Modular Codes for NLP, QP, and KKT Systems

Marc C. Steinbach (ZIB)
http://www.zib.de/steinbach

2005-06-01

Outline

- Motivation: DAE Boundary Value Problems
- Algorithm: Hierarchical Solution Approach
- Codes
- Further Problem Classes

Model Problem

ODE Boundary Value Problem (ODE-BVP) on $I = [0, 1]$ with $0 = t_1 < \cdots < t_k = 1$:

$$\begin{aligned} \min_{x,u} \quad & \sum_{i=1}^{k} \phi_i(x(t_i), u(t_i)) \\ \text{subject to} \quad & r(x(t_1), \ldots, x(t_k)) = 0 \\ & \dot{x}(t) - f(x(t), u(t)) = 0 \quad \text{on } I \\ & g(x(t), u(t)) \geq 0 \quad \text{on } I \end{aligned}$$

Second Order Decoupling

$$r(x_1, \ldots, x_k) = \begin{pmatrix} r^0(x_1, \ldots, x_k) \\ r^1(x_1, \ldots, x_k) \end{pmatrix} = \begin{pmatrix} r_1^0(x_1) \\ \vdots \\ r_k^0(x_k) \\ r_1^1(x_1) + \cdots + r_k^1(x_k) \end{pmatrix}$$

Model Problem

General Case

- DAE-BVP (index 1)
- $r(x_1, u_1, \ldots, x_k, u_k)$
- Non-autonomous BVP, t_i free, parameters p
- Multi-phase BVP
- BVP with branching (orbit transfer, . . .)

$\longrightarrow$ Trajectory Optimization, Feedback Control, . . .

Discretized BVP

Multistage NLP

$$\min_{x,u} \quad \sum_{i=1}^{k} \phi_i(x_i, u_i)$$

$$\text{subject to} \qquad r_i^0(x_i, u_i) = 0 \qquad i = 1, \dots k$$

$$\sum_{i=1}^{k} r_i^1(x_i, u_i) = 0$$

$$f_i(x_i, u_i) - x_{i+i} = 0 \qquad i = 1, \dots k-1$$

$$g_i(x_i, u_i) \geq 0 \qquad i = 1, \dots k$$

Discretized BVP

Hierarchical Solution Approach

- NLP solved iteratively by SQP; step direction from CQP subproblem
- CQP solved iteratively by IPM; step direction from KKT subproblem
- KKT system solved "directly"

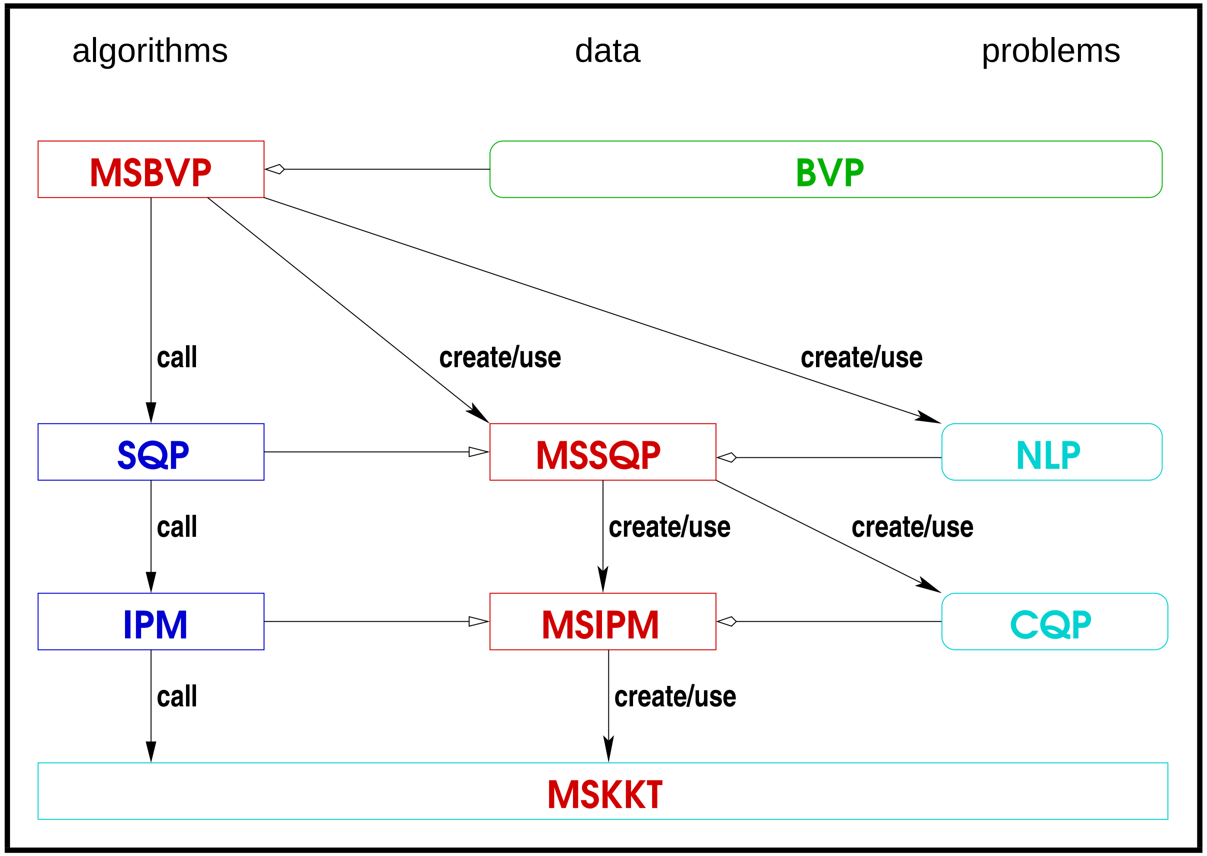
Observations

- NLP, CQP, and KKT subproblems have multistage structure
- SQP and IPM iterations generic: structure irrelevant
- KKT system often dominates computational effort
- KKT solver problem-specific: exploit structure

$\longrightarrow$ careful modular implementation rather than "SQP method for BVP"

Codes

Multistage BVP Modules (C Implementation)



Codes

Features

SQP and IPM

- Exchangeable subproblem solvers
  - QP solver in SQP
  - KKT solver in IPM
- SQP uses hot-started IPM

MSKKT

- Solves general multistage CQP
  – here BVP discretization
  – also rigid multibody dynamics (index 1 descriptor form)
- Implements dense block operations using BLAS, LAPACK

Further Problem Classes

Multistage Stochastic Programs

- NLP (typically LP) on scenario tree
- Often large trees: 10^4–10^6 nodes
- Often sparse blocks
- General problem comes in three flavors:
  - implicit control: $g_i(y_i) = f_i(y_{i\_})$
  - outgoing control: $x_i = f_i(x_{i\_}, u_{i\_})$
  - incoming control: $x_i = f_i(x_{i\_}, u_i)$
- KKT solvers generated as C++ source code
  (model-specific classes wrapped within generic interface)
- Block operations alternatively dense or element-wise

Further Problem Classes

Operative Planning in Supply Networks (Gas/Water)

Custom KKT solvers: work in progress

- Sparse block operations: spatial projections (network topology)
- Dense block operations: recursions over time
- Ill-conditioning, iterative refinement
- ...

Code Status

- Mostly research
- Some components included in DEVA (commercial portfolio optimization package)