Problem2

September 30, 2016

In [3]:
```
"""
‘C = matmul(A,B)‘ returns the matrix-matrix product of C=A*B
function matmul(A,B)
end
"""
function matmul(A,B)
    if ndims(A) < 2
        A_rows = length(A)
        A_cols = 1
    else
        A_rows, A_cols = size(A)
    end
    if ndims(B) < 2
        B_rows = length(B)
        B_cols = 1
    else
        B_rows, B_cols = size(B)
    end

    if A_cols != B_rows
        return "Error: inner dimensions must agree"
    end

    C = zeros(A_rows, B_cols)
    for ii=1:A_rows
        for jj=1:B_cols
            for kk=1:A_cols
                C[ii,jj] += A[ii,kk]*B[kk,jj]
            end
        end
    end

    return C
end
```

Out[3]:
```
matmul (generic function with 1 method)
```

In [3]:
```
@show matmul([1 2], [2; 1])
@show matmul([1 2; -2 4; 0 3], [-1; 1])
@show matmul([-1 1], [1 -2 0; 2 4 3])
@show matmul(3, 5)
t = pi/2
@show matmul([cos(t) -sin(t); sin(t) cos(t)],
    [cos(t) sin(t); -sin(t) cos(t)]);
```

```
matmul([1 2],[2;1]) = [4.0]
matmul([1 2;-2 4;0 3],[-1;1]) = [1.0
 6.0
 3.0]
matmul([-1 1],[1 -2 0;2 4 3]) = [1.0 6.0 3.0]
matmul(3,5) = [15.0]
matmul([cos(t) -(sin(t));sin(t) cos(t)],[cos(t) sin(t);-(sin(t)) cos(t)]) = [1.0 0.0
 0.0 1.0]
```

In [19]:
```
nMats = 50
nSamples = 50

N = linspace(10, 1000, nMats)
timediff = zeros(nMats, nSamples)
dt = zeros(nMats, nSamples)
dt2 = zeros(nMats, nSamples)
diffSols = zeros(nMats, nSamples)

for i=1:nMats
    M = round(Int,N[i])
    for j=1:nSamples
        A = randn(M, M)
        B = randn(M, M)
        dt[i,j] = @elapsed C = matmul(A,B);
        dt2[i,j] = @elapsed D = A*B;
        diffSols[i,j] = vecnorm(C-D,)
        timediff[i,j] = dt2[i,j]-dt[i,j]
    end
end
```

In [28]:
```
using Plots
plot(N, mean(dt,2), xlabel= "matrix size", ylabel="Time", title="Average time for matmul",label
plot!(N, mean(dt2,2),label="base")
```

In [29]:
```
maxdiff = zeros(50,1)
for i=1:50
    maxdiff[i], trash = findmax(diffSols[i,:])
end
using Plots
plot(N, maxdiff, xlabel= "matrix size", ylabel="Difference", title="Max different between base
```

In [12]:
```
"""
`matmul2`
========

Compute Matrix-matrix multiplication or MatMul faster

Functions
---------
- `C = matmul2(A,B)` returns the matrix-matrix product of C=A*B
"""
function matmul2(A,B)
    A_rows, A_cols = size(A)
    B_rows, B_cols = size(B)
    C = zeros(A_rows, B_cols)
```

```
            nAr= div(A_rows, 16)
            nBc= div(B_cols, 16)
            NAc = div(A_cols, 16)
            for ii=1:nAr
                I = (16*(ii-1)+1):(16(ii))
                for jj=1:nBc
                    J = (16*(jj-1)+1):(16(jj))
                    for kk=1:NAc
                        K = (16*(kk-1)+1):(16(kk))
                        C[I,J] += A[I,K]*B[K,J]
                    end
                end
            end

            return C
        end
```

Out[12]: matmul2 (generic function with 1 method)

In [16]:
```
        nMats = 10
        nSamples = 50

        N = linspace(16, 1600, nMats)
        timediff = zeros(nMats, nSamples)
        dt = zeros(nMats, nSamples)
        dt2 = zeros(nMats, nSamples)

        for i=1:nMats
            M = round(Int,N[i])
            for j=1:nSamples
                A = randn(M, M)
                B = randn(M, M)
                dt[i,j] = @elapsed C = matmul(A,B);
                dt2[i,j] = @elapsed D = matmul2(A,B);
            end
        end
```

In [23]:
```
        using Plots
        plot(N, mean(dt,2), xlabel= "matrix size", ylabel="Time", title="Average time for matmul",label
        plot!(N, mean(dt2,2),label="matmul2")
        savefig("problem2partD")
```

In [ ]: