Interior-Point Methods for Nonlinear Programming

Robert Vanderbei

July 1, 2001

Erice

Operations Research and Financial Engineering, Princeton University

http://www.princeton.edu/~rvdb

1 Acknowledgements

- Newton
- Lagrange
- Fiacco, McCormick
- Karmarkar
- Megiddo

- Gay
- Conn, Gould, Toint, Bongartz

- Shanno, Benson

1 Acknowledgements

- Newton
- Lagrange
- Fiacco, McCormick
- Karmarkar
- Megiddo

- Gay
- Conn, Gould, Toint, Bongartz

- Shanno, Benson

1 Acknowledgements

- Newton
- Lagrange
- Fiacco, McCormick
- Karmarkar
- Megiddo

- Gay
- Conn, Gould, Toint, Bongartz

- Shanno, Benson

1 Acknowledgements

- Newton
- Lagrange
- Fiacco, McCormick
- Karmarkar
- Megiddo

- Gay
- Conn, Gould, Toint, Bongartz

- Shanno, Benson

1 Acknowledgements

- Newton
- Lagrange
- Fiacco, McCormick
- Karmarkar
- Megiddo
- Gay
- Conn, Gould, Toint, Bongartz
- Shanno, Benson

1 Acknowledgements

- Newton
- Lagrange
- Fiacco, McCormick
- Karmarkar
- Megiddo

- Gay
- Conn, Gould, Toint, Bongartz

- Shanno, Benson

1 Acknowledgements

- Newton
- Lagrange
- Fiacco, McCormick
- Karmarkar
- Megiddo

- Gay
- Conn, Gould, Toint, Bongartz

- Shanno, Benson

1 Acknowledgements

- Newton
- Lagrange
- Fiacco, McCormick
- Karmarkar
- Megiddo

- Gay
- Conn, Gould, Toint, Bongartz

- Shanno, Benson

1 Acknowledgements

- Newton
- Lagrange
- Fiacco, McCormick
- Karmarkar
- Megiddo

- Gay
- Conn, Gould, Toint, Bongartz

- Shanno, Benson

2 Outline

- Algorithm
  - Basic Paradigm
  - Step-Length Control
  - Diagonal Perturbation
  - Jamming
  - Free Variables

- Some Applications
  - Celestial Mechanics
  - Putting on an Uneven Green
  - Goddard Rocket Problem

Outline

- Algorithm
  - Basic Paradigm
  - Step-Length Control
  - Diagonal Perturbation
  - Jamming
  - Free Variables
- Some Applications
  - Celestial Mechanics
  - Putting on an Uneven Green
  - Goddard Rocket Problem

Outline

- Algorithm
  - Basic Paradigm
  - Step-Length Control
  - Diagonal Perturbation
  - Jamming
  - Free Variables
- Some Applications
  - Celestial Mechanics
  - Putting on an Uneven Green
  - Goddard Rocket Problem

2 Outline

- Algorithm
  - Basic Paradigm
  - Step-Length Control
  - Diagonal Perturbation
  - Jamming
  - Free Variables

- Some Applications
  - Celestial Mechanics
  - Putting on an Uneven Green
  - Goddard Rocket Problem

2 Outline

- Algorithm
  - Basic Paradigm
  - Step-Length Control
  - Diagonal Perturbation
  - Jamming
  - Free Variables

- Some Applications
  - Celestial Mechanics
  - Putting on an Uneven Green
  - Goddard Rocket Problem

Outline

- Algorithm
  - Basic Paradigm
  - Step-Length Control
  - Diagonal Perturbation
  - Jamming
  - Free Variables
- Some Applications
  - Celestial Mechanics
  - Putting on an Uneven Green
  - Goddard Rocket Problem

2 Outline

- Algorithm
  - Basic Paradigm
  - Step-Length Control
  - Diagonal Perturbation
  - Jamming
  - Free Variables
- Some Applications
  - Celestial Mechanics
  - Putting on an Uneven Green
  - Goddard Rocket Problem

2 Outline

- Algorithm
  - Basic Paradigm
  - Step-Length Control
  - Diagonal Perturbation
  - Jamming
  - Free Variables

- Some Applications
  - Celestial Mechanics
  - Putting on an Uneven Green
  - Goddard Rocket Problem

Outline

- Algorithm
  - Basic Paradigm
  - Step-Length Control
  - Diagonal Perturbation
  - Jamming
  - Free Variables
- Some Applications
  - Celestial Mechanics
  - Putting on an Uneven Green
  - Goddard Rocket Problem

2 Outline

- Algorithm
  - Basic Paradigm
  - Step-Length Control
  - Diagonal Perturbation
  - Jamming
  - Free Variables
- Some Applications
  - Celestial Mechanics
  - Putting on an Uneven Green
  - Goddard Rocket Problem

Outline

- Algorithm
  - Basic Paradigm
  - Step-Length Control
  - Diagonal Perturbation
  - Jamming
  - Free Variables
- Some Applications
  - Celestial Mechanics
  - Putting on an Uneven Green
  - Goddard Rocket Problem

The Interior-Point Algorithm

3 Introduce Slack Variables

- Start with an optimization problem—for now, the simplest NLP:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & h_i(x) \geq 0, \qquad i = 1, \ldots, m \end{array}$$

- Introduce slack variables to make all inequality constraints into nonnegativities:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & h(x) - w = 0, \\ & w \geq 0 \end{array}$$

3 Introduce Slack Variables

- Start with an optimization problem—for now, the simplest NLP:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & h_i(x) \geq 0, \qquad i = 1, \ldots, m \end{array}$$

- Introduce slack variables to make all inequality constraints into nonnegativities:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & h(x) - w = 0, \\ & w \geq 0 \end{array}$$

3 Introduce Slack Variables

- Start with an optimization problem—for now, the simplest NLP:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & h_i(x) \geq 0, \qquad i = 1, \ldots, m \end{array}$$

- Introduce slack variables to make all inequality constraints into nonnegativities:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & h(x) - w = 0, \\ & w \geq 0 \end{array}$$

4 Associated Log-Barrier Problem

- Replace nonnegativity constraints with logarithmic barrier terms in the objective:

$$\begin{array}{ll} \text{minimize} & f(x) - \mu \sum_{i=1}^{m} \log(w_i) \\ \text{subject to} & h(x) - w = 0 \end{array}$$

4 Associated Log-Barrier Problem

- Replace nonnegativity constraints with logarithmic barrier terms in the objective:

$$\begin{array}{ll} \text{minimize} & f(x) - \mu \sum_{i=1}^{m} \log(w_i) \\ \text{subject to} & h(x) - w = 0 \end{array}$$

5 First-Order Optimality Conditions

- Incorporate the equality constraints into the objective using Lagrange multipliers:

$$L(x, w, y) = f(x) - \mu \sum_{i=1}^{m} \log(w_i) - y^T(h(x) - w)$$

- Set all derivatives to zero:

$$\begin{aligned} \nabla f(x) - \nabla h(x)^T y &= 0 \\ -\mu W^{-1} e + y &= 0 \\ h(x) - w &= 0 \end{aligned}$$

5 First-Order Optimality Conditions

- Incorporate the equality constraints into the objective using Lagrange multipliers:

$$L(x, w, y) = f(x) - \mu \sum_{i=1}^{m} \log(w_i) - y^T(h(x) - w)$$

- Set all derivatives to zero:

$$\begin{aligned} \nabla f(x) - \nabla h(x)^T y &= 0 \\ -\mu W^{-1} e + y &= 0 \\ h(x) - w &= 0 \end{aligned}$$

5 First-Order Optimality Conditions

- Incorporate the equality constraints into the objective using Lagrange multipliers:

$$L(x, w, y) = f(x) - \mu \sum_{i=1}^{m} \log(w_i) - y^T(h(x) - w)$$

- Set all derivatives to zero:

$$\begin{aligned} \nabla f(x) - \nabla h(x)^T y &= 0 \\ -\mu W^{-1} e + y &= 0 \\ h(x) - w &= 0 \end{aligned}$$

6 Symmetrize Complementarity Conditions

- Rewrite system:

$$
\begin{aligned}
\nabla f(x) - \nabla h(x)^T y &= 0 \\
WYe &= \mu e \\
h(x) - w &= 0
\end{aligned}
$$

6 Symmetrize Complementarity Conditions

- Rewrite system:

$$\begin{aligned} \nabla f(x) - \nabla h(x)^T y &= 0 \\ WYe &= \mu e \\ h(x) - w &= 0 \end{aligned}$$

7 Apply Newton's Method

- Apply Newton's method to compute search directions, Δx, Δw, Δy:

$$\begin{bmatrix} H(x,y) & 0 & -A(x)^T \\ 0 & Y & W \\ A(x) & -I & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta w \\ \Delta y \end{bmatrix} = \begin{bmatrix} -\nabla f(x) + A(x)^T y \\ \mu e - WYe \\ -h(x) + w \end{bmatrix}.$$

Here,

$$H(x,y) = \nabla^2 f(x) - \sum_{i=1}^m y_i \nabla^2 h_i(x)$$

and

$$A(x) = \nabla h(x)$$

- Note: $H(x,y)$ is positive semidefinite if f is convex, each h_i is concave, and each $y_i \geq 0$.

7 Apply Newton's Method

- Apply Newton's method to compute search directions, Δx, Δw, Δy:

$$\begin{bmatrix} H(x,y) & 0 & -A(x)^T \\ 0 & Y & W \\ A(x) & -I & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta w \\ \Delta y \end{bmatrix} = \begin{bmatrix} -\nabla f(x) + A(x)^T y \\ \mu e - WYe \\ -h(x) + w \end{bmatrix}.$$

Here,

$$H(x,y) = \nabla^2 f(x) - \sum_{i=1}^m y_i \nabla^2 h_i(x)$$

and

$$A(x) = \nabla h(x)$$

- Note: $H(x,y)$ is positive semidefinite if f is convex, each h_i is concave, and each $y_i \geq 0$.

7 Apply Newton's Method

- Apply Newton's method to compute search directions, Δx, Δw, Δy:

$$\begin{bmatrix} H(x,y) & 0 & -A(x)^T \\ 0 & Y & W \\ A(x) & -I & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta w \\ \Delta y \end{bmatrix} = \begin{bmatrix} -\nabla f(x) + A(x)^T y \\ \mu e - WYe \\ -h(x) + w \end{bmatrix}.$$

  Here,

$$H(x,y) = \nabla^2 f(x) - \sum_{i=1}^m y_i \nabla^2 h_i(x)$$

  and

$$A(x) = \nabla h(x)$$

- Note: $H(x,y)$ is positive semidefinite if f is convex, each h_i is concave, and each $y_i \geq 0$.

8 Reduced KKT System

- Use second equation to solve for Δw. Result is the reduced KKT system:

$$\begin{bmatrix} -H(x,y) & A^T(x) \\ A(x) & WY^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \nabla f(x) - A^T(x)y \\ -h(x) + \mu Y^{-1}e \end{bmatrix}$$

- Iterate:

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} \Delta x^{(k)}$$
$$w^{(k+1)} = w^{(k)} + \alpha^{(k)} \Delta w^{(k)}$$
$$y^{(k+1)} = y^{(k)} + \alpha^{(k)} \Delta y^{(k)}$$

8 Reduced KKT System

- Use second equation to solve for Δw. Result is the reduced KKT system:

$$\begin{bmatrix} -H(x,y) & A^T(x) \\ A(x) & WY^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \nabla f(x) - A^T(x)y \\ -h(x) + \mu Y^{-1}e \end{bmatrix}$$

- Iterate:

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} \Delta x^{(k)}$$
$$w^{(k+1)} = w^{(k)} + \alpha^{(k)} \Delta w^{(k)}$$
$$y^{(k+1)} = y^{(k)} + \alpha^{(k)} \Delta y^{(k)}$$

8 Reduced KKT System

- Use second equation to solve for Δw. Result is the reduced KKT system:

$$\begin{bmatrix} -H(x,y) & A^T(x) \\ A(x) & WY^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \nabla f(x) - A^T(x)y \\ -h(x) + \mu Y^{-1}e \end{bmatrix}$$

- Iterate:

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} \Delta x^{(k)}$$
$$w^{(k+1)} = w^{(k)} + \alpha^{(k)} \Delta w^{(k)}$$
$$y^{(k+1)} = y^{(k)} + \alpha^{(k)} \Delta y^{(k)}$$

9 Modifications for Convex Optimization

For convex nonquadratic optimization, it does not suffice to choose the steplength α simply to maintain positivity of nonnegative variables.

- Consider, e.g., minimizing

$$f(x) = (1 + x^2)^{1/2}.$$

- The iterates can be computed explicitly:

$$x^{(k+1)} = -(x^{(k)})^3$$

- Converges if and only if $|x| \leq 1$.
- Reason: away from 0, function is too linear.

9 Modifications for Convex Optimization

For convex nonquadratic optimization, it does not suffice to choose the steplength α simply to maintain positivity of nonnegative variables.

- Consider, e.g., minimizing

$$f(x) = (1 + x^2)^{1/2}.$$

- The iterates can be computed explicitly:

$$x^{(k+1)} = -(x^{(k)})^3$$

- Converges if and only if $|x| \leq 1$.
- Reason: away from 0, function is too linear.

9 Modifications for Convex Optimization

For convex nonquadratic optimization, it does not suffice to choose the steplength α simply to maintain positivity of nonnegative variables.

- Consider, e.g., minimizing

$$f(x) = (1 + x^2)^{1/2}.$$

- The iterates can be computed explicitly:

$$x^{(k+1)} = -(x^{(k)})^3$$

- Converges if and only if $|x| \leq 1$.
- Reason: away from 0, function is too linear.

9 Modifications for Convex Optimization

For convex nonquadratic optimization, it does not suffice to choose the steplength α simply to maintain positivity of nonnegative variables.

- Consider, e.g., minimizing

$$f(x) = (1 + x^2)^{1/2}.$$

- The iterates can be computed explicitly:

$$x^{(k+1)} = -(x^{(k)})^3$$

- Converges if and only if $|x| \leq 1$.
- Reason: away from 0, function is too linear.

9 Modifications for Convex Optimization

For convex nonquadratic optimization, it does not suffice to choose the steplength α simply to maintain positivity of nonnegative variables.

- Consider, e.g., minimizing

$$f(x) = (1 + x^2)^{1/2}.$$

- The iterates can be computed explicitly:

$$x^{(k+1)} = -(x^{(k)})^3$$

- Converges if and only if $|x| \leq 1$.
- Reason: away from 0, function is too linear.

10 Step-Length Control: Merit Function

A merit function is used to guide the choice of steplength α.

We use the Fiacco–McCormick merit function

$$\Psi_{\beta,\mu}(x, w) = f(x) - \mu \sum_{i=1}^{m} \log(w_i) + \frac{\beta}{2} \|h(x) - w\|_2^2.$$

Define the dual normal matrix:

$$N(x, y, w) = H(x, y) + A^T(x) W^{-1} Y A(x).$$

Theorem 1 *Suppose that $N(x, y, w)$ is positive definite.*

1. *For β sufficiently large, $(\Delta x, \Delta w)$ is a descent direction for $\Psi_{\beta,\mu}$.*
2. *If current solution is primal feasible, then $(\Delta x, \Delta w)$ is a descent direction for the barrier function.*

Note: minimum required value for β is easy to compute.

10 Step-Length Control: Merit Function

A merit function is used to guide the choice of steplength α.

We use the Fiacco–McCormick merit function

$$\Psi_{\beta,\mu}(x,w) = f(x) - \mu \sum_{i=1}^{m} \log(w_i) + \frac{\beta}{2}\|h(x) - w\|_2^2.$$

Define the dual normal matrix:

$$N(x,y,w) = H(x,y) + A^T(x)W^{-1}YA(x).$$

Theorem 1 *Suppose that $N(x,y,w)$ is positive definite.*

1. *For β sufficiently large, $(\Delta x, \Delta w)$ is a descent direction for $\Psi_{\beta,\mu}$.*
2. *If current solution is primal feasible, then $(\Delta x, \Delta w)$ is a descent direction for the barrier function.*

Note: minimum required value for β is easy to compute.

10 Step-Length Control: Merit Function

A merit function is used to guide the choice of steplength α.

We use the Fiacco–McCormick merit function

$$\Psi_{\beta,\mu}(x,w) = f(x) - \mu \sum_{i=1}^{m} \log(w_i) + \frac{\beta}{2}\|h(x) - w\|_2^2.$$

Define the dual normal matrix:

$$N(x,y,w) = H(x,y) + A^T(x)W^{-1}YA(x).$$

Theorem 1 *Suppose that $N(x,y,w)$ is positive definite.*

1. *For β sufficiently large, $(\Delta x, \Delta w)$ is a descent direction for $\Psi_{\beta,\mu}$.*
2. *If current solution is primal feasible, then $(\Delta x, \Delta w)$ is a descent direction for the barrier function.*

Note: minimum required value for β is easy to compute.

10 Step-Length Control: Merit Function

A merit function is used to guide the choice of steplength α.

We use the Fiacco–McCormick merit function

$$\Psi_{\beta,\mu}(x,w) = f(x) - \mu \sum_{i=1}^{m} \log(w_i) + \frac{\beta}{2}\|h(x) - w\|_2^2.$$

Define the dual normal matrix:

$$N(x,y,w) = H(x,y) + A^T(x) W^{-1} Y A(x).$$

Theorem 1 *Suppose that $N(x,y,w)$ is positive definite.*

1. *For β sufficiently large, $(\Delta x, \Delta w)$ is a descent direction for $\Psi_{\beta,\mu}$.*
2. *If current solution is primal feasible, then $(\Delta x, \Delta w)$ is a descent direction for the barrier function.*

Note: minimum required value for β is easy to compute.

Step-Length Control: Merit Function

A merit function is used to guide the choice of steplength α.

We use the Fiacco–McCormick merit function

$$\Psi_{\beta,\mu}(x, w) = f(x) - \mu \sum_{i=1}^{m} \log(w_i) + \frac{\beta}{2} \|h(x) - w\|_2^2.$$

Define the dual normal matrix:

$$N(x, y, w) = H(x, y) + A^T(x) W^{-1} Y A(x).$$

Theorem 1 *Suppose that $N(x, y, w)$ is positive definite.*

1. *For β sufficiently large, $(\Delta x, \Delta w)$ is a descent direction for $\Psi_{\beta,\mu}$.*
2. *If current solution is primal feasible, then $(\Delta x, \Delta w)$ is a descent direction for the barrier function.*

Note: minimum required value for β is easy to compute.

10 Step-Length Control: Merit Function

A merit function is used to guide the choice of steplength α.

We use the Fiacco–McCormick merit function

$$\Psi_{\beta,\mu}(x, w) = f(x) - \mu \sum_{i=1}^{m} \log(w_i) + \frac{\beta}{2}\|h(x) - w\|_2^2.$$

Define the dual normal matrix:

$$N(x, y, w) = H(x, y) + A^T(x)W^{-1}YA(x).$$

Theorem 1 *Suppose that $N(x, y, w)$ is positive definite.*

1. *For β sufficiently large, $(\Delta x, \Delta w)$ is a descent direction for $\Psi_{\beta,\mu}$.*
2. *If current solution is primal feasible, then $(\Delta x, \Delta w)$ is a descent direction for the barrier function.*

Note: minimum required value for β is easy to compute.

10 Step-Length Control: Merit Function

A merit function is used to guide the choice of steplength α.

We use the Fiacco–McCormick merit function

$$\Psi_{\beta,\mu}(x,w) = f(x) - \mu \sum_{i=1}^{m} \log(w_i) + \frac{\beta}{2}\|h(x) - w\|_2^2.$$

Define the dual normal matrix:

$$N(x,y,w) = H(x,y) + A^T(x)W^{-1}YA(x).$$

Theorem 1 *Suppose that $N(x,y,w)$ is positive definite.*

1. *For β sufficiently large, $(\Delta x, \Delta w)$ is a descent direction for $\Psi_{\beta,\mu}$.*
2. *If current solution is primal feasible, then $(\Delta x, \Delta w)$ is a descent direction for the barrier function.*

Note: minimum required value for β is easy to compute.

10 Step-Length Control: Merit Function

A merit function is used to guide the choice of steplength α.

We use the Fiacco–McCormick merit function

$$\Psi_{\beta,\mu}(x, w) = f(x) - \mu \sum_{i=1}^{m} \log(w_i) + \frac{\beta}{2}\|h(x) - w\|_2^2.$$

Define the dual normal matrix:

$$N(x, y, w) = H(x, y) + A^T(x)W^{-1}YA(x).$$

Theorem 1 *Suppose that $N(x, y, w)$ is positive definite.*

1. *For β sufficiently large, $(\Delta x, \Delta w)$ is a descent direction for $\Psi_{\beta,\mu}$.*
2. *If current solution is primal feasible, then $(\Delta x, \Delta w)$ is a descent direction for the barrier function.*

Note: minimum required value for β is easy to compute.

11 Nonconvex Optimization: Diagonal Perturbation

- If $H(x, y)$ is not positive semidefinite then $N(x, y, w)$ might fail to be positive definite.
- In such a case, we lose the descent properties given in previous theorem.
- To regain those properties, we perturb the Hessian: $\tilde{H}(x, y) = H(x, y) + \lambda I$.
- And compute search directions using $\tilde{H}$ instead of H.

Notation: let $\tilde{N}$ denote the dual normal matrix associated with $\tilde{H}$.

Theorem 2 *If $\tilde{N}$ is positive definite, then $(\Delta x, \Delta w, \Delta y)$ is a descent direction for*

1. *the primal infeasibility, $\|h(x) - w\|$;*
2. *the noncomplementarity, $w^T y$.*

11 Nonconvex Optimization: Diagonal Perturbation

- If $H(x, y)$ is not positive semidefinite then $N(x, y, w)$ might fail to be positive definite.
- In such a case, we lose the descent properties given in previous theorem.
- To regain those properties, we perturb the Hessian: $\tilde{H}(x, y) = H(x, y) + \lambda I$.
- And compute search directions using $\tilde{H}$ instead of H.

Notation: let $\tilde{N}$ denote the dual normal matrix associated with $\tilde{H}$.

Theorem 2 *If $\tilde{N}$ is positive definite, then $(\Delta x, \Delta w, \Delta y)$ is a descent direction for*

1. *the primal infeasibility, $\|h(x) - w\|$;*
2. *the noncomplementarity, $w^T y$.*

Nonconvex Optimization: Diagonal Perturbation

- If $H(x, y)$ is not positive semidefinite then $N(x, y, w)$ might fail to be positive definite.
- In such a case, we lose the descent properties given in previous theorem.
- To regain those properties, we perturb the Hessian: $\tilde{H}(x, y) = H(x, y) + \lambda I$.
- And compute search directions using $\tilde{H}$ instead of H.

Notation: let $\tilde{N}$ denote the dual normal matrix associated with $\tilde{H}$.

Theorem 2 *If $\tilde{N}$ is positive definite, then $(\Delta x, \Delta w, \Delta y)$ is a descent direction for*

1. *the primal infeasibility, $\|h(x) - w\|$;*
2. *the noncomplementarity, $w^T y$.*

11 Nonconvex Optimization: Diagonal Perturbation

- If $H(x, y)$ is not positive semidefinite then $N(x, y, w)$ might fail to be positive definite.
- In such a case, we lose the descent properties given in previous theorem.
- To regain those properties, we perturb the Hessian: $\tilde{H}(x, y) = H(x, y) + \lambda I$.
- And compute search directions using $\tilde{H}$ instead of H.

Notation: let $\tilde{N}$ denote the dual normal matrix associated with $\tilde{H}$.

Theorem 2 *If $\tilde{N}$ is positive definite, then $(\Delta x, \Delta w, \Delta y)$ is a descent direction for*

1. *the primal infeasibility, $\|h(x) - w\|$;*
2. *the noncomplementarity, $w^T y$.*

11 Nonconvex Optimization: Diagonal Perturbation

- If $H(x,y)$ is not positive semidefinite then $N(x,y,w)$ might fail to be positive definite.
- In such a case, we lose the descent properties given in previous theorem.
- To regain those properties, we perturb the Hessian: $\tilde{H}(x,y) = H(x,y) + \lambda I$.
- And compute search directions using $\tilde{H}$ instead of H.

Notation: let $\tilde{N}$ denote the dual normal matrix associated with $\tilde{H}$.

Theorem 2 *If $\tilde{N}$ is positive definite, then $(\Delta x, \Delta w, \Delta y)$ is a descent direction for*

1. *the primal infeasibility, $\|h(x) - w\|$;*
2. *the noncomplementarity, $w^T y$.*

11 Nonconvex Optimization: Diagonal Perturbation

- If $H(x, y)$ is not positive semidefinite then $N(x, y, w)$ might fail to be positive definite.
- In such a case, we lose the descent properties given in previous theorem.
- To regain those properties, we perturb the Hessian: $\tilde{H}(x, y) = H(x, y) + \lambda I$.
- And compute search directions using $\tilde{H}$ instead of H.

Notation: let $\tilde{N}$ denote the dual normal matrix associated with $\tilde{H}$.

Theorem 2 *If $\tilde{N}$ is positive definite, then $(\Delta x, \Delta w, \Delta y)$ is a descent direction for*

1. *the primal infeasibility, $\|h(x) - w\|$;*
2. *the noncomplementarity, $w^T y$.*

11 Nonconvex Optimization: Diagonal Perturbation

- If $H(x, y)$ is not positive semidefinite then $N(x, y, w)$ might fail to be positive definite.
- In such a case, we lose the descent properties given in previous theorem.
- To regain those properties, we perturb the Hessian: $\tilde{H}(x, y) = H(x, y) + \lambda I$.
- And compute search directions using $\tilde{H}$ instead of H.

Notation: let $\tilde{N}$ denote the dual normal matrix associated with $\tilde{H}$.

Theorem 2 *If $\tilde{N}$ is positive definite, then $(\Delta x, \Delta w, \Delta y)$ is a descent direction for*

1. *the primal infeasibility, $\|h(x) - w\|$;*
2. *the noncomplementarity, $w^T y$.*

11 Nonconvex Optimization: Diagonal Perturbation

- If $H(x, y)$ is not positive semidefinite then $N(x, y, w)$ might fail to be positive definite.
- In such a case, we lose the descent properties given in previous theorem.
- To regain those properties, we perturb the Hessian: $\tilde{H}(x, y) = H(x, y) + \lambda I$.
- And compute search directions using $\tilde{H}$ instead of H.

Notation: let $\tilde{N}$ denote the dual normal matrix associated with $\tilde{H}$.

Theorem 2 *If $\tilde{N}$ is positive definite, then $(\Delta x, \Delta w, \Delta y)$ is a descent direction for*

1. *the primal infeasibility, $\|h(x) - w\|$;*
2. *the noncomplementarity, $w^T y$.*

11 Nonconvex Optimization: Diagonal Perturbation

- If $H(x, y)$ is not positive semidefinite then $N(x, y, w)$ might fail to be positive definite.
- In such a case, we lose the descent properties given in previous theorem.
- To regain those properties, we perturb the Hessian: $\tilde{H}(x, y) = H(x, y) + \lambda I$.
- And compute search directions using $\tilde{H}$ instead of H.

Notation: let $\tilde{N}$ denote the dual normal matrix associated with $\tilde{H}$.

Theorem 2 *If $\tilde{N}$ is positive definite, then $(\Delta x, \Delta w, \Delta y)$ is a descent direction for*

1. *the primal infeasibility, $\|h(x) - w\|$;*
2. *the noncomplementarity, $w^T y$.*

11 Nonconvex Optimization: Diagonal Perturbation

- If $H(x, y)$ is not positive semidefinite then $N(x, y, w)$ might fail to be positive definite.
- In such a case, we lose the descent properties given in previous theorem.
- To regain those properties, we perturb the Hessian: $\tilde{H}(x, y) = H(x, y) + \lambda I$.
- And compute search directions using $\tilde{H}$ instead of H.

Notation: let $\tilde{N}$ denote the dual normal matrix associated with $\tilde{H}$.

Theorem 2 *If $\tilde{N}$ is positive definite, then $(\Delta x, \Delta w, \Delta y)$ is a descent direction for*

1. *the primal infeasibility, $\|h(x) - w\|$;*
2. *the noncomplementarity, $w^T y$.*

Notes:

- Not necessarily a descent direction for dual infeasibility.
- A line search is performed to find a value of λ within a factor of 2 of the smallest permissible value.

Notes:

- Not necessarily a descent direction for dual infeasibility.

- A line search is performed to find a value of λ within a factor of 2 of the smallest permissible value.

Notes:

- Not necessarily a descent direction for dual infeasibility.

- A line search is performed to find a value of λ within a factor of 2 of the smallest permissible value.

13 Nonconvex Optimization: Jamming

Theorem 3 *If the problem is convex and and the current solution is not optimal and ..., then for any slack variable, say* w_i*, we have* $w_i = 0$ *implies* $\Delta w_i \geq 0$*.*

- To paraphrase: for convex problems, as slack variables get small they tend to get large again. This is an antijamming theorem.
- A recent example of Wächter and Biegler shows that for nonconvex problems, jamming really can occur.
- Recent modification:
  - if a slack variable gets small and
  - its component of the step direction contributes to making a very short step,
  - then increase this slack variable to the average size of the variables the “mainstream” slack variables.
- This modification corrects all examples of jamming that we know about.

13 Nonconvex Optimization: Jamming

Theorem 3 *If the problem is convex and and the current solution is not optimal and ..., then for any slack variable, say* w_i*, we have* $w_i = 0$ *implies* $\Delta w_i \geq 0$*.*

- To paraphrase: for convex problems, as slack variables get small they tend to get large again. This is an antijamming theorem.
- A recent example of Wächter and Biegler shows that for nonconvex problems, jamming really can occur.
- Recent modification:
  - if a slack variable gets small and
  - its component of the step direction contributes to making a very short step,
  - then increase this slack variable to the average size of the variables the "mainstream" slack variables.
- This modification corrects all examples of jamming that we know about.

Nonconvex Optimization: Jamming

Theorem 3 *If the problem is convex and and the current solution is not optimal and ..., then for any slack variable, say* w_i*, we have* $w_i = 0$ *implies* $\Delta w_i \geq 0$*.*

- To paraphrase: for convex problems, as slack variables get small they tend to get large again. This is an antijamming theorem.
- A recent example of Wächter and Biegler shows that for nonconvex problems, jamming really can occur.
- Recent modification:
  - if a slack variable gets small and
  - its component of the step direction contributes to making a very short step,
  - then increase this slack variable to the average size of the variables the "mainstream" slack variables.
- This modification corrects all examples of jamming that we know about.

13 Nonconvex Optimization: Jamming

Theorem 3 *If the problem is convex and and the current solution is not optimal and ..., then for any slack variable, say* w_i*, we have* $w_i = 0$ *implies* $\Delta w_i \geq 0$*.*

- To paraphrase: for convex problems, as slack variables get small they tend to get large again. This is an antijamming theorem.
- A recent example of Wächter and Biegler shows that for nonconvex problems, jamming really can occur.
- Recent modification:
  - if a slack variable gets small and
  - its component of the step direction contributes to making a very short step,
  - then increase this slack variable to the average size of the variables the "mainstream" slack variables.
- This modification corrects all examples of jamming that we know about.

13 Nonconvex Optimization: Jamming

Theorem 3 *If the problem is convex and and the current solution is not optimal and ..., then for any slack variable, say* w_i*, we have* $w_i = 0$ *implies* $\Delta w_i \geq 0$*.*

- To paraphrase: for convex problems, as slack variables get small they tend to get large again. This is an antijamming theorem.
- A recent example of Wächter and Biegler shows that for nonconvex problems, jamming really can occur.
- Recent modification:
  - if a slack variable gets small and
  - its component of the step direction contributes to making a very short step,
  - then increase this slack variable to the average size of the variables the "mainstream" slack variables.
- This modification corrects all examples of jamming that we know about.

13 Nonconvex Optimization: Jamming

Theorem 3 *If the problem is convex and and the current solution is not optimal and ..., then for any slack variable, say* w_i, *we have* $w_i = 0$ *implies* $\Delta w_i \geq 0$.

- To paraphrase: for convex problems, as slack variables get small they tend to get large again. This is an antijamming theorem.
- A recent example of Wächter and Biegler shows that for nonconvex problems, jamming really can occur.
- Recent modification:
  - if a slack variable gets small and
  - its component of the step direction contributes to making a very short step,
  - then increase this slack variable to the average size of the variables the "mainstream" slack variables.
- This modification corrects all examples of jamming that we know about.

13 Nonconvex Optimization: Jamming

Theorem 3 *If the problem is convex and and the current solution is not optimal and ..., then for any slack variable, say* w_i*, we have* $w_i = 0$ *implies* $\Delta w_i \geq 0$.

- To paraphrase: for convex problems, as slack variables get small they tend to get large again. This is an antijamming theorem.
- A recent example of Wächter and Biegler shows that for nonconvex problems, jamming really can occur.
- Recent modification:
  - if a slack variable gets small and
  - its component of the step direction contributes to making a very short step,
  - then increase this slack variable to the average size of the variables the "mainstream" slack variables.
- This modification corrects all examples of jamming that we know about.

13 Nonconvex Optimization: Jamming

Theorem 3 *If the problem is convex and and the current solution is not optimal and ..., then for any slack variable, say* w_i*, we have* $w_i = 0$ *implies* $\Delta w_i \geq 0$*.*

- To paraphrase: for convex problems, as slack variables get small they tend to get large again. This is an antijamming theorem.
- A recent example of Wächter and Biegler shows that for nonconvex problems, jamming really can occur.
- Recent modification:
  - if a slack variable gets small and
  - its component of the step direction contributes to making a very short step,
  - then increase this slack variable to the average size of the variables the "mainstream" slack variables.
- This modification corrects all examples of jamming that we know about.

13 Nonconvex Optimization: Jamming

Theorem 3 *If the problem is convex and and the current solution is not optimal and ..., then for any slack variable, say* w_i, *we have* $w_i = 0$ *implies* $\Delta w_i \geq 0$.

- To paraphrase: for convex problems, as slack variables get small they tend to get large again. This is an antijamming theorem.
- A recent example of Wächter and Biegler shows that for nonconvex problems, jamming really can occur.
- Recent modification:
  - if a slack variable gets small and
  - its component of the step direction contributes to making a very short step,
  - then increase this slack variable to the average size of the variables the "mainstream" slack variables.
- This modification corrects all examples of jamming that we know about.

14 Modifications for General Problem Formulations

- Bounds, ranges, and free variables are all treated implicitly as described in Linear Programming: Foundations and Extensions (LP:F&E).
- Net result is following reduced KKT system:

$$\begin{bmatrix} -(H(x,y)+D) & A^T(x) \\ A(x) & E \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix}$$

- Here, D and E are positive definite diagonal matrices.
- Note that D helps reduce frequency of diagonal perturbation.
- Choice of barrier parameter μ and initial solution, if none is provided, is described in the paper.
- Stopping rules, matrix reordering heuristics, etc. are as described in LP:F&E.

14 Modifications for General Problem Formulations

- Bounds, ranges, and free variables are all treated implicitly as described in Linear Programming: Foundations and Extensions (LP:F&E).
- Net result is following reduced KKT system:

$$\begin{bmatrix} -(H(x,y)+D) & A^T(x) \\ A(x) & E \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix}$$

- Here, D and E are positive definite diagonal matrices.
- Note that D helps reduce frequency of diagonal perturbation.
- Choice of barrier parameter μ and initial solution, if none is provided, is described in the paper.
- Stopping rules, matrix reordering heuristics, etc. are as described in LP:F&E.

14 Modifications for General Problem Formulations

- Bounds, ranges, and free variables are all treated implicitly as described in Linear Programming: Foundations and Extensions (LP:F&E).
- Net result is following reduced KKT system:

$$\begin{bmatrix} -(H(x,y)+D) & A^T(x) \\ A(x) & E \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix}$$

- Here, D and E are positive definite diagonal matrices.
- Note that D helps reduce frequency of diagonal perturbation.
- Choice of barrier parameter μ and initial solution, if none is provided, is described in the paper.
- Stopping rules, matrix reordering heuristics, etc. are as described in LP:F&E.

Modifications for General Problem Formulations

- Bounds, ranges, and free variables are all treated implicitly as described in Linear Programming: Foundations and Extensions (LP:F&E).
- Net result is following reduced KKT system:

$$\begin{bmatrix} -(H(x,y)+D) & A^T(x) \\ A(x) & E \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix}$$

- Here, D and E are positive definite diagonal matrices.
- Note that D helps reduce frequency of diagonal perturbation.
- Choice of barrier parameter μ and initial solution, if none is provided, is described in the paper.
- Stopping rules, matrix reordering heuristics, etc. are as described in LP:F&E.

14 Modifications for General Problem Formulations

- Bounds, ranges, and free variables are all treated implicitly as described in Linear Programming: Foundations and Extensions (LP:F&E).
- Net result is following reduced KKT system:

$$\begin{bmatrix} -(H(x,y)+D) & A^T(x) \\ A(x) & E \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix}$$

- Here, D and E are positive definite diagonal matrices.
- Note that D helps reduce frequency of diagonal perturbation.
- Choice of barrier parameter μ and initial solution, if none is provided, is described in the paper.
- Stopping rules, matrix reordering heuristics, etc. are as described in LP:F&E.

14 Modifications for General Problem Formulations

- Bounds, ranges, and free variables are all treated implicitly as described in Linear Programming: Foundations and Extensions (LP:F&E).
- Net result is following reduced KKT system:

$$\begin{bmatrix} -(H(x,y)+D) & A^T(x) \\ A(x) & E \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix}$$

- Here, D and E are positive definite diagonal matrices.
- Note that D helps reduce frequency of diagonal perturbation.
- Choice of barrier parameter μ and initial solution, if none is provided, is described in the paper.
- Stopping rules, matrix reordering heuristics, etc. are as described in LP:F&E.

14 Modifications for General Problem Formulations

- Bounds, ranges, and free variables are all treated implicitly as described in Linear Programming: Foundations and Extensions (LP:F&E).
- Net result is following reduced KKT system:

$$\begin{bmatrix} -(H(x,y)+D) & A^T(x) \\ A(x) & E \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix}$$

- Here, D and E are positive definite diagonal matrices.
- Note that D helps reduce frequency of diagonal perturbation.
- Choice of barrier parameter μ and initial solution, if none is provided, is described in the paper.
- Stopping rules, matrix reordering heuristics, etc. are as described in LP:F&E.

15 Focus of Free Variables

- For each free variable, say x_j, we introduce a new constraint expressing the variable as a difference between two nonnegative variables:

$$x_j = g_j - t_j, \quad g_j \geq 0, \quad t_j \geq 0.$$

- The variable x_j is not removed from the problem.
- The Newton system involves new rows/columns corresponding to the new constraints and variables.
- These new rows/columns are eliminated algebraically to produce a reduced KKT system with the original dimensions.
- The net result is an entry in the diagonal matrix D in $H(x, y) + D$.
- Letting d_{jj} denote the diagonal entry of D, we have

$$d_{jj} = \left(\frac{g_j}{z_j} + \frac{t_j}{s_j} \right)^{-1},$$

  where z_j (s_j) is a dual variable complementary to g_j (t_j, respectively).
- We see that the net effect is a regularization of the reduced KKT system.

15 Focus of Free Variables

- For each free variable, say x_j, we introduce a new constraint expressing the variable as a difference between two nonnegative variables:

$$x_j = g_j - t_j, \quad g_j \geq 0, \quad t_j \geq 0.$$

- The variable x_j is not removed from the problem.
- The Newton system involves new rows/columns corresponding to the new constraints and variables.
- These new rows/columns are eliminated algebraically to produce a reduced KKT system with the original dimensions.
- The net result is an entry in the diagonal matrix D in $H(x, y) + D$.
- Letting d_{jj} denote the diagonal entry of D, we have

$$d_{jj} = \left(\frac{g_j}{z_j} + \frac{t_j}{s_j} \right)^{-1},$$

  where z_j (s_j) is a dual variable complementary to g_j (t_j, respectively).
- We see that the net effect is a regularization of the reduced KKT system.

Focus of Free Variables

- For each free variable, say x_j, we introduce a new constraint expressing the variable as a difference between two nonnegative variables:

$$x_j = g_j - t_j, \quad g_j \geq 0, \quad t_j \geq 0.$$

- The variable x_j is not removed from the problem.
- The Newton system involves new rows/columns corresponding to the new constraints and variables.
- These new rows/columns are eliminated algebraically to produce a reduced KKT system with the original dimensions.
- The net result is an entry in the diagonal matrix D in $H(x, y) + D$.
- Letting d_{jj} denote the diagonal entry of D, we have

$$d_{jj} = \left(\frac{g_j}{z_j} + \frac{t_j}{s_j} \right)^{-1},$$

  where z_j (s_j) is a dual variable complementary to g_j (t_j, respectively).
- We see that the net effect is a regularization of the reduced KKT system.

15 Focus of Free Variables

- For each free variable, say x_j, we introduce a new constraint expressing the variable as a difference between two nonnegative variables:

$$x_j = g_j - t_j, \quad g_j \geq 0, \quad t_j \geq 0.$$

- The variable x_j is not removed from the problem.
- The Newton system involves new rows/columns corresponding to the new constraints and variables.
- These new rows/columns are eliminated algebraically to produce a reduced KKT system with the original dimensions.
- The net result is an entry in the diagonal matrix D in $H(x, y) + D$.
- Letting d_{jj} denote the diagonal entry of D, we have

$$d_{jj} = \left(\frac{g_j}{z_j} + \frac{t_j}{s_j} \right)^{-1},$$

  where z_j (s_j) is a dual variable complementary to g_j (t_j, respectively).
- We see that the net effect is a regularization of the reduced KKT system.

15 Focus of Free Variables

- For each free variable, say x_j, we introduce a new constraint expressing the variable as a difference between two nonnegative variables:

$$x_j = g_j - t_j, \quad g_j \geq 0, \quad t_j \geq 0.$$

- The variable x_j is not removed from the problem.
- The Newton system involves new rows/columns corresponding to the new constraints and variables.
- These new rows/columns are eliminated algebraically to produce a reduced KKT system with the original dimensions.
- The net result is an entry in the diagonal matrix D in $H(x, y) + D$.
- Letting d_{jj} denote the diagonal entry of D, we have

$$d_{jj} = \left(\frac{g_j}{z_j} + \frac{t_j}{s_j} \right)^{-1},$$

  where z_j (s_j) is a dual variable complementary to g_j (t_j, respectively).
- We see that the net effect is a regularization of the reduced KKT system.

15 Focus of Free Variables

- For each free variable, say x_j, we introduce a new constraint expressing the variable as a difference between two nonnegative variables:

$$x_j = g_j - t_j, \quad g_j \geq 0, \quad t_j \geq 0.$$

- The variable x_j is not removed from the problem.
- The Newton system involves new rows/columns corresponding to the new constraints and variables.
- These new rows/columns are eliminated algebraically to produce a reduced KKT system with the original dimensions.
- The net result is an entry in the diagonal matrix D in $H(x, y) + D$.
- Letting d_{jj} denote the diagonal entry of D, we have

$$d_{jj} = \left(\frac{g_j}{z_j} + \frac{t_j}{s_j} \right)^{-1},$$

  where z_j (s_j) is a dual variable complementary to g_j (t_j, respectively).
- We see that the net effect is a regularization of the reduced KKT system.

15 Focus of Free Variables

- For each free variable, say x_j, we introduce a new constraint expressing the variable as a difference between two nonnegative variables:

$$x_j = g_j - t_j, \quad g_j \geq 0, \quad t_j \geq 0.$$

- The variable x_j is not removed from the problem.
- The Newton system involves new rows/columns corresponding to the new constraints and variables.
- These new rows/columns are eliminated algebraically to produce a reduced KKT system with the original dimensions.
- The net result is an entry in the diagonal matrix D in $H(x, y) + D$.
- Letting d_{jj} denote the diagonal entry of D, we have

$$d_{jj} = \left(\frac{g_j}{z_j} + \frac{t_j}{s_j} \right)^{-1},$$

  where z_j (s_j) is a dual variable complementary to g_j (t_j, respectively).
- We see that the net effect is a regularization of the reduced KKT system.

15 Focus of Free Variables

- For each free variable, say x_j, we introduce a new constraint expressing the variable as a difference between two nonnegative variables:

$$x_j = g_j - t_j, \quad g_j \geq 0, \quad t_j \geq 0.$$

- The variable x_j is not removed from the problem.
- The Newton system involves new rows/columns corresponding to the new constraints and variables.
- These new rows/columns are eliminated algebraically to produce a reduced KKT system with the original dimensions.
- The net result is an entry in the diagonal matrix D in $H(x, y) + D$.
- Letting d_{jj} denote the diagonal entry of D, we have

$$d_{jj} = \left(\frac{g_j}{z_j} + \frac{t_j}{s_j} \right)^{-1},$$

  where z_j (s_j) is a dual variable complementary to g_j (t_j, respectively).
- We see that the net effect is a regularization of the reduced KKT system.

Some Applications

Celestial Mechanics—Periodic Orbits

- Find periodic orbits for the planar gravitational n-body problem.
- Minimize action:

$$\int_0^{2\pi} (K(t) - P(t))dt,$$

- where $K(t)$ is kinetic energy,

$$K(t) = \frac{1}{2} \sum_i \left(\dot{x}_i^2(t) + \dot{y}_i^2(t) \right),$$

- and $P(t)$ is potential energy,

$$P(t) = -\sum_{i<j} \frac{1}{\sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2}}.$$

- Subject to periodicity constraints:

$$x_i(2\pi) = x_i(0), \qquad y_i(2\pi) = y_i(0).$$

Celestial Mechanics—Periodic Orbits

- Find periodic orbits for the planar gravitational n-body problem.
- Minimize action:

$$\int_0^{2\pi} (K(t) - P(t))dt,$$

- where $K(t)$ is kinetic energy,

$$K(t) = \frac{1}{2} \sum_i \left(\dot{x}_i^2(t) + \dot{y}_i^2(t) \right),$$

- and $P(t)$ is potential energy,

$$P(t) = -\sum_{i<j} \frac{1}{\sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2}}.$$

- Subject to periodicity constraints:

$$x_i(2\pi) = x_i(0), \qquad y_i(2\pi) = y_i(0).$$

Celestial Mechanics—Periodic Orbits

- Find periodic orbits for the planar gravitational n-body problem.
- Minimize action:

$$\int_0^{2\pi} (K(t) - P(t))dt,$$

- where $K(t)$ is kinetic energy,

$$K(t) = \frac{1}{2} \sum_i \left(\dot{x}_i^2(t) + \dot{y}_i^2(t) \right),$$

- and $P(t)$ is potential energy,

$$P(t) = -\sum_{i<j} \frac{1}{\sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2}}.$$

- Subject to periodicity constraints:

$$x_i(2\pi) = x_i(0), \qquad y_i(2\pi) = y_i(0).$$

Celestial Mechanics—Periodic Orbits

- Find periodic orbits for the planar gravitational n-body problem.
- Minimize action:

$$\int_0^{2\pi} (K(t) - P(t))dt,$$

- where $K(t)$ is kinetic energy,

$$K(t) = \frac{1}{2}\sum_i \left(\dot{x}_i^2(t) + \dot{y}_i^2(t)\right),$$

- and $P(t)$ is potential energy,

$$P(t) = -\sum_{i<j} \frac{1}{\sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2}}.$$

- Subject to periodicity constraints:

$$x_i(2\pi) = x_i(0), \qquad y_i(2\pi) = y_i(0).$$

Celestial Mechanics—Periodic Orbits

- Find periodic orbits for the planar gravitational n-body problem.
- Minimize action:

$$\int_0^{2\pi} (K(t) - P(t))dt,$$

- where $K(t)$ is kinetic energy,

$$K(t) = \frac{1}{2} \sum_i \left(\dot{x}_i^2(t) + \dot{y}_i^2(t) \right),$$

- and $P(t)$ is potential energy,

$$P(t) = -\sum_{i<j} \frac{1}{\sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2}}.$$

- Subject to periodicity constraints:

$$x_i(2\pi) = x_i(0), \qquad y_i(2\pi) = y_i(0).$$

16 Celestial Mechanics—Periodic Orbits

- Find periodic orbits for the planar gravitational n-body problem.
- Minimize action:

$$\int_0^{2\pi} (K(t) - P(t))dt,$$

- where $K(t)$ is kinetic energy,

$$K(t) = \frac{1}{2} \sum_i \left(\dot{x}_i^2(t) + \dot{y}_i^2(t) \right),$$

- and $P(t)$ is potential energy,

$$P(t) = -\sum_{i<j} \frac{1}{\sqrt{(x_i(t) - x_j(t))^2 + (y_i(t) - y_j(t))^2}}.$$

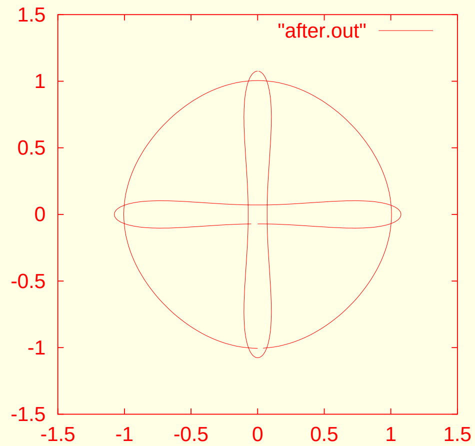
- Subject to periodicity constraints:

$$x_i(2\pi) = x_i(0), \qquad y_i(2\pi) = y_i(0).$$

17 Specific Example

Orbits.mod with $n = 3$ and $(0, 2\pi)$ discretized into a 160 pieces gives the following results:

constraints	0	
variables	960	
time (secs)		
LOQO	1.1	
LANCELOT	8.7	
SNOPT	287	(no change for last 80% of iterations)



18 Putting on an Uneven Green

Given:

- $z(x, y)$ elevation of the green.
- Starting position of the ball (x_0, y_0).
- Position of hole (x_f, y_f).
- Coefficient of friction μ.

Find: initial velocity vector so that ball will roll to the hole and arrive with minimal speed.

Variables:

- $u(t) = (x(t), y(t), z(t))$—position as a function of time t.
- $v(t) = (v_x(t), v_y(t), v_z(t))$—velocity.
- $a(t) = (a_x(t), a_y(t), a_z(t))$—acceleration.
- T—time at which ball arrives at hole.

18 Putting on an Uneven Green

Given:

- $z(x, y)$ elevation of the green.
- Starting position of the ball (x_0, y_0).
- Position of hole (x_f, y_f).
- Coefficient of friction μ.

Find: initial velocity vector so that ball will roll to the hole and arrive with minimal speed.

Variables:

- $u(t) = (x(t), y(t), z(t))$—position as a function of time t.
- $v(t) = (v_x(t), v_y(t), v_z(t))$—velocity.
- $a(t) = (a_x(t), a_y(t), a_z(t))$—acceleration.
- T—time at which ball arrives at hole.

18 Putting on an Uneven Green

Given:

- $z(x, y)$ elevation of the green.
- Starting position of the ball (x_0, y_0).
- Position of hole (x_f, y_f).
- Coefficient of friction μ.

Find: initial velocity vector so that ball will roll to the hole and arrive with minimal speed.

Variables:

- $u(t) = (x(t), y(t), z(t))$—position as a function of time t.
- $v(t) = (v_x(t), v_y(t), v_z(t))$—velocity.
- $a(t) = (a_x(t), a_y(t), a_z(t))$—acceleration.
- T—time at which ball arrives at hole.

18 Putting on an Uneven Green

Given:

- $z(x, y)$ elevation of the green.
- Starting position of the ball (x_0, y_0).
- Position of hole (x_f, y_f).
- Coefficient of friction μ.

Find: initial velocity vector so that ball will roll to the hole and arrive with minimal speed.

Variables:

- $u(t) = (x(t), y(t), z(t))$—position as a function of time t.
- $v(t) = (v_x(t), v_y(t), v_z(t))$—velocity.
- $a(t) = (a_x(t), a_y(t), a_z(t))$—acceleration.
- T—time at which ball arrives at hole.

19 Putting—Two Approaches

- Problem can be formulated with two decision variables:

  $$v_x(0) \quad \text{and} \quad v_y(0)$$

  and two constraints:

  $$x(T) = x_f \quad \text{and} \quad y(T) = y_f.$$

  In this case, $x(T)$, $y(T)$, and the objective function are complicated functions of the two variables that can only be computed by integrating the appropriate differential equation.

- A discretization of the complete trajectory (including position, velocity, and acceleration) can be taken as variables and the physical laws encoded in the differential equation can be written as constraints.

To implement the first approach, one would need an ode integrator that provides, in addition to the quantities being sought, first and possibly second derivatives of those quantities with respect to the decision variables.

The modern trend is to follow the second approach.

19 Putting—Two Approaches

- Problem can be formulated with two decision variables:

$$v_x(0) \quad \text{and} \quad v_y(0)$$

  and two constraints:

$$x(T) = x_f \quad \text{and} \quad y(T) = y_f.$$

  In this case, $x(T)$, $y(T)$, and the objective function are complicated functions of the two variables that can only be computed by integrating the appropriate differential equation.

- A discretization of the complete trajectory (including position, velocity, and acceleration) can be taken as variables and the physical laws encoded in the differential equation can be written as constraints.

To implement the first approach, one would need an ode integrator that provides, in addition to the quantities being sought, first and possibly second derivatives of those quantities with respect to the decision variables.

The modern trend is to follow the second approach.

19 Putting—Two Approaches

- Problem can be formulated with two decision variables:

$$v_x(0) \quad \text{and} \quad v_y(0)$$

  and two constraints:

$$x(T) = x_f \quad \text{and} \quad y(T) = y_f.$$

  In this case, $x(T)$, $y(T)$, and the objective function are complicated functions of the two variables that can only be computed by integrating the appropriate differential equation.

- A discretization of the complete trajectory (including position, velocity, and acceleration) can be taken as variables and the physical laws encoded in the differential equation can be written as constraints.

To implement the first approach, one would need an ode integrator that provides, in addition to the quantities being sought, first and possibly second derivatives of those quantities with respect to the decision variables.

The modern trend is to follow the second approach.

19 Putting—Two Approaches

- Problem can be formulated with two decision variables:

$$v_x(0) \quad \text{and} \quad v_y(0)$$

  and two constraints:

$$x(T) = x_f \quad \text{and} \quad y(T) = y_f.$$

  In this case, $x(T)$, $y(T)$, and the objective function are complicated functions of the two variables that can only be computed by integrating the appropriate differential equation.

- A discretization of the complete trajectory (including position, velocity, and acceleration) can be taken as variables and the physical laws encoded in the differential equation can be written as constraints.

To implement the first approach, one would need an ode integrator that provides, in addition to the quantities being sought, first and possibly second derivatives of those quantities with respect to the decision variables.

The modern trend is to follow the second approach.

19 Putting—Two Approaches

- Problem can be formulated with two decision variables:

$$v_x(0) \quad \text{and} \quad v_y(0)$$

  and two constraints:

$$x(T) = x_f \quad \text{and} \quad y(T) = y_f.$$

  In this case, $x(T)$, $y(T)$, and the objective function are complicated functions of the two variables that can only be computed by integrating the appropriate differential equation.

- A discretization of the complete trajectory (including position, velocity, and acceleration) can be taken as variables and the physical laws encoded in the differential equation can be written as constraints.

To implement the first approach, one would need an ode integrator that provides, in addition to the quantities being sought, first and possibly second derivatives of those quantities with respect to the decision variables.

The modern trend is to follow the second approach.

Putting—Continued

Objective:

$$\text{minimize } v_x(T)^2 + v_y(T)^2.$$

Constraints:

$$\begin{aligned} v &= \dot{u} \\ a &= \dot{v} \\ ma &= N + F - mge_z \\ u(0) &= u_0 \qquad u(T) = u_f, \end{aligned}$$

where

- m is the mass of the golf ball.
- g is the acceleration due to gravity.
- e_z is a unit vector in the positive z direction.

and ...

Putting—Continued

Objective:

$$\text{minimize } v_x(T)^2 + v_y(T)^2.$$

Constraints:

$$\begin{aligned} v &= \dot{u} \\ a &= \dot{v} \\ ma &= N + F - mge_z \\ u(0) &= u_0 \qquad u(T) = u_f, \end{aligned}$$

where

- m is the mass of the golf ball.
- g is the acceleration due to gravity.
- e_z is a unit vector in the positive z direction.

and ...

Putting—Continued

Objective:

$$\text{minimize } v_x(T)^2 + v_y(T)^2.$$

Constraints:

$$\begin{aligned} v &= \dot{u} \\ a &= \dot{v} \\ ma &= N + F - mge_z \\ u(0) &= u_0 \qquad u(T) = u_f, \end{aligned}$$

where

- m is the mass of the golf ball.
- g is the acceleration due to gravity.
- e_z is a unit vector in the positive z direction.

and ...

Putting—Continued

Objective:

$$\text{minimize } v_x(T)^2 + v_y(T)^2.$$

Constraints:

$$\begin{aligned} v &= \dot{u} \\ a &= \dot{v} \\ ma &= N + F - mge_z \\ u(0) &= u_0 \qquad u(T) = u_f, \end{aligned}$$

where

- m is the mass of the golf ball.
- g is the acceleration due to gravity.
- e_z is a unit vector in the positive z direction.

and ...

Putting—Continued

- $N = (N_x, N_y, N_z)$ is the normal force:

$$N_z = m \frac{g - a_x(t)\frac{\partial z}{\partial x} - a_y(t)\frac{\partial z}{\partial y} + a_z(t)}{(\frac{\partial z}{\partial x})^2 + (\frac{\partial z}{\partial y})^2 + 1}$$

$$N_x = -\frac{\partial z}{\partial x} N_z$$

$$N_y = -\frac{\partial z}{\partial y} N_z.$$

- F is the force due to friction:

$$F = -\mu \|N\| \frac{v}{\|v\|}.$$

21 Putting—Continued

- $N = (N_x, N_y, N_z)$ is the normal force:

$$\begin{aligned} N_z &= m\frac{g - a_x(t)\frac{\partial z}{\partial x} - a_y(t)\frac{\partial z}{\partial y} + a_z(t)}{(\frac{\partial z}{\partial x})^2 + (\frac{\partial z}{\partial y})^2 + 1} \\ N_x &= -\frac{\partial z}{\partial x} N_z \\ N_y &= -\frac{\partial z}{\partial y} N_z. \end{aligned}$$

- F is the force due to friction:

$$F = -\mu \|N\| \frac{v}{\|v\|}.$$

21 Putting—Continued

- $N = (N_x, N_y, N_z)$ is the normal force:

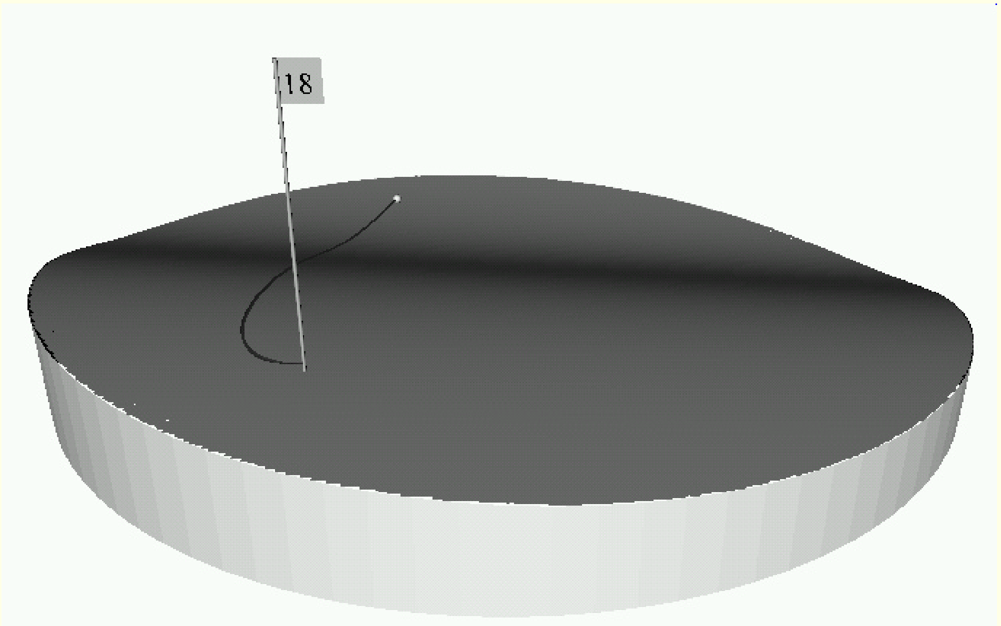
$$
\begin{aligned}
N_z &= m\frac{g - a_x(t)\frac{\partial z}{\partial x} - a_y(t)\frac{\partial z}{\partial y} + a_z(t)}{(\frac{\partial z}{\partial x})^2 + (\frac{\partial z}{\partial y})^2 + 1} \\
N_x &= -\frac{\partial z}{\partial x} N_z \\
N_y &= -\frac{\partial z}{\partial y} N_z.
\end{aligned}
$$

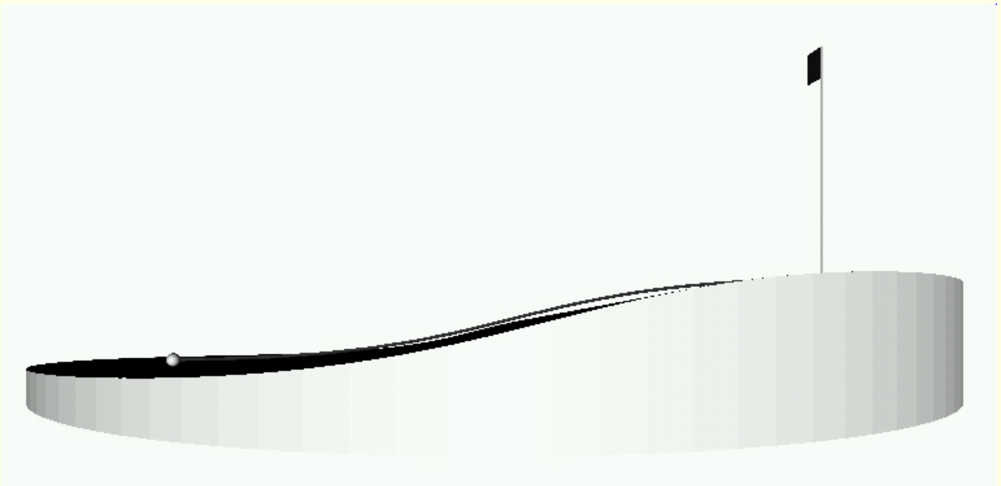
- F is the force due to friction:

$$F = -\mu \|N\| \frac{v}{\|v\|}.$$

Putting—Specific Example

- Discretize continuous time into $n = 200$ discrete time points.
- Use finite differences to approximate the derivatives.





constraints	597
variables	399
time (secs)	
LOQO	14.1
LANCELOT	> 600.0
SNOPT	4.1

23 Goddard Rocket Problem

Objective:

$$\text{maximize } h(T);$$

Constraints:

$$\begin{aligned}
v &= \dot{h} \\
a &= \dot{v} \\
\theta &= -c\dot{m} \\
ma &= (\theta - \sigma v^2 e^{-h/h_0}) - gm \\
0 &\le \theta \le \theta_{\max} \\
m &\ge m_{\min} \\
h(0) &= 0 \quad v(0) = 0 \quad m(0) = 3
\end{aligned}$$

where

- $\theta =$ Thrust , $m =$ mass
- $\theta_{\max}$, g, σ, c, and h_0 are given constants
- h, v, a, T_h, and m are functions of time $0 \le t \le T$.

23 Goddard Rocket Problem

Objective:

$$\text{maximize } h(T);$$

Constraints:

$$\begin{aligned}
v &= \dot{h} \\
a &= \dot{v} \\
\theta &= -c\dot{m} \\
ma &= (\theta - \sigma v^2 e^{-h/h_0}) - gm \\
0 &\le \theta \le \theta_{\max} \\
m &\ge m_{\min} \\
h(0) &= 0 \qquad v(0) = 0 \qquad m(0) = 3
\end{aligned}$$

where

- $\theta =$ Thrust , $m =$ mass
- $\theta_{\max}$, g, σ, c, and h_0 are given constants
- h, v, a, T_h, and m are functions of time $0 \le t \le T$.

Goddard Rocket Problem—Solution

constraints	**399**
variables	**599**
time (secs)	
LOQO	**5.2**
LANCELOT	(IL)
SNOPT	(IL)