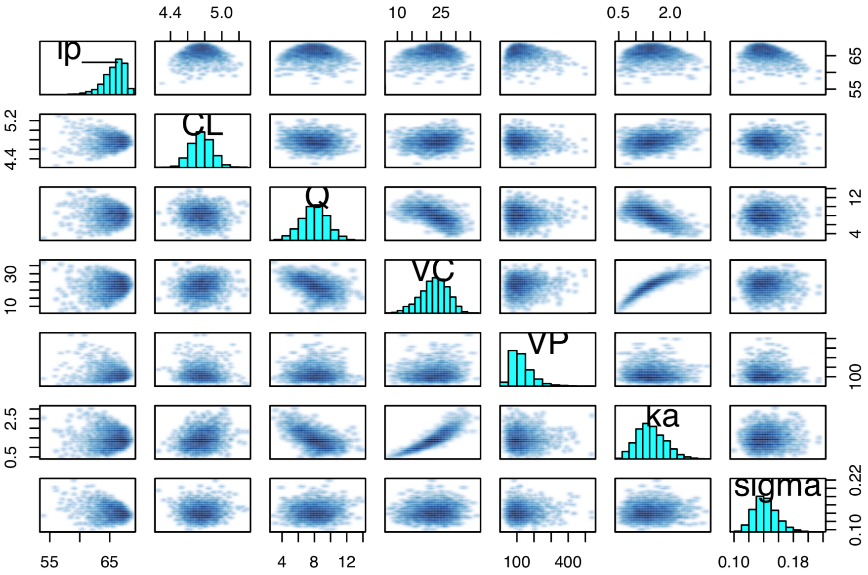
Understanding Automatic Differentiation to Improve Performance

Charles Margossian

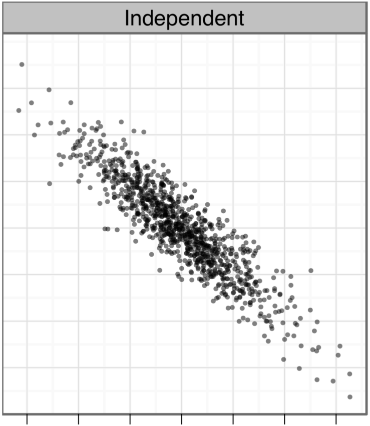
Columbia University, Department of Statistics

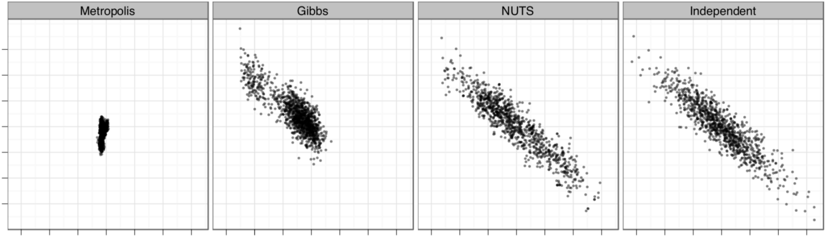
July 22nd 2018



lp
CL
Q
VC
VP
ka
sigma
4.4
5.0
10
25
0.5
2.0
65
55
5.2
4.4
12
4
30
10
100
2.5
0.5
0.22
0.10
55
65
4
8
12
100
400
0.10
0.18

Sampler	**Order of derivative**
Metropolis Hasting, Gibbs	0 (value)
Hamiltonian Monte Carlo	1 (gradient)
Riemannian HMC	2 (Hessian) and 3


Independent

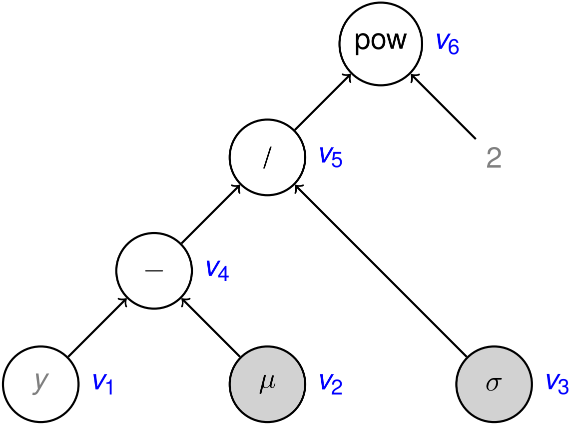

Metropolis
Gibbs
NUTS
Independent

How do we efficiently compute

$$\nabla \log(\pi(\theta|x)) = \left(\frac{\partial}{\partial \theta_1} \log(\pi(\theta|x), ..., \frac{\partial}{\partial \theta_n} \log(\pi(\theta|x) \right)?$$

$$f(y, \mu, \sigma) = \left(\frac{y - \mu}{\sigma}\right)^2$$

Expression graph



Solving algebraic equations

Find x^* such that $f(x^*, \theta) = 0$ and compute $\frac{\partial}{\partial \theta} x^*(\theta)$.
Example – Newton's algorithms:

$$x_{i+1} = x_i - \frac{f'(x_i, \theta)}{f''(x_i, \theta)}$$

Computing derivatives

- $x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)}$

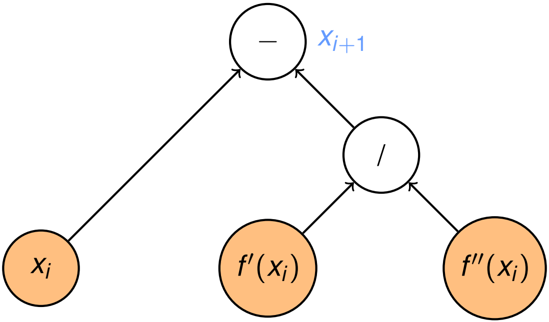


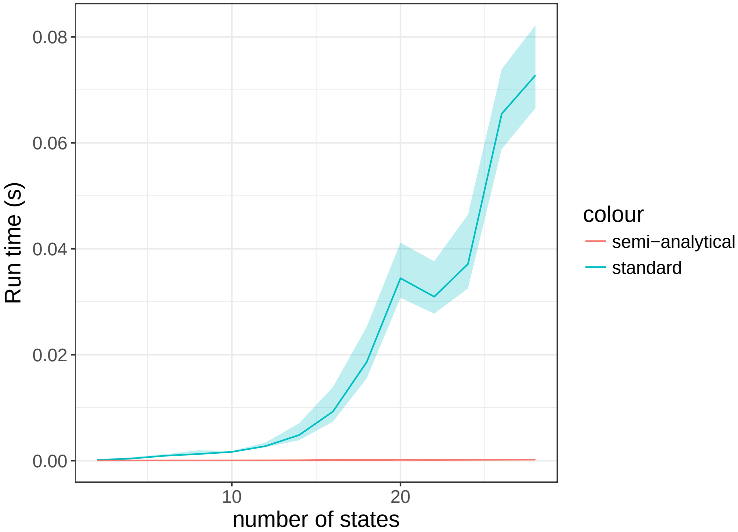
Figure: Topological graph for automatic differentiation. *The orange nodes further expand into topological graphs, across which we apply the chain rule.*

Using semi-analytical solutions

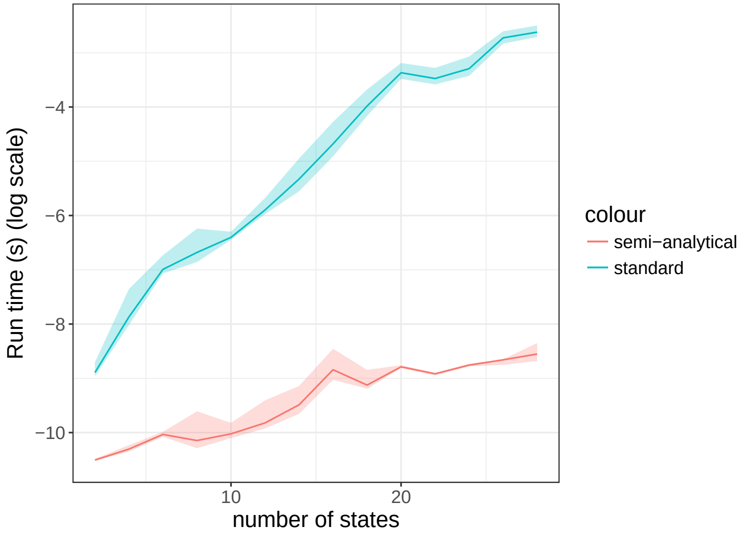
Under certain regularity conditions:

$$\frac{\partial}{\partial\theta}x^*(\theta) = -\left(\frac{\partial f}{\partial x}\right)^{-1}\frac{\partial f}{\partial\theta}$$

- The result extends to higher dimensions, by using Jacobian matrices.



Run time (s)
0.08
0.06
0.04
0.02
0.00
10
20
number of states
colour
semi−analytical
standard



Run time (s) (log scale)
−4
−6
−8
−10
10
20
number of states
colour
semi−analytical
standard

Example: ordinary differential equations

$$y'(t) = f(y, t, \theta)$$

where $y \in \mathbb{R}^n$ and $\theta \in \mathbb{R}^p$.

Example: ordinary differential equations

- $y'(t) = f(y, t, \theta)$

Need to compute:

- the solution: y^*
- the derivatives:

$$J = \begin{bmatrix} \frac{\partial y_1}{\partial \theta_1} & \cdots & \frac{\partial y_1}{\partial \theta_p} \\ \ldots & \ldots & \ldots \\ \frac{\partial y_n}{\partial \theta_1} & \cdots & \frac{\partial y_n}{\partial \theta_p} \end{bmatrix}$$

Components which may require sensitivities

- model parameters, $\theta \in \mathbb{R}^P$
- initial states, $y \in \mathbb{R}^N$
- time, $t_1 \in \mathbb{R}$

$$J = \begin{bmatrix} \frac{\partial y_1}{\partial \theta_1} & \cdots & \frac{\partial y_1}{\partial \theta_p} & \frac{\partial y_1}{\partial y_1^0} & \cdots & \frac{\partial y_1}{\partial y_n^0} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \\ \frac{\partial y_n}{\partial \theta_1} & \cdots & \frac{\partial y_n}{\partial \theta_p} & \frac{\partial y_n}{\partial y_1^0} & \cdots & \frac{\partial y_n}{\partial y_n^0} \end{bmatrix}$$

Coupled Ordinary Differential Equations:

$$
\begin{aligned}
y_1' &= f_1(y, t, \theta) \\
y_2' &= f_2(y, t, \theta) \\
&\cdots \\
\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial y_1}{\partial \theta_1} &= f_{1,1}(y, t, \theta) \\
&\cdots \\
\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial y_n}{\partial \theta_p} &= f_{n,p}(y, t, \theta) \\
&\cdots \\
\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial y_1}{\partial y_1^0} &= f_{n,p}(y, t, \theta) \\
&\cdots
\end{aligned}
$$

Coupled Ordinary Differential Equations:

$$
\begin{aligned}
y_1' &= f_1(y, t, \theta) \\
y_2' &= f_2(y, t, \theta) \\
&\cdots \\
\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial y_1}{\partial \theta_1} &= f_{1,1}(y, t, \theta) \\
&\cdots \\
\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial y_n}{\partial \theta_p} &= f_{n,p}(y, t, \theta) \\
&\cdots \\
\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial y_1}{\partial y_1^0} &= f_{n,p}(y, t, \theta) \\
&\cdots
\end{aligned}
$$

Coupled Ordinary Differential Equations:

$$
\begin{aligned}
y_1' &= f_1(y, t, \theta) \\
y_2' &= f_2(y, t, \theta) \\
&\cdots \\
\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial y_1}{\partial \theta_1} &= f_{1,1}(y, t, \theta) \\
&\cdots \\
\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial y_n}{\partial \theta_p} &= f_{n,p}(y, t, \theta) \\
&\cdots \\
\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial y_1}{\partial y_1^0} &= f_{n,p}(y, t, \theta) \\
&\cdots
\end{aligned}
$$

Coupled Ordinary Differential Equations:

$$
\begin{aligned}
y_1' &= f_1(y, t, \theta) \\
y_2' &= f_2(y, t, \theta) \\
&\dots \\
\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial y_1}{\partial \theta_1} &= f_{1,1}(y, t, \theta) \\
&\dots \\
\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial y_n}{\partial \theta_p} &= f_{n,p}(y, t, \theta) \\
&\dots \\
\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial y_1}{\partial y_1^0} &= f_{n,p}(y, t, \theta) \\
&\dots
\end{aligned}
$$

Number of evaluations when we require sensitivities for model parameters and initial states

$$\mathcal{C} \propto N(N + N^2 + P + P \times N)$$

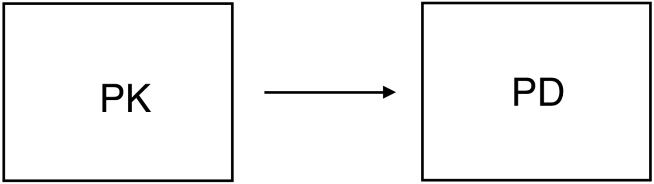
Number of evaluations when we require sensitivities for model parameters and initial states

$$\mathcal{C} \propto N(N + N^2 + P + P \times N)$$

Number of evaluations when we require sensitivities for model parameters and initial states

$$\mathcal{C} \propto N(N + N^2 + P + P \times N)$$

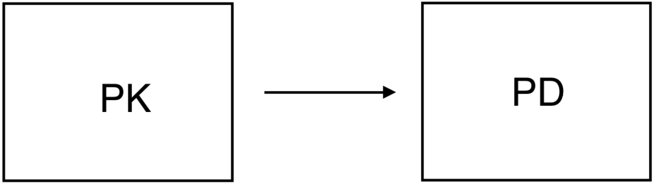
PK / PD ordinary differential equation



$$
\begin{aligned}
y'_{\mathrm{PK}} &= f_{\mathrm{PK}}(y_{\mathrm{PK}}, t) \\
y'_{\mathrm{PD}} &= f_{\mathrm{PD}}(y_{\mathrm{PK}}, y_{\mathrm{PD}}, t)
\end{aligned}
$$

where we note $y_{\mathrm{PK}} \in \mathbb{R}^{N_{\mathrm{PK}}}$ and $y_{\mathrm{PD}} \in \mathbb{R}^{N_{\mathrm{PD}}}$.

PK / PD ordinary differential equation

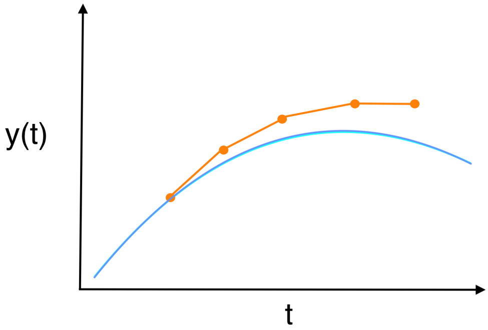


$$\begin{aligned} y'_{\mathrm{PK}} &= f_{\mathrm{PK}}(y_{\mathrm{PK}}, t) \\ y'_{\mathrm{PD}} &= f_{\mathrm{PD}}(y_{\mathrm{PK}}, y_{\mathrm{PD}}, t) \end{aligned}$$

where we note $y_{\mathrm{PK}} \in \mathbb{R}^{N_{\mathrm{PK}}}$ and $y_{\mathrm{PD}} \in \mathbb{R}^{N_{\mathrm{PD}}}$.

Full integration

$$y = \int f(y, t, \theta)\mathrm{d}t$$



Mixed Solving

$$\begin{aligned} y_{\mathrm{PK}} &= F_{\mathrm{PK}}(t, \theta) \\ y_{\mathrm{PD}} &= \int f_{\mathrm{PK}}(F_{\mathrm{PK}}, y_{\mathrm{PK}}, t, \theta)\mathrm{d}t \end{aligned}$$

- Computing F_{PK} is more expensive than computing f!

Computer experiment

- PK model with $N_{\mathrm{PK}} = 3$
- PD model with $N_{\mathrm{PD}} = 5$

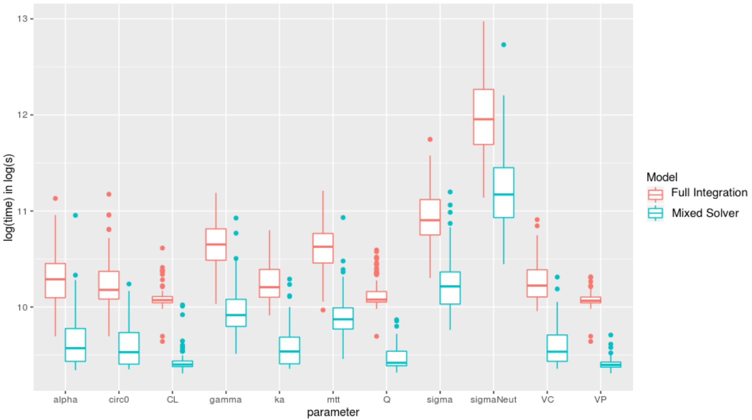
Theoretical relative cost: 0.42

Note $5/8 = 0.625 > 0.42$!

More theoretical results

Initial State for y_1	*Initial State for y_2*	*Parameters*	$\mathcal{R}$
-	-	-	0.625
-	-	+	0.419
-	+	-	0.265
-	+	+	0.345
+	+	+	0.418

Empirical result



$$\mathcal{R} = 51.11 \pm 13.51(\%)$$

Drawbacks:

- Coding analytical solutions is time consuming and error prone.
- There is some difficult bookkeeping when doing mixed solving.

Torsten has routines to do so when the PK is a one or two compartment model.

- `mixedOde1CptModel`
- `mixedOde2CptModel`
- Torsten also uses mixed solving for algebraic equations.

Acknowledgment

- Individuals:
  - Bill Gillespie (Metrum Research Group)
  - Bob Carpenter (Columbia)
  - Andrew Gelman (Columbia)
  - Sebastian Weber (Novartis)
  - Michael Betancourt (Symplectomorphic LLC)
  - Ben Goodrich (Columbia)
  - Yi Zhang (Metrum Research Group)
- Institutions:
  - Office of Naval Research, Bill & Melinda Gates Foundation
  - Columbia University, Metrum Research Group, AstraZeneca