

PS3 Programming

Week 6. Math Libraries
(Chap 16,17,18)

Outline

- Linear algebra
- FFT
- Multiprecision processing
- Monte Carlo methods
- Homework

LINEAR ALGEBRA

Linear algebra

- Vector library
 - dot product, cross product, vector length, ...
- Matrix library (4x4)
 - Transport, scale, multiply, cast, split, rotation
- Large matrix library
 - Transport, scale, multiply, madd, nmsub, LU decomposition, solve triangular, solve general
- BLAS

Basic Linear Algebra Subroutines

- Over 50 subroutines
 - <http://www.netlib.org/blas>
 - Level 1: dot product, rotation, vector length...
 - Level 2: matrix-vector multiplication, Solve triangular system, ...
 - Level 3: matrix-matrix multiplication, ...
- It's for PowerPC and the SDK's BLAS (libblas)
 - PPU has a complete library, but only few in SPU.

Multiprocessor matrix multiplication

- In `/opt/cell/sdk/src/demos/matrix_mul`
- Algorithm
 - PPU creates the matrix and SPU contexts
 - SPUs receive information and load matrix blocks
 - SPUs process the block matrix multiplication and then store back to memory

FAST FOURIER TRANSFORM

Do you remember what FFT is?

- Transform data from time domain to frequency domain
- Used in DSP, sampling, and many applications.
- A special matrix vector multiplication that can be done in $O(n \log n)$: 1D FFT
- Four floats for two complex numbers

in[1].real

in[1].imag

in[2].real

in[2].imag

Ex: One dimensional problem

```
void fft_1d_r2 (vector float *out,  
               vector float *in,  
               vector float *W,  
               int log2_size)
```

- `log2_size` must be ≥ 5 (`size` ≥ 32)
- `W` is defined as

```
for (i=0; i<N/2; i+=2) {  
    W[i]    = cos((double) i*PI/N);  
    W[i+1] = sin((double) i*PI/N);  
}
```

The FFT library

- In `/opt/cell/sdk/prototype/libfft`
 - PPU's file is in `/opt/cell/sdk/prototype/usr`
 - SPUs' file is in `/opt/cell/sdk/prototype/spu`
- Functions
 - 1D problem: `fft_1d_initialize`, `fft_1d_perform`, `fft_1d_terminate`
 - 2D problem: `fft_2d_initialize`, `fft_2d_perform`, `fft_2d_terminate`

MULTIPRECISION PROCESS AND MONTE CARLO METHODS

Multiprecision process

- In the library: libmpm
- Big integer, big numbers
- Applications in encryption, numerical analysis, ...
- Basic functions: abs, add, sub, square, multiply, madd, cmpeq, cmpge, cmpgt, ...
- Division and modular: div, mod, mod_exp, ...

Monte Carlo methods

- In the library: `libmc_rand`
- Monte Carlo methods
 - Generate a series of random inputs
 - Send the random inputs into the simulated system
 - Analyze output statistics
- Random number generation

HOMEWORK

Homework

- Read chap 16,17,18
- Try the example of distributed matrix-matrix multiplication