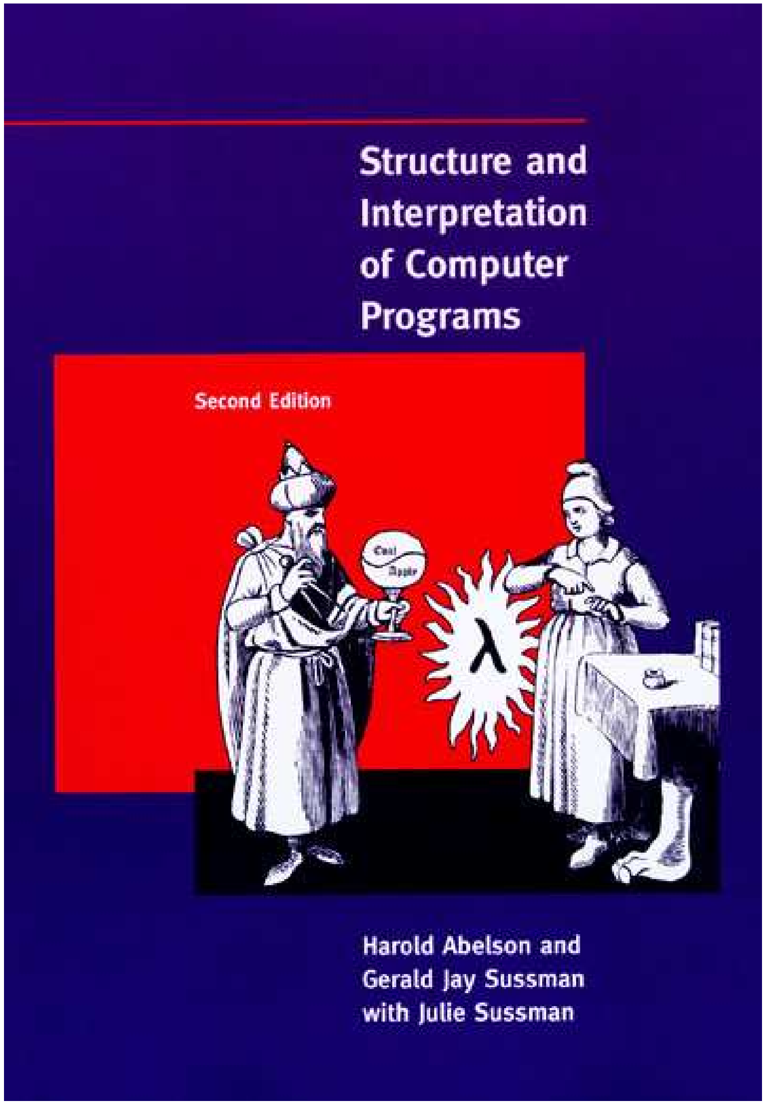
Programming for the Expression of Ideas

Gerald Jay Sussman

`gjs@mit.edu`

MASSACHVSETTS INSTITVTE OF TECHNOLOGY

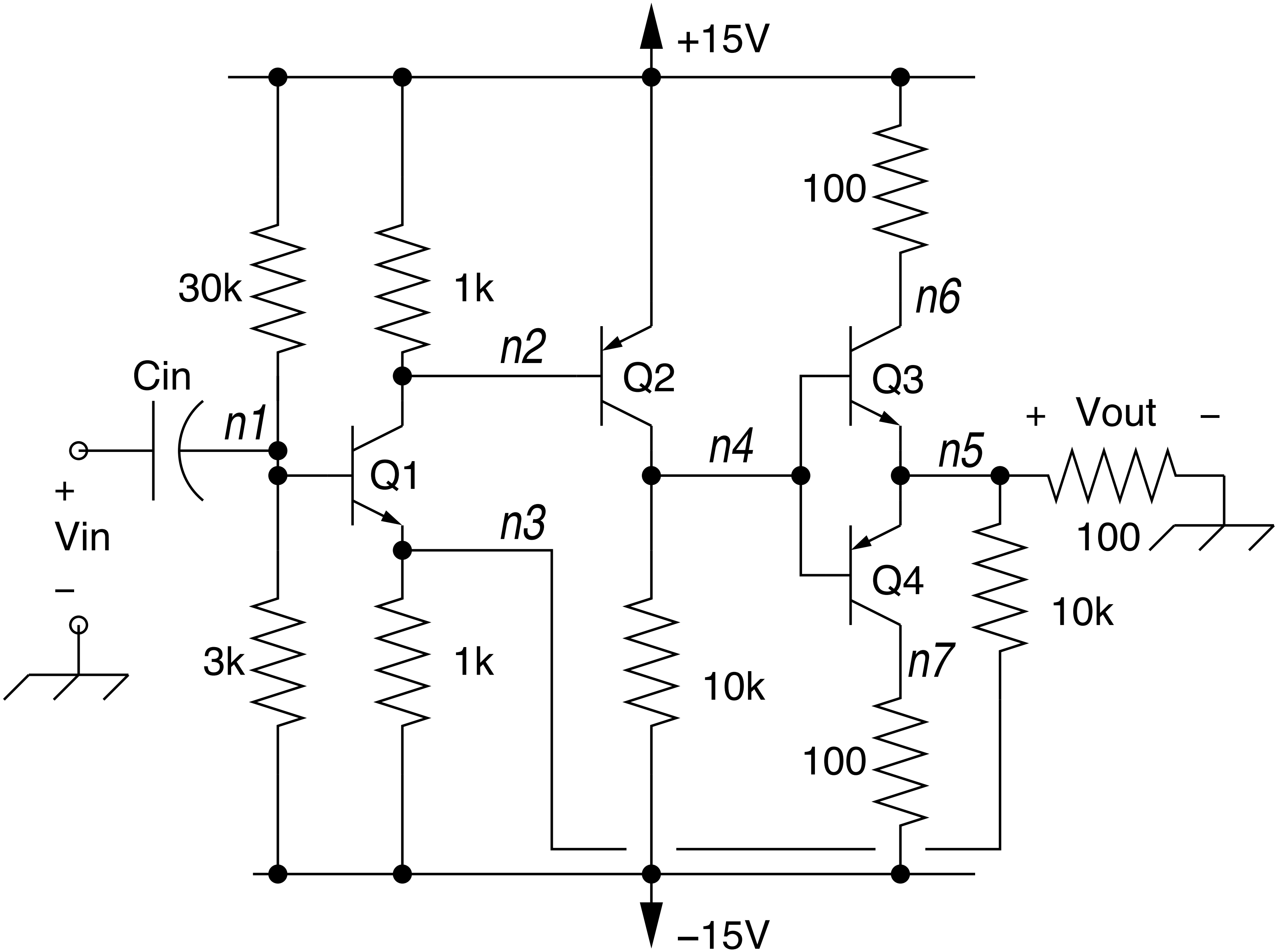
SICP



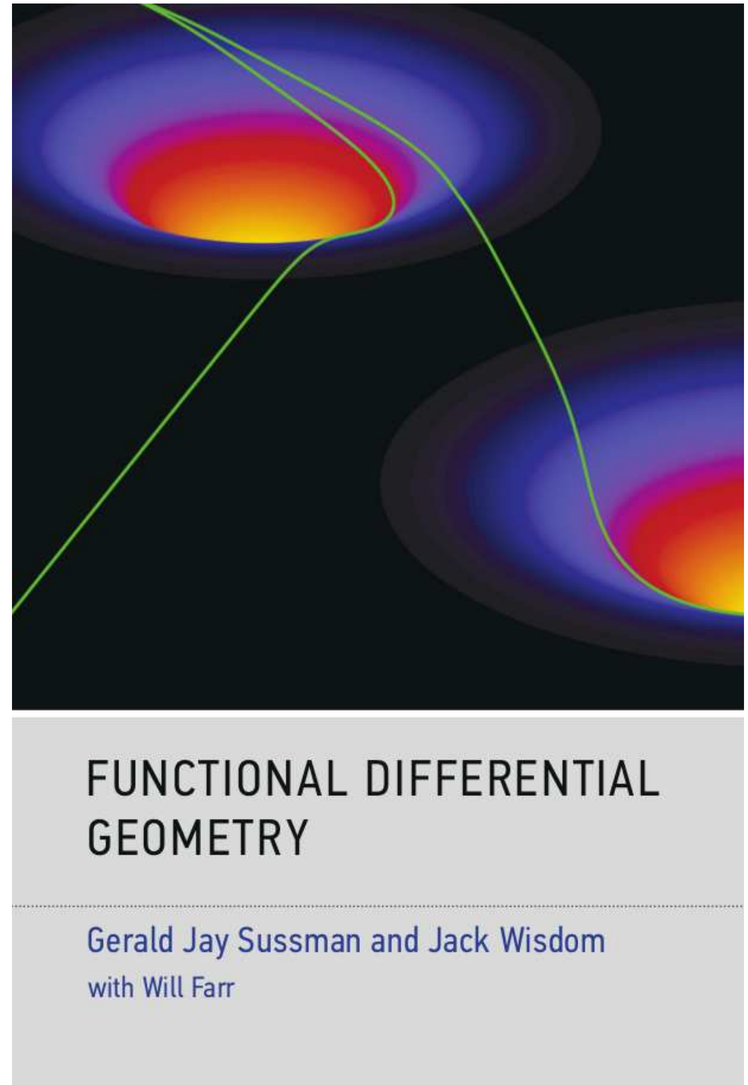
Preparing the Way

"The computer revolution is a revolution in the way we think and in the way we express what we think."

Teaching Circuits



Now: Physics and Math



First: Something Really Stupid

$$\cos^2\theta = (\cos\theta)^2$$

But

$$\cos^{-1}\theta \neq \frac{1}{\cos\theta}$$

Duh?!

Impressionistic Math

What does $\vec{F} = m\vec{a}$ really mean?

Particle in gravitational field with trajectory $\vec{r}$.
$\vec{r}$ is a function of time. Its value is a radial position vector from the center of field.

The trajectory must satisfy:

$$-\frac{GMm}{\|\vec{r}(t)\|^2}\frac{\vec{r}(t)}{\|\vec{r}(t)\|} = m\frac{d^2\vec{r}(t)}{dt^2}$$

Newton's equation is really a macro,

with untyped and undeclared parameters.

Leibniz notation trouble!

Let $e(x, y) = f(g(x, y), h(x, y))$. We want $\frac{\partial e(x,y)}{\partial x}$.
We let $u = g(x, y)$ and $v = h(x, y)$ then we write:

$$\frac{\partial f(g(x, y), h(x, y))}{\partial x} = \frac{\partial f(u, v)}{\partial u}\frac{\partial u}{\partial x} + \frac{\partial f(u, v)}{\partial v}\frac{\partial v}{\partial x}.$$

Or, more telegraphically,

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial u}\frac{\partial u}{\partial x} + \frac{\partial f}{\partial v}\frac{\partial v}{\partial x}.$$

Note that f means something different on the two sides of this equation!

(Paraphrase of Spivak, *Calculus on Manifolds*, p.44)

The Correct Chain Rule

$$\frac{\partial f(g(x,y),h(x,y))}{\partial x}$$

$$= \quad \left.\frac{\partial f(u,v)}{\partial u}\right|_{\substack{u=g(x,y)\\ v=h(x,y)}} \frac{\partial g(x,y)}{\partial x}$$

$$+ \left.\frac{\partial f(u,v)}{\partial v}\right|_{\substack{u=g(x,y)\\ v=h(x,y)}} \frac{\partial h(x,y)}{\partial x}$$

Ugh bletch!

Lagrange's Equations of Motion

Lagrange's equations are satisfied by the possible motions.

In traditional telegraphic Leibniz notation we write:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}^i}\right) - \frac{\partial L}{\partial q^i} = 0.$$

The Lagrangian

$$L : \mathbf{R} \times \mathbf{R}^n \times \mathbf{R}^n \to \mathbf{R}$$

is a real-valued function of time, generalized coordinates, and generalized velocities.

Type violation!

The Explanation

Let w be the path to be tested, then:

$$\frac{d}{dt}\left(\left.\frac{\partial L(t,q,\dot{q})}{\partial \dot{q}}\right|_{\substack{q\,=\,w(t)\\ \dot{q}\,=\,\frac{dw(t)}{dt}}} \right) - \left.\frac{\partial L(t,q,\dot{q})}{\partial q}\right|_{\substack{q\,=\,w(t)\\ \dot{q}\,=\,\frac{dw(t)}{dt}}} = 0.$$

Pretty ugly, but correct.

Functional Abstraction 1

1. Get rid of q and $\dot{q}$, and use positional notation.

$$\frac{d}{dt}((\partial_2 L)(t, w(t), \frac{d}{dt}w(t))) - (\partial_1 L)(t, w(t), \frac{d}{dt}w(t)) = 0$$

2. Introduce functional derivative.

$$(Df)(t) = \left. \frac{d}{dx} f(x) \right|_{x=t}$$

3. Construct the state-space path from the configuration-space path.

$$\Gamma[w](t) = (t, w(t), Dw(t))$$

Functional Abstraction 2

Lagrange's equations are now:

$$\frac{d}{dt}((\partial_2 L)(\Gamma[w](t))) - (\partial_1 L)(\Gamma[w](t)) = 0$$

4. Using composition $(f \circ g)(x) = f(g(x))$:

$$D((\partial_2 L) \circ (\Gamma[w])) - (\partial_1 L) \circ (\Gamma[w]) = 0$$

This is clear, complete, unambiguous, and functional.
There are no missing parameters or extraneous symbols.

As Scheme Code

$$D((\partial_2 L) \circ (\Gamma[w])) - (\partial_1 L) \circ (\Gamma[w]) = 0$$

```
(define ((Lagrange-equations Lagrangian) w)
  (- (D (compose ((partial 2) Lagrangian)
        (Gamma w)))
     (compose ((partial 1) Lagrangian)
              (Gamma w))))
```

$$\Gamma[w](t) = (t, w(t), Dw(t))$$

```
(define ((Gamma w) t)
  (up t (w t) ((D w) t)))
```

Example: Harmonic Oscillator

A Lagrangian is (kinetic energy $-$ potential energy).

$$L(t, q, v) = \frac{mv^2}{2} - \frac{kq^2}{2}$$

```
(define ((L-harmonic m k) local)
  (let ((q (coordinate local))
        (v (velocity local)))
    (- (* 1/2 m (square v))
       (* 1/2 k (square q)))))
```

We know the general solution is $x(t) = a\cos(\omega t + \varphi)$.

```
(define (proposed-solution t)
  (* 'a (cos (+ (* 'omega t) 'phi))))
```

Example continued

Let's try the proposed solution:

```
(show-expression
 (((Lagrange-equations (L-harmonic 'm 'k))
   proposed-solution)
  't))
```

The residual is

$$\cos\left(\omega t + \varphi\right) a \left(k - m\omega^2\right)$$

So for the residual to be zero $\omega = \sqrt{k/m}$.

Example continued

But if we don't know the form of the solution, we can try a literal function:

```
(show-expression
 (((Lagrange-equations (L-harmonic 'm 'k))
   (literal-function 'x))
  't))
```

The residual is Newton's equation of motion:

$$kx\left(t\right) + mD^2x\left(t\right)$$

The Moral

Originally, Lagrange's equations had missing parameters and a type error. Programming them forced an elegant and *effective* statement.

Free Motion

If there are no forces acting on a particle a Lagrangian for the motion of the particle is its kinetic energy.

```
(define ((Lfree mass) state)
  (* 1/2 mass
     (square (velocity state))))
```

On a Sphere

```
;;; Transformation of coordinates
(define ((sphere->R3 R) state)
  (let ((q (coordinate state)))
    (let ((theta (ref q 0)) (phi (ref q 1)))
      (up (* R (sin theta) (cos phi))      ; x
          (* R (sin theta) (sin phi))      ; y
          (* R (cos theta))))))            ; z

;;; Transformation of states
(define ((F->C F) state)
  (up (time state)
      (F state)
      (+ (((partial 0) F) state)
         (* (((partial 1) F) state)
            (velocity state)))))
```

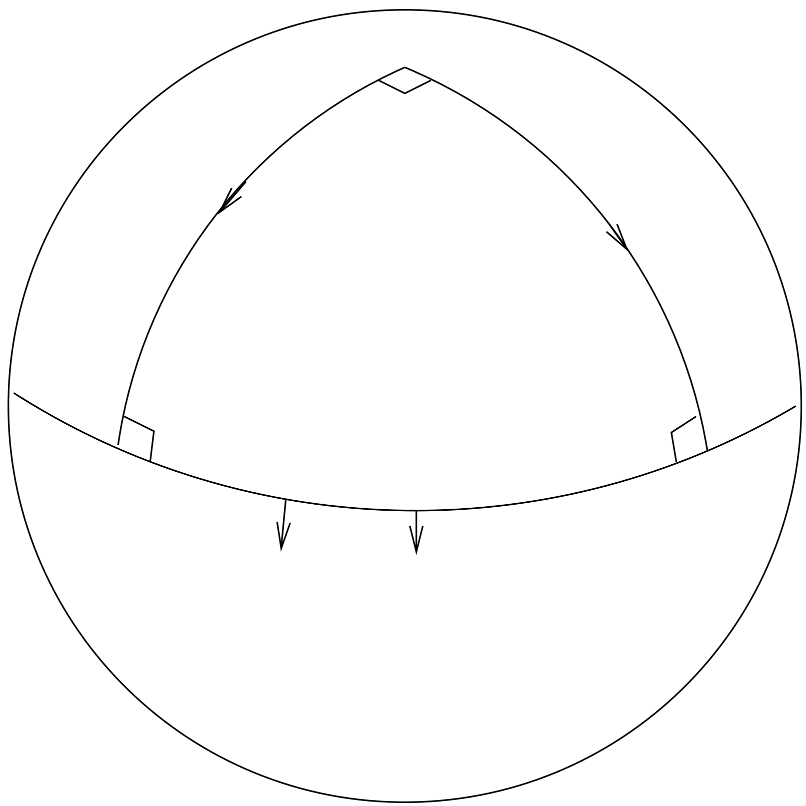
Free Motion on a Sphere

```
;;; On a sphere
(define (Lsphere m R)
  (compose (Lfree m)
           (F->C (sphere->R3 R))))

;;; Equations of motion are:
(((Lagrange-equations Lsphere)
  (up (literal-function 'theta) ; Path
      (literal-function 'phi)))
 't)                            ; time
```

mess

Parallel Transport



Geodesics

Let's take a long walk.

Start at North Pole.
Scratch a line pointing at Greenwich.

Walk south, down Longitude 0°, pointing a stick south.

At Equator walk east.
Keep stick pointing south.

At 90°E walk north.
Keep stick pointing south.

At North Pole compare stick with scratch.

Geodesics

Geodesic motion is motion with a velocity that is transported parallel to itself.

$$\nabla_{\mathsf{v}}\mathsf{v} = 0$$

More precisely,

$$\nabla^{\gamma}_{\partial/\partial\mathsf{t}} d\gamma(\partial/\partial\mathsf{t}) = 0.$$

In coordinates

$$D^2 q^i(t) + \sum_{jk} \Gamma^i_{jk}(\gamma(t)) Dq^j(t) Dq^k(t) = 0$$

Gaack!

Geodesic Equations

```
(define (make-Cartan metric coords)
  (Christoffel->Cartan
   (metric->Christoffel-2 metric
     (coordinate-system->basis coords))))

(define
    (((geodesic-eqns metric coords) path) t)
  (let ((C (make-Cartan metric coords)))
    ((raise metric coords)
     (((((covariant-derivative C path) d/dt)
        ((differential path) d/dt))
       (chart coords))
      ((point R1-rect) t)))))
```

Metric for a Sphere

Metric for a sphere of radius R is:

$$\mathsf{g}(\mathsf{u},\mathsf{v}) = R^2 \mathsf{d}\theta(\mathsf{u})\mathsf{d}\theta(\mathsf{v}) + R^2(\sin\theta)^2\mathsf{d}\phi(\mathsf{u})\mathsf{d}\phi(\mathsf{v})$$

```
(define ((SphereMetric R) u v)
  (define-coordinates (up theta phi)
                      S2-spherical)
  (* (square R)
     (+ (* (dtheta u)
           (dtheta v))
        (* (square (sin theta))
           (dphi u)
           (dphi v)))))
```

Free Motion is Geodesic Motion

```
(define a-path
  (literal-manifold-map 'gamma
                        R1-rect S2-spherical))

(define coord-path
  (compose (chart S2-spherical)
           a-path
           (point R1-rect)))

(- (((Lagrange-equations Lsphere)
     coord-path)
    't)
   (((geodesic-eqns SphereMetric) a-path) 't))
```

(down 0 0) ; They are the same!

The Newton Metric

```
(define (Newton-metric M G c V)
  (let ((a (+ 1 (* (/ 2 (square c))
                   (compose V (up x y z))))))
    (define (g v1 v2)
      (+ (* -1 (square c) a (dt v1) (dt v2))
         (* (dx v1) (dx v2))
         (* (dy v1) (dy v2))
         (* (dz v1) (dz v2))))
    g))

(define nabla
  (covariant-derivative
    (connection Newton-metric 'M ':G ':c V)))
```

Geodesic motion is $F = ma$

The geodesic equations for spacetime with the Newton metric g for potential field V

$$\begin{aligned} \mathsf{g}(\mathsf{v}_1, \mathsf{v}_2) = & - & c^2 \left(1 + \frac{2V}{c^2}\right) \mathsf{dt}(\mathsf{v}_1)\mathsf{dt}(\mathsf{v}_2) \\ & + & \mathsf{dx}(\mathsf{v}_1)\mathsf{dx}(\mathsf{v}_2) \\ & + & \mathsf{dy}(\mathsf{v}_1)\mathsf{dy}(\mathsf{v}_2) \\ & + & \mathsf{dz}(\mathsf{v}_1)\mathsf{dz}(\mathsf{v}_2) \end{aligned}$$

are Newton's equations

$$D^2\vec{x}(t) = -(\mathrm{grad}V)(\vec{x}(t))$$

to lowest order in V/c^2.

Einstein's Field Equations

The local energy-momentum distribution determines the local shape of spacetime.
Einstein's field equations, in traditional tensor notation

$$R_{\mu\nu} - \frac{1}{2} R g_{\mu\nu} + \Lambda g_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}$$

can also be written as

$$R_{\mu\nu} = \frac{8\pi G}{c^4} \left(T_{\mu\nu} - \frac{1}{2} T g_{\mu\nu} \right) - \Lambda g_{\mu\nu}$$

Analogous to the Poisson equation for Newtonian gravity:

$$\mathrm{Lap}(V) = 4\pi G \rho$$

Setup for Derivation

```
(define a-typical-event
  ((point spacetime-rect)  (up 't 'x 'y 'z)))

(define V    ;A general potential field
  (literal-function 'V (-> (UP Real 3) Real)))

(define (Ricci nabla basis)
  (define (Ricci-tensor u v)
    (contract
     (lambda (ei wi)
       ((Riemann nabla) wi u ei v))
     basis))
  Ricci-tensor)
```

LHS: Ricci and Laplacian

We compute the time-time component of the left-hand side of the Einstein equations:

```
(((Ricci nabla ...) d/dt d/dt) a-typical-event)
```

mess

The leading terms of this mess are

$$(\partial_0^2 V + \partial_1^2 V + \partial_2^2 V)(x, y, z) = \mathrm{Lap}(V)(x, y, z),$$

the left-hand side of the Poisson equation!
Other terms are smaller by V/c^2.

RHS: Energy-Momentum and Density

If all the stress-energy is matter density ρ we have

```
(define (Tdust rho)
  (lambda (w1 w2)
    (* rho (w1 d/dt) (w2 d/dt))))
```

We evaluate the right-hand side of the Einstein equations:

```
(let* ((g (Newton-metric 'M 'G ':c V))
       (T_ij ((drop2 g) (Tdust 'rho)))
       (T ((trace2down g) T_ij)))
  (* (/ (* 8 ':pi ':G) (expt ':c 4))
     ((- (T_ij d/dt d/dt)
         (* 1/2 T (g d/dt d/dt)))
      a-typical-event)))
```

(4 :pi :G rho)*

So we got the RHS of the Poisson equation!

Einstein's Field Equations

Einstein's equations:

```
(define (Einstein metric)
  (let* (... setup stuff ... )
    (define (E v1 v2)
      (- (Ricci v1 v2)
         (* 1/2 Ricci-scalar (metric v1 v2))))
    E))

(define (Einstein-field-equation metric L T)
  (let ((E (Einstein metric)))
    (define EFE-residuals
      (- (+ E (* L metric))
         (* (/ (* 8 :pi :G) (expt :c 4))
            T)))
    EFE-residuals))
```

Friedmann Metric

```
(define (FLRW-metric c k R)
  (define-coordinates (up t r theta phi)
                      spacetime-sphere)
  (let ((a (/ (square (compose R t))
              (- 1 (* k (square r)))))
        (b (square (* (compose R t) r))))
    (define (g v1 v2)
      (+ (*  -1 (square c) (dt v1) (dt v2))
         (* a (dr v1) (dr v2))
         (* b (+ (* (dtheta v1) (dtheta v2))
                 (* (square (sin theta))
                    (dphi v1) (dphi v2))))))
    g))
```

Perfect Fluid

```
(define (Tperfect-fluid rho p c metric)
  (define-coordinates (up t r theta phi)
                      spacetime-sphere)
  (define (T w1 w2)
    (+ (* (+ (compose rho t)
             (/ (compose p t) (square c)))
          (w1 d/dt) (w2 d/dt))
       (* (compose p t)
          ((metric:invert metric) w1 w2))))
  T)
```

Cosmology

With just this setup the computer can derive the Robertson-Walker equations for the evolution of the universe:

$$\left(\frac{DR(t)}{R(t)}\right)^2 + \frac{kc^2}{(R(t))^2} - \frac{\Lambda c^2}{3} = \frac{8\pi G}{3}\rho(t),$$

$$2\frac{D^2R(t)}{R(t)} - \frac{2}{3}\Lambda c^2 = -8\pi G\left(\frac{\rho(t)}{3} + \frac{p(t)}{c^2}\right).$$

Conclusion

> A computer is like a violin. You can imagine a novice trying first a phonograph and then a violin. The latter, he says, sounds terrible. That is the argument we have heard from our humanists and most of our computer scientists. Computer programs are good, they say, for particular purposes, but they aren't flexible. Neither is a violin, or a typewriter, until you learn how to use it.

Marvin Minsky, "WHY PROGRAMMING IS A GOOD MEDIUM FOR EXPRESSING POORLY UNDERSTOOD AND SLOPPILY-FORMULATED IDEAS" in *Design and Planning II – Computers in Design and Communication*, (Martin Krampen and Peter Seitz, eds.), Visual Committee Books, Hastings House Publishers, New York, 1967.