

## **TACC Technical Report IMP-21**

### **Adaptive mesh refinement**

Victor Eijkhout\*

February 22, 2016

This technical report is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that anyone wanting to cite or reproduce it ascertains that no published version in journal or proceedings exists.

Permission to copy this report is granted for electronic viewing and single-copy printing. Permissible uses are research and browsing. Specifically prohibited are *sales* of any copy, whether electronic or hardcopy, for any purpose. Also prohibited is copying, excerpting or extensive quoting of any report in another work without the written permission of one of the report's authors.

The University of Texas at Austin and the Texas Advanced Computing Center make no warranty, express or implied, nor assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed.

\* [eijkhout@tacc.utexas.edu](mailto:eijkhout@tacc.utexas.edu), Texas Advanced Computing Center, The University of Texas at Austin

## **Abstract**

AMR stuff

The following IMP reports are available or under construction:

- IMP-00** The IMP Elevator Pitch
- IMP-01** IMP Distribution Theory
- IMP-02** The deep theory of the Integrative Model
- IMP-03** The type system of the Integrative Model
- IMP-04** Task execution in the Integrative Model
- IMP-05** Processors in the Integrative Model
- IMP-06** Definition of a ‘communication avoiding’ compiler in the Integrative Model
- IMP-07** Associative messaging in the Integrative Model (under construction)
- IMP-08** Resilience in the Integrative Model (under construction)
- IMP-09** Tree codes in the Integrative Model
- IMP-10** Thoughts on models for parallelism
- IMP-11** A gentle introduction to the Integrative Model for Parallelism
- IMP-12** K-means clustering in the Integrative Model
- IMP-13** Sparse Operations in the Integrative Model for Parallelism
- IMP-14** 1.5D All-pairs Methods in the Integrative Model for Parallelism (under construction)
- IMP-15** Collectives in the Integrative Model for Parallelism
- IMP-16** Processor-local code generation (under construction)
- IMP-17** The CG method in the Integrative Model for Parallelism (under construction)
- IMP-18** A tutorial introduction to IMP software (under construction)
- IMP-19** Report on NSF EAGER 1451204.
- IMP-20** A mathematical formalization of data parallel operations
- IMP-21** Adaptive mesh refinement (under construction)
- IMP-22** Implementing LULESH in IMP (under construction)

## 1 Story

## 2 Implementation

We use partial distributions a lot.

One. We set the same mask on the beta as on the output. Does that save any purpose?

```
beta_distribution = distribution_from_structure
(outvector->get_distribution(), pstruct);
if (outvector->has_mask())
    beta_distribution->add_mask(outvector->get_mask());
```

Two. The beta mask is unrelated to the input masks. For instance, in a broadcast the input has a mask, but the output and beta not.

## References