

# DONLP2 short users guide

P. SPELLUCCI  
Technische Universität Darmstadt  
Schloßgartenstraße 7, D64289 Darmstadt  
Germany  
email: spellucci@mathematik.tu-darmstadt.de

May 6, 1999

## Abstract

This paper gives a short introduction into the design and the usage of the nonlinear programming code DONLP2. Only information pertinent to the users of the AMPL interface is given here. A more detailed description of the internal organization of the code, e.g. for users which might want to modify the code according to their needs, can be obtained from the sites which provide source code, e.g. netlib in directory opt/donlp2.

## 1 INTRODUCTION

DONLP2 implements a variant of the SQP-method. It is designed to solve problems of the general form

$$\text{NLP: } f(x) \stackrel{!}{=} \min_{x \in \Omega}, \quad (1.1)$$

where

$$\Omega = \{x \in \mathbb{R}^n : h(x) = 0 \in \mathbb{R}^p, g(x) \geq 0 \in \mathbb{R}^m\}. \quad (1.2)$$

(In the following description we assume  $\Omega \neq \emptyset$  of course. If this does not hold, then the code will terminate near a stationary point of a penalty term constructed from the constraints, see below.) It is assumed that  $f$ ,  $h$  and  $g$  are defined on an open superset of  $\Omega$  and two times continuously differentiable there. Given some regularity conditions, the local solutions of NLP can be characterized by the well known Kuhn-Tucker conditions

$$\begin{aligned} \nabla f(x^*) - \nabla g(x^*)\lambda^* - \nabla h(x^*)\mu^* &= 0, \quad \lambda^* \geq 0, \quad x^* \in \Omega, \\ \lambda_i^* g_i(x^*) &= 0, \quad i = 1, \dots, m. \end{aligned} \quad (1.3)$$

This is the case e.g. if for every  $x \in \Omega$  Mangasarian-Fromowitz condition

$$\text{rank}(\nabla h(x)) = p, \quad \nabla h(x)^T z = 0, \quad \nabla g_{\mathcal{A}}(x)^T z > 0 \quad \text{for some } z \in \mathbb{R}^n, \quad (1.4)$$

holds, where

$$\mathcal{A} = \mathcal{A}(x) = \{i : g_i(x) = 0\}.$$

The usual SQP methods compute a correction to a given approximation  $x^k$  of a local solution of NLP by solving a QP problem

$$\varphi_k(d) := \nabla f(x^k)^T d + \frac{1}{2} d^T B_k d \stackrel{!}{=} \min_{d \in S_k}, \quad (1.5)$$

where

$$S_k = \{d \in \mathbb{R}^n : \nabla h(x^k)^T d + h(x^k) = 0, \quad \nabla g(x^k)^T d + g(x^k) \geq 0\}.$$

Here,  $B_k$  is usually chosen symmetric positive (semi)definite. Finally  $x^{k+1} = x^k + \sigma_k d^k$  for a suitably chosen stepsize  $\sigma_k > 0$ .  $\sigma_k$  is computed by a descent test for a so called merit function. This implementation uses an exact penalty function as a merit function, namely the  $l_1$ -penalty function

$$\Phi(x; \vec{\beta}, \vec{\gamma}) = f(x) - \sum_{i=1}^m \beta_i \min\{0, g_i(x)\} + \sum_{j=1}^p \gamma_j |h_j(x)|. \quad (1.6)$$

Sufficient conditions for exact penalization by this function are

**H:**

For some  $\tau_0 > 0$

$$\Omega(\tau_0) := \{x \in \mathbb{R}^n : \psi(x) \leq \tau_0\}$$

is compact, where

$$\psi(x) := \|h(x)\|_1 + \|g(x)^-\|_1$$

and there holds the extended Mangasarian-Fromowitz-condition:

$$\text{for every } x \in \Omega(\tau_0) : \quad \text{rank}(\nabla h(x)) = p \quad \text{and both} \quad \nabla h(x)^T z = 0 \quad \text{and} \quad \nabla g_{\mathcal{V}}(x)^T z > 0 \quad \text{for some } z \in \mathbb{R}^n, \quad (1.7)$$

where

$$\mathcal{V} = \mathcal{V}(x) = \{i : g_i(x) \leq 0\}. \quad (1.8)$$

Under H the  $l_1$ -penalty function  $\Phi$  (1.6) is an exact one on  $\Omega(\tau_0)$ , if the  $\beta_i$ 's and  $\gamma_j$ 's are large enough (see e.g. Spellucci, [5]). It is easy to show that assumption (1.7) implies that for  $x \in \Omega(\tau_0)$  the set  $M_0(x)$  of abnormal multipliers in Clarke's sense [2] satisfies  $M_0(x) = \{0\}$  on  $\Omega(\tau_0)$ . This is the weakest known condition for proving exact penalizability of NLP based on properties of  $g$  and  $h$ , see e.g. Burke, [1]. It is, of course, a rather stringent one. DONLP2 does not require this condition in order of being well-defined. It often works successfully in cases, where it is not. In order to prove its global convergence to a (feasible) stationary point however (1.7) has been used. The SQP-approach as stated above requires the evaluations of all constraint gradients for any  $x^k$ . DONLP2 tries to avoid this by estimating a working set for any  $x^k$ . This includes the equality constraints and those inequality constraints  $g_i$  for which

$$g_i(x^k) \leq \delta_k \max\{1, \|\nabla g_i(x^k)\|\}$$

where  $\delta_k$  is computed adaptively. The parameter `del0` which may be set by the user is used as an upper bound for  $\delta_k$ , but during iteration  $\delta_k$  is bounded from above and from below by  $\mathcal{O}(\|x^k - x^*\|^\alpha)$  for some  $\alpha$ ,  $0 < \alpha < 1$ . Clearly increasing `del0` increases the working set in the early steps. This might improve the quality of the directions, but also increases the computational effort. As long as the gradients of the constraints in the working set are linearly independent, DONLP2 avoids to solve an inequality constrained QP. Rather it computes a direction of descent for  $\Phi$  by solving an equality constrained one, i.e. a system of linear equations, using a suitable modification of the current constraint values and the gradient of the objective as right hand side. This linear system has the form

$$\begin{pmatrix} B & N_{\mathcal{A}} \\ N_{\mathcal{A}}^T & 0 \end{pmatrix} \begin{pmatrix} d \\ (-\mu) \\ (-\lambda_{\mathcal{A}}) \end{pmatrix} = - \begin{pmatrix} g \\ c \end{pmatrix}, \quad (1.9)$$

where  $\mathcal{A} = \mathcal{A}(x, \delta)$  describes the working set,  $N$  is made up from the gradients of the constraints and  $g = \nabla f(x)$ . If the gradients in the working set are linearly dependent or nearly so, then DONLP2 computes the direction of descent from an inequality constrained QP as considered above, however, including constraints from the working set only. Since the QP-problem may be infeasible, a kind of flexible mode as known from LP algorithms is entered by adding slacks to all constraints and penalizing these slacks in the objective function of the QP problem in the hope to drive them to zero. In the following,  $u$  and  $v$  are these slack variables, which make the linearized constraints consistent. Slack variables  $u_i$  and QP-slack-weights  $a_i$  with  $i \in \mathcal{A}$  are involved only in the following. It is silently assumed that  $u_i = 0$  and  $a_i > 0$  (arbitrary) for  $i \notin \mathcal{A}$ . We let

$$\varphi(d, u_{\mathcal{A}}, v) := \nabla f(x)^T d + \frac{1}{2} d^T B d + a_{\mathcal{A}}^T u_{\mathcal{A}} + b^T v + \frac{\alpha}{2} (\|u_{\mathcal{A}}\|^2 + \|v\|^2) \quad (1.10)$$

and

$$S := \left\{ (d, u_{\mathcal{A}}, v) : \begin{array}{l} \nabla h(x)^T d + h(x) + \Theta v = 0, \quad v \geq 0 \\ \nabla g_{\mathcal{A}}(x)^T d + g_{\mathcal{A}}(x) + u_{\mathcal{A}} \geq 0, \quad u_{\mathcal{A}} \geq 0 \end{array} \right\}. \quad (1.11)$$

$\Theta$  is a diagonal matrix with entries

$$\Theta = \text{diag}(\theta_i), \quad \theta_i = 1 \quad \text{if} \quad h_i(x) < 0 \quad \text{and} \quad \theta_i = -1 \quad \text{if} \quad h_i(x) \geq 0. \quad (1.12)$$

We solve QP  $(x, \alpha, B, a_{\mathcal{A}}, b)$ :

$$\varphi(d, u_{\mathcal{A}}, v) = \min_{(d, u_{\mathcal{A}}, v) \in S}. \quad (1.13)$$

If the regularity condition **H** given above is satisfied, then for sufficiently large  $v$  and  $u_{\mathcal{A}}$  zero slacks result.  $d$  is a direction of descent for  $\Phi$  for sufficiently large penalty weights. The QP-problem is solved using the method of Goldfarb and Idnani [3]. Having computed  $d$  and the multiplier estimates the weights of the penalty function  $\Phi$  are adapted appropriately. These weights may be increased or decreased individually. This is done in a manner which guarantees fixed and sufficiently large weights after a finite number of steps. Then, a stepsize  $\sigma_k$  is computed by a combination of interpolation and backtracking which reduces  $\Phi$  sufficiently to insure convergence of  $\{x^k\}$  to a Kuhn-Tucker point. Then, an update for the estimate  $B_k$  of the Hessian of the augmented Lagrangian of the problem with some regularization (the Pantoja-Mayne update) is computed and a new cycle is begun. See the papers with the complete theoretical analysis [6], [7] of the method for details. In the following we restrict ourselves to a discussion of the impact of DONLP2's parameters which can be modified by a user of the AMPL interface

## 2 Concerning the choice of *tau0* and *del0*

The implementation requires  $\text{tau0} = \tau_0$  being specified by the user. Of course, this cannot always be done on account of **H**. But it should be obvious, that  $\tau_0$  must be chosen such that any of  $f$ ,  $g$  and  $h$  can be evaluated safely in  $\Omega(\tau_0)$ . As a simple academic example assume that  $n = 2$ ,  $m = 2$ ,  $p = 0$  and

$$\begin{aligned} f(x) &= -\exp(x_1 x_2 - 1) / (x_1 x_2 - \frac{1}{2}), \\ g(x) &= (x_1 x_2 - \frac{3}{4}, -(x_2)^2 + 2 - x_1)^T. \end{aligned}$$

For  $\tau_0 = \frac{1}{4}$  there exist points in  $\Omega(\tau_0)$  with  $x_1 x_2 - \frac{1}{2} = 0$ , that means  $f(x)$  undefined. Moreover,  $f$  is not bounded from below on the interior of  $\Omega(\frac{1}{4})$ . If  $x^0 = (\xi, \xi)^T$  with  $\xi > 1$  then we might obtain a direction  $d^0$  which results in violation of  $x^0 + d^0 \in \Omega(\frac{1}{4} - \epsilon)$ , that is, without introduction of the safeguard to stay "near" the feasible set the method might fail. If a problem has a nonlinear equality constraint

$$h(x) = \sum_{i=1}^n (x_i)^2 - 1 = 0,$$

then it is obvious that one should have  $\tau_0 < 1$  since otherwise the point  $x = 0$  with  $\nabla h(x) = 0$  would be an element of  $\Omega(\tau_0)$ , violating **H**. Indeed, such cases cause trouble in practice. If no such problems occur, then a user might choose  $\text{tau0}$  of the order of one or greater. If it turns out that **H** is not satisfied with  $\delta = \delta_0$  then a reduction of  $\text{tau0}$  might help. As a rule of thumb, reducing  $\text{tau0}$  increases the robustness of the code but makes it more and more inefficient as well. If the initial guess is not in  $\Omega(\tau_0)$ , then  $f$  is replaced by  $f = 0$  and the iteration proceeds until a point in  $\Omega(\tau_0)$  is found. Usually this works fine. It may however be disastrous in some cases, where a steep gradient of the objective function might drive the iteration into a region where the problem behaves nicely in the very first steps, whereas the point found as above might be poor.

In most cases  $\text{del0} = \delta_0 = 1$  will be a reasonable choice. If one has finite lower and upper bounds for a variable,  $\alpha \leq x_1 \leq \beta$  (say), then  $\delta_0 < (\beta - \alpha)/2$  is required in order to satisfy **H**, of course. The

present implementation corrects the value of *del0* automatically to fit such requirements. If the strict complementarity condition holds, a small value of *del0* will cause no problem, but otherwise it might delay identification of the correct working set. A large *del0* might give a working set which violates the linear independence condition which results in a (sometimes tremendous) increase of computational complexity of the run. This should be avoided. Typically this situation arises for finely discretized continuous problems, e.g. control problems, robot design, variational inequalities. Here *del0* should be decreased with decreasing discretization parameter and some experimentation might be necessary to optimize the performance of the code.

### 3 Concerning the role of *nreset*

DONLP2 has some automatic restart features implemented, in the hope to overcome problems which might result from illconditioned quasi-Newton updates, too large penalty weights or highly nonconvex local behaviour of  $\Phi$ . Some of these are governed by the variable *nreset*. In addition, premature termination of the run is controlled choosing it. These features are the following:

- 1 A step is called a "step with small change in  $x$ ", if

$$|x_i^{k+1} - x_i^k| \leq \varepsilon_x \left( |x_i^k| + \frac{1}{100} \right) \forall i$$

DONLP2 admits *nreset* successive such steps, then terminating prematurely with an appropriate message.  $\varepsilon_x$  is to be chosen by the user.

- 2 If the BFGS-update for the ordinary Lagrangian was not well defined for *nreset* successive steps or there had been *nreset* successive steps with small stepsizes

$$\sigma_k \leq 0.05,$$

then a reinitialization of the quasi-Newton update  $B_k$  is issued, using an appropriate multiple of the unit matrix.

The default for *nreset* is  $n$ , the primal dimension. For larger values of  $n$  a smaller choice might be advisable, say  $nreset = \min\{n, 20\}$ . *nreset* is bounded from below by 4 automatically.

### 4 Concerning the choice of *maxit*

Much numerical experimentation has shown that  $30n$  should be a reasonable upper bound for the number of steps. For singular problems this is not true however. The present implementation uses 4000 as the maximum possible number of steps. Together with the hot-start feature of the code this poses no limitation. The AMPL interface in its present form does not support the hot start feature. Hence the iteration limit might be set to its maximum allowable value.

## 5 Control of output

DONLP2 allows the user a lot of control concerning the volume of output. Six logical variables are used for this purpose. The user of the AMPL interface sets these via the variables *silent*, *intakt* and *outlev*. The variables are

- 1 *intakt*. If true, then any output to the protocol-file *out.pro* is sent also to *std-out*.
- 2 *silent*. If true, no output files are generated and the user must extract the results desired from DONLP2's data area. The AMPL interface does this under control of the *wantsol* variable.
- 3 *te0*. If true, then a short information of one line is sent to *std-out* for every step performed. this looks like:

```

1 FX= .0000000D+00 UPSI= .53D+00 B2N= -.10D+01 UMI= .00D+00 NR 4 SI-1
2 FX= .0000000D+00 UPSI= .11D+00 B2N= -.10D+01 UMI= .00D+00 NR 4 SI-1
3 FX= .5577251D+03 UPSI= .41D-01 B2N= .52D+03 UMI= .00D+00 NR 4 SI-1
4 FX= .1500269D+04 UPSI= .23D-01 B2N= .45D+04 UMI= -.89D+04 NR 4 SI-1
5 FX= .1525384D+04 UPSI= .18D-01 B2N= .44D+04 UMI= -.30D+05 NR 4 SI-1
.....

```

Here *FX* is the value of the objective  $f$ , *UPSI* is the value of the l1-norm of the constraint violations, *B2N* is the Euclidean norm of the gradient of the Lagrangian (with respect to  $x$ ), *UMI* is the most negative multiplier or zero, *NR* is the cardinality of the working set and *SI* equals -1 for a regular working set and 1 otherwise. The combination of *FX*= .000000D+00 and *B2N*= -.10D+01 indicates the presence of the infeasibility improvement phase of the method.

- 4 *te1*. If true, then a short protocol of the run is sent to the protocol file , summarizing the most important data for each step. This looks like

```

1 FX= .0000000E+00 SCF= .0000000E+00 PSI= .5260381E+00 UPS= .5260381E+00
  DEL= .5000000E-05 B20= .0000000E+00 B2N= -.1000000E+01 NR= 4
  SI= -1 U-= .0000000E+00 C-R= .1141352E+01 C-D= .1000000E+01
  XN= .1587451E+02 DN= .5772724E+01 PHA= -1 CL= 0
  SKM= .0000000E+00 SIG= .1000000E+01 CF+= 0 DIR= -.5260381E+00
  DSC= .1000000E+01 COS= .0000000E+00 VI0= 0
  UPD= 0 TK= .0000000E+00 XSI= .0000000E+00
2 FX= .0000000E+00 SCF= .0000000E+00 PSI= .1082978E+00 UPS= .1082978E+00
  DEL= .5000000E-05 B20= .0000000E+00 B2N= -.1000000E+01 NR= 4
  SI= -1 U-= .0000000E+00 C-R= .1708276E+01 C-D= .1000000E+01
  XN= .1330892E+02 DN= .6223501E+01 PHA= -1 CL= 0
  SKM= .0000000E+00 SIG= .1000000E+01 CF+= 0 DIR= -.1082978E+00
  DSC= .1000000E+01 COS= .5246703E+00 VI0= 0
  UPD= 0 TK= .0000000E+00 XSI= .0000000E+00

```

The meaning of these variables is as follows:

- 1 step-nr.
- FX current value of objective (zero in feasibility improvement phase (-1) ).
- SCF internal scaling of objective (zero in phase -1).
- PSI the weighted penalty-term.
- UPS the unweighted penalty-term (l1-norm of constraint vector).
- DEL bound for currently active constraints ( $\delta_k$ ).

- B20 12-norm of projected gradient, based on constraints in a reduced working set with  $\delta_k$  replaced by  $\delta_{\min}$ .
  - B2N weighted norm of projected gradient based on the full working set with inverse of Cholesky-factor of  $B_k$  as weight.
  - NR cardinality of the working set.
  - SI if 1, the binding constraints do not satisfy the linear independence condition.
  - U- infinity norm of dual infeasibility (most negative multiplier for inequalities).
  - C-R condition number of diagonal part of qr-decomposition of normalized gradients of binding constraints.
  - C-D condition number of diagonal of cholesky-factor of updated full Hessian ( $B_k$ ).
  - XN 12-norm of  $x^k$ .
  - DN 12-norm of  $d^k$  (correction from qp -subproblem, unscaled).
  - PHA -1 : infeasibility improvement phase,  
0: initial optimization phase (varying working set),  
1: binding constraints unchanged ,  
2:  $d$  small, Maratos correction in use.
  - CL total number of steps with a decrease of at least one penalty weight.
  - SKM infinity norm of penalty weights.
  - SIG  $\sigma_k$ , stepsize from unidimensional minimization (backtracking).
  - CF+ number of objective function evaluations during line search.
  - DIR directional derivative of penalty-function along  $d$  times DSC .
  - DSC internal scaling factor for  $d$ .
  - COS the cosine of the angle between  $d^k$  and  $d^{k-1}$ . if larger than the internal variable theta (0.9), stepsizes larger than one (up to the internal variable sigla (2048.)) are tried too, if  $\sigma = 1$  passes the test of descent.
  - VIO number of constraints not binding at  $x^k$ , but hit during line search
  - UPD Type of update for  $B_k$ :  
1: normal Pantoja-Mayne update,  
0: update suppressed (e.g. "y=0"),  
-1= restart with scaled unit matrix ,  
2= standard BFGS with NR=0 ,  
3= BFGS modified by Powells device if NR=0.
  - TK regularization term for damping the projector in the Pantoja- Mayne update or  $(\tilde{y}^k)^T s^k / ((s^k)^T B_k s^k)$  ( in the case UPD=2,3 ).
  - XSI the parameter for the extended Lagrangian in the Pantoja-Mayne update if UPD=1 or the regularization parameter for Powell's modification of BFGS, if UPD=2,3.
  - QPT 0, if SI=-1,  
1: successful termination indicator of qp-solver,  
-1: weight for slacks becomes larger than maximum value allowed ( $10^8$ ) without slack variables becoming sufficiently small ,  
-2: infeasible qp-problem (theoretically impossible).
  - TQP weight of slack-variables in qp-solver.
  - SLA 11-norm of slack-variables in qp-solver. There should hold  $SLA < UPSI$ . Otherwise the penalty term increases in the direction  $d^k$ . For  $UPSI > \tau_0/2$  this results in breakdown.
- 5 *te2*. If true, then a detailed protocol is sent to the protocol file, showing all primal and dual variables, the relevant values from the working set, progress in the backtracking etc. The output looks like

```

=====
                1-TH ITERATION STEP
SCF= .00000D+00 PSIST= .00000D+00 PSI= .31508D+00 UPSI= .31508D+00
FXST= .00000D+00 FX= .00000D+00
X=
.80000D+00 .83000D+00 .85000D+00 .87000D+00 .90000D+00 .10000D+00
.12000D+00 .19000D+00 .25000D+00 .29000D+00 .51200D+01 .13100D+00
.71800D+00 .64000D+01 .65000D+01 .57000D-01
VALID PERMUTATION OF X

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

DEL= .500000D-06 B2N0= .000000D+00 B2N= -.100000D+01 GFN= .000000D+00

VALUES OF RESTRICTIONS
( 39 -.21516D-01 .82663D+01) ( 40 -.11610D+00 .17539D+01)
( 41 -.14338D+00 .23683D+01) ( 43 -.34090D-01 .11756D+01)
( 5 .10000D-07 .10000D+01) ( 22 .10000D-07 .10000D+01)

DIAG(R)=
-.10000D+01 -.86767D+00 -.64439D+00 -.62219D+00 -.99984D+00 .10000D+01
CONDITION NUMBER OF R 1.6072133626688

PHASE= -1 SCF0= .10000D+01
D =
.88915D-03 .51825D-01 -.59878D-01 .28985D-01 -.10000D-07 .10000D-07
-.34826D-01 -.16261D-01 .25319D-01 -.30441D+00 .00000D+00 .46154D-02
.46195D-01 -.34312D-02 -.41399D-01 .24918D-02

START UNIMIN

PHI= .31508D+00 DPHI= -.31508D+00 PSI= .31508D+00 TAU0/2= .45000D-02
FX= .00000D+00 DSCAL= .10000D+01 SCF= .00000D+00 UPSI= .31508D+00
SIG= .10000D+01 FX= .00000D+00 PSI= .24486D+01 UPSI= .24486D+01
SIG= .50000D+00 FX= .00000D+00 PSI= .13597D+01 UPSI= .13597D+01
SIG= .12500D+00 FX= .00000D+00 PSI= .30859D+00 UPSI= .30859D+00

END UNIMIN

SIG= .12500D+00 NUM. F-EVALUATIONS 0
LIST OF INACTIVE HIT CONSTRAINTS
34 37 38 48 50 34 37 38 48 50 34 38 50

```

In a addition to the variables explained already there are GFN, the l2-norm of  $\nabla f(x^k)$  and SCF0, a damping factor for the "horizontal" part of  $d^k$ . The permutation of the components of  $x$  is constructed to make the QR-decomposition of the gradients in the working set continous. Under

VALUES OF RESTRICTIONS the number, the value and the maximum of 1 and the norm of the gradient are listed.

6 *te3*. If true, then in addition the gradients in the working set and the quasi-Newton update are printed.

These parameters are computed from the variable *outlev* as follows: *outlev* is represented as a 4 bit binary with *te0*,...,*te3* being represented by bit 1 through 4. If the bit is set, then the corresponding variable is set to true. *silent* and *intakt* are set directly by the user.

The standard output with *outlev* = 1 is as follows

DONLP2, V1, COPYRIGHT P. SPELLUCCI  
POLYGON

1998 FEB 10 12H 1M 8S

1	FX=	.0000000D+00	UPSI=	.18D+01	B2N=	-.10D+01	UMI=	.00D+00	NR	7	SI-1
2	FX=	-.7323344D+00	UPSI=	.63D+00	B2N=	.21D+00	UMI=	.00D+00	NR	7	SI-1
3	FX=	-.6622516D+00	UPSI=	.27D-01	B2N=	.14D+00	UMI=	.00D+00	NR	7	SI-1
4	FX=	-.6698409D+00	UPSI=	.57D-05	B2N=	.60D-01	UMI=	.00D+00	NR	7	SI-1
5	FX=	-.6713666D+00	UPSI=	.55D-05	B2N=	.21D-01	UMI=	.00D+00	NR	7	SI-1
6	FX=	-.6717718D+00	UPSI=	.21D-05	B2N=	.24D-01	UMI=	.00D+00	NR	7	SI-1
7	FX=	-.6729540D+00	UPSI=	.72D-06	B2N=	.69D-01	UMI=	.00D+00	NR	8	SI-1
8	FX=	-.6739375D+00	UPSI=	.60D-05	B2N=	.58D-01	UMI=	.00D+00	NR	8	SI-1
9	FX=	-.6743592D+00	UPSI=	.21D-08	B2N=	.64D-01	UMI=	.00D+00	NR	8	SI-1
10	FX=	-.6743830D+00	UPSI=	.81D-09	B2N=	.64D-01	UMI=	.00D+00	NR	8	SI-1
11	FX=	-.6744703D+00	UPSI=	.83D-08	B2N=	.67D-01	UMI=	.00D+00	NR	8	SI-1
12	FX=	-.6748709D+00	UPSI=	.40D-05	B2N=	.21D-01	UMI=	.00D+00	NR	8	SI-1
13	FX=	-.6749596D+00	UPSI=	.55D-06	B2N=	.62D-02	UMI=	.00D+00	NR	8	SI-1
14	FX=	-.6749801D+00	UPSI=	.83D-08	B2N=	.15D-02	UMI=	.00D+00	NR	8	SI-1
15	FX=	-.6749814D+00	UPSI=	.89D-10	B2N=	.21D-03	UMI=	.00D+00	NR	8	SI-1
16	FX=	-.6749814D+00	UPSI=	.24D-12	B2N=	.15D-04	UMI=	.00D+00	NR	8	SI-1
17	FX=	-.6749814D+00	UPSI=	.00D+00	B2N=	.33D-06	UMI=	.00D+00	NR	8	SI-1

N= 12 NH= 2 NG= 20

EPSX= .1000D-06 SIGSM= .1490D-07

STARTVALUE

.52359878D+00	.10471976D+01	.15707963D+01	.20943951D+01	.26179939D+01
.31415927D+01	.48979592D+00	.81632653D+00	.97959184D+00	.97959184D+00
.81632653D+00	.48979592D+00			

EPS=	.222D-15	TOL=	.198-322	DELO=	.100D+01	DELM=	.100D-05	TAU0=	.100D+01
TAU=	.100D+00	SD=	.100D+00	SW=	.367D-10	RHO=	.100D-05	RHO1=	.100D-09
SCFM=	.100D+05	C1D=	.100D-01	EPDI=	.100D-07				
NRE=	12	ANAL=	T						

TERMINATION REASON:

KT-CONDITIONS SATISFIED, NO FURTHER CORRECTION COMPUTED

EVALUATIONS OF F	31
EVALUATIONS OF GRAD F	17
EVALUATIONS OF CONSTRAINTS	720
EVALUATIONS OF GRADS OF CONSTRAINTS	107
FINAL SCALING OF OBJECTIVE	.1000000D+01
NORM OF GRAD(F)	.8732285D+00
LAGRANGIAN VIOLATION	.6282064D-08

```

FEASIBILITY VIOLATION .2220446D-15
DUAL FEASIBILITY VIOLATION .0000000D+00
OPTIMIZER RUNTIME SEC'S .4000000E-01

```

OPTIMAL VALUE OF F = -.674981442930105D+00

OPTIMAL SOLUTION X =

```

.422092805482513D+00 .109969388585251D+01 .166621824605869D+01
.197382067847328D+01 .238722802966953D+01 .314159265358979D+01
.641783511852838D+00 .100000000000000D+01 .100000000000000D+01
.779220565062775D+00 .558978622460956D+00 .000000000000000D+00

```

MULTIPLIERS ARE RELATIV TO SCF=1

NR.	CONSTRAINT	NORMGRAD (OR 1)	MULTIPLIER
1	.00000000D+00	.10000000D+01	.00000000D+00
2	.00000000D+00	.10000000D+01	-.13145988D+00
3	.67760108D+00	.14142136D+01	.00000000D+00
4	.56652436D+00	.14142136D+01	.00000000D+00
5	.30760243D+00	.14142136D+01	.00000000D+00
6	.41340735D+00	.14142136D+01	.00000000D+00
7	.75436462D+00	.14142136D+01	.00000000D+00
8	.58811391D+00	.10000000D+01	.00000000D+00
9	.00000000D+00	.24268869D+01	.41592032D-01
10	-.22204460D-15	.24341731D+01	.17188573D+00
11	.00000000D+00	.25313837D+01	.41592032D-01
12	.58811392D+00	.10000000D+01	.00000000D+00
13	.68754291D+00	.10000000D+01	.00000000D+00
14	.39281530D+00	.10000000D+01	.00000000D+00
15	.00000000D+00	.23376604D+01	.16013124D+00
16	.00000000D+00	.21963579D+01	.99649164D-01
17	.87810669D+00	.10000000D+01	.00000000D+00
18	.52728476D+00	.10000000D+01	.00000000D+00
19	.00000000D+00	.20090572D+01	.16013124D+00
20	.87810670D+00	.10000000D+01	.00000000D+00
21	.39281531D+00	.10000000D+01	.00000000D+00
22	.68754290D+00	.10000000D+01	.00000000D+00

EVALUATIONS OF RESTRICTIONS AND THEIR GRADIENTS

```

( 0, 0) ( 0, 0) ( 32, 1) ( 32, 1) ( 32, 1)
( 32, 1) ( 32, 1) ( 32, 0) ( 42, 12) ( 46, 18)
( 46, 18) ( 32, 0) ( 32, 0) ( 32, 0) ( 46, 18)
( 46, 18) ( 32, 0) ( 32, 0) ( 46, 18) ( 32, 0)
( 32, 0) ( 32, 0)

```

```

LAST ESTIMATE OF CONDITION OF ACTIVE GRADIENTS .2310E+02
LAST ESTIMATE OF CONDITION OF APPROX. HESSIAN .3280E+05
ITERATIVE STEPS TOTAL 17
# OF RESTARTS 0
# OF FULL REGULAR UPDATES 15
# OF UPDATES 16
# OF FULL REGULARIZED SQP-STEPS 0

```

Date and time of the run are reported first. It follows the short protocol mentioned above. Then the setting of internal and external parameters is given together with the initial point. The parameters have the following meaning:

N primal dimension.  
 NH number of equality constraints.  
 NG number of inequality constraints, including simple bounds.  
 EPSX  $\varepsilon_x$ .  
 SIGSM smallest stepsize allowed during backtracking.  
 EPS  $\varepsilon_{\text{mach}}$ .  
 TOL smallest positive nonzero machine number.  
 DEL0  $\delta_0$ .  
 DELM  $\delta_{\text{min}}$ .  
 TAU0  $\tau_0$ .  
 TAU weight for "horizontal" part of direction, if infeasibility is large, i.e.  $\text{UPSI} > \text{TAU0}/2$ .  
 SD "small  $d$ ". If the working set is constant and  $\|d\| < \text{SD}$ , then the code enters PHASE=2 and uses the Maratos correction.  
 SW "small multiplier". Tolerance for negative multipliers and the error in the complementarity condition. ( $\varepsilon_\lambda$  in the section "termination criteria" below).  
 RHO The working set is considered as "singular" if  $\min\{|r_{ii}|\} / \max\{|r_{ii}|\} \leq \text{RHO}$ , where  $R$  denotes the triangular part of its QR-decomposition.  
 RHO1 A reinitialization of the quasi-Newton update is issued if  $\min\{l_{ii}\} / \max\{l_{ii}\} \leq \sqrt{\text{RHO1}}$ , where  $L$  is its Cholesky-factor.  
 SCFM The automatic scaling of the objective function is bounded from below by  $1/\text{SCFM}$  and from above by  $\text{SCFM}$ . The maximum weight of the slacks in the QP-problem is  $\text{SCFM}^2$ .  
 C1D scale factor for modification of function values by negative multipliers ( $\psi_A$  in the paper [6]).  
 EPDI assumed discretization stepsize for numerical gradients, if any.  
 NRE *nreset*, see above.  
 ANAL T=true for analytical gradients.

The lines of output which follow are selfexplanatory. The final scaling reported is the internal scaling of the objective function. Output of the norm values and multipliers is done relative to  $\text{SCF}=1$ , that is, to the users scaling. "LAGRANGIAN VIOLATION" is the Euclidean norm of  $\nabla_x L(\dots)$ . The user might compare this with "NORM OF GRAD(F)" =  $\nabla f(x)$  in order to assess the precision obtained. "FEASIBILITY VIOLATION" is  $\|h(x)\|_1 + \|g(x)^-\|_1$ . "DUAL FEASIBILITY VIOLATION" is the absolute value of the most negative multiplier for the inequality constraints. The output then reports the optimal primal and dual solutions. In the list of the constraints there come the equality constraints first, followed by the inequality constraints. Within each group the order is general linear constraints followed by simple bounds followed by the nonlinear constraints, if any. Under "NORM(GRAD) OR 1" appears  $\max\{1, \|\nabla c(x)\|\}$ . Then the evaluation counts for the restrictions are listed (in the same ordering as above).

Under "... CONDITION OF ACTIVE GRADIENTS" there appears  $\max\{|r_{ii}|\} / \min\{|r_{ii}|\}$  with  $R$  as the triangular part of the QR-decomposition of the working set (if  $\text{SI}=-1$ ) or the working set in the QP-solver otherwise.

Under "... CONDITION OF APPROX. HESSIAN" there appears  $(\max\{l_{ii}\} / \min\{l_{ii}\})^2$ , where  $L$  is the Cholesky factor of the last valid quasi-Newton update of the Hessian. This might be misleading, if a reinitialization ( $\text{UPD}=-1$ ) occurred less than  $n$  steps before.

The number of steps reported refers to the number of occurrences of  $\text{UPD}=-1$ ,  $\text{UPD}=1$ ,  $\text{UPD}=1,2,3$  and  $\text{SI}=1$ , see the description given above under "te1".

## 6 Concerning the choice of $epsx$

The parameter  $epsx$  describes the desired precision of the solution, see the termination criteria specified in the next section. For a well scaled and well conditioned problem it essentially specifies a desired relative precision in the solution. It should never be chosen less than the square root of the machine precision, since the control of progress in the method is based on the comparison of function values, usually taken from the constraining manifold, where the objective function varies like  $\mathcal{O}(\|x^k - x^*\|^2)$ . Even this requirement may be too strong. The default value for  $epsx$  is  $10^{-5}$ , approximately the third root of the machine precision. The user should be aware of the fact that *DONLP2* relaxes the precision requirement automatically, if it considers a problem as "singular". If the precision seems to be too poor in such a case, a decrease of  $epsx$  might help.

## 7 Termination criteria used

The following termination criteria are used for the present algorithm:

0.  $\psi(x) \leq \varepsilon_\psi$  and  
 $\|\nabla L\| \leq \varepsilon_x(1 + \|\nabla f(x)\|)$  and  
 $(\lambda_{\mathcal{A}_3})^T g_{\mathcal{A}_3}(x) \leq |\mathcal{A}_3| \varepsilon_\psi \varepsilon_\lambda$  .  
 Here  $\mathcal{A}_3$  is the index set of inequality constraints which are strictly satisfied.
1.  $\|d\| \leq \varepsilon_x(\|x\| + 1)$  and  
 $\psi(x) \leq \varepsilon_\psi$  and  
 $\|\nabla L\| \leq \varepsilon_x(1 + \|\nabla f\|)$ .
2.  $-D\Phi(x, d; u, v) \leq 100\varepsilon_{\text{mach}}(|\Phi(x; u, v)| + \varepsilon_{\text{mach}})$  and  
 $\psi(x) \leq \varepsilon_\psi(m + p)$  . Here  $D\Phi$  denotes the directional derivative of  $\Phi$  in the direction  $d$  at  $x$ .

Here  $\varepsilon_{\text{mach}}$  is the machine precision. The first criterion states the Kuhn-Tucker-conditions for NLP . The second one is quite similar and avoids usage of a too small correction which would give no significant further progress. The third one states that the directional derivative of  $\Phi$  is of the order of the round-off-level of evaluating  $\Phi$ . Using such a  $d$  might merely result in useless evaluations of  $f$ ,  $g$  and  $h$ . This termination criterion avoids useless computations for very illconditioned or even singular cases. The following default settings are used:

$$\begin{aligned} \varepsilon_x &= 10^{-5} , \\ \varepsilon_\lambda &= 10^{-10} , \\ \varepsilon_\psi &= 10^{-6} . \end{aligned}$$

$\varepsilon_{\text{mach}}$  is computed internally.  $\varepsilon_x$  may also be chosen by the user. Useful values are in the range

$$\varepsilon_{\text{mach}}^{1/2} \leq \varepsilon_x \leq 10^{-3}.$$

"Successful termination" means that one of these criteria has been satisfied. There are a lot of additional termination reasons for the code. These are summarized in the following list of terminating messages:

- 1 "cannot evaluate constraints",
- 2 "cannot evaluate objective",
- 3 "dual extended QP failure: singular working set",
- 4 "infeasible QP",
- 5 "no descent in QP",
- 6 "tiny QP step from infeasible point",

```

7      "nondescent direction from QP",
8      "reached maxit steps",
9      "no acceptable step size",
10     "tiny correction from QP at infeasible point",
11     "Success! KKT conditions satisfied",
12     "tiny step",
13     "nearly feasible with small directional deriv.",
14     "relaxed KKT conditions satisfied: singular point",
15     "slow primal progress: singular or ill-conditioned problem?",
16     "more than NRESET small primal corrections",
17     "small QP correction: nearly feasible and singular",
18     "max(N,10) small penalty function differences"

```

Messages 1 and 2 are selfexplanatory. Indeed users must provide for a mechanism to detect this behaviour in their function code. Messages 3 and 4 are theoretically impossible. Nevertheless, for a very badly scaled problem with linearly (almost or exactly) dependent gradients from the working set, it may occur. In this case the user should try to reformulate the problem. Messages 5,6 and 10 mean that locally at least the constraints in the working set contradict each to the other, (that is, the extended Mangasarian-Fromowitz condition is not satisfied there). In that case DONLP2 tries to drive the penalty and slack weights to infinity and stops near a local minimizer of the pure penalty term. It may be that the problem is infeasible indeed, but for a nonconvex problem this is not necessarily so. Message 7 is theoretically impossible again but may occur for the same reasons as 5 and 6. Message 8 is self explaining. If this occurs with large values of *maxit*, it is reasonable to assume that the problem is singular (i.e. the Kuhn-Tucker system has a singular Jacobian near the solution). Message 9 can have different reasons. Mostly it occurs due to imprecise gradients (and in many cases simply comes from a programming error in these). Since the AMPL version uses automatic differentiation, this cannot be the case. If however the precision requirement (by choosing  $\epsilon_{psx} = \epsilon_x$ ) is too stringent (e.g. for a illconditioned or singular problem), it may occur that no acceptable stepsize is found near the solution. In many cases the solution is found to a good precision nevertheless. Messages 11, 12 and 13 correspond to the termination criteria listed above. Normally also in the case of message 13 the solution is fine . Message number 14 occurs only for a working set with linearly (almost) dependent gradients. In that case  $\epsilon_x$  is replaced by  $100\epsilon_x$  in the termination criterion 0 given above. Message number 15 corresponds to the following termination criterion: The condition number of the diagonal of the Cholesky factor of  $B_k$  is greater than  $\sqrt{(10^3)}$  or the gradients are done numerically (that is, precision of  $d^k$  is necessarily impaired) and for  $n$  successive steps the Kuhn-Tucker criteria are almost satisfied in the sense

$$\begin{aligned}
\|\lambda^{(-)}\|_{\infty} &\leq \epsilon_{\text{mach}}^{2/3}, \\
|\lambda^T g| &\leq (m+p)\epsilon_{\text{mach}}^{2/3} \min\{\text{del}0/10, \max\{10^{-6}\text{del}0, \epsilon_{\text{mach}}^{2/3}\}\} \\
\|h(x)\|_1 + \|g(x)^{(-)}\|_1 &\leq (m+p)\epsilon_{\text{mach}}^{2/3} \\
\|\nabla_x L\| &\leq 100\epsilon_x(\|\nabla f\| + 1)
\end{aligned}$$

and in addition

$$|f_k - f_{k-1}| \leq \min\{\epsilon_x, \epsilon_{\text{mach}}^{1/2}\}(|f_k| + 1)$$

In this situation no much further progress can be assumed. Possibly the problem is very illconditioned or even singular. Message number 16 has been discussed in connection with choosing *nreset* already. Message number 17 means that the working set is not regular and

$$\begin{aligned}
\|h(x)\|_1 + \|g(x)^{(-)}\|_1 &\leq (m+p)\epsilon_{\text{mach}}^{2/3} \\
\|d\| &\leq 10^{-2}\epsilon_x(\min\{1, \|x\|\} + \epsilon_x)
\end{aligned}$$

such that no significant further progress can be obtained from this direction. Finally, message 18 says that for  $\max\{10, n\}$  successive steps the change in the penalty function was less than  $10^3\epsilon_{\text{mach}}|\Phi|$ . Again that means that not much further progress seems possible. These additional termination criteria avoid useless computations for illconditioned problems. The user should consider the possibility to improve the formulation of the problem in such cases.

## References

- [1] Burke, J. V.: An exact penalization viewpoint of constrained optimization. *SIAM J. Control and Optimization* 29(1991), 968–998.
- [2] Clarke, F. H.: Optimization and nonsmooth analysis. John Wiley, New York, 1987.
- [3] Goldfarb, D.; Idnani, A.: A numerically stable dual method for solving strictly convex quadratic programs, *Math. Prog.* 27, (1983), 1–33 .
- [4] Mayne, D.Q.; Pantoja, J.F.A.: Exact penalty function algorithm with simple updating of the penalty parameter *J.O.T.A.* 69, 441–467 (1991).
- [5] Spellucci, P.: *Numerische Verfahren der nichtlinearen Optimierung* Basel: Birkhäuser 1993.
- [6] Spellucci, P.: A SQP method for general nonlinear programs using only equality constrained sub-problems . To appear in *Math. Prog.* 1998
- [7] Spellucci, P.: A new technique for inconsistent QP problems in the SQP method . To appear in *Math. Meth. OR*, Vol. 47, (1998)