# Computational Methods in Optimal Control
# Lecture 2. First-Order Optimality Conditions

William W. Hager

July 23, 2018

# Model Problem and Some Terminology

$$\begin{aligned} \text{minimize} \quad & C(\mathbf{x}(1)) \\ \text{subject to} \quad & \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad \mathbf{u}(t) \in \mathcal{U}, \quad t \in \Omega_0, \\ & \mathbf{x}(0) = \mathbf{x}_0 \quad (\mathbf{x}, \mathbf{u}) \in \mathcal{C}^1(\Omega_0;\ \mathbb{R}^n) \times \mathcal{C}^0(\Omega_0;\ \mathbb{R}^m), \end{aligned}$$

Some Notation:

- $\nabla C$ is the gradient vector, a row vector whose $i$-th element is
$$(\nabla C(\mathbf{x}))_i = \frac{\partial C(\mathbf{x})}{\partial x_i}.$$
- $\nabla_x \mathbf{f}$ is the Jacobian matrix, an $n$ by $n$ matrix whose $(i, j)$ element is
$$(\nabla_x \mathbf{f}(\mathbf{x}, \mathbf{u}))_{ij} = \frac{\partial f_i(\mathbf{x}, \mathbf{u})}{\partial x_j}.$$
- The Hamiltonian $H : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ associated with the control problem is $H(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}) = \boldsymbol{\lambda}\mathbf{f}(\mathbf{x}, \mathbf{u})$, where $\boldsymbol{\lambda} \in \mathbb{R}^n$ denotes a row vector.

# First-Order Optimality Conditions for Optimal Control

Assume $f$ and $C$ are continuously differentiable, and that the control problem has a solution $(\mathbf{x}^*, \mathbf{u}^*)$. Let $\boldsymbol{\lambda}^* : [0,1] \to \mathbb{R}^n$ denote the solution of the linear co-dynamics:

$$\dot{\boldsymbol{\lambda}}^*(t) = -\nabla_x H(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t)), \quad \boldsymbol{\lambda}^*(1) = \nabla C(\mathbf{x}^*(1)).$$

According to the Pontryagin minimum principle,

$$\mathbf{u}^*(t) = \arg\min\{H(\mathbf{x}^*(t), \mathbf{u}, \boldsymbol{\lambda}^*(t)) : \mathbf{u} \in \mathcal{U}\} \text{ for all } t \in \Omega_0.$$

The first-order optimality conditions consist of the co-dynamics, the minimum principle, and the dynamics:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad \mathbf{x}(0) = \mathbf{x}_0$$

# Reformulate the Minimum Principle

Suppose $F : \mathbb{R}^m \to \mathbb{R}$ and $\mathbf{u}^* \in \mathcal{U}$ with

$$F(\mathbf{u}^*) = \text{arg min } \{F(\mathbf{u}) : \mathbf{u} \in \mathcal{U}\}.$$

Since $\mathcal{U}$ is convex, $\mathbf{u}^* + s(\mathbf{u} - \mathbf{u}^*) \in \mathcal{U}$ for all $s \in [0, 1]$. Hence, we have

$$F(\mathbf{u}^* + s(\mathbf{u} - \mathbf{u}^*)) \geq F(\mathbf{u}^*) \quad \text{for all } \alpha \in [0, 1].$$

This implies

$$\frac{F(\mathbf{u}^* + s(\mathbf{u} - \mathbf{u}^*)) - F(\mathbf{u}^*)}{s} \geq 0 \quad \text{for all } \alpha \in [0, 1].$$

If $F$ is differentiable at $\mathbf{u}^*$, then letting $s \to 0^+$ gives:

$$\nabla F(\mathbf{u}^*)(\mathbf{u} - \mathbf{u}^*) \geq 0 \quad \text{for all } \mathbf{u} \in \mathcal{U}.$$

# Normal Cone

For any $\mathbf{u} \in \mathcal{U}$, the normal cone $N_{\mathcal{U}}$ is defined by

$$N_{\mathcal{U}}(\mathbf{u}) = \begin{cases} \{\mathbf{w} \in \mathbb{R}^m : \mathbf{w}^\mathsf{T}(\mathbf{v} - \mathbf{u}) \leq 0 \text{ for all } \mathbf{v} \in \mathcal{U}\} & \text{if } \mathbf{u} \in \mathcal{U} \\ \emptyset \text{ if } \mathbf{u} \notin \mathcal{U}. \end{cases}$$

Thus the following relations are equivalent:

$$\begin{aligned} \nabla F(\mathbf{u}^*)(\mathbf{v} - \mathbf{u}^*) &\geq 0 \text{ for all } \mathbf{v} \in \mathcal{U} \\ -\nabla F(\mathbf{u}^*) &\in N_{\mathcal{U}}(\mathbf{u}^*) \end{aligned}$$

Moreover, the minimum principle implies that

$$-\nabla_u H(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t)) \in N_{\mathcal{U}}(\mathbf{u}^*(t)) \quad \text{for all } t \in \Omega_0,$$

In summary, the first-order optimality conditions can be stated:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \quad \mathbf{u}(t) \in \mathcal{U}, \quad \mathbf{x}(0) = \mathbf{x}_0 \\ \dot{\boldsymbol{\lambda}}^*(t) &= -\nabla_x H(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t)), \quad \boldsymbol{\lambda}^*(1) = \nabla C(\mathbf{x}^*(1)) \\ N_{\mathcal{U}}(\mathbf{u}^*(t)) &\ni -\nabla_u H(\mathbf{x}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t)) \quad \text{for all } t \in \Omega_0 \end{aligned}$$

# Discretization

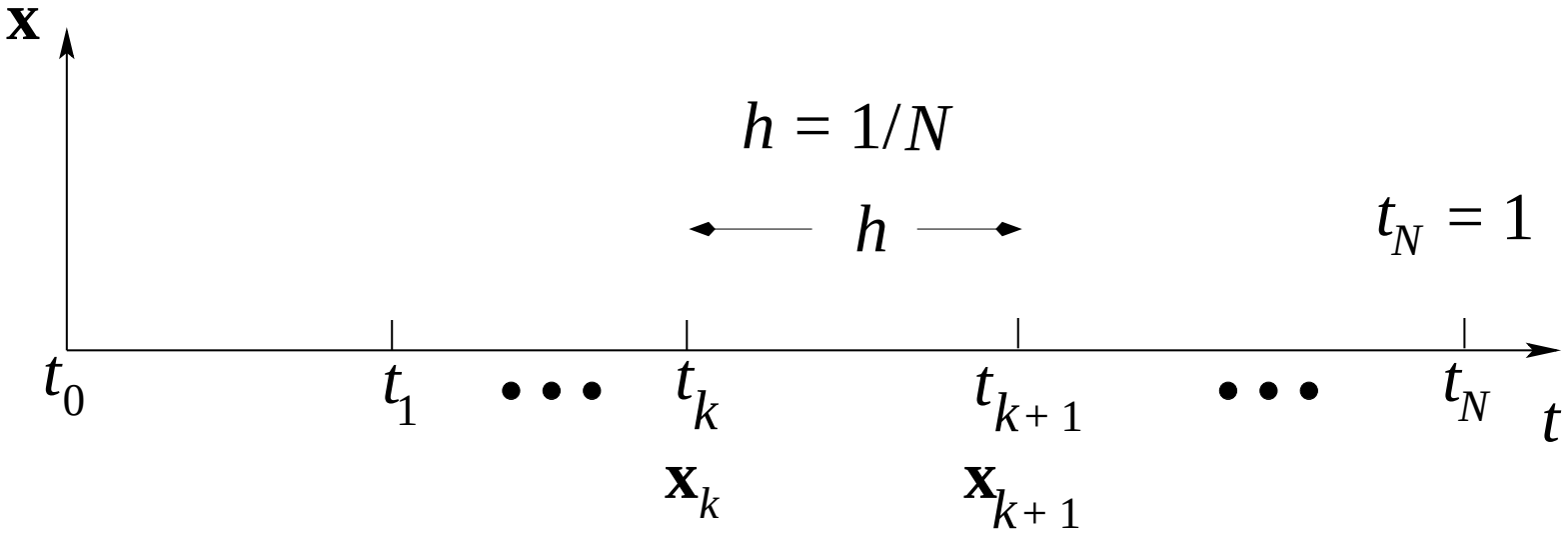
Continuous problem:

$$\min\ C(\mathbf{x}(1)) \text{ subject to } \begin{cases} \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \\ \mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{u}(t) \in \mathcal{U}, \quad t \in \Omega_0 \end{cases}$$

Notation:

$$\dot{\mathbf{x}}_k := \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{h}$$

Discretization:



# $s$-stage Runge-Kutta discretization

$$\min \quad C(\mathbf{x}_N)$$

$$\text{subject to} \quad \mathbf{y}_i = \mathbf{x}_k + h\sum_{j=1}^{s} a_{ij}\mathbf{f}(\mathbf{y}_j, \mathbf{u}_{kj}), \quad i = 1, \ldots, s$$

$$\dot{\mathbf{x}}_k = \sum_{i=1}^{s} b_i \mathbf{f}(\mathbf{y}_i, \mathbf{u}_{ki}), \quad \mathbf{u}_{ki} \in \mathcal{U}$$

Euler's Improved Method:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 \\ 1/2 & 0 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{aligned} \mathbf{y}_1 &= \mathbf{x}_k \\ \mathbf{y}_2 &= \mathbf{x}_k + (h/2)\mathbf{f}(\mathbf{y}_1, \mathbf{u}_{k1}) = \mathbf{x}_{k+1/2} \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + h\mathbf{f}(\mathbf{y}_2, \mathbf{u}_{k2}) \qquad = \mathbf{x}_k + h\mathbf{f}(\mathbf{x}_{k+1/2}, \mathbf{u}_{k+1/2}) \end{aligned}$$

# Explicit and Implicit Schemes

- If $\mathbf{A}$ is strictly lower triangular, then the scheme is explicit since $\mathbf{y}_i$ is explicitly expressed in terms of $\mathbf{y}_1$, $\mathbf{y}_2$, $\ldots$, $\mathbf{y}_{i-1}$.
- If $\mathbf{A}$ is not strictly lower triangular, then the scheme is implicit. In this case, we generally need to solve a system of $s$ equations in the $s$ unknowns $\mathbf{y}_i$, $1 \leq i \leq s$, to obtain the $\mathbf{y}_i$.
- If $h$ is sufficiently small and $\mathbf{f}(\mathbf{x}, \mathbf{u})$ is Lipschitz continuous in $\mathbf{x}$, then $\mathbf{y}$ in an implicit scheme is often computed using the iteration

$$\mathbf{y}_i^{\text{new}} = \mathbf{x}_k + h \sum_{j=1}^{s} a_{ij} \mathbf{f}(\mathbf{y}_j^{\text{old}}, \mathbf{u}_{kj}),$$

  where initially, we take $\mathbf{y}_i = \mathbf{x}_k$ for $1 \leq i \leq s$. This is convergent by the contraction mapping principle.

# Lagrangian for the Discrete Problem

Let $\chi_{ki} \in \mathbb{R}^n$ denote the Lagrange multiplier (row vector) associated with the equation for $\mathbf{y}_i$, $1 \leq i \leq s$, and let $\psi_{k+1}$ denote the multiplier associated with the equation for $\dot{\mathbf{x}}_k$. Let us define $L_k(\mathbf{x}, \mathbf{y}, \mathbf{u}, \psi, \chi) =$

$$\sum_{i=1}^{s} \left( b_i \psi_{k+1} \mathbf{f}(\mathbf{y}_{ki}, \mathbf{u}_{ki}) + \chi_{ki} \left[ \mathbf{x}_k - \mathbf{y}_{ki} + h \sum_{j=1}^{s} a_{ij} \mathbf{f}(\mathbf{y}_{kj}, \mathbf{u}_{kj}) \right] \right)$$

The Lagrangian $L$ for the discrete problem is

$$L(\mathbf{x}, \mathbf{y}, \mathbf{u}, \psi, \chi) = C(\mathbf{x}_N) + \sum_{k=0}^{N-1} L_k(\mathbf{x}, \mathbf{y}, \mathbf{u}, \psi, \chi) - \psi_{k+1} \dot{\mathbf{x}}_k.$$

Suppose that $(\mathbf{x}, \mathbf{y}, \mathbf{u})$ is a local minimizer for the discrete control problem, and a constraint regularity condition holds. Then the derivative of the Lagrangian with respect to $\mathbf{x}$ and $\mathbf{y}$ vanish, and the negative derivative with respect to $\mathbf{u}$ lies in the normal cone $N_{\mathcal{U}}$.

# First-order Optimality Conditions

- Derivative of $L$ with respect to $\mathbf{x}_k$ yields:

$$\psi_k - \psi_{k+1} = \sum_{i=1}^{s} \chi_{ki}, \quad \psi_N = \nabla C(\mathbf{x}_N), \tag{D1}$$

- Derivative of $L$ with respect to $\mathbf{y}_{ki}$ yields

$$h\left(b_j \psi_{k+1} + \sum_{i=1}^{s} a_{ij} \chi_{ki}\right) \nabla_x \mathbf{f}(\mathbf{y}_{kj}, \mathbf{u}_{kj}) = \chi_{kj}, \tag{D2}$$

- Derivative of $L$ with respect to $\mathbf{u}_{kj}$ yields

$$-\left(b_j \psi_{k+1} + \sum_{i=1}^{s} a_{ij} \chi_{ki}\right) \nabla_u \mathbf{f}(\mathbf{y}_{kj}, \mathbf{u}_{kj}) \in N_{\mathcal{U}}(\mathbf{u}_{kj}) \tag{D3}$$

Here $1 \leq j \leq s$ and $0 \leq k \leq N-1$.

# Connect with Continuous Optimality Conditions

Suppose that $b_j > 0$ and define

$$\boldsymbol{\lambda}_{kj} = \boldsymbol{\psi}_{k+1} + \sum_{i=1}^{s} \frac{a_{ij}}{b_j} \boldsymbol{\chi}_{ki}, \quad 1 \leq j \leq s.$$

Hence, (D3) is equivalent to

$$-\boldsymbol{\lambda}_{kj} \nabla_u \mathbf{f}(\mathbf{y}_{kj}, \mathbf{u}_{kj}) \in N_{\mathcal{U}}(\mathbf{u}_{kj})$$

Moreover, (D2) becomes

$$h b_j \boldsymbol{\lambda}_{kj} \nabla_x \mathbf{f}(\mathbf{y}_{kj}, \mathbf{u}_{kj}) = \boldsymbol{\chi}_{kj}. \qquad (\dagger)$$

# Continue . . .

Multiply (†) by $a_{ji}/b_i$ and sum over $j$ to obtain

$$h\sum_{j=1}^{s}\frac{b_j a_{ji}}{b_i}\boldsymbol{\lambda}_{kj}\nabla_x\mathbf{f}(\mathbf{y}_{kj},\mathbf{u}_{kj}) = \sum_{j=1}^{s}\frac{a_{ji}}{b_i}\boldsymbol{\chi}_{kj} = \boldsymbol{\lambda}_{ki} - \boldsymbol{\psi}_{k+1}. \qquad (\ddagger)$$

Sum (†) over $j$ and combine with (D1) to obtain

$$\boldsymbol{\psi}_{k+1} - \boldsymbol{\psi}_k = -\sum_{j=1}^{s} b_j\boldsymbol{\chi}_j\nabla_x\mathbf{f}(\mathbf{y}_j,\mathbf{u}_{kj}).$$

Substitute this in (‡) yields

$$\boldsymbol{\chi}_i = \boldsymbol{\psi}_k - h\sum_{j=1}^{s}\bar{a}_{ij}\boldsymbol{\chi}_j\nabla_x\mathbf{f}(\mathbf{y}_j,\mathbf{u}_{kj}), \quad \bar{a}_{ij} = \frac{b_i b_j - b_j a_{ji}}{b_i}.$$

# Transformed Discrete Optimality Conditions

$$
\begin{aligned}
\mathbf{y}_{ki} &= \mathbf{x}_k + h\sum_{j=1}^{s} a_{ij}\mathbf{f}(\mathbf{y}_{kj}, \mathbf{u}_{kj}), \\
\dot{\mathbf{x}}_k &= \sum_{i=1}^{s} b_i \mathbf{f}(\mathbf{y}_{ki}, \mathbf{u}_{ki}), \quad \mathbf{x}_0 \text{ given} \\
\boldsymbol{\lambda}_{ki} &= \boldsymbol{\psi}_k - h\sum_{j=1}^{s} \bar{a}_{ij}\boldsymbol{\lambda}_{kj}\nabla_x \mathbf{f}(\mathbf{y}_{kj}, \mathbf{u}_{kj}), \quad \bar{a}_{ij} = \frac{b_i b_j - b_j a_{ji}}{b_i}, \\
\dot{\boldsymbol{\psi}}_k &= -\sum_{i=1}^{s} b_i \boldsymbol{\lambda}_{ki}\nabla_x \mathbf{f}(\mathbf{y}_{ki}, \mathbf{u}_{ki}), \quad \boldsymbol{\psi}_N = \nabla C(\mathbf{x}_N), \\
N_{\mathcal{U}}(\mathbf{u}_{ki}) &\ni -\boldsymbol{\lambda}_{ki}\nabla_u \mathbf{f}(\mathbf{y}_{ki}, \mathbf{u}_{ki})
\end{aligned}
$$

$1 \le i \le s$ and $0 \le k \le N-1$.

# Computing Derivatives in Direct Transcription

The manipulations used to derive the transformed discrete optimality conditions also yield a systematic way for computing the objective derivative as we now show. Consider the finite dimensional problem

$$\min\ g(\mathbf{x}, \mathbf{u}) \text{ subject to } \mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{0},$$

where $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$. Suppose that for each $\mathbf{u} \in \mathbb{R}^m$, the equation $\mathbf{f}(\mathbf{x}, \mathbf{u}) = \mathbf{0}$ ($n$ equations in $n$ unknowns) has a unique solution $\mathbf{x} = \mathbf{x}(\mathbf{u})$. Substituting $\mathbf{x} = \mathbf{x}(\mathbf{u})$ in the objective yields a real-valued function of $\mathbf{u}$ given by

$$G(\mathbf{u}) = g(\mathbf{x}(\mathbf{u}), \mathbf{u}).$$

# Chain Rule

We compute the gradient of the objective with respect to $\mathbf{u}$ by the chain rule, assuming both $f$ and $g$ are smooth:

$$\nabla G(\mathbf{u}) = \nabla_u g(\mathbf{x}(\mathbf{u}), \mathbf{u}) + \nabla_x g(\mathbf{x}(\mathbf{u}), \mathbf{u}) \nabla \mathbf{x}(\mathbf{u})$$

Here $\nabla_x g$ and $\nabla_u g$ are both row vectors and $\nabla \mathbf{x}(\mathbf{u})$ is the $n$ by $n$ Jacobian matrix. Next, differentiate the identity $\mathbf{f}(\mathbf{x}(\mathbf{u}), \mathbf{u}) = \mathbf{0}$

$$\nabla_x \mathbf{f}(\mathbf{x}(\mathbf{u}), \mathbf{u}) \nabla \mathbf{x}(\mathbf{u}) + \nabla_u \mathbf{f}(\mathbf{x}(\mathbf{u}), \mathbf{u}) = \mathbf{0}.$$

If $\nabla_x \mathbf{f}(\mathbf{x}(\mathbf{u}), \mathbf{u})$ is an invertible matrix, then

$$\nabla \mathbf{x}(\mathbf{u}) = -\nabla_x \mathbf{f}(\mathbf{x}(\mathbf{u}), \mathbf{u})^{-1} \nabla_u \mathbf{f}(\mathbf{x}(\mathbf{u}), \mathbf{u}).$$

Hence, we have

$$\nabla G(\mathbf{u}) = \nabla_u g(\mathbf{x}(\mathbf{u}), \mathbf{u}) - \nabla_x g(\mathbf{x}(\mathbf{u}), \mathbf{u}) \nabla_x \mathbf{f}(\mathbf{x}(\mathbf{u}), \mathbf{u})^{-1} \nabla_u \mathbf{f}(\mathbf{x}(\mathbf{u}), \mathbf{u})$$

# Connect to the Lagrangian

The Lagrangian of the optimization problem is

$$L(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}) = g(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}\mathbf{f}(\mathbf{x}, \mathbf{u}),$$

where $\boldsymbol{\lambda} \in \mathbb{R}^n$ is a row vector. Let us set the derivative of the Lagrangian with respect to $\mathbf{x}$ to zero:

$$\mathbf{0} = \nabla_x L(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}) = \nabla_x g(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}\nabla_x \mathbf{f}(\mathbf{x}, \mathbf{u})$$

Multiplying on the right by $\nabla_x \mathbf{f}(\mathbf{x}, \mathbf{u})^{-1}$ gives

$$\boldsymbol{\lambda} = -\nabla_x g(\mathbf{x}, \mathbf{u})\nabla_x \mathbf{f}(\mathbf{x}, \mathbf{u})^{-1}.$$

THUS

$$\nabla G(\mathbf{u}) = \nabla_u L(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}) = \nabla_u g(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}\nabla_u \mathbf{f}(\mathbf{x}, \mathbf{u})$$

evaluated at $\mathbf{x} = \mathbf{x}(\mathbf{u})$ where $\boldsymbol{\lambda}$ is chosen so that $\nabla_x L(\mathbf{x}, \mathbf{u}, \boldsymbol{\lambda}) = \mathbf{0}$.

# Return to the Control Problem

If $G(\mathbf{u})$ denotes the objective of the control problem, then

$$
\begin{aligned}
[\nabla G(\mathbf{u})]_{k\ell} &= b_\ell \psi_{k+1} \nabla_u \mathbf{f}(\mathbf{y}_{k\ell}, \mathbf{u}_{k\ell}) + h \sum_{i=1}^{s} \chi_{ki} a_{i\ell} \nabla_u \mathbf{f}(\mathbf{y}_{k\ell}, \mathbf{u}_{k\ell}) \\
&= \left( b_\ell \psi_{k+1} + h \sum_{i=1}^{s} \chi_{ki} a_{i\ell} \right) \nabla_u \mathbf{f}(\mathbf{y}_{k\ell}, \mathbf{u}_{k\ell}) \\
&= b_\ell \left( \psi_{k+1} + h \sum_{i=1}^{s} \chi_{ki} \frac{a_{i\ell}}{b_\ell} \right) \nabla_u \mathbf{f}(\mathbf{y}_{k\ell}, \mathbf{u}_{k\ell})
\end{aligned}
$$

Hence, we have

$$
[\nabla G(\mathbf{u})]_{k\ell} = b_\ell \boldsymbol{\lambda}_{k\ell} \nabla_u \mathbf{f}(\mathbf{y}_{k\ell}, \mathbf{u}_{k\ell}) \qquad (\nabla)
$$

# Computation of State and Multipliers in $(\nabla)$

Sweep forward to compute $\mathbf{x}$ and $\mathbf{y}$. Sweep backwards to compute $\psi$ and $\boldsymbol{\lambda}$.

$$
\begin{aligned}
\mathbf{y}_{ki} &= \mathbf{x}_k + h\sum_{j=1}^{s} a_{ij}\mathbf{f}(\mathbf{y}_{kj}, \mathbf{u}_{kj}), \\
\dot{\mathbf{x}}_k &= \sum_{i=1}^{s} b_i \mathbf{f}(\mathbf{y}_{ki}, \mathbf{u}_{ki}), \quad \mathbf{x}_0 \text{ given} \\
\boldsymbol{\lambda}_{ki} &= \psi_{k+1} + h\sum_{j=1}^{s}\left(\frac{b_j a_{ji}}{b_i}\right)\boldsymbol{\lambda}_{kj}\nabla_x \mathbf{f}(\mathbf{y}_{kj}, \mathbf{u}_{kj}), \\
\dot{\psi}_k &= -\sum_{i=1}^{s} b_i \boldsymbol{\lambda}_{ki}\nabla_x \mathbf{f}(\mathbf{y}_{ki}, \mathbf{u}_{ki}), \quad \psi_N = \nabla C(\mathbf{x}_N),
\end{aligned}
$$

$1 \le i \le s$ and $0 \le k \le N-1$.

# A Comment

This technique is the basis for reverse mode automatic differentiation techniques. A computer code provides a step by step process for computing an output number from its inputs. A Lagrange multiplier is associated with each step in the calculation. The derivative is obtained by first evaluating the function in a forward sweep, and then evaluating the multipliers in a backward sweep.