The Mathematica® Journal

Tricks of the Trade

Edited by Paul Abbott

Department of Physics
University of Western Australia
35 Stirling Highway
Nedlands WA 6009, Australia
tmj@physics.uwa.edu.au
physics.uwa.edu.au/~paul

This is a column of programming tricks and techniques, most of which, we hope, will be contributed by our readers.

Note

To improve the appearance of graphics appearing in this column, **TraditionalForm** will be used as the default format type.

```
$FormatType = TraditionalForm;
```

The default text style is set as follows.

```
$TextStyle = {FontSize → 10};
```

Inflight Conundrums

Here are two puzzles taken from an inflight magazine.

The Mathematica Journal ***8***:*4 (2002)* *© 2002 Wolfram Media, Inc.*

□ Puzzle 1

Which symbol should replace the ⚠?

✸	□	✸	□	26
□	○	○	✸	
○	✸	○	□	
○	□	⚠	⚠	29
23	20			

First, express this table of symbols as a matrix.

$$\mathcal{M} = \begin{pmatrix} ✸ & □ & ✸ & □ \\ □ & ○ & ○ & ✸ \\ ○ & ✸ & ○ & □ \\ ○ & □ & ⚠ & ⚠ \end{pmatrix};$$

Then, introduce the unknowns a and b to represent the sum of the second and third rows respectively, so as to obtain equations for each row.

$$\textbf{rowsums} = \mathcal{M}.\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} == \begin{pmatrix} 26 \\ a \\ b \\ 29 \end{pmatrix}$$

$$\begin{pmatrix} 2\,□ + 2\,✸ \\ 2\,○ + □ + ✸ \\ 2\,○ + □ + ✸ \\ ○ + □ + 2\,⚠ \end{pmatrix} == \begin{pmatrix} 26 \\ a \\ b \\ 29 \end{pmatrix}$$

Similarly, introduce c and d for the sum of the third and fourth columns respectively.

$$\textbf{colsums} = (1 \quad 1 \quad 1 \quad 1).\mathcal{M} == (23 \quad 20 \quad c \quad d)$$

$$(2\,○ + □ + ✸ \quad ○ + 2\,□ + ✸ \quad 2\,○ + ✸ + ⚠ \quad 2\,□ + ✸ + ⚠) == (23 \quad 20 \quad c \quad d)$$

Together, these equations allow us to determine all the unknowns.

```
Solve[{rowsums, colsums}, Join[Variables[ℳ], {a, b, c, d}]] // First
```

{○ → 5, □ → 2, ✸ → 11, ⚠ → 11, a → 23, b → 23, c → 32, d → 26}

To determine which symbol should replace the ⚠, we use pattern-matching.

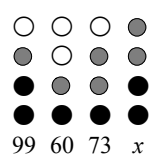
```
Cases[% /. (x_ → (⚠ /. %)) → x, _Symbol]
```

{✸, ⚠}

Hence ✸ should replace ⚠.

□ Puzzle 2

What is the value of the column with total x?



The solution is immediate.

```
Solve[(1 1 1 1).(○ ○ ○ ●
                 ● ○ ● ●
                 ● ● ● ●
                 ● ● ● ●) == (99 60 73 x), x]
```

{{x → 112}}

Indeed, you can solve for all symbols if you want.

```
Solve[(1 1 1 1).⎛○ ○ ○ ●⎞ == (99 60 73 x), {x, ○, ●, ●}]
                ⎜● ○ ● ●⎟
                ⎜● ● ● ●⎟
                ⎝● ● ● ●⎠
```

$\{\{x \to 112, \bigcirc \to 2, \bullet \to 15, \bullet \to 41\}\}$

■ Integral Equations

Inhomogeneous Fredholm integral equations of the second kind,

$$f(x) = g(x) + \lambda \int_a^b \kappa(x, t)\, f(t)\, dt, \quad a \le x \le b, \tag{1}$$

involve solving for $f(x)$, where $\kappa(x, t)$ is the integral kernel, λ is a parameter, and $g(x)$ is the inhomogeneous term [1, 2]. *Mathematica* does not include built-in routines or packages for integral equations. However, it is straightforward to implement quite general methods for solving such equations.

As a specific example, consider Love's equation

$$f(x) = 1 + \frac{1}{\pi} \int_{-1}^{1} \frac{f(t)}{(x-t)^2 + 1}\, dt, \tag{2}$$

which arises in electrostatics [3, 4]. An excellent approximation to the solution $f(x)$ can be found in the form of a *Chebyshev-series*

$$f(x) = \sum_{i=0}^{n} c_i\, T_{2i}(x). \tag{3}$$

The coefficients c_i for $i = 0, 1, \ldots, n$, are determined by solving numerically the simultaneous linear algebraic equations that result by evaluating the integral equation (2) at the *Chebyshev points* $x_j = \cos\left(\frac{j\pi}{2n}\right)$ for $j = 0, 1, \ldots, 2n$, using the Chebyshev-series (3) [5, 6].

Here is a direct implementation for $n = 4$. The Chebyshev points are calculated using arbitrary-precision arithmetic.

```
n = 4; xs = N[Cos[Range[0, 2 n] π / (2 n)], 20]
```

{1.0000000000000000000, 0.92387953251128675613,
0.70710678118654752440, 0.38268343236508977173, 0,
−0.38268343236508977173, −0.70710678118654752440,
−0.92387953251128675613, −1.0000000000000000000}

Instead of computing the points using **Table**, we have used the **Listable** attribute of **Cos**.

We introduce the undetermined coefficients c_i.

```
cs = Thread[c_Range[0,n]]
```

$\{c_0, c_1, c_2, c_3, c_4\}$

We evaluate the left-hand side of equation (2) at the Chebyshev points.

```
lhs = cs.Table[T_{2 i}(xs), {i, 0, n}];
```

Note that we have replaced the sum in the Chebyshev-series (3) by a dot product and have used the **Listable** attribute of **ChebyshevT** to compute the value at *all* the Chebyshev points at once.

```
Attributes[ChebyshevT]
```

{Listable, NumericFunction, Protected, ReadProtected}

To compute the right-hand side of equation (2) we could use a Chebyshev quadrature formula [5, 6]. Instead we use **NIntegrate** for simplicity, again using arbitrary-precision arithmetic.

```
rhs = 1 + 1/π cs.Table[NIntegrate[Evaluate[T2i(t)/((xs - t)^2 + 1)],
        {t, -1, 1}, WorkingPrecision → 20], {i, 0, n}];
```

Solving the resulting simultaneous linear algebraic equations we obtain the coefficients c_i for $i = 0, 1, \ldots, n$.

```
Solve[lhs == rhs, cs]
```

$$\{\{c_0 \to 1.774444805,\ c_1 \to -0.1400430745,\ c_2 \to 0.004962498177,\ c_3 \to 0.0003746723341,\ c_4 \to -0.00004368650886\}\}$$

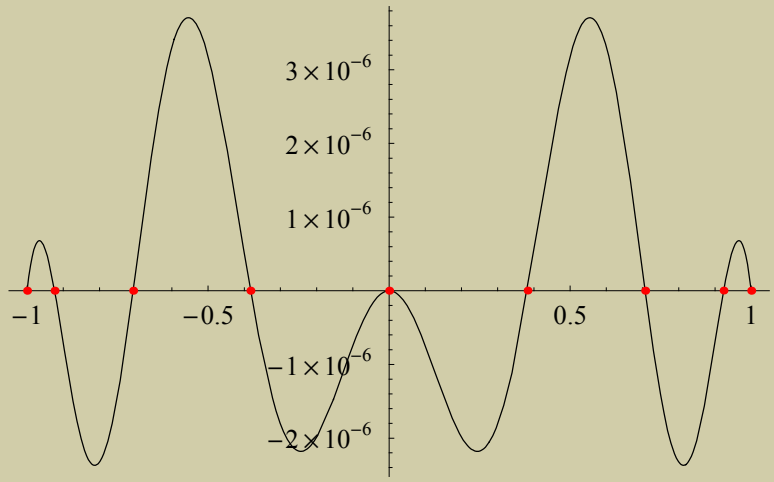
The solution $f(x)$ is obtained from the Chebyshev-series (3).

```
f(x_) = Σ_{i=0}^{n} c_i T2i(x) /. First[%]
```

$$-0.1400430745\,(2\,x^2 - 1) + 0.004962498177\,(8\,x^4 - 8\,x^2 + 1) + 0.0003746723341\,(32\,x^6 - 48\,x^4 + 18\,x^2 - 1) - 0.00004368650886\,(128\,x^8 - 256\,x^6 + 160\,x^4 - 32\,x^2 + 1) + 1.774444805$$

To see how well the Chebyshev-series solution satisfies Love's equation (2), we substitute it back into the equation and plot the difference of the left- and right-hand sides. The Chebyshev points are also displayed and it is apparent that the *minimum error* in the Chebyshev-series solution occurs at these points.

```
Plot[1 + 1/π NIntegrate[f(t)/((x - t)^2 + 1), {t, -1, 1}] - f(x),
    {x, -1, 1}, Epilog → {Hue[0], AbsolutePointSize[3],
      Point /@ (xs /. x_?NumberQ → {x, 0})}, PlotRange → All];
```



As an aside, we can get a bound for the *maximum error* directly from the **PlotRange** of this plot. We could use this error-bound in a recursive procedure so as to find a solution for a given maximum error.

```
Max[Abs[Last[PlotRange[%]]]]
```

3.85916×10^{-6}

■ Matrix Differential Operator

The curl is often represented as a differential operator.

```
∇ /: ∇×{f_, g_, h_} := {∂h/∂y - ∂g/∂z, ∂f/∂z - ∂h/∂x, ∂g/∂x - ∂f/∂y}
```

Here is an example.

```
∇×{f(y), g(z), h(x)}
```

$$\{-g'(z), -h'(x), -f'(y)\}$$

Alternatively, curl can be represented as a matrix differential operator, C.

$$C = \begin{pmatrix} 0 & -\partial_z & \partial_y \\ \partial_z & 0 & -\partial_x \\ -\partial_y & \partial_x & 0 \end{pmatrix}.$$

Using pure function notation (see Section 2.2.5 of *The Mathematica Book*) this reads as follows.

$$C := \begin{pmatrix} 0\ \& & -\partial_z\#\ \& & \partial_y\#\ \& \\ \partial_z\#\ \& & 0\ \& & -\partial_x\#\ \& \\ -\partial_y\#\ \& & \partial_x\#\ \& & 0\ \& \end{pmatrix}$$

However, we cannot **Dot** this operator by a vector to compute the curl because each element resulting from the dot product operation must be *composed* instead of being multiplied. **Inner** generalizes **Dot** and allows us to do what we want. Here we use the **CenterDot** operator (see Section 3.10.4 of *The Mathematica Book*) to define a suitable shorthand notation.

```
a_ · b_ := Inner[#1[#2] &, a, b]
```

Now we can compute the curl using the matrix differential operator.

```
C · {f(y), g(z), h(x)}
```

$$\{-g'(z), -h'(x), -f'(y)\}$$

As a second example, Kurasov and Naboko [7] study the essential spectrum of the following matrix differential operator:

$$\mathcal{D} = \begin{pmatrix} -\frac{d}{dx}\,\rho(x)\,\frac{d}{dx} + q(x) & \frac{\beta(x)}{x}\,\frac{d}{dx} \\ -\frac{\beta(x)}{x}\,\frac{d}{dx} & \frac{m(x)}{x^2} \end{pmatrix}. \tag{4}$$

The four differential operators in the matrix $\mathcal{D}$ can be implemented as follows.

$$\mathcal{D}_{1,1}[\mathrm{f_}] := -\partial_x\left(\rho(x)\,\frac{\partial f}{\partial x}\right) + q(x)\,f$$

$$\mathcal{D}_{1,2}[\mathrm{f_}] := \partial_x\left(\frac{\beta(x)}{x}\,f\right)$$

$$\mathcal{D}_{2,1}[\mathrm{f_}] := -\frac{\beta(x)}{x}\,\frac{\partial f}{\partial x}$$

$$\mathcal{D}_{2,2}[\mathrm{f_}] := \frac{m(x)}{x^2}\,f$$

Let us modify the action of the **CenterDot** operator by applying **HoldForm** to the first argument, so that it does not evaluate the operators.

```
a_ · b_ := Inner[HoldForm[#1][#2] &, a, b]
```

We now can examine the application of the matrix differential operator to a vector or matrix before we explicitly evaluate the matrix entries using **ReleaseHold**.

$$\mathrm{Table}[\mathcal{D}_{i,j}, \{i, 2\}, \{j, 2\}] \cdot \begin{pmatrix} f(x) \\ g(x) \end{pmatrix}$$

$$\begin{pmatrix} \mathcal{D}_{1,1}[f(x)] + \mathcal{D}_{1,2}[g(x)] \\ \mathcal{D}_{2,1}[f(x)] + \mathcal{D}_{2,2}[g(x)] \end{pmatrix}$$

```
ReleaseHold[%]
```

$$\begin{pmatrix} f(x)\,q(x) - \frac{g(x)\,\beta(x)}{x^2} + \frac{\beta(x)\,g'(x)}{x} + \frac{g(x)\,\beta'(x)}{x} - f'(x)\,\rho'(x) - \rho(x)\,f''(x) \\ \frac{g(x)\,m(x)}{x^2} - \frac{\beta(x)\,f'(x)}{x} \end{pmatrix}$$

As an aside, if we use the **SmallCircle** operator to denote **Composition**, then formal products of differential operators can be computed quite elegantly.

```
SmallCircle = Composition;
```

For example, the action of the product of the antidiagonal elements in $\mathcal{D}$ can be computed as follows.

$$(\mathcal{D}_{1,2} \circ \mathcal{D}_{2,1})[f(x)]$$

$$\frac{2 f'(x) \beta(x)^2}{x^3} - \frac{f''(x) \beta(x)^2}{x^2} - \frac{2 f'(x) \beta'(x) \beta(x)}{x^2}$$

■ References

[1] C. T. H. Baker, *The Numerical Treatment of Integral Equations*, Oxford: Clarendon Press, 1977 pp. 358–60.

[2] C. E. Pearson, *Handbook of Applied Mathematics*, New York: Van Nostrand Reinhold, 1990.

[3] E. R. Love, “The Electrostatic Field of Two Equal Circular Co-axial Conducting Disks,” *Quarterly Journal of Mechanics and Applied Mathematics*, **2**, 1949 pp. 428–451.

[4] L. Fox and E. T. Goodwin, “The Numerical Solution of Non-singular Linear Integral Equations,” *Philosophical Transactions of the Royal Society of London*, Series A, **245**, 1953 pp. 501–534.

[5] C. W. Clenshaw and A. R. Curtis, “A Method for Numerical Integration on an Automatic Computer,” *Numerische Mathematik*, **2**, 1960 pp. 197–205.

[6] S. E. El-gendi, “Chebyshev Solution of Differential, Integral and Integro-differential Equations,” *The Computer Journal*, **12**, 1969 pp. 282–287. http://www3.oup.co.uk/computer_journal/hdb/Volume_12/Issue_03.

[7] P. Kurasov and S. Naboko, “On the Essential Spectrum of a Class of Singular Matrix Differential Operators. I Quasiregularity Conditions and Essential Self-adjointness,” *Research Reports in Mathematics, Department of Mathematics, Stockholm University*, **12**, 2001. http://www.matematik.su.se/reports/2001/12.