	Math	Simple Lisp
	Math	Simple Lisp
Addition	$a + b + c$	`(sum a b c)`
Subtraction	$a - b$	`(sum a (minus b))`
Multiplication	abc	`(product a b c)`
Division	a/b	`(product a (oneover b))`
Power	a^b	`(power a b)`
Component	${A^1}_2$	`(product A12 (tensor 1 2))`

Symbolic expressions

Products of sums are expanded.

```
? (product a (sum b c))
(sum (product a b) (product a c))

? (power (sum a b) 2)
(sum (power a 2) (power b 2) (product 2 a b))
```

Sums in an exponent are expanded.

```
? (power a (sum b c))
(product (power a b) (power a c))
```

Vectors, matrices, and tensors are written as sums of components.

The following example computes the inner product of two vectors A and B.

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}, \quad B = \begin{pmatrix} B_1 \\ B_2 \end{pmatrix}, \quad A \cdot B = A_1 B_1 + A_2 B_2$$

```
? (setq A (sum (product A1 (tensor 1)) (product A2 (tensor 2))))
? (setq B (sum (product B1 (tensor 1)) (product B2 (tensor 2))))
? (dot A B)
(sum (product A1 B1) (product A2 B2))
```

Tensor components can use symbolic indices. The following example is the same as above except x and y are used for the index names.

```
? (setq A (sum (product A1 (tensor x)) (product A2 (tensor y))))
? (setq B (sum (product B1 (tensor x)) (product B2 (tensor y))))
? (dot A B)
(sum (product A1 B1) (product A2 B2))
```

GR example

Define the metric tensor.

$$g_{\mu\nu} = \begin{pmatrix} -\xi(r) & 0 & 0 & 0 \\ 0 & 1/\xi(r) & 0 & 0 \\ 0 & 0 & r^2 & 0 \\ 0 & 0 & 0 & r^2 \sin^2\theta \end{pmatrix}$$

```
(setq gtt (product -1 (xi r)))
(setq grr (power (xi r) -1))
(setq gthetatheta (power r 2))
(setq gphiphi (product (power r 2) (power (sin theta) 2)))

(setq gdd (sum
  (product gtt (tensor t t))
  (product grr (tensor r r))
  (product gthetatheta (tensor theta theta))
  (product gphiphi (tensor phi phi))
))
```

Compute $g^{\mu\nu} = g_{\mu\nu}^{-1}$.

```
(setq g (determinant gdd t r theta phi))
(setq guu (product (power g -1) (adjunct gdd t r theta phi)))
```

Compute connection coefficients.

$$\Gamma_{\mu\beta\gamma} = \frac{1}{2}(g_{\mu\beta,\gamma} + g_{\mu\gamma,\beta} - g_{\beta\gamma,\mu})$$

```
(define gradient (sum
  (product (derivative arg t) (tensor t))
  (product (derivative arg r) (tensor r))
  (product (derivative arg theta) (tensor theta))
  (product (derivative arg phi) (tensor phi))
))

(setq grad (gradient gdd))

(setq GAMDDD (product 1/2 (sum
  grad
  (transpose 2 3 grad)
  (minus (transpose 1 2 (transpose 1 2 grad)))
)))
```

$$\Gamma^{\alpha}{}_{\beta\gamma} = g^{\alpha\mu}\Gamma_{\mu\beta\gamma}$$

```
(setq GAMUDD (contract 2 3 (product guu GAMDDD)))
```

Compute Riemann tensor.

$$R^{\alpha}{}_{\beta\gamma\delta} = \frac{\partial \Gamma^{\alpha}{}_{\beta\delta}}{\partial x^{\gamma}} - \frac{\partial \Gamma^{\alpha}{}_{\beta\gamma}}{\partial x^{\delta}} + \Gamma^{\alpha}{}_{\mu\gamma}\Gamma^{\mu}{}_{\beta\delta} - \Gamma^{\alpha}{}_{\mu\delta}\Gamma^{\mu}{}_{\beta\gamma}$$

```
(setq grad (gradient GAMUDD))

(setq GAMGAM (contract 2 4 (product GAMUDD GAMUDD)))

(setq RUDDD (sum
  (transpose 3 4 grad)
  (product -1 grad)
  (transpose 2 3 GAMGAM)
  (product -1 (transpose 3 4 (transpose 2 3 GAMGAM)))
))
```

Compute Ricci tensor.

$$R_{\mu\nu} = R^{\alpha}{}_{\mu\alpha\nu}$$

```
(setq RDD (contract 1 3 RUDDD))
```

Compute Ricci scalar.

$$R = R^{\mu}{}_{\mu}$$

```
(setq R (contract 1 2 (contract 2 3 (product guu RDD))))
```

Compute Einstein tensor.

$$G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2} g_{\mu\nu} R$$

```
(setq GDD (sum RDD (product -1/2 gdd R)))
```