CE722/822 = MSIM-695: CLUSTER PARALLEL COMPUTING

Tentative Course Syllabus
****

Semester: Summer 2010 (Session 5 = 7 weeks, June 28 through August 15, 2010)
Date (lectures): Monday & Wednesday, 6:00pm - 09:45pm
Place:Room # 219, GORNTO Bldg.

This televised (T.V.) course will have BROADCASTING/TELEVISING lectures to other ODU sites (in Hampton, and Newport News ...) to accommodate those course participants who do NOT want to drive to ODU main campus !

Instructor: Professor Duc T. Nguyen
1319 ECSB
Tel= (757) 683-3761, Fax= -5354, Email= dnguyen@odu.edu
Res Tel= Res Fax= (757) 465-9110

ODU-OCCS (parallel computer accounts): Ruben (Email= rigloria@odu.edu; Tel# 757-683-4842)

Required Text:

Book: "Finite Element Methods: Parallel-Sparse Statics and Eigen-Solutions"
Author: Duc T. Nguyen
Publisher: Springer Publisher (April'2006), Hard-cover, 542 pages, US $99.00
ISBN #: 0387293302

Basic Course Lectures Materials can be downloaded (and printed by your own) from the following ODU web-site address:

http://www.lions.odu.edu/~skadi002/ (select CEE-722/822 Course Materials)

Course Descriptions: Efficient numerical algorithms and detailed computer implementation for solving large-scale engineering/ science (in civil /mechanical /aerospace /electrical engineering /chemistry/biology ...) applications on serial and parallel computer platforms.

Pre-requisites: Self-explained course. Minimum (undergraduate) knowledge in linear algebra, and some basic working knowledge in computer coding (C++, or FORTRAN etc...) is helpful (but NOT a requirement).

Course's Features: No tests, no final exam (homeworks and "case study" computer project(s) are emphasized and required !).

- Improving your chance to get research oriented jobs in the emerging areas of "HIGH-PERFORMANCE PARALLEL SCIENTIFIC COMPUTING".

- Increasing your understanding of numerical aspects behind the popular, commercialized finite element codes, such as MSC-NASTRAN, ABUQUS, ANSYS etc... State-of-the art algorithms are thoroughly explained.

- Hands-on practical applications (including practical, NASA data) are emphasized.

- Course participants will have access to ODU/SUN/LIONS2 with 1-32 parallel processors). Option for MPI/Pro on your own LAPTOP_PC is also discussed.

- Officially registered participants to this course may select either "AUDIT" mode (just to learn the subjects, no class grades are assigned), or in the usual "class GRADES" mode.

"TENTATIVE" TOPICS:

[0] Review of FORTRAN_90 language (GNU/F77 is available, 125 KAUF) (COMPACT VISUAL FORTRAN, commercialized) (free MPI/FORTRAN/C/C++ under LINUX pc O.S.) Transparencies' page numbers: 239,240

HW0:
For "even" year - Given the class' test scores {90,15,34,67,89,55,77,100,85,79}Write a computer program (using MATLAB, MathCAD, MATHEMATICA, Maple, EXCEL, FORTRAN, C++ ...) to find the "lowest, highest, and average" scores.

For "odd" year – Given the function f(x) = 2x^2 -8x + 6, write a computer program (using MATLAB, MathCAD, MATHEMATICA, Maple, EXCEL, FORTRAN, C++ …) to compute the integral of f(x), between the 2 limits [a=0; b=3], by computing the total area under the function f(x), and with dx=0.002.

[1] Vectorized code (loop-unrolling & vector unrolling techniques). Simple Example: Efficient "matrix * vector" computation Transparencies' page numbers: 1-11

Text Book Reading Materials: Portions of Chapter 2; pages 75-77, 81-83

HW1 (see Nguyen's Book)
For "even" year - Chapter 2 (problems # 2.1-2.3)

For "odd" year - Chapter 2 (problems # 2.1-2.3), with A(i,j) = 8i – j and x(j) = j + 1 and n = 6000.

[2] Parallel-Vector Hardware/Software Tools Available for "HANDS-ON" Applications:
---> Using Sun (LIONS2, cluster, with 1-32 nodes, supported by ODU/OOCS)

---> Developing/Debuging Parallel MPI Application Code on Your Own LapTop
---> MPI (Message Passing Interface) FORTRAN/C++ are supported.
Transparencies' page numbers: 223

Text Book Reading Materials: Portions of Chapter 2; pages 63-75, 95-103

HW2 (see Nguyen's Book)
For "even" year - Chapter 2 (problem # 2.7)

For "odd" year - Chapter 2 (problem # 2.7), with the modified data as indicated in HW1 ("odd" year)

Using 1 and 4 processors with MPI on your PC LapTop/DeskTop (or using ODU LIONS2 processors).

[3] Some basic definitions of parallel performance.
---> Simple Example: MPI/FORTRAN Parallel computation of "pai=3.14159.."
---> Simple Example: MPI/FORTRAN Parallel "matrix * matrix" computation
Transparencies' page numbers: 12-17

Text Book Reading Materials: Portions of Chapter 2; pages 63-75, 95-103
HW3 (see Nguyen's Book)
For "even" year - Chapter 2 (problem # 2.8)

For "odd" year - Chapter 2 (problem # 2.8), with the modified requirement to print "N" numbers in "DECREASING" order.

Using 1 and 4 processors with MPI on your PC LapTop/DeskTop (or using ODU LIONS2 processors)

[4] Parallel Vector Equation-Solver Algorithm and Software:
---> skyline & variable sparse storage schemes for the (stiffness) matrix
Transparencies' page numbers: 18-25, 156,157

---> Choleski and Gauss algorithms
Transparencies' page numbers: 26-30, 33,34, 43,44,46,47, 52-58)

---> "Sparse" algorithms: Positive/Negative/Indefinite systems of equations
Transparencies' page numbers: 151-161, 164, 167,168

---> "Indefinite" systems of linear equations
Transparencies' page numbers: 168.1-168.11

TextBook Reading Materials: Portions of Chapter 3
HW4 (see Nguyen's Book)

For “even” year - Chapter 3 (problems # 3.1-3.5) + "extra, special" HW4's problem

Given the following matrix equation: [K] * {x} = {b}, where:

$$[K] = \begin{array}{cccccc} 112 & 7 & 0 & 0 & 0 & 2 \\ 7 & 110 & 5 & 4 & 3 & 0 \\ 0 & 5 & 88 & 0 & 0 & 1 \\ 0 & 4 & 0 & 66 & 0 & 0 \\ 0 & 3 & 0 & 0 & 44 & 0 \\ 2 & 0 & 1 & 0 & 0 & 11 \end{array} \qquad \begin{array}{c} 1 \\ 0 \\ -1 \\ 2 \\ 3 \\ -2 \end{array}$$

and

$$\{b\} = \begin{array}{c} 108 \\ 19 \\ -90 \\ 132 \\ 132 \\ -21 \end{array}$$

Questions:

(1) Prove that the above given 6x6 matrix [K] is Symmetric Positive Definite (SPD) ??
(2) Using "truely sparse" storage schemes, determine the numerical values for the arrays {ia}, {ja}, {ad}, and {an}, corresponding to [K] ??
(3) Without any actual computation, find (& explain) the LOCATIONS of ALL fill-in terms, based on the data given in the above matrix [K] ??
(4) Assuming the degree-of-freedom (or dof) between the NEW and OLD numbering systems is given as:

	(new=1)		(old=6)
	(new=2)		(old=5)
	(new=3)		(old=4)
iperm	(new=4)	=	(old=3)
	(new=5)		(old=2)
	(new=6)		(old=1)

Find the NEW (or modified) matrix [K*] and the new rhs vector {b*} ??
(5) Using "truely sparse" storage schemes, determine the numerical values for the arrays {ia}, {ja}, {ad}, and {an}, corresponding to [K*] ??
(6) Without any actual computation, find (& explain) the LOCATIONS of ALL fill-in terms, based on the data given in the above matrix [K*] ??
(7) Repeat part (5), but coresponding to [K*] after including fill-in terms ??
(8) Using the Choleski (or L D L_transpose, or LU) method, solve for {x*}from the following systems of eqs.

[K*] * {x*} = {b*}

(9) Using the information provided by iperm(new) = (old), shown in item (4), rearrange the solution vector {x*} in order to get the original unknown vector {x} ??

For "odd" year - Chapter 3 (problems # 3.1-3.5), with a different/given matrix A as following:

$$[A] = \begin{bmatrix} 112 & 7 & 0 & 0 & 0 & 2 \\ 7 & 110 & 5 & 4 & 3 & 0 \\ 0 & 5 & 88 & 0 & 0 & 1 \\ 0 & 4 & 0 & 66 & 0 & 0 \\ 0 & 3 & 0 & 0 & 44 & 0 \\ 2 & 0 & 1 & 0 & 0 & 11 \end{bmatrix}$$

+ a different "extra, special" HW4's problem:

$$[K] = \begin{bmatrix} 11 & 0 & 0 & 1 & 0 & 2 \\ 0 & 44 & 0 & 0 & 3 & 0 \\ 0 & 0 & 66 & 0 & 4 & 0 \\ 1 & 0 & 0 & 88 & 5 & 0 \\ 0 & 3 & 4 & 5 & 110 & 7 \\ 2 & 0 & 0 & 0 & 7 & 112 \end{bmatrix} \qquad \begin{matrix} 1 \\ 0 \\ -1 \\ 2 \\ 3 \\ -2 \end{matrix}$$

and

$$\{b\} = \begin{Bmatrix} 9 \\ 15 \\ -54 \\ 192 \\ 322 \\ -201 \end{Bmatrix}$$

Questions:

(1) Prove that the above given 6x6 matrix [K] is Symmetric Positive Definite (SPD) ??
(2) Using "truely sparse" storage schemes, determine the numerical values for the arrays {ia}, {ja}, {ad}, and {an}, corresponding to [K] ??
(3) Without any actual computation, find (& explain) the LOCATIONS of ALL fill-in terms, based on the data given in the above matrix [K] ??
(4) Assuming the degree-of-freedom (or dof) between the NEW and OLD numbering systems is given as:

(new=1)	(old=6)
(new=2)	(old=5)

	(new=3)		(old=4)
iperm	(new=4)	=	(old=3)
	(new=5)		(old=2)
	(new=6)		(old=1)

Find the NEW (or modified) matrix [K*] and the new rhs vector {b*} ??
(5) Using "truely sparse" storage schemes, determine the numerical values for the arrays {ia}, {ja}, {ad}, and {an}, corresponding to [K*] ??
(6) Without any actual computation, find (& explain) the LOCATIONS of ALL fill-in terms, based on the data given in the above matrix [K*] ??
(7) Repeat part (5), but coresponding to [K*] after including fill-in terms ??
(8) Using the Choleski (or L D L_transpose, or LU) method, solve for {x*}from the following systems of eqs.

[K*] * {x*} = {b*}
(9) Using the information provided by iperm(new) = (old), shown in item (4), rearrange the solution vector {x*} in order to get the original unknown vector {x} ??

[5] Iterative Equation Solvers
---> Pre-conditioned Conjugate Gradient Algorithms (symmetrical matrix)
Transparencies' page numbers: 230

---> Various Pre-Conditioned Schemes
Transparencies' page numbers: 230

TextBook Reading Materials: Portions of Chapter 6; pages 379-404
HW5 (see Nguyen's Book)
For "even" year - Chapter 6 (problems # 6.2)

For "odd" year - Chapter 6 (problems # 6.2), with the following different data:

	11	0	0	1	0	2	1
	0	44	0	0	3	0	0
	0	0	66	0	4	0	-1
[A] =	1	0	0	88	5	0	2
	0	3	4	5	110	7	3
	2	0	0	0	7	112	-2

and

	9			0			2	0	0	0	0	0
	15			1			0	2	0	0	0	0
	-54			0			0	0	2	0	0	0
{b} =	192	;	{xo} =	2	;	[B] =	0	0	0	2	0	0
	322			0			0	0	0	0	2	0
	-201			3			0	0	0	0	0	1

[6] Parallel Domain Decomposition (DD) Formulation
Transparencies' page numbers: 193-195, 230.1-230.6

TextBook Reading Materials: Portions of Chapter 6; pages 379-404
HW6 (see Nguyen's Book)
For "even" year - Chapter 6 (problems # 6.1)

Extra HW Problem:
Given the following DD matrices [A] * {x} = {b}, where

	2	-1	0	0	0	0	1	-1		1
	-1	2	-1	0	0	0	0	2		2
	0	-1	1	0	0	0	2	1		3
[A] =	0	0	0	2	-1	0	1	1	{b} =	3
	0	0	0	-1	2	-1	2	-1		1
	0	0	0	0	-1	1	2	2		4
	1	0	2	1	2	2	4	-1		11
	-1	2	1	1	-1	2	-1	3		6

(a) Identify Kii(-,-) for sub_domain r=1 ??
(b) Identify Kii(-,-) for sub_domain r=2 ??
(c) Identify Kib(-,-) for sub_domain r=1 ??
(d) Identify Kib(-,-) for sub_domain r=2 ??
(e) Identify Summation of Kbb(-,-) for r=1 & 2 ??
(f) Find boundary unknown vector { Xb(-) } ??
(g) Find interior unknown vector { Xi(-) } for r=1 ??
(h) Find interior unknown vector { Xi(-) } for r=2 ??

For "odd" year - Chapter 6 (problems # 6.1), with the following changes:
E = 40,000 K/in^2 (for all members)
A = 3 in^2 (for all members)
Substructure 1 consists of members 1, 2, 7 and nodes 1, 2, 6 (see Figure 6.2).
Substructure 2 consists of members 3, 4, 5, 6, 8, 9 and nodes 2, 3, 4, 5, 6

Extra HW Problem:
Given the following DD matrices [A] * {x} = {b}, where

$$[A] = \begin{bmatrix} 3 & -1 & 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 3 & -1 & 0 & 0 & 0 & 0 & 2 \\ 0 & -1 & 1 & 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 3 & -1 & 0 & 1 & 1 \\ 0 & 0 & 0 & -1 & 3 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 & 2 & 2 \\ 1 & 0 & 2 & 1 & 2 & 2 & 5 & -1 \\ -1 & 2 & 1 & 1 & -1 & 2 & -1 & 1 \end{bmatrix} \qquad \{b\} = \begin{Bmatrix} 1 \\ -2 \\ 3 \\ -3 \\ 1 \\ -4 \\ 11 \\ -6 \end{Bmatrix}$$

(a) Identify Kii(-,-) for sub_domain r=1 ??
(b) Identify Kii(-,-) for sub_domain r=2 ??
(c) Identify Kib(-,-) for sub_domain r=1 ??
(d) Identify Kib(-,-) for sub_domain r=2 ??
(e) Identify Summation of Kbb(-,-) for r=1 & 2 ??
(f) Find boundary unknown vector { Xb(-) } ??
(g) Find interior unknown vector { Xi(-) } for r=1 ??
(h) Find interior unknown vector { Xi(-) } for r=2 ??

[7] Parallel "dense" equation solver (if time permits)
Transparencies' page numbers: 196-208

Text Book Reading Materials: Portions of Chapter 2; pages 77-95
HW7 (see Nguyen's Book) : Chapter 2 (not applicable !)

==
Possible "case study" Class Projects (No Tests, No Final Exam !):
Each team (1 or 2 students) can select any of the mutually agreeable topics
==

IMPORTANT:

Due to the requirement of SACS (for accreditation purpose), students registered at the higher level courses (such as CEE-8xx, or MSIM-8xx, or MAE-8xx " = formerly known as ME-8xx, or AE-8xx", ECE-8xx etc….) will be required to do additional work (as comparing to those students registered under lower level courses (such as CEE-7xx, or MSIM-7xx, etc….).