

Copyright ©1991-2009 by K. Pattipati

- □ Motivation for Successive Quadratic Programming (SQP) Methods
- **Example 3 Key SQP Ideas**
- **Newton Version of SQP**
- Descent Property of Merit Function *f+cP*
- Quasi-Newton Version of SQP
- \Box SQP with second order correction

Motivation for SQP - 1

- \Box Consider unconstrained minimization problem: $\min_{x} f(x)$
	- Given the current estimate x_k the next estimate x_{k+1} is obtained via

Given the current estimate
$$
\underline{x}_k
$$
 the next estimate \underline{x}_{k+1} is obtained via
a quadratic approximation of $f(\underline{x}^*)$ around \underline{x}_k :

$$
f(\underline{x}^*) \approx f(\underline{x}_k) + \nabla \underline{f}^T(\underline{x}_k)(\underline{x}^* - \underline{x}_k) + \frac{1}{2}(\underline{x}^* - \underline{x}_k)^T \nabla^2 f(\underline{x}_k)(\underline{x}^* - \underline{x}_k)
$$
min at $\underline{x}_{k+1} \equiv \underline{x}^*$
$$
\underline{x}_{k+1} = \underline{x}_k - [\nabla^2 f(\underline{x}_k)]^{-1} \nabla \underline{f}(\underline{x}_k)
$$
 "PURE NEWTON TIERATION"

* An alternate viewpoint is to consider solving the first order necessary condition : $k+1 = 1$

in alternal state
 $\frac{f(x^*)}{f(x^*)}$ $\nabla f(\underline{x}^*) = 0$

order necessary conditions around \underline{x}_k

- Also, know that Newton's method is locally convergent and that we need to modify it via step size selection or trust region approach and employ strategies for indefinite Hessian (e.g., modified Cholesky, Levenberg-Marquardt, double dog-leg, trust region)
- Quasi-Newton methods to avoid having to compute the Hessian $(\Rightarrow$ secant approximation)

SQP for Constrained Optimization

Can we extend this idea to constrained minimization problems: Yes!

- Consider min $f(\underline{x})$ such that $\underline{h}(\underline{x}) = \underline{0}$. Lagrangian function is given as: $L(\underline{x}, \underline{\lambda}) = f(\underline{x}) + \underline{\lambda}^T \underline{h}(\underline{x})$. First order necessary conditions of optimality: First order necessary conditions of

)= $\nabla f(\underline{x}^) + \nabla \underline{h}(\underline{x}^*) \underline{\lambda} = \nabla f(\underline{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla \underline{h}_i(\underline{x}^*)$ 1 $(\mathcal{A}^*) = \nabla f(\mathbf{x}^*) + \nabla \underline{h}(\mathbf{x}^*) \underline{\lambda} = \nabla f(\mathbf{x}^*) + \sum_{i=1}^m \lambda_i \nabla \underline{h}_i$
, $\mathcal{A}^ = \underline{h}(\mathbf{x}^*) = 0$; $(m+n)$ unknowns: $\mathbf{x}^*, \mathcal{A}^*$ er min $f(\underline{x})$ such that $\underline{h}(\underline{x}) = \underline{0}$. Lagrangian function is gi
= $f(\underline{x}) + \underline{\lambda}^T \underline{h}(\underline{x})$. First order necessary conditions of opt
equations: $\nabla_x L(\underline{x}^*, \underline{\lambda}^*) = \nabla f(\underline{x}^*) + \nabla \underline{h}(\underline{x}^*) \underline{\lambda} = \nabla f(\underline{x}^*) + \sum_{i=1$ = $f(\underline{x}) + \underline{\lambda}^t \underline{h}(\underline{x})$. First order necessary condition
equations: $\nabla_x L(\underline{x}^*, \underline{\lambda}^*) = \nabla f(\underline{x}^*) + \nabla \underline{h}(\underline{x}^*) \underline{\lambda} = \nabla f(\underline{x}^*) + \sum_{i=1}^m \lambda_i \nabla$
equations: $\nabla_{\lambda} L(\underline{x}^*, \underline{\lambda}^*) = \underline{h}(\underline{x}^*) = \underline{0}$; $(m+n)$ unknown *m* α *i* α *j* = γ β α *j* + γ β α *j* α = γ β α *j* + γ α *j* $\$ der min $f(\underline{x})$ such that $\underline{h}(\underline{x}) = \underline{0}$. Lagrangian function
 Δx) = $f(\underline{x}) + \underline{\lambda}^T \underline{h}(\underline{x})$. First order necessary conditions of
 n equations: $\nabla_x L(\underline{x}^*, \underline{\lambda}^*) = \nabla f(\underline{x}^*) + \nabla \underline{h}(\underline{x}^*) \underline{\lambda} = \nabla f(\underline{x}^*) + \sum$ $j = f(\underline{x}) + \underline{\lambda}^t \underline{h}(\underline{x})$. First order necessary condition
 m equations: $\nabla_x L(\underline{x}^*, \underline{\lambda}^*) = \nabla f(\underline{x}^*) + \nabla \underline{h}(\underline{x}^*) \underline{\lambda} = \nabla f(\underline{x}^*) + \sum_{i=1}^{m} \lambda_i$
 m equations: $\nabla_{\lambda} L(\underline{x}^*, \underline{\lambda}^*) = \underline{h}(\underline{x}^*) = \underline{0}$; $(m+n)$ h that <u> $h(x) = 0$ </u>. Lagrangian function is giv

First order necessary conditions of optin
 $\underline{\lambda}^*$)= $\nabla \underline{f}(\underline{x}^*) + \nabla \underline{h}(\underline{x}^*) \underline{\lambda} = \nabla \underline{f}(\underline{x}^*) + \sum_{i=1}^m \lambda_i \nabla \underline{h}_i(\underline{x}^*) = 0$ First order necessary conditions of optima
 $(\mathbf{x}^*) = \nabla f(\mathbf{x}^*) + \nabla \underline{h}(\mathbf{x}^*) \underline{\lambda} = \nabla f(\mathbf{x}^*) + \sum_{i=1}^m \lambda_i \nabla h_i(\mathbf{x}^*) = 0$
 $(\mathbf{x}^*) = \underline{h}(\mathbf{x}^*) = 0;$ (*m*+*n*) unknowns: $\mathbf{x}^*, \mathbf{x}^*$ \ast x) such that $\underline{h}(\underline{x}) = \underline{0}$. Lagrangian function is given as:
 $\frac{d}{dx} \underline{h}(\underline{x})$. First order necessary conditions of optimality:
 $\nabla_x L(\underline{x}^*, \underline{\lambda}^*) = \nabla f(\underline{x}^*) + \nabla \underline{h}(\underline{x}^*) \underline{\lambda} = \nabla f(\underline{x}^*) + \sum_{i=1}^m \lambda_i \nabla \underline{h$ $\underline{h}(\underline{x})$. First order necessary condi
 $\nabla_x L(\underline{x}^*, \underline{\lambda}^*) = \nabla f(\underline{x}^*) + \nabla \underline{h}(\underline{x}^*) \underline{\lambda} = \nabla f(\underline{x}^*) + \sum_{i=1}^{n} \nabla_{\underline{\lambda}} L(\underline{x}^*, \underline{\lambda}^*) = \underline{h}(\underline{x}^*) = \underline{0};$ (*m*+*n*) unknowns $\sum_{i=1}^{n}$
- \mathbf{u} \mathbf{c}
 \mathbf{q}
 λ_{k+1} current estimates, want to
 $\sum_{k=1}^{2} L(x_k, \lambda_k)(x_{k+1} - x_k) + \nabla_x^2$ If $(\underline{x}_k, \underline{\lambda}_k)$, the current estimates, want to find hew estimate
 ${}_{x}L(\underline{x}_k, \underline{\lambda}_k) + \nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k) (\underline{x}_{k+1} - \underline{x}_k) + \nabla_{x\lambda}^2 L(\underline{x}_k, \underline{\lambda}_k) (\underline{\lambda}_{k+1} - \underline{\lambda}_k)$ 2 1 *m* equations: $V_{\lambda}L(\underline{x}, \underline{\lambda}) = \underline{n}(\underline{x}) = \underline{0}$; $(m+n)$ unknowns: $\underline{x}, \underline{\lambda}$
call Newton's method for solving a system of non-linear equations
Given $(\underline{x}_k, \underline{\lambda}_k)$, the current estimates, want to find new estimates $(\underline{x$ call Newton's method for solving a system of non-linear equation
Given $(\underline{x}_k, \underline{\lambda}_k)$, the current estimates, want to find new estimates $(\underline{x}_{k+1}, \underline{\lambda}_{k+1})$
 $\Rightarrow \nabla_x L(\underline{x}_k, \underline{\lambda}_k) + \nabla_x^2 L(\underline{x}_k, \underline{\lambda}_k)(\underline{x}_{k+1} - \underline{x}_k) + \nab$ Given $(\underline{x}_k, \underline{\lambda}_k)$, the current estimates, want t
 $\Rightarrow \nabla_x L(\underline{x}_k, \underline{\lambda}_k) + \nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k)(\underline{x}_{k+1} - \underline{x}_k) + \nabla_{x}^2$
 $\nabla_x L(\underline{x}_k, \underline{\lambda}_k) + \nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k)(\underline{x}_{k+1} - \underline{x}_k) = 0$ Given $(\underline{x}_k, \underline{\lambda}_k)$, the current estimates, v
 $\Rightarrow \nabla_x L(\underline{x}_k, \underline{\lambda}_k) + \nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k)(\underline{x}_{k+1} - \underline{x}_k)$
 $\nabla_{\lambda} L(\underline{x}_k, \underline{\lambda}_k) + \nabla_{\lambda x}^2 L(\underline{x}_k, \underline{\lambda}_k)(\underline{x}_{k+1} - \underline{x}_k)$

Using: $\nabla_x L(\underline{x}_k, \underline{\lambda}_k) = \nabla f(\underline{x}_k) + \nab$ $\mathcal{X}_k, \underline{\mathcal{A}}_k$ + $\mathcal{V}_{\lambda x}^{\perp} L(\underline{x}_k, \underline{\mathcal{A}}_k)(\underline{x}_{k+1} - \underline{x}_{k+1})$
 $\mathcal{X}_x L(\underline{x}_k, \underline{\mathcal{A}}_k) = \nabla \underline{f}(\underline{x}_k) + \nabla \underline{h}(\underline{x}_k)$ π (\underline{x}_k , $\underline{\lambda}_k$), the current estimates, w.
 π $L(\underline{x}_k, \underline{\lambda}_k) + \nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k)(\underline{x}_{k+1} - \underline{x}_k)$
 π $L(\underline{x}_k, \underline{\lambda}_k) + \nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k)(\underline{x}_{k+1} - \underline{x}_k)$
 π : $\nabla_x L(\underline{x}_k, \underline{\lambda}_k) = \nabla f(\underline{x}_k) + \nabla \underline$ λ λ , ven $(\underline{x}_k, \underline{\lambda}_k)$, the current estimates, want to find new estimates
 $\nabla_x L(\underline{x}_k, \underline{\lambda}_k) + \nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k) (\underline{x}_{k+1} - \underline{x}_k) + \nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k) (\underline{\lambda}_{k+1} - \underline{\lambda}_k)$
 $\nabla_x L(\underline{x}_k, \underline{\lambda}_k) + \nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k) (\underline{x}_{$ $^{+}$ $(\underline{x}_k, \underline{\lambda}_k) + \nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k)(\underline{x}_{k+1} - \underline{x}_k) + \nabla_{x}^2$
 $(\underline{x}_k, \underline{\lambda}_k) + \nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k)(\underline{x}_{k+1} - \underline{x}_k) = 0$
 $\nabla_x L(\underline{x}_k, \underline{\lambda}_k) = \nabla_{\underline{f}}(\underline{x}_k) + \nabla_{\underline{h}}(\underline{x}_k)\underline{\lambda}_k, \nabla_{x}$ sing: $\nabla_x L(\underline{x}_k, \underline{\lambda}_k) = \nabla f(\underline{x}_k) + \nabla \underline{h}(\underline{x}_k)$
 $\frac{\partial}{\partial x} L(\underline{x}_k, \underline{\lambda}_k) = \nabla^2 f(\underline{x}_k) + \sum_{i=1}^m (\lambda_i)_{i} \nabla^2$ $\begin{aligned} &\frac{2}{\chi_{xx}}L(\underline{x}_k,\underline{\lambda}_k)=\nabla^2 f(\underline{x}_k)+\sum_{x\in\mathcal{L}}L(\underline{x}_k,\underline{\lambda}_k)=\nabla h(\underline{x}_k)=\nabla \overline{\overline{Y}}_k^2, \end{aligned}$ 2 1 1 $\begin{aligned} &\n\chi_k) + \nabla^2_{x\lambda} L(\underline{x}_k, \underline{\lambda}_k)(\underline{\lambda}_{k+1} - \underline{\lambda}_{k}) \ &\quad \chi_k) = \underline{0} \ &\quad \chi_k, \nabla_{\lambda} L(\underline{x}_k, \underline{\lambda}_k) = \underline{h}(\underline{x}_k) \ &\quad \chi_k, \end{aligned}$ $\nabla_{\lambda} L(\underline{x}_k, \underline{\lambda}_k) + \nabla_{\lambda x}^2 L(\underline{x}_k, \underline{\lambda}_k) (\underline{x}_{k+1} - \underline{x}_k) = 0$
 $\mathbf{g}: \nabla_{x} L(\underline{x}_k, \underline{\lambda}_k) = \nabla \underline{f} (\underline{x}_k) + \nabla \underline{h} (\underline{x}_k) \underline{\lambda}_k, \nabla_{\lambda x}$
 $(\underline{x}_k, \underline{\lambda}_k) = \nabla^2 f (\underline{x}_k) + \sum_{i=1}^m (\lambda_i)_i \nabla^2 h_i (\underline{x}_k)$ g: $\nabla_x L(\underline{x}_k, \underline{\lambda}_k) = \nabla_{\underline{f}}^T(\underline{x}_k) + \nabla_{\underline{f}}^T(\underline{x}_k)$
 $(\underline{x}_k, \underline{\lambda}_k) = \nabla^2 f(\underline{x}_k) + \sum_{i=1}^m (\lambda_i)_i \nabla^2 h$
 $(\underline{x}_k, \underline{\lambda}_k) = \nabla_{\underline{f}}^T(\underline{x}_k) = [\nabla_{\lambda x}^2 L(\underline{x}_k, \underline{\lambda}_k)]$ $\sum_{k} (X_k) - \nabla \cdot \frac{1}{2} \sum_{k=1}^N (X_k) - \sum_{i=1}^N (X_k) - \sum_{k}^N (X_k) - \sum_{k}^N$
 $(\underline{x}_k, \underline{\lambda}_k) - \nabla \underline{h}(\underline{x}_k) - \sum_{k=1}^N (\underline{x}_k, \underline{\lambda}_k) - \sum_{k=1}^N (\underline{x}_k) - \sum_{k=1}^N (\underline{x}_$ $(\sum_k \lambda_k, \Delta_k) = \mathbf{V} \underline{h}(\underline{x}_k) = [\mathbf{V}_{\lambda x} L(\underline{x}_k, \underline{x}_k)] = N_k$
 $(\sum_k \lambda_k, \underline{\lambda}_k) \quad \nabla \underline{h}(\underline{x}_k) = \begin{bmatrix} \Delta_{k+1} - \underline{x}_k \\ \Delta_{k+1} - \underline{\lambda}_k \end{bmatrix} = \begin{bmatrix} -\nabla \underline{f}(\underline{x}_k) - \nabla \underline{h}(\underline{x}_k) \underline{\lambda}_k \\ -\underline{h}(\underline{x}_k) \end{bmatrix}$ *m* $\nabla_{\lambda} L(\underline{x}_k, \underline{\lambda}_k) + \nabla_{\lambda x}^2 L(\underline{x}_k, \underline{\lambda}_k)(\underline{x}_{k+1} - \underline{x}_k) = 0$

sing: $\nabla_{x} L(\underline{x}_k, \underline{\lambda}_k) = \nabla_{\underline{f}} (\underline{x}_k) + \nabla_{\underline{h}} (\underline{x}_k) \underline{\lambda}_k, \nabla_{\lambda} L(\underline{x}_k)$
 $\frac{2}{\lambda x} L(\underline{x}_k, \underline{\lambda}_k) = \nabla^2 f(\underline{x}_k) + \sum_{i=1}^m (\lambda_i)_i \nabla^2 h_i(\underline{x}_k$ *T* $\chi_x L(\underline{x}_k, \underline{x}_k) = \mathbf{v} \int (\underline{x}_k) + \sum_{i=1}^k (\lambda_k) i \mathbf{v} \cdot n_i (\underline{x}_k)$
 $\chi_x L(\underline{x}_k, \underline{\lambda}_k) = \nabla \underline{h}(\underline{x}_k) = [\nabla^2_{\lambda x} L(\underline{x}_k, \underline{\lambda}_k)]^T = N_k$ $\left(\frac{\lambda_k}{k}\right)$ $V_{\frac{n}{2}(X_k)}$ $\left(\frac{x_{k+1} - x_k}{\lambda_{k+1} - \lambda_k}\right) = \begin{bmatrix} -V_{\frac{1}{2}(X_k)} - V_{\frac{n}{2}(X_k)} \ -h_{\frac{n}{2}(X_k)} \end{bmatrix}$ ng: $\nabla_x L(\underline{x}_k, \underline{\lambda}_k) = \nabla_{\underline{f}}^T(\underline{x}_k) + \nabla_{\underline{h}}^T(\underline{x}_k) \underline{\lambda}_k, \nabla_{\lambda}$
 $L(\underline{x}_k, \underline{\lambda}_k) = \nabla^2 f(\underline{x}_k) + \sum_{i=1}^m (\lambda_i)_i \nabla^2 h_i(\underline{x}_k)$
 $L(\underline{x}_k, \underline{\lambda}_k) = \nabla_{\underline{h}}^T(\underline{x}_k) = [\nabla_{\lambda x}^2 L(\underline{x}_k, \underline{\lambda}_k)]^T = N$ $\sum_{k} \sum_{k} y - \mathbf{v} \int (\underline{x}_{k})^{T} \sum_{i=1}^{L} (X_{k})^{T} \mathbf{v} R_{i} (\underline{x}_{k}) - \mathbf{H}_{k}$
 $\sum_{k} \sum_{k} \sum_{k} = \nabla \underline{h} (\underline{x}_{k}) = [\nabla^{2}_{\lambda x} L(\underline{x}_{k}, \underline{\lambda}_{k})]^{T} = N_{k}$
 $L(\underline{x}_{k}, \underline{\lambda}_{k}) \nabla \underline{h} (\underline{x}_{k}) \bigg| \bigg[\underline{x}_{k+1} - \underline{x}_{k} \bigg] = \bigg[-\nabla \underline{f} (\underline{x$ $\begin{aligned} \n\mathcal{L}_{k} & \mathcal{L}_{k} & \mathcal{$ λ) + $\nabla_{x\lambda}^2 L(\underline{x}_k, \underline{\lambda}_k)(\underline{\lambda}_{k+1} - \underline{\lambda}_k) = 0$
 $\underline{\lambda}_k$, $\nabla_{\lambda} L(\underline{x}_k, \underline{\lambda}_k) = \underline{h}(\underline{x}_k)$ $(\underline{x}_k, \underline{\lambda}_k) + \nabla^2_{\lambda x} L(\underline{x}_k, \underline{\lambda}_k) (\underline{x}_{k+1} - \underline{x}_k) = 0$
 $(\underline{x}_k, \underline{\lambda}_k) = \nabla \underline{f} (\underline{x}_k) + \nabla \underline{h} (\underline{x}_k) \underline{\lambda}_k, \nabla_{\lambda} \underline{\lambda}_k)$
 $(\underline{\lambda}_k) = \nabla^2 f (\underline{x}_k) + \sum_{i=1}^m (\lambda_i)_{i} \nabla^2 h_i (\underline{x}_k)$ $\chi_L(\underline{x}_k, \underline{\lambda}_k) = \nabla \underline{f}(\underline{x}_k) + \nabla \underline{h}(\underline{x}_k) \underline{\lambda}_k, \nabla_{\lambda} L(\underline{x}_k)$
 $\underline{\lambda}_k$) = $\nabla^2 f(\underline{x}_k) + \sum_{i=1}^m (\lambda_i)_i \nabla^2 h_i(\underline{x}_k) = H$
 $\underline{\lambda}_k$) = $\nabla \underline{h}(\underline{x}_k) = [\nabla^2_{\lambda x} L(\underline{x}_k, \underline{\lambda}_k)]^T = N_k$ λ_{α}) $\nabla h(x_{\alpha})$ $\left[\begin{array}{cc} x_{\alpha} - x_{\alpha} \end{array} \right]$ $\left[-\nabla f(x_{\alpha}) - \nabla h(x_{\alpha}) \lambda_{\alpha} \right]$ $[L(\underline{x}_k, \underline{\lambda}_k)]^{\dagger} = N_k$
 $\underline{x}_{k+1} - \underline{x}_k$
 $\underline{\lambda}_{k+1} - \underline{\lambda}_k$ = $\begin{bmatrix} -\nabla f(x) \\ -\nabla f(x) \end{bmatrix}$ $\nabla_{\lambda} L(\underline{x}_k, \underline{\lambda}_k) + \nabla_{\lambda x}^2 L(\underline{x}_k, \underline{\lambda}_k)(\underline{x}_{k+1} - \underline{x}_k) = 0$
Using: $\nabla_{x} L(\underline{x}_k, \underline{\lambda}_k) = \nabla f(\underline{x}_k) + \nabla \underline{h}(\underline{x}_k) \underline{\lambda}_k, \nabla_{\lambda} L(\underline{x}_k, \underline{\lambda}_k) = \underline{h}(\nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k) = \nabla^2 f(\underline{x}_k) + \sum_{i=1}^m (\lambda_i)_{i} \nabla^2 h_i$ $^{+}$ $^{+}$ Using: $\nabla_x L(\underline{x}_k, \underline{\lambda}_k) = \nabla f(\underline{x}_k) + \nabla \underline{h}(\underline{x}_k) \underline{\lambda}_k, \nabla_{\lambda} L(\underline{x}_k, \underline{\lambda}_k)$
 $\nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k) = \nabla^2 f(\underline{x}_k) + \sum_{i=1}^m (\lambda_i)_i \nabla^2 h_i(\underline{x}_k) = H_k$
 $\nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k) = \nabla \underline{h}(\underline{x}_k) = [\nabla_{\lambda x}^2 L(\underline{x}_k, \underline{\lambda}_$ $L(\underline{x}_k, \underline{\lambda}_k) = \mathbf{V} \int (\underline{x}_k) + \sum_{i=1}^k (\lambda_k)_i \mathbf{V} h_i(\underline{x}_k) = \mathbf{H}_k$
 $L(\underline{x}_k, \underline{\lambda}_k) = \nabla \underline{h}(\underline{x}_k) = [\nabla^2_{\lambda x} L(\underline{x}_k, \underline{\lambda}_k)]^T = N_k$
 $\begin{bmatrix} \nabla^2_{xx} L(\underline{x}_k, \underline{\lambda}_k) & \nabla \underline{h}(\underline{x}_k) \end{bmatrix} \begin{bmatrix} \underline{x}_{k+1} - \underline{x}_k \\ \underline{x}_{k+1} - \underline{x}_$ $\nabla_{x\lambda}^2 L(\underline{x}_k, \underline{\lambda}_k) = \nabla_{\underline{h}}^1(\underline{x}_k) = [\nabla_{\lambda x}^2 L(\underline{x}_k, \underline{\lambda}_k)]^T = N_k$
 $\Rightarrow \begin{bmatrix} \nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k) & \nabla_{\underline{h}}^1(\underline{x}_k) \\ \nabla_{\underline{h}}^T(\underline{x}_k) & 0 \end{bmatrix} \begin{bmatrix} \underline{x}_{k+1} - \underline{x}_k \\ \underline{\lambda}_{k+1} - \underline{\lambda}_k \end{bmatrix} = \begin{bmatrix} -\nabla_{\underline{f}}^1(\$ \sum • Recall Newton's method for solving a system of non-linear equations: add -1^I to (2,2) block if N_k is not full rank *k I c* \overline{a}

Solution of Linearized Equations = QPP

• Let $\overline{x_{k+1} - x_k} = \underline{d}_k$ and add $\nabla_{\underline{p}(\underline{x}_k)} \underline{\lambda}_k$ to first equation: 2 1 $\underline{x}_k = \underline{d}_k$ and add $\nabla_{\underline{h}} (\underline{x}_k) \underline{\lambda}_k$ to first equal
 $(\underline{x}_k, \underline{\lambda}_k) \quad \nabla_{\underline{h}} (\underline{x}_k)$ $\begin{bmatrix} \underline{d}_k \end{bmatrix} = \begin{bmatrix} -\nabla_{\underline{f}} (\underline{x}_k) \end{bmatrix}$ $(*)$ $\left[\nabla_{\underline{k}}^2 L(\underline{x}_k, \underline{\lambda}_k) \quad \nabla_{\underline{m}}^2(\underline{x}_k) \right] = \begin{bmatrix} \underline{d}_k \\ \underline{\lambda}_{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla_{\underline{f}}(\underline{x}_k) \\ -\underline{h}(\underline{x}_k) \end{bmatrix}$ *T* k *k* k **k** k $-\underline{x}_k = \underline{d}_k$ and add $\nabla_{\underline{R}}(\underline{x}_k) \underline{\lambda}_k$ to first eq
 $L(\underline{x}_k, \underline{\lambda}_k)$ $\nabla_{\underline{R}}(\underline{x}_k)$ $\begin{bmatrix} \underline{d}_k \end{bmatrix} = \begin{bmatrix} -\nabla_{\underline{f}}(\underline{x}_k) \end{bmatrix}$ $\left[\begin{array}{cc} L(\underline{x}_k, \underline{\lambda}_k) & \nabla \underline{h}(\underline{x}_k) \\ \frac{h^T}{2}(\underline{x}_k) & 0 \end{array} \right] = \left[\begin{array}{c} \underline{d}_k \\ \underline{\lambda}_{k+1} \end{array} \right] = \left[\begin{array}{c} -\nabla f(\underline{x}_k) \\ -\underline{h}(\underline{x}_k) \end{array} \right]$ $\underline{\lambda}_{\scriptscriptstyle k+}$ $\overline{x}_{k+1} - \underline{x}_k = \underline{d}_k$ and add $\nabla_{\underline{h}}(\underline{x}_k) \underline{\lambda}_k$ to first equation:
 $\begin{bmatrix} \nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k) & \nabla_{\underline{h}}(\underline{x}_k) \\ \nabla_{\underline{h}}^T(\underline{x}_k) & 0 \end{bmatrix} = \begin{bmatrix} \underline{d}_k \\ \underline{\lambda}_{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla_{\underline{f}}(\underline{x}_k) \\ -\underline{h}(\underline{x}_k) \end{$

Claim: These are the necessary conditions of optimality for the

following quadratic programming problem:
\n
$$
\min_{\underline{d}_k} \frac{1}{2} \underline{d}_k^T \nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k) \underline{d}_k + \nabla \underline{f}^T(\underline{x}_k) \underline{d}_k
$$
\n(QPP)
\ns.t.
$$
\nabla \underline{h}^T(\underline{x}_k) \underline{d}_k + \underline{h}(\underline{x}_k) = 0
$$

First order necessary conditions of optimality:

 $1 \frac{1}{d^T \nabla^2}$ * $\lambda^{k} = \lambda^{k} - \lambda^{k}$ 2 1 irst order necessary conditions of optimality:
Define: $L(\underline{d}, \underline{y}) = \frac{1}{2} \underline{d}^T \nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k) \underline{d} + \nabla \underline{f}^T(\underline{x}_k) \underline{d}_k + \underline{y}^T [\nabla \underline{h}^T(\underline{x}_k) \underline{d}_k + \underline{h}(\underline{x}_k)]$ Define: $L(\underline{d}, \underline{y}) = \frac{1}{2} \underline{d}^T \nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k) \underline{d} + \nabla_{\underline{x}}$
Optimality Conditions of $(\underline{d}_k, \underline{y}^* = \underline{\lambda}_{k+1})$ mality Conditions of $(\underline{d}_k, \underline{y}^* = \underline{\lambda}_{k+1})$
 $(\underline{x}_k, \underline{\lambda}_k) \underline{d}_k + \nabla \underline{h}(\underline{x}_k) \underline{\lambda}_{k+1} = -\nabla \underline{f}(\underline{x}_k)$ same as punianty Conditions of $(\underline{a}_k, \underline{r}) = \underline{a}_{k+1}$
 $\frac{d}{dx} L(\underline{x}_k, \underline{\lambda}_k) \underline{d}_k + \nabla \underline{h}(\underline{x}_k) \underline{\lambda}_{k+1} = -\nabla \underline{f}(\underline{x}_k)$
 $\nabla \underline{h}^T(\underline{x}_k) \underline{d}_k = -\underline{h}(\underline{x}_k)$ *T* ∇ conditions of optimality:
 $\begin{aligned} \nT \nabla^2_{xx} L(x_t, \lambda_t) d + \nabla f^T(x_t) d_t + \gamma^T [\nabla h^T] \n\end{aligned}$ Let necessary conditions of optimality:
 $L(\underline{d}, \underline{r}) = \frac{1}{2} \underline{d}^T \nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k) \underline{d} + \nabla \underline{f}^T(\underline{x}_k) \underline{d}_k + \underline{\gamma}^T [\nabla \underline{h}^T(\underline{x}_k) \underline{d}_k + \underline{h}(\underline{x}_k)]$ *T* $\sum_{k=1}^{n} (x_k - \mu_k) d_k = -h(x_k)$ \underline{d}_{k} , $\underline{\gamma}^{*}=\underline{\lambda}_{k+1}$ $2^{\frac{1}{2}}$ *x* $\frac{1}{2^k}$ *x* $\frac{1}{2^k}$ *x* $\frac{1}{2^k}$
 L(\underline{x}_k , $\underline{\lambda}_k$) \underline{d}_k + ∇ <u>*h*</u>(\underline{x}_k) $\underline{\lambda}_{k+1}$ = $-\nabla f$ (\underline{x}_k) $\underline{h}^T(\underline{x}_k) \underline{\lambda}_{k+1} = -\nabla \underline{f}(\underline{x}_k) \underline{d}_k = -\underline{h}(\underline{x}_k)$ $^{+}$ Defined \underline{a} , \underline{b} , \underline{b} and \underline{a} , \underline{b} , \underline{b} and \underline{b}

Optimality Conditions of $(\underline{d}_k, \underline{y}^* = \underline{\lambda}_{k+1})$
 $\nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k) \underline{d}_k + \nabla \underline{h}(\underline{x}_k) \underline{\lambda}_{k+1} = -\nabla \underline{f}(\underline{x}_k)$ $\}$ same as $*$ tions of $(\underline{a}_k, \underline{y} = \underline{A}_{k+1})$
 $\nabla \underline{h}(\underline{x}_k) \underline{\lambda}_{k+1} = -\nabla \underline{f}(\underline{x}_k)$
 $\nabla \underline{h}^T(\underline{x}_k) \underline{d}_k = -\underline{h}(\underline{x}_k)$ sar γ ions of optimality:
 $\underline{\lambda}_k \underline{d} + \nabla \underline{f}^T(\underline{x}_k) \underline{d}_k + \underline{\gamma}^T [\nabla \underline{h}^T(\underline{x}_k)]$ ty Conditions of $(\underline{d}_k, \underline{y}^* = \underline{\lambda}_k)$
 $(\underline{x}_k) \underline{d}_k + \nabla \underline{h}(\underline{x}_k) \underline{\lambda}_{k+1} = -\nabla \underline{f}(\underline{x}_k)$

6

Summary of SQP Ideas - 1

- Let us summarize results so far and list unresolved issues:
- 1) Can obtain \underline{d}_k and the multiplier vector \underline{d}_{k+1} form the solution of a quadratic programming problem with linear equality constraints.
- 2) In essence, we are approximating the nonlinear equality problem by a series of quadratic programming problems, one at each iteration.
- 3) Again, can get only local convergence. Need strategies for: a) Indefinite $\nabla_{xx}^2 L \Rightarrow$ Modified Cholesky, Quasi-Newton, Augmented Lagrangian

b)Global convergence – Line search. Q: Line Search on What?

- 4) What about inequality constraints?
	- One way of ensuring positive definiteness of $\nabla_{xx}^2 L$ is to convexify the Lagrangian by adding a quadratic penalty term:
 $L_c(\underline{x}, \underline{\lambda}) = f(\underline{x}) + \underline{\lambda}^T \underline{h}(\underline{x}) + \frac{1}{2} c \underline{h}^T(\underline{x}) \underline{h}(\underline{x})$ Lagrangian by adding a quadratic penalty term: One way of ensuring positive defin

	agrangian by adding a quadratic $\frac{d(x, \lambda)}{dt} = f(\frac{x}{\lambda}) + \frac{\lambda^T h(\frac{x}{\lambda}) + \frac{1}{2} c h(\frac{x}{\lambda}) h(\frac{x}{\lambda})}$ dding a qua
 $\frac{1}{r}h(x) + \frac{1}{r}ch^T$

$$
L_c(\underline{x}, \underline{\lambda}) = f(\underline{x}) + \underline{\lambda}^T \underline{h}(\underline{x}) + \frac{1}{2} c \underline{h}^T(\underline{x}) \underline{h}(\underline{x})
$$

$$
L_c(\underline{x}, \underline{\lambda}) = f(\underline{x}) + \underline{\lambda}^T \underline{h}(\underline{x}) + \frac{1}{2} c \underline{h}^T(\underline{x}) \underline{h}(\underline{x})
$$

Use $\nabla_{xx}^2 L_c(\underline{x}_k, \underline{\lambda}_k) = \nabla_{xx}^2 L_o(\underline{x}_k, \underline{\lambda}_k) + \sum_{i=1}^m c_i h_i(\underline{x}_k) \nabla^2 h_i(\underline{x}_k) + c_k \nabla_{\underline{h}}^1(\underline{x}_k) \nabla_{\underline{h}}^1(\underline{x}_k)$

Summary of SQP Ideas - 2

Alternatively, use only

ernatively, use only
 ${}_{xx}L_c(\underline{x}_k, \underline{\lambda}_k) \approx \nabla_{xx}^2 L_o(\underline{x}_k, \underline{\lambda}_k) + c_k \nabla_{\underline{u}}^L(\underline{x}_k) \nabla_{\underline{u}}^T(\underline{x}_k),$ *T xx c k k xx o k k k k k L x L x c h x h x*

Iternatively, use only
 $\nabla_{xx}^2 L_c(\underline{x}_k, \underline{\lambda}_k) \approx \nabla_{xx}^2 L_o(\underline{x}_k, \underline{\lambda}_k) + c_k \nabla_{\underline{R}}(\underline{x}_k) \nabla_{\underline{R}}^T(\underline{x}_k),$

RHS will be modified as: $-\nabla_{\underline{f}}(\underline{x}_k) \rightarrow -[\nabla_{\underline{f}}(\underline{x}_k) + c_k \nabla_{\underline{R}}(\underline{x}_k) \underline{h}(\underline{x}_k)]$

xtonsion to i

• Extension to inequality constraints:

Extension to inequality constraints:

min $f(x)$, s.t. $\underline{h}(x) = 0$; $\underline{h} \in R^m$; $\underline{g}(x) \le 0$; $\underline{g} \in R^r$

Lagrangian Function: $L(x, \lambda, \mu) = f(x) + \lambda^T \underline{h}(x) + \mu^T \underline{g}(x)$

$$
\frac{\text{Necessary Conditions:}}{\nabla \underline{f}(\underline{x}^*) + \nabla \underline{h}(\underline{x}^*) \underline{\lambda}^* + \nabla \underline{g}(\underline{x}^*) \underline{\mu}^* = 0}
$$

$$
\begin{aligned} \n\mathbf{v} \underline{f}(\underline{x}) + \mathbf{v} \underline{n}(\underline{x}) \underline{\lambda} + \mathbf{v} \underline{g}(\underline{x}) \underline{\mu} &= \underline{0} \\ \n\frac{h(x^*) = 0}{\mu_i^* g_i(\underline{x}^*)} &= 0 \qquad i = 1, 2, \dots, r \quad (or) \underline{g}(\underline{x}^*) \leq \underline{0} \n\end{aligned}
$$

Linearization leads to: $\underline{\mu}\geq\underline{0}$

$$
\mu_i^* g_i(\underline{x}^*) = 0 \qquad i = 1, 2, ..., r \text{ (or) } \underline{g}(\underline{x}^*) \leq \underline{0}
$$
\n
$$
\text{Linearization leads to:}
$$
\n
$$
\begin{bmatrix}\n\nabla^2_{xx} L(\underline{x}_k, \underline{\lambda}_k, \underline{\mu}_k) & \nabla \underline{h}(\underline{x}_k) & \nabla \underline{g}(\underline{x}_k) \\
\nabla \underline{h}^T(\underline{x}_k) & 0 & 0 \\
\nabla \underline{g}^T(\underline{x}_k) & 0 & 0\n\end{bmatrix}\n\begin{bmatrix}\n\underline{d}_k \\
\underline{\lambda}_{k+1} \\
\underline{\mu}_{k+1}\n\end{bmatrix} = \begin{bmatrix}\n-\nabla \underline{f}(\underline{x}_k) \\
-\underline{h}(\underline{x}_k) \\
-\underline{g}(\underline{x}_k)\n\end{bmatrix}
$$
\nwhere $\nabla^2_{xx} L(\underline{x}_k, \underline{\lambda}_k, \underline{\mu}_k) = \nabla^2 f(\underline{x}_k) + \sum_{i=1}^m (\lambda_i)_i \nabla^2 h_i(\underline{x}_k) + \sum_{j=1}^r (\mu_k)_j \nabla^2 g_j(\underline{x}_k)$ \n
$$
\text{where } \nabla^2_{xx} L(\underline{x}_k, \underline{\lambda}_k, \underline{\mu}_k) = \nabla^2 f(\underline{x}_k) + \sum_{i=1}^m (\lambda_i)_i \nabla^2 h_i(\underline{x}_k) + \sum_{j=1}^r (\mu_k)_j \nabla^2 g_j(\underline{x}_k)
$$

8

Summary of SQP Ideas - 3

This is equivalent to the following QPP with linear equality and inequality constraints : nts :

min $\underline{d}_{k}^{T} \nabla^{2}{}_{xx} L(\underline{x}_{k}, \underline{\lambda}_{k}, \underline{\mu}_{k}) \underline{d}_{k} + \nabla \underline{f}^{T}(\underline{x}_{k})$ *T T* $\underline{d}_k^T \nabla^2_{xx} L(\underline{x}_k, \underline{\lambda}_k, \underline{\mu}_k) \underline{d}_k + \nabla \underline{f}^T(\underline{x}_k) \underline{d}_k$ $\nabla^2_{xx} L(\underline{x}_k, \underline{\lambda}_k, \underline{\mu}_k) \underline{d}_k + \nabla \underline{f}^T(\underline{x}_k)$

$$
\min_{\underline{d}_k} \underline{d}_k^T \nabla^2_{xx} L(\underline{x}_k, \underline{\lambda}_k, \underline{\mu}_k) \underline{d}_k + \nabla \underline{f}^T(\underline{x}_k) \underline{d}_k
$$
\ns.t.
$$
\nabla \underline{h}^T(\underline{x}_k) \underline{d}_k + \underline{h}(\underline{x}_k) = 0
$$
\n
$$
\nabla \underline{g}^T(\underline{x}_k) \underline{d}_k + \underline{g}(\underline{x}_k) \le 0
$$

Questions:

- How do we use this idea in a general SQP algorithm?
- Need to solve a quadratic programming problem at each iteration. How to solve QPP?
- How to ensure global convergence? Line search on what function? 0 0 Step 1: Given an initial estimate , , comp 0 0 0 0 0 0 ute (, ,), (), (). Set =0 onvergence? Line search on what function?

<u>on Version</u>
 x_0, λ_0, μ_0 compute $\nabla_{xx}^2 L(\underline{x}_0, \underline{\lambda}_0, \mu_0), \nabla h(\underline{x}_0), \nabla g(\underline{x}_0)$. Set $k = 0$
- General Algorithm: Newton Version

2 *xx*

Step 1: Given an initial estimate
$$
\underline{x}_0, \underline{\lambda}_0, \underline{\mu}_0
$$
 compute $\nabla_x^2 L(\underline{x}_0, \underline{\lambda}_0, \underline{\mu}_0), \nabla \underline{h}(\underline{x}_0), \nabla \underline{g}(\underline{x}_0)$. Set $k = 0$
\nStep 2: Solve the QPP
\n
$$
\begin{array}{c}\n\min_{\underline{d}_k} \frac{1}{2} \underline{d}_k^T \nabla_{xx}^2 L(\underline{x}_k, \underline{\lambda}_k, \underline{\mu}_k) \underline{d}_k + \nabla \underline{f}^T(\underline{x}_k) \underline{d}_k \\
\text{s.t. } \nabla \underline{h}^T(\underline{x}_k) \underline{d}_k + \underline{h}(\underline{x}_k) = 0 \\
\hline\n\nabla \underline{g}^T(\underline{x}_k) \underline{d}_k + \underline{g}(\underline{x}_k) \leq 0\n\end{array}
$$
\n
$$
\Rightarrow \text{RESULT: } \underline{d}_k, \underline{\lambda}_{k+1}, \underline{\mu}_{k+1}
$$

General SQP Algorithm

Step 3: Select a step size α_k along \underline{d}_k to minimize a penalty (merit) function
Choices for P: 1. P = max{0, g₁(<u>x</u>), g₂(<u>x</u>),..., g_r(<u>x</u>), | h₁(x)|, | h₂(x)|,..., | h_m(version (continued)

a step size α_k along \underline{d}_k to minimize a penalty (merit) function = f

: 1. P = max{0, g₁(<u>x</u>), g₂(<u>x</u>),..., g_r(<u>x</u>),| h₁(<u>x</u>)|,| h₂(<u>x</u>)|,...,| h_m(<u>x</u>)|} Ch oices for *k* along \underline{d}_k to minimize a penalty (merit) function
 $x\{0, g_1(\underline{x}), g_2(\underline{x}), ..., g_r(\underline{x}), |h_1(\underline{x})|, |h_2(\underline{x})|, ..., |h_m(\underline{x})| + \sum_{i=1}^m |h_i(\underline{x})|$ $\frac{1}{\sqrt{2}}$
(1) General Algorithm: Newton Version (continued)
Step 3: Select a step size α_k along \underline{d}_k to minimize a penalty (1 α_k arong \underline{a}_k to minim

nax {0, g₁(<u>x)</u>, g₂(<u>x)</u>,. **ral SQP Algorithm**
 d_k to minimize a penalty (merit) function = $f + cP$
 $f(x) = a(x) + b(x) + b(x) + b(x) + b(x)$ *P* and *P P* g $P = \max\{0, g_1(\underline{x}), g_2(\underline{x}), ..., g_r(\underline{x}), |h_1(\underline{x})|, |h_2(\underline{x})|, ..., |h_m(\underline{x})|, |h_m(\underline{x})|$ *P* = max {0, $g_1(\underline{x})$, $g_2(\underline{x})$, ..., $g_r(\underline{x})$, $h_1(\underline{P}) = \sum_{j=1}^r \max(0, g_r(\underline{x})) + \sum_{i=1}^m |h_i(\underline{x})|$ $= f + cP$ Newton Version (continued)
 α_k along \underline{d}_k to minimize a penalty (merit)

max {0, g₁(<u>x</u>), g₂(<u>x</u>),..., g_r(<u>x</u>), | h₁(<u>x</u>)|, | h₂(<u>x</u>)
 $\sum_{j=1}^r \max(0, g_r(\underline{x})) + \sum_{i=1}^m |h_i(\underline{x})|$ α

P: 1. *P* = max{0,
$$
g_1(\underline{x})
$$
, $g_2(\underline{x})$,..., $g_r(\underline{x})$, $|h_1|$
\n2. *P* = $\sum_{j=1}^r \max(0, g_r(\underline{x})) + \sum_{i=1}^m |h_i(\underline{x})|$
\n3. *f* + *cP* = $\underbrace{f(\underline{x}) + \underline{\lambda}^T \underline{h}(\underline{x}) + \underline{\mu}^T \underline{g}(\underline{x})}_{I(\underline{x})}$ + $\frac{1}{2}$

1.
$$
P = \max\{0, g_1(\underline{x}), g_2(\underline{x}), ..., g_r(\underline{x}), |h_1(\underline{x})|, |h_2(\underline{x})|, ..., |h_m(\underline{x})|\}
$$

\n2. $P = \sum_{j=1}^r \max(0, g_r(\underline{x})) + \sum_{i=1}^m |h_i(\underline{x})|$
\n3. $f + cP = \underbrace{f(\underline{x}) + \underline{\lambda}^T \underline{h}(\underline{x}) + \underline{\mu}^T \underline{g}(\underline{x})}_{L(\underline{x}, \underline{\lambda}, \underline{\mu})} + \frac{1}{2} c \sum_{i=1}^r [\max(0, g_r(\underline{x}))]^2 + \frac{1}{2} c \underline{h}^T(\underline{x}) \underline{h}(\underline{x})$
\n4. $f + cP = L(\underline{x}, \underline{\lambda}, \underline{\mu}) + \frac{1}{2} c \underline{h}^T(\underline{x}) \underline{h}(\underline{x}) + \frac{1}{2} ||\nabla_x L(\underline{x}, \underline{\lambda}, \underline{\mu})||^2$

3.
$$
f + cP = \underbrace{f(\underline{x}) + \underline{\lambda}^T \underline{h}(\underline{x}) + \underline{\mu}^T \underline{g}(\underline{x})}_{L(\underline{x}, \underline{\lambda}, \underline{\mu})} + \frac{1}{2} c \sum_{i=1}^T [\max(0, g_r(\underline{x}))]^2
$$

\n4. $f + cP = L(\underline{x}, \underline{\lambda}, \underline{\mu}) + \frac{1}{2} c \underline{h}^T(\underline{x}) \underline{h}(\underline{x}) + \frac{1}{2} || \nabla_x L(\underline{x}, \underline{\lambda}, \underline{\mu}) ||^2$
\n5. $\beta f + cP = 0.f(\underline{x}) + \frac{1}{2} \nabla_x^T \underline{L}(\underline{x}, \underline{\lambda}) \nabla_x \underline{L}(\underline{x}, \underline{\lambda}) + \frac{1}{2} c \underline{h}^T(\underline{x}) \underline{h}(\underline{x})$
\nIn any case, $\alpha_k = \arg \min_{\alpha} \{ f(\underline{x}_k + \alpha \underline{d}_k) + cP(\underline{x}_k + \alpha \underline{d}_k) \}$

α

Note that some of the Penalty functions are non-differentiable. DO NOT USE LINE SEARCH TECHNIQUES THAT USE DERIVATIVE INFO. ONLY THOSE THAT USE FUNCTION EVALUATIONS (e.g., GS+QI) Step 4: Check e of the Penalty functions are non-differentiable.

LINE SEARCH TECHNIQUES THAT USE DERIVATIVE INE

E THAT USE FUNCTION EVALUATIONS (e.g., GS+QI)

for convergence. If not converged, $k = k + 1$ and go back to Step 2

Descent Property of *f+cP -* **1**

Descent Property of *f+cP* Consider the inequality constrained case: $f + cP = f(\underline{x}) + c\sum_{i=1}^{r} [g_i(\underline{x})]$ No loss of generality since 1 $f + cP = f(\underline{x}) + c \sum_{i=1}^{r} [g_i(\underline{x})]^+$ *j* $=$ $(\underline{x}) \leq 0$ $h_i(\underline{x}) = 0$ $h_i(\underline{x}) \leq 0$
 $h_i(\underline{x}) \leq 0$ *i i* $h_i(\underline{x})$ $h_i(\underline{x})$ $h_i(\underline{x})$ $=0 \Rightarrow \begin{cases} h_i(\underline{x}) \leq \\ h_i(\underline{x}) \end{cases}$ $\begin{cases} -h_i(\underline{x}) \leq 0 \end{cases}$ s of generality since $h_i(\underline{x}) = 0 \Rightarrow \begin{cases} h_i(\underline{x}) \le 0 \\ -h_i(\underline{x}) \le 0 \end{cases}$
 $a(\underline{x}) = f(\underline{x}) + c \sum_{j=1}^r [g_j(\underline{x})]^+, \text{Let } J(\underline{x}) = \{j : g_j(\underline{x}) > 0\}$ $\frac{1}{1-r}$
 $\frac{1}{1-\alpha}$ ($\frac{1}{2\alpha}$) $\frac{1}{1-\alpha}$ ($\frac{1}{2\alpha}$) $\frac{1}{1-\alpha}$ ($\frac{1}{2\alpha}$ ($\frac{1}{2\alpha}$) $\frac{1}{2\alpha}$ ($\frac{1}{2\alpha}$ + α) $\frac{1}{2\alpha}$ ($\frac{1}{2\alpha}$ + α) $\frac{1}{2\alpha}$ ($\frac{1}{2\alpha}$ + α) $\frac{1}{2\alpha}$ (\frac $\int_{-r}^{r} f(x+\alpha d) + c \sum_{j=1}^{r} [g_j(x+\alpha d)]^+$
 $(\underline{x}) + \alpha \nabla \underline{f}^T(\underline{x}) \underline{d} + c \sum_{j=1}^{r} [g_j(\underline{x}) + \alpha \nabla \underline{g}_j^T(\underline{x}) \underline{d})]^+ + O(\alpha)$ (x) + $\alpha \nabla f^T(\underline{x}) \underline{d} + c \sum_{j=1}^r [g_j(\underline{x}) + \alpha \nabla g_j^T(\underline{x}) \underline{d}]^+ + O(\alpha)$
(x) + $\alpha \nabla f^T(\underline{x}) \underline{d} + c \sum_{j=1}^r [g_j(\underline{x})]^+ + \alpha c \sum_{j \in J(\underline{x})} \nabla g_j^T(\underline{x}) \underline{d} + O(\alpha)$ Proof: This is because ∇g *r* $\lim_{n_i \to \infty} n_i(\underline{x}) = 0 \Rightarrow \begin{cases} -h_i(\underline{x}) \le 0 \\ -\infty \end{cases}$
 $\sum_{j=1}^r [g_j(\underline{x})]^+, \text{Let } J(\underline{x}) = \{j : g_j\}$ *r j Proof:* $a(x) = f(x) + c \sum_{j=1}^{r} [g_j(x)]^+$, Let *J*
 $a(x + \alpha \underline{d}) = f(x + \alpha \underline{d}) + c \sum_{j=1}^{r} [g_j(x + \alpha \underline{d})]$ *r* $+c\sum_{j=1}^{r}[g_j(\underline{x}+\alpha\underline{d})]^+$
 $T(\underline{x})\underline{d}+c\sum^r[g_j(\underline{x})+\alpha\nabla \underline{g}_j^T]$ $\sum_{j=1}^r [g_j(\underline{x}) + \alpha \nabla \underline{g}_j^T]$ *r* $T^T(\underline{x})\underline{d} + c \sum_{j=1}^r [g_j(\underline{x}) + \alpha \nabla \underline{g}_j^T(\underline{x})\underline{d})]^+$
 $T(\underline{x})\underline{d} + c \sum_{j=1}^r [g_j(\underline{x})]^+ + \alpha c \sum_{j=1}^r \nabla \underline{g}_j^T$ $\frac{1}{j}(\underline{x}) + \alpha \sqrt{g} \sum_{j \in J(\underline{x})} \nabla \underline{g}^T_j$
 $\frac{1}{j}(\underline{x})^+ + \alpha c \sum_{j \in J(\underline{x})} \nabla \underline{g}^T_j$ $f(\underline{x}) + \alpha \nabla \underline{f}^T(\underline{x}) \underline{d} + c \sum_{j=1}^r [g_j(\underline{x}) + \alpha \nabla \underline{g}_j^T(\underline{x}) \underline{d}]^+ + O(\alpha)$
 $f(\underline{x}) + \alpha \nabla \underline{f}^T(\underline{x}) \underline{d} + c \sum_{j=1}^r [g_j(\underline{x})]^+ + \alpha c \sum_{j \in J(\underline{x})} \nabla \underline{g}_j^T(\underline{x}) \underline{d} + O(\alpha)$ of generality since $h_i(\underline{x}) = 0 \Rightarrow \begin{cases} h_i(\underline{x}) \le 0 \\ -h_i(\underline{x}) \le 0 \end{cases}$
 $a(\underline{x}) = f(\underline{x}) + c \sum_{j=1}^r [g_j(\underline{x})]^+, \text{Let } J(\underline{x}) = \{ j : g_j(\underline{x}) \}$ $f(\underline{x} + \alpha \underline{d}) + c \sum_{j=1}^{r} [g_j(\underline{x} + \alpha \underline{d})]^+$
 $f(\underline{x}) + \alpha \nabla \underline{f}^T(\underline{x}) \underline{d} + c \sum_{j=1}^{r} [g_j(\underline{x}) + \alpha \nabla \underline{g}_j^T(\underline{x}) \underline{d})]^+ + O(\alpha)$ $\therefore a(\underline{x}) = f(\underline{x}) + c \sum_{j=1}^{r} [g_j(\underline{x})]^+, \text{Let } J(\underline{x}) = \{j : g$
 $\alpha \underline{d}$) = $f(\underline{x} + \alpha \underline{d}) + c \sum_{j=1}^{r} [g_j(\underline{x} + \alpha \underline{d})]^+$ $^{+}$ $\ddot{}$ enerality since $h_i(\underline{x}) = 0 \Rightarrow \begin{cases} h_i(\underline{x}) \le 0 \\ -h_i(\underline{x}) \le 0 \end{cases}$
= $f(\underline{x}) + c \sum_{j=1}^r [g_j(\underline{x})]^+$, Let $J(\underline{x}) = \{ j : g_j(\underline{x}) > 0 \}$ $[-h_i(\underline{x}) \le 0$

of: $a(\underline{x}) = f(\underline{x}) + c \sum_{j=1}^r [g_j(\underline{x})]^+$, Let $J(\underline{x}) = \{ j :$
 $+\alpha \underline{d}) = f(\underline{x} + \alpha \underline{d}) + c \sum_{j=1}^r [g_j(\underline{x} + \alpha \underline{d})]^+$ $^{+}$ $\overline{j=1}$
= $f(\underline{x} + \alpha \underline{d}) + c \sum_{j=1}^{r} [g_j(\underline{x} + \alpha \underline{d})]^+$
= $f(\underline{x}) + \alpha \nabla \underline{f}^T(\underline{x}) \underline{d} + c \sum_{j=1}^{r} [g_j(\underline{x}) + \alpha \nabla \underline{g}_j^T(\underline{x}) \underline{d})]^+ + O(\alpha)$ $\ddot{}$ $= f(\underline{x}) + \alpha \nabla \underline{f}^T(\underline{x}) \underline{d} + c \sum_{j=1}^r [g_j(\underline{x}) + \alpha \nabla \underline{g}_j^T(\underline{x}) \underline{d}]^+ + O(\alpha)$
= $f(\underline{x}) + \alpha \nabla \underline{f}^T(\underline{x}) \underline{d} + c \sum_{j=1}^r [g_j(\underline{x})]^+ + \alpha c \sum_{j \in J(\underline{x})} \nabla \underline{g}_j^T(\underline{x}) \underline{d} + O(\alpha)$ \sum^{\prime} $\sum_{i=1}^{r}$ \sum' $(\underline{x} + \alpha \underline{d})^+$
 $\sum_{j=1}^r [g_j(\underline{x}) + \alpha \nabla \underline{g}_j^T(\underline{x}) \underline{d})]^+ + O(\alpha)$
 $\sum_{j=1}^r [g_j(\underline{x})]^+ + \alpha c \sum_{j \in J(\underline{x})} \nabla \underline{g}_j^T(\underline{x}) \underline{d} + O(\alpha)$ = $f(\underline{x}) + \alpha \nabla \underline{f}^T(\underline{x}) \underline{d} + c \sum_{j=1}^r [g_j(\underline{x})]^+ + \alpha c \sum_{j \in J(\underline{x})}$
This is because $\nabla \underline{g}_j^T(\underline{x}) \underline{d} + g_j(\underline{x}) \le 0 \Rightarrow \nabla \underline{g}_j^T(\underline{x}) \underline{d} \le 0$
So, $a(\underline{x} + \alpha \underline{d}) = a(\underline{x}) + \alpha [\nabla \underline{f}^T(\underline{x}) \underline{d} + c \sum_{j \in J(\underline{x})} \nabla \underline{g}_j^T(\underline$ $\sum_{j \in J(\underline{x})} \nabla \underline{g}_j^T(\underline{x}) d \leq c \sum_{j \in J(\underline{x})} -g_j(\underline{x}) = -c \sum_{j=1}^r$ $\nabla \underline{f}^T(\underline{x}) \underline{d} + c \sum_{j=1}^r [g_j(\underline{x})]^+ + \alpha c \sum_{j \in J(\underline{x})} \nabla \underline{g}_j^T(\underline{x}) \underline{d} + c \sum_{j=1}^r [g_j(\underline{x})]^+ + \alpha c \sum_{j \in J(\underline{x})} \nabla \underline{g}_j^T(\underline{x}) \underline{d} + c \sum_{j \in J(\underline{x})} \nabla \underline{g}_j^T(\underline{x}) \underline{d} + c \sum_{j \in J(\underline{x})} \nabla \underline{g}_j^T(\underline{x}) \underline{d} + c \sum_{j \in J(\underline{x})} \nab$ This is because $\nabla g_j^T(\underline{x}) \underline{d} + g_j(\underline{x}) \le 0 \Rightarrow \nabla g_j^T(\underline{x}) \underline{d} \le 0$

So, $a(\underline{x} + \alpha \underline{d}) = a(\underline{x}) + \alpha [\nabla f^T(\underline{x}) \underline{d} + c \sum_{j \in J(\underline{x})} \nabla g_j^T(\underline{x}) \underline{d}]$

Since $c \sum_{j \in J(\underline{x})} \nabla g_j^T(\underline{x}) \underline{d} \le c \sum_{j \in J(\underline{x})} -g_j(\underline{x}) = -c \sum_{j=1}^r [g_j$ $\alpha \nabla \underline{f}^T(\underline{x}) \underline{d} + c \sum_{j=1}^r [g_j(\underline{x})]^+$
 *T*_j $(\underline{x}) \underline{d} + g_j(\underline{x}) \le 0 \Rightarrow \nabla \underline{g}_j^T$ $\mathcal{E}(\mathbf{y}, \mathbf{y}) = \sum_{j=1}^{r} [g_j(\mathbf{x}) + \alpha \cdot \mathbf{y}] \cdot \mathbf{y}$
 $\mathcal{E}(\mathbf{y}, \mathbf{y}) = \sum_{j=1}^{r} [g_j(\mathbf{x})] + \alpha \mathbf{y} \sum_{j \in J(\mathbf{x})} \nabla \mathbf{y}^T_j(\mathbf{x})$
 $\mathbf{y} = \sum_{j=1}^{r} [g_j(\mathbf{x})] \cdot \mathbf{y} \cdot \mathbf{y} \cdot \mathbf{y}$ $g_j(\underline{x}) \leq 0 \Rightarrow \nabla g_j^T(\underline{x})$
 $f'(x) \underline{d} + c \sum \nabla g_j^T$ $f(x) + \alpha \nabla f^{T}(x)dt + c \sum_{j=1}^{r} [g_{j}(x)]^{+} + \alpha c \sum_{j \in J}$
 s is because $\nabla g_{j}^{T}(x)dt + g_{j}(x) \le 0 \Rightarrow \nabla g_{j}^{T}(x)dt \le a(x + \alpha d) = a(x) + \alpha[\nabla f^{T}(x)dt + c \sum_{j \in J(x)} \nabla g_{j}^{T}(x)dt]$ *r T* $a(\underline{x}) + a[\underline{v}]$ $(\underline{x})\underline{a} + c$ $\sum_{j \in J(\underline{x})} \underline{v} g_j(\underline{x})$
 $f'_j(\underline{x})\underline{d} \le c$ $\sum_{j \in J(\underline{x})} -g_j(\underline{x}) = -c \sum_{j=1}^r [g_j]$ $\sum_{j \in J(\underline{x})} \nabla \underline{g}_j^T(\underline{x}) d \leq c \sum_{j \in J(\underline{x})} -g_j(\underline{x}) = -c \sum_{j \in J(\underline{x})}$ *s* because $\nabla g_j^T(\underline{x})\underline{d} + g_j(\underline{x}) \le 0 \Rightarrow \nabla g_j^T(\underline{x})\underline{d} \le$
 $\underline{x} + \alpha \underline{d} = a(\underline{x}) + \alpha[\nabla f^T(\underline{x})\underline{d} + c \sum_{j \in J(\underline{x})} \nabla g_j^T(\underline{x})\underline{d}$
 $c \sum_{j \in J(\underline{x})} \nabla g_j^T(\underline{x})\underline{d} \le c \sum_{j \in J(\underline{x})} -g_j(\underline{x}) = -c \sum_{j=1}^r [g_j(\underline{x})\underline{d} + c \sum_{j=$ = $f(\underline{x}) + \alpha \nabla \underline{f}^T(\underline{x}) \underline{d} + c \sum_{j=1}^r [\underline{f}]$

ecause $\nabla \underline{g}_j^T(\underline{x}) \underline{d} + g_j(\underline{x}) \leq 0$
 $\alpha \underline{d} = a(\underline{x}) + \alpha [\nabla \underline{f}^T(\underline{x}) \underline{d} + \alpha \underline{g}]$ ϵ $\ddot{}$ $\sum_{j \in J(\underline{x})} \nabla g_j^T(\underline{x}) d \leq c \sum_{j \in J(\underline{x})} -g_j(\underline{x}) = -c \sum_{j=1}^r$ $\sum_{j=1}^{N} [\mathcal{B}_j(\underline{x}) + \alpha \sqrt{\mathcal{B}_j}(\underline{x}) \underline{\mathcal{B}}] + O(\alpha)$
 $\sum_{j=1}^{N} [\mathcal{B}_j(\underline{x})]^+ + \alpha \mathcal{C} \sum_{j \in J(\underline{x})} \nabla \underline{\mathcal{B}}_j^T(\underline{x}) \underline{d} + O(\alpha)$
 $+ g_j(\underline{x}) \leq 0 \Rightarrow \nabla \underline{\mathcal{B}}_j^T(\underline{x}) \underline{d} \leq 0 \text{ if } g_j(\underline{x}) \geq 0$ = $f(x) + \alpha \nabla f^{T}(x) d + c \sum_{j=1}^{r} [g_{j}(x)]^{+} + \alpha c \sum_{j \in J(x)} \nabla g$
because $\nabla g_{j}^{T}(x) d + g_{j}(x) \le 0 \Rightarrow \nabla g_{j}^{T}(x) d \le 0$ if $g + \alpha d$) = $a(x) + \alpha [\nabla f^{T}(x) d + c \sum_{j \in J(x)} \nabla g_{j}^{T}(x) d]$ use $\nabla g_j^T(\underline{x})\underline{d} + g_j(\underline{x}) \le 0 \Rightarrow \nabla g_j^T(\underline{x})\underline{d} \le 0$
 $= a(\underline{x}) + \alpha[\nabla f^T(\underline{x})\underline{d} + c \sum_{j \in J(\underline{x})} \nabla g_j^T(\underline{x})\underline{d}]$
 $\nabla g_j^T(\underline{x})\underline{d} \le c \sum_{j \in J(\underline{x})} -g_j(\underline{x}) = -c \sum_{j=1}^r [g_j(\underline{x})]^+$ \sum = $f(x) + \alpha \nabla f^T(x) d + c \sum_{j=1} [g_j(x)]^+ + \alpha c \sum_{j \in J(x)} \nabla g^T_j(x) d$

because $\nabla g^T_j(x) d + g_j(x) \leq 0 \Rightarrow \nabla g^T_j(x) d \leq 0$ if $g_j(x) \geq$
 $-\alpha d) = a(x) + \alpha [\nabla f^T(x) d + c \sum_{j \in J(x)} \nabla g^T_j(x) d]$
 $\sum_{i \in J(x)} \nabla g^T_j(x) d \leq c \sum_{j \in J(x)} -g_j(x) = -c \sum_{j=1$

Descent Property of *f*+*cP* (continued)
\nSo,
$$
a(\underline{x} + \alpha \underline{d}) \le a(\underline{x}) + \alpha \nabla \underline{f}^T(\underline{x}) \underline{d} - c \sum_{j=1}^r [g_j(\underline{x})]^+
$$

\nFrom the necessary conditions of optimality, we have
\n
$$
\nabla \underline{f}^T(\underline{x}) \underline{d} = -\underline{d}^T \nabla_{xx}^2 L \underline{d} - \sum_{j=1}^r \mu_j \nabla \underline{g}^T_j(\underline{x}) \underline{d}
$$
\n
$$
= -\underline{d}^T \nabla_{xx}^2 L \underline{d} + \sum_{j=1}^r \mu_j g_j(\underline{x}) \qquad \qquad [\mu_j [\nabla \underline{g}^T_j(\underline{x}) \underline{d} + g_j(\underline{x})] = 0 \Rightarrow -\mu_j \nabla \underline{g}^T_j(\underline{x}) \underline{d} = \mu_j g_j(\underline{x})
$$
\n
$$
\le -\underline{d}^T \nabla_{xx}^2 L \underline{d} + \max_j (\mu_j) \sum_{j=1}^r [g_j(\underline{x})]^+
$$
\n
$$
a(\underline{x} + \alpha \underline{d}) \le a(\underline{x}) + \alpha \{-\underline{d}^T \nabla_{xx}^2 L \underline{d} - [c - \max(\mu_j)] \sum_{j=1}^r [g_j(\underline{x})]^+ \} + O(\alpha)
$$
\nSince $\nabla_{xx}^2 L$ is PD and if $c > \max_j \mu_j$, \exists an α a $a(\underline{x} + \alpha \underline{d}) < a(\underline{x})$

- What if $\nabla_{xx}^2 L$ is not PD? Use Augmented Lagrangian
- If don't want to compute Hessian, use Quasi-Newton Method

SAMP

Quasi-Newton Version of SQP - 1

General Algorithm: Quasi-Newton Version

 $\frac{1}{0}$ and a PD matrix B_0 (approximation to the Hessian) or its square root L_0 for square root version Step 1: Given an initial estimate of x_0 and a PD matrix estimate of \underline{x}_0 and a PD n
essian) or its square root
min $\frac{1}{2} \underline{d}_k^T B_k \underline{d}_k + \nabla \underline{f}^T(\underline{x}_k)$ **EXECUTE:** $\frac{x_0}{x_0}$ and a PD matrix B_0 *d* a PD matrix
 d or its square root L_0 for $\underline{d}_k^T B_k \underline{d}_k + \nabla \underline{f}^T(\underline{x}_k) \underline{d}_k$

(approximation to the Hessian) or its square root
$$
L_0
$$
 for square root version
\n
$$
\min \frac{1}{2} \underline{d}_k^T B_k \underline{d}_k + \nabla \underline{f}^T (\underline{x}_k) \underline{d}_k
$$
\nStep 2: Solve the QPP: s.t. $\nabla \underline{h}^T (\underline{x}_k) \underline{d}_k + \underline{h} (\underline{x}_k) = 0 \Rightarrow$ Get $\underline{d}_{k+1}, \underline{\lambda}_{k+1}, \underline{\mu}_{k+1}$
\n
$$
\nabla \underline{g}^T (\underline{x}_k) \underline{d}_k + \underline{g} (\underline{x}_k) \leq 0
$$
\nStep 3: Perform Line search to obtain α_k , where $\alpha_k = \arg \min_{\alpha} \{f + cP\}$

α

Step 4: Check for convergence. If not go to Step 5 r conve Eq. 1 Line set B_k (or L_k)

Step 5: Update B_k (or L_k) via generalize d BFGS update

Step 4: Check for convergence. If not go to Step 5
\nStep 5: Update
$$
B_k
$$
 (or L_k) via generalized BFGS update
\n
$$
B_{k+1} = B_k + \frac{\hat{\omega}_k \hat{\omega}_k^T}{\hat{p}_k^T \hat{\omega}_k} - \frac{B_k p_k p_k^T B_k}{p_k^T B_k p_k}; \quad \hat{\omega}_k = \theta_k \underline{q}_k + (1 - \theta_k) B_k \underline{p}_k; \quad \boxed{\frac{q_k = \nabla_x L(\underline{x}_{k+1}, \underline{\lambda}_{k+1}, \underline{\mu}_{k+1}) - \nabla_x L(\underline{x}_k, \underline{\lambda}_{k+1}, \underline{\mu}_{k+1})}{\frac{p_k = \underline{x}_{k+1} - \underline{x}_k = \alpha_k \underline{d}_k}{\frac{p_k}{p_k^T B_k \underline{p}_k}}}
$$
\nsuggested value of θ_k (empirical): $\theta_k = \begin{cases} 1 & \text{if } \underline{p}_k^T \underline{q}_k \geq 0.2 \underline{p}_k^T B_k \underline{p}_k \\ \frac{0.8 \underline{p}_k^T B_k \underline{p}_k}{\underline{p}_k - \underline{p}_k^T \underline{q}_k} & \text{if } \underline{p}_k^T \underline{q}_k \leq 0.2 \underline{p}_k^T B_k \underline{p}_k \end{cases}$

• Powell (Math Programming, Vol. 15, 1978) shows that if $\alpha_k \approx 1$, the method has super linear convergence. However, one can find problems where $\alpha_k \neq 1$ for \underline{x}_k arbitrarily close to \underline{x}^* . Known as Maratos effect
 $f(\underline{x} + \underline{d}) + c \sum_j |g_j^+(\underline{x} + \underline{d})| > f(\underline{x}) + c \sum_j |g_j^+(\underline{x})|$

f+cP

$$
f(\underline{x}+\underline{d})+c\sum_{j}|g_{j}^{+}(\underline{x}+\underline{d})|>f(\underline{x})+c\sum_{j}|g_{j}^{+}(\underline{x})|
$$

Maratos Effect

15

• Solution of QPP: At step *k*, we have **SQP Algorithm with 2nd Order Correction - 1** (1) min $\nabla f^T(\underline{x}_k) \underline{d}_k + \frac{1}{2}$ 2 s.t. $\underline{h}(\underline{x}_k) + \nabla \underline{h}^T(\underline{x}_k) \underline{d}_k = 0$ $(\underline{x}_k) + \nabla g^T(\underline{x}_k) \underline{d}_k \leq \underline{0}$ $T_{(r)}$ _d $\pm \frac{1}{d}$ ^T $\nabla \underline{f}^T(\underline{x}_k) \underline{d}_k + \frac{1}{2} \underline{d}_k^T B_k \underline{d}_k$ (2) min $\frac{1}{2}$ $\frac{h(x_k) + \nabla \underline{h}^T(x_k) d_k = 0}{2^{-k} - k}$ *S.t.* $h(x_k + d_k) + \nabla \underline{h}^T(x_k) \underline{p}_k = 0$ $\underline{g}(\underline{x}_k) + \nabla \underline{g}^T(\underline{x}_k) \underline{d}_k \leq \underline{0}$ mprov
(2) min 2 $(\underline{x}_k + \underline{d}_k) + \nabla \underline{g}^T(\underline{x}_k) \underline{p}_k \leq \underline{0}$ • Solve two quadratic programs to improve convergence rate:

(1) $\min \nabla f^T(\underline{x}_k) \underline{d}_k + \frac{1}{2} \underline{d}_k^T B_k \underline{d}_k$ (2) $\min \frac{1}{2} \underline{p}_k^T \underline{p}_k$ 2^{-k} ²
 h(x_k + d_k) + ∇ *h*^{*T*}(x_k) p_k = <u>0</u> S.t. $\underline{h}(\underline{x}_k + \underline{a}_k) + \underline{v} \underline{h}(\underline{x}_k) \underline{p}_k = \underline{v}$

<u>g</u> ($\underline{x}_k + \underline{d}_k$) + $\nabla \underline{g}^T(\underline{x}_k) \underline{p}_k \leq \underline{0}$ 2 $g(\underline{x}_k) + \nabla \underline{g}^T(\underline{x}_k) \underline{d}_k \leq \underline{0}$ $g(\underline{x}_k + \underline{d}_k) + \nabla \underline{g}^T(\underline{x}_k)$
 $\underline{x}_{k+1} = \underline{x}_{k+1} + \alpha_k \underline{d}_k + \alpha_k^2 \underline{p}_k; \alpha_k = \arg \min_{\alpha} (f + cP)$ α s.t. $A_1^T \underline{d} = \underline{b}_1$ $A_1 \sim \nabla_{\underline{A}}^2$, $A_2 \sim \nabla_{\underline{B}}^2$
 $A_3 \sim \nabla_{\underline{B}}^2$, $A_2 \sim \nabla_{\underline{B}}^2$ $A_2^T \underline{d} \leq \underline{b}_2$ $\underline{b}_1 = -\underline{h}, \ \underline{b}_2$ 1 min 2 T_{d} ¹ d^T $\lim_{d} g_k^T d + \frac{1}{2} d^T B_k d$ $\frac{g_k}{g_k} \sim \frac{\nabla f(x_k)}{\nabla^2 L} ON \nabla^2$ have
 (\underline{x}_k) $k_k \sim \nabla f(\underline{x}_k)$
 $k_k \sim \nabla^2_{xx} L, QN,$ $\frac{1}{\sqrt{2}}$ $\frac{1}{2}$ $\nabla_k \sim \nabla_{xx}^2 L, QN, \nabla_{xx}^2 L + c_k \nabla_{\underline{H}}^2 \nabla_{\underline{H}}^2$ *k*, we ha $\underline{g}_k \sim \nabla \underline{f}(\underline{x}_k)$
 $B_k \sim \nabla_{xx}^2 L$, QN, $\nabla_{xx}^2 L + c_k \nabla \underline{h} \nabla \underline{h}^T$ $A_1 \sim \nabla \underline{h}$, $A_2 \sim \nabla \underline{g}$
 $\underline{b}_1 = -\underline{h}$, $\underline{b}_2 = -\underline{g}$ $\nabla_{xx}^2 L$, QN, $\nabla_{xx}^2 L$
 $\nabla_{\underline{B}}$, $A_2 \sim \nabla_{\underline{g}}$ $\sim \nabla \underline{h}$, $A_2 \sim \nabla \underline{g}$
= $-\underline{h}$, $\underline{b}_2 = -\underline{g}$ Suppose we have a feasible point \overline{d}_l , $\mathbf{u}_1 \underline{\mathbf{u}}_1 = \underline{\mathbf{v}}_1$ $\frac{1}{2} \underline{a}_l \geq \underline{b}_2$ *T l* $l \sim \frac{I}{I}$ *l* $A_1^T \underline{d}_l = \underline{b}$ *d* $A_2^T \underline{d}_l \leq \underline{b}$ $=$ $b_{\scriptscriptstyle 1}$ | $\left\{\begin{array}{c}\n\frac{1}{2} \frac{d}{dt} - \frac{1}{2} \\
\frac{1}{2} \frac{1$ $\leq \underline{b}_2$ \int $\frac{\text{min}}{1}$ $\sum_{i=1}$ $\frac{z_i + \sum_{j=1}^{n} z_j}{1}$ s.t. $A_1^T \underline{d}_l + \underline{z} = \underline{b}_1, A_2^T \underline{d}_l + \underline{y} = \underline{b}_2, \underline{y} \ge \underline{0}$ Solve Phase I LP min *m r* $i + \sum y_j$ $\sum_{i=1} \frac{\lambda_i}{j} + \sum_{j}$ $\underline{d}_l + \underline{z} = \underline{b}_1, \ \mathbf{A}_2^T \underline{d}_l + \underline{y} = \underline{b}_2, \underline{y} \ge \underline{0}$ $z_i + \sum y$ $\sum_{i=1}^{j} \frac{z_i}{j} + \sum_{j=1}^{j} y_j$ $\ddot{}$ re Phase I LP
 $\sum_{i=1}^{m} z_i + \sum_{i=1}^{r} y_i$

1 2 * $\frac{1}{1} \underline{\lambda}^* - \hat{A}_2$ olution of QPP (continued):
At \underline{d}_l : Equality constraints are satisfied and some inequality constraints equality Define \hat{A} \hat{A}^T $\left| r^* \right|$ active inequality At optimum $g_k + B_k d^* = -A_1 \lambda^* - \hat{A}$ $[A_2^{\dagger}]^T$ acuve mequanty
At optimum $\underline{g}_k + B_k \underline{d}^* = -A_1 \underline{\lambda}^* - \hat{A}_2 \underline{\mu}^*$
If we know active constraints at \underline{d}^* , we can *l d T T T* A_1^T *m* \hat{A}^T_2 | r *m* equality
 r^* active inequality
 $B_k \underline{d}^* = -A_1 \underline{\lambda}^* - \hat{A}_2 \underline{\mu}^*$ d^* , we can actually solve an equality constrained * active inequality
 $d^* = -A_1 \lambda^* - \hat{A}_2 \mu^*$ ty contraints are $\begin{bmatrix} A_1^T \\ A_2^T \end{bmatrix}$ *m* equal ality constraints are $=\begin{bmatrix} A_1^T \\ \hat{A}_2^T \end{bmatrix} r^*$ active ine $\begin{cases} \frac{T}{2} \int r^* \text{ active inequality} \\ + B_k \underline{d}^* = -A_1 \underline{\lambda}^* - \hat{A}_2 \underline{\mu}^* \end{cases}$ 1 * If we know active constraints at \underline{d}^* , we can actually solve an problem: min $\underline{g}_k^T \underline{d} + \frac{1}{2} \underline{d}^T B_k \underline{d}$ s.t. $\hat{A}^T \underline{d} = \hat{\underline{b}}$; $\hat{\underline{b}} = \begin{pmatrix} \underline{b}_1 \\ \hat{b}_2^* \end{pmatrix}$ problem: min $\underline{g}_k^T \underline{d} + \frac{1}{2} \underline{d}^T B_k \underline{d}$ s.t. $\hat{A}^T \underline{d} = \hat{\underline{b}}$; $\hat{\underline{b}} = \begin{pmatrix} \underline{b}_1 \\ \hat{b}_2^* \end{pmatrix}$
Unfortunately don't know r^{*}, so our procedure is iterative: • Start with the current working set S_l • Go to the next point d_{l+1} *T* $\frac{1}{k}d + \frac{1}{2}d^T B_k d$ s.t. \hat{A}^T $\sum_k^T \underline{d} + \frac{1}{2} \underline{d}^T B_k$ *b* ive constraints at \underline{d}^* , we can actually
 $\underline{g}_k^T \underline{d} + \frac{1}{2} \underline{d}^T B_k \underline{d}$ s.t. $\hat{A}^T \underline{d} = \hat{\underline{b}}$; $\hat{\underline{b}} = \begin{pmatrix} \frac{b}{b} \\ \frac{b}{b} \end{pmatrix}$ *r*^{*}, so our procedure is iterative:
the current working set S_l
next point $\underline{d}_{l+1} = \underline{d}_l + \underline{p}_l$ Ily solve an equali
 $\left(\frac{b_1}{a}\right)$ constraints at \underline{d}^* , we can actually solve an equals
+ $\frac{1}{2} \underline{d}^T B_k \underline{d}$ s.t. $\hat{A}^T \underline{d} = \hat{\underline{b}}$; $\hat{\underline{b}} = \begin{pmatrix} \underline{b}_1 \\ \hat{b}_2^* \end{pmatrix}$ See if we need to update $S_l \to S_{l+1}$ procedure is
 *d*_{*l*+1} = <u>*d*_{*l*} + *p*_{*l*}</u> *l*
 l = \underline{d}_l + \underline{p}_l
 $S_l \rightarrow S_l$ Repeat until \longrightarrow \bullet Go to the next point \underline{d}_{l+1} $^{+}$ • Start with the current working set S_l

• Go to the next point $\underline{d}_{l+1} = \underline{d}_l + \underline{p}_l$

• See if we need to update $S_l \rightarrow S_{l+1}$ Convergence Solution of QPP (continued): **SQP Algorithm with 2nd Order Correction - 2**

 \Box How to get the best \underline{p}_l ? The solution of $\hat{A} = [\hat{A}_1 \ \hat{A}_2]$; Suppose $\hat{A} = [Q_i \ \overline{Q}_i] \begin{bmatrix} R_i \\ 0 \end{bmatrix} = Q_i R_i \Rightarrow Q_i^T \hat{A} = R_i$ 1 1 ¹ \hat{A}_2]; Suppose $\hat{A} = [Q_i \ \bar{Q}_i] \begin{bmatrix} A_i \\ 0 \end{bmatrix} = Q_i R_i \Rightarrow Q_i^T$

(\hat{A}) column space of \hat{A} $R(\hat{A})$ column space of \hat{A}
Orthogonal to $\hat{A} \Rightarrow \overline{Q}_l^T \hat{A} = 0 \Rightarrow \hat{A}^T \overline{Q}_l = 0$ \overline{Q}_l = Orthogonal to $\hat{A} \Rightarrow \overline{Q}_l^T \hat{A} = 0 \Rightarrow \hat{A}^T \overline{Q}_l = 0$
Also, $Q_l^T \overline{Q}_l = 0$; Since \underline{d}_l and \underline{d}_{l+1} are feasible Also, $Q_l^T \overline{Q}_l = 0$; Since \underline{d}_l and \underline{d}_{l+1} are f
 $\hat{A}^T \underline{d}_{l+1} = \hat{A}^T \underline{d}_l + \hat{A}^T \underline{p}_l = \underline{0} \Rightarrow \hat{A}^T \underline{p}_l = \underline{0}$ Since columns $\begin{bmatrix} Q_l \end{bmatrix} \begin{bmatrix} R_l \ 0 \end{bmatrix} = Q_l R_l \Rightarrow Q_l^T \hat{A} = R_l$ $l =$ $A^T \hat{A} = 0 \Rightarrow \hat{A}^T$ $Q_l = R(\hat{A})$ column space of \hat{A}
 \overline{Q}_l = Orthogonal to $\hat{A} \Rightarrow \overline{Q}_l^T \hat{A} = 0 \Rightarrow \hat{A}^T \overline{Q}_l$ *T* $\overline{Q}_l^T \overline{Q}_l = 0$; Since \underline{d}_l and \underline{d}_{l+1} Iso, $Q_i^T Q_i = 0$; Since \underline{d}_l and \underline{d}_l
 $T d_{l+1} = \hat{A}^T d_l + \hat{A}^T p_l = 0 \Rightarrow \hat{A}^T$ Also, $Q_l^T \overline{Q}_l = 0$; Since \underline{d}_l and \underline{d}_{l+1}
 $\hat{A}^T \underline{d}_{l+1} = \hat{A}^T \underline{d}_l + \hat{A}^T \underline{p}_l = \underline{0} \Rightarrow \hat{A}^T \underline{p}_l$ Follow to get the best \underline{P}_l ?
 $\hat{A} = [\hat{A}_1 \ \hat{A}_2]$; Suppose $\hat{A} = [Q_l \ \overline{Q}_l] \begin{bmatrix} R_l \\ 0 \end{bmatrix} = Q_l R_l \Rightarrow Q_l^T \hat{A} = R_l$ $A = [A_1 \ A_2]$; Suppose $A = [Q_i]$
 $Q_i = R(\hat{A})$ column space of \hat{A} Drthogonal to $\hat{A} \Rightarrow \overline{Q}_l^T \hat{A} = 0$
 $Q_l^T \overline{Q}_l = 0$; Since \underline{d}_l and \underline{d}_l $\ddot{}$ $\begin{bmatrix} R_l \end{bmatrix} = O_l R_l =$ w to get the best \underline{P}_l ?
= [\hat{A}_1 \hat{A}_2]; Suppose $\hat{A} = [Q_l \ \overline{Q}_l] \begin{bmatrix} R_l \\ 0 \end{bmatrix} = Q_l R_l \Rightarrow Q_l^T \hat{A} = R_l$ = $R(\hat{A})$ column space of \hat{A}
= Orthogonal to $\hat{A} \Rightarrow \overline{Q}_l^T \hat{A} = 0 \Rightarrow \hat{A}^T \overline{Q}_l = 0$ $Q_l^T \overline{Q}_l = 0$; Since \underline{d}_l and \underline{d}_{l+1} are feasib
= $\hat{A}^T \underline{d}_l + \hat{A}^T \underline{p}_l = \underline{0} \Rightarrow \hat{A}^T \underline{p}_l = \underline{0}$ $+\hat{A}^T \underline{p}_l = \underline{0} \Rightarrow \hat{A}^T \underline{p}_l = \underline{0}$
of Q_l and \overline{Q}_l span R^n , we can write $p_{l} = Q_{l} y_{l} + \overline{Q}_{l} z_{l}$
 $\hat{A}^{T} p_{l} = \hat{A}^{T} Q_{l} y_{l} + \hat{A}^{T} \overline{Q}_{l} z_{l} = 0 \Rightarrow R_{l}^{T} y_{l} = 0 \Rightarrow y_{l} = 0$ *n* Since columns of Q_l and \overline{Q}_l
 $\underline{p}_l = Q_l \underline{y}_l + \overline{Q}_l \underline{z}_l$ $p_l = Q_l y_l + Q_l z_l$ $I_l = Q_l \underline{y}_l + Q_l \underline{z}_l$
 $I_l = \hat{A}^T O_l v_l + \hat{A}^T \overline{O_l} z_l = 0 \Rightarrow R_l^T$ $p_l = Q_l \underline{y}_l + \overline{Q}_l \underline{z}_l$
 $\hat{A}^T \underline{p}_l = \hat{A}^T Q_l \underline{y}_l + \hat{A}^T \overline{Q}_l \underline{z}_l = \underline{0} \Rightarrow R_l^T \underline{y}_l = 0 \Rightarrow \underline{y}_l$ \therefore $p_l = \overline{Q}_l \underline{z}_l$ $\frac{d^T p_l}{dt^T} = \frac{0}{Q_l} \Rightarrow \hat{A}^T \underline{p}_l =$
 \overline{Q}_l and \overline{Q}_l span R $p_l = \hat{A}^T Q$
 $p_l = \overline{Q}_l \underline{z}$ $Q_l \underline{y}_l + \overline{Q}_l \underline{z}_l$
= $\hat{A}^T Q_l \underline{y}_l + \hat{A}^T \overline{Q}_l \underline{z}_l = \underline{0} \Rightarrow R_l^T \underline{y}_l = 0 \Rightarrow \underline{y}_l = \underline{0}$ $\hat{A}^T \underline{p}_l = \hat{A}^T \underline{Q}_l \underline{y}_l$
 $\therefore \boxed{\underline{p}_l = \overline{Q}_l \underline{z}_l}$ **P Algorithm with 2nd Order Correction - 3**

1 he problem of finding \underline{p}_l can be w
min $\underline{g}_k^T(\underline{d}_l + \underline{p}_l) + \frac{1}{2}(\underline{d}_l + \underline{p}_l)^T B_k(\underline{d}_l + \underline{p}_l)$ s.t. $\hat{A}^T p_i = 0$ 1 $\hat{A}^T \underline{p}_l = \underline{0}$

min($\underline{g}_k + B_k \underline{d}_l$)^T $\underline{p}_l + \frac{1}{2}$ $\hat{\mathbf{A}}^T$ min($\underline{g}_k + B_k$
s.t. $\hat{A}^T \underline{p}_l = 0$ 1 s.t. $A^T \underline{p}_l = \underline{0}$
min $\tilde{g}_k^T \underline{p}_l + \frac{1}{2}$ $\hat{\mathbf{A}}^T$ \Rightarrow min $\tilde{g}_k^T \underline{p}_l$
s.t. $\hat{A}^T \underline{p}_l = 0$ s.t. $\hat{A}^T \underline{p}_l = \underline{0}$
where $\tilde{g}_k = \underline{g}_k + B_k \underline{d}_l$ So, the problem of finding p_l is another QPP, <u>but</u> simpler $constants \Rightarrow can solve very easily!!!$ T_t $(d, +p)$, T_t $(d, +p)$, T_t blem of finding \underline{p}_l can be $\frac{r}{k}(\underline{d}_l + \underline{p}_l) + \frac{1}{2}(\underline{d}_l + \underline{p}_l)^T B_k(\underline{d}_l + \underline{p}_l)$ $l =$ $T_{\dot{p}_1} + \frac{1}{p_1} p_1^T$ $=\underline{0}$
 k + $B_k \underline{d}_l$ ^T \underline{p}_l + $\frac{1}{2} \underline{p}_l^T B_k \underline{p}_l$ $l =$ $\frac{T}{T}$
 $\frac{T}{T}$
 $\frac{1}{T}$
 $\frac{1}{T}$ $\sum_{k}^{T} p_{l} + \frac{1}{2} p_{l}^{T} B_{k} p_{l}$ $l =$ $A^T \underline{p}^T$ $A^T \underline{p}$
 $\hat{A}^T \underline{p}$ min
 $\hat{A}^T \underline{p}$ s.t. $\hat{A}^T \underline{p}_l = \underline{0}$
 $\Rightarrow \min(\underline{g}_k + B_k \underline{d}_l)^T \underline{p}_l + \frac{1}{2} \underline{p}_l^T B_k \underline{I}$ s.t. $\hat{A}^T \underline{p}_l = \underline{0}$
 $\Rightarrow \min \tilde{g}_k^T \underline{p}_l + \frac{1}{2} \underline{p}_l^T B_k \underline{I}$ • The problem of finding p_l can be written as: **SQP Algorithm with 2nd Order Correction**

5QP Algorithm with 2nd Order Correction - 5
\n
$$
\begin{bmatrix}\nB_k & \hat{A} \\
\hat{A}^T & 0\n\end{bmatrix}\n\begin{bmatrix}\n\underline{p}_l \\
\hat{Z}_{l+1}\n\end{bmatrix} =\n\begin{bmatrix}\n-\underline{g}_k - B_k \underline{d}_l \\
\vdots \\
\hat{Q}\n\end{bmatrix}
$$
\nor\n
$$
\begin{bmatrix}\nB_k & \hat{A} \\
\hat{A}^T & 0\n\end{bmatrix}\n\begin{bmatrix}\n\underline{d}_{l+1} \\
\hat{Z}_{l+1}\n\end{bmatrix} =\n\begin{bmatrix}\n-\underline{g}_k \\
\hat{L}\n\end{bmatrix}
$$
\nSince $\underline{d}_{l+1} = Q_l c_{l+1} + \overline{Q}_l \underline{a}_{l+1}$ \n
$$
\Rightarrow \begin{bmatrix}\nB_k Q_l & B_k \overline{Q}_l & Q_l R_l \\
R_l^T Q_l^T Q_l & 0 & 0\n\end{bmatrix}\n\begin{bmatrix}\n\underline{e}_{l+1} \\
\hat{Z}_{l+1}\n\end{bmatrix} =\n\begin{bmatrix}\n-\underline{g}_k \\
\hat{L}\n\end{bmatrix}
$$
\n
$$
\Rightarrow R_l^T c_{l+1} = \hat{b} \Rightarrow \text{solve for } c_{l+1} \text{ in } O(\frac{n^2}{2}) \text{ operations.}
$$
\n
$$
\overline{Q}_l^T B_k \overline{Q}_l \underline{a}_{l+1} = -\overline{Q}_l^T [g_k + B_k Q_l c_{l+1}]
$$

. ,,,,,,

 $\overline{}$ L. \mathcal{L} $\overline{}$ Ų L. **D** D

SQP Algorithm with 2nd Order Correction - 6
Do Cholesky on
$$
\overline{Q}_l^T B_k \overline{Q}_l = \overline{U}_l \overline{U}_l^T
$$
, so $\underline{a}_{l+1} = -[\overline{U}_l \overline{U}_l^T]^{-1} \overline{Q}_l^T [\underline{g}_k + B_k Q_l \underline{c}_{l+1}]$
 $\Rightarrow \underline{d}_{l+1} = Q_l \underline{c}_{l+1} - \overline{Q}_l [\overline{U}_l \overline{U}_l^T]^{-1} \overline{Q}_l^T [\underline{g}_k + B_k Q_l \underline{c}_{l+1}]$
Finally, $R_l \underline{A}_{l+1} = -Q_l^T [\underline{g}_k + B_k Q_l \underline{c}_{l+1} + B_k \overline{Q}_l \underline{a}_{l+1}] = -Q_l^T [\underline{g}_k + B_k \underline{d}_{l+1}]$
or $\underline{A}_{l+1} = -R_l^{-1} Q_l^T [\underline{g}_k + B_k \underline{d}_{l+1}]$ Multiplier vector in $O(n^2)$ operations.

 $\Rightarrow \underline{a}_{l+1} - \underline{Q}_l \underline{c}_{l+1} - \underline{Q}_l [\underline{U}_l \underline{U}_l] \quad \underline{Q}_l [\underline{g}_k + B_k \underline{Q}_l]$
Finally, $R_l \underline{\lambda}_{l+1} = -\underline{Q}_l^T [\underline{g}_k + B_k \underline{Q}_l \underline{c}_{l+1} + B_k \overline{Q}_l]$
or $\underline{\lambda}_{l+1} = -R_l^{-1} \underline{Q}_l^T [\underline{g}_k + B_k \underline{d}_{l+1}]$ Multiplier v $\begin{aligned} & [\bar{U}_l \, \bar{U}_l^T]^{-1} \bar{Q}_l^T [\, \underline{g}_k + B_k \mathcal{Q}_l \underline{c}_{l+1}] \ & \int_l^T [\, g_{k} + B_k \mathcal{Q}_l \underline{c}_{l+1} + B_k \bar{\mathcal{Q}}_l \underline{a}_{l+1} \,] = - \mathcal{Q}_l^T \end{aligned}$ = $Q_l C_{l+1} - Q_l U_l U_l$ J Q_l
 Q_l , $R_l \underline{\lambda}_{l+1} = -Q_l^T [\underline{g}_k + B_k Q_l]$
 $Q_l = -R_l^{-1} Q_l^T [\underline{g}_k + B_k \underline{d}_{l+1}]$ Mu λ g_{l+1}] = $-Q_{l}^{T}$ [g_{k} + $B_{k}d_{l+1}$]
ector in O(*n*²) operations.

or $\underline{\lambda}_{l+1} = -R_l^{-1} Q_l^T [\underline{g}_k + B_k \underline{d}_{l+1}]$ Multiplier vector in O(*n*²) operations

[pdate of working set:

If $\underline{d}_{l+1} = \underline{d}_l \Rightarrow \underline{p}_l = 0 \Rightarrow \underline{d}_l$ is optimal w.r.t current set of constraints S or $\underline{\lambda}_{l+1} = -R_l^{-1} Q_l^T [\underline{g}_k + B_k \underline{d}_{l+1}]$ Mu

Update of working set:

• If $\underline{d}_{l+1} = \underline{d}_l \Rightarrow \underline{p}_l = 0 \Rightarrow \underline{d}_l$ is optin 1y, $R_1 \underline{A}_{l+1} - Q_l L_1 \underline{B}_k + D_l$
 $\frac{1}{2} = -R_l^{-1} Q_l^T L_2 \underline{B}_k + B_k \underline{d}_{l+1}$ *T* $\mathbf{u}_1 \mathbf{y}, \mathbf{A}_l \mathbf{A}_{l+1} = -\mathbf{Q}_l \mathbf{I} \mathbf{g}_k +$
 $\mathbf{A}_{l+1} = -R_l^{-1} \mathbf{Q}_l^T [\mathbf{g}_k + B_k \mathbf{d}_l]$ \overline{a} μ_{1} , $N_{l}\underline{A}_{l+1}$ - Q_{l} L_{l} L_{k} + $D_{k}\underline{Q}_{l}\underline{C}_{l+1}$
 Q_{l} = $-R_{l}^{-1}Q_{l}^{T}$ $[\underline{g}_{k}$ + $B_{k}\underline{d}_{l+1}]$ Mul $\lambda_{1,1} = -R_1^{-1}O_1^T[g_1 + B_1d_{1,1}]$ Multiplier vector in $O(n^2)$

- \Box Update of working set:
	- 1 ate of working set:
 $\mu_{l+1} = \underline{d}_l \Rightarrow \underline{p}_l = 0 \Rightarrow \underline{d}_l$ is optimal w.r.t current set of constraints S_l $^{+}$
- 1 If $\underline{d}_{l+1} = \underline{d}_l \Rightarrow \underline{p}_l = 0 \Rightarrow \underline{d}_l$ is optimal w.r.t current set of constraints S_l
If $\underline{p}_l \neq 0$ but is feasible for <u>all</u> constraints, $\underline{d}_{l+1} = \underline{d}_l + \underline{p}_l$ is the new point. If $\underline{d}_{l+1} = \underline{d}_l \Rightarrow \underline{p}_l = 0 \Rightarrow \underline{d}_l$ is

If $\underline{p}_l \neq 0$ but is feasible for \underline{a}

If $\mu_q \geq 0 \forall q$ of inequality co all $\mu_q \geq 0 \,\forall q$ of inequality constraints, stop \Rightarrow Optimal late of working set:
 $\underline{d}_{l+1} = \underline{d}_l \Rightarrow \underline{p}_l = 0 \Rightarrow \underline{d}_l$ is optimal w.r.t current set of $\underline{p}_l \neq 0$ but is feasible for <u>all</u> constraints, $\underline{d}_{l+1} = \underline{d}_l + \underline{p}_l$ $^{+}$ Update of working set:

■ If $\underline{d}_{l+1} = \underline{d}_l \Rightarrow \underline{p}_l = 0 \Rightarrow \underline{d}_l$ is optimal w.r.t current set of constrain

■ If $\underline{p}_l \neq 0$ but is feasible for <u>all</u> constraints, $\underline{d}_{l+1} = \underline{d}_l + \underline{p}_l$ is the ne $\mu_1 = \underline{d}_l \Rightarrow \underline{p}_l = 0 \Rightarrow \underline{d}_l$ is optimal w.r.t current set of $\alpha \neq 0$ but is feasible for <u>all</u> constraints, $\underline{d}_{l+1} = \underline{d}_l + \underline{p}_l$ i
≥ 0 $\forall q$ of inequality constraints, stop \Rightarrow Optimal 0 but is feasible for <u>all</u> constraints, $d_{l+1} = d_l + p_l$ is the new point.
 $\cdot 0 \forall q$ of inequality constraints, stop ⇒ Optimal

is not feasible ⇒ some constraint is voilated. So, let $\underline{d}_{l+1} = \underline{d}_l + \alpha_l \underline{p}_l$
	-

• If
$$
\underline{d}_{l+1}
$$
 is not feasible \Rightarrow some constraint is violated. So, let $\underline{d}_{l+1} = \underline{d}_l + \alpha_l$
where: $\alpha_l = \min_{\substack{a_i^T p_l > 0 \\ i \notin S_l}} \{1, \frac{b_i - \underline{a}_i^T \underline{d}_l}{\underline{a}_i^T \underline{p}_l}\}$
 $i_a = \arg \min_{\substack{a_i^T p_l > 0 \\ i \notin S_l}} \{1, \frac{b_i - \underline{a}_i^T \underline{d}_l}{\underline{a}_i^T \underline{p}_l}\} \Rightarrow S_{l+1} = S_l \cup \{i_a\}$

SQP Algorithm with 2nd Order Correction - 7

- How to drop active inequality constraints:
	- to drop active inequality constraints:
 d_{l+1} is feasible for all constraints in S_l
	- $-\underline{d}_{l+1}$ is feasible for all co
 $-\text{ If all }\mu_q > 0 \Rightarrow \text{Optimal}$
	- If all $\mu_q > 0 \Rightarrow$ Optimal
- Find i_d = arg min{ μ_q }, $S_{l+1} = S_l \{i_d\}$
- Algorithm:

- Find i_d = arg min{ μ_q }, $S_{l+1} = S_l - \{i_d\}$
Algorithm:
Step1: Start with an initial feasible \underline{d}_0 and the corresponding working set S_0 . Set $l = 0$. 1 Algorithm:
Step1: Start with
Step2: Solve for Step1: Start with an initial feasible \underline{d}_0 and the corresponding working set S_0
Step2: Solve for $\underline{p}_l \Rightarrow \underline{d}_{l+1}$
Step3: Find step length α_l . If $\alpha_l < 1$, append corresponding constraint i_a . So an initia
 $l_l \Rightarrow \underline{d}_l$, $S_{l+1} = S_l - \{l_d\}$
 \underline{d}_0 and the corresponding working set S_0 . Set *l* an initia
 $p_l \Rightarrow \underline{d}$ ial feasible \underline{d}_0 and the corresponding working set S_0 . Set $l = \underline{d}_{l+1}$
 α_l . If $\alpha_l < 1$, append corresponding constraint i_a . So \Rightarrow \underline{d}_{l+}

^{*l*+1}
i. If $\alpha_i < 1$, append corresponding constraint i_a *i* \lt

Step 2: Solve for
$$
\underline{p}_l \Rightarrow \underline{d}_{l+1}
$$

\nStep 3: Find step length α_l . If $\alpha_l < 1$, append correspondi
\n
$$
\tilde{A} = [\hat{A} \quad \hat{\underline{a}}] = Q \begin{pmatrix} R & Q^T \hat{\underline{a}} \\ 0 & 1 \end{pmatrix} = \overline{Q} \begin{bmatrix} \overline{R} \\ 0 \end{bmatrix}
$$
\n
$$
\tilde{Q} = QQ_l, \text{New } Q_l = [Q_l : \underline{q}_{m+r_l} + 1]
$$

New Q_l complete change

Return to Step 2 ; else go to Step 4

If $\mu_{i_d} \ge 0$ Stop

else drop constraint correspon
 $\tilde{A} = [\underline{a}_1 \ \underline{a}_2 ... \underline{a}_{i_d-1} \ \underline{a}_{i_d+1} ... \underline{a}_{m+\hat{r}}]$
 $Q^T \tilde{A} = \begin{bmatrix} M \\ 0 \end{bmatrix}$ **SQP Algorithm with 2nd Order Cor**
Algorithm (continued)
Step 4: If $\alpha_j = 1$, compute $\underline{\lambda}$ (last r components are $\underline{\mu}$) Find μ_i = 1, compute
Find μ_i = min(μ_i) If $\mu_{i_d} \geq 0$ Stop else drop constraint corresponding to i_d 0 upper triangular cols. 1 to $i_d - 1$. *d* λ_j = 1, compute λ (last *r* $= 1$, compu
 $i_{i_d} = \min_i (\mu_i)$ $T \tilde{A}$ \Box M else $\tilde{A} = \begin{bmatrix} \overline{A} & \overline{A} \\ \overline{Q}^T \tilde{A} & \overline{A} \end{bmatrix}$ $A = [\underline{a}_1 \ \ \underline{a}_2 ... \underline{a}_{i_d-1} \ \ \underline{a}_{i_d+1} ... \underline{a}_{m+\hat{r}}]$
 $Q^T \tilde{A} = \begin{bmatrix} M \\ 0 \end{bmatrix}$
 $M =$ upper triangular cols. 1 to $i_d - 1$. Has elemen
 i to $m + \hat{r} - 1$ $\frac{d_1(\text{continued})}{\mu_i} = 1, \text{ compute } \underline{\lambda} \text{ (last } r \text{ cor})$
 $\mu_{i_d} = \min_i(\mu_i)$ op constraint corre $\frac{a_2...a_{i_d-1}}{a_{i_d+1}...a_m}$
 $\begin{bmatrix} M \ 0 \end{bmatrix}$ $=$ $=$ $=$ $=$ $M =$ upper triangular cols
 i_d to $m + \hat{r} - 1$

New Q_l = [$\underline{q}_1 \underline{q}_2 ... \underline{q}_{i_d-1}, \hat{Q}$], Has elements in subdiagonals for columns i_d to $m + \hat{r} - 1$ i_d to $m + \hat{r} - 1$
New $Q_l = [q_1 q_2 ... q_{i_d}]$
New $\overline{Q}_l = [\hat{g}, \overline{Q}_{l \text{old}}]$ Go to Step 2 μ - <u>Lg</u>₁ \underline{q}_2 … \underline{q}_{i_d} *l* = $[\underline{q}_1 \ \underline{q}_2 ... \underline{q}_l]$
l = $[\hat{g}, \ \overline{Q}_l \]$ $Q^T \tilde{A} = \begin{bmatrix} M \\ 0 \end{bmatrix}$
 M = upper
 i_d to *m* + \hat{r} friangular col
-1
 $\frac{q_1 q_2 ... q_{i_d-1}, \hat{Q}}{\hat{Q}}$ $\frac{q_1}{q_2}$
 $\frac{q_2}{\hat{g}}, \overline{Q}$
 $\frac{2}{\hat{g}}, \overline{Q}$ $=[q_1 q_2 ... q_{i_d}]$ $\begin{bmatrix} M \\ 0 \end{bmatrix}$
per triangular cols.
+ \hat{r} - 1 New $\overline{Q}_i = [\hat{g}, \overline{Q}_{i \text{old}}]$ • Algorithm (continued) **SQP Algorithm with 2nd Order Correction - 8**

SQP Algorithm with 2nd Order Correction - 9

What if QPP is infeasible? Add artificial variables to detect it.

$$
\min \frac{1}{2} \underline{d}^T B_k \underline{d} + \underline{g}_k^T \underline{d} + C \underline{\xi}^T \underline{1}
$$
\n
\ns.t. $\nabla \underline{h}^T (\underline{x}_k) \underline{d}_k + \underline{h} (\underline{x}_k) = 0$ Always feasible\n
$$
\nabla \underline{g}^T (\underline{x}_k) \underline{d}_k + \underline{g} (\underline{x}_k) \le \underline{\xi}
$$
\n
$$
\underline{\xi} \ge 0
$$

Q Other Methods:

- M.JD Powell, "On the QP Algorithm of Goldfarb and Idnani", MP, 1985, pp.46-61
- Goldfarb and Idnani, "A numerically stable dual method for solving strictly quadratic programs convex"MP,1983,pp. 1-33

- Motivation for Successive Quadratic Programming (SQP) Methods
- **Example 3 Key SQP Ideas**
- **Newton Version of SQP**
- Descent Property of Merit Function *f+cP*
- Quasi-Newton Version of SQP
- \Box SQP with second order correction