# Automatic Differentiation in the Analysis of Strategies for Mitigation of Global Change

I. Enting

MASCOS
The University of Melbourne

# Acknowledgments

The Center of Excellence for Mathematics and Statistics (MASCOS) is funded by the Australian Research Council.

My fellowship at MASCOS is supported by CSIRO through a sponsorship agreement.

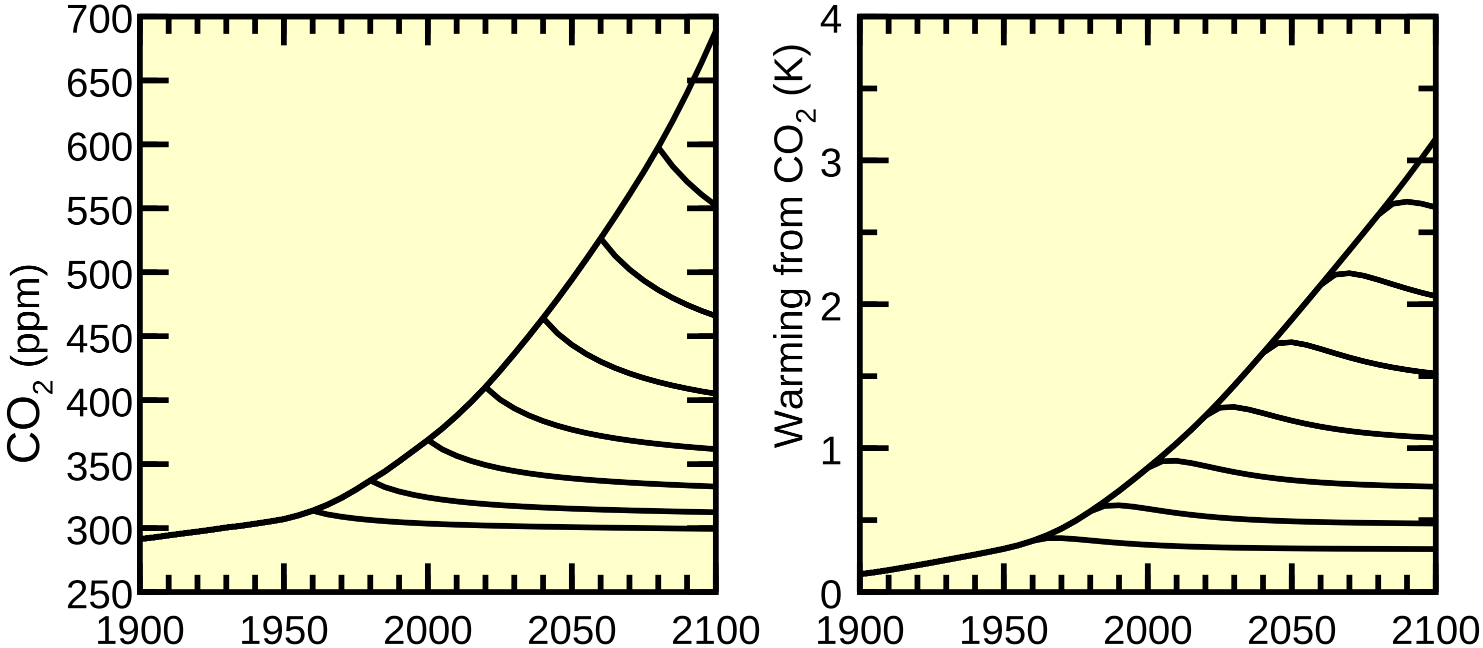
Much of this work builds on work with the members of the MATCH working group studying the Brazilian Proposal, particularly Cathy Trudinger of CSIRO Marine and Atmospheric Research

# Summary

- Modelling global change
- Automatic (algorithmic) differentiation in analysing models
- Algorithmic differentiation in analysing the Brazilian proposal for greenhouse mitigation targets

# Timescales

$CO_2$ concentrations and consequent warming, partitioned according to time of emission.




Lowest bands are from pre-1960 emissions, next from 1960 to 1980 emissions, etc.
Increase in contribution to warming after time of emissions from 'committed warming' effect.

# Modelling Spectrum

| Characteristics | Carbon Cycle | Climate System |
|---|---|---|
| **Black box** | | |
| Empirical | Curve fitting | Curve fitting |
| Stochastic | Airborne fraction | |
| **Grey box** | | |
| | Response function | Response function |
| | Box model | Energy balance model |
| | | |
| **White box** | | |
| Deterministic | | Atmos/ocean GCM |
| Reductionist | Spatially resolved | |
| Mechanistic | | Earth system model |

Spectrum concept: Karlpus 1977; carbon examples: Enting 1987.

# Model Analysis

Most common model calculation is forward projection by (numerical) integration of DEs.

However many aspects of analysing models involve differentiation:

**Sensitivity analysis** — derivatives with respect to parameters;

**Calibration** — techniques such as Maximum Likelihood imply optimisations, facilitated by use of derivatives;

**Data assimilation** — real-time model adjustment — dynamic calibration.

# Tangent Linear Model (TLM)

For a model expressed as $N$ DEs:

$$\frac{d}{dt}x_j = g_j(\{x_k\}, \alpha, t) \qquad \text{for } j = 1, N$$

we can define sensitivities as

$$y_j = \frac{\partial}{\partial \alpha}x_j \qquad \text{for } j = 1, N$$

to give 'the tangent linear model':

$$\frac{d}{dt}y_m = \frac{\partial}{\partial \alpha}g_m(\{x_k\}, \alpha, t) + \sum_n \frac{\partial}{\partial x_n}g_m(\{x_k\}, \alpha, t)\, y_n$$

# Algorithmic Differentiation (AD)

Differentiation by successive use of chain rule.

For binary operation $c = f(a, b)$,

$$\frac{\partial c}{\partial \alpha} = \frac{\partial f}{\partial a} * \frac{\partial a}{\partial \alpha} + \frac{\partial f}{\partial b} * \frac{\partial b}{\partial \alpha}$$

e.g.

$$c = a + b \quad \rightarrow \frac{\partial c}{\partial \alpha} = \frac{\partial a}{\partial \alpha} + \frac{\partial b}{\partial \alpha}$$

$$c = a * b \quad \rightarrow \frac{\partial c}{\partial \alpha} = b * \frac{\partial a}{\partial \alpha} + a * \frac{\partial b}{\partial \alpha}$$

Convert program to code for derivatives, one operation at a time.

# Computational Complexity of AD

For $u_k \to x_j(t) \to y_j$, Jacobian is: $J_{jk} = \frac{\partial y_j}{\partial u_k}$

$$J_{jk} = \sum \frac{\partial y_j}{\partial x_n(T)} \cdots \frac{\partial x_{n'}(t)}{\partial x_{m'}(t-1)} \cdots \frac{\partial x_n(1)}{\partial x_n(0)} \frac{\partial x_n(0)}{\partial u_k}$$

Tangents: $$\frac{\partial y_n}{\partial \alpha} = \sum_k J_{jk} \frac{\partial u_k}{\partial \alpha}$$

TLM is successive product of vector $\times$ sparse matrix.

Gradients: $$\frac{\partial \phi}{\partial u_k} = \sum_j J_{jk} \frac{\partial \phi}{\partial y_j}$$

Adjoint model achieves efficiency of vector $\times$ sparse matrix by using chain rule backwards in time.

# Approaches to AD

Hand-code program to calculate derivatives — laborious, error-prone and must be repeated each time the model changes.

Symbolic algebra (e.g. Mathematica) — problematic for adjoints.

Tangent/adjoint compilers — transform source into code for tangent or adjoint models.

Operator overloading to produce a 'script' that is analysed to give code for the derivatives.

Use operator overloading capabilities directly — straightforward for tangent-linear-model, but restricted applicability to adjoint models.

# Operator Overloading

Replace real variable $x$, (type `double`), with composite variable $\tilde{x}$ (type `Xvar`), representing both value $x$ and its derivatives with respect to $K$ model quantities, $\alpha_k$ as:

$$\tilde{x}_0 = x \quad \text{and} \quad \tilde{x}_k = \frac{\partial}{\partial \alpha_k} x \quad \text{for } k = 1, K$$

Operator overloading implements $\tilde{c} = \tilde{a} * \tilde{b}$, representing:

$$\tilde{c}_0 = \tilde{a}_0 * \tilde{b}_0 \quad \text{and} \quad \tilde{c}_k = \tilde{a}_0 * \tilde{b}_k + \tilde{a}_k * \tilde{b}_0$$

Overloaded functions, $\tilde{c} = f(\tilde{a})$, represent:

$$\tilde{c}_0 = f(\tilde{a}_0) \quad \text{and} \quad \tilde{c}_k = f'(\tilde{a}_0) * \tilde{a}_k$$

where $f'(.)$ denotes the derivative of $f(.)$

# Class Definitions

Fragment of C++ class definition to implement operator overloading:

```
class Xvar{
public :
static const int ns = _NUMDERIVS+1;
double xs[_NUMDERIVS+1];
Xvar operator*(Xvar);
...
};

Xvar Xvar::operator*(Xvar b){ Xvar c;
for (int i=1; i < ns; i++) c.xs[i] =
xs[i]*b.xs[0]+xs[0]*b.xs[i];
c.xs[0] = xs[0]*b.xs[0];
return c;} ;
...
```

# Brazilian Proposal

Tabled by Brazil during negotiations leading to Kyoto Protocol — Flicked-passed to Subsidiary Body of Scientific and Technical Advice (SBSTA).

Proposes that emission reduction targets should be proportional to nation's relative responsibility for the greenhouse effect.

Issues:

- Indicator? What quantity is used as a measure of the greenhouse effect?
- For what period of emissions is responsibility attributed?
- How are non-linear responses attributed?

# Brazilian Proposal as Derivatives

As example, use indicator $T^* = T_{\mathrm{CO2}}(2100)=$ warming in 2100 from $CO_2$ emissions.

$T^*$ is to be attributed to emissions $E_j(t)$ from country $j$ with $E(t) = \sum_j E_j(t)$.

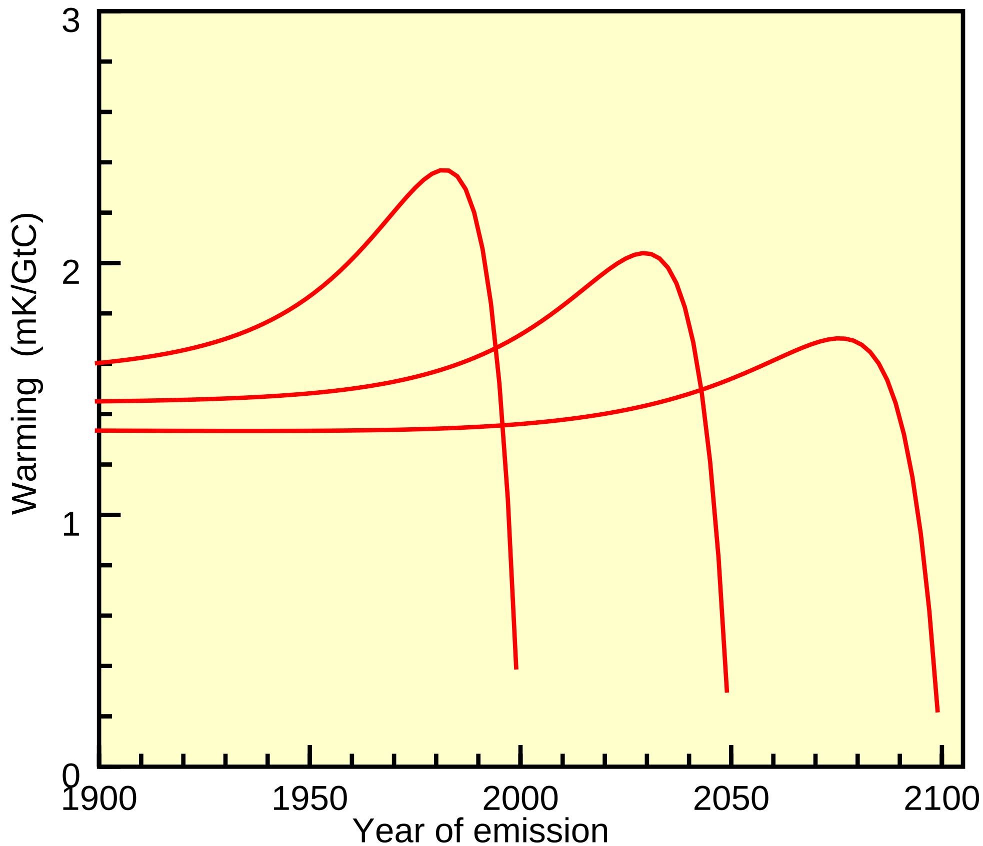
Differential attribution to country $j$ of emissions at time $t$ is

$$\frac{\partial T^*}{\partial E_j(t)} E_j(t) = \frac{\partial T^*}{\partial E(t)} E_j(t) = S(t) E_j(t)$$

where $S(t)$ is a functional derivative.

Cumulated attribution: $T_j^* = \int S(t) E_j(t)\, dt$

# Results: Functional Derivatives




Assumes IS92a emissions. Represents temperature by response function. Linear responses for ocean and biotic carbon, coupled non-linearly to atmospheric $CO_2$ (as in CSIRO study).

$\frac{\partial}{\partial E(t)} T(\tau)$ for $\tau = 2000, 2050, 2100$.

Decrease as $t \rightarrow \tau$ shows 'committed warming'.

At any time, warming from most recent releases is yet to happen.

# Implications

- For a given indicator, $T^*$, calculation of $S(t)$ allows attribution to any nation.

- $S(t)$ most efficiently calculated from adjoint model, but for multiple indicator times, tangent linear model not too inefficient.

- Sensitivity of $T_j^*$ to model uncertainties can be obtained as second derivatives.

- Sensitivity of $T_j^*$ to uncertainties in emissions can be obtained as

$$\text{Var}[T_j^*] = \int \int S(t) \text{Cov}[E_j(t), E_j(t')] S(t') \, dt' \, dt$$

# Conclusions

**Algorithmic differentiation —**

Operator overloading is a straightforward way of developing tangent linear models (and obtaining higher derivatives if needed).

**Brazilian Proposal —**

Attribution in terms of derivatives is readily calculated using algorithmic differentiation. Higher derivatives give sensitivities.

**Global change —**

Potential should extend to other analyses of uncertainties in global change.

# Further Information

**Algorithmic Differentation** —
*Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, Andreas Griewank (SIAM).

**Brazilian Proposal** — MATCH website:
*http://www/match-info.net*
My site:
*http://ms.unimelb.edu.au/~enting/brazil.html*

**This study** — Extended abstracts from MODSIM 2005.