# ODE and DDE in PODE

Martin Fink
Pharmacological Modeler
Modeling & Simulation, Novartis

# Overview
*ODEs and DDEs*

- Introduction
- Sensitivities & (P)FIM
- Deriving sensitivities from ODEs & DDEs
  - Comparison of various methods
  - Speed and numerical robustness
- Graphical displays of information
  - Sensitivities & Generalized sensitivities


NOVARTIS

# Introduction

*ODEs and DDEs*

- Models with closed-form solutions often intuitive enough
  - PK sampling time points from clinical pharmacologist fine
- Non-linear PK and PK/PD models need design information
  - Michaelis-Menten elimination
  - Binding model and its approximations
  - Indirect response models
  - Lifespan models
  - …
- More complex models: ODEs or DDEs

# Derivation of (P)FIM starts with sensitivities

*Sensitivities of solution w.r.t. parameters*

- For approximation of the (P)FIM sensitivities are necessary

$$M_F(\Psi, \xi) \cong \begin{bmatrix} \frac{\partial f^{T}(\beta, \xi)}{\partial \beta} V^{-1} \frac{\partial f(\beta, \xi)}{\partial \beta} & 0 \\ 0 & \frac{1}{2}F \end{bmatrix}$$

Retout et al., 2001

$$\text{where } F_{jk} = \mathrm{tr}\left( V^{-1} \frac{\partial V}{\partial \lambda_j} V^{-1} \frac{\partial V}{\partial \lambda_k} \right)$$

$$V \cong \frac{\partial f^{T}(\beta, \xi)}{\partial \beta} \Omega \frac{\partial f(\beta, \xi)}{\partial \beta} + \sigma^2 I_n$$

- PFIM block of fixed effects and variance terms are derived using the sensitivities



NOVARTIS

# Definition of Sensitivities

*Partial derivatives of solution – possibly normalized*

- Sensitivities: derivatives of solution w.r.t. parameters

The sensitivity of a state $x_k$ to a parameter $p_j$ is defined as the derivative

$$\frac{dx_k}{dp_j}(p, q, \ldots) = \frac{\partial x_k}{\partial p_j} + \sum_{i \neq j} \frac{\partial x_k}{\partial p_i} \frac{\partial p_i}{\partial p_j} + \sum_i \frac{\partial x_k}{\partial q_i} \frac{\partial q_i}{\partial p_j} + \ldots.$$

The sensitivity can be normalized by the parameter and/or the state variable

$$\frac{dx_k(t)}{dp_j} \frac{p_j}{x_k(t)}.$$

- Different methods are available to derive these sensitivities
- Computationally most costly part when using ODEs/DDEs



NOVARTIS

# Sensitivities: Numerical Derivatives of Results

*Similar for analytical functions and ODEs/DDEs*

- Numerical approximations using finite differences
  - Higher order finite difference equations
    - 2nd or 3rd order central or forward/backward finite differences
      - No error control
    - *fdHess* uses a second-order response surface design known as a Koschal design – then gradient is by-product
      - High precision, but needs resources for deriving Hessian
  - Iterative approach with finite differences
    - *jacobian* using Richardson's extrapolation
      - High accuracy with little overhead

*Function names from "R"*



NOVARTIS

# Sensitivities: By Integration of Partial Derivatives

*Extend right-hand side in integration partial derivatives*

- By changing the order of integral and differential one can derive the sensitivities within the integration routine

For ODEs this gives in detail

$$\frac{dx_k}{dp_j}(\mathbf{x}, \mathbf{p}, t) = \int \frac{\partial}{\partial p_j} \frac{\partial x_k}{\partial t} + \sum_i \frac{\partial}{\partial x_i} \frac{\partial x_k}{\partial t} \cdot \frac{\partial x_i}{\partial p_j} dt.$$

**variables for integral**

partial deriv to param

**RHS of ODEs**

partial deriv to states

**variables for integral**

- Partial derivatives of the right-hand side (RHS) are needed
  - Can be specified manually or derived using automatic differentiation



NOVARTIS

# Sensitivities: By Integration of Partial Derivatives

*Similarly for DDEs*

- DDEs can be handled similarly – except for the delay itself

For ODEs this gives in detail

$$\frac{dx_k}{dp_j}(\mathbf{x}, \mathbf{p}, t) = \int \frac{\partial}{\partial p_j} \frac{\partial x_k}{\partial t} + \sum_i \frac{\partial}{\partial x_i} \frac{\partial x_k}{\partial t} \cdot \frac{\partial x_i}{\partial p_j} \, dt.$$

Whereas for DDEs with delay $\tau$ the sensitivity can be calculated as follows

$$\frac{dx_k}{dp_j}(\mathbf{x}, \mathbf{p}, t) = \int \frac{\partial}{\partial p_j} \frac{\partial x_k}{\partial t} + \sum_i \frac{\partial}{\partial x_i} \frac{\partial x_k}{\partial t} \cdot \frac{\partial x_i}{\partial p_j} + \sum_i \frac{\partial}{\partial x_i^\tau} \frac{\partial x_k}{\partial t} \cdot \frac{\partial x_i^\tau}{\partial p_j} \, dt.$$

- Colleagues at NCSU & Graz University tried to come up with a general form for the sensitivity to the delay…
  - to no avail

# Automatic differentiation

*No package available on CRAN*

- Automatic differentiation comes in different flavours
  - Deriving the derivative “on-the-fly” with operator-overloading
  - Parsing the code and providing the derivative as a function
- Matlab has several packages
  - myAD (M Fink, 2005)
    - http://www.mathworks.ch/matlabcentral/fileexchange/15235-automatic-differentiation-for-matlab
  - adiff (W McIlhagga, 2010)
    - http://www.mathworks.ch/matlabcentral/fileexchange/26807-automatic-differentiation-with-matlab-objects
  - MAD (TOMLAB) – commercial software
- Now, preliminary implementation in R as S4-class



NOVARTIS

# Comparison of methods for ODEs

*Various methods available in R*

- Analytical solution (if available)
- Numerical derivative of solution
  - fdHess
  - Jacobian
- Simultaneous integration using partial differentials
  - User provided partial differentials
  - Automatic differentiation
  - Numerical differentiation
- Optimal solution would be user specified model in C/C++



NOVARTIS

# Comparison of methods

*2 examples to investigate accuracy and speed*

- Example 1: $dy = -k_e * y$
  - Analytical solution known
  - Investigate both ODE & analytical solution
- Example 2: PK/PD
  - Example 1 coupled with turnover model (stimulating kout)
  - Only keeping sensitivities of PD
  - 5 parameters



NOVARTIS

# Comparison of methods

*High accuracy: automatic differentiation, speed: iterative jacobian*

| | Analytic | ODE User | ODE myRAD | Hessian Solution | Jacobian Solution |
|---|---|---|---|---|---|
| Ex 1 accuracy | Ref | 1.5e-5 | 1.5e-5 | 3.5e-5 | 3.3e-5 |
| Ex 1 time | <0.01 | 0.19 | 34.25 | 15.97 | 1.37 |
| Ex 2 accuracy | | Ref | as Ref | 1.1e-5 | 6.0e-6 |
| Ex 2 time | | 0.56 | 107.70 | 416.67 | 35.13 |

- Solving ODE with jacobian instead of myRAD does not work
- Time given in sec for 10 runs
- Accuracy given as mean diff from analytic or user supplied solution

# Comparison of methods

*How much accuracy is necessary?*

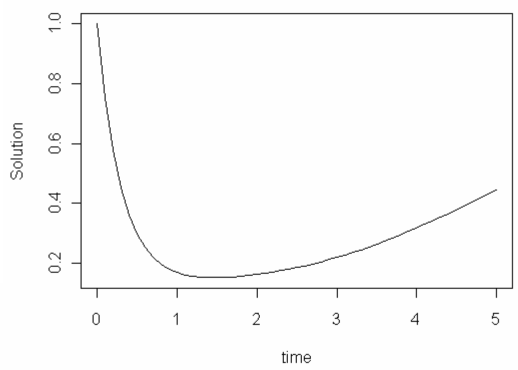
- How much influence does accuracy of sensitivities have on PFIM?
- How accurate are the standard error-estimates?
- Important to investigate numerical error propagation

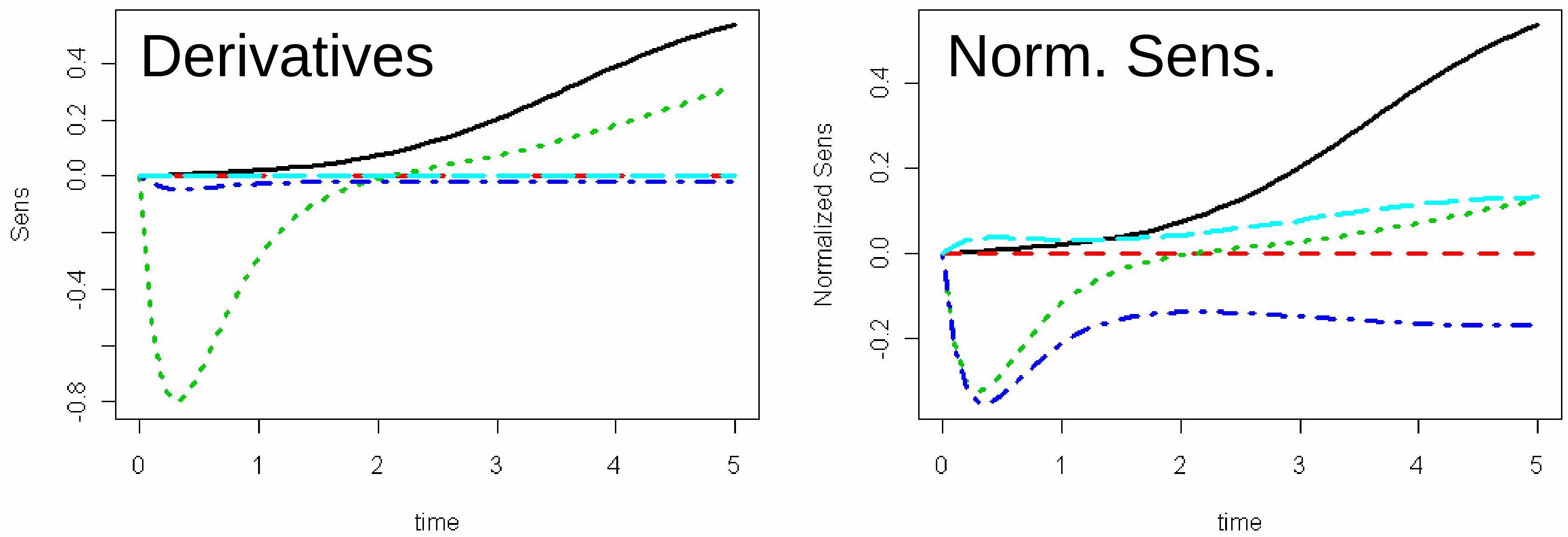


NOVARTIS

# Graphical display of information
*Plotting normalized sensitivities*

- Important to plot normalized sensitivities!
  - Proportional changes important






NOVARTIS

# Generalized sensitivity function (GSF)

*K Thomaseth & C Cobelli. Ann Biomed Eng 27:607-616, 1999*

- Display information increase over time - normalized to [0, 1]
  - No variance-covariance included (possible to extend to PFIM)

$$y(t_k) = f(t_k, \boldsymbol{\theta}) + e(t_k), \quad k = 1, \ldots, M, \qquad (1)$$

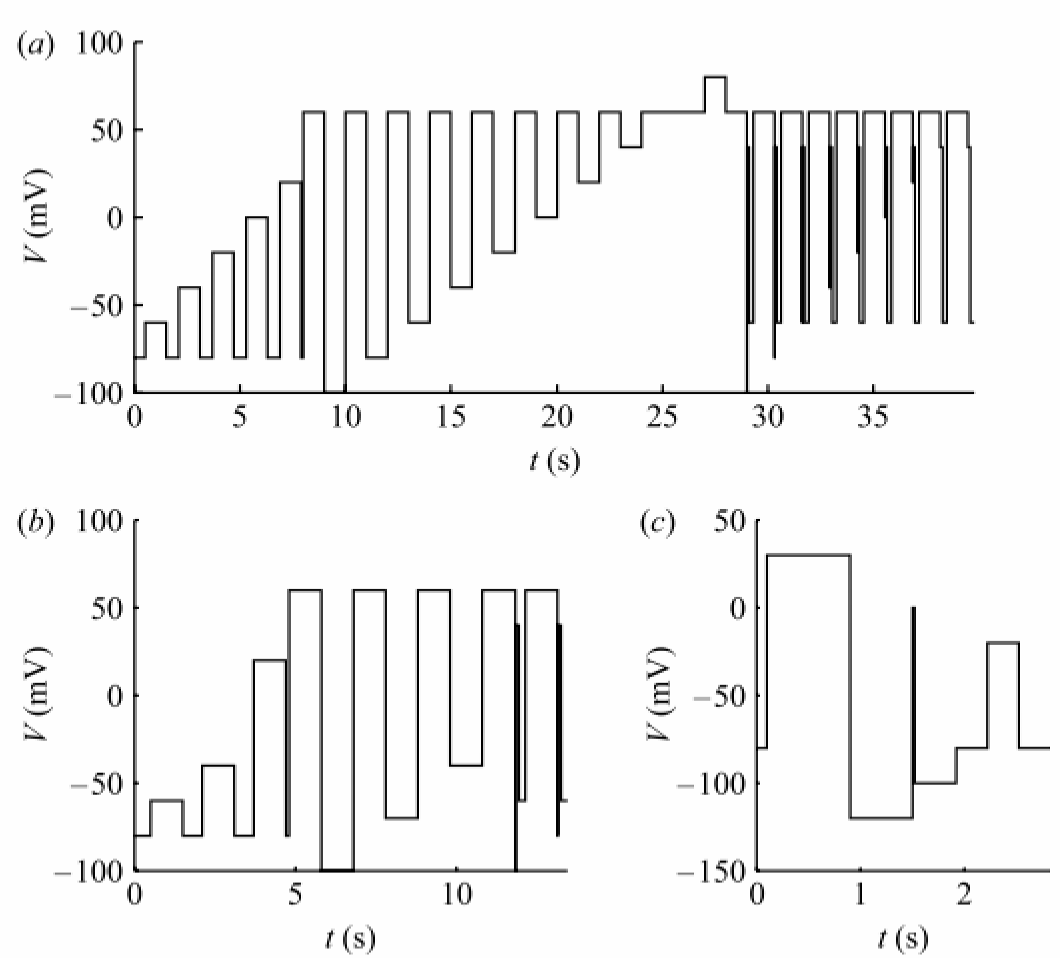
... motivates the introduction of GSF defined at the time points $\{t_k, k = 1, \ldots, M\}$, $\mathbf{gs}(t_k)$, that show how the effect of variations in the true parameters on their estimates distributes during the experiment

$$\mathbf{gs}(t_k) = \sum_{i=1}^{k} \left\{ \left( \left[ \sum_{j=1}^{M} \frac{1}{\sigma^2(t_j)} \nabla_{\boldsymbol{\theta}} f(t_j, \boldsymbol{\theta}_0) \nabla_{\boldsymbol{\theta}} f(t_j, \boldsymbol{\theta}_0)' \right]^{-1} \times \frac{\nabla_{\boldsymbol{\theta}} f(t_i, \boldsymbol{\theta}_0)}{\sigma^2(t_i)} \right) \bullet \nabla_{\boldsymbol{\theta}} f(t_i, \boldsymbol{\theta}_0) \right\}. \qquad (17)$$

# Generalized Sensitivity: Cardiac electrophysiology

*General experimental setup very elaborate – which parts to select?*

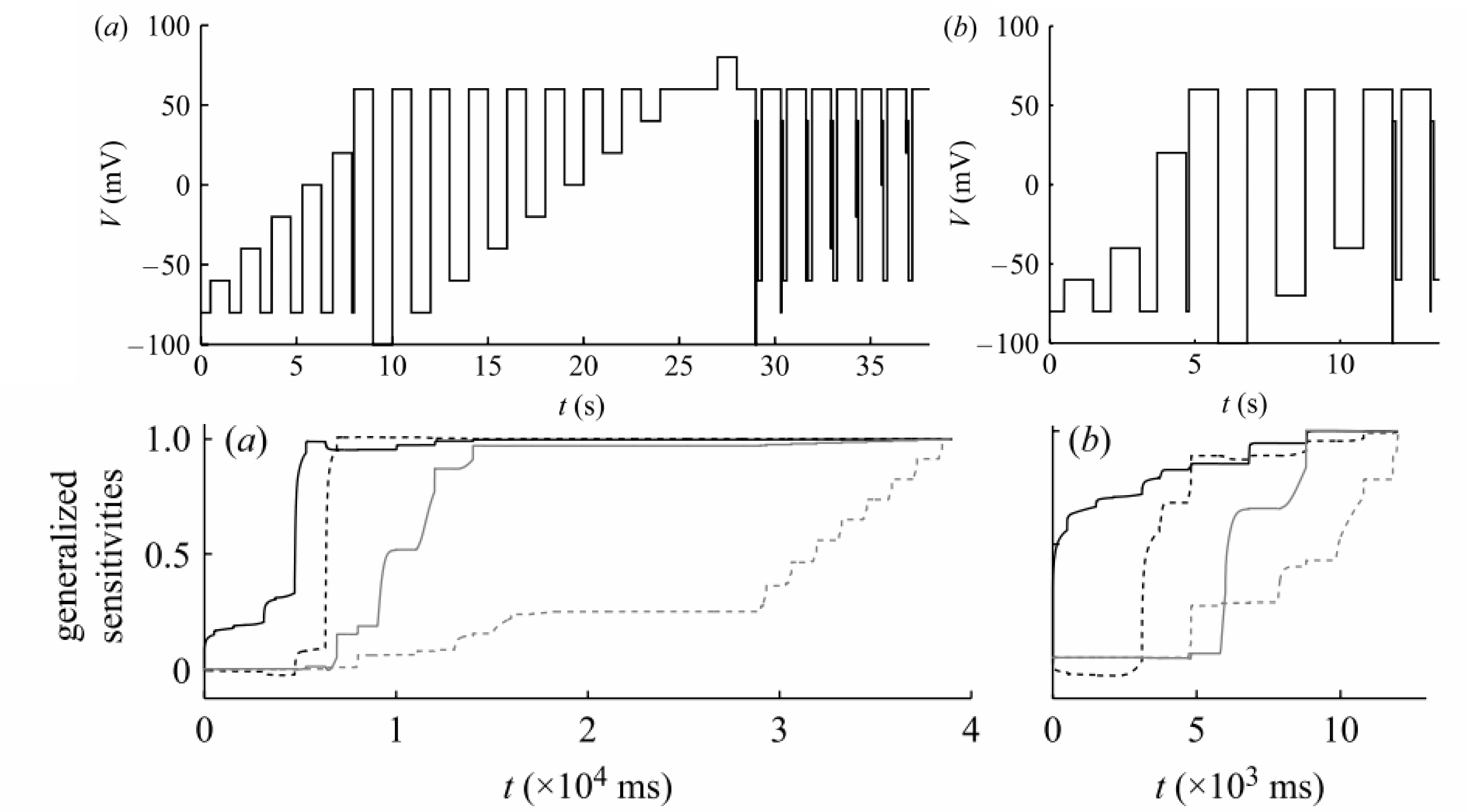
- Ion channel measurements using various protocols
  - All collated in (a)
  - Clamping the cross-membrane voltage while measuring the current
  - A lot of redundant information collected – can one skip parts? (b) or even (c)?




Fink & Noble, Phil Trans R Soc A, 2009.

# Generalized Sensitivity: Cardiac electrophysiology

*Protocol (b) was designed using the generalized sensitivities*

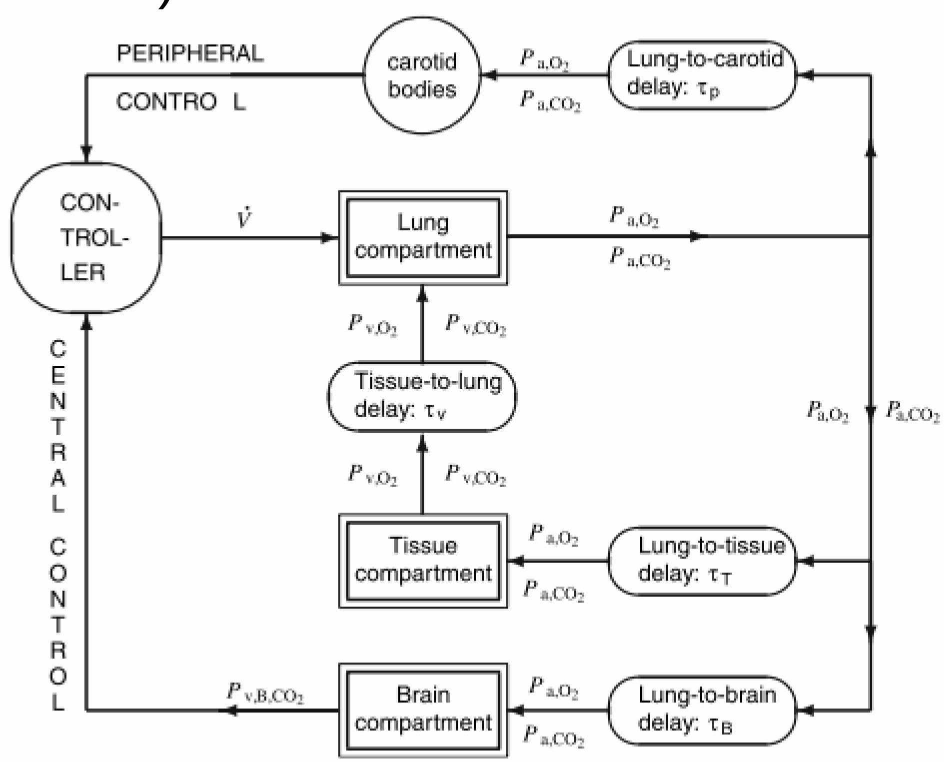



Fink & Noble, Phil Trans R Soc A, 2009.

# Generalized Sensitivity: Respiratory system

*A complex system with many parameters*

- Two main influences/controls of respiration
  - Central control (brain)
  - Peripheral control (carotid arteries)
- Several measurements
  - Partial pressures $CO_2/O_2$
    - Transcutaneous
    - Arterial
  - Ventilation
- Clinical: $CO_2$ challenge

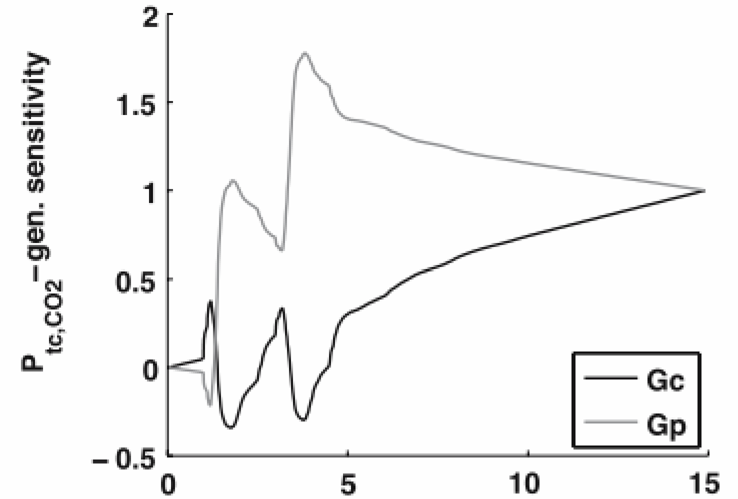


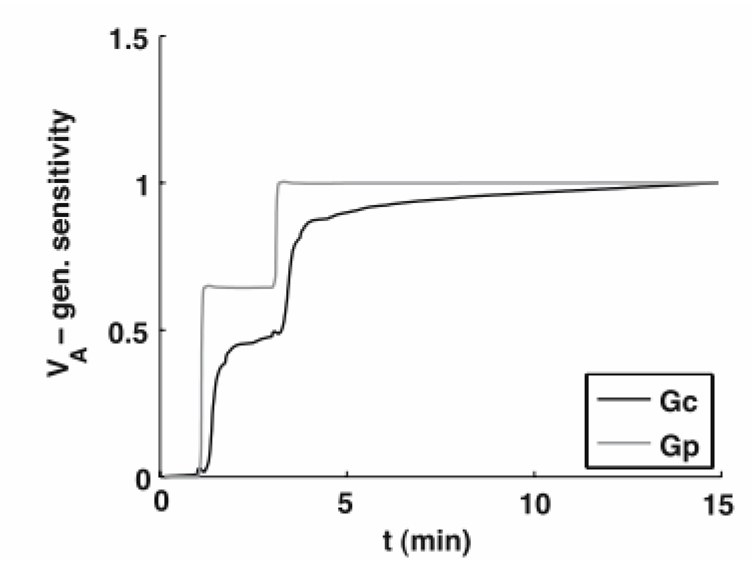

Fink et al., Cardiovasc Eng, 2007.

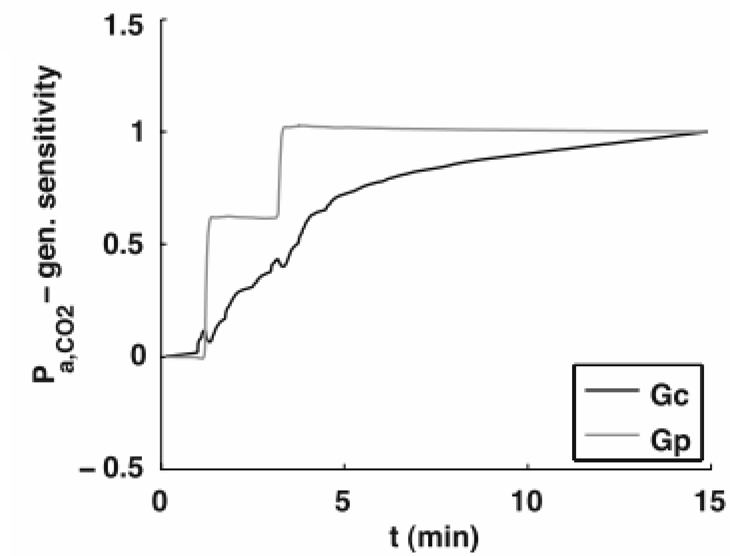
# Generalized Sensitivity: Respiratory system

*A complex system with many parameters*

- Measurements
  - Transcutaneous was sub-optimal Information interdependent
  - Arterial measurements better
  - Ventilation best as non-invasive




Fink et al., Cardiovasc Eng, 2007.



NOVARTIS

# Summary

*Sensitivities are essential to derive and understand*

- Automatic differentiation is great but a bit slow
- Iterative numerical derivatives seem perfect
- Open question
  - How much influence does accuracy of sensitivities have on PFIM?
- Plotting normalized sensitivities for user information
- Generalized sensitivities
  - Great concept and has been very helpful
  - Still some properties not well understood