GPU Implementation of Algorithm NCL

Michael Saunders
MS&E and ICME, Stanford University

Alexis Montoison, Dominique Orban, François Pacaud, Sungho Shin

JuMP-dev 2025
Auckland, New Zealand, November 17–20, 2025

Abstract

GPU Implementation of Algorithm NCL

For constrained optimization, LANCELOT solves about 10 subproblems that minimize an augmented Lagrangian subject to bounds and are immune to LICQ and MPEC difficulties. Algorithm NCL solves equivalent subproblems that are suited to nonlinear interior methods. We focus on reducing the associated KKT systems to smaller systems that can be solved by Cholesky-type factorizations. Our NCL implementation is based on MadNLP.jl (a nonlinear optimization solver working on GPU) and CUDSS.jl (a Julia interface to the NVIDIA library cuDSS). We present numerical results on large SCOPF problems (which are MPECS).

Constrained Optimization

$$\boxed{\min \phi(x) \quad \text{st} \quad c(x) = 0, \qquad\qquad c \in \mathbb{R}^m, \;\; m < n}$$

Penalty function

$$P(x, \rho_k) = \phi(x) + \tfrac{1}{2}\rho_k c(x)^T c(x)$$

Penalty parameter $\rho_k \to \infty$

Constrained Optimization

$$\boxed{\min \phi(x) \quad \text{st} \quad c(x) = 0, \qquad\qquad c \in \mathbb{R}^m, \;\; m < n}$$

Penalty function

$$P(x, \rho_k) = \phi(x) + \tfrac{1}{2}\rho_k c(x)^T c(x)$$

Penalty parameter $\rho_k \to \infty$

Augmented Lagrangian

$$L(x, y_k, \rho_k) = \phi(x) - y_k^T c(x) + \tfrac{1}{2}\rho_k c(x)^T c(x)$$

If Lagrange multiplier estimate $y_k \to y^*$, $\quad \rho_k$ can remain finite

LANCELOT's BCL algorithm for general NLP

Conn, Gould & Toint (1992)

LANCELOT

$$\boxed{\min \phi(x) \quad \text{st} \quad c(x) = 0, \quad \ell \le x \le u}$$

BCL subproblems (Bound-Constrained augmented Lagrangian):

$$\boxed{\begin{array}{lll} \mathrm{BC}_k & \underset{x}{\text{minimize}} & \phi(x) - y_k^T c(x) + \frac{1}{2}\rho_k c(x)^T c(x) \\ & \text{subject to} & \ell \le x \le u \end{array}}$$

LANCELOT

$$\boxed{\min \phi(x) \quad \text{st} \quad c(x) = 0, \quad \ell \le x \le u}$$

BCL subproblems (Bound-Constrained augmented Lagrangian):

$$\boxed{\mathrm{BC}_k \qquad \begin{array}{ll} \underset{x}{\text{minimize}} & \phi(x) - y_k^T c(x) + \frac{1}{2}\rho_k c(x)^T c(x) \\ \text{subject to} & \ell \le x \le u \end{array}}$$

Loop:

solve BC_k to get x_k^*	decreasing opttol ω_k
if $\|c(x_k^*)\| \le \eta_k$, $y_{k+1} \leftarrow y_k - \rho_k c(x_k^*)$	decreasing featol η_k
else $\rho_{k+1} \leftarrow 10\rho_k$	

LANCELOT

$$\boxed{\min \phi(x) \quad \text{st} \quad c(x) = 0, \quad \ell \le x \le u}$$

BCL subproblems (Bound-Constrained augmented Lagrangian):

BC_k

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \phi(x) - y_k^T c(x) + \tfrac{1}{2}\rho_k c(x)^T c(x) \\ \text{subject to} & \ell \le x \le u \end{array}$$

Loop:

solve BC_k to get x_k^*	decreasing opttol ω_k
if $\|c(x_k^*)\| \le \eta_k$, $y_{k+1} \leftarrow y_k - \rho_k c(x_k^*)$	decreasing featol η_k
else $\rho_{k+1} \leftarrow 10\rho_k$	

Only about 10(!) subproblems BC_k No LICQ worries

The optimal taxation policy problem that started it all

Kenneth Judd, economist

Hoover Institution, Stanford

Optimal Tax Policy

Kenneth Judd and Che-Lin Su 2011


Ken's taxation problem

NLP

$$\begin{aligned} &\underset{x\in\mathbb{R}^n}{\text{minimize}} && \phi(x) \\ &\text{subject to} && c(x) \geq 0, \quad \ell \leq x \leq u \end{aligned}$$

Many inequalities $c(x) \geq 0$ might not satisfy LICQ at x^*

Example: $m = 571,000, \quad n = 1500$

10,000 constraints essentially active: $c_i(x^*) \leq 10^{-6}$

BCL

LCL

NCL

Sequence of subproblems minimizing
X-constrained (augmented) Lagrangian

BCL	LANCELOT	Conn, Gould & Toint (1992)
LCL	linearized constraints	Robinson (1972)
	MINOS	Murtagh and S (1982)
sLCL	KNOSSOS	Friedlander (2002)
SQP	SNOPT	Gill, Murray, S (2005)
NCL	New form of **BCL**	Ma, Judd, Orban, S (2017)
		using IPOPT, KNITRO or MADNLP

Algorithm NCL
for general NLP

NCL subproblems

NLP

$$\begin{array}{ll}\underset{x}{\text{minimize}} & \phi(x) \\ \text{subject to} & c(x) = 0, \quad \ell \le x \le u\end{array}$$

LANCELOT-type subproblems:

BC_k

$$\begin{array}{ll}\underset{x}{\text{minimize}} & L(x, y_k, \rho_k) = \phi(x) - y_k^T c(x) + \frac{1}{2}\rho_k c(x)^T c(x) \\ \text{subject to} & \ell \le x \le u\end{array}$$

NCL subproblems

NLP
$$\underset{x}{\text{minimize}} \quad \phi(x)$$
$$\text{subject to} \quad c(x) = 0, \quad \ell \le x \le u$$

LANCELOT-type subproblems:

BC_k
$$\underset{x}{\text{minimize}} \quad L(x, y_k, \rho_k) = \phi(x) - y_k^T c(x) + \tfrac{1}{2}\rho_k c(x)^T c(x)$$
$$\text{subject to} \quad \ell \le x \le u$$

Introduce $r = -c(x)$: $(r \to 0)$

NC_k
$$\underset{x,\,r}{\text{minimize}} \quad \phi(x) + y_k^T r + \tfrac{1}{2}\rho_k r^T r$$
$$\text{subject to} \quad c(x) + r = 0, \quad \ell \le x \le u$$

Free vars r make the constraints independent and feasible

Interior solvers happy!

NCL subproblems

NLP

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \phi(x) \\ \text{subject to} & c(x) = 0, \quad \ell \le x \le u \end{array}$$

NC_k

$$\begin{array}{ll} \underset{x,\, r}{\text{minimize}} & \phi(x) + y_k^T r + \tfrac{1}{2}\rho_k r^T r \\ \text{subject to} & c(x) + r = 0, \quad \ell \le x \le u \end{array}$$

Free vars r make the nonlinear constraints independent and feasible Interior solvers happy!

NCL subproblems for Ken's problem

NLP

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \phi(x) \\ \text{subject to} \quad & c(x) \geq 0, \quad \ell \leq x \leq u \end{aligned}$$

NC$_k$

$$\begin{aligned} \underset{x,\,r}{\text{minimize}} \quad & \phi(x) + y_k^T r + \tfrac{1}{2}\rho_k r^T r \\ \text{subject to} \quad & c(x) + r \geq 0, \quad \ell \leq x \leq u \end{aligned}$$

Free vars r make the nonlinear constraints independent and feasible   Interior solvers happy!

Numerical results

Warm-start options for Nonlinear Interior Methods

IPOPT	warm_start_init_point=yes	
	mu_init=1e-4	(1e-5, ..., 1e-8)

Warm-start options for Nonlinear Interior Methods

```
IPOPT      warm_start_init_point=yes
           mu_init=1e-4                    (1e-5, ..., 1e-8)
```

mu_init is the initial value of μ (the barrier parameter)

$$\mu \to 0$$

NCL/IPOPT on problem TAX

$na, nb, nc, nd, ne = 5, 3, 3, 2, 2 \quad m = 32220 \quad n = 360$

k	ρ_k	η_k	$\|r_k^*\|_\infty$	$\phi(x_k^*)$	mu_init	Itns	Time
1	10^2	10^{-2}	7.0e-03	-4.2038075e+02	10^{-1}	95	41.1
2	10^2	10^{-3}	4.1e-03	-4.2002898e+02	10^{-4}	17	7.2
3	10^3	10^{-3}	1.3e-03	-4.1986069e+02	10^{-4}	20	8.1
4	10^4	10^{-3}	4.4e-04	-4.1972958e+02	10^{-4}	48	25.0
5	10^4	10^{-4}	2.2e-04	-4.1968646e+02	10^{-4}	43	20.5
6	10^5	10^{-4}	9.8e-05	-4.1967560e+02	10^{-4}	64	32.9
7	10^5	10^{-5}	6.6e-05	-4.1967177e+02	10^{-4}	57	26.8
8	10^6	10^{-5}	4.2e-06	-4.1967150e+02	10^{-4}	87	46.2
9	10^6	10^{-6}	9.4e-07	-4.1967138e+02	10^{-4}	96	53.6

527 iterations, 5 mins CPU

NCL/IPOPT on problem TAX

$na, nb, nc, nd, ne = 5, 3, 3, 2, 2 \quad m = 32220 \quad n = 360$

k	ρ_k	η_k	$\|r_k^*\|_\infty$	$\phi(x_k^*)$	mu_init	Itns	Time
1	10^2	10^{-2}	7.0e-03	-4.2038075e+02	10^{-1}	95	40.8
2	10^2	10^{-3}	4.1e-03	-4.2002898e+02	10^{-4}	17	7.0
3	10^3	10^{-3}	1.3e-03	-4.1986069e+02	10^{-4}	20	8.5
4	10^4	10^{-3}	4.4e-04	-4.1972958e+02	10^{-5}	57	32.6
5	10^4	10^{-4}	2.2e-04	-4.1968646e+02	10^{-5}	29	14.6
6	10^5	10^{-4}	9.8e-05	-4.1967560e+02	10^{-6}	36	18.7
7	10^5	10^{-5}	3.9e-05	-4.1967205e+02	10^{-6}	35	19.7
8	10^6	10^{-5}	4.2e-06	-4.1967150e+02	10^{-7}	18	7.7
9	10^6	10^{-6}	9.4e-07	-4.1967138e+02	10^{-7}	15	6.8

322 iterations, 3 mins CPU

NCL/IPOPT bigger example

$na, nb, nc, nd, ne = 21, 3, 3, 2, 2 \quad m = 570780 \quad n = 1512$

k	ρ_k	η_k	$\lVert r_k^* \rVert_\infty$	$\phi(x_k^*)$	mu_init	Itns	Time
1	10^2	10^{-2}	5.1e-03	-1.7656816e+03	10^{-1}	825	7763
2	10^2	10^{-3}	2.4e-03	-1.7648480e+03	10^{-4}	66	473
3	10^3	10^{-3}	1.3e-03	-1.7644006e+03	10^{-4}	106	771
4	10^4	10^{-3}	3.8e-04	-1.7639491e+03	10^{-5}	132	1347
5	10^4	10^{-4}	3.2e-04	-1.7637742e+03	10^{-5}	229	2451
6	10^5	10^{-4}	8.6e-05	-1.7636804e+03	10^{-6}	104	1097
7	10^5	10^{-5}	4.9e-05	-1.7636469e+03	10^{-6}	143	1633
8	10^6	10^{-5}	1.5e-05	-1.7636252e+03	10^{-7}	71	786
9	10^7	10^{-5}	2.8e-06	-1.7636196e+03	10^{-7}	67	726
10	10^7	10^{-6}	5.1e-07	-1.7636187e+03	10^{-8}	18	171

1761 iterations, 5 hours CPU

NCL/IPOPT bigger example

$na, nb, nc, nd, ne = 21, 3, 3, 2, 2 \quad m = 570780 \quad n = 1512$

Constraints within tol of being active: $c_i(x) \leq tol$

tol	*count*	*count*/*n*	
10^{-10}	3888	2.6	
10^{-9}	3941	2.6	
10^{-8}	4430	2.9	
10^{-7}	7158	4.7	
$\rightarrow 10^{-6}$	10074	6.6 $\leftarrow$	$\approx 6.6n$ active constraints
10^{-5}	11451	7.6	
10^{-4}	13109	8.7	
10^{-3}	23099	15.3	
10^{-2}	66361	43.9	
10^{-1}	202664	134.0	

Warm-start options for Nonlinear Interior Methods

IPOPT	warm_start_init_point=yes	
	mu_init=1e-4	(1e-5, ..., 1e-8)
KNITRO	algorithm=1	Thanks, Richard Waltz!
	bar_directinterval=0	
	bar_initpt=2	
	bar_murule=1	
	bar_initmu=1e-4	(1e-5, ..., 1e-8)
	bar_slackboundpush=1e-4	(1e-5, ..., 1e-8)

Linear Algebra

Interior Methods (IPMs)

For general problems $\min_x \; \phi(x) \;\; \text{st} \;\; c(x) = 0, \quad x \ge 0,$
search directions $(\Delta x, \Delta y)$ come from

$$\begin{pmatrix} -(H + X^{-1}Z) & J^T \\ J & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_2 \\ r_1 \end{pmatrix}$$

Interior Methods (IPMs)

For general problems $\min_x \ \phi(x) \ \text{ st } \ c(x) = 0, \quad x \ge 0,$
search directions $(\Delta x, \Delta y)$ come from

$$\begin{pmatrix} -(H + X^{-1}Z) & J^T \\ J & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_2 \\ r_1 \end{pmatrix}$$

For NCL subproblems $\min_{x,\,r} \ \phi(x) + y_k^T r + \frac{1}{2}\rho_k r^T r \ \text{ st } \ c(x) + r = 0, \quad x \ge 0$
the linear system is

$$\begin{pmatrix} -(H + X^{-1}Z) & & J^T \\ & -\rho_k I & I \\ J & I & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta r \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_2 \\ r_3 \\ r_1 \end{pmatrix}$$

Linear Algebra

$$\begin{pmatrix} -(H + X^{-1}Z) & & J^T \\ & -\rho_k I & I \\ J & I & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta r \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_2 \\ r_3 \\ r_1 \end{pmatrix} \tag{K3}$$

Linear Algebra

$$\begin{pmatrix} -(H + X^{-1}Z) & & J^T \\ & -\rho_k I & I \\ J & I & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta r \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_2 \\ r_3 \\ r_1 \end{pmatrix} \tag{K3}$$

$$\begin{pmatrix} -(H + X^{-1}Z) & J^T \\ J & \frac{1}{\rho_k} I \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_2 \\ r_1 + \frac{r_3}{\rho_k} \end{pmatrix}, \qquad \Delta r = \frac{1}{\rho_k}(\Delta y - r_3). \tag{K2}$$

Linear Algebra

$$\begin{pmatrix} -(H + X^{-1}Z) & & J^T \\ & -\rho_k I & I \\ J & I & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta r \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_2 \\ r_3 \\ r_1 \end{pmatrix} \tag{K3}$$

$$\begin{pmatrix} -(H + X^{-1}Z) & J^T \\ J & \frac{1}{\rho_k} I \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_2 \\ r_1 + \frac{r_3}{\rho_k} \end{pmatrix}, \qquad \Delta r = \frac{1}{\rho_k}(\Delta y - r_3). \tag{K2}$$

$$(H + X^{-1}Z + \rho_k J^T J)\Delta x = -r_2 + J^T(r_3 + \rho_k r_1), \qquad \Delta y = r_3 + \rho_k(r_1 - J\Delta x), \tag{K1}$$

Linear Algebra

$$\begin{pmatrix} -(H + X^{-1}Z) & & J^T \\ & -\rho_k I & I \\ J & I & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta r \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_2 \\ r_3 \\ r_1 \end{pmatrix} \tag{K3}$$

$$\begin{pmatrix} -(H + X^{-1}Z) & J^T \\ J & \frac{1}{\rho_k} I \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_2 \\ r_1 + \frac{r_3}{\rho_k} \end{pmatrix}, \qquad \Delta r = \frac{1}{\rho_k}(\Delta y - r_3). \tag{K2}$$

$$(H + X^{-1}Z + \rho_k J^T J)\Delta x = -r_2 + J^T(r_3 + \rho_k r_1), \qquad \Delta y = r_3 + \rho_k(r_1 - J\Delta x), \tag{K1}$$

- IPOPT, KNITRO must solve (K3). MadNCL/MadNLP can solve (K2) or (K1)

Linear Algebra

$$\begin{pmatrix} -(H+X^{-1}Z) & & J^T \\ & -\rho_k I & I \\ J & I & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta r \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_2 \\ r_3 \\ r_1 \end{pmatrix} \tag{K3}$$

$$\begin{pmatrix} -(H+X^{-1}Z) & J^T \\ J & \frac{1}{\rho_k} I \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_2 \\ r_1 + \frac{r_3}{\rho_k} \end{pmatrix}, \qquad \Delta r = \frac{1}{\rho_k}(\Delta y - r_3). \tag{K2}$$

$$(H + X^{-1}Z + \rho_k J^T J)\Delta x = -r_2 + J^T(r_3 + \rho_k r_1), \qquad \Delta y = r_3 + \rho_k(r_1 - J\Delta x), \tag{K1}$$

- IPOPT, KNITRO must solve (K3). MadNCL/MadNLP can solve (K2) or (K1)
- (K2) usually needs MA27 Inertia-correction eventually makes (K2) SQD, (K1) SPD

Linear Algebra

$$\begin{pmatrix} -(H + X^{-1}Z) & & J^T \\ & -\rho_k I & I \\ J & I & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta r \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_2 \\ r_3 \\ r_1 \end{pmatrix} \tag{K3}$$

$$\begin{pmatrix} -(H + X^{-1}Z) & J^T \\ J & \frac{1}{\rho_k} I \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} r_2 \\ r_1 + \frac{r_3}{\rho_k} \end{pmatrix}, \qquad \Delta r = \frac{1}{\rho_k}(\Delta y - r_3). \tag{K2}$$

$$(H + X^{-1}Z + \rho_k J^T J)\Delta x = -r_2 + J^T(r_3 + \rho_k r_1), \qquad \Delta y = r_3 + \rho_k(r_1 - J\Delta x), \tag{K1}$$

- IPOPT, KNITRO must solve (K3). MadNCL/MadNLP can solve (K2) or (K1)
- (K2) usually needs MA27 Inertia-correction eventually makes (K2) SQD, (K1) SPD
- (K2) and (K1) allow sparse $\mathrm{LDL^T}$, GPUs

SCOPF

Security Constrained Optimal Power Flow

(Evidently an MPEC)

MadNCL on SCOPF problems

Results from François during INFORMS Oct 2024:

- MadNCL convergence tolerance `1e-6`
- Return status `st` = 1 if locally optimal, -3 if step is becoming too small
- CUDSS uses GPUs

		MadNCL+K2+MA27					MadNCL+K2+CUDSS					MadNCL+K1+CUDSS				
	K	st	obj	it	linsolve	total	st	obj	it	linsolve	tota l	st	obj	it	linsolve	total
118ieee	64	1	13.4	152	20.7	26.0	1	13.4	108	3.1	4.8	1	13.4	108	2.0	3.3
1354pegase	8	1	7.4	144	36.6	43.0	1	7.4	143	6.5	8.8	1	7.4	144	5.0	6.8
1354pegase	16	1	7.4	282	235.3	259.7	-3	7.4	295	30.0	35.0	1	7.4	231	17.9	21.1
ACTIVSg2000	8	1	122.9	296	543.2	564.1	1	122.9	314	29.1	33.9	-3	122.9	429	31.7	37.0
2869pegase	8	-3	13.4	331	305.0	340.0	1	13.4	211	21.5	26.7	-3	13.4	244	19.1	23.2

MadNCL on SCOPF problems

Results from François during INFORMS Oct 2024:

- MadNCL convergence tolerance 1e−6
- Return status st = 1 if locally optimal, -3 if step is becoming too small
- CUDSS uses GPUs

		MadNCL+K2+MA27					MadNCL+K2+CUDSS					MadNCL+K1+CUDSS				
	K	st	obj	it	linsolve	total	st	obj	it	linsolve	tota l	st	obj	it	linsolve	total
118ieee	64	1	13.4	152	20.7	26.0	1	13.4	108	3.1	4.8	1	13.4	108	2.0	3.3
1354pegase	8	1	7.4	144	36.6	43.0	1	7.4	143	6.5	8.8	1	7.4	144	5.0	6.8
1354pegase	16	1	7.4	282	235.3	259.7	-3	7.4	295	30.0	35.0	1	7.4	231	17.9	21.1
ACTIVSg2000	8	1	122.9	296	543.2	564.1	1	122.9	314	29.1	33.9	-3	122.9	429	31.7	37.0
2869pegase	8	-3	13.4	331	305.0	340.0	1	13.4	211	21.5	26.7	-3	13.4	244	19.1	23.2

Conferences are worthwhile!

Comparing MadNLP and MadNCL

Comparing the performance of MadNLP and MadNCL with GPU on large-scale OPF instances from PGLIB and large-scale COPS instances, `tol=1e-8`.

	MadNLP-K2-Ma27				MadNLP-HyKKT-cuDSS				MadNLP-LiftedKKT-cuDSS				MadNCL-K2r-cuDSS				MadNCL-K1s-cuDSS			
case	flag	it	lin	total	flag	it	lin	total	flag	it	lin	total	flag	it	lin	total	flag	it	lin	total
8387_pegase	1	77	7.26	**11.33**	1	78	98.02	**99.15**	2	99	7.68	**8.77**	5	482	62.14	**64.48**	5	455	60.74	**64.21**
9241_pegase	1	75	8.60	**11.62**	1	72	9.92	**11.02**	6	1000	104.64	**118.74**	5	215	35.86	**36.89**	5	227	37.78	**39.30**
9591_goc	1	68	15.88	**18.68**	2	183	21.07	**23.06**	1	72	8.88	**9.94**	1	90	3.56	**3.99**	1	95	5.42	**5.98**
10000_goc	1	85	8.45	**11.30**	1	86	8.93	**9.92**	1	66	8.58	**9.36**	1	96	4.75	**5.20**	1	89	4.60	**5.12**
10192_epigrids	1	56	11.22	**14.45**	2	69	9.90	**11.05**	1	61	8.31	**9.36**	1	86	3.17	**3.59**	1	89	5.35	**5.89**
10480_goc	1	72	17.34	**21.11**	2	204	26.02	**28.12**	1	69	5.91	**6.98**	1	132	6.95	**7.56**	1	134	8.43	**9.21**
13659_pegase	1	65	10.32	**14.24**	1	64	9.16	**10.39**	1	71	6.45	**7.64**	1	146	7.00	**7.67**	1	146	8.09	**9.07**
19402_goc	1	72	50.48	**57.19**	1	222	34.96	**38.90**	1	72	6.65	**8.36**	1	126	6.16	**6.80**	5	1000	75.08	**84.78**
20758_epigrids	1	53	22.75	**28.18**	1	55	8.86	**10.52**	1	242	33.11	**35.76**	1	174	24.58	**25.79**	5	1000	212.58	**221.85**
24464_goc	1	64	31.38	**38.46**	6	1000	141.06	**148.86**	1	65	5.87	**7.77**	1	214	9.83	**10.88**	6	1000	213.77	**222.71**
30000_goc	1	229	137.64	**158.13**	1	232	29.95	**32.68**	1	155	14.47	**16.50**	1	503	81.88	**84.70**	6	1000	190.10	**198.14**
78484_epigrids	1	104	291.93	**328.40**	-1	689	111.68	**160.59**	1	105	19.74	**34.39**	1	354	52.47	**66.83**	5	1000	195.23	**255.93**
bearing	1	18	17.12	**32.83**	1	18	7.59	**20.43**	1	15	0.68	**6.16**	1	14	0.66	**5.72**	1	14	2.31	**28.41**
catmix	1	20	1.47	**7.68**	1	17	0.69	**4.43**	1	51	7.90	**13.28**	1	71	7.04	**9.10**	1	69	6.70	**11.09**
channel	1	6	0.75	**7.55**	1	6	0.58	**4.60**	1	7	0.52	**4.22**	1	19	1.43	**3.25**	1	24	2.23	**4.55**
elec	1	209	98.41	**121.59**	1	112	4.73	**13.02**	1	145	6.29	**15.53**	1	169	7.20	**13.86**	1	268	15.40	**41.12**
gasoil	1	20	1.35	**18.23**	1	20	2.77	**7.69**	1	87	24.50	**34.21**	1	24	1.65	**4.43**	1	26	3.22	**6.97**
marine	1	14	2.09	**9.38**	1	19	1.71	**6.57**	2	37	17.02	**23.23**	1	31	1.81	**4.37**	1	31	5.47	**8.11**
pinene	1	12	1.76	**12.43**	1	11	2.07	**3.69**	1	20	2.94	**4.94**	1	54	3.37	**4.95**	1	64	10.71	**10.99**
polygon	1	32	0.03	**47.18**	1	31	0.75	**12.31**	1	225	7.31	**19.45**	1	41	1.22	**9.37**	1	41	1.07	**37.14**
robot	1	31	1.60	**10.88**	1	87	184.89	**193.69**	1	26	0.57	**5.53**	1	25	0.59	**3.72**	1	27	3.09	**9.56**
steering	1	17	0.23	**7.80**	1	16	0.36	**4.52**	1	14	0.44	**4.24**	1	32	0.91	**4.00**	1	17	0.42	**4.57**
torsion	1	14	0.96	**1.20**	1	14	1.07	**1.35**	1	15	0.51	**0.78**	1	10	0.34	**0.40**	1	10	0.33	**0.39**

Summary of Algorithm NCL

NLP

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \phi(x) \\ \text{subject to} & c(x) = 0, \quad \ell \le x \le u \end{array}$$

LANCELOT subproblems:

BC_k

$$\begin{array}{ll} \underset{x}{\text{minimize}} & L(x, y_k, \rho_k) = \phi(x) - y_k^T c(x) + \frac{1}{2}\rho_k c(x)^T c(x) \\ \text{subject to} & \ell \le x \le u \end{array}$$

Summary of Algorithm NCL

NLP

$$\begin{array}{ll} \underset{x}{\text{minimize}} & \phi(x) \\ \text{subject to} & c(x) = 0, \quad \ell \le x \le u \end{array}$$

LANCELOT subproblems:

BC_k

$$\begin{array}{ll} \underset{x}{\text{minimize}} & L(x, y_k, \rho_k) = \phi(x) - y_k^T c(x) + \tfrac{1}{2}\rho_k c(x)^T c(x) \\ \text{subject to} & \ell \le x \le u \end{array}$$

NCL subproblems:

NC_k

$$\begin{array}{ll} \underset{x,\, r}{\text{minimize}} & \phi(x) + y_k^T r + \tfrac{1}{2}\rho_k r^T r \\ \text{subject to} & c(x) + r = 0, \quad \ell \le x \le u \end{array}$$

Free variabless $r \to 0$ make the constraints independent, feasible. (IPM solvers happy)

References

- D. Ma, K. L. Judd, D. Orban and M. A. Saunders, Stabilized optimization via an NCL algorithm, pp 173–191 in M. Al-Baali et al. (eds.), Numerical Analysis and Optimization, NAO-IV, Muscat, Oman, January 2017, Springer Proceedings in Mathematics & Statistics, Volume 235, 2018.
- https://github.com/MadNLP/MadNCL.jl

JuMP Support in `MadNCL.jl` (Thanks Alexis)

- Unlike its siblings `MadIPM.jl` and `MadNLP.jl`, `MadNCL.jl` does not yet provide a dedicated MOI extension. JuMP models must currently be wrapped as an `NLPModel` via `ExaModels.jl` (GPU-friendly) or `NLPModelsJuMP.jl` (CPU only).

JuMP Support in MadNCL.jl (Thanks Alexis)

- Unlike its siblings `MadIPM.jl` and `MadNLP.jl`, `MadNCL.jl` does not yet provide a dedicated MOI extension. JuMP models must currently be wrapped as an `NLPModel` via `ExaModels.jl` (GPU-friendly) or `NLPModelsJuMP.jl` (CPU only).

- Duplicating the MOI extension of `MadNLP.jl` would create significant maintenance overhead—an issue already seen in `Ipopt.jl`, `UnoSolver.jl`, and other nonlinear solvers.

JuMP Support in `MadNCL.jl` (Thanks Alexis)

- Unlike its siblings `MadIPM.jl` and `MadNLP.jl`, `MadNCL.jl` does not yet provide a dedicated MOI extension. JuMP models must currently be wrapped as an `NLPModel` via `ExaModels.jl` (GPU-friendly) or `NLPModelsJuMP.jl` (CPU only).

- Duplicating the MOI extension of `MadNLP.jl` would create significant maintenance overhead—an issue already seen in `Ipopt.jl`, `UnoSolver.jl`, and other nonlinear solvers.

- The ongoing modernization of the MOI interface in `MadNLP.jl` opens a promising direction: expose `MadNCL` as an *algorithmic option*, enabling full reuse of a unified MOI layer.

JuMP Support in `MadNCL.jl` (Thanks Alexis)

- Unlike its siblings `MadIPM.jl` and `MadNLP.jl`, `MadNCL.jl` does not yet provide a dedicated MOI extension. JuMP models must currently be wrapped as an `NLPModel` via `ExaModels.jl` (GPU-friendly) or `NLPModelsJuMP.jl` (CPU only).
- Duplicating the MOI extension of `MadNLP.jl` would create significant maintenance overhead—an issue already seen in `Ipopt.jl`, `UnoSolver.jl`, and other nonlinear solvers.
- The ongoing modernization of the MOI interface in `MadNLP.jl` opens a promising direction: expose `MadNCL` as an *algorithmic option*, enabling full reuse of a unified MOI layer.
- This would also make it easier to support the new `VectorNonlinearOracle` with minimal additional code.

JuMP Support in `MadNCL.jl`

- Open question: can we further abstract and factor out common logic across MOI extensions for nonlinear solvers? Several packages still depend on `QPBlockData`, which currently lives only inside solver-specific extensions.

JuMP Support in `MadNCL.jl`

- Open question: can we further abstract and factor out common logic across MOI extensions for nonlinear solvers? Several packages still depend on `QPBlockData`, which currently lives only inside solver-specific extensions.
- With Ipopt.jl as baseline, 84% of it is in the MOI extension of unosolver.jl, 62% in the one of MadNLP.jl and 26% in the one of KNITRO.jl.

Special thanks

- Taxation problem: Ken Judd, Che-lin Su
- LANCELOT: Andy Conn, Nick Gould, Philippe Toint
- NCL coauthors: Ding Ma, Dominique Orban

Special thanks

- Taxation problem: Ken Judd, Che-lin Su
- LANCELOT: Andy Conn, Nick Gould, Philippe Toint
- NCL coauthors: Ding Ma, Dominique Orban
- AMPL: Bob Fourer, Dave Gay
- IPOPT: Larry Biegler, Carl Laird, Andreas Wächter
- KNITRO: Richard Waltz, Jorge Nocedal, Todd Plantenga, Richard Byrd

Special thanks

- Taxation problem: Ken Judd, Che-lin Su
- LANCELOT: Andy Conn, Nick Gould, Philippe Toint
- NCL coauthors: Ding Ma, Dominique Orban
- AMPL: Bob Fourer, Dave Gay
- IPOPT: Larry Biegler, Carl Laird, Andreas Wächter
- KNITRO: Richard Waltz, Jorge Nocedal, Todd Plantenga, Richard Byrd
- MadNLP/MadNCL: Alexis Montoison, François Pacaud, Sungho Shin, Dominique Orban

Special thanks

- Taxation problem: Ken Judd, Che-lin Su
- LANCELOT: Andy Conn, Nick Gould, Philippe Toint
- NCL coauthors: Ding Ma, Dominique Orban
- AMPL: Bob Fourer, Dave Gay
- IPOPT: Larry Biegler, Carl Laird, Andreas Wächter
- KNITRO: Richard Waltz, Jorge Nocedal, Todd Plantenga, Richard Byrd
- MadNLP/MadNCL: Alexis Montoison, François Pacaud, Sungho Shin, Dominique Orban
- Oscar Dowson and team

Special thanks

- Taxation problem: Ken Judd, Che-lin Su
- LANCELOT: Andy Conn, Nick Gould, Philippe Toint
- NCL coauthors: Ding Ma, Dominique Orban
- AMPL: Bob Fourer, Dave Gay
- IPOPT: Larry Biegler, Carl Laird, Andreas Wächter
- KNITRO: Richard Waltz, Jorge Nocedal, Todd Plantenga, Richard Byrd
- MadNLP/MadNCL: Alexis Montoison, François Pacaud, Sungho Shin, Dominique Orban
- Oscar Dowson and team

Yesterday on hotel TV

- The speedway

Special thanks

- Taxation problem: Ken Judd, Che-lin Su
- LANCELOT: Andy Conn, Nick Gould, Philippe Toint
- NCL coauthors: Ding Ma, Dominique Orban
- AMPL: Bob Fourer, Dave Gay
- IPOPT: Larry Biegler, Carl Laird, Andreas Wächter
- KNITRO: Richard Waltz, Jorge Nocedal, Todd Plantenga, Richard Byrd
- MadNLP/MadNCL: Alexis Montoison, François Pacaud, Sungho Shin, Dominique Orban
- Oscar Dowson and team

Yesterday on hotel TV

- The speedway
- International rugby: NZ vs England

Special thanks

- Taxation problem: Ken Judd, Che-lin Su
- LANCELOT: Andy Conn, Nick Gould, Philippe Toint
- NCL coauthors: Ding Ma, Dominique Orban
- AMPL: Bob Fourer, Dave Gay
- IPOPT: Larry Biegler, Carl Laird, Andreas Wächter
- KNITRO: Richard Waltz, Jorge Nocedal, Todd Plantenga, Richard Byrd
- MadNLP/MadNCL: Alexis Montoison, François Pacaud, Sungho Shin, Dominique Orban
- Oscar Dowson and team

Yesterday on hotel TV

- The speedway
- International rugby: NZ vs England
- International cricket: NZ vs West Indies