

Десятая независимая научно-практическая
конференция «Разработка ПО 2014»

23 - 25 октября, Москва



Портирование приложения с гранично- элементным моделированием пороупругой динамики на системы с распределенной памятью



Нижегородский
государственный
университет
им. Н. И. Лобачевского

Воробцов Игорь
НИИ механики ННГУ



Метод Граничного Элемента (МГЭ)

- Может быть Метод Конечных Элементов (МКЭ)?
 - нет, всё верно – МГЭ
 - численный метод для решения краевых задач
- МГЭ
 - 3 место по использованию после МКЭ и МКР
 - Более эффективный для ряда задач
- Активно используется в НИИ ННГУ

Математическая модель и гранично-интегральные уравнения

$$G\hat{u}_{i,jj} + \left(K + \frac{1}{3}G\right)\hat{u}_{j,ij} - (\alpha - \beta)\hat{p}_{,i} - s^2(\rho - \beta\rho_f)\hat{u}_i = 0,$$

$$\frac{\beta}{s\rho_f}\hat{p}_{,ii} - \frac{\varphi^2 s}{R}\hat{p} - (\alpha - \beta)s\hat{u}_{i,i} = 0,$$

$$\beta = \varphi^2 s + s^2 k(\rho_a + \varphi\rho_f),$$

$$\bar{u}(x, s) = \tilde{u}, \quad x \in \Gamma^u,$$

$$\bar{t}_n(x, s) = \tilde{t}_n, \quad x \in \Gamma^\sigma,$$

$$\begin{bmatrix} \hat{u}_j \\ \hat{p} \end{bmatrix} = \int_{\Gamma} \begin{bmatrix} \hat{U}_{ij}^s & -\hat{P}_j^s \\ \hat{U}_i^f & -\hat{P}^f \end{bmatrix} \begin{bmatrix} \hat{t}_i \\ \hat{q} \end{bmatrix} d\Gamma - \int_{\Gamma} \begin{bmatrix} \hat{T}_{ij}^s & -\hat{Q}_j^s \\ \hat{T}_i^f & -\hat{Q}^f \end{bmatrix} \begin{bmatrix} \hat{u}_i \\ \hat{p} \end{bmatrix} d\Gamma.$$



Математическая модель и гранично-интегральные уравнения



$$\hat{P}_i^s = O(r^0),$$

$$\hat{U}_i^f = O(r^0),$$

$$\hat{U}_{ij}^s = \frac{1+\nu}{8\pi\lambda(1-\nu)} \left\{ r_{,i}r_{,j} + \delta_{ij}(3-4\nu) \right\} \frac{1}{r} + O(r^0),$$

$$\hat{P}^f = \frac{\rho_f P}{4\pi\beta} \frac{1}{r} + O(r^0),$$

$$\hat{Q}_j^s = \frac{1+\nu}{8\pi\lambda(1-\nu)} \left\{ \alpha(1-2\nu)(r_{,n}r_{,j} - n_j) - 2\beta(1-\nu)(r_{,n}r_{,j} + n_j) \right\} \frac{1}{r} + O(r^0)$$

$$\hat{T}_i^f = \frac{\rho_f P^2}{8\pi\beta} \left\{ (\alpha - \beta) \frac{1-2\nu}{1-\nu} r_{,i}r_{,j} + n_i \frac{\alpha + \beta(1-2\nu)}{1-\nu} \right\} \frac{1}{r} + O(r^0)$$

$$\hat{T}_{ij}^s = \frac{-[(1-2\nu)\delta_{ij} + 3r_{,i}r_{,j}]r_{,n} + (1-2\nu)(r_{,j}n_i - r_{,i}n_j)}{8\pi(1-\nu)r^2} + O(r^0)$$



$$\hat{Q}^f = -\frac{r_{,n}}{4\pi\lambda^2} + O(r^0),$$

Что для разработчиков?

- Портировать приложение на кластер
 - Компиляция
 - Успешный запуск и расчет тестовых задач
 - Сохранение численных результатов
 - Загрузка ядер
 - Масштабируемость

Приложение

- Создавалось многие годы на языке Фортран
- Только проект VS на Windows
- Компилируется Intel Fortran v.11.1
 - Используется MKL
 - Используется IMSL
- 139 файлов (Fortran 77 и Fortran 90)

Первые шаги

- Перенос на Linux
 - Нет готовых средств для портирования из проекта Intel Fortran в VS на Линукс
 - Зависимости
 - Сложности с созданием makefil'а «руками»
 - Скрипт: <http://www.fortran.com/makemake.perl>
 - Циклические зависимости
 - Наконец-то, собралось! Но были и проблемы...

Проблемы

- Ошибки:

```
$ ifort -c test.f90
test.f90(8): warning #7319: This argument's data type is incompatible with this intrinsic procedure; procedure
assumed EXTERNAL.      [DSQRT]
      R=DSQRT((X1FP-X1SP)*(X1FP-X1SP)+(X2FP-X2SP)*(X2FP-X2SP)+(X3FP-X3SP)*(X3FP-X3SP))
test.f90: catastrophic error: **Internal compiler error: segmentation violation signal raised**
```

- Компилятор на Linux более «требователен»

- MKL

- Факторизация с `zgetrf` не выполняется при несоответствии типа (`complex` вместо `double complex`)
 - Заменена на `cgetrf`
 - На Windows все работало

FP модель

$$2^{26} - 2^{26} + 1 = 1$$

$$(2^{26} - 2^{26}) + 1 = 1 \text{ (верно согласно IEEE)}$$

$$2^{26} - (2^{26} - 1) = 0 \text{ (верно согласно IEEE)}$$

| опция | безопасность | точность | FMA | окружение | исключения |
|-----------------------------|--------------|------------|-----------|-----------|------------|
| precise source | варьируется | код код | да | нет | нет |
| strict | варьируется | код | нет | да | да |
| fast=1 (по умолчанию) | небезопасно | неизвестно | да | нет | нет |
| fast=2 | небезопасно | неизвестно | да | нет | нет |
| except except- | не меняет | код код | не меняет | не меняет | да нет |

```
4.012345678901111
4.012345678902222
4.012345678902222
4.012345678901111
4.012345678902222
4.012345678901111
4.012345678901111
4.012345678901111
4.012345678902222
4.012345678902222
4.012345678901111
4.012345678902222
4.012345678901111
4.012345678902222
4.012345678902222
4.012345678901111
```

...

Параллельность

- Часть распараллелено через секции OpenMP
- Внешний цикл
 - static scheduling
- Intel Vtune™ Amplifier XE показал почти идеальную загрузку ядер.
- Дальше - MPI

