found with a total of 10 boxes (in 2-dimensional space, but an 8-dimensional interval Newton method). If `RUN_GLOBAL_OPTIMIZATION` was configured to use (5.3) instead of (5.7), a total of 11 boxes (in 2 dimensions, but a 4-dimensional interval Newton method) were considered, in only about 5% of the CPU time. These results are summarized in Table 5.3.

Additional experiments in [121] corroborate the above: It seems to usually be better not to include the bound constraints in the Fritz John function. These results may be different in high-dimensional problems in which the "peeling" process of §5.2.3 leads to a combinatorial explosion in the number of boxes.

In verifying points obtained by floating point constrained optimization software, approximate Lagrange multiplier values may be available along with the approximate optimizers.

5.2.6 Exercises

1. Draw a picture analogous to Figure 5.5 for the problem:

$$\begin{array}{rrcl} \text{minimize} & \phi(x) & = & -\left(x_1^2 + x_2^2\right) \\ \text{subject to} & c(x) & = & x_2 - x_1 = 0, \\ & -1 \le \; x_1 & \le 1 & \end{array}$$