Experimental results with Algorithm NCL for constrained optimization

Michael Saunders
MS&E and ICME, Stanford University

Ding Ma, Ken Judd, and Dominique Orban
Pierre-Élie Personnaz

ICCOPT 2019
TUB, Berlin, August 5–8, 2019

Abstract

We reimplement the LANCELOT augmented Lagrangian method as a short sequence of nonlinearly constrained subproblems that can be solved efficiently by IPOPT and KNITRO, with warm starts on each subproblem. NCL succeeds on degenerate tax policy models that can't be solved directly. These models (and algorithm NCL) are coded in AMPL. A Julia implementation of NCL gives results
for standard test problems.

Partially supported by the
National Institute of General Medical Sciences
of the National Institutes of Health (NIH)
Award U01GM102098

Constrained Optimization

NCO

$$\begin{array}{ll} \underset{x \in \mathbb{R}^n}{\text{minimize}} & \phi(x) \\ \text{subject to} & c(x) = 0 \qquad (c \in \mathbb{R}^m,\ m < n) \end{array}$$

Penalty function

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

Penalty parameter $\rho_k \to \infty$

Augmented Lagrangian

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

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

LANCELOT's BCL algorithm for general NLP

Conn, Gould & Toint (1992)

LANCELOT

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

BCL subproblems (Bound-Constrained augmented Lagrangian):

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

LANCELOT

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

BCL subproblems (Bound-Constrained augmented Lagrangian):

$$\text{BC}_k \qquad \underset{x}{\text{minimize}} \quad \phi(x) - y_k^T c(x) + \tfrac{1}{2}\rho_k \|c(x)\|^2$$
$$\text{subject to} \quad \ell \le x \le u$$

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 $\rho_{k+1} \leftarrow 10\rho_k$	

Our optimization problem

Our NLP problem

NLP

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \phi(x)$$
$$\text{subject to} \quad c(x) \ge 0, \quad \ell \le x \le u$$

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

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

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

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)
NCL	New form of **BCL**	
	AMPL or Julia loop + IPOPT or KNITRO	

Algorithm NCL
for general NLP

Temporary variables

In Fortran, C, Matlab, . . . , we replace gory expressions by temporaries.

```
f = a(1)*b(2)/c(3) + sqrt(a(1)*b(2)/c(3)) + log(a(1)*b(2)/c(3))
```

becomes

```
t = a(1)*b(2)/c(3)
f = t + sqrt(t) + log(t)
```

NCL subproblems

NLP	$\displaystyle \min_{x} \ \phi(x)$ subject to $c(x) = 0, \quad \ell \le x \le u$

LANCELOT-type subproblems:

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

NCL subproblems

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

LANCELOT-type subproblems:

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

Introduce $r = -c(x)$:

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

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

NCL subproblems

NLP

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

NC_k

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

Free vars r make the nonlinear constraints independent and feasible

Solvers happy!

NCL subproblems for our problem

NLP

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

NC_k

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

Free vars r make the nonlinear constraints independent and feasible — Solvers happy!

IPMs

$$\underset{x}{\text{minimize}} \;\; \phi(x) \;\; \text{st} \;\; c(x) = 0, \quad x \geq 0$$

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

$$\underset{x,\, r}{\text{minimize}} \;\; \phi(x) + y_k^T r + \tfrac{1}{2}\rho_k \|r\|^2 \;\; \text{st} \;\; c(x) + r = 0, \quad x \geq 0$$

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

Optimal Tax Policy

Kenneth Judd and Che-Lin Su 2011

Optimal tax policy

TAX	$\text{maximize}_{c,\,y}$	$\sum_i \lambda_i U^i(c_i, y_i)$	
	subject to	$U^i(c_i, y_i) - U^i(c_j, y_j) \geq 0$	for all i, j $(*)$
		$\lambda^T(y - c) \geq 0$	
		$c,\ y \geq 0$	

where c_i and y_i are the consumption and income of taxpayer i, and λ is a vector of positive weights. The utility functions $U^i(c_i, y_i)$ are each of the form

$$U(c, y) = \frac{(c - \alpha)^{1 - 1/\gamma}}{1 - 1/\gamma} - \psi \frac{(y/w)^{1/\eta + 1}}{1/\eta + 1}$$

where w is the wage rate and α, γ, ψ and η are taxpayer heterogeneities

$(*)$ = zillions of incentive-compatibility constraints

Optimal tax policy

More precisely,

$$U^{i,j,k,g,h}(c_{p,q,r,s,t}, y_{p,q,r,s,t}) = \frac{(c_{p,q,r,s,t} - \alpha_k)^{1-1/\gamma_h}}{1 - 1/\gamma_h} - \psi_g \frac{(y_{p,q,r,s,t}/w_i)^{1/\eta_j+1}}{1/\eta_j + 1}$$

where (i, j, k, g, h) and (p, q, r, s, t) run over 5 dimensions:

na	wage types	$= 5$	21
nb	elasticities of labor supply	$= 3$	3
nc	basic need types	$= 3$	3
nd	levels of distaste for work	$= 2$	2
ne	elasticities of demand for consumption	$= 2$	2
$T =$	$na \times nb \times nc \times nd \times ne$	$= 180$	756
$m =$	$T(T-1)$ nonlinear constraints	$= 32220$	570780
$n =$	$2T$ variables	$= 360$	1512

AMPL model

TAX	$\text{maximize}_{c,\,y}$	$\sum_i \lambda_i U^i(c_i, y_i)$	
	subject to	$U^i(c_i, y_i) - U^i(c_j, y_j) \geq 0$	for all $i \neq j$
		$\lambda^T(y - c) \geq 0$	
		$c,\ y \geq 0$	

```
Incentive{(i,j,k,g,h) in T, (p,q,r,s,t) in T:
          !(i=p and j=q and k=r and g=s and h=t)}:
      (c[i,j,k,g,h] - alpha[k])^(1-1/gamma[h]) / (1-1/gamma[h])
       - psi[g]*(y[i,j,k,g,h]/w[i])^mu1[j] / mu1[j]
    - (c[p,q,r,s,t] - alpha[k])^(1-1/gamma[h]) / (1-1/gamma[h])
       + psi[g]*(y[p,q,r,s,t]/w[i])^mu1[j] / mu1[j]
      >= 0;

Technology:
   sum{(i,j,k,g,h) in T} lambda[i,j,k,g,h]*(y[i,j,k,g,h] - c[i,j,k,g,h]) >= 0;
```

Piecewise-smooth extension

```
Incentive{(i,j,k,g,h) in T, (p,q,r,s,t) in T:
          !(i=p and j=q and k=r and g=s and h=t)}:
   (if c[i,j,k,g,h] - alpha[k] >= epsilon then
      (c[i,j,k,g,h] - alpha[k])^(1-1/gamma[h]) / (1-1/gamma[h])
       - psi[g]*(y[i,j,k,g,h]/w[i])^mu1[j] / mu1[j]
    else
       -  0.5/gamma[h] *epsilon^(-1/gamma[h]-1)*(c[i,j,k,g,h] - alpha[k])^2
       + (1+1/gamma[h])*epsilon^(-1/gamma[h]  )*(c[i,j,k,g,h] - alpha[k])
       + (1/(1-1/gamma[h]) - 1 - 0.5/gamma[h])*epsilon^(1-1/gamma[h])
       - psi[g]*(y[i,j,k,g,h]/w[i])^mu1[j] / mu1[j]
   )
 - (if c[p,q,r,s,t] - alpha[k] >= epsilon then
      ...
   ) >= 0;
```

SNOPT on problem TAX (1st derivs)

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

Major	Minors	Step	nCon	Feasible	Optimal	MeritFunction	nS	condHz	Penalty	
0	866		1	(3.7E-15)	4.9E-04	4.1745522E+02	4	4.1E+08	1.0E+04	_ r t
1	503	2.7E-02	6	(3.6E-15)	6.5E-02	4.1746922E+02	24	3.2E+05	1.0E+04	_n r t
2	134	1.0E-01	11	(1.4E-07)	2.7E-05	4.1755749E+02	8	2.6E+09	1.8E+06	_s
3	313	9.8E-02	16	(1.4E-07)	8.9E-05	4.1764438E+02	43	1.0E+07	1.8E+06	_
4	153	2.8E-02	21	(5.5E-08)	1.8E-04	4.1767129E+02	35	2.2E+04	1.8E+06	_
5	103	2.2E-02	26	(5.4E-08)	9.5E-04	4.1769616E+02	34	6.7E+07	1.8E+06	_
194	30811	1.0E+00	795	8.6E-01	9.7E-01	2.8330244E+21	2	1.8E+01	3.5E+13	_n it
195	1819	1.1E-04	800	8.6E-01	1.0E+00	2.6326936E+22	3	1.4E+02	1.1E+15	_n R it
195	3314		800	8.6E-01	1.0E+00	2.8661156E+22			1.0E+04	_n r it
195	4439		800	8.6E-01	9.9E-01	2.8661156E+22			1.0E+04	_n r it

```
SNOPTB EXIT  40 -- terminated after numerical difficulties
SNOPTB INFO  41 -- current point cannot be improved
```

IPOPT on problem TAX (2nd derivs)

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

```
This is Ipopt version 3.12.4, running with linear solver mumps.

iter    objective    inf_pr   inf_du lg(mu)  ||d||  lg(rg) alpha_du alpha_pr  ls
   0 -4.1745522e+02 0.00e+00 2.52e+00  -1.0 0.00e+00    -  0.00e+00 0.00e+00   0
   1 -4.1734473e+02 6.18e-03 7.36e+00  -1.0 1.34e+00    -  7.69e-01 2.05e-01f  1
   2 -4.1682694e+02 4.93e-03 1.78e+01  -1.0 5.48e+00    -  2.23e-01 1.34e-01f  1

  10 -4.1428766e+02 1.22e-03 1.50e+04  -1.0 3.01e-01   0.6 4.75e-01 5.39e-01h  1

 160 -4.1641067e+02 0.00e+00 1.50e-03  -3.8 1.25e-01    -  1.00e+00 1.00e+00f  1

 449r-4.1630403e+02 1.13e-05 2.79e-05  -8.1 2.92e-01    -  1.00e+00 9.77e-01h  1

                                   (scaled)                 (unscaled)
Dual infeasibility......:   1.1130803588695777e+00    1.1130803588695777e+00
Constraint violation....:   0.0000000000000000e+00    0.0000000000000000e+00
Complementarity.........:   1.3412941119075164e-08    1.3412941119075164e-08
```

LANCELOT on problem TAX (2nd derivs)

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

k	rhok	omegak	etak	Obj	itns	CGit	TRradius	active
1	1.0e+1	1.0e-1	1.0e-1	-417.455	18	12000	4.1e-01	2831
2	1.0e+1	1.0e-2	1.2e-2	-421.606	39	9000	1.6e-01	2568
3	1.0e+2	1.0e-2	7.9e-2	-421.011	23	11000	2.4e-01	1662
4	1.0e+2	1.0e-4	1.3e-3	-420.188	282	104000	8.6e-02	1444
5	1.0e+3	1.0e-3	6.3e-2	-419.967	134	64000	5.7e-02	1004
6	1.0e+3	1.0e-6	1.3e-4	-419.819	198	156000	3.1e-02	901
7	1.0e+4	1.0e-4	5.0e-2	-419.741	300	308000	3.1e-12	710
8	1.0e+4	1.0e-6	1.3e-5	-419.698	327	623000	5.5e-04	709
9	1.0e+5	1.0e-5	4.0e-2	-419.682	253	724000	4.7e-03	653
10	1.0e+5	1.0e-6	1.3e-6	-419.676	154	1031000	4.2e-11	663
11	1.0e+6	1.0e-6	3.2e-2	...				

1970 iterations, 8 hours CPU on NEOS

AMPL implementation of NCL

pTax5Dnclipopt.run

```
reset;   model pTax5Dinitial.run;    # Get initial values

reset;   model pTax5Dncl.mod;
         data  pTax5Dncl.dat;
         data; var include p5Dinitial.dat;

model;   option solver ipopt;
         option ipopt_options  'dual_inf_tol=1e-6   max_iter=5000';
```

pTax5Dnclipopt.run

```
option opt2 $ipopt_options ' warm_start_init_point=yes';

for {K in 1..kmax}
{  if K == 2 then {option ipopt_options $opt2 ' mu_init=1e-4'};
   if K == 4 then {option ipopt_options $opt2 ' mu_init=1e-5'};
   if K == 6 then {option ipopt_options $opt2 ' mu_init=1e-6'};
   if K == 8 then {option ipopt_options $opt2 ' mu_init=1e-7'};
   if K ==10 then {option ipopt_options $opt2 ' mu_init=1e-8'};

   solve;

   let rmax := max({(i,j,k,g,h) in T, (p,q,r,s,t) in T:
      !(i=p and j=q and k=r and g=s and h=t)} R[i,j,k,g,h,p,q,r,s,t]);
   let rmin := ...
   let rnorm := max(abs(rmax), abs(rmin));
   if rnorm <= rtol then { printf "Stopping: rnorm is small\n"; break; }
```

Numerical results

Interior Methods (IPMs)

FOLKLORE: We don't know how to warm-start IPMs

Interior Methods (IPMs)

FOLKLORE: We don't know how to warm-start IPMs

NCL:

- Sequence of related subproblems
- Only the objective changes: $\phi(x) + y_k^T r + \frac{1}{2}\rho_k \|r\|^2$
- Many extra variables r
- r stabilizes iterations, doesn't affect sparsity of factorizations
- IPOPT, KNITRO have some helpful run-time options

Interior Methods (IPMs)

FOLKLORE: We don't know how to warm-start IPMs

NCL:

- Sequence of related subproblems
- Only the objective changes: $\phi(x) + y_k^T r + \frac{1}{2}\rho_k \|r\|^2$
- Many extra variables r
- r stabilizes iterations, doesn't affect sparsity of factorizations
- IPOPT, KNITRO have some helpful run-time options

In this context, IPM warm starts are practical after all!

Warm-start options for Nonlinear Interior Methods

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

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: $c_i(x) \le tol$

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)

Comparison of IPOPT, KNITRO, NCL (2nd derivs)

na = increasing $nb = 3$ $nc = 3$ $nd = 2$ $ne = 2$

			IPOPT		KNITRO		NCL/IPOPT		NCL/KNITRO	
na	*m*	*n*	itns	time	itns	time	itns	time	itns	time
5	32220	360	449	217	168	53	322	146	2320	8.0mins
9	104652	648	> 98*	> 360*	928	825	655	1023	9697	1.9hrs
11	156420	792	> 87*	∞!	2769	4117	727	1679	26397	7.0hrs
17	373933	1224			2598	11447	1021	6347		
21	570780	1512					1761	17218	45039	1.9 days

*duals diverge
MUMPS needs more mem
!Loop

Warm starts Cold starts

NCL/KNITRO with Warm Starts

na = increasing $nb = 3$ $nc = 3$ $nd = 2$ $ne = 2$

			IPOPT		KNITRO		NCL/IPOPT		NCL/KNITRO	
na	m	n	itns	time	itns	time	itns	time	itns	time
5	32220	360	449	217	168	53	322	146	339	63
9	104652	648	$> 98^*$	$> 360^*$	928	825	655	1023	307	239
11	156420	792	$> 87^*$	$\infty!$	2769	4117	727	1679	383	420
17	373933	1224			2598	11447	1021	6347	486	1200
21	570780	1512					1761	17218	712	2880

Warm starts Warm starts

Julia/NCL

Thanks Dominique and Pierre-Élie!

A Julia Implementation of NCL

Features:

- generic implementation using a full-blown programming language
- rests upon the JuliaSmoothOptimizers[1] infrastructure for optimization
- here, we use the AMPL models for the TAX problems
- can use IPOPT, KNITRO[2] interchangeably

Differences from AMPL/NCL:

- accepts problems modeled with SIF, AMPL, JuMP or plain Julia
- subproblems solved inexactly ($\omega_k \searrow$)
- we are currently experimenting with warm-starting multipliers

[1] https://juliasmoothoptimizers.github.io
[2] Thanks to the authors of IPOPT.jl and to Artelys for supporting KNITRO.jl

Illustration on TAX Problems with Realistic Data

- Use KNITRO 12
- Progressively decrease ω_k
- Stop when $\|r\| \leq$ `feas_tol` and $\|\nabla L\| \leq$ `opt_tol`

```
julia> using NCL

julia> using AmplNLReader  # Julia module to read a nl file

julia> tax1D = AmplModel("data/tax1D")
Maximization problem data/tax1D
nvar = 24, ncon = 133 (1 linear)

julia> NCLSolve(tax1D, outlev=0)

outer  inner    NCL obj      ||r||        η     ||∇L||        ω        ρ   μ init      ||y||      ||x||     time
    1      5  -8.00e+02  9.7e-02  1.0e-02  7.6e-03  1.0e-02  1.0e+02  1.0e-01  1.0e+00  2.0e+02   0.13
    2     12  -7.89e+02  4.2e-02  1.0e-02  4.3e-03  1.0e-02  1.0e+03  1.0e-03  1.0e+00  1.9e+02   0.00
    3      7  -7.83e+02  5.7e-03  1.0e-02  1.0e-03  1.0e-02  1.0e+04  1.0e-03  1.0e+00  1.9e+02   0.00
    4      3  -7.82e+02  1.3e-04  1.0e-03  1.0e-05  1.0e-03  1.0e+04  1.0e-05  5.8e+01  1.9e+02   0.00
    5      2  -7.82e+02  2.3e-06  1.0e-04  1.0e-05  1.0e-04  1.0e+04  1.0e-05  5.9e+01  1.9e+02   0.00
    6      2  -7.82e+02  9.3e-08  1.0e-05  1.0e-06  1.0e-05  1.0e+04  1.0e-06  5.9e+01  1.9e+02   0.00
    7      2  -7.82e+02  7.7e-09  1.0e-06  1.0e-08  1.0e-06  1.0e+04  1.0e-06  5.9e+01  1.9e+02   0.00
```

TAX Problems with Realistic Data

```
julia> tax2D = AmplModel("data/tax2D")
Maximization problem data/tax2D
nvar = 120, ncon = 3541 (1 linear)

julia> NCLSolve(tax2D, outlev=0)
outer  inner    NCL obj      ||r||        η     ||∇L||        ω         ρ   μ init      ||y||      ||x||     time
    1     16  -4.35e+03  6.1e-02  1.0e-02  4.2e-03  1.0e-02  1.0e+02  1.0e-01  1.0e+00  4.0e+02    0.15
    2     15  -4.31e+03  2.5e-02  1.0e-02  2.7e-04  1.0e-02  1.0e+03  1.0e-03  1.0e+00  4.0e+02    0.13
    3     16  -4.29e+03  7.8e-03  1.0e-02  3.5e-04  1.0e-02  1.0e+04  1.0e-03  1.0e+00  4.0e+02    0.16
    4     15  -4.28e+03  5.1e-03  1.0e-03  1.0e-05  1.0e-03  1.0e+04  1.0e-05  7.9e+01  4.0e+02    0.14
    5     32  -4.28e+03  1.2e-03  1.0e-03  1.0e-05  1.0e-03  1.0e+05  1.0e-05  7.9e+01  4.0e+02    0.32
    6     12  -4.28e+03  1.5e-04  1.0e-03  1.5e-05  1.0e-03  1.0e+06  1.0e-06  7.9e+01  4.0e+02    0.15
    7      4  -4.28e+03  1.8e-05  1.0e-04  2.7e-06  1.0e-04  1.0e+06  1.0e-06  2.0e+02  4.0e+02    0.06
    8      4  -4.28e+03  1.2e-06  1.0e-05  1.3e-07  1.0e-05  1.0e+06  1.0e-07  2.0e+02  4.0e+02    0.05
    9      3  -4.28e+03  3.5e-07  1.0e-06  1.0e-07  1.0e-06  1.0e+06  1.0e-07  2.0e+02  4.0e+02    0.05
```

TAX Problems with Realistic Data

```
julia> pTax3D = AmplModel("data/pTax3D")
Maximization problem data/pTax3D
nvar = 216, ncon = 11557 (1 linear)

julia> NCLSolve(pTax3D, outlev=0)
outer  inner    NCL obj     ||r||        η       ||∇L||        ω         ρ    μ init      ||y||      ||x||     time
    1      9  -6.97e+03   4.5e-02  1.0e-02  9.1e-03  1.0e-02  1.0e+02  1.0e-01  1.0e+00  5.7e+02   0.54
    2     18  -6.87e+03   1.7e-02  1.0e-02  2.4e-04  1.0e-02  1.0e+03  1.0e-03  1.0e+00  5.6e+02   0.99
    3     16  -6.83e+03   7.8e-03  1.0e-02  1.7e-03  1.0e-02  1.0e+04  1.0e-03  1.0e+00  5.7e+02   1.01
    4     17  -6.81e+03   5.2e-03  1.0e-03  1.5e-05  1.0e-03  1.0e+04  1.0e-05  7.9e+01  5.6e+02   0.99
    5     54  -6.80e+03   2.6e-03  1.0e-03  1.2e-05  1.0e-03  1.0e+05  1.0e-05  7.9e+01  5.6e+02   3.15
    6     22  -6.80e+03   4.5e-04  1.0e-03  8.0e-05  1.0e-03  1.0e+06  1.0e-06  7.9e+01  5.6e+02   1.30
    7      9  -6.80e+03   1.1e-04  1.0e-04  1.0e-06  1.0e-04  1.0e+06  1.0e-06  5.2e+02  5.6e+02   0.56
    8      8  -6.80e+03   1.1e-05  1.0e-04  1.1e-07  1.0e-04  1.0e+07  1.0e-07  5.2e+02  5.6e+02   0.49
    9      5  -6.80e+03   1.1e-06  1.0e-05  1.0e-07  1.0e-05  1.0e+07  1.0e-07  5.3e+02  5.6e+02   0.32
   10      3  -6.80e+03   8.9e-08  1.0e-06  1.0e-08  1.0e-06  1.0e+07  1.0e-08  5.3e+02  5.6e+02   0.22
```

TAX Problems with Realistic Data

```
julia> pTax4D = AmplModel("data/pTax4D")
Minimization problem data/pTax4D
nvar = 432, ncon = 46441 (1 linear)

julia> NCLSolve(pTax4D, outlev=0)
outer  inner    NCL obj     ||r||        η      ||∇L||        ω         ρ      μ init     ||y||      ||x||     time
    1     12  -1.34e+04  3.3e-02  1.0e-02  5.4e-03  1.0e-02  1.0e+02  1.0e-01  1.0e+00  7.2e+02    3.38
    2     12  -1.31e+04  1.3e-02  1.0e-02  4.0e-03  1.0e-02  1.0e+03  1.0e-03  1.0e+00  7.2e+02    3.23
    3     15  -1.30e+04  5.1e-03  1.0e-02  2.1e-04  1.0e-02  1.0e+04  1.0e-03  1.0e+00  7.1e+02    3.86
    4     31  -1.30e+04  3.2e-03  1.0e-03  1.3e-05  1.0e-03  1.0e+04  1.0e-05  5.2e+01  7.0e+02    7.95
    5     37  -1.30e+04  1.8e-03  1.0e-03  1.2e-05  1.0e-03  1.0e+05  1.0e-05  5.2e+01  7.0e+02    9.89
    6     44  -1.29e+04  5.0e-04  1.0e-03  1.1e-06  1.0e-03  1.0e+06  1.0e-06  5.2e+01  7.0e+02   11.93
    7     16  -1.29e+04  2.6e-04  1.0e-04  1.2e-05  1.0e-04  1.0e+06  1.0e-06  5.3e+02  7.0e+02    3.74
    8     30  -1.29e+04  4.4e-05  1.0e-04  1.2e-07  1.0e-04  1.0e+07  1.0e-07  5.3e+02  7.0e+02    8.15
    9      9  -1.29e+04  2.3e-05  1.0e-05  1.2e-07  1.0e-05  1.0e+07  1.0e-07  8.2e+02  7.0e+02    2.49
   10     11  -1.29e+04  3.8e-06  1.0e-05  1.0e-08  1.0e-05  1.0e+08  1.0e-08  8.2e+02  7.0e+02    3.09
   11      6  -1.29e+04  1.7e-07  1.0e-06  1.3e-08  1.0e-06  1.0e+08  1.0e-08  9.4e+02  7.0e+02    1.74
```

TAX Problems with Realistic Data

```
julia> pTax5D = AmplModel("data/pTax5D")
Minimization problem data/pTax5D
nvar = 864, ncon = 186193 (1 linear)

julia> NCLSolve(pTax5D, outlev=0)
outer  inner    NCL obj    ||r||        η     ||∇L||        ω        ρ   μ init     ||y||     ||x||     time
    1     64  -1.76e+05  2.0e-01  1.0e-02  2.3e-03  1.0e-02  1.0e+02  1.0e-01  1.0e+00  1.1e+04    80.43
    2     29  -1.74e+05  4.9e-02  1.0e-02  1.2e-03  1.0e-02  1.0e+03  1.0e-03  1.0e+00  1.1e+04    35.02
    3     23  -1.74e+05  1.6e-02  1.0e-02  1.0e-03  1.0e-02  1.0e+04  1.0e-03  1.0e+00  1.1e+04    28.96
    4     46  -1.74e+05  4.1e-03  1.0e-02  3.6e-05  1.0e-02  1.0e+05  1.0e-05  1.0e+00  1.1e+04    54.50
    5     41  -1.74e+05  2.8e-03  1.0e-03  1.7e-05  1.0e-03  1.0e+05  1.0e-05  4.1e+02  1.1e+04    52.72
    6     28  -1.74e+05  6.1e-04  1.0e-03  1.0e-06  1.0e-03  1.0e+06  1.0e-06  4.1e+02  1.1e+04    34.38
    7     13  -1.74e+05  2.1e-04  1.0e-04  1.4e-06  1.0e-04  1.0e+06  1.0e-06  1.0e+03  1.1e+04    14.81
    8     12  -1.74e+05  5.3e-05  1.0e-04  1.2e-07  1.0e-04  1.0e+07  1.0e-07  1.0e+03  1.1e+04    14.80
    9      7  -1.74e+05  4.5e-06  1.0e-05  1.0e-07  1.0e-05  1.0e+07  1.0e-07  1.0e+03  1.1e+04     9.49
   10      5  -1.74e+05  8.0e-07  1.0e-06  1.2e-08  1.0e-06  1.0e+07  1.0e-08  1.0e+03  1.1e+04     7.02
```

Related work

- C. M. Maes, A Regularized Active-Set Method for Sparse Convex Quadratic Programming. PhD thesis, ICME, Stanford University, 2010.
- M. P. Friedlander and D. Orban, A primal-dual regularized interior-point method for convex quadratic programs. Math. Prog. Comp., 4(1):71–107, 2012.
- S. Arreckx and D. Orban, A regularized factorization-free method for equality-constrained optimization, Technical Report GERAD G-2016-65, GERAD, Montréal, QC, Canada, 2016, doi:10.13140/RG.2.2.20368.00007.
- 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.

- P. E. Gill, V. Kungurtsev, and D. P. Robinson, A stabilized SQP method: global convergence, IMA J. Numer. Anal., 37 (2017), 407–443.
- P. E. Gill, V. Kungurtsev, and D. P. Robinson, A stabilized SQP method: superlinear convergence, Math. Program., Ser. A, 163 (2017), 369–410.

- O. Hinder and Y. Ye, A one-phase IPM for nonconvex optimization, Oliver's MS&E PhD thesis (2019).

Special thanks

- LANCELOT: Andy Conn, Nick Gould, Philippe Toint
- AMPL: Bob Fourer, Dave Gay
- Julia developers
- IPOPT: Larry Biegler, Carl Laird, Andreas Wächter
- KNITRO: Richard Waltz, Jorge Nocedal, Todd Plantenga, Richard Byrd
- Coauthors: Ken Judd, Che-Lin Su, Ding Ma, Dominique Orban
- Pierre-Élie Personnaz
- Yuja Wang, YouTube (and YouKu!)

Coauthors Ken, Che-Lin, Dominique, Ding