Unstructured Adaptive (UA) NAS Parallel Benchmark, Version 1.0

Huiyu Feng[*], Rob F. Van der Wijngaart[*], Rupak Biswas[†], Catherine Mavriplis[‡]
NASA Ames Research Center
Moffett Field, CA 94035-1000

NASA Technical Report NAS-04-006

July 2004

Abstract

We present a complete specification of a new benchmark for measuring the performance of modern computer systems when solving scientific problems featuring irregular, dynamic memory accesses. It complements the existing NAS Parallel Benchmark suite. The benchmark involves the solution of a stylized heat transfer problem in a cubic domain, discretized on an adaptively refined, unstructured mesh.

1 Introduction

The NAS Parallel Benchmarks (NPB) [1, 2] were originally formulated to measure the performance of high-performance computer systems, especially parallel machines, when applied to computational problems of importance to the scientific community in general, and to NASA in particular. Despite the relatively limited scope of the eight problems that made up NPB, the benchmarks became quite popular and have been widely used by researchers, computer vendors/buyers, and software tool developers. Since their initial release as paper-and-pencil specifications in 1991, and as source code implementations (in MPI) in 1995 [2], it has become increasingly clear that the NPB problems lack in the area of irregular and dynamically changing memory accesses, which may defeat the hardware and software support for memory traffic in modern computer architectures. The original NPB featured applications with

[*] Employees of Computer Sciences Corporation. This work was supported by the NASA Computing, Information, and Communications Technology (CICT) program, and performed under Task Order A61812D (ITOP Contract DTTS59-99-D-00437/TO #A61812D) awarded to Advanced Management Technology Incorporated.

[†] NASA Advanced Supercomputing Division

[‡] George Washington University

non-varying, (mostly) fixed-stride memory access, which can be exploited by compilers and specialized hardware to reduce the cost of memory traffic. Not all applications of importance to NASA have such simple, predictable memory accesses, however, and it was found in a study by Oliker and Biswas [8] that some strongly dynamic applications fare poorly when implemented on certain widely used parallel computers. This situation is exacerbated when the programming environment selected offers limited control over data placement and granularity of data traffic, as is often the case with distributed shared memory systems.

The new benchmark specified here, to be incorporated in the NPB suite, aims to provide a standardized method for gauging the performance of computer systems when running scientific applications whose memory access patterns are irregular and continually changing. The motivation for the specific design choices that led to the problem specification were presented in a previous report [3].

1.1 Governing equation

The mathematical model for the heat transfer problem is as follows:

$$T_t + \mathbf{v} \cdot \nabla T = \epsilon \nabla^2 T + S(\mathbf{x}, t)\,, \tag{1}$$

where T is temperature; $\mathbf{v}$ is the flow velocity; ϵ is the heat diffusion coefficient; t is time; and $S(\mathbf{x}, t)$ is a source term defined by

$$S(\mathbf{x}, t) = \begin{cases} \cos(\frac{\pi}{\alpha}||\mathbf{x} - \mathbf{x_0} - \mathbf{v}t||) + 1 & \text{if} \quad ||\mathbf{x} - \mathbf{x_0} - \mathbf{v}t|| \leq \alpha \\ \\ 0 & \text{if} \quad ||\mathbf{x} - \mathbf{x_0} - \mathbf{v}t|| > \alpha \end{cases} \tag{2}$$

Here $|| \cdot ||$ signifies the Euclidian norm. The location vector $\mathbf{x}$ is defined by (x, y, z), and $\mathbf{x_0}$ is the initial location of the center of the source. The initial temperature distribution on the domain is zero, and the Dirichlet boundary conditions are fixed at zero as well. The prescribed velocity field $\mathbf{v} = (u, v, w)$ is uniform and constant, and equals the speed of the source.

Thus, it is known explicitly when and where mesh refinement and coarsening are required. Fine mesh cells are prescribed where large temperature gradients exist, i.e. near the traveling source. After the source has passed a certain region, high resolutions are no longer needed there, and grid coarsening is applied. Even though the elements are rectangular, the circular shape of the source results in a nonuniform grid, see Section 4.

1.2 Temporal discretization

Temporal discretization is based on an operator splitting technique developed by Maday, Patera, and Rønquist [5]. The two-step time splitting scheme for time integration of the governing equation is:

$$\begin{aligned} \hat{T}^{n+1} &= \mathrm{RK}_4(-\mathbf{v} \cdot \nabla T^n + S(\mathbf{x}, t)) && (3) \\ \frac{T^{n+1} - \hat{T}^{n+1}}{\Delta t} &= \epsilon \nabla^2 T^{n+1} && (4) \end{aligned}$$

The explicit, fourth-order Runge-Kutta method, RK_4, in Eq. (3) is defined as follows.

$$
\begin{aligned}
\hat{T}^{n+1} &= T^n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\
k_1 &= \Delta t(-\mathbf{v} \cdot \nabla T^n + S(\mathbf{x}, t)) \\
k_2 &= \Delta t(-\mathbf{v} \cdot \nabla (T^n + \frac{k_1}{2}) + S(\mathbf{x}, t + \frac{\Delta t}{2})) \\
k_3 &= \Delta t(-\mathbf{v} \cdot \nabla (T^n + \frac{k_2}{2}) + S(\mathbf{x}, t + \frac{\Delta t}{2})) \\
k_4 &= \Delta t(-\mathbf{v} \cdot \nabla (T^n + k_3) + S(\mathbf{x}, t + \Delta t))
\end{aligned}
\tag{5}
$$

The Euler Implicit method used for the time advancement of the diffusion term (4) leads to a large system of equations after spatial discretization. It is solved using a Preconditioned Conjugate Gradient (PCG) method, described in Section 3.3.

2 Grid system

For spatial discretization of the problem we use the Spectral Element Method (SEM) [9], a high order technique for solving partial differential equations. It is a hybrid of spectral and finite element methods, combining their respective high accuracy and geometrical flexibility. In this benchmark we use a non-conforming, adaptively refined grid that changes as the heat source travels through the domain. Details about grid adaptation are presented in Section 4. Non-conformity of the grid is restricted; a grid cell–or *element*–is refined by dividing all its sides by two, and if two elements have a common face or edge, the ratio of the lengths of their sides is at most two. Hence, if two elements share a face, the faces are either equal in size, or one is four times the size of the other. Simple examples of a conforming and a non-conforming grid are shown in Figure 1.

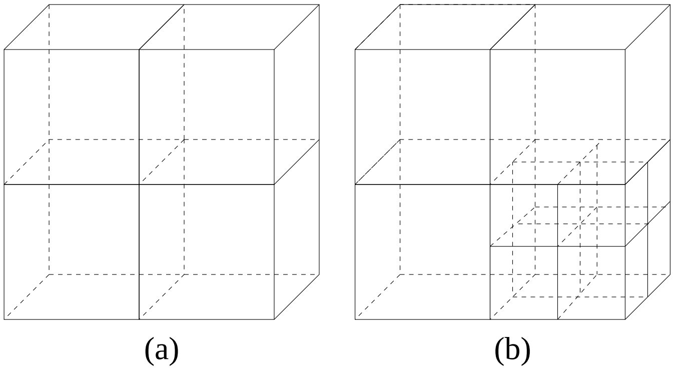


Figure 1: (a) Conforming grid, (b) Non-conforming grid

2.1 Collocation points

Each element is covered by a Cartesian product of $(N+1)^3$ collocation points, where N is the order of the Legendre-based polynomials used to evaluate the interpolated solution on the whole element. For this benchmark N is fixed at four for all elements. There are collocation points on all element boundaries. Their locations within each element are those corresponding to Gauss-Lobatto-Legendre (GLL) quadrature, see Section 3.2. If the grid has K elements, the total number of collocation points is $(N+1)^3K = 125K$. Where elements meet, some of the collocation points belonging to the different elements coincide in space—they are not topologically unique. The collocation points are used for the advancement of the convection as well as the diffusion term. For each such point an elemental contribution is computed, as defined in Section 3, which can be evaluated locally using just the collocation points within a single element.

2.2 Grid points

The purpose of SEM is to compute a single solution value for each unique point in space contained in the grid system (a *grid point*). Each such value is called a *degree of freedom*. There are always more collocation points than grid points, except for a trivial grid containing a single element. For example, a conforming grid of $M \times M \times M$ elements has $(M(N+1))^3$ collocation points and $(MN+1)^3$ grid points. For a non-conforming grid the number of degrees of freedom depends on the particular grid configuration. Each grid point coincides with one or more collocation points, but not all collocation points coincide with a grid point. Where a coarse and fine element meet along an edge or face, only the collocation points on the edge or face of the finer element are included as grid points. Figure 2 shows an example of the configuration of grid points and collocation points for a simple grid.

2.3 Mappings

In the course of the solution process it is necessary to map values from collocation points to grid points, and vice versa. The former is called a *gather*, and the latter a *scatter*. These operations form the basis of the Mortar Element Method (MEM) [6]. Both gather and scatter are the identity operation (copy) for points in element interiors, and for points on grid boundaries. In that case a grid point coincides with a single collocation point. Below we define the mappings for all other points in the grid, all of which are on interfaces between elements.

2.3.1 Scatter

CONFORMING FACE INTERIOR, EDGE

If all grid points in the interior of a face of an element coincide with collocation points of that element, the face interior is conforming. If all grid points on an edge of an element coincide with collocation points of that element, the edge is conforming.

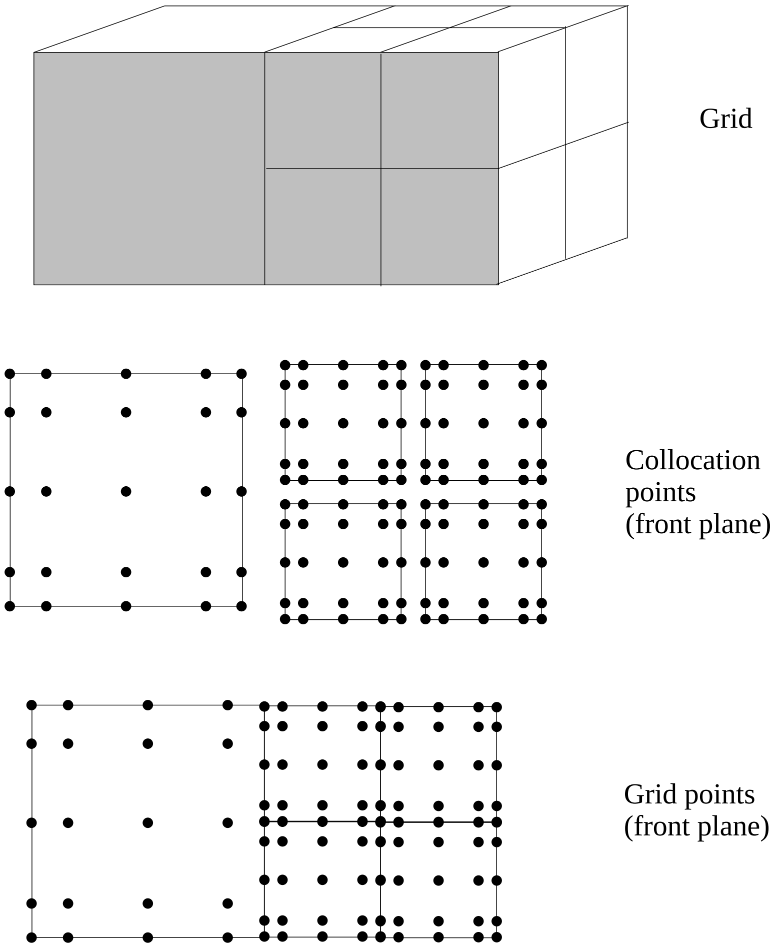


Figure 2: Collocation and grid points for simple grid

The scattering consists of copying values at grid points to the coincident collocation points in the conforming loci defined above.

NON-CONFORMING FACE

A face of an element is non-conforming if neither the face interior nor its four edges are conforming.

The scattering of values satisfies a jump condition that minimizes the difference in values across the interface in an integral sense [6]. This is accomplished as follows. Let the grid points that lie on the face of the coarse element be indicated by $(m,k), m = 1,\ldots,2N+1, k = 1,\ldots,2N+1$, and let the quantity to be scattered be called ϕ. The scattered values Φ_{ij} at collocation points on the coarse-element side of the interface are:

$$\Phi_{ij} = \sum_{m=1}^{2N+1} \sum_{k=1}^{2N+1} Q_{im} Q_{jk} \phi_{mk}\,, \quad i,j = 1,\ldots,N+1\,. \tag{6}$$

This scatter operation to the collocation points of a single element face can be viewed as a collection of one-dimensional scatterings in two respective coordinate directions. For the derivation of the one-dimensional scatter matrix Q we refer to [6]. Its elements (transposed) are listed in Table 1. It follows from the shape of Eq. 6 that the order in which the one-dimensional scatterings are carried out does not matter.

NON-CONFORMING EDGE

An edge of an element face is non-conforming if it is adjacent to a conforming face interior, but is itself not conforming. In the scattering to a non-conforming edge, either $i\&m$ or $j\&k$ in Eq. 6 are fixed. The fixed value is 1 or $N+1$ for either i or j, and 1 or $2N+1$ for the other fixed index, depending on the location of the edge.

Q_{ij}	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
$j=1$	1.0	-0.1772843218615690	9.375E-2	Q_{29}	0.0
$j=2$	0.0	0.7152146412463197	-0.2285757930375471	Q_{28}	0.0
$j=3$	0.0	0.4398680650316104	0.2083333333333333	Q_{27}	0.0
$j=4$	0.0	8.333333333333333E-2	0.3561799597042137	Q_{26}	0.0
$j=5$	0.0	0.0	0.140625	Q_{25}	0.0
$j=6$	0.0	-4.854797457965334E-2	Q_{34}	Q_{24}	0.0
$j=7$	0.0	-5.891568407922938E-2	Q_{33}	Q_{23}	0.0
$j=8$	0.0	8.333333333333333E-2	Q_{32}	Q_{22}	0.0
$j=9$	0.0	-3.700139242414530E-2	Q_{31}	Q_{21}	1.0

Table 1: One-dimensional scatter matrix Q_{ij} for $N=4$

The global scattering operation from the grid points to *all* collocation points can be written as: $\mathbf{R}_c \leftarrow \theta\mathbf{R}_g$, where θ is the global scatter matrix.

2.3.2 Gather

The global gathering of residuals from the collocation points to all grid points simply involves the transpose of the global scatter matrix: $\mathbf{R}_g \leftarrow \theta^T\mathbf{R}_c$ [7]. Use of the (θ^T, θ)-pair is critical for the solution of the linear system of equations for advancing the diffusion term, since it guarantees symmetry and positive definiteness of the overall discretization matrix.

The gathered value of the residual at any grid point consists of the sum of all contributions from neighboring elements, i.e. the sum of the results of all relevant local gathering operations, to be defined below. This is termed *direct stiffness summation*.

CONFORMING FACE INTERIOR, EDGES

On conforming face interiors and edges, the value at a grid point is the sum of the residuals at all coincident collocation points.

NON-CONFORMING FACE

On a non-conforming face, the contribution δr_{mk} of the elemental residuals Φ at collocation points to the residual at a grid point (m, k) that lies on the same face is:

$$\delta r_{mk} = \sum_{i=1}^{N+1} \sum_{j=1}^{N+1} Q_{mi}^T Q_{kj}^T \Phi_{ij} = \sum_{i=1}^{N+1} \sum_{j=1}^{N+1} Q_{im} Q_{jk} \Phi_{ij} \, , \quad m, k = 1, \ldots, 2N+1 \, . \tag{7}$$

NON-CONFORMING EDGE

For the computation of elemental contributions to the residuals at grid points on non-conforming edges we again use Eq. 7, but either $i \& m$ or $j \& k$ are fixed. The fixed value is 1 or $N+1$ for either i or j, and 1 or $2N+1$ for the other index, depending on the location of the edge.

3 Spatial discretization

Within each element the dependent variable, temperature T, is approximated by:

$$T(r,s,t) \approx \sum_{i=1}^{N+1} \sum_{j=1}^{N+1} \sum_{k=1}^{N+1} T_{ijk} h_i(r) h_j(s) h_k(t) \tag{8}$$

$$h_p(\xi) = \begin{cases} 1 & \text{if} \quad \xi = \xi_p \\ -\frac{1}{N(N+1)L_N(\xi_p)} \frac{(1-\xi^2)L'_N(\xi)}{\xi - \xi_p} & \text{if} \quad \xi \neq \xi_p \end{cases} \tag{9}$$

T_{ijk} is the value of the temperature at collocation point (i, j, k) within the element; ξ_p is the p^{th} GLL collocation point; r, s, and t are local coordinates that sweep out the cubical element; mapping from global to local coordinates is as follows:

$$r = \frac{x - x_1}{x_2 - x_1} \times 2 - 1, \qquad s = \frac{y - y_1}{y_2 - y_1} \times 2 - 1, \qquad t = \frac{z - z_1}{z_2 - z_1} \times 2 - 1 \tag{10}$$

where (x_1, y_1, z_1) and (x_2, y_2, z_2) are the coordinates of the diagonally opposite corner points of the element with the smallest and largest coordinate values, respectively. The interpolating polynomial h, which is a function of the N^{th} order Legendre polynomial, L_N, and its derivative, L'_N, has the property that $h_p(\xi_q) = \delta_{pq}$ for $p, q \in \{1, ..., N+1\}$, where δ_{pq} is the Kronecker delta. We also note that $L'_N(\xi_q) = 0$ for $q \in \{2, ..., N\}$.

3.1 Advancing the convection term

At the start of each time step the values of T at the grid points are scattered to the collocation points. Eq. 5 is used to advance the convection term using only information local to the element. We use the spatial discretization of T, defined in Eq. 8, to compute the element-wise approximation to its directional derivative at collocation point (i, j, k):

$$(\mathbf{v} \cdot \nabla T)_{ijk} \approx \sum_{q=1}^{N+1} \left(v_x \frac{2}{L_1} D_{iq} T_{qjk} + v_y \frac{2}{L_2} D_{jq} T_{iqk} + v_z \frac{2}{L_3} D_{kq} T_{ijq} \right) , \tag{11}$$

where L_i is the size of the element in the i^{th} coordinate direction, and D_{ij} is defined as $\frac{dh_j}{d\xi}(\xi_i)$. The values of D are listed in Table 2.

D_{ij}	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$
$i = 1$	-5.0	6.756502488724238	-2.666666666666667	-D_{52}	-D_{51}
$i = 2$	-1.240990253030982	0.0	1.745743121887939	-D_{42}	-D_{41}
$i = 3$	0.375	-1.336584577695453	0.0	-D_{32}	-D_{31}
$i = 4$	-0.2590097469690172	0.7637626158259734	-D_{23}	-D_{22}	-D_{21}
$i = 5$	0.5	-1.410164177942427	-D_{13}	-D_{12}	-D_{11}

Table 2: Derivatives D_{ij} of one-dimensional elemental basis functions h_j for $N = 4$

3.2 Advancing the diffusion term

The variational form of Eq. (4) is:

$$(\frac{T^{n+1} - \hat{T}^{n+1}}{\Delta t}, v) \quad = \quad (-\epsilon \nabla T^{n+1}, \nabla v) \, , \tag{12}$$

where (·,·) represents the L^2 inner product, and v is a test function.

We select for the basis of the discrete space of test functions and solutions a set of functions $V_h \equiv \{v(p)_h\}$, each corresponding to exactly one grid point $\mathbf{x}_p$. The basis function corresponding to $\mathbf{x}_p$ is obtained by the residual gathering operation θ^T, applied to the elemental basis functions $u(r,s,t)_{ijk} = h_i(r) h_j(s) h_k(t)$ of all elements that contain $\mathbf{x}_p$. This operation guarantees continuity of the solution at element vertices and at conforming interfaces, as well as a minimum jump—in an integral sense—across non-conforming element interfaces [6].

Hence, we can compute elemental residual contributions Φ_{ijk} for each collocation point in each element and compute the total residual at a grid point by applying the gather operation θ^T (see Section 2.3.2). We define:

$$\Phi_{ijk} \equiv (\frac{T^{n+1} - \hat{T}^{n+1}}{\Delta t}, u_{ijk}) - (-\epsilon \nabla T^{n+1}, \nabla u_{ijk}) \, , \tag{13}$$

where u_{ijk} is the elemental basis function for collocation point (i, j, k).

GLL quadrature is used to calculate the integrals in Eq. 13. One-dimensional GLL quadrature of a function τ is defined as follows:

$$\int_{\xi=-1}^{1} \tau(\xi) d\xi \approx \sum_{i=1}^{N+1} \rho_i \tau(\xi_i) \, . \tag{14}$$

GLL quadrature locations ξ_i and the corresponding weights ρ_i are given in Table 3. Multi-dimensional versions of GLL quadrature are derived by applying Eq. 14 successively in the respective coordinate directions.

i	1	2	3	4	5
ξ_i	-1.0	-0.6546536707079771	0.0	0.6546536707079771	1.0
ρ_i	1/10	49/90	32/45	49/90	1/10

Table 3: GLL collocation points ξ_i with associated weights ρ_i for $N = 4$.

Substituting Eq. 8 into Eq. 13 and applying GLL quadrature yields:

$$
\begin{aligned}
\Phi_{ijk} = \Bigg[\sum_{l=1}^{N+1} \sum_{m=1}^{N+1} \sum_{o=1}^{N+1} |J| \Bigg(& \epsilon(\frac{2}{L_1})^2 \rho_m \rho_o \sum_{q=1}^{N+1} \rho_q D_{qi} D_{ql} \delta_{jm} \delta_{ko} + \\
& \epsilon(\frac{2}{L_2})^2 \rho_l \rho_o \sum_{q=1}^{N+1} \rho_q D_{qj} D_{qm} \delta_{il} \delta_{ko} + \\
& \epsilon(\frac{2}{L_3})^2 \rho_l \rho_m \sum_{q=1}^{N+1} \rho_q D_{qk} D_{qo} \delta_{il} \delta_{jm} + \frac{\rho_i \rho_j \rho_k \delta_{il} \delta_{jm} \delta_{ko}}{\Delta t} \Bigg) \Bigg] T_{lmo}^{n+1} - \\
& \left\{ \frac{1}{\Delta t} \sum_{l=1}^{N+1} \sum_{m=1}^{N+1} \sum_{o=1}^{N+1} |J| \quad \rho_i \rho_j \rho_k \delta_{il} \delta_{jm} \delta_{ko} \right\} \hat{T}_{lmo}^{n+1} \,,
\end{aligned}
\tag{15}
$$

where $|J|$ is proportional to the volume of the element, i.e. $|J| = \frac{L_1 L_2 L_3}{8}$. Upon gathering the residual contributions at the grid points and setting the total residual equal to zero, we symbolically obtain the following system of equations:

$$
\theta^T (A \theta T_g^{n+1} - b) = 0 \,, \tag{16}
$$

where $b = B \hat{T}_c^{n+1}$. $\hat{T}_c^{n+1}$ and T_g^{n+1} signify the temperature at the collocation points after advancement of the convection term, and at the grid points after advancement of the diffusion term, respectively. The matrix A represents the assembled elemental discretization matrix of the term in square brackets in Eq. 15 , which is symmetric, positive definite. Obviously, $\theta^T A \theta$ is symmetric as well. Since the scatter matrix θ is of full rank, $\theta^T A \theta$ is also positive definite. B represents the assembled elemental discretization matrix of the term in curly braces in Eq. 15.

3.3 Conjugate Gradient method

We use the Preconditioned Conjugate Gradient (PCG) method to solve Eq. 16.

The initial guess T_0 is based on $\hat{T}_c^{n+1}$: $T_0 = \sigma^T \hat{T}_c^{n+1}$. The σ^T operator is applied to the solution on collocation points, $\hat{T}_c^{n+1}$, to obtain T_0 on grid points. It is defined as follows: on element interfaces the solution at a grid point is computed by averaging values (arithmetic mean) at the coincident collocation points of all neighboring *fine* elements. If a grid point is on a conforming interface, all neighboring elements are considered fine elements. At points not on element interfaces the grid point value is set to the value at the coincident collocation point.

It should be noted that although both σ^T and θ^T constitute gather operations, the latter cannot be used to compute actual solution values. It is specifically derived to gather residuals, such that the solution of the resulting system of equations satisfies a minimum jump condition when the aggregate residual vanishes. This gather operation does not correspond to any kind of projection or averaging, and hence cannot be used to gather solution values.

$$\begin{gathered}
\text{initialization:} \\
T_0 = \sigma^T \hat{T}_c^{n+1}; \\
r_0 = \theta^T B \hat{T}_c^{n+1} - \theta^T A \theta T_0 = \theta^T (B \hat{T}_c^{n+1} - A \theta T_0); \\
q_0 = P^{-1} r_0; \\
p_0 = q_0;
\end{gathered}$$

$$\begin{gathered}
\text{iteration:} \\
a_m = (q_m, r_m)/(p_m, \theta^T A \theta p_m); \\
T_{m+1} = T_m + a_m p_m; \\
r_{m+1} = r_m - a_m \theta^T A \theta p_m; \\
q_{m+1} = P^{-1} r_{m+1}; \\
b_m = (q_{m+1}, r_{m+1})/(q_m, r_m); \\
p_{m+1} = r_{m+1} + b_m p_m;
\end{gathered}$$

where m refers to iteration number, r_m is the residual, p_m the search direction, P is the preconditioner, q_m is a vector associated with the preconditioning and (·,·) is the usual discrete inner product. Note that the dependent variable T_m does not have to be updated within the PCG iterative loop. It is sufficient to accumulate the quantities $a_m p_m$ at each grid point and scatter their sum to the collocation points after the PCG iterations have been completed, where it gets added to $\hat{T}_c^{n+1}$. The stop criterion for the iterations is given in Section 5.

A diagonal preconditioner P is used. See the Appendix for how to compute it.

$$P = Diag(\theta^T A \theta)\,. \tag{17}$$

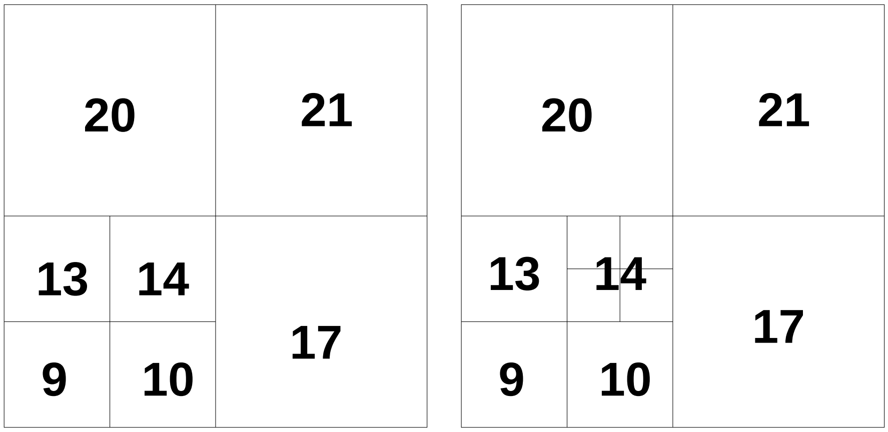
4 Mesh adaptation

The mesh adaptation is subject to two restrictions:

1. the maximum number of levels of refinement is constant for each problem size (Class).

2. the difference in size between any two elements that have a face or edge in common is at most a factor of eight (one level of refinement).

Figures 3 (a) and (b) show a valid and an invalid grid, respectively. In Fig. 3 (b), the refinement of element 14 results in a situation where elements 17, 20 and 21 have two levels of refinement difference with some of their neighboring elements.



(a) (b)

Figure 3: (a) 2D front view of a valid 3D mesh, (b) Invalid mesh that violates criterion 2

4.1 Refinement and coarsening

Both refinement and coarsening are isotropic. For refinement, an element is split evenly in the x, y and z directions into eight child elements. For coarsening, the eight children are merged into one element, the parent. The coarsening is performed only when all eight child elements are identified to be coarsened. Both refinement and coarsening must obey the two adaptation criteria. If an element is marked for refinement (see Section 4.3), but can not be refined due to restriction 2, its neighboring elements which prohibit the refinement will be marked to be refined first. Any element not marked for refinement will be coarsened, if possible.

If a refinement will result in a difference of more than one level of refinement between neighboring elements, additional refinements will be applied as needed until all required refinements are obtained. For example, in the situation of Fig. 3 (b), elements 17, 20 and 21 will be refined first to allow the refinement of element 14.

In Fig. 4, the coarsening of elements 61, 62, 64 and 65 may be desired. However, the coarsening will result in element 17 having two levels of refinement difference with element 61, with which it shares an edge. Thus, the coarsening is cancelled. If all elements in Fig. 4 are to be coarsened, this is accomplished in two steps. The merge of elements 12, 13, 16, and 17 is performed first, then the other merge is performed. Therefore, no invalid grid will appear at any time.

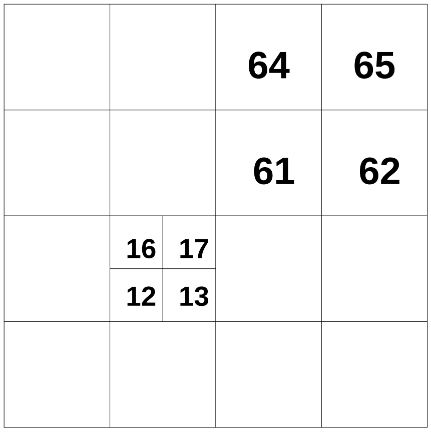


Figure 4: 2-D front view of a 3-D mesh; coarsening of elements 61, 62, 64 and 65 alone is not allowed.

4.2 Grid mapping

After a new grid has been generated, the old solution must be transferred to it. This is accomplished by mapping collocation point values from coarse to fine elements through multidimensional interpolation. It consists of sampling the solution T^f at the collocation points of the finer elements using Eq. 8. For evaluation of function values at just one level of refinement difference we tabulate $h_i(\xi_j)$, where h_i is the i^{th} interpolation polynomial pertaining to the coarser element ($i = 1, \ldots, N$), and ξ_j is the j^{th} collocation point of the union of finer elements contained in the coarse element. See Table 4. That is, collocation points 1 through $2N + 1$ are contained in the two adjacent fine elements that span the coarse element in a certain coordinate direction, i.e. 1 through $N + 1$ in the first, and $N + 1$ through $2N + 1$ in the second. If interpolation across more than one level of refinement needs to be performed, the interpolation can be defined recursively, using Table 4 repeatedly, or directly, by sampling T at the appropriate collocation point locations.

$h_{ij} = h_i(\xi_j)$	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
$j = 1$	1.0	0.0	0.0	h_{29}	h_{19}
$j = 2$	0.3385078435248143	0.7898516348912331	-0.1884018684471238	h_{28}	h_{18}
$j = 3$	-0.1171875	0.8840317166357952	0.3125	h_{27}	h_{17}
$j = 4$	-7.065070066767144E-2	0.2829703269782467	0.9026875827328380	h_{26}	h_{16}
$j = 5$	0.0	0.0	1.0	h_{25}	h_{15}
$j = 6$	4.984442584781999E-2	-0.1648516348912333	h_{34}	h_{24}	h_{14}
$j = 7$	0.0390625	-0.1184067166357950	h_{33}	h_{23}	h_{13}
$j = 8$	-3.198728299067715E-2	9.202967302175333E-2	h_{32}	h_{22}	h_{12}
$j = 9$	0.0	0.0	h_{31}	h_{21}	h_{11}

Table 4: Interpolation coefficients from coarse to fine elements (one level of refinement)

The mapping from fine to coarse elements is always defined recursively. The interpolation formula Eq. 8 for the fine element is used to evaluate T at the collocation points of the element at the next coarser level that fall within the fine element. For those coarse-element collocation points that are on the boundary between fine elements it does not matter which fine element contributes the interpolated values. The solution process guarantees that coincident fine-element collocation point solution values are identical at the end of each time step. We use a variation of Eq. 8 to compute the value of T^c at the coarse-element collocation points from the fine-element values T^f:

$$T^c_{ijk} = \sum_{o=1}^{2N+1} \sum_{p=1}^{2N+1} \sum_{q=1}^{2N+1} T^f_{opq} \hat{h}_o(\xi_i) \hat{h}_p(\xi_j) \hat{h}_q(\xi_k) \,, \quad i,j,k = 1,\ldots,N \,, \tag{18}$$

where $\hat{h}_i$ is one of $2N+1$ interpolation polynomials of two adjacent fine elements within the coarse element, ξ_j is the j^{th} collocation point of the coarse element, and T^f_{ijk} is the value of the temperature at collocation point (i,j,k) of the union of all eight adjacent fine elements making up the coarse element. Since coincident collocation points at fine-element boundaries share the same value of T^f, we count them as the same point in Eq. 18. We list the values of $\hat{h}_i(\xi_j)$ in Table 5 below.

$\hat{h}_i(\xi_j)$	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$
$i=1$	1.0	-0.1179652785083428	0.0	0.0	0.0
$i=2$	0.0	0.5505046330389332	0.0	0.0	0.0
$i=3$	0.0	0.7024534364259963	0.0	0.0	0.0
$i=4$	0.0	-0.1972224518285866	0.0	0.0	0.0
$i=5$	0.0	6.222966087199998E-2	1.0	6.222966087199998E-2	0.0
$i=6$	0.0	0.0	0.0	-0.1972224518285866	0.0
$i=7$	0.0	0.0	0.0	0.7024534364259963	0.0
$i=8$	0.0	0.0	0.0	0.5505046330389332	0.0
$i=9$	0.0	0.0	0.0	-0.1179652785083428	1.0

Table 5: Interpolation coefficients from fine to coarse elements

4.3 Adaptation criteria, frequency

Any element that has a nonzero overlap with the heat source gets refined. The refinement procedure is performed recursively until all elements that have nonzero overlap with the heat source have the minimal element size of that class. Overlap is considered nonzero if the point of the element closest to the center of the heat source is at a distance of less than α, where α is the radius of the heat source. Other elements are refined up to the *minimum* level required to satisfy mesh restriction 2 (page 10). Due to the movement of the heat source, some elements may no longer have overlap with the heat source. They will be coarsened recursively until all possible coarsenings are done, obeying restriction 2. The first adaptation takes place before computations start, i.e., before the first time step. Subsequently, after a certain prescribed number of time steps (differs for the different benchmark classes, see Table 6) the grid gets re-adapted using the same criteria as above.

5 Verification

The initial grid has one element, $[0, 1]^3$, which makes up the whole domain. The center of the heat source is located at $\mathbf{x}_0 = (\frac{3}{7}, \frac{2}{7}, \frac{2}{7})$ at the beginning of the computation, i.e. $t = 0$. It moves at a speed of $\mathbf{v} = (3, 3, 3)$. The size of the source is defined by α. We set the heat diffusion coefficient ϵ to 0.005 for all cases. The number of PCG iterations is always ten, and the adaptation frequency is once every five time steps. The correctness of the result is verified by the integral of the solution $\int_\Omega T d\Omega$ at the end of the time stepping. It is computed as follows:

$$\int_\Omega T d\Omega = \sum_{\textit{all elements}} \sum_{k=1}^{N+1} \sum_{j=1}^{N+1} \sum_{i=1}^{N+1} |J| \rho_k \rho_j \rho_i T_{ijk} \, . \tag{19}$$

We list the parameters and verification values for all benchmark classes in Table 6. nt is the number of time steps, nl is the maximum number of levels of refinement. The size of the time step Δt is related to the maximum number of levels of refinement as follows: $\Delta t = 0.04 * 2^{-nl}$.

For convenience we also list the number of elements, ne, at the end of the computation. A solution is considered incorrect if the absolute value of the difference between computed integral and verification value, divided by the verification value, exceeds the error threshold of 10^{-8}.

Class	nt	nl	α	$\int_\Omega T d\Omega$	ne
S	50	4	0.04	1.890013110962E-3	246
W	100	5	0.06	2.569794837076E-5	526
A	200	6	0.076	8.939996281443E-5	2038
B	200	7	0.076	4.507561922901E-5	7841
C	200	8	0.067	1.544736587100E-5	31641
D	250	10	0.046	1.577586272355E-6	506297

Table 6: UA verification values

6 Summary

We now give an overview of the steps in the completion of the benchmark.

1. refine the initial grid based on the refinement criteria.
2. set the solution at all collocation points to zero.
3. compute the source at all collocation points, using Eq. 2.
4. advance the solution (convection term) at all collocation points, using Eqs. 5 and 11.
5. set T_0 for PCG, using the operation σ^T, described in Section 3.3.
6. advance the solution (diffusion term) at all grid points, using PCG described in Section 3.3.
   (a) compute the element-wise residual contribution defined in Eq. 13.
   (b) collect the residual at the grid points, using the gather operation θ^T.
   (c) scale the residual, using diagonal preconditioning.
   (d) scatter the scaled residual to the collocation points.
   (e) apply the element-wise diffusion operator (Eq. 13) to the scaled residual.
   (f) gather the resulting residual increments at the grid points.
   (g) accumulate the temperature updates at grid points.
   (h) compute a new residual vector.
   (i) scale the residual.
   (j) compute a new search direction.
   (k) return to step 6e as long as the stopping criterion has not been satisfied.
   (l) scatter the temperature update to collocation points.
7. if the step qualifies, refine/coarsen the grid based on the new location of the source and map the solution at the collocation points to the new set of collocation points.
8. return to step 3 as long as the total number of time steps has not been exceeded.

References

[1] Bailey, D.H., Barton, J., Lasinski, T. and Simon, H. (Eds.), The NAS Parallel Benchmarks, *NAS Technical Report RNR-91-002*, NASA Ames Research Center, Moffett Field, CA, 1991.

[2] Bailey, D.H., Harris, T., Saphir, W.C., Van der Wijngaart, R.F., Woo, A.C. and Yarrow, M., The NAS Parallel Benchmarks 2.0., *NAS Technical Report NAS-95-020*, NASA Ames Research Center, Moffett Field, CA, 1995.

[3] Feng, H, Van der Wijngaart, R.F. and Biswas. R., Design of Unstructured Adaptive (UA) NAS Parallel Benchmark Featuring Irregular, Dynamic Memory Accesses, *NAS Technical Report NAS-01-012*, NASA Ames Research Center, Moffett Field, CA, 2001.

[4] Maday, Y. and Patera, A.T., Spectral element methods for the Navier-Stokes equations. in Noor, A.K., editor, *State-of-the-art surveys in computational mechanics*, ASME, New York, 1988.

[5] Maday, Y., Patera, A.T. and Rønquist, E.M., An operator integration-factor splitting method for time-dependent problems: application to incompressible fluid flow, *Journal on Scientific Computing*, Volume: 5, Number: 4, Page: 263-292, 1990.

[6] Maday, Y., Mavriplis, C.A. and Patera, A.T., Non-Conforming Mortar Element Methods: Application to Spectral Discretizations, in *Domain Decomposition Methods*, Page: 392-418, 1989.

[7] Mavriplis, C.A., *Nonconforming Discretizations and a Posteriori Error Estimators for Adaptive Spectral Element Techniques*, PhD thesis, Massachusetts Institute of Technology, 1989.

[8] Oliker, L. and Biswas, R. Parallelization of a Dynamic Unstructured Algorithm Using Three Leading Programming Paradigms, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 11, No. 9, pages 931-940, 2000.

[9] Patera, A.T., A Spectral Element Method for Fluid Dynamics: Laminar Flow in a Channel Expansion, *Journal of Computational Physics*, Volume: 54, Number: 4, Page: 468-488, 1984.

Appendix: Simple method for computing the preconditioner

The system to be solved in the diffusion step can be written as $\theta^T A\theta\mathbf{x} = \theta^T\mathbf{b}$, where A is the assembled discretization matrix. Therefore, the diagonal preconditioner P satisfies the relation $P_{ij}^{-1} = (\theta^T A\theta)_{ij}\delta_{ij}$. However, $\theta^T A\theta$ is never computed explicitly, because it is inefficient to do so and because it would be too large to store. A straightforward way to obtain P_{ii} is to evaluate the matrix-vector products $\mathbf{u}_i = (\theta^T A\theta)\mathbf{e}_i$ for as many unit vectors $\mathbf{e}_i$ as there are grid points. The i^{th} entry of $\mathbf{u}_i$ is the i^{th} diagonal element of $(\theta^T A\theta)$. This procedure, which we call GENERAL, comprises three steps:

1. Calculate $\theta\mathbf{e}_i$, the resulting vector is the i^{th} column of θ.

2. Calculate $A\theta\mathbf{e}_i$. Each row of A relates to one elemental collocation point. The sum over all elements $\sum_{j=1}^{ntot} A_{pj}(\theta\mathbf{e}_i)_j$ is the p^{th} entry of $A\theta\mathbf{e}_i$. A is a block diagonal matrix, each block corresponding to one element, $ntot$ refers to the total number of collocation points. In conforming cases, each element of θ is either zero or one, and, hence, so is each element of vector $\theta\mathbf{e}_i$. The product $A\theta\mathbf{e}_i$ is a vector, whose p^{th} entry is nonzero iff $(\theta e_i)_p = 1$, and has the value A_{pp} in that case.

3. Calculate $(\theta^T A\theta\mathbf{e}_i)$. Since we only need the i^{th} entry of the resulting vector $(\theta^T A\theta\mathbf{e}_i)$, we only need to perform the inner product of the i^{th} row of θ^T with $A\theta\mathbf{e}_i$. Note that the i^{th} row of θ^T is just the i^{th} column of θ. In conforming cases, the resulting $(\theta^T A\theta\mathbf{e}_i)$ is vector $\mathbf{u}_i$, whose i^{th} entry is $\sum_{p\in G_i} A_{pp},\ G_i = \{g|(\theta e_i)_g = 1\}$.

In conforming cases, this procedure can be SIMPLIFIED.

1. Evaluate A_{mm} locally within each element.

2. Construct a vector $\mathbf{a}$ whose m^{th} entry equals A_{mm}.

3. Perform a global stiffness summation on $\mathbf{a}$: $\hat{\mathbf{a}} = \theta^T\mathbf{a}$.

4. Preconditioner P is obtained from $\hat{\mathbf{a}}$: $P_{ii}^{-1} = \hat{a}_i$.

To solve the overall problem we use a hybrid approach. The SIMPLIFIED procedure is used to compute the preconditioner for grid points in the interior of elements and on conforming boundaries. The GENERAL procedure is used on nonconforming boundaries.