Matrix Programming

Introduction

Studying patients with Babbage's Syndrome

How effective are available treatments?

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>2.5</td>
<td>3.5</td>
<td>3.0</td>
</tr>
<tr>
<td>Mary</td>
<td>3.0</td>
<td>1.5</td>
<td>3.0</td>
</tr>
<tr>
<td>Zura</td>
<td>2.5</td>
<td>2.0</td>
<td>5.5</td>
</tr>
</tbody>
</table>

How similar are patients' responses?
Can we use similarity to recommend treatments?
Answer these questions with matrix operations
How to implement them in software?
Option 1: write loops
- Makes programs many times longer than the corresponding mathematics
- And it's hard code to debug...
- ...and tune

Option 2: use libraries written in low-level, high-performance languages like Fortran and C
- Someone else has written, debugged, and tuned all the loops
- But the interface is...awkward

SUBROUTINE CAXPY(N, CA, CX, INCX, CY, INCY)
Option 3: use a high-level language like MATLAB
Or a library like Python's NumPy
Present a data-parallel programming model
– Operate on entire arrays at once
– No loops!
Hide details of optimizations
– Particularly differences between machines
All provide basically the same features
– Often wrappers around the same underlying libraries

created by
Richard T. Guy
November 2010

Copyright © Software Carpentry 2010
This work is licensed under the Creative Commons Attribution License
See http://software-carpentry.org/license.html for more information.